Price: $13.50

SDS 930 COMPUTER EXAMINER DIAGNOSTIC SYSTEM TECHNICAL MANUAL

SDS 90 00 97B September 1965

(Volume I)

SCIENTIFIC DATA SYSTEMS/1649 Seventeenth Street/Santa Monica, California/UP 1-0960

Printed in U.S.A.

INTRODUCTION

GENERAL INFORMATION

The SDS 930 Computer Examiner System is a complete diagnostic package that enables the operator to exercise and/or diagnose the memory and the computer main frame logic. For easy, convenient handling, the entire system is on one tape or on cards; its form is modular to facilitate expansion. The system consists of three main programs: Memory Diagnostic, Instruction Diagnostic, and The P and S Register Tester.

The Memory Diagnostic program tests memory, monitors the test for memory circuitry failure, and provides the user with pertinent information concerning the location and nature of the failure.

The Instruction Diagnostic tests the entire instruction set, exercises every gate and flip-flop pertaining to the instruction logic, monitors the tests, and provides pertinent information about the instruction in case of failure.

The P and S Register Tester determines whether the P Register and S Register operate properly over the entire addressing range.

Included in Section III under "Automatic Instruction Test Program" are detailed logic equation reference material, operating details, and detailed diagnostic constants for use in checking the multiply instruction (the Right Shift Adder) and the divide instruction (the BZO "eye-ball" circuit).

SYSTEM TAPE ORGANIZATION

The SDS 930 EXAMINER System is on one tape or on cards, as follows:

1. Memory Diagnostic
2. Instruction Diagnostic
3. P and S Register Tester

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 304001B

IDENTIFICATION:	Memory Diagnostic
AUTHOR:	Resnick/Mulholland, SDS
ACCEPTED:	August 23, 1965
COMPUTER CONFIGURATION:	SDS 930
PURPOSE:	To exercise memory with a checkerboard memory word pattern; to monitor memory for errors and aid in diagnosing memory failures.
SUBROUTINES REQUIRED:	None
STORAGE:	All of memory
TIMING:	Not applicable
SOURCE LANGUAGE:	SYMBOL 8 Assembler
LOADING PROCEDURE:	Self-loading, paper tape or cards. Clear memory before loading.

USE:

INTRODUCTION

The SDS 930 Examiner Memory Diagnostic Program tests any core memory, 4096 through 32,768 words. The two versions of this diagnostic are "16k or less" and "over 16k." Operationally, both programs are similar. In performing the test, the program writes a regular checkerboard pattern to generate noise throughout memory (1's in "positive" cores and 0's in "negative" cores*), then tests each memory word individually, inverts the word, tests again, and then restores the original word. If, during the testing process, the program detects an error, it can display the failing address and bit positions, aiding the operator in determining the circuitry that is malfunctioning. Also, the overflow light is turned on, and remains on, until the end of the half-cycle.

*Those cores inducing a positive polarity voltage into the sense winding are assumed to be positive cores, and the remainder negative cores.

USE: (cont.)

The program uses the breakpoint switches to provide various modes of program operation. See the next section, "Operating the Memory Diagnostic Program," for the mode discussion.

Special limits may be invoked at load time that may be used as a breakpoint option during execution. In the case of the "over 16k" version, new special limits may be invoked without reloading the program. During normal operation, the program tests all core memory above it ("16k or less" version) or all core other than resident 4k block of program ("over 16k" version). The program then moves itself to the upper end of core and tests all core below it ("16k or less" version), or all other core except the resident 4K block of program ("over 16k" version). When the program completes this cycle, it returns to lower core again and repeats the cycle with an "inverted" checkerboard pattern (0's in "positive" cores and 1's in "negative" cores). The ability to remain in one area and repeatedly test the remainder of core is a breakpoint option; the pattern will be inverted every second cycle in this mode. When special limits are used, the program will remain in the area that is not being tested. In the case of the "over 16k" version, the program moves only if the special start is in the first 4k block and/or the special limit is in the last 4k block of core. (If limits other than integer multiples of 128 are invoked, the program truncates the least significant 7 bits of the limits.)

OPERATING THE MEMORY DIAGNOSTIC PROGRAM

If an "over 16k" computer is to be tested, use the fill procedure* to load "over 16k" version over "16k or less" version (paper tape version).

1. During loading select breakpoint options as follows:

   Breakpoint No. 1

   SET - allows operator to change machine size, as displayed in the A Register (e.g., make 16k memory look as if it were only 12k or 8k, etc.).

   A(9-23) – location plus one of last word in memory

   B(9-23) – first location in memory, zero (do not change)

   RESET – no effect

*See SDS 930 Computer Reference Manual, 90 00 64.

USE: (cont.)

Breakpoint No. 2

No effect

Breakpoint No. 3

SET – change limits to be tested, in multiples of 128 (e.g., normal limits – 0 to 030000, special limits – 0200 to 020000*)

A(9-23) – location plus one of last word in memory

B(9-23) – first location of memory ("16k or less" version also has a NOP instruction in the A and B registers).

At this point the operator may change the lower and upper limits of memory by inserting the desired lower and upper limits in B and A, respectively. If special limits are to be used, the NOP instruction must be placed in these registers as well for the "16k or less" version.

RESET – No effect

Breakpoint No. 4

No effect

2. During running of the program select breakpoint options as follows:

Breakpoint No. 1

SET – repeat testing cycle in same area. Pattern will be inverted every second cycle.

RESET – normal testing cycle.

Breakpoint No. 2

SET – inhibit error halts

RESET – normal halts for errors.

Location of halts (both versions):

Low Memory – 0115
High Memory – 0X7715

---

*Leading 0 indicates octal number.

USE: (cont.)

Breakpoint No. 3

SET – special limits used (Do not set Bpt. 3 unless the program halts due to an error or to the setting of Bpt. 4).

RESET – normal limits used.

Breakpoint No. 4

SET – halt before each half-cycle. The A register contains the number of errors incurred. The operator may change the number at this time. Set Bpt. 3 if special limits are to be used.

In the "over 16k" version, if new machine size and/or special limits are desired, set Bpt. 1 (new machine size) and/or Bpt. 3 (new special limits) and branch to location 0404. Do this only if program is in low memory.

Location of halts (both versions):

Low Memory – 047
High Memory – 0X7647

RESET – No effect

CHECKERBOARD PATTERN MEMORY TEST

The checkerboard pattern consists of binary words of all 1's and words of all 0's. An example is:

Location	Contents
00000	77777777
00001	00000000
00002	77777777
00003	00000000
.	.
.	.
.	.

Shown in the form of X and Y address components, where a single digit represents a full word, this pattern is:

USE: (cont.)

Y→

X ↓	00	01	02	03	. . . 77
0000	1	0	1	0	0
0100	0	1	0	1	1
0200	1	0	1	0	0
0300	0	1	0	1	1
.					.
.					.
.					.
7700	0	1	0	1	1

The logical function that determines the pattern depends directly on the least significant bit of the X and Y sink addresses. Assuming that the address bits go from 9 through 23 in the C Register, the X sink bits and the Y sink bits are 15, 16, 17, and 21, 22, 23, respectively. This function is:

1. If address bit 17 is equal to address bit 23, the program fills the location with all 1-bits.
2. If address bit 17 is not equal to address bit 23, the program fills the location with all 0-bits.

Examples:

Address	Bit 17	Bit 23	Word Put in Location
01000	0	0	77777777
01001	0	1	00000000
00304	1	0	00000000
00503	1	1	77777777

The program alternates the above pattern at the end of each memory testing cycle. The change occurs each time the memory area tested goes from lower memory to upper memory. Thus, a complete check of memory requires two complete test cycles. Setting Breakpoint 3 causes the pattern to change between each half of the test cycle.

USE: (cont.)

The checkerboard pattern used in this program imposes the maximum noise on the output of the sense winding during reading or writing memory. The memory circuitry can detect such noise as information resulting in memory error.

ERROR CHECK

1. Error Halts in Program

   If Breakpoint No.2 is RESET when an error is detected, the program halts and displays the error. The program halts in one of several locations depending on program placement and memory size. When the program is testing upper core, the error halt is location 0115. When testing lower core, the error halt location is 0X7715.

2. Use of Error Indications

   Initially, if nothing loads into memory, check the power supply for proper power levels.

   If the program loads and then detects errors during the operation, the operator can suspect the memory circuitry or memory adjustment.

   When the operator determines the memory stack type – 4K, 8K and/or 16K – he can use the following to help determine the exact failure point.

X-Y Drivers/Sinks

When an error occurs, the C Register contains the error-word's location: C_9 through C_{23}. These address bits correspond to the X, Y drivers and sinks according to the following tables.

USE: (cont.)

X Sinks

	C18	C19	C20
SRX0 SWX0	0	0	0
SRX1 SWX1	0	0	1
SRX2 SWX2	0	1	0
SRX3 SWX3	0	1	1
SRX4 SWX4	1	0	0
SRX5 SWX5	1	0	1
SRX6 SWX6	1	1	0
SRX7 SWX7	1	1	1

Y Sinks

	C12	C13	C14
SRY0 SWY0	0	0	0
SRY1 SWY1	0	0	1
SRY2 SWY2	0	1	0
SRY3 SWY3	0	1	1
SRY4 SWY4	1	0	0
SRY5 SWY5	1	0	1
SRY6 SWY6	1	1	0
SRY7 SWY7	1	1	1

USE: (cont.)

X Drivers

	C11	C21	C22	C23
DX0	0	0	0	0
DX1	0	0	0	1
DX2	0	0	1	0
DX3	0	0	1	1
DX4	0	1	0	0
DX5	0	1	0	1
DX6	0	1	1	0
DX7	0	1	1	1
DX10	1	0	0	0
DX11	1	0	0	1
DX12	1	0	1	0
DX13	1	0	1	1
DX14	1	1	0	0
DX15	1	1	0	1
DX16	1	1	1	0
DX17	1	1	1	1

Y Drivers

	C10	C15	C16	C17
DY0	0	0	0	0
DY1	0	0	0	1
DY2	0	0	1	0
DY3	0	0	1	1
DY4	0	1	0	0
DY5	0	1	0	1
DY6	0	1	1	0
DY7	0	1	1	1
DY10	1	0	0	0
DY11	1	0	0	1

USE: (cont.)

DY12	1	0	1	0
DY13	1	0	1	1
DY14	1	1	0	0
DY15	1	1	0	1
DY16	1	1	1	0
DY17	1	1	1	1

Z Inhibit Drivers

The Z Inhibit Drivers are functions of bit position and memory stack size. Number the bit positions in memory as M0 (most significant) through M23 (least significant) and M24 (parity).

used with 4K, 8K or 16K stacks when $C_{10} C_{11} = 00$

$Z0d0 = \overline{M0}$

$Z1d0 = \overline{M1}$

.

.

.

$Z23d0 = \overline{M23}$

$Z24d0 = \overline{M24}$ (parity)

used with 8K or 16K stacks when $C_{10} C_{11} = 01$

$Z0d1 = \overline{M0}$

$Z1d1 = \overline{M1}$

.

.

.

$Z24d1 = \overline{M24}$ (parity bit)

used with 16K stacks only when $C_{10} C_{11} = 10$

$Z0d2 = \overline{M0}$

.

.

.

$Z24d2 = \overline{M24}$ (parity bit)

USE: (cont.)

used with 16K stacks only when $C_{10} C_{11} = 11$

$Z0d3 = \overline{M0}$

.

.

.

$Z24d3 = \overline{M24}$ (parity bit)

Sense Amplifiers

The Sense Amplifiers are functions of the following bit positions:

Md0 = M0

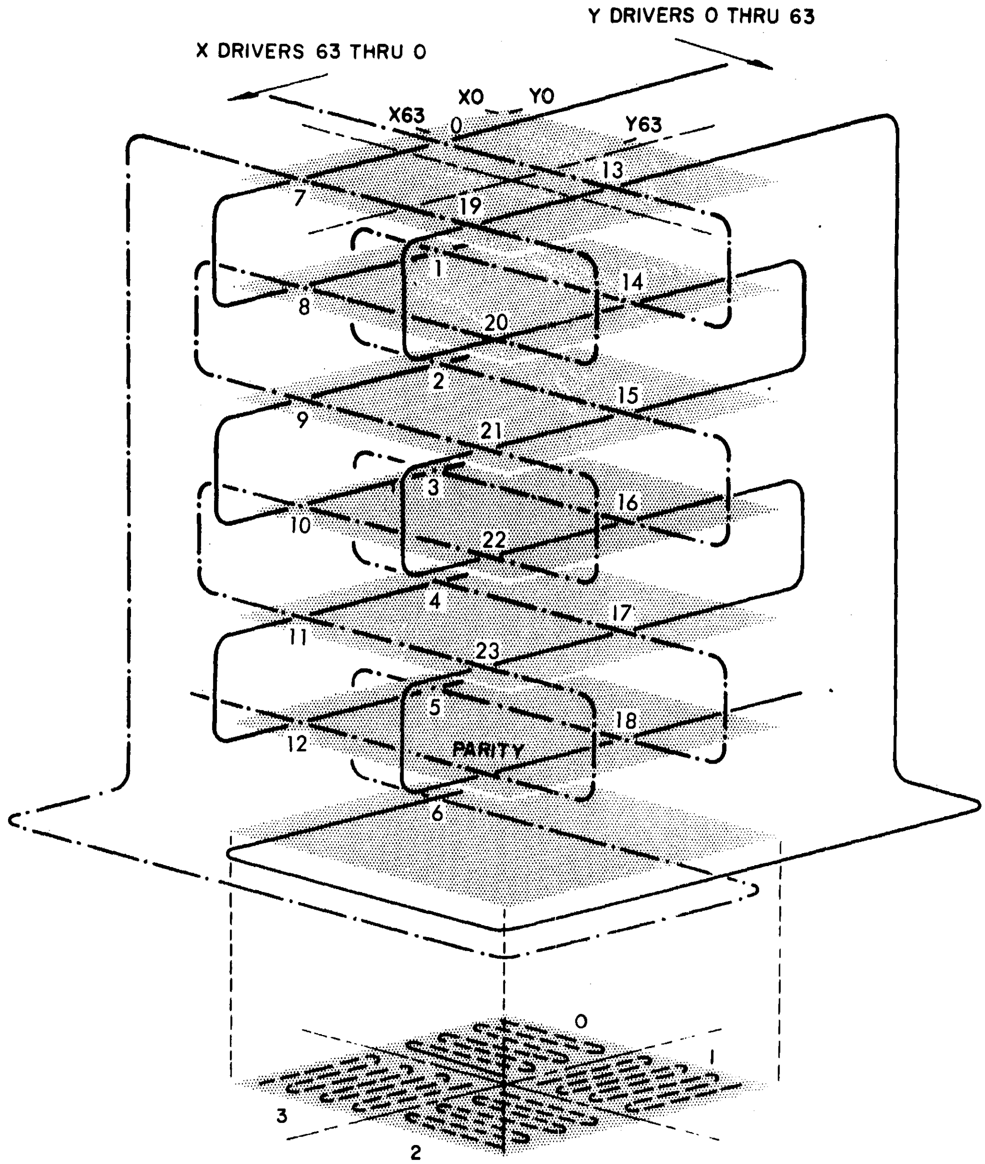
Md1 = M1

.

.

.

Md24 = M24 (parity bit)



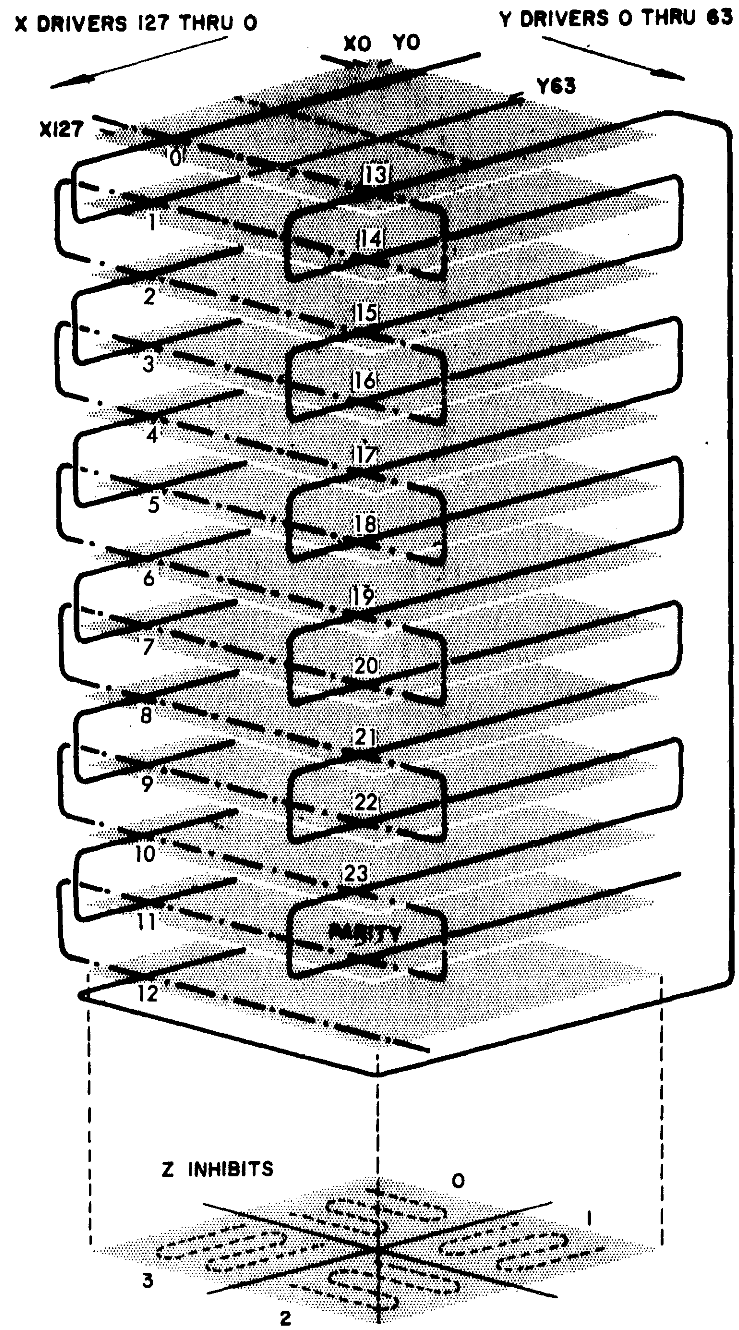
Quadrant 0 is Y(0-63) X(0-63) Address 0000 - 7777 bits 0 - 6

Quadrant 1 is Y(0-63) X(0-63) Address 0000 - 7777 bits 13 - 18

Quadrant 2 is Y(0-63) X(0-63) Address 0000 - 7777 bits 19 - Parity

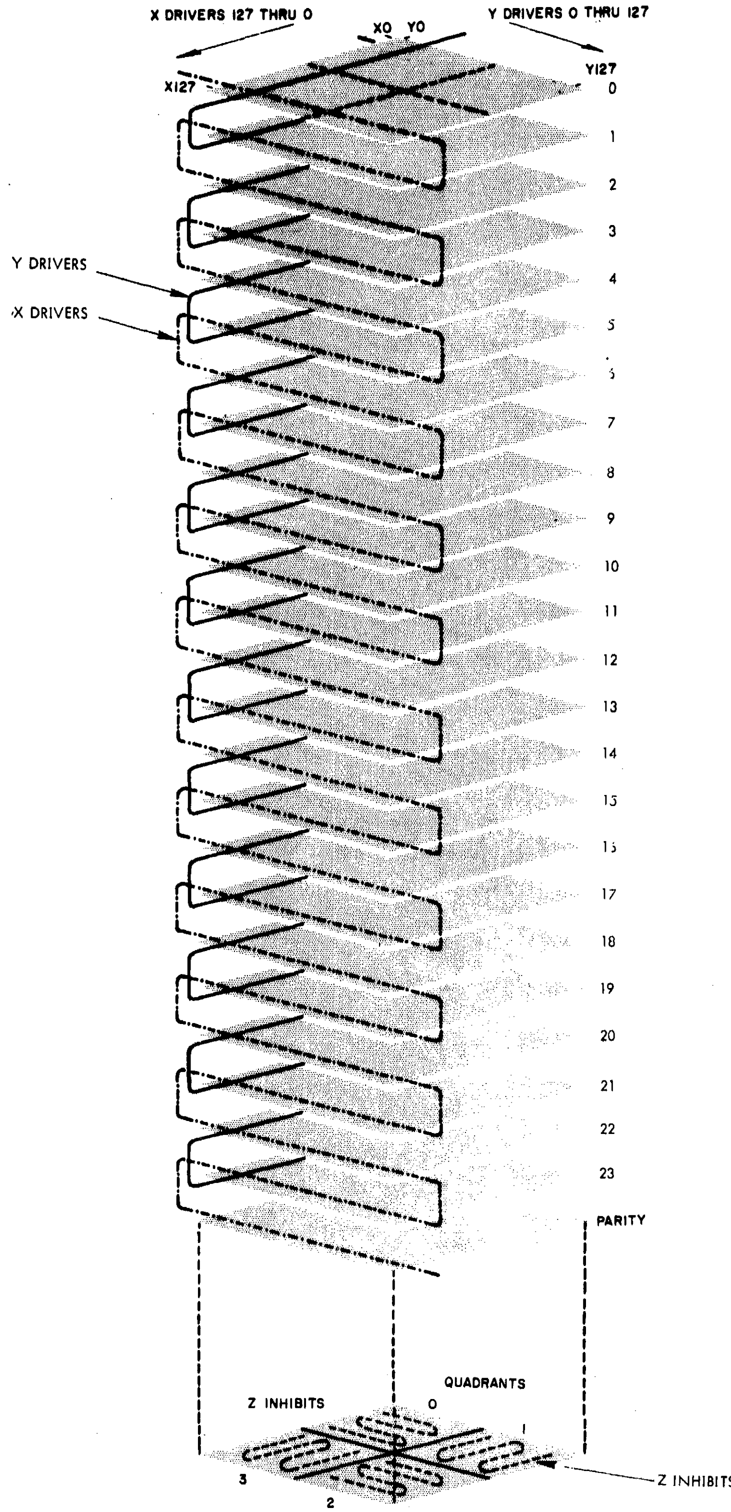
Quadrant 3 is Y(0-63) X(0-63) Address 0000 - 7777 bits 7 - 12

Figure 1. Memory Stack, 4K



Quadrant 0 is Y(0-63) X(0-63)	Address	0000 -	7777	bits	0 - 12
Quadrant 1 is (Y0-63) X(0-63)	Address	0000 -	7777	bits	13 - Parity
Quadrant 2 is (Y0-63) X(64-127)	Address	10000 -	17777	bits	13 - Parity
Quadrant 3 is Y(0-63) X(64-127)	Address	10000 -	17777	bits	0 - 12

Figure 2. Memory Stack, 8K



Z inhibit lines run parallel to the X drivers in Quadrants 0 and 2 and run parallel to the Y drivers in Quadrants 1 and 3.

The sense windings run diagonally by Quadrants.

Quadrant 0 is Y(0-63)	· X(0-63)	Address	0000 - 7777
Quadrant 3 is Y(0-63)	· X(64-127)	Address	10000 - 17777
Quadrant 1 is Y(64-127)	· X(0-63)	Address	20000 - 27777
Quadrant 2 is Y(64-127)	· X(64-127)	Address	30000 - 37777

Figure 3. Memory Stack, 16K

USE:
(cont.)

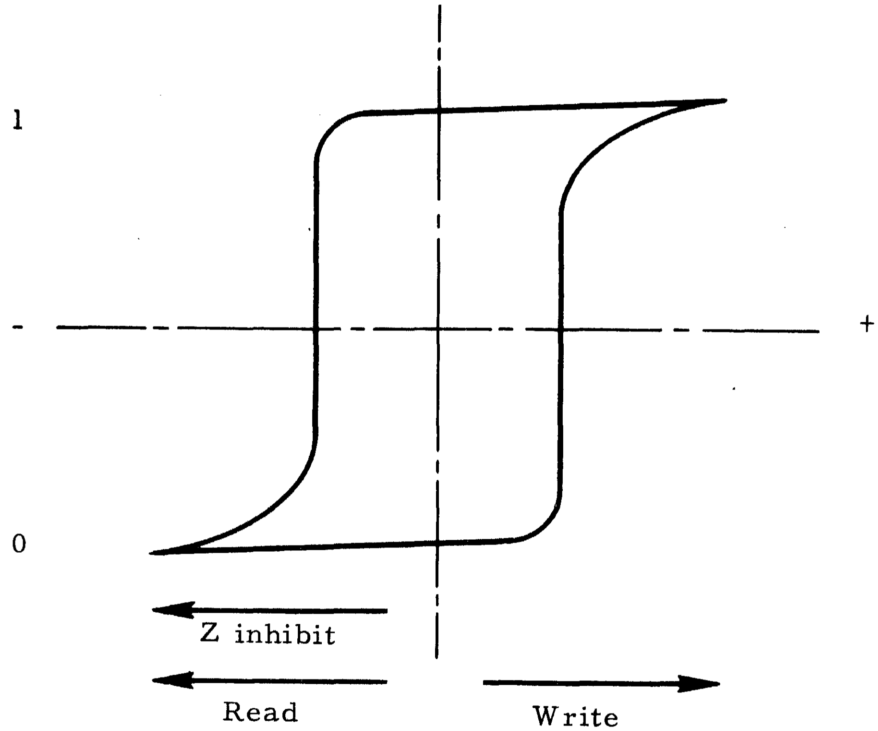


Figure 4. Current Flow

There are two switches, a read switch and a write switch, for each X and Y sink. The current flow in a read sink is equal to and opposite that in a write sink. The Z inhibit current is in the same direction as the read sink current.

If an address selects a core in a "one's" state, the X and Y read sinks combine two half-currents in the read direction and cause the core to change state. The change of state induces a voltage across the sense windings and sends a one-bit into the corresponding M register bit position. During the write cycle portion of the memory cycle, the X and Y write sinks restore the core to its original state by combining two half-currents in the write direction, causing the core to change state.

If, after the read phase and prior to the write phase of the memory cycle, an instruction changes a "one", read out of memory and into the M register, to a "zero" during the write cycle, the Z inhibit driver yields one half-current opposing the X and Y half-currents; therefore, no data is read back into core, which is a zero in memory. (A store instruction can cause such a change.)

USE: (cont.)

If, during the read phase, a "zero" is already in a core, the core does not change state when the two half-currents select it, and no change in potential occurs across the sense windings; this reads out a "zero". During the write phase of the memory cycle, the zero in the memory flip-flop triggers the Z inhibit driver and the core retains a "zero" state. However, if a "one" enters the memory flip-flop after the read but before the write, the Z inhibit driver does not trigger and the core changes to a "one's" state.

MEMORY EXTENSION SYSTEM

Programs set the 930 Extend Memory Registers, EM3 and EM2, with an EOD instruction. Bits 16 and 17 select the register that the EOD changes; bits 18, 19, 20 and/or 21,22,23 contain the new contents for the registers.

EOD		S3	S2	E M 3	E M 2	EOD Format
		16	17	18 19 20	21 22 23	
EOD		0	0	Ignored	Ignored	No register set
EOD		0	1	Ignored	X X X	Set Bits XXX into EM2.
EOD		1	0	Y Y Y	Ignored	Set Bits YYY into EM3.
EOD		1	1	Y Y Y	X X X	Set Bits YYY into EM3. Set Bits XXX into EM2.

USE: (cont.)

The two most significant bits of an address, bits 10 and 11, determine the action of the Memory Extension; it reacts as follows:

Address bits 10	11	
0	0	Do not use Extend Memory Register
0	1	Do not use Extend Memory Register
1	0	Use contents of EM2 as "fifth" digit of address
1	1	Use contents of EM3 as "fifth" digit of address

The following examples show the relationship between the setting of EM3 and EM2 and the leading address bits used in an instruction. The EOD sets the new value(s) in the register(s); the addresses show the effect of the value.

EOD	New EM3 Value	New EM2 Value	Sample Address Written	Actual Address Used
EOD 20300	0	0	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 0xxxx 0xxxx
EOD 20305	0	5	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 5xxxx 0xxxx
EOD 20354	5	4	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 4xxxx 5xxxx
EOD 20332	3	2	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 2xxxx 3xxxx
This latter register setting is the normal setting.				

PUNCHED PAPER TAPE VERSION

SCIENTIFIC DATA SYSTEMS

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

 Catalog No. 304001

IDENTIFICATION: Memory Diagnostic

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: August 28, 1964

COMPUTER CONFIGURATION: Any SDS 930

PURPOSE: The program exercises memory with the worst-case memory word pattern; it monitors the memory for errors and aids in diagnosing memory failures.

PROGRAMMED OPERATORS: None

STORAGE: All of memory

TIMING: N/A

USE:

A. INTRODUCTION

The 930 Memory Diagnostic Program tests core memories of any size, 4096 through 32,768 words In performing the test, the program writes a worst case pattern to generate noise throughout memory, and then tests each memory word individually. The worst case pattern is a checkerboard of words of all zeros and of all ones. If, during the testing process, the program detects an error, it can display the failing address and bit positions, aiding the operator in determining the circuitry which is malfunctioning.

The program uses the breakpoint switches to provide various modes of program operation. See Section B for the mode discussion.

For ease of troubleshooting, the program provides a "sync" bit on one word of the word testing loop. The operator can synchronize an oscilloscope on this bit to observe the action of continuously testing one word memory. An additional pulse generated by an EOM instruction external to the above-mentioned testing loop provides a broad sync loop.

USE: (cont.)

The program also provides the facility for observing "ringing" in the memory circuitry. During operation the program tests upper core, then moves itself to that area and tests lower core where it had been stored. When the program completes the lower core test (one test cycle), it returns to test upper core again (performs another test cycle). Depending on memory size--greater than or equal to 4K--the program tests different amounts of memory during the upper/lower parts of the test cycle. The operator also can select the portion of memory to be tested.

In each part of the cycle, the program covers the core area to be tested with the worst case pattern. Then the program accesses, inverts, tests, and restores to the test pattern each word in that area. The program inverts the worst case pattern between each repetition of the test cycle.

B. OPERATING THE MEMORY DIAGNOSTIC PROGRAM

1. Load program by normal fill procedure.
2. Select program options as follows:

Breakpoint No. 1

RESET position causes the program to test each word in sequence one time per test cycle.

SET position causes the program to test repeatedly the first erroneous word encountered.

Breakpoint No. 2

RESET position causes computer to halt when it encounters an error. To continue, clear the halt.

SET position causes computer to continue (not halt) when it encounters an error.

Error Indication

If Breakpoint No. 2 is RESET when the program encounters an error as it reads the inverted test word, the program displays the error as follows:

Register	Meaning
C_{9-23}	The true 15-bit address capable of displaying 32K

USE: (cont.)

B	The correct contents of the specified address
X	The erroneous contents detected in the specified address.

The discrepancies between (B) and (X) are the error positions. If a 1-bit appears in X in bit position i but not in B in that position, the failure is a bit picked up in position i. If a 1-bit appears in B in bit position j but not in $\overline{X}$ in that position, the failure is a bit dropped from position j. This determines the bit positions that failed since the address uniquely determines the drive and sink circuitry.

Breakpoint No. 3

During normal fill procedure

RESET position causes the normal fill to take place and causes the Memory Diagnostic program to be loaded and executed.

SET position causes the computer to halt and display:

The location plus one of the last word in memory in $A_{(9-23)}$

The location of the first word in memory in $B_{(9-23)}$

At this point the operator can change the lower and upper limits of memory by inserting the desired lower and upper limits in B and A, respectively.

During the running of the program

RESET position causes the Memory Diagnostic to use the actual lower and upper limits of the memory being tested.

SET position causes the Memory Diagnostic to use those limits provided during loading instead of the actual limits of the memory being tested.

By resetting Breakpoint 3, the program returns to the normal mode.

Breakpoint No. 4

RESET position causes no action.

USE: (cont.)

Breakpoint No.	SET	RESET
1	Repeated test of first erroneous word	Test all words
2	Continue on Error	Halt on Error Error Indication C_{9-23} = 15-bit address B_{9-23} = correct contents X = erroneous contents 1 in X_i but not in B_i implies bit picked up in position i 1 in B_i but not in X_i implies bit dropped from position i
3	During Fill Load and halt with display A_{9-23} = Location plus one of last word in memory B_{9-23} = Location plus one of first word in memory To change limits, insert into A and B	During Fill Load and run
	During Program Execution Program uses limits provided at load time	During Program Execution Program uses actual limits of memory being tested
4	Halt at each half-cycle Display A = current 1's word B = current 0's word X = accumulation of errors	Ignore

USE: (cont.)

CHANGE TO AN ODD PARITY

SET position causes the program to halt before beginning each half of the test cycle and to display:

The current one's word in A

The current zero's word in B, and

The number of failures that have occurred in X (binary integer scaled at 23).

At this point, the operator can change any of the registers by clearing it and inserting the desired bit pattern. To obtain a count of errors for each half of the test cycle, clear X since the program continues to accumulate errors from cycle to cycle.

The computer halts at location 3 or at 7703/17703 depending on memory size and on the program's position when the Breakpoint is SET.

If SET, during the testing of upper memory, the computer halts at 3; if SET when testing lower memory, it halts at 7703 for a 4K core size, or at 17703 for a greater than 4K core size.

C. WORST-CASE PATTERN MEMORY TEST

The worst-case pattern consists of a checkerboard pattern of binary words of all 1's and words of all 0's. An example is:

Location	Contents
00000	77777777
00001	00000000
00002	77777777
00003	00000000
.	.
.	.
.	.

Shown in the form of X and Y address components, where a single digit represents a full word, this pattern is:

USE: (cont.)

Y →

X ↓	00	01	02	03	. . . 77
0000	1	0	1	1	0
0100	0	1	0	1	1
0200	1	0	1	0	0
0300	0	1	0	1	1
.					.
.					.
.					.
7700	0	1	0	1	1

The logical function that determines the pattern is directly dependent on the least significant bit of the X and Y sink addresses. Assuming that the address bits go from 9 through 23 in the C Register, the X sink bits and the Y sink bits are 15, 16, 17, and 21, 22, 23, respectively. This function is:

1. If address bit 17 is equal to address bit 23, the program fills the location with all 1-bits.
2. If address bit 17 is not equal to address bit 23, the program fills the location with all 0-bits.

Examples:

Address	Bit 17	Bit 23	Word Put in Location
01000	0	0	77777777
01001	0	1	00000000
00304	1	0	00000000
00503	1	1	77777777

The program alternates the above pattern at the end of each memory testing cycle. The change occurs each time the memory area tested goes from lower memory to upper memory. Thus, a complete check of memory requires two complete test cycles. Setting Breakpoint 3 causes the pattern to change between each half of the test cycle.

USE: (cont.)

The worst-case pattern used in this program imposes the maximum noise on the output of the sense winding during reading or writing memory. The memory circuitry can detect such noise as information resulting in memory error.

D. ERROR CHECK

1. Error Halts in Program

If Breakpoint No. 2 is RESET when an error is detected, the program halts and displays the error. The program halts in one of several locations depending on program placement and memory size. When the program is testing upper core, the error halt is location 73_8. When testing lower core in a machine with 4K, the error halt location is 7773_8. When testing lower core in a machine with greater than 4K, the error halt location is 17773_8.

2. Error Checking Aids

The program provides a sync bit on one word in the test loop. With Breakpoint No. 1 SET, the operator can sync an oscilloscope on the sync bit. An EOM executed at the beginning of each half of the test cycle gives the operator a sync pulse on a larger loop. A right-cycle-48 executed once during each half of the test cycle turns memory off for six complete machine cycles; this allows the operator to observe "ringing" effects in memory and to observe its severity.

3. Use of Error Indications

Initially, if nothing loads into memory, check the power supply for proper power levels.

If the program loads and then detects errors during the operation, the operator can suspect the memory circuitry or memory adjustment.

When the operator determines the memory stack type -- 4K, 8K and/or 16K -- he can use the following to help determine the exact failure point.

USE: (cont.)

X-Y Drivers/Sinks

When an error occurs, the C Register contains the error-word's location: C_9 through C_{23}. These address bits correspond to the X, Y drivers and sinks according to the following tables:

X Sinks

	C21	C22	C23
SRX0 SWX0	0	0	0
SRX1 SWX1	0	0	1
SRX2 SWX2	0	1	0
SRX3 SWX3	0	1	1
SRX4 SWX4	1	0	0
SRX5 SWX5	1	0	1
SRX6 SWX6	1	1	0
SRX7 SWX7	1	1	1

Y Sinks

	C15	C16	C17
SRY0 SWY0	0	0	0
SRY1 SWY1	0	0	1
SRY2 SWY2	0	1	0

USE: (cont.)

SRY3 SWY3	0	1	1
SRY4 SWY4	1	0	0
SRY5 SWY5	1	0	1
SRY6 SWY6	1	1	0
SRY7 SWY7	1	1	1

	C11	C18	C19	C20
DX0	0	0	0	0
DX1	0	0	0	1
DX2	0	0	1	0
DX3	0	0	1	1
DX4	0	1	0	0
DX5	0	1	0	1
DX6	0	1	1	0
DX7	0	1	1	1
DX10	1	0	0	0
DX11	1	0	0	1
DX12	1	0	1	0
DX13	1	0	1	1
DX14	1	1	0	0
DX15	1	1	0	1
DX16	1	1	1	0
DX17	1	1	1	1

<u>Y Drivers</u>

	C10	C11	C12	C13
DY0	0	0	0	0
DY1	0	0	0	1

USE: (cont.)

DY2	0	0	1	0
DY3	0	0	1	1
DY4	0	1	0	0
DY5	0	1	0	1
DY6	0	1	1	0
DY7	0	1	1	1
DY10	1	0	0	0
DY11	1	0	0	1
DY12	1	0	1	0
DY13	1	0	1	1
DY14	1	1	0	0
DY15	1	1	0	1
DY16	1	1	1	0
DY17	1	1	1	1

Z Inhibit Drivers

The Z Inhibit Drivers are functions of bit position and memory stack size. Number the bit positions in memory as M0 (most significant) through M23 (least significant) and M24 (parity).

used with 4K; 8K or 16K stacks when $C_{10}\ C_{11} = 00$

Z0d0 = M0

Z1d0 = M1

.
.
.

Z23d0 = M23

Z24d0 = M24 (parity)

used with 8K or 16K stacks when $C_{10}\ C_{11} = 01$

Z0d1 = M0

Z1d1 = M1

.
.
.

Z24d1 = M24 (parity bit)

USE: (cont.)

used with 16K stacks only when $C_{10} C_{11} = 10$

Z0d2 = M0

.

.

.

Z24d2 = M24 (parity bit)

used with 16K stacks only when $C_{10} C_{11} = 11$

Z0d3 = M0

.

.

.

Z24d3 = M24 (parity bit)

Sense Amplifiers

The Sense Amplifiers are functions of the following bit positions:

Sense Amplifiers

Md0 = M0

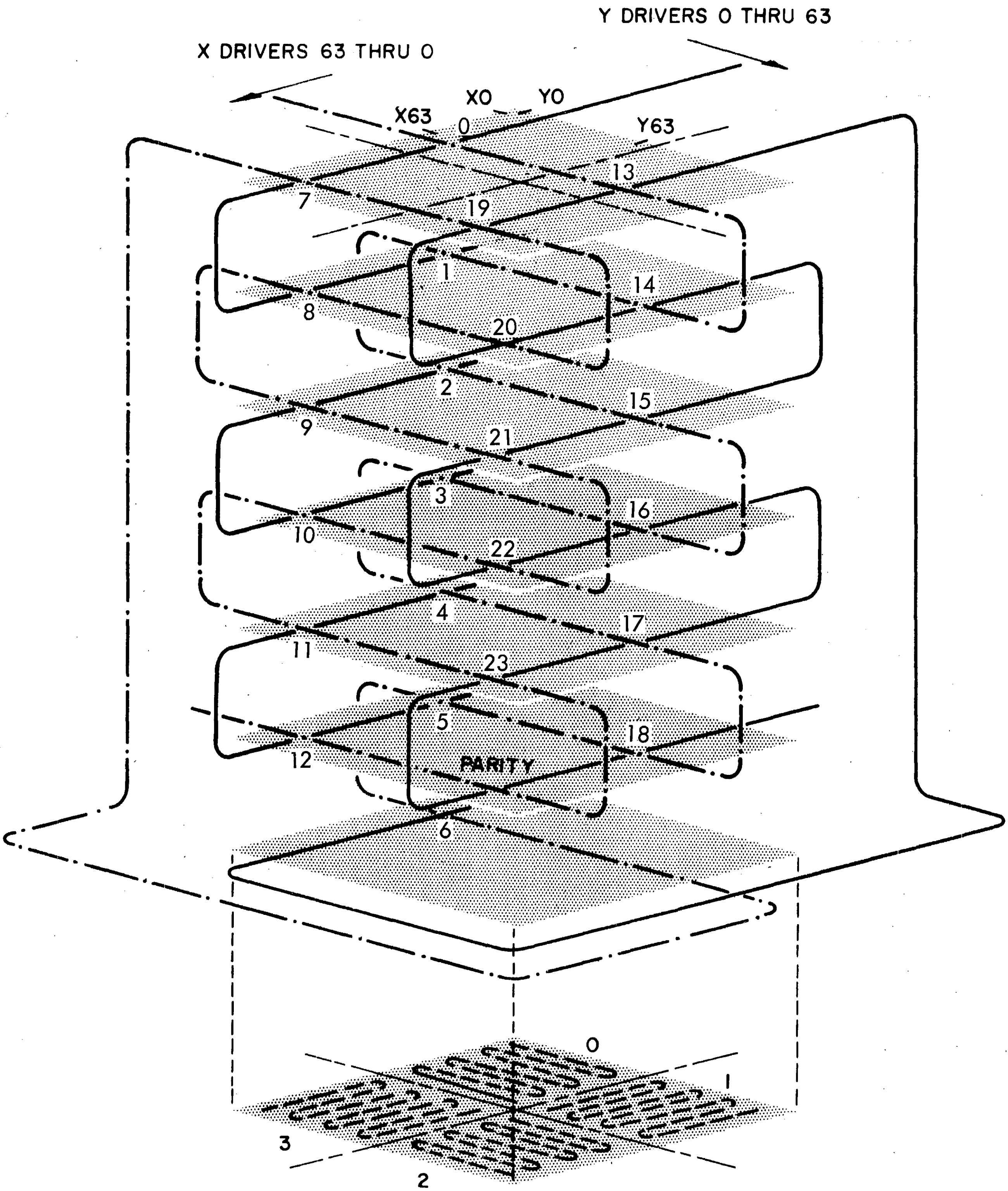
Md1 = M1

.

.

.

Md24 = M24 (parity bit)



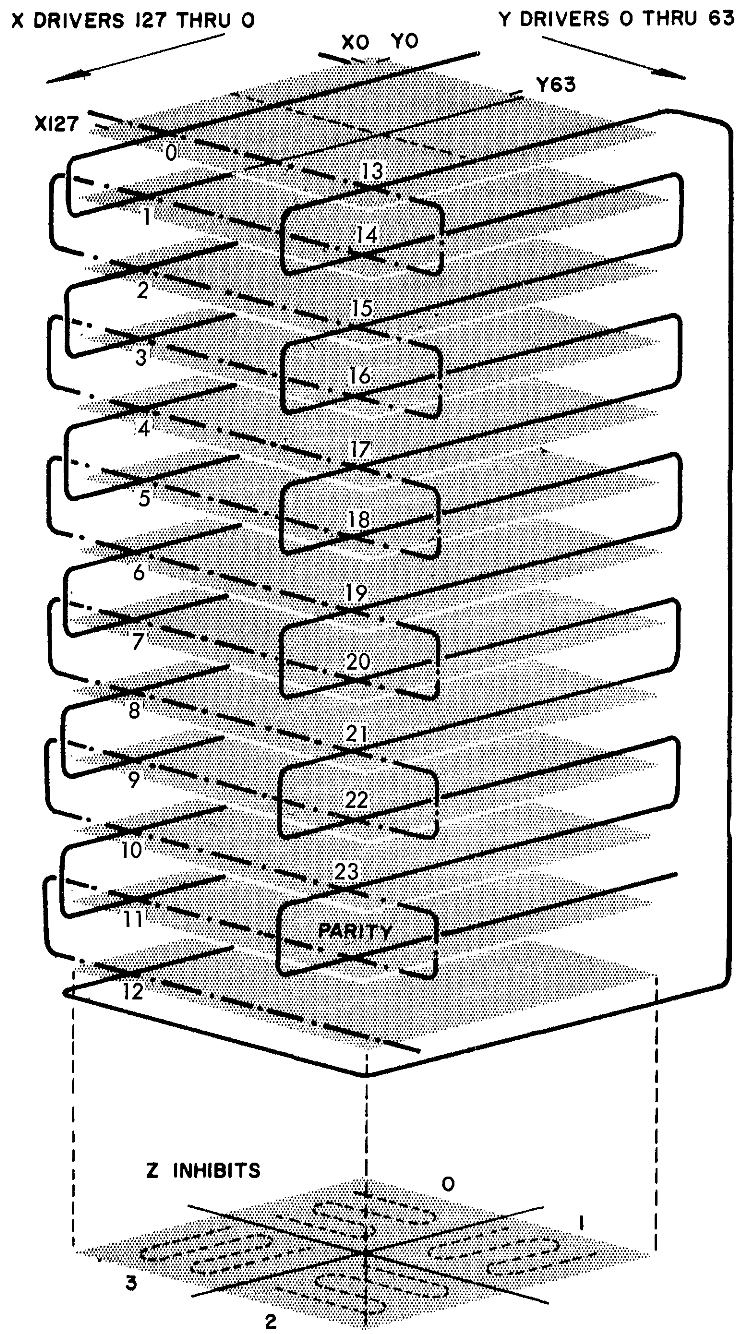
Quadrant 0 is Y(0-63) X(0-63) Address 0000 - 7777 bits 0 - 6

Quadrant 1 is Y(0-63) X(0-63) Address 0000 - 7777 bits 13 - 18

Quadrant 2 is Y(0-63) X(0-63) Address 0000 - 7777 bits 19 - Parity

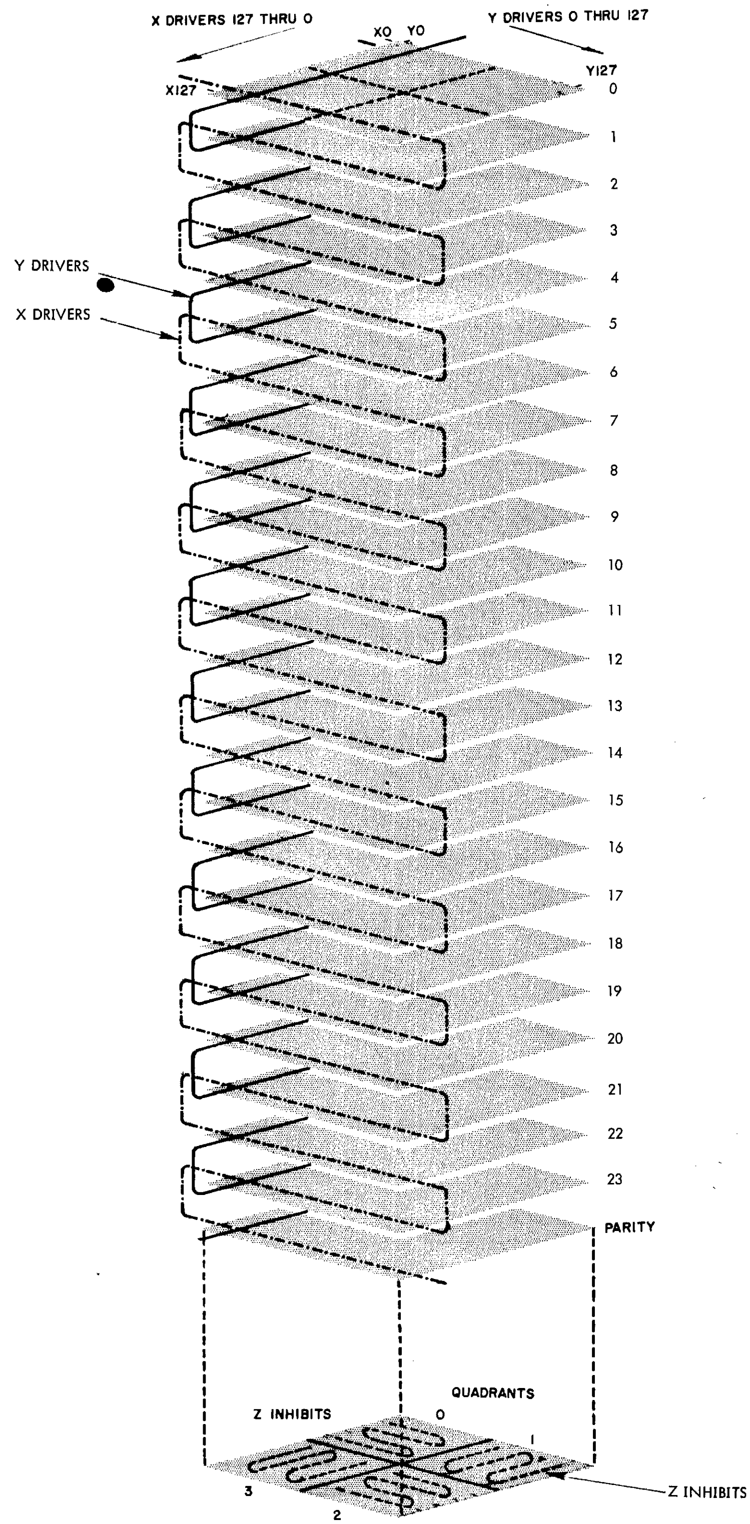
Quadrant 3 is Y(0-63) X(0-63) Address 0000 - 7777 bits 7 - 12

Figure 1. Memory Stack, 4K



Quadrant 0 is Y(0-63) X(0-63)	Address	0000 - 7777	bits	0 - 12
Quadrant 1 is Y(0-63) X(0-63)	Address	0000 - 7777	bits	13 - Parity
Quadrant 2 is Y(0-63) X(64-127)	Address	10000 - 17777	bits	13 - Parity
Quadrant 3 is Y(0-63) X(64-127)	Address	10000 - 17777	bits	0 - 12

Figure 2. Memory Stack, 8K



Z inhibit lines run parallel to the X drivers in Quadrants 0 and 2 and run parallel to the Y drivers in Quadrants 1 and 3.

The sense windings run diagonally by Quadrants.

Quadrant 0 is Y(0-63)	· X(0-63)	Address	0000 - 7777
Quadrant 3 is Y(0-63)	· X(64-127)	Address	10000 - 17777
Quadrant 1 is Y(64-127)	· X(0-63)	Address	20000 - 27777
Quadrant 2 is Y(64-127)	· X(64-127)	Address	30000 - 37777

Figure 3. Memory Stack, 16K

USE: (cont.)

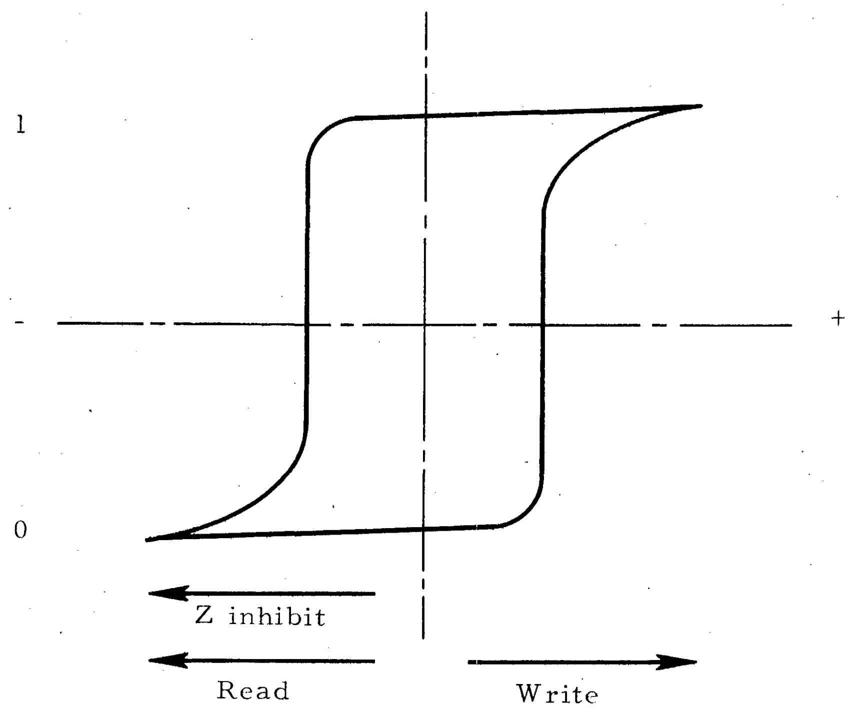


Figure 4. Current Flow

There are two switches, a read switch and a write switch, for each X and Y sink. The current flow in a read sink is equal to and opposite that in a write sink. The Z inhibit current is in the same direction as the read sink current.

If an address selects a core in a "one's" state, the X and Y read sinks combine two half-currents in the read direction and cause the core to change state. The change of state induces a voltage across the sense windings and sends a one-bit into the corresponding M Register bit position. During the write cycle portion of the memory cycle, the X and Y write sinks restore the core to its original state by combining two half-currents in the write direction, causing the core to change state.

If, after the read phase and prior to the write phase of the memory cycle, an instruction changes a "one", read out of memory and into the MRegister, to a "zero" during the write cycle, the Z inhibit driver yields one half-current opposing the X and Y half-currents; therefore, no data is read back into core, which is a zero in memory. (A store instruction can cause such a change.)

USE: (cont.)

If, during the read phase, a "zero" is already in a core, the core does not change state when the two half-currents select it, and no change in potential occurs across the sense windings; this reads out a "zero". During the write phase of the memory cycle, the zero in the memory flip-flop triggers the Z inhibit driver and the core retains a "zero" state. However, if a "one" enters the memory flip-flop after the read but before the write, the Z inhibit driver does not trigger and the core changes to a "one's" state.

E. MEMORY EXTENSION SYSTEM

Programs set the 930 Extend Memory Registers, EM3 and EM2, with an EOD instruction. Bits 16 and 17 select the register that the EOD changes; bits 18, 19, 20 and/or 21,22,23 contain the new contents for the registers.

EOD		S3	S2	E M 3	E M 2	EOD Format
		16	17	18 19 20	21 22 23	
EOD		0	0	Ignored	Ignored	No register set
EOD		0	1	Ignored	X X X	Set Bits XXX into EM2.
EOD		1	0	Y Y Y	Ignored	Set Bits YYY into EM3.
EOD		1	1	Y Y Y	X X X	Set Bits YYY into EM3. Set Bits XXX into EM2.

USE: (cont.)

The two most significant bits of an address, bits 10 and 11, determine the action of the Memory Extension; it reacts as follows:

Address bits 10	11	
0	0	Do not use Extend Memory Register
0	1	Do not use Extend Memory Register
1	0	Use contents of EM2 as "fifth" digit of address
1	1	Use contents of EM3 as "fifth" digit of address

The following examples show the relationship between the setting of EM3 and EM2 and the leading address bits used in an instruction. The EOD sets the new value(s) in the register(s); the addresses show the effect of the value.

EOD	New EM3 Value	New EM2 Value	Sample Address Written	Actual Address Used
EOD 00300	0	0	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 0xxxx 0xxxx
EOD 00305	0	5	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 5xxxx 0xxxx
EOD 00354	5	4	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 4xxxx 5xxxx
EOD 00332	3	2	0xxxx 1xxxx 2xxxx 3xxxx	0xxxx 1xxxx 2xxxx 3xxxx
This latter register setting is the normal setting.				

USE: (cont.)

F. MEMORY DIAGNOSTIC (MD) PROGRAM PARTICULARS

This program contains its own loader, which the standard loader loads into memory starting at location 200_8. This loader loads MD into locations 0 through 177_8, performs a checksum, and halts at P=204_8 if a checksum failure occurs.

The loader tests for memory size greater than 4K; if it finds the larger memory, the loader loads the appropriate "greater than 4K" routine as detailed in the next paragraph. If the memory is 4K, the loader reponds to Breakpoint 3, and adjusts the pertinent program parameters. The loader then reads past the "greater than 4K" routine on the tape and transfers control to the Memory Diagnostic. MD stores the worst case pattern in memory from 200_8 through the last location in core. MD then tests each cell within the worst case pattern. The program then moves itself into locations 7600_8 through 7777_8. MD writes the worst case pattern into locations 0 through 7600_8 and tests each cell. This represents one complete test cycle. The program then moves itself to locations 0 through 177_8 and repeats the cycle with the worst case pattern inverted.

If the MD loader finds a memory of more than 4K words, it reads a new loader into location 300_8. This loader loads the "greater than 4K" MD into core at location 0 through 277_8. The loader performs a checksum and halts with the P counter equal to 316_8 in case of checksum failure. The loader determines the exact memory size, responds to Breakpoint 3, adjusts pertinent parameters, and transfers to MD. MD stores the worst case pattern in memory from location 10000_8 through the last location in memory. MD tests each cell with the pattern. MD then writes a simple constant into locations 10000_8 through 17777_8 to nullify the worst case pattern and moves itself into locations 17500_8 through 17777_8. The program writes the worst case pattern in lower core, locations 0 through 7777_8, and tests each cell with the pattern. This represents one complete test cycle. MD then writes a simple constant into locations 0 through 7777_8 to nullify the worst case pattern and moves itself to locations 0 through 277_8 and repeats the cycle with the worst case pattern inverted.

G. TAPE DESCRIPTION

(front of tape) →

> 16K MD	>16K Loader	Symbol Loader	<16K MD	<16K Loader	Symbol Loader

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 304002

IDENTIFICATION: Instruction Diagnostic

AUTHOR: R. Resnick, M. Mulholland

ACCEPTED: August 28, 1964

COMPUTER CONFIGURATION: SDS 930

PURPOSE: This program aids in diagnosing faulty computers by verifying proper execution of computer logic.

PROGRAM OPERATORS: N/A

STORAGE: 4096 words starting at Location 00000

TIMING: Approximately 53 milliseconds per pass when in the Automatic mode

USE: There are three reasons for testing the performance of a computer:

> Evaluation of the computer's performance,
> Detection of errors when the computer has failed, and
> Verification of the computer's performance.

This program provides for all three.

To test the computer's performance, it is necessary to perform:

> a Manual Test that involves the execution of a minimum number of specific instructions using the control console and comparing the resulting contents of the programmable registers with predetermined constants,
>
> a Semi-automatic Test, which uses a simple program principally consisting of those instructions tested during the Manual test; that executes a few more instructions and halts after each execution to let the operator inspect the registers for predetermined results, and
>
> an Automatic Test, which uses a complicated program containing those instructions tested during the previous two tests, that performs the remainder of the Instruction Repertoire, compares the results with predetermined constants and gives error indications when necessary.

USE: (cont.)

The Evaluation procedure involves all three tests and is used principally when there is no past history of the computer's performance as would be for a computer just off the assembly line. The Error Detection procedure involves the Semi-automatic and Automatic test and is used when a previously operative computer is suspected of having failed. The Verification procedure involves only the Automatic test and is used with Preventive Maintenance as well as for periodic computer reliability verification.

USE: (cont.)

I. MANUAL INSTRUCTION TEST

The Manual Instruction Test is a procedure that the operator performs without the use of an internally stored program. The operator presets the programmable registers, executes via the control console one of the five instructions to be tested, and compares the resulting constants in the registers with the list of predetermined constants. When these constants differ from the predetermined constants, an error (failure) has occurred.

In the case of an error, the operator should investigate it by observing via an oscilloscope, the circuitry operations during the execution of the instruction. Before using the oscilloscope,

1) raise the HOLD switch to hold the P Counter,

2) store the instruction being tested, with a 1-bit in bit 0, into the location currently displayed in the P Register,

3) raise the mode switch to RUN, and

4) observe the appropriate signal levels via the oscilloscope.

Note that the HOLD switch itself may be faulty.

The procedure tests the following instructions in the order given:

HLT	Halt	00
BRU	Branch Unconditionally	01
BRX	Increment Index and Branch (no branch case)	41
BRX	Increment Index and Branch (branch case)	41
LDX	Load X	71
Automatic Fill		

USE: (cont.)

The following gives the individual instruction test procedure:

A. HALT (00) TEST

1. Set mode switch to "IDLE," and press the START button.
2. Preset the registers as follows:

A	=	77700070
B	=	66611161
X	=	55522252
P	=	00000
C	=	00000000
Overflow	=	reset

3. Set the mode switch to "RUN". The computer should wait (half light goes on.)
4. Set the mode switch to "IDLE" and check the registers for the following results:

A	=	77700070
B	=	66611161
X	=	55522252
P	=	00001
Overflow	=	reset

5. If an error occurs, follow the procedure outlined on the first page of the Manual Instruction Test program.

B. BRANCH UNCONDITIONALLY (01) TEST

1. Press the Start Button.
2. Preset the registers as follows:

A	=	66611161
B	=	55522252

USE: (cont.)

X = 44433343

P = 00000

C = 00107700

Overflow = reset

3. Set the mode switch to "STEP" and back to "IDLE".
4. Check the registers for the following results:

A = 66611161

B = 55522252

X = 44433343

P = 07700

Overflow = reset

5. If an error occurs, follow the procedure outlined on the first page of the Manual Instruction Test program.

C. INCREMENT INDEX AND BRANCH (41) TEST, BRANCH CASE

1. Press the start button.
2. Preset the registers as follows:

A = 55522252

B = 44433343

X = 33344434

P = 00000

C = 04107654

Overflow = reset

3. Set the mode switch to "STEP" and back to "IDLE".
4. Check the registers for the following results:

A = 55522252

B = 44433343

X = 33344435

P = 07654

Overflow = reset

5. If an error occurs, follow the procedure outlined on the first page of the Manual Instruction Test program.

USE: (cont.)

D. INCREMENT INDEX AND BRANCH (41) TEST, NOT BRANCH CASE

1. Press the start button.
2. Preset the registers as follows:

   A = 00077707

   B = 11166616

   X = 55522252

   P = 00000

   C = 04107654

   Overflow = reset

3. Set the mode switch to "STEP" and back to "IDLE".
4. Check the registers for the following results:

   A = 00077707

   B = 11166616

   X = 55522253

   P = 00001

   Overflow = reset

5. If an error occurs, follow the procedure outlined on the first page of the Manual Instruction Test program.

E. AUTOMATIC FILL TEST

1. Press start button. Load reader with WIM Test, (or WIM test loop).
2. Preset the register as follows:

   A = 44433343

   B = 22255525

   X = 77777777

   P = 00000

   C = 00000000

   Overflow = reset

USE: (Cont.)

3. Set the mode switch to "RUN". Raise paper tape fill switch and release.
4. The WIM test should load and then the computer should halt.
5. Set the mode switch to "IDLE" and check the registers for the following results:

   A = 44433343

   B = 22255525

   X = 11111111

   P = 00011

   C = 77777777

   Overflow = reset
6. If an error occurs refer to the Automatic Fill portion of the Peripheral Diagnostic Manual.

USE: (cont.)

II. SEMI-AUTOMATIC INSTRUCTION TEST PROGRAM

The Semi-Automatic Instruction Test is a sequence of repeatable, program-controlled tests. As the program performs each test, it periodically halts, leaving preselected constants in all of the programmable registers. When the actual contents of these registers differ from the constants given in the test procedure, an error has occurred.

The operator is provided with the following breakpoint options:

Breakpoint 1 Set	=	repeat the preceding test
Breakpoint 1 Reset	=	normal operation
Breakpoint 2 Set	=	bypass programmed halts
Breakpoint 2 Reset	=	halt and inspect the registers during each test

Prior to loading:

Breakpoint 4 Set	=	Load semi-automatic Instruction Program Test
Breakpoint 4 Reset	=	Bypass Semi-automatic Instruction Test Program and load the Automatic Instruction Test Program

Upon completion of the Semi-automatic Instruction Test Program:

Breakpoint 4 Set	=	Program to repeat Semi-automatic Instruction Test Program
Breakpoint 4 Reset	=	Load Automatic Instruction Test Program

USE: (cont.)

The Breakpoint 1 option is available for all tests except the Breakpoint 1 tests.

The Breakpoint 2 option is available for all tests except the Breakpoint 2 tests.

A sync bit is provided in bit zero in all instructions being tested. With Breakpoints 1 and 2 set, the program repeatedly performs each test without halts; only the instruction being tested has a sync bit. This provides a means to synchronize an oscilloscope with the test loop to aid the operator in observing signal levels of any circuitry in question.

The program tests the following instructions in the order given:

Mnemonic Op Code	Test Case	Octal Op Code
BPT 1	BREAKPOINT NO. 1 TEST, reset case	40
BPT 1	BREAKPOINT NO. 1 TEST, set case	40
BPT 2	BREAKPOINT NO. 2 TEST, reset case	40
BPT 2	BREAKPOINT NO. 2 TEST, set case	40
BPT 3	BREAKPOINT NO. 3 TEST, reset case	40
BPT 3	BREAKPOINT NO. 3 TEST, set case	40
BPT 4	BREAKPOINT NO. 4 TEST, reset case	40
BPT 4	BREAKPOINT NO. 4 TEST, set case	40
LDX	LOAD X	71
LDB	LOAD B	75
LDA	LOAD A	76
LDX*	LOAD X With Indirect Addressing	71
LDB*	LOAD B With Indirect Addressing	75
LDA*	LOAD A With Indirect Addressing	76

USE: (cont.)

STB	STORE B	36
STX	STORE X	37
STB*	Store B With Indirect Addressing	36
STX*	Store X With Indirect Addressing	37
BRR	RETURN BRANCH, Overflow Set	51
OVT	SKIP IF OVERFLOW NOT SET AND RESET OVERFLOW, set case	40
BRR	RETURN BRANCH, Overflow not set	51
BRR*	RETURN BRANCH, with Indirect Addressing - Overflow not set	51
BRX	INCREMENT INDEX AND BRANCH (no branch case)	41
OVT	SKIP IF OVERFLOW NOT SET AND RESET OVERFLOW, not set case	40
BRX	INCREMENT INDEX AND BRANCH (branch case)	41
BRX*	INCREMENT INDEX AND BRANCH for branch case with indirect addressing	41
SKE	SKIP IF A EQUAL M FOR A = M case	50
SKE	SKIP IF A EQUAL M FOR A $\neq$ M case	50

The following gives the loading procedure, test entry procedure, and the individual instruction tests operation procedures.

A. LOAD PROGRAM

1. Set Breakpoint 4, reset Breakpoints 1, 2, and 3.
2. Perform normal fill procedure.
3. After the tape loads, the computer halts with P = 211.

B. ENTER PROGRAM

1. Reset all Breakpoints.
2. Put the mode switch in RUN to initiate the tests.

USE: (cont.)

C. BREAKPOINT NO. 1 TEST (40) TEST

1. Program loops as long as Bpt 1 remains reset.
2. Setting Bpt 1 causes the program to halt. When it halts, move the mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00220
C	=	44020400
Overflow	=	0

3. Move the mode switch to RUN; the program then loops as long as Bpt 1 remains set.
4. Resetting Bpt 1 causes the program to halt. Move the mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00223
C	=	44020200
Overflow	=	0

5. Otherwise, move the mode switch to RUN. To repeat Bpt 1 test, place BRU 00214 in the C Register and move the mode switch to RUN.

D. BREAKPOINT NO. 2 TEST (40) TEST

1. Program loops while Bpt 2 remains reset.
2. Setting Bpt 2 causes the program to halt. Move the mode switch to IDLE.

USE: (cont.)

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00227
C	=	44020200
Overflow	=	0

3. To repeat this test, set Bpt 1, reset Bpt 2, and move mode switch to RUN. Otherwise, reset Bpt 1 and move the mode switch to RUN; the program loops as long as Bpt 2 remains set.

4. Resetting Bpt 2 causes the program to halt. Move the mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00234
C	=	04020400
Overflow	=	0

5. To repeat this test, set Bpt 1, set Bpt 2, and move the mode switch to RUN. Otherwise, reset Bpt 1 and move the mode switch to RUN.

E. BREAKPOINT NO. 3 TEST (40) TEST

1. The program loops as long as Bpt 3 remains reset.

2. Setting Bpt 3 causes the program to halt. Set Bpt 1 if a repeat of the test is desired. Move mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00242
C	=	44020100
Overflow	=	0

USE: (cont.)

3. To repeat this test, set Bpt 1, reset Bpt 3, and move mode switch to RUN. Otherwise, reset Bpt 1 and move mode switch to RUN; the program loops as long as Bpt 3 remains set.
4. Resetting Bpt 3 causes the program to halt. Move the mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00247
C	=	04020400
Overflow	=	0

5. To repeat this test, set Bpt 1, set Bpt 3 and move the mode switch to RUN. Otherwise, reset Bpt 1 and move mode switch to RUN.

F. BREAKPOINT NO. 4 TEST (40) TEST

1. Program loops as long as Bpt 4 remains reset.
2. Setting Bpt 4 causes the program to halt. Move the mode switch to IDLE.

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00255
C	=	44020040
Overflow	=	0

3. To repeat this test, set Bpt 1, reset Bpt 4, and move mode switch to RUN. Otherwise, reset Bpt 1 and move the mode switch to RUN. The program loops as long as Bpt 4 remains set.
4. Resetting Bpt 4 causes the program to halt. Move mode switch to IDLE.

USE: (cont.)

Correct Register Contents:

A	=	70707070
B	=	01010101
X	=	40000001
P	=	00252
C	=	04020400
Overflow	=	0

5. To repeat this test, set Bpt 1, set Bpt 4, and move the mode switch to RUN. Otherwise, reset Bpt 1 and move mode switch to RUN.

   In the following tests, set Bpt 2 if the operator wishes to inhibit halts for checking registers and erroneous branching and skipping. Set Bpt 1 if the operator wishes to repeat a particular test.

G. LOAD X (71) TEST

1. Upon entering this test, the computer halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	52525252
B	=	25252525
X	=	70707070
P	=	00272
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2, and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

GG. LOAD X, (71) TEST, WITH INDIRECT ADDRESSING

1. Upon entering this test, the computer halts. Move the mode switch to IDLE and check the registers.

USE: (cont.)

Correct Register Contents:

A	=	40000001
B	=	37777776
X	=	70707070
P	=	00302
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

H. LOAD B (75) TEST

1. Upon entering this test, the computer halts. Move mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	25252525
B	=	70707070
X	=	52525252
P	=	00312
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

HH. LOAD B (75) TEST, WITH INDIRECT ADDRESSING

1. Upon entering this test, the computer halts. Move the mode switch to IDLE and check the registers.

USE: (cont.)

Correct Register Contents:

A	=	37777776
B	=	40000001
X	=	07070707
P	=	00322
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise, proceed to the next test.

I. LOAD A (76) TEST

1. Upon entering this test, the computer halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	70707070
B	=	52525252
X	=	25252525
P	=	00332
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

II. LOAD A (76) TEST, WITH INDIRECT ADDRESSING

1. Upon entering this test, the computer halts. Move the mode switch to IDLE and check the registers.

USE: (cont.)

Correct Register Contents:

A	=	40000001
B	=	37777776
X	=	70707070
P	=	00342
C	=	04020400
Overflow	=	0

2. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

3. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

J. STORE B (36) TEST

1. Upon entering this test, the computer halts. Move the mode switch to IDLE, and check the registers.

Correct Register Contents:

A	=	25255252
B	=	77777777
X	=	00000000
P	=	00354
C	=	07600750
Overflow	=	0

2. Move the mode switch to RUN

3. The computer halts again. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	77777777
B	=	77777777
C	=	00000000
P	=	00360
C	=	04020400
Overflow	=	0

USE: (cont.)

4. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.
5. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to next test.

JJ. STORE B (36) TEST, WITH INDIRECT ADDRESSING

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	11114444
B	=	07070707
X	=	70707070
P	=	00372
C	=	07600760
Overflow	=	0

2. Move the mode switch to RUN.
3. The computer halts again. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	07070707
B	=	07070707
X	=	70707070
P	=	00376
C	=	04020400
Overflow	=	0

4. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.
5. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

USE: (cont.)

K. STORE X (37) TEST

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	44441111
B	=	10101010
X	=	32323232
P	=	00410
C	=	07600754
Overflow	=	0

2. Move the mode switch to RUN.

3. The computer halts again. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	32323232
B	=	10101010
X	=	32323232
P	=	00414
C	=	04020400
Overflow	=	0

4. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

5. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

KK. STORE X (37) TEST, WITH INDIRECT ADDRESSING

1. Upon encountering the test, the computer halts. Move the mode switch to IDLE and check the registers.

USE: (cont.)

Correct Register Contents:

A	=	01014040
B	=	00007777
X	=	77770000
P	=	00426
C	=	07600765
Overflow	=	0

2. Move the mode switch to RUN.
3. The computer halts again. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	77770000
B	=	00007777
X	=	77770000
P	=	00432
C	=	04020400
Overflow	=	0

4. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.
5. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

L. RETURN BRANCH (51) TEST, SET OVERFLOW CASE

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00443, the branch failed to occur. Set Bpts 1 and 2 and set up the oscilloscope. Move mode switch to RUN.
3. If correct branch occurs, check the registers.

USE: (cont.)

Correct Register Contents:

A	=	41111114
B	=	14141414
X	=	43434343
P	=	00446
C	=	04020400
Overflow	=	1

4. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

5. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the next test.

M. OVERFLOW INDICATOR TEST AND RESET (40) TEST, SET CASE

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.

2. If the P Register is 00461, the instruction skipped, when it should not have.

3. Set Bpts 1 and 2 and set up the oscilloscope. Move the mode switch to RUN.

4. If the P Register is 00464, check the registers.

Correct Register Contents:

A	=	12121212
B	=	21212121
X	=	01010101
P	=	00464
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.

6. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats. Otherwise proceed to the next test.

USE: (cont.)

N. RETURN BRANCH (51) TEST, OVERFLOW NOT SET

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00475, the instruction failed to branch.
3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
4. If the correct branch occurs, check the registers.

Correct Register Contents:

A	=	10101010
B	=	01010101
X	=	40000001
P	=	00500
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

O. RETURN BRANCH (51) TEST, OVERFLOW NOT SET, WITH INDIRECT ADDRESSING

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00511, the instruction failed to branch.
3. Set Bpts 1 and 2 and set up the oscilloscope. Move mode switch to RUN.
4. If the correct branch occurs, check the registers.

Correct Register Contents:

A	=	00000000
B	=	70707070
X	=	07070707
P	=	00514
C	=	04020400
Overflow	=	0

USE: (cont.)

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the next test.

P. INCREMENT INDEX AND BRANCH (41) TEST, FOR "NOT BRANCH" CASE

1. Upon completion of the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00526, the instruction branched when it should not have.
3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
4. If there is no branch, check the registers.

Correct Register Contents:

A	=	77777777
B	=	00000000
X	=	00100000
P	=	00531
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the next test.

Q. OVERFLOW INDICATOR TEST AND RESET (40) TEST, FOR "NOT SET" CASE

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00531, the instruction failed to skip.
3. Set Bpts 1 and 2 and set up the oscilloscope.
4. If the instruction works properly, the computer halts. Move the mode switch to RUN.
5. If the P Register is 00547, check the registers.

USE: (cont.)

Correct Register Contents:

A	=	07707707
B	=	70070070
X	=	77777777
P	=	00547
C	=	04020400
Overflow	=	0

6. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

7. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the next test.

R. INCREMENT INDEX AND BRANCH (41) TEST FOR BRANCH CASE

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.

2. If the P register is 00560, the instruction failed to branch.

3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.

4. If the branch occurs correctly, check the registers.

Correct Register Contents:

A	=	77000077
B	=	70777707
X	=	00040001
P	=	00563
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.

6. Move the mode switch to RUN. If Bpt 1 is SET, the rest repeats, otherwise proceed to the next test.

USE: (cont.)

S. INSTRUCTION INCREMENT INDEX AND BRANCH (41) TEST WITH INDIRECT ADDRESSING FOR BRANCH CASE

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P register is 00574, the instruction failed to branch.
3. Set Bpts 1 and 2 and use the oscillscope. Move the mode switch to RUN.
4. If the branch occurs correctly, check the registers.

Correct Register Contents:

A	=	37777776
B	=	40000001
X	=	70040007
P	=	00577
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

T. SKIP IF A EQUAL TO M (50) TEST FOR CASE A=M

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P register is 00613, the instruction failed to skip.
3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
4. If the computer skips, check the registers.

Correct Register Contents:

A	=	00000000
B	=	00000000
X	=	77777776

USE: (cont.)

P	=	00616
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and set up the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the second case.
7. Now the program tests the second case. Upon reentering the test, the computer halts. Move the mode switch to IDLE and check the registers.
8. If the P Register is 00613, the instruction failed to skip.
9. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
10. If the computer skips, check the registers.

Correct Register Contents:

A	=	77777777
B	=	77777777
X	=	77777777
P	=	00616
C	=	04020400
Overflow	=	0

11. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
12. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

U. SKIP IF A EQUAL TO M (50) TEST, FOR CASE A $\neq$ M

1. Upon enetering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00634, the instruction skipped.
3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
4. If the computer doesn't skip, check the registers.

USE: (cont.)

Correct Register Contents:

A	=	40000000
B	=	00000000
X	=	77777776
P	=	00637
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the second case.
7. Now the program tests the second case. Upon re-entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
8. If the P Register is 00634, the instruction skipped.
9. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
10. If the computer doesn't skip, check the registers.

Correct Register Contents:

A	=	00000000
B	=	40000000
X	=	77777777
P	=	00637
C	=	04020400
Overflow	=	0

11. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
12. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the next test.

V. SKIP IF A EQUAL TO M (50) TEST, FOR CASE A $\neq$ M

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00655, the instruction skipped.

USE: (cont.)

3. Set Bpts 1 and 2 and use oscilloscope. Move the mode switch to RUN.
4. If the computer doesn't skip, it halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	20000000
B	=	00000000
X	=	77777776
P	=	00660
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the second case.
7. Now the program tests the second case. Upon reentering the test, the computer halts. Move the mode switch to IDLE and check the registers.
8. If the P Register is 00655, the instruction skipped.
9. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
10. If the computer doesn't skip, check the registers.

Correct Register Contents:

A	=	00000000
B	=	20000000
X	=	77777777
P	=	00660
C	=	04020400
Overflow	=	0

11. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
12. Move the mode switch to RUN. If Bpt 1 is set, the test repeats, otherwise proceed to the next test.

USE: (cont.)

W. SKIP IF A EQUAL TO M (50) TEST, FOR CASE A $\neq$ M

1. Upon entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
2. If the P Register is 00676, the instruction skipped.
3. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
4. If the computer doesn't skip, it halts. Move the mode switch to IDLE and check the registers.

Correct Register Contents:

A	=	10000000
B	=	00000000
X	=	77777776
P	=	00701
C	=	04020400
Overflow	=	0

5. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
6. Move the mode switch to RUN. If Bpt 1 is SET, the test repeats, otherwise proceed to the second case.
7. Now the program tests the second case. Upon re-entering the test, the computer halts. Move the mode switch to IDLE and check the registers.
8. If the P Register is 00676, the instruction skipped.
9. Set Bpts 1 and 2 and use the oscilloscope. Move the mode switch to RUN.
10. If the computer doesn't skip, check the registers.

Correct Register Contents:

A	=	00000000
B	=	10000000
X	=	77777777
P	=	00701
C	=	04020400
Overflow	=	0

USE: (cont.)

11. If an error occurs, set Bpts 1 and 2 and use the oscilloscope.
12. If Bpt 1 is set, move the mode switch to RUN; the test repeats.
13. If the operator wishes to repeat the entire Semi-automatic Instruction Diagnostic, set Bpt 4.
14. Move the mode switch to RUN. If Bpt 4 is reset, the program loads the Automatic portion of the Instruction diagnostic.

```
                           *
                           *      LOADER FOR 930 SEMI-AUTOMATIC INSTRUCTION DIAGNOSTIC
                           *
             00002                ORG     2
00002  2 32 00012                 WIM     10,2
00003  0 41 00002                 BRX     2
00004  0 32 00176                 WIM     126
00005  0 32 00177                 WIM     127
00006  0 71 00010                 LDX     8
00007  0 01 00176                 BRU     126
00010  00077142                   OCT     77142
00011  00000000                   OCT     0
00012        00164                BSS     116
```

USE: (cont.)

III. AUTOMATIC INSTRUCTION TEST PROGRAM

The Automatic Instruction test is a "tight-looped" program that performs and tests those instructions not tested by the Manual or Semi-automatic Tests. The Instructions to be tested and all the data needed to perform the tests are in twelve word data blocks in memory. Each data block is associated with only one Instruction. The program, under index control, operates successively on each of the Data Blocks.

Each Data Block contains the register settings prior to performing the Instructions, and the resulting register settings after the Instruction execution. Appendix 3B contains a description of the Data Blocks.

The program operates on a Data Block by setting the registers with the proper data, executing the Instruction of the Data Block and comparing the resulting register contents against those in the Data Block. When an error occurs, the computer halts. Appendix 3B contains a description of the error halts.

The following breakpoint settings provide special operator control.

Breakpoint 1 Set	=	Repeat the test now in progress
Reset	=	Normal operation
Breakpoint 2 Set	=	By-pass error halts
Reset	=	Normal operation
Breakpoint 3 Set	=	Halt after operating on each Data Block. The X Register contains the location of the Data Block just operated on
Reset	=	Normal operation

A sync bit is provided in bit zero of all Instructions being tested. With Breakpoints 1 and 2 set, the program repeatedly performs each test without any halts; only the Instruction being tested has a sync bit. This provides a way to synchronize an oscilloscope with the test loop to aid the operator in observing signal levels of any circuitry in question.

USE: (cont.)

Also provided to aid the operator in debugging is a description of each Instruction in terms of the logic equations which make up the Instruction logic. Using these descriptions along with program provided information when an error halt occurs, the operator can quickly localize the error.

The Data Blocks and their contents are shown on the program listings starting at octal location 130. The program tests the Instructions in the order their Data Blocks appear in the coding sheets.

Special handling of Multiply and Divide errors is described in Appendices 3C and 3D.

APPENDIX 3A

ERROR HALTS

Errors such as incorrect register contents, incorrect overflow indications, and incorrect skip performance are indicated by halts with Breakpoint 2 reset. The following defines the contents of the registers when such a halt occurs.

P register	Location+1 of the error halt
A register	Erroneous contents of the defined register after the instruction has been performed
B register	Correct results of the defined register after the instruction has been performed
$X_{(10-23)}$	First location of data pertinent to instruction being tested
$C_{(10-23)}$	Same as X Register for skip or not skip errors only

The following table defines the halt locations:

P Register	Definition of error	Register in which data address is found
26	Failed to SKIP when SKIP should occur	C
37	SKIPPED when no SKIP should occur	C
50	A Register error	X
56	B Register error	X
64	X Register error	X
72	Memory Register error	X
77	Overflowed when it should not occur	X
105	Failed to Overflow when it should occur	X

APPENDIX 3B

DATA BLOCKS

The Instruction Diagnostic is a tight-looped program that operates under Index Control on a field of data consisting of Data Blocks. These are 12-word blocks that contain the Instruction to be tested and all the data needed to perform the test of the Instruction. The following is a description of a Data Block (assume that the first location of the Block is B):

Location	Contents
B+10	Address of Data Block with Indirect Address bit for indexing*
+1	Instruction
+2	Original contents of A
+3	Resultant contents of A after performing Instruction
+4	Original contents of B
+5	Resultant contents of B
+6	Original contents of X
+7	Resultant contents of X
+8	Original contents of memory (the operand)**
+9	Resultant contents of memory
+10	Determines if SKIP should or should not occur*** (SKIP or NO SKIP)
+11	Determines if overflow should or should not occur (OVFLO or NOVFLO)

*The Indirect address bit is necessary to allow X9=1 and force a BRX to branch. The last Data Block has no Ia bit and allows the BRX to fall through.

**For those instructions in which an operand address has no meaning (e.g., Register exchange, shift, no-op, etc.), this word could contain an EOM which could be necessarily executed prior to performing the Instruction to be tested (e.g., it is necessary to perform an EOM prior to testing the BRTW instruction).

***If the contents of this location are tagged with a "4", the overflow toggle will be set prior to performing the instruction to be tested.

APPENDIX 3C

MULTIPLY AND THE RIGHT SHIFT ADDER

The following sets of constants allow the user to diagnose completely any Right Shift Adder and/or multiply problem. If a failure is suspected, the user should reset all breakpoints and load the tape using the Diagnostic load procedure beginning at the Automatic Instruction Diagnostic.

If the program halts at a multiply error halt, determine the pertinent constant set via Appendix 3A. Then set Breakpoints 1 and 2 and place the mode switch in RUN. This repeatedly performs the current test with the error halt inhibited. The user can now set up the oscilloscope and follow the test through each T-time to determine the exact circuit area

FIGURE MUL 1 A*M = 77652714 03224436 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 74121733	0 00000000	00000000	01305135	
T7	1 36050755	0 01305135	00000000	01305135	*
T6	1 57024366	0 01047503	01000110	01305135	*
T5	1 67412173	0 00023605	01000100	01305135	
T4	1 73605075	0 01716077	00001000	01305135	*
T3	1 75702436	0 01244564	01010010	01305135	*
T2	1 36741217	0 00126276	01000000	01305135	
T1	1 17360507	0 01750264	00010010	01305135	*
TC	1 07570243	0 01275263	01000010	01305135	*
TR	1 43674121	0 02433662	00010010	01305135	*
TP	1 21736050	0 02526062	00001010	01305135	*
T8	1 10757024	0 01253435	00000000	01305135	
T7	1 44367412	0 00525616	00000000	01305135	
T6	1 22173605	0 00252707	00000000	01305135	
T5	1 51075702	0 01422470	00010010	01305135	*
T4	1 24436741	0 00615230	00000010	01305135	
T3	1 12217360	0 01603655	00010000	01305135	*
T2	1 45107570	0 00705726	00000000	01305135	
T1	1 22443674	0 00342753	00000000	01305135	
T0	1 51221736	0 00161365	00000000	01305135	
TR	1 64510757	0 00070572	00000000	01305135	
TP	1 32244367	0 01331322	00010110	01305135	*
T8	1 15122173	0 01065742	01000010	01305135	*
T7	1 06451075	0 02337012	00001110	01305135	*
T6	1 03224436	1 77542613	00110101	76472642	*
TP	1 77652714	1 03224436	00110101	76472642	

FIGURE MUL 2 A*M = 02200343 65100504 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 45666162	0 00000000	00000000	75175331	
T7	1 22733071	0 00000000	00000000	75175331	
T6	1 11355434	1 75175331	00000000	75175331	*
T5	1 44566616	1 76476554	00000000	75175331	
T4	1 22273307	1 77237266	00000000	75175331	
T3	1 11135543	1 64604064	10111000	75175331	*
T2	1 04456661	1 63483763	10110000	75175331	*
T1	1 42227330	1 52756222	10101100	75175331	*
T0	1 21113554	1 75327551	00100000	75175331	
TR	1 50445666	1 76513664	00100000	75175331	
TP	1 24222733	1 77205732	00100000	75175331	
T8	1 12111355	1 64637206	10101100	75175331	*
T7	1 05044566	1 63446474	10110000	75175331	*
T6	1 02422273	1 75666636	00000000	75175331	
T5	1 01211135	1 63020640	11110010	75175331	*
T4	1 00504456	1 63141655	10110000	75175331	*
T3	1 40242227	1 75424726	00100000	75175331	
T2	1 20121113	1 63747604	10100100	75175331	*
T1	1 10050445	1 62110273	11111000	75175331	*
T0	1 44024222	1 62605066	10101000	75175331	*
T8	1 22012111	1 75302033	00001000	75175331	
TP	1 51005044	1 63656746	10100000	75175331	*
T8	1 24402422	1 75767363	00000000	75175331	
T7	1 52201211	1 76773571	00000000	75175331	
T6	1 65100504	0 21177282	11001111	02602446	*
TP	1 02200343	0 65100504	11001111	02602446	

107

FIGURE MUL 3

A*M = 01225457 06470510 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 76536132	0 00000000	00000000	40451332	
T7	1 37257055	0 00000000	00000000	40451332	
T6	1 17527426	1 40451332	00000000	40451332	*
T5	1 07653613	1 60224555	00000000	40451332	
T4	1 43725705	1 30563510	00000110	40451332	*
T3	1 21752742	1 14642242	00101000	40451332	*
T2	1 10765361	1 46361521	00000000	40451332	
T1	1 44372570	1 23641102	00101100	40451332	*
TC	1 22176274	1 51620001	00101100	40451332	
TR	1 61076536	1 64750440	00000000	40451332	
TP	1 24437257	1 72364220	00000000	40451332	
T8	1 12217527	1 35543442	00100000	40451332	*
T7	1 05107653	1 16372153	01001000	40451332	*
T6	1 42443725	1 07146717	01100100	40451332	*
T5	1 61221752	1 04574731	00000010	40451332	*
T4	1 70510765	1 42276350	00000010	40451332	
T3	1 34244372	1 21500422	00110100	40451332	*
T2	1 16122175	1 50604251	00100000	40451332	
T1	1 47061076	1 24713456	00100000	40451332	*
TC	1 23424437	1 52305627	00100000	40451332	
TR	1 61612217	1 25653245	00001000	40451332	*
TP	1 64705107	1 12377454	01000000	40451332	*
T8	1 32342443	1 05140150	01111010	40451332	*
T7	1 15161221	1 03575722	00000100	40451332	*
T6	1 06470510	0 70114456	11111001	37326445	*
TP	1 01225457	0 06470510	11111001	37326445	

FIGURE MUL 4

A*M = 14411426 44633734 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37124376	0 00000000	00000000	14665211	
T7	0 17452177	0 00000000	00000000	14665211	
T6	0 07625077	0 14665211	00000000	14665211	*
T5	0 43712437	0 12117715	11100000	14665211	*
T4	0 61745217	0 25364157	01011000	14665211	*
T3	0 70762507	0 27753670	00110010	14665211	*
T2	0 34371243	0 27616151	01101000	14665211	*
T1	0 56174521	0 20124675	11110000	14665211	*
T0	0 67076250	0 20373547	11010000	14665211	*
TR	0 73437124	0 14571663	00010000	14665211	
TP	0 75617452	0 06270731	00010000	14665211	
T8	0 76707625	0 03130354	00010000	14665211	
T7	0 37343712	0 15245377	01100000	14665211	*
T6	0 57561745	0 06162577	01000000	14665211	
T5	0 67670762	0 17256400	01100110	14665211	*
T4	0 33734371	0 07167244	01000000	14665211	
T3	0 15756174	0 10650733	10110000	14665211	*
T2	0 46767076	0 00360355	10010000	14665211	
T1	0 63373437	0 04174166	00000000	14665211	
T0	0 31575617	0 16653204	00110100	14665211	*
TR	0 14676707	0 13156753	11100000	14665211	*
TP	0 46337343	0 26704576	00110000	14665211	*
T8	0 23157561	0 27263400	01010110	14665211	*
T7	0 11467670	0 20712055	10111000	14665211	*
T6	0 44633734	0 14301426	00110000	63112566	
TP	0 14411426	0 44633734	00110000	63112566	

FIGURE MUL 5

A*M = 72505325 45501024 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 55340547	0 00000000	00000000	11177146	
T7	1 26560263	0 11177146	00000000	11177146	*
T6	1 13270131	0 15566521	00110110	11177146	*
T5	1 45534054	0 17026452	01110010	11177146	*
T4	1 22656026	0 07057221	01000010	11177146	
T3	1 51327013	0 03027514	01000000	11177146	
T2	1 24553405	0 13102704	00110110	11177146	*
T1	1 12265602	0 16674544	00010010	11177146	*
T0	1 05132701	0 07332266	00010000	11177146	
TR	1 02455340	0 14650271	00110010	11177146	*
TP	1 41226560	0 06360130	00010010	11177146	
T8	1 20513270	0 03174050	00000010	11177146	
T7	1 10245534	0 01476020	00000010	11177146	
T6	1 04122656	0 00637014	00000000	11177146	
T5	1 02051327	0 00317406	00000000	11177146	
T4	1 01024553	0 11236741	00110010	11177146	*
T3	1 40412265	0 15652422	00110110	11177146	*
T2	1 20205132	0 17060313	01110110	11177146	*
T1	1 50102455	0 07074101	01000110	11177146	
T0	1 64041226	0 15125242	00110010	11177146	*
TR	1 32020513	0 06416525	00100000	11177146	
TP	1 55010245	0 14336310	00110110	11177146	*
T8	1 26404122	0 17312356	00110000	11177146	*
T7	1 13202051	0 07501167	00110000	11177146	
T6	1 45501024	1 61504224	11001101	66600631	*
TP	1 72505325	1 45501024	11001101	66600631	

FIGURE MUL 6 A*M = 46133723 57100650 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37252204	0 00000000	00000000	45502225	
T7	0 17525102	0 00000000	00000000	45502225	
T6	0 07652441	0 00000000	00000000	45502225	
T5	0 03725220	1 45502225	00000000	45502225	*
T4	0 41752510	1 62641112	00000000	45502225	
T3	0 20765244	1 71320445	00000000	45502225	
T2	0 50372522	1 74550222	00000000	45502225	
T1	0 24175251	1 76264111	00000000	45502225	
TC	0 52076524	1 34634261	10000010	45502225	*
TR	0 65037252	1 52316134	10000000	45502225	
TP	0 32417525	1 61147056	10000000	45502225	
T8	0 15207652	1 31165644	11000010	45502225	*
T7	0 06503725	1 50072726	11000000	45502225	
T6	0 03241752	1 35137570	01000010	45502225	*
T5	0 01520765	1 56057670	01000010	45502225	
T4	0 00650372	1 24121165	11011000	45502225	*
T3	0 40324175	1 56454072	00001000	45502225	
T2	0 20152076	1 24720652	10010010	45502225	*
T1	0 10065037	1 56354321	00000010	45502225	
TC	0 44032417	1 24660371	10010010	45502225	*
TR	0 62015207	1 13036325	11000100	45502225	*
TP	0 71006503	1 17511437	00010000	45502225	*
T8	0 74403241	1 04352034	11001010	45502225	*
T7	0 36201520	1 03267647	11000000	45502225	*
T6	0 57100650	1 45133723	01000000	32275552	
TP	0 46133723	1 57100650	01000000	32275552	

FIGURE MUL 7 A*M = 06437742 35001420 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 11565130	0 00000000	00000000	25453363	
T7	0 04672454	0 00000000	00000000	25453363	
T6	0 02335226	0 00000000	00000000	25453363	
T5	0 01156513	0 00000000	00000000	25453363	
T4	0 00467245	0 25453363	00000000	25453363	*
T3	0 40233522	0 37270054	01011100	25453363	*
T2	0 20115651	0 17130466	01010000	25453363	
T1	0 10046724	0 25433516	10100100	25453363	*
TC	0 04023352	0 16655607	00000100	25453363	
TR	0 42011565	0 07326743	00000000	25453363	
TP	0 61004672	0 20126644	11100100	25453363	*
T8	0 30402335	0 14413362	00100000	25453363	
T7	0 14201156	0 23610054	10111100	25453363	*
T6	0 06100467	0 15740466	00010000	25453363	
T5	0 03040233	0 23337516	11100100	25453363	*
T4	0 01420115	0 33662262	10011010	25453363	*
T3	0 00610046	0 46300010	01111110	25453363	*
T2	0 00304023	0 23504440	00100010	25453363	
T1	0 00142011	0 36355507	01000100	25453363	*
TC	0 40061004	0 34131266	11111000	25453363	*
TR	0 20030402	0 12410133	10111000	25453363	
TP	0 50014201	0 01240455	10010000	25453363	
T8	0 64006100	0 21177501	11000110	25453363	*
T7	0 72003040	0 14077604	01000100	25453363	
T6	0 35001420	0 06437742	00000000	52324414	
TP	0 06437742	0 35001420	00000000	52324414	

FIGURE MUL 8 A*M = 04045712 32372146 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 72672751	0 00000000	00000000	46470673	
T7	1 35335364	1 46470673	00000000	46470673	*
T6	1 56556572	1 63234335	00000000	46470673	
T5	1 67267275	1 71516156	00000000	46470673	
T4	1 33533536	1 32237652	11100110	46470673	*
T3	1 15655657	1 51557761	10000010	46470673	
T2	1 46726727	1 36257567	01101100	46470673	*
T1	1 63353363	1 15150126	11111100	46470673	*
T0	1 31565565	1 11520306	10101100	46470673	*
TR	1 14672672	1 16300476	01101000	46470673	*
TP	1 06335335	1 47500637	00100000	46470673	
T8	1 43156556	1 21370102	11001110	46470673	*
T7	1 21467267	1 54174405	01000100	46470673	
T6	1 50633533	1 24066035	11101100	46470673	*
T5	1 64315665	1 14063241	11101110	46470673	*
T4	1 72146726	1 10062757	11100100	46470673	*
T3	1 75063353	1 40471327	10000100	46470673	
T2	1 76431565	1 22624406	10101100	46470673	*
T1	1 37214672	1 13742436	10101100	46470673	*
T0	1 17506335	1 41721657	10100000	46470673	
TR	1 42643156	1 22400512	11101110	46470673	*
TP	1 23721467	1 55640601	00000110	46470673	
T8	1 51750633	1 24310137	11101100	46470673	*
T7	1 64764315	1 14274302	11001110	46470673	*
T6	1 32372146	0 03035701	01010011	31307104	*
TP	1 04045712	0 32372146	01010011	31307104	

FIGURE MUL 9 A*M = 10406001 20330000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 51007400	0 00000000	00000000	64114650	
T7	1 24403600	0 00000000	00000000	64114650	
T6	1 12201700	0 00000000	00000000	64114650	
T5	1 05100740	0 00000000	00000000	64114650	
T4	1 02440360	0 00000000	00000000	64114650	
T3	1 01220170	0 00000000	00000000	64114650	
T2	1 00510074	0 00000000	00000000	64114650	
T1	1 00244036	0 00000000	00000000	64114650	
TC	1 00122017	0 00000000	00000000	64114650	
TR	1 00051007	1 64114650	00000000	64114650	*
TP	1 00024403	1 56152174	00011000	64114650	*
T8	1 00012201	1 43105246	10101100	64114650	*
T7	1 00005100	1 41617733	10000100	64114650	*
T6	1 40002440	1 64707715	00000100	64114650	
T5	1 60001220	1 72343706	00000100	64114650	
T4	1 30000510	1 75161703	00000100	64114650	
T3	1 54000244	1 76470701	00000100	64114650	
T2	1 66000122	1 77234300	00000100	64114650	
T1	1 33000051	1 77516100	00000100	64114650	
TC	1 15400024	1 53753750	10010000	64114650	*
TR	1 06600012	1 51761764	10010000	64114650	
TP	1 03300005	1 74774772	00000000	64114650	
T8	1 01540002	1 52402145	10111100	64114650	*
T7	1 40660001	1 61245422	10000100	64114650	
T6	1 20330000	0 07305770	01100011	13663127	*
TP	1 10406001	0 20330000	01100011	13663127	

FIGURE MUL10 A*M = 13604526 64505442 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 15447261	0 00000000	00000000	33556341	
T7	0 06623530	0 33556341	00000000	33556341	*
T6	0 43311654	0 15667160	00000000	33556341	
T5	0 21544726	0 06733470	00000000	33556341	
T4	0 10662353	0 03355634	00000000	33556341	
T3	0 04331165	0 34234257	01111000	33556341	*
T2	0 42154472	0 41230060	11111010	33556341	*
T1	0 21066235	0 24150434	01010000	33556341	
T0	0 10433116	0 45146557	01100000	33556341	*
TR	0 44215447	0 22023267	01100000	33556341	
TP	0 62106623	0 44127074	01101000	33556341	*
T8	0 31043311	0 55262377	01010000	33556341	*
T7	0 54421544	0 52603430	10110110	33556341	*
T6	0 26210662	0 21345650	10000010	33556341	
T5	0 13104331	0 14562720	00000010	33556341	
T4	0 05442154	0 31747615	10100100	33556341	*
T3	0 42621066	0 10723746	10100000	33556341	
T2	0 21310433	0 00311763	10100000	33556341	
T1	0 50544215	0 37752232	00011100	33556341	*
T0	0 24262106	0 42447016	11101100	33556341	*
TR	0 12131043	0 25663047	00001000	33556341	
TP	0 45054421	0 45400364	01110000	33556341	*
T8	0 22426210	0 56712433	00110100	33556341	*
T7	0 51213104	0 27301255	00110000	33556341	
T6	0 64505442	0 13604526	00100000	44221436	
TP	0 13604526	0 64505442	00100000	44221436	

FIGURE MUL11

A*M = 77163427 76765550 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 03532233	0 00000000	00000000	71205134	
T7	0 01655115	1 71205134	00000000	71205134	*
T6	0 00726446	1 65707502	00000110	71205134	*
T5	0 00353223	1 72743605	00000100	71205134	
T4	0 40165511	1 66566076	00001000	71205134	*
T3	0 20072644	1 64470563	00010010	71205134	*
T2	0 50035322	1 72230275	00010000	71205134	
T1	0 64016551	1 75110136	00010000	71205134	
TC	0 32007264	1 67655103	00000110	71205134	*
TR	0 55003532	1 73726405	00000100	71205134	
TP	0 66401655	1 75753242	00000000	71205134	
T8	0 33200726	1 67162655	01010000	71205134	*
T7	0 55500353	1 73075326	01000000	71205134	
T6	0 26640165	1 67233607	00010100	71205134	*
T5	0 53320072	1 64726077	00001000	71205134	*
T4	0 65550035	1 72353437	00000000	71205134	
T3	0 72664016	1 66362743	00010010	71205134	*
T2	0 75332007	1 73175365	00000000	71205134	
T1	0 76555003	1 66673626	00010100	71205134	*
T0	0 37266401	1 64546007	00001100	71205134	*
TR	0 57633200	1 63460577	00010000	71205134	*
TP	0 67655500	1 71634277	00000000	71205134	
T8	0 73726640	1 74716137	00000000	71205134	
T7	0 75753320	1 76347057	00000000	71205134	
T6	0 76765550	1 77163427	00000000	06572643	
TP	0 77163427	1 76765550	00000000	06572643	

FIGURE MUL12 A*M = 76723246 25616366 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 22541417	0 00000000	00000000	76106725	
T7	0 11260607	1 76106725	00000000	76106725	*
T6	0 44530303	1 65141277	10011000	76106725	*
T5	0 62254141	1 64563054	10011010	76106725	*
T4	0 31126060	1 64374757	10010000	76106725	*
T3	0 54453030	1 76172367	00010000	76106725	
T2	0 66225414	1 77071173	00010000	76106725	
T1	0 73112606	1 77430476	00010000	76106725	
T0	0 75445303	1 77610236	00010000	76106725	
TR	0 36622541	1 65016034	11001010	76106725	*
TP	0 17311260	1 65105347	10011000	76106725	*
T8	0 47544530	1 76446163	00001000	76106725	
T7	0 63662254	1 77223471	00000000	76106725	
T6	0 71731126	1 77511634	00000000	76106725	
T5	0 34754453	1 77644716	00000000	76106725	
T4	0 16366225	1 65020264	11011010	76106725	*
T3	0 07173112	1 65112453	10011010	76106725	*
T2	0 43475445	1 76441621	00010010	76106725	
T1	0 61636622	1 65322631	10011010	76106725	*
T0	0 70717311	1 76555710	00000010	76106725	
TR	0 34347544	1 65364675	10011000	76106725	*
TP	0 56163662	1 76576736	00000000	76106725	
T8	0 27071731	1 77277357	00000000	76106725	
T7	0 53434754	1 65635404	10011110	76106725	*
T6	0 25616366	1 76712246	00011000	01671002	
TP	0 76723246	1 25616366	00011000	01671002	

FIGURE MUL13 A*M = 56603537 21726302 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 53451005	0 00000000	00000000	33012655	
T7	1 25624402	0 33012655	00000000	33012655	*
T6	1 52712201	0 15405326	00000000	33012655	
T5	1 25345100	0 31614320	10001110	33012655	*
T4	1 12562440	0 10706514	10000100	33012655	
T3	1 05271220	0 00343206	10000100	33012655	
T2	1 02534510	0 04161543	00000000	33012655	
T1	1 41256244	0 02070661	00000000	33012655	
T0	1 60527122	0 01034330	00000000	33012655	
TR	1 30253451	0 00416154	00000000	33012655	
TP	1 14125624	0 33211633	00010110	33012655	*
T8	1 46052712	0 15500751	00010010	33012655	
T7	1 63025345	0 06644360	00000010	33012655	
T6	1 31412562	0 36334741	00000110	33012655	*
T5	1 54605271	0 17156324	00000100	33012655	
T4	1 26302534	0 32471067	10011000	33012655	*
T3	1 53141256	0 11230033	10011000	33012655	
T2	1 65460527	0 00510415	10010000	33012655	
T1	1 72630253	0 37262053	00001010	33012655	*
T0	1 75314125	0 42643206	10001100	33012655	*
TR	1 36646052	0 50274710	10000110	33012655	*
TP	1 17263025	0 20136300	10000110	33012655	
T8	1 07531412	0 47061051	00011010	33012655	*
T7	1 43654605	0 23434020	00001010	33012655	
T6	1 21726302	1 55573536	01010001	44765122	*
TP	1 56603537	1 21726302	01010001	44765122	

FIGURE MUL14 A*M = 77040237 52035164 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 15101573	0 00000000	00000000	75556616	
T7	0 06440675	1 75556616	00000000	75556616	*
T6	0 03220336	1 63335115	11111010	75556616	*
T5	0 41510157	1 75112042	01111010	75556616	
T4	0 20644067	1 64667233	10001010	75556616	*
T3	0 50322033	1 63002737	11110000	75556616	*
T2	0 64151015	1 63513165	10111010	75556616	*
T1	0 72064406	1 62460604	11010110	75556616	*
T0	0 35032203	1 75634346	00000000	75556616	
TR	0 16415101	1 63464771	11010010	75556616	*
TP	0 47206440	1 63704116	10111100	75556616	*
T8	0 23503220	1 75706407	00100100	75556616	
T7	0 51641610	1 76703243	00100000	75556616	
T6	0 64720644	1 77301521	00100000	75556616	
T5	0 72350322	1 77500650	00100000	75556616	
T4	0 35164151	1 77600324	00100000	75556616	
T3	0 16472064	1 64416760	11100010	75556616	*
T2	0 07235032	1 76647374	00000000	75556616	
T1	0 03516415	1 77323576	00000000	75556616	
T0	0 01647206	1 64227405	11101110	75556616	*
TR	0 40723503	1 76553246	00001000	75556616	
TP	0 20351641	1 64734731	10110010	75556616	*
T8	0 50164720	1 63170176	11011000	75556616	*
T7	0 24072350	1 75070477	01010000	75556616	
T6	0 52035164	1 76030237	01010000	02221161	
TP	0 77040237	1 52035164	01010000	02221161	

FIGURE MUL15 A*M = 00066032 51624720 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 71042702	0 00000000	00000000	77406364	
T7	1 34421341	0 00000000	00000000	77406364	
T6	1 16210560	1 77406364	00000000	77406364	*
T5	1 07104270	1 77603172	00000000	77406364	
T4	1 03442134	1 77701475	00000000	77406364	
T3	1 41621056	1 77740636	00000000	77406364	
T2	1 20710427	1 77760317	00000000	77406364	
T1	1 50344213	1 66376423	11000110	77406364	*
TC	1 64162105	1 66175531	11010110	77406364	*
TR	1 72071042	1 66000204	11111100	77406364	*
TP	1 35034421	1 77444542	00000000	77406364	
T8	1 16416210	1 66220545	11010100	77406364	*
T7	1 47207104	1 77514222	00000100	77406364	
T6	1 23503442	1 77646151	00000000	77406364	
T5	1 51641621	1 77723064	00000000	77406364	
T4	1 24720710	1 66357716	11000100	77406364	*
T3	1 12350344	1 77567707	00000100	77406364	
T2	1 45164162	1 77673703	00000100	77406364	
T1	1 62472071	1 77735701	00000100	77406364	
TC	1 71235034	1 66354364	11011000	77406364	*
TR	1 34516416	1 77562572	00010000	77406364	
TP	1 16247207	1 77675275	00000000	77406364	
T8	1 47123503	1 66334012	11011110	77406364	*
T7	1 23451641	1 66160735	11010100	77406364	*
T6	1 51624720	0 77765021	00101011	00371413	*
TP	1 00066032	0 51624720	00101011	00371413	

FIGURE MUL16 A*M = 03223757 67567714 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 17457571	0 00000000	00000000	06577626	
T7	0 07627674	0 06577626	00000000	06577626	*
T6	0 03713736	0 03277713	00000000	06577626	
T5	0 41745757	0 01537745	00000000	06577626	
T4	0 60762767	0 06346500	01111110	06577626	*
T3	0 30371373	0 01316522	11111010	06577626	*
T2	0 14174575	0 03702473	10111010	06577626	*
T1	0 46076276	0 03574467	11011000	06577626	*
T0	0 63037137	0 05272633	01010000	06577626	
TR	0 71417457	0 01630133	10111010	06577626	*
TP	0 74607627	0 02457207	11101100	06577626	*
T8	0 76303713	0 03357761	11110010	06577626	*
T7	0 77141745	0 04722512	10111110	06577626	*
T6	0 37460762	0 04104537	11111000	06577626	*
T5	0 57630371	0 06406657	00100000	06577626	
T4	0 67714174	0 01732145	10111010	06577626	*
T3	0 73746076	0 04711466	00110000	06577626	
T2	0 35763037	0 02300633	00110000	06577626	
T1	0 56771417	0 07773133	00011010	06577626	*
T0	0 67374607	0 01471607	11110100	06577626	*
TR	0 73576303	0 03777561	10101010	06577626	*
TP	0 75677141	0 03527012	11111110	06577626	*
T8	0 36737460	0 04007677	11110000	06577626	*
T7	0 57357630	0 06447737	00000000	06577626	
T6	0 67567714	0 03223757	00000000	71200151	
TP	0 03223757	0 67567714	00000000	71200151	

FIGURE MUL17 A*M = 57066216 56071300 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 53075604	0 00000000	00000000	31710530	
T7	1 25436702	0 00000000	00000000	31710530	
T6	1 12617341	0 00000000	00000000	31710530	
T5	1 05307560	0 31710530	00000000	31710530	*
T4	1 02543670	0 14744254	00000000	31710530	
T3	1 01261734	0 06362126	00000000	31710530	
T2	1 00530756	0 03171053	00000000	31710530	
T1	1 40254367	0 01474425	00000000	31710530	
TC	1 60126173	0 31546742	01000000	31710530	*
TR	1 30053075	0 46173011	01001100	31710530	*
TP	1 54025436	0 54306574	01100000	31710530	*
T8	1 26012617	0 26503276	00100000	31710530	
T7	1 13005307	0 44211267	01001000	31710530	*
T6	1 45402543	0 53415663	01000000	31710530	*
T5	1 62601261	0 57116461	01001000	31710530	*
T4	1 71300530	0 51757360	10001000	31710530	*
T3	1 34540254	0 20767170	10001000	31710530	
T2	1 16260126	0 14373074	00001000	31710530	
T1	1 07130053	0 06175036	00001000	31710530	
TC	1 03454025	0 34707547	00100000	31710530	*
TR	1 41626012	0 47313313	01001100	31710530	*
TP	1 60713005	0 23145105	01001100	31710530	
T8	1 70345402	0 43773632	00000000	31710530	*
T7	1 34162601	0 21775715	00000000	31710530	
T6	1 56071300	1 56755105	00111111	46067247	*
TP	1 57066216	1 56071300	00111111	46067247	

FIGURE MUL18 A*M = 12145626 04303070 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 55617111	0 00000000	00000000	56055074	
T7	1 26707444	1 56055074	00000000	56055074	*
T6	1 13343622	1 67026436	00000000	56055074	
T5	1 05561711	1 73413217	00000000	56055074	
T4	1 42670744	1 43652573	10010010	56055074	*
T3	1 61334362	1 65721271	00010010	56055074	
T2	1 70556171	1 72754530	00000010	56055074	
T1	1 34267074	1 43333254	10110100	56055074	*
TC	1 16133436	1 65511566	00110000	56055074	
TR	1 07055617	1 72600673	00110000	56055074	
TP	1 43426707	1 43311321	10110110	56055074	*
T8	1 61613343	1 33665600	10000110	56055074	*
T7	1 30705561	1 37707730	00100110	56055074	*
T6	1 14342670	1 25050014	11011100	56055074	*
T5	1 06161334	1 56020446	01010000	56055074	
T4	1 03070556	1 67414223	00000000	56055074	
T3	1 41434267	1 73606111	00000000	56055074	
T2	1 60616133	1 43750030	10010110	56055074	*
T1	1 30307055	1 33745054	10100100	56055074	*
TC	1 14143426	1 37077452	01000110	56055074	*
TR	1 06061613	1 57037661	01000010	56055074	
TP	1 43030705	1 36064720	10010110	56055074	*
T8	1 21414342	1 31003400	10110110	56055074	*
T7	1 10606161	1 50445644	10000000	56055074	
T6	1 04303070	0 11144625	01001001	21722703	*
TP	1 12145626	0 04303070	01001001	21722703	

FIGURE MUL19 A*M = 76276151 24566024 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 72447427	0 00000000	00000000	11470206	
T7	1 35223613	0 11470206	00000000	11470206	*
T6	1 16511705	0 15224301	01100010	11470206	*
T5	1 47244742	0 10542342	10100010	11470206	*
T4	1 23522361	0 00221165	10100000	11470206	
T3	1 51651170	0 15540670	00100010	11470206	*
T2	1 24724474	0 06620330	00100010	11470206	
T1	1 12352236	0 03350150	00000010	11470206	
TC	1 05165117	0 01564060	00000010	11470206	
TR	1 02472447	0 11262232	01100010	11470206	*
TP	1 01235223	0 16561327	00100000	11470206	*
T8	1 40516511	0 10720751	10100010	11470206	*
T7	1 60247244	0 21000576	01100000	11470206	*
T6	1 30123522	0 10040277	01000000	11470206	
T5	1 54051651	0 04420137	00000000	11470206	
T4	1 66024724	0 13600255	00100010	11470206	*
T3	1 73012352	0 05740122	00000010	11470206	
T2	1 35405165	0 02760055	00000000	11470206	
T1	1 56602472	0 12760224	00100010	11470206	*
TC	1 27301235	0 05330116	00100000	11470206	
TR	1 13540516	0 13204245	01100010	11470206	*
TP	1 45660247	0 05142126	01000000	11470206	
T8	1 22730123	0 14451251	00100010	11470206	*
T7	1 51354051	0 17654736	00100000	11470206	*
T6	1 24566024	1 65265040	11011111	66307571	*
TP	1 76276151	1 24566024	11011111	66307571	

FIGURE MUL20 A*M = 71173165 72126104 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 36427241	0 00000000	00000000	70726342	
T7	0 17213520	1 70726342	00000000	70726342	*
T6	0 07505650	1 74353161	00000000	70726342	
T5	0 43642724	1 76165470	00000000	70726342	
T4	0 21721352	1 77072634	00000000	70726342	
T3	0 10750565	1 77435316	00000000	70726342	
T2	0 04364272	1 67534001	01011110	70726342	*
T1	0 42172135	1 73252444	01010000	70726342	
T0	0 21075056	1 66057564	01000000	70726342	*
TR	0 10436427	1 73427672	00000000	70726342	
TP	0 04217213	1 65531277	01011000	70726342	*
T8	0 42107505	1 63107471	01100010	70726342	*
T7	0 61043642	1 62021102	01011100	70726342	*
T6	0 30421721	1 71414001	00001100	70726342	
T5	0 54210750	1 64524702	01010100	70726342	*
T4	0 26104364	1 72656301	00000100	70726342	
T3	0 53042172	1 75327100	00000100	70726342	
T2	0 25421075	1 76553400	00000100	70726342	
T1	0 12610436	1 67103102	01111100	70726342	*
T0	0 05304217	1 73005001	01101100	70726342	
TR	0 42542107	1 66761302	00010100	70726342	*
TP	0 21261043	1 63212143	01111000	70726342	*
T8	0 50530421	1 62077723	01000100	70726342	*
T7	0 64254210	1 61355353	01011000	70726342	*
T6	0 72126104	1 70162165	01011000	07051435	
TP	0 71173165	1 72126104	01011000	07051435	

FIGURE MUL21 A*M = 77004325 77001212 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 10150477	0 00000000	00000000	74104173	
T7	0 04064237	1 74104173	00000000	74104173	*
T6	0 42032117	1 62146160	10000110	74104173	*
T5	0 21015047	1 61167227	10000100	74104173	*
T4	0 50406423	1 60577646	10000100	74104173	*
T3	0 24203211	1 60373056	10011100	74104173	*
T2	0 12101504	1 60206152	10100110	74104173	*
T1	0 05040642	1 74143021	00000110	74104173	
TC	0 42420321	1 76061454	00000000	74104173	
TR	0 21210150	1 63134711	10000110	74104173	*
TP	0 50504064	1 75456300	00000110	74104173	
T8	0 24242032	1 76627104	00000100	74104173	
T7	0 12121015	1 77313402	00000100	74104173	
T6	0 05050406	1 63641734	10010100	74104173	*
T5	0 02424203	1 75724716	00000100	74104173	
T4	0 01212101	1 62056572	11000010	74104173	*
T3	0 00505040	1 61523374	10010100	74104173	*
T2	0 00242420	1 74655536	00000100	74104173	
T1	0 00121210	1 76326617	00000100	74104173	
TC	0 40050504	1 77153347	00000000	74104173	
TR	0 60024242	1 77465563	00000000	74104173	
TP	0 70012121	1 77632671	00000000	74104173	
T8	0 74005050	1 63011427	11010100	74104173	*
T7	0 76002424	1 75000653	01010000	74104173	
T6	0 77001212	1 76004325	01000000	03673604	
TP	0 77004325	1 77001212	01000000	03673604	

FIGURE MUL22 A*M = 65577047 53467714 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 61010463	0 00000000	00000000	25712202	
T7	1 30404231	0 25712202	00000000	25712202	*
T6	1 14202114	0 37657303	01000000	25712202	*
T5	1 46101046	0 17327541	01000000	25712202	
T4	1 63040423	0 07153660	01000000	25712202	
T3	1 31420211	0 21777132	10001000	25712202	*
T2	1 14610104	0 31602257	11110000	25712202	*
T1	1 46304042	0 10345127	11000000	25712202	
T0	1 63142021	0 00562453	10000000	25712202	
TR	1 71461010	0 21103427	11100000	25712202	*
TP	1 74630404	0 14001613	01100000	25712202	
T8	1 76314202	0 06440705	00000000	25712202	
T7	1 77146101	0 03220342	00000000	25712202	
T6	1 37463040	0 26422363	01000000	25712202	*
T5	1 57631420	0 13611171	00000000	25712202	
T4	1 67714610	0 05704474	00000000	25712202	
T3	1 33746304	0 02742236	00000000	25712202	
T2	1 15763142	0 01361117	00000000	25712202	
T1	1 46771461	0 00570447	00000000	25712202	
T0	1 63374630	0 25106425	01100000	25712202	*
TR	1 71576314	0 12003212	01100000	25712202	
TP	1 34677146	0 05441505	00000000	25712202	
T8	1 56337463	0 02620642	00000000	25712202	
T7	1 27157631	0 26222523	01000000	25712202	*
T6	1 53467714	1 65576746	00000101	52065575	*
TP	1 65577047	1 53467714	00000101	52065575	

FIGURE MUL23

A*M = 15523762 54111166 U

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62111173	0 00000000	00000000	40344101	
T7	1 31044475	1 40344101	00000000	40344101	*
T6	1 54422236	1 20426141	00100000	40344101	*
T5	1 66211117	1 50253060	00000000	40344101	
T4	1 33104447	1 24461531	00010000	40344101	*
T3	1 55442223	1 12570755	00010000	40344101	*
T2	1 66621111	1 05544467	00100000	40344101	*
T1	1 73310444	1 02266334	01000000	40344101	*
TC	1 35544222	1 41533156	00000000	40344101	
TR	1 16662111	1 60655467	00000000	40344101	
TP	1 47331044	1 30662734	00010000	40344101	*
T8	1 23554422	1 54335356	00000000	40344101	
T7	1 11666211	1 66156567	00000000	40344101	
T6	1 44733104	1 33323374	00110000	40344101	*
T5	1 22355442	1 55515576	00100000	40344101	
T4	1 11166621	1 66606677	00100000	40344101	
T3	1 44473310	1 33607430	00100010	40344101	*
T2	1 22235544	1 55743610	00000010	40344101	
T1	1 11116662	1 66761700	00000010	40344101	
T0	1 04447331	1 73370744	00000000	40344101	
TR	1 02223554	1 35030463	01110000	40344101	*
TP	1 41111666	1 56050231	01010000	40344101	
T8	1 60444733	1 67420114	00010000	40344101	
T7	1 30222355	1 33150147	01010000	40344101	*
T6	1 54111166	0 05423651	10100111	37433676	*
TP	1 15523762	0 54111166	10100111	37433676	

FIGURE MUL24 A*M = 67201543 12433276 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 65503143	0 00000000	00000000	32774725	
T7	1 32641461	0 32774725	00000000	32774725	*
T6	1 55320630	0 47262277	01111000	32774725	*
T5	1 66550314	0 23175537	01000000	32774725	
T4	1 73264146	0 11076657	01000000	32774725	
T3	1 75532063	0 04037327	01000000	32774725	
T2	1 76655031	0 34303470	01111010	32774725	*
T1	1 37326414	0 40572165	11011000	32774725	*
TC	1 57553206	0 24671472	00010000	32774725	
TR	1 27665503	0 12330635	00010000	32774725	
TP	1 53732641	0 37054233	01101010	32774725	*
T8	1 65755320	0 42752446	10111000	32774725	*
T7	1 32766550	0 25321623	00110000	32774725	
T6	1 55373264	0 12514711	00100000	32774725	
T5	1 66575532	0 05206344	00100000	32774725	
T4	1 33276655	0 02543162	00000000	32774725	
T3	1 15537326	0 33155316	01101100	32774725	*
T2	1 06657553	0 15026107	01101100	32774725	
T1	1 43327665	0 31747420	10101010	32774725	*
TC	1 21553732	0 52710131	01111010	32774725	*
TR	1 50665755	0 25700450	00110010	32774725	
TP	1 24332766	0 44770155	01011000	32774725	*
T8	1 52155373	0 22770466	00010000	32774725	
T7	1 25066575	0 43274150	01101010	32774725	*
T6	1 12433276	1 67171442	00010101	45003052	*
TP	1 67201543	1 12433276	00010101	45003052	

FIGURE MUL25

A*M = 16425421 75165600 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 57411100	0 00000000	00000000	43610437	
T7	1 27604440	0 00000000	00000000	43610437	
T6	1 13702220	0 00000000	00000000	43610437	
T5	1 05741110	0 00000000	00000000	43610437	
T4	1 02760444	0 00000000	00000000	43610437	
T3	1 01370222	0 00000000	00000000	43610437	
T2	1 00574111	0 00000000	00000000	43610437	
T1	1 00276044	1 43610437	00000000	43610437	*
TC	1 40137022	1 61704217	00000000	43610437	
TR	1 60057411	1 70742107	00000000	43610437	
TP	1 70027604	1 37171472	01000010	43610437	*
T8	1 34013702	1 57074631	01000010	43610437	
T7	1 56005741	1 67036310	01000010	43610437	
T6	1 27002760	1 37627507	00000100	43610437	*
T5	1 53401370	1 57713603	00000100	43610437	
T4	1 65600574	1 67745741	00000000	43610437	
T3	1 72700276	1 73762760	00000000	43610437	
T2	1 35340137	1 75771370	00000000	43610437	
T1	1 16560057	1 31604123	11101110	43610437	*
TC	1 47270027	1 24112144	01001010	43610437	*
TR	1 23534013	1 15255016	01001110	43610437	*
TP	1 51656005	1 02737401	10000110	43610437	*
T8	1 64727002	1 00167273	11001010	43610437	*
T7	1 72353401	1 44473131	00001010	43610437	
T6	1 75165600	0 16316320	00110101	34167340	*
TP	1 16425421	0 75165600	00110101	34167340	

FIGURE MUL26 A*M = 00764773 16[illegible]132 [illegible]

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 15051535	0 00000000	00000000	02311421	
T7	0 06424656	0 02311421	00000000	02311421	*
T6	0 43212327	0 01144610	00000000	02311421	
T5	0 21505153	0 02773725	00000000	02311421	*
T4	0 50642465	0 03606373	00101000	02311421	*
T3	0 64321232	0 03254116	01001100	02311421	*
T2	0 32150515	0 01126407	01000100	02311421	
T1	0 55064246	0 03364664	00000000	02311421	*
TC	0 26432123	0 01572332	00000000	02311421	
TR	0 13215051	0 02106576	01100000	02311421	*
TP	0 05506424	0 03014610	01000110	02311421	*
T8	0 02643212	0 01006340	01000010	02311421	
T7	0 01321505	0 00003164	01000000	02311421	
T6	0 00550642	0 02712013	00001100	02311421	*
T5	0 40264321	0 01345445	00000000	02311421	
T4	0 60132150	0 02073243	01001000	02311421	*
T3	0 70056064	0 01435121	00001000	02311421	
T2	0 74026432	0 00616050	00001000	02311421	
T1	0 36013215	0 00307424	00000000	02311421	
TC	0 17005506	0 02454233	00001000	02311421	*
TR	0 47402643	0 01226515	00000000	02311421	
TP	0 63601321	0 02024667	01000000	02311421	*
T8	0 71700550	0 03723754	00000000	02311421	*
T7	0 34740264	0 01751766	00000000	02311421	
T6	0 16360132	0 00764773	00000000	75466356	
TP	0 00764773	0 16360132	00000000	75466356	

FIGURE MUL27 A*M = 17312204 00443740 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 57330170	0 00000000	00000000	42220642	
T7	1 27554074	0 00000000	00000000	42220642	
T6	1 13666036	0 00000000	00000000	42220642	
T5	1 05733017	0 00000000	00000000	42220642	
T4	1 02755407	1 42220642	00000000	42220642	*
T3	1 01366603	1 23330163	00001000	42220642	*
T2	1 40573301	1 13774233	00001100	42220642	*
T1	1 60275540	1 07116317	01101100	42220642	*
T0	1 70136660	1 43007507	01100100	42220642	
TR	1 74057330	1 61043603	01000100	42220642	
TP	1 76027554	1 70021741	01000000	42220642	
T8	1 77013666	1 74410760	00000000	42220642	
T7	1 37405733	1 76204370	00000000	42220642	
T6	1 17602755	1 31322736	10000100	42220642	*
T5	1 07701366	1 22771251	00001010	42220642	*
T4	1 43740573	1 51374120	00001010	42220642	
T3	1 21760275	1 26716216	00101100	42220642	*
T2	1 10770136	1 15627301	00001110	42220642	*
T1	1 44374057	1 46713104	00001100	42220642	
T0	1 22176027	1 25566744	00000000	42220642	*
TR	1 11077013	1 14013124	01101100	42220642	*
TP	1 04437405	1 00666754	10000000	42220642	*
T8	1 02217602	1 06553120	00001110	42220642	*
T7	1 01107701	1 43265014	00001100	42220642	
T6	1 00443740	0 16202173	01110011	35557135	*
TP	1 17312204	0 00443740	01110011	35557135	

FIGURE MUL28 A*M = 00355615 23630064 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 03008166	0 00000000	00000000	04737527	
T7	0 01402473	0 00000000	00000000	04737527	
T6	0 00601235	0 04737527	00000000	04737527	*
T5	0 40300516	0 06206372	01111010	04737527	*
T4	0 20140247	0 03547571	00000010	04737527	
T3	0 50060123	0 05512327	01111100	04737527	*
T2	0 64030051	0 07240332	01011010	04737527	*
T1	0 32014024	0 00053200	11011110	04737527	*
T0	0 15006012	0 04421144	00011000	04737527	
TR	0 06403005	0 02214062	00001000	04737527	
TP	0 03201402	0 05035150	01011010	04737527	*
T8	0 01500601	0 02012060	01011010	04737527	
T7	0 00640300	0 05340153	01011010	04737527	*
T6	0 40320140	0 02164461	01000010	04737527	
T5	0 60150060	0 01472234	00000000	04737527	
T4	0 30064030	0 00635116	00000000	04737527	
T3	0 14032014	0 00316447	00000000	04737527	
T2	0 46015006	0 00147223	00000000	04737527	
T1	0 63006403	0 00063511	00000000	04737527	
T0	0 71403201	0 04760363	00011010	04737527	*
TR	0 74601500	0 06223214	01111110	04737527	*
TP	0 36300640	0 03555142	00001010	04737527	
T8	0 17140320	0 01666065	00001000	04737527	
T7	0 47460150	0 00733432	00000000	04737527	
T6	0 23630064	0 00355615	00000000	73040250	
TP	0 00355615	0 23630064	00000000	73040250	

FIGURE MUL29 A*M = 67532402 10014460 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62761652	0 00000000	00000000	24343434	
T7	1 31370725	0 00000000	00000000	24343434	
T6	1 14574352	0 24343434	00000000	24343434	*
T5	1 06276165	0 12161616	00000000	24343434	
T4	1 03137072	0 21333333	10101010	24343434	*
T3	1 41457435	0 14515151	00101010	24343434	
T2	1 60627616	0 22642424	10010100	24343434	*
T1	1 30313707	0 15325252	00000000	24343434	
TC	1 14145743	0 22015151	11101010	24343434	*
TR	1 46062761	0 31702424	10110100	24343434	*
TP	1 23031370	0 44340606	01010100	24343434	*
T8	1 11414574	0 22564343	00000000	24343434	
T7	1 44606276	0 11272161	00000000	24343434	
T6	1 62303137	0 04535070	00000000	24343434	
T5	1 31141457	0 26511060	00111010	24343434	*
T4	1 14460627	0 37654460	00000010	24343434	*
T3	1 06230313	0 33261660	11010010	24343434	*
T2	1 03114145	0 46477360	00001010	24343434	*
T1	1 01446062	0 47573520	00010110	24343434	*
T0	1 00623031	0 23671614	00010100	24343434	
TR	1 00311414	0 35203372	01101010	24343434	*
TP	1 00144606	0 16141171	01001010	24343434	
T8	1 40062303	0 07460070	00001010	24343434	
T7	1 20031141	0 27173074	01001000	24343434	*
T6	1 10014460	1 57422371	00110011	53434343	*
TP	1 67532402	1 10014460	00110011	53434343	

FIGURE MUL30 A*M = 06745073 46420734 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 32527245	0 00000000	00000000	10253726	
T7	0 15253522	0 10253726	00000000	10253726	*
T6	0 06525651	0 04125753	00000000	10253726	
T5	0 43252724	0 12225603	00101110	10253726	*
T4	0 61525352	0 05152345	00001000	10253726	
T3	0 70652565	0 02465562	00000000	10253726	
T2	0 34325272	0 11405517	00101100	10253726	*
T1	0 56152535	0 04642207	00001100	10253726	
T0	0 67065256	0 12574461	00001010	10253726	*
TR	0 73432527	0 05276634	00000000	10253726	
TP	0 35615253	0 12702234	00111010	10253726	*
T8	0 16706525	0 15650440	00011010	10253726	*
T7	0 07343252	0 16103542	01101010	10253726	*
T6	0 03561525	0 07401265	00101000	10253726	
T5	0 41670652	0 13014050	01101010	10253726	*
T4	0 20734325	0 05046420	01000010	10253726	
T3	0 10356152	0 13276132	00001010	10253726	*
T2	0 04167065	0 05537451	00000010	10253726	
T1	0 42073432	0 12022556	01111000	10253726	*
T0	0 21035615	0 05455667	00000000	10253726	
TR	0 50416706	0 12071651	01011010	10253726	*
TP	0 64207343	0 05430320	00011010	10253726	
T8	0 32103561	0 12073472	01001010	10253726	*
T7	0 15041670	0 15601167	00111000	10253726	*
T6	0 46420734	0 06744073	00001000	67524051	
TP	0 06745073	0 46420734	00001000	67524051	

FIGURE MUL31 A*M = 77401510 42064640 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 01242560	0 00000000	00000000	63737373	
T7	0 00521270	0 00000000	00000000	63737373	
T6	0 00250534	0 00000000	00000000	63737373	
T5	0 00124256	0 00000000	00000000	63737373	
T4	0 00052127	0 00000000	00000000	63737373	
T3	0 00025053	1 63737373	00000000	63737373	*
T2	0 40012425	1 54606060	01111110	63737373	*
T1	0 20005212	1 41676767	11010100	63737373	*
TC	0 50002505	1 64333333	01010100	63737373	
TR	0 64001242	1 55410100	01111110	63737373	*
TP	0 32000521	1 66240404	01010100	63737373	
T8	0 15000250	1 56453535	01010100	63737373	*
T7	0 46400124	1 67621616	00010100	63737373	
T6	0 23200052	1 73714747	00000000	63737373	
T5	0 51500025	1 75746363	00000000	63737373	
T4	0 64640012	1 51612464	11110100	63737373	*
T3	0 32320005	1 60341272	11010000	63737373	
T2	0 15150002	1 57413020	01111110	63737373	*
T1	0 06464001	1 67241054	01011000	63737373	
TC	0 03232000	1 57054311	01010110	63737373	*
TR	0 41515000	1 67022100	01010110	63737373	
TP	0 20646400	1 73015004	01000100	63737373	
T8	0 10323200	1 75006442	01000000	63737373	
T7	0 04151500	1 76003221	01000000	63737373	
T6	0 42064640	1 77401510	00000000	14040404	
TP	0 77401510	1 42064640	00000000	14040404	

FIGURE MUL32

A*M = 00361521 66277232 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 76310175	0 00000000	00000000	66471421	
T7	1 37144076	1 66471421	00000000	66471421	*
T6	1 57462037	1 73234610	00000000	66471421	
T5	1 27631017	1 53107725	11100000	66471421	*
T4	1 53714407	1 50574373	10001000	66471421	*
T3	1 65746203	1 56667116	00101100	66471421	*
T2	1 32763101	1 45765520	10100010	66471421	*
T1	1 15371440	1 44423275	11101000	66471421	*
TC	1 46574620	1 66651136	00001000	66471421	
TR	1 23276310	1 73324057	00001000	66471421	
TP	1 51537144	1 75552427	00000000	66471421	
T8	1 64657462	1 76665213	00000000	66471421	
T7	1 72327631	1 77332505	00000000	66471421	
T6	1 75153714	1 55146663	11100000	66471421	*
T5	1 76465746	1 62023331	11100000	66471421	
T4	1 77232763	1 75451554	00000000	66471421	
T3	1 37515371	1 54215207	11101100	66471421	*
T2	1 57646574	1 50130564	11110000	66471421	*
T1	1 27723276	1 60410272	10110000	66471421	
TC	1 13751537	1 74240135	00010000	66471421	
TR	1 45764657	1 54515477	10100000	66471421	*
TP	1 62772327	1 50777250	10001010	66471421	*
T8	1 31375153	1 56761551	00110000	66471421	*
T7	1 54576465	1 45025205	11101100	66471421	*
T6	1 66277232	0 70351410	10010111	11306356	*
TP	1 00361521	0 66277232	10010111	11306356	

FIGURE MUL33 A*M = 06361472 36302432 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 32300533	0 00000000	00000000	07664067	
T7	0 15140255	0 07664067	00000000	07664067	*
T6	0 46460126	0 02516012	11100110	07664067	*
T5	0 23230053	0 05607041	00100010	07664067	
T4	0 51514025	0 01527403	11100110	07664067	*
T3	0 64646012	0 04177624	11000110	07664067	*
T2	0 32323005	0 06477756	00000000	07664067	
T1	0 15151402	0 02013746	11110110	07664067	*
TC	0 06464601	0 05441727	00010100	07664067	
TR	0 43232300	0 01400072	11111010	07664067	*
TP	0 21515140	0 04244431	01000010	07664067	
T8	0 50646460	0 02522210	00000010	07664067	
T7	0 24323230	0 01251100	00000010	07664067	
T6	0 12151514	0 00524444	00000000	07664067	
T5	0 05064646	0 00252222	00000000	07664067	
T4	0 02432323	0 00125111	00000000	07664067	
T3	0 41215151	0 07636423	00100110	07664067	*
T2	0 60506464	0 02533234	11110110	07664067	*
T1	0 30243232	0 05611552	00110010	07664067	
TC	0 14121515	0 02740661	00010010	07664067	
TR	0 46050646	0 00140313	11110110	07664067	*
TP	0 63024323	0 04424101	00100110	07664067	
T8	0 71412151	0 01036163	11100010	07664067	*
T7	0 74605064	0 04633054	10110110	07664067	*
T6	0 36302432	0 06361462	00010010	70113710	
TP	0 06361472	0 36302432	00010010	70113710	

FIGURE MUL34 A*M = 63723734 26006610 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 23402515	0 00000000	00000000	54127724	
T7	0 11601246	1 54127724	00000000	54127724	*
T6	0 04700523	1 66053752	00000000	54127724	
T5	0 02340251	1 47144601	00011110	54127724	*
T4	0 41160124	1 37505660	00111010	54127724	*
T3	0 20470052	1 57606334	00101000	54127724	
T2	0 10234025	1 67743556	00000000	54127724	
T1	0 04116012	1 47000503	01111110	54127724	*
T0	0 42047005	1 63044605	01000100	54127724	
TR	0 61023402	1 46141266	00011000	54127724	*
TP	0 30411601	1 63064133	00001000	54127724	
T8	0 54204700	1 45551371	00011010	54127724	*
T7	0 66102340	1 62660170	00011010	54127724	
T6	0 33041160	1 71334470	00000010	54127724	
T5	0 15420470	1 74556230	00000010	54127724	
T4	0 06610234	1 76267110	00000010	54127724	
T3	0 03304116	1 77133440	00000010	54127724	
T2	0 01542047	1 77455624	00000000	54127724	
T1	0 00661023	1 43745636	10011000	54127724	*
T0	0 00330411	1 31006233	11111010	54127724	*
TR	0 40154204	1 35166445	00011000	54127724	*
TP	0 60066102	1 56477622	00000000	54127724	
T8	0 30033041	1 67237711	00000000	54127724	
T7	0 54015420	1 47636660	00011010	54127724	*
T6	0 26006610	1 63713734	00010000	23650053	
TP	0 63723734	1 26006610	00010000	23650053	

FIGURE MUL35   A*M = 10550561 33016416 0

TIME	A REGISTER		B REGISTER		#C REGISTER	C REGISTER	
T8	1	62360411	0	00000000	00000000	53323417	
T7	1	31170204	1	53323417	00000000	53323417	*
T6	1	54474102	1	65551607	00000000	53323417	
T5	1	66236041	1	72664703	00000000	53323417	
T4	1	73117020	1	40555750	10000010	53323417	*
T3	1	35447410	1	64326750	00000010	53323417	
T2	1	16623604	1	72153374	00000000	53323417	
T1	1	07311702	1	75065576	00000000	53323417	
TC	1	03544741	1	76432677	00000000	53323417	
TR	1	41662360	1	42530746	10010010	53323417	*
TP	1	20731170	1	65250367	00010000	53323417	
T8	1	50354474	1	72520173	00010000	53323417	
T7	1	64166236	1	75254075	00000000	53323417	
T6	1	72073117	1	76526036	00000000	53323417	
T5	1	35035447	1	42576426	10000010	53323417	*
T4	1	16416623	1	30512626	10110010	53323417	*
T3	1	07207311	1	33634726	00000010	53323417	*
T2	1	03503544	1	20231766	11010010	53323417	*
T1	1	01641662	1	54510777	00010000	53323417	
TC	1	40720731	1	66240377	00010000	53323417	
TR	1	60350364	1	46447506	00000110	53323417	*
TP	1	30164166	1	63223607	00000100	53323417	
T8	1	54072073	1	71511743	00000000	53323417	
T7	1	66035035	1	47167370	01001010	53323417	*
T6	1	33016416	0	00440460	10110101	24454360	*
TP	1	10550561	0	33016416	10110101	24454360	

FIGURE MUL36 A*M = 30536721 76064526 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 44436405	0 00000000	00000000	43154757	
T7	1 22217202	1 43154757	00000000	43154757	*
T6	1 51107501	1 61466367	00000000	43154757	
T5	1 64443640	1 33707042	00101110	43154757	*
T4	1 32221720	1 55703065	00101000	43154757	
T3	1 55110750	1 66701032	00101000	43154757	
T2	1 26444364	1 73300015	00101000	43154757	
T1	1 53222172	1 75500406	00100000	43154757	
TC	1 25511075	1 76600203	00100000	43154757	
TR	1 52644436	1 32414050	10101010	43154757	*
TP	1 25322217	1 51246420	10000010	43154757	
T8	1 12551107	1 33677163	00001010	43154757	*
T7	1 45264443	1 10004744	11110110	43154757	*
T6	1 22532221	1 13512375	00111010	43154757	*
T5	1 51255110	1 00065451	11001110	43154757	*
T4	1 64526444	1 44432260	00001010	43154757	
T3	1 32253222	1 62215534	00000000	43154757	
T2	1 15125511	1 71106656	00000000	43154757	
T1	1 06452644	1 37517276	00101010	43154757	*
TC	1 03225322	1 57607133	00101010	43154757	
TR	1 41512551	1 67743051	00001010	43154757	
TP	1 60645264	1 36036707	01100100	43154757	*
T8	1 70322532	1 57457303	00000100	43154757	
T7	1 74151255	1 67627501	00000100	43154757	
T6	1 76064526	0 27536720	01000001	34623020	*
TP	1 30536721	0 76064526	01000001	34623020	

FIGURE MUL37

A*M = 06521556 21500510 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 12254114	0 00000000	00000000	24474123	
T7	0 05126046	0 00000000	00000000	24474123	
T6	0 02453023	0 00000000	00000000	24474123	
T5	0 01225411	0 24474123	00000000	24474123	*
T4	0 40512604	0 36622174	00110000	24474123	*
T3	0 20245302	0 17355076	00000000	24474123	
T2	0 10122541	0 07566437	00000000	24474123	
T1	0 44051260	0 27267332	01100010	24474123	*
TC	0 22024530	0 13173551	01000010	24474123	
TR	0 51012254	0 05075660	01000010	24474123	
TP	0 24405126	0 02036734	01000000	24474123	
T8	0 12202453	0 01417356	00000000	24474123	
T7	0 05101225	0 24273602	01010110	24474123	*
T6	0 02440512	0 36135060	01101010	24474123	*
T5	0 01220245	0 17416034	00101000	24474123	
T4	0 00510122	0 23233531	11110010	24474123	*
T3	0 40244051	0 15151650	01010010	24474123	
T2	0 20122024	0 23464053	10101000	24474123	*
T1	0 50051012	0 15672425	00000000	24474123	
TC	0 64024405	0 06736212	00000000	24474123	
TR	0 32012202	0 27742620	00110010	24474123	*
TP	0 15005101	0 13725314	00100000	24474123	
T8	0 06402440	0 21406661	11100010	24474123	*
T7	0 43201220	0 14243334	01000000	24474123	
T6	0 21500510	0 06521556	00000000	53303654	
TP	0 06521556	0 21500510	00000000	53303654	

FIGURE MUL38 A*M = 07204746 64342230 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 10310731	0 00000000	00000000	33534454	
T7	0 04144354	0 33534454	00000000	33534454	*
T6	0 02062166	0 15656226	00000000	33534454	
T5	0 01031073	0 06727113	00000000	33534454	
T4	0 40414435	0 36007011	01101110	33534454	*
T3	0 60206216	0 42170424	11010100	33534454	*
T2	0 30103107	0 25470252	00010000	33534454	
T1	0 14041443	0 45374571	01000010	33534454	*
TC	0 46020621	0 56622654	00110100	33534454	*
TR	0 23010310	0 52001732	11110110	33534454	*
TP	0 11404144	0 21444711	10000110	33534454	
T8	0 44602062	0 14622300	00000110	33534454	
T7	0 22301031	0 06311104	00000100	33534454	
T6	0 11140414	0 36670156	00011000	33534454	*
T5	0 04460206	0 17330467	00010000	33534454	
T4	0 42230103	0 07550233	00010000	33534454	
T3	0 61114041	0 36324561	01100010	33534454	*
T2	0 70446020	0 42346640	11000110	33534454	*
T1	0 34223010	0 25563364	00000000	33534454	
TC	0 16111404	0 12671572	00000000	33534454	
TR	0 07044602	0 05334675	00000000	33534454	
TP	0 43422301	0 02556336	00000000	33534454	
T8	0 21611140	0 34713523	00110110	33534454	*
T7	0 50704460	0 16301615	00110100	33534454	
T6	0 64342230	0 07104746	00100000	44243323	
TP	0 07204746	0 64342230	00100000	44243323	

FIGURE MUL39 A*M = 05043667 76312562 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 16516613	0 00000000	00000000	13044513	
T7	0 07247306	0 13044513	00000000	13044513	*
T6	0 43523542	0 10466750	10000010	13044513	*
T5	0 21651661	0 00233360	10000010	13044513	
T4	0 10724730	0 17151207	00011100	13044513	*
T3	0 44352354	0 07460143	00011000	13044513	
T2	0 62165166	0 03634461	00000000	13044513	
T1	0 71072473	0 01716230	00000000	13044513	
T0	0 34435235	0 13703627	00110000	13044513	*
TR	0 56216516	0 10041426	11011000	13044513	*
TP	0 27107247	0 00424213	10001000	13044513	
T8	0 53443523	0 17256210	00001010	13044513	*
T7	0 25621651	0 12563223	10011000	13044513	*
T6	0 52710724	0 24232624	00110000	13044513	*
T5	0 25344352	0 12161312	00010000	13044513	
T4	0 12562165	0 05060545	00010000	13044513	
T3	0 45271072	0 15470775	00010000	13044513	*
T2	0 62534435	0 06630376	00010000	13044513	
T1	0 31256216	0 16364602	00000110	13044513	*
TC	0 14527107	0 07172345	00000000	13044513	
TR	0 46253443	0 16431675	00110000	13044513	*
TP	0 63125621	0 12224441	10101010	13044513	*
T8	0 71452710	0 24116337	00101000	13044513	*
T7	0 74625344	0 12007557	00100000	13044513	
T6	0 76312562	0 05043667	00000000	64733264	
TP	0 05043667	0 76312562	00000000	64733264	

FIGURE MUL40 A*M = 02073303 11151642 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 03573443	0 00000000	00000000	22062173	
T7	0 01675621	0 22062173	00000000	22062173	*
T6	0 40736710	0 33013160	00100110	22062173	*
T5	0 20357344	0 15445434	00000100	22062173	
T4	0 10157562	0 06622656	00000000	22062173	
T3	0 04073671	0 03311327	00000000	22062173	
T2	0 42035734	0 23526646	00100100	22062173	*
T1	0 21016756	0 11613363	00100000	22062173	
T0	0 50407367	0 04745571	00000000	22062173	
TR	0 64203573	0 24344767	00100100	22062173	*
TP	0 72101675	0 34204526	00100100	22062173	*
T8	0 35040736	0 30124406	10100100	22062173	*
T7	0 16420357	0 10012243	10100000	22062173	
T6	0 47210167	0 32027214	00100100	22062173	*
T5	0 23504073	0 37035631	00100110	22062173	*
T4	0 51642035	0 31430053	10111100	22062173	*
T3	0 64721016	0 42642550	00100110	22062173	*
T2	0 32350407	0 21361220	00000110	22062173	
T1	0 15164203	0 32552647	00100100	22062173	*
T0	0 46472101	0 37307456	00100100	22062173	*
TR	0 23235040	0 31665752	10000110	22062173	*
TP	0 11516420	0 10732721	10000110	22062173	
T8	0 44647210	0 00355314	10000100	22062173	
T7	0 22323504	0 04166506	00000100	22062173	
T6	0 11151642	0 02073203	00000100	55716604	
TP	0 02073303	0 11151642	00000100	55716604	

FIGURE MUL41 A*M = 77116344 30635022 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 34330013	0 00000000	00000000	77027673	
T7	0 16154005	1 77027673	00000000	77027673	*
T6	0 47066002	1 66432520	10011110	77027673	*
T5	0 23433001	1 77211614	00010100	77027673	
T4	0 11615400	1 66537531	10001110	77027673	*
T3	0 44706600	1 77257210	00001110	77027673	
T2	0 22343300	1 77527140	00001010	77027673	
T1	0 11161540	1 77653064	00001000	77027673	
T0	0 04470660	1 77725032	00001000	77027673	
TR	0 02234330	1 77752015	00001000	77027673	
TP	0 41116154	1 77765406	00000000	77027673	
T8	0 20447066	1 77772603	00000000	77027673	
T7	0 50223433	1 77775301	00000000	77027673	
T6	0 64111615	1 66715333	10111100	77027673	*
T5	0 72044706	1 66432000	10011110	77027673	*
T4	0 35022343	1 77211444	00010000	77027673	
T3	0 16411161	1 66537415	10001100	77027673	*
T2	0 47204470	1 66277031	10011110	77027673	*
T1	0 63502234	1 77133050	00011010	77027673	
T0	0 31641116	1 77451020	00011010	77027673	
TR	0 14720447	1 77620014	00011000	77027673	
TP	0 06350223	1 66733271	10011010	77027673	*
T8	0 43164111	1 66301733	10110100	77027673	*
T7	0 61472044	1 66223600	10011110	77027673	*
T6	0 30635022	1 77115344	00001000	00750104	
TP	0 77116344	1 30635022	00001000	00750104	

FIGURE MUL42 A*M = 02675464 65657500 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 15470352	0 00000000	00000000	06577220	
T7	0 06634165	0 00000000	00000000	06577220	
T6	0 03316072	0 06577220	00000000	06577220	*
T5	0 01547035	0 03277510	00000000	06577220	
T4	0 00663416	0 07226064	01111000	06577220	*
T3	0 00331607	0 03157432	01000000	06577220	
T2	0 00154703	0 00556035	10111000	06577220	*
T1	0 40066341	0 02022636	11110000	06577220	*
TC	0 20033160	0 03144537	11110000	06577220	*
TR	0 50015470	0 05026257	01100000	06577220	
TP	0 64006634	0 02053127	01000000	06577220	
T8	0 72003316	0 01425453	00000000	06577220	
T7	0 75001547	0 00612625	00000000	06577220	
T6	0 76400663	0 06074532	01010000	06577220	*
T5	0 37200331	0 01131475	11110000	06577220	*
T4	0 57500154	0 03617056	10101000	06577220	*
T3	0 27640066	0 05747027	00001000	06577220	
T2	0 53720033	0 02763013	00001000	06577220	
T1	0 65750015	0 07061225	01110000	06577220	*
TC	0 72764006	0 02563732	10110000	06577220	*
TR	0 35372003	0 05235755	00100000	06577220	
TP	0 56575001	0 00245106	11111100	06577220	*
T8	0 27276400	0 02256323	11110000	06577220	*
T7	0 53537200	0 05563151	00010000	06577220	
T6	0 65657500	0 02675464	00000000	71200557	
TP	0 02675464	0 65657500	00000000	71200557	

FIGURE MUL43 A*M = 03303112 44670636 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 47453333	0 00000000	00000000	73443035	
T7	1 23625555	1 73443035	00000000	73443035	*
T6	1 51712666	1 60264443	11000010	73443035	*
T5	1 64745333	1 74532225	00000000	73443035	
T4	1 72362555	1 61610147	10110000	73443035	*
T3	1 75171266	1 67313010	01100110	73443035	*
T2	1 36474533	1 73105440	01100010	73443035	
T1	1 17236255	1 61545651	10000010	73443035	*
TC	1 47517126	1 67225765	01100000	73443035	*
TR	1 63647453	1 73152772	01000000	73443035	
TP	1 31723625	1 61420322	10110110	73443035	*
T8	1 14751712	1 67217242	01100010	73443035	*
T7	1 06364745	1 73147525	01000000	73443035	
T6	1 43172362	1 61426607	10100100	73443035	*
T5	1 61475171	1 74653343	00000000	73443035	
T4	1 70636474	1 61760516	10010100	73443035	*
T3	1 34317236	1 74774207	00000100	73443035	
T2	1 56147517	1 76376143	00000000	73443035	
T1	1 67063647	1 62532016	10110100	73443035	*
TC	1 33431723	1 60764074	10000010	73443035	*
TR	1 15614751	1 67735077	00100000	73443035	*
TP	1 46706364	1 66451464	01010010	73443035	*
T8	1 23343172	1 73620636	00010000	73443035	
T7	1 11561475	1 75714317	00000000	73443035	
T6	1 44670636	0 72272001	11011111	04334742	*
TP	1 03303112	0 44670636	11011111	04334742	

FIGURE MUL44 A*M = 05536742 02312574 0

TIME	A REGISTER	B REGISTER	SC REGISTER	C REGISTER	
T8	0 32432635	0 00000000	00000000	06664126	
T7	0 15215316	0 06664126	00000000	06664126	*
T6	0 06506547	0 03332053	00000000	06664126	
T5	0 43243263	0 07331143	01110010	06664126	*
T4	0 61521531	0 02004503	11100110	06664126	*
T3	0 73650654	0 03226423	11100010	06664126	*
T2	0 74324326	0 05153215	01000000	06664126	
T1	0 76152153	0 02065506	01000000	06664126	
T0	0 37065065	0 07216761	01100010	06664126	*
TR	0 57432432	0 02723412	10110110	06664126	*
TP	0 27615215	0 05315641	00100010	06664126	
T8	0 53706506	0 00462042	11011010	06664126	*
T7	0 25743243	0 04635425	00000000	06664126	
T6	0 52761521	0 00172730	11010010	06664126	*
T5	0 25370650	0 02265406	11100100	06664126	*
T4	0 12574324	0 05572643	00000000	06664126	
T3	0 45276152	0 02675321	00000000	06664126	
T2	0 62537065	0 01336550	00000000	06664126	
T1	0 31257432	0 06333302	01110110	06664126	*
T0	0 14527615	0 03511505	00110100	06664126	
TR	0 46253706	0 07574020	01001010	06664126	*
TP	0 23125743	0 03276414	01000000	06664126	
T8	0 11452761	0 00713324	10110010	06664126	*
T7	0 04625370	0 02275674	11000010	06664126	*
T6	0 02312574	0 05536732	00000010	71113651	
TP	0 05536742	0 02312574	00000010	71113651	

149

FIGURE MUL45 A*M = 07221555 20577040 0

TIME	A REGISTER	B REGISTER	SC REGISTER	C REGISTER	
T8	0 30556344	0 00000000	00000000	11334744	
T7	0 14267162	0 00000000	00000000	11334744	
T6	0 06133471	0 00000000	00000000	11334744	
T5	0 03055634	0 11334744	00000000	11334744	*
T4	0 01426716	0 04556362	00000000	11334744	
T3	0 00613347	0 02267171	00000000	11334744	
T2	0 40305563	0 12467330	00001110	11334744	*
T1	0 20142671	0 16560164	00011000	11334744	*
TC	0 10061334	0 10520336	10111100	11334744	*
TK	0 04030556	0 00214517	10100100	11334744	
TP	0 42014267	0 04146207	00000100	11334744	
T8	0 61006133	0 13317007	00101100	11334744	*
T7	0 70403055	0 16134707	01010100	11334744	*
T6	0 74201426	0 10716347	10101000	11334744	*
T5	0 76100613	0 00307563	10100000	11334744	
T4	0 77040305	0 15537535	00001100	11334744	*
T3	0 77420142	0 17104252	01111010	11334744	*
T2	0 37610061	0 07006521	01100010	11334744	
T1	0 57704030	0 15377110	00001110	11334744	*
TC	0 27742014	0 06577000	00001110	11334744	
TK	0 13761006	0 03277044	00001000	11334744	
TP	0 05770403	0 01537022	00001000	11334744	
T8	0 02774201	0 11104755	01110000	11334744	*
T7	0 41376100	0 16432222	00011110	11334744	*
T6	0 20577040	0 07211555	00010000	66443033	
TP	0 07221555	0 20577040	00010000	66443033	

FIGURE MUL46 A*M = 04002564 42550272 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 11430455	0 00000000	00000000	15330761	
T7	0 04614226	0 15330761	00000000	15330761	*
T6	0 42306113	0 06564370	00000000	15330761	
T5	0 21143045	0 10516055	10101100	15330761	*
T4	0 50461422	0 25637347	00001100	15330761	*
T3	0 64230611	0 12717123	00001100	15330761	
T2	0 72114304	0 12670072	10011000	15330761	*
T1	0 35046142	0 01330435	10010000	15330761	
T0	0 56423061	0 04550216	00010000	15330761	
TR	0 27211430	0 17520060	00101010	15330761	*
TP	0 13504614	0 07610434	00100000	15330761	
T8	0 05642306	0 03744216	00000000	15330761	
T7	0 02721143	0 01762107	00000000	15330761	
T6	0 41350461	0 16221724	01100100	15330761	*
T5	0 20564230	0 14400773	10101000	15330761	*
T4	0 50272114	0 02240775	10000000	15330761	
T3	0 64135046	0 05120376	00000000	15330761	
T2	0 32056423	0 02450177	00000000	15330761	
T1	0 55027211	0 16554750	00000110	15330761	*
T0	0 26413504	0 14516311	10101100	15330761	*
TR	0 53205642	0 02207504	10100100	15330761	
TP	0 25502721	0 05143602	00000100	15330761	
T8	0 12641350	0 17711622	00101100	15330761	*
T7	0 05320564	0 07704351	00101000	15330761	
T6	0 42550272	0 03702564	00100000	62447016	
TP	0 04002564	0 42550272	00100000	62447016	

FIGURE MUL47 A*M = 00006074 35126000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 75513000	0 00000000	00000000	77654641	
T7	1 36645400	0 00000000	00000000	77654641	
T6	1 17322600	0 00000000	00000000	77654641	
T5	1 07551300	0 00000000	00000000	77654641	
T4	1 03664540	0 00000000	00000000	77654641	
T3	1 01732260	0 00000000	00000000	77654641	
T2	1 00755130	0 00000000	00000000	77654641	
T1	1 00366454	0 00000000	00000000	77654641	
TC	1 00173226	0 00000000	00000000	77654641	
TR	1 00075513	0 00000000	00000000	77654641	
TP	1 00036645	1 77654641	00000000	77654641	*
T8	1 40017322	1 66572161	11011000	77654641	*
T7	1 60007551	1 77671470	00010000	77654641	
T6	1 30003664	1 66514475	11101000	77654641	*
T5	1 54001732	1 77606636	00100000	77654641	
T4	1 26000755	1 77743317	00000000	77654641	
T3	1 53000366	1 66535300	11101110	77654641	*
T2	1 25400173	1 77616104	00101100	77654641	
T1	1 12600075	1 66513343	11111000	77654641	*
TC	1 45300036	1 66666722	11000100	77654641	*
TR	1 22540017	1 77673311	00000100	77654641	
TP	1 51260007	1 66501445	11111000	77654641	*
T8	1 64530003	1 66551063	11011000	77654641	*
T7	1 72254001	1 66445672	11100000	77654641	*
T6	1 35126000	0 77706063	00101011	00123136	*
TP	1 00006074	0 35126000	00101011	00123136	

FIGURE MUL48 A*M = 00404502 17747764 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 06431271	0 00000000	00000000	02371312	
T7	0 03214534	0 02371312	00000000	02371312	*
T6	0 01506256	0 01174545	00000000	02371312	
T5	0 40643127	0 00476262	00000000	02371312	
T4	0 20321453	0 02520443	00110000	02371312	*
T3	0 50150625	0 03605533	00100000	02371312	*
T2	0 64064312	0 03233167	01101000	02371312	*
T1	0 72032145	0 01155073	01001000	02371312	
TC	0 75015062	0 03350347	00110000	02371312	*
TR	0 76406431	0 01620163	00110000	02371312	
TP	0 77203214	0 02205303	01100100	02371312	*
T8	0 77501506	0 01542501	00000100	02371312	
T7	0 77640643	0 00661200	00000100	02371312	
T6	0 37720321	0 02621052	00101000	02371312	*
T5	0 17750150	0 03642337	00100000	02371312	*
T4	0 47764064	0 01761167	00000000	02371312	
T3	0 63772032	0 00770467	00000000	02371312	
T2	0 71775015	0 00374233	00000000	02371312	
T1	0 74776406	0 02467427	00100000	02371312	*
TC	0 76377203	0 01273613	00000000	02371312	
TR	0 77177501	0 02026217	01101000	02371312	*
TP	0 77477640	0 03744011	00101010	02371312	*
T8	0 77637720	0 01722400	00100010	02371312	
T7	0 37717750	0 00711204	00100000	02371312	
T6	0 17747764	0 00304502	00100000	75406465	
TP	0 00404502	0 17747764	00100000	75406465	

FIGURE MUL49 A*M = 02216520 77651600 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 33330610	0 00000000	00000000	02521270	
T7	0 15554304	0 00000000	00000000	02521270	
T6	0 06666142	0 00000000	00000000	02521270	
T5	0 03333061	0 00000000	00000000	02521270	
T4	0 01555430	0 02521270	00000000	02521270	*
T3	0 00666614	0 01250534	00000000	02521270	
T2	0 00333306	0 00524256	00000000	02521270	
T1	0 00155543	0 00252127	00000000	02521270	
T0	0 40066661	0 02646243	00000100	02521270	*
TR	0 60033330	0 03044351	01000100	02521270	*
TP	0 70015554	0 01022124	01000100	02521270	
T8	0 34006666	0 00011012	01000100	02521270	
T7	0 16003333	0 00404445	00000000	02521270	
T6	0 47001555	0 02723412	00000100	02521270	*
T5	0 23400666	0 03072035	01001100	02521270	*
T4	0 51600333	0 01035456	01000000	02521270	
T3	0 24700155	0 03537017	00001100	02521270	*
T2	0 52340066	0 03371237	01010100	02521270	*
T1	0 65160033	0 01170557	01010000	02521270	
T0	0 72470015	0 03521457	00100100	02521270	*
TR	0 75234006	0 03431057	01001100	02521270	*
TP	0 76516003	0 01214067	01001000	02521270	
T8	0 77247001	0 03627623	00000100	02521270	*
T7	0 77523400	0 03434141	01001100	02521270	*
T6	0 77651600	0 01216420	01000100	75256507	
TP	0 02216520	0 77651600	01000100	75256507	

FIGURE MUL50 A*M = 74002276 11743170 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62512352	0 00000000	00000000	11445546	
T7	1 31245165	0 00000000	00000000	11445546	
T6	1 14522472	0 11445546	00000000	11445546	*
T5	1 06251235	0 04622663	00000000	11445546	
T4	1 43124516	0 13756077	00001000	11445546	*
T3	1 61452247	0 05767437	00000000	11445546	
T2	1 70625123	0 13330355	01111010	11445546	*
T1	1 74312451	0 17665230	00001110	11445546	*
TC	1 36145224	0 10370626	11010100	11445546	*
TR	1 17062512	0 00570353	10010000	11445546	
TP	1 47431245	0 04270165	00010000	11445546	
T8	1 63614522	0 13505530	00100110	11445546	*
T7	1 31706251	0 05602610	00100110	11445546	
T6	1 14743124	0 13306016	01101100	11445546	*
T5	1 06361452	0 05103447	01100000	11445546	
T4	1 43170625	0 02001623	01100000	11445546	
T3	1 61474312	0 12005457	01101000	11445546	*
T2	1 70636145	0 05442227	00001000	11445546	
T1	1 74317062	0 13266251	01001010	11445546	*
TC	1 76147431	0 06133520	01000010	11445546	
TR	1 37063614	0 14412312	00111110	11445546	*
TP	1 17431706	0 06241501	00010110	11445546	
T8	1 47614743	0 03124604	00000100	11445546	
T7	1 23706361	0 12017000	01101110	11445546	*
T6	1 11743170	1 63772275	10010001	66332231	*
TP	1 74002276	1 11743170	10010001	66332231	

FIGURE MULS1 A*M = 40000000 00000000 1

TIME	A REGISTER		B REGISTER		BC REGISTER	C REGISTER	
T8	1	40000000	0	00000000	00000000	40000000	
T7	1	20000000	0	00000000	00000000	40000000	
T6	1	10000000	0	00000000	00000000	40000000	
T5	1	04000000	0	00000000	00000000	40000000	
T4	1	02000000	0	00000000	00000000	40000000	
T3	1	01000000	0	00000000	00000000	40000000	
T2	1	00400000	0	00000000	00000000	40000000	
T1	1	00200000	0	00000000	00000000	40000000	
T0	1	00100000	0	00000000	00000000	40000000	
TR	1	00040000	0	00000000	00000000	40000000	
TP	1	00020000	0	00000000	00000000	40000000	
T8	1	00010000	0	00000000	00000000	40000000	
T7	1	00004000	0	00000000	00000000	40000000	
T6	1	00002000	0	00000000	00000000	40000000	
T5	1	00001000	0	00000000	00000000	40000000	
T4	1	00000400	0	00000000	00000000	40000000	
T3	1	00000200	0	00000000	00000000	40000000	
T2	1	00000100	0	00000000	00000000	40000000	
T1	1	00000040	0	00000000	00000000	40000000	
T0	1	00000020	0	00000000	00000000	40000000	
TR	1	00000010	0	00000000	00000000	40000000	
TP	1	00000004	0	00000000	00000000	40000000	
T8	1	00000002	0	00000000	00000000	40000000	
T7	1	00000001	0	00000000	00000000	40000000	
T6	1	00000000	0	37777777	00000001	37777777	*
TP	1	40000000	0	00000000	00000001	37777777	

APPENDIX 3D

DIVIDE AND THE BZO "EYE-BALL CIRCUIT"

The following sets of constants allow the user to diagnose completely any BZO and/or divide problem. If a failure is suspected, the user should reset all breakpoints and load the tape using the Diagnostic load procedure beginning at the Automatic Instruction Diagnostic.

If the program halts at a divide error halt, determine the pertinent constant set via Appendix 3A. Then, set Breakpoints 1 and 2 and place the mode switch in RUN. This repeatedly performs the current test with the error halt inhibited. The user can now use the oscilloscope to follow the test through each T-time to determine the exact circuit area.

FIGURE DIV 1 A/M = 40000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000000	00000000	37777777	
T8	0 00000000	0 00000000	00000000	40000000	
T5	0 00000001	1 40000000	00000001	40000000	
T2	0 00000002	0 37777771	00000010	37777777	
TR	0 00000004	0 26666771	11111010	37777777	*
T7	0 00000010	0 26666771	11111010	37777777	*
T4	0 00000020	0 26666771	11111010	37777777	*
T1	0 00000040	0 26666771	11111010	37777777	*
TP	0 00000100	0 26666771	11111010	37777777	*
T6	0 00000200	0 26666771	11111010	37777777	*
T3	0 00000400	0 26666771	11111010	37777777	*
T0	0 00001000	0 26666771	11111010	37777777	*
T8	0 00002000	0 26666771	11111010	37777777	*
T5	0 00004000	0 26666771	11111010	37777777	*
T2	0 00010000	0 26666771	11111010	37777777	*
TR	0 00020000	0 26666771	11111010	37777777	*
T7	0 00040000	0 26666771	11111010	37777777	*
T4	0 00100000	0 26666771	11111010	37777777	*
T1	0 00200000	0 26666771	11111010	37777777	*
TP	0 00400000	0 26666771	11111010	37777777	*
T6	0 01000000	0 26666771	11111010	37777777	*
T3	0 02000000	0 26666771	11111010	37777777	*
T0	0 04000000	0 26666771	11111010	37777777	*
T8	0 10000000	0 26666771	11111010	37777777	*
T5	0 20000000	0 26666771	11111010	37777777	*
T2	0 40000000	0 77777702	00000100	37777777	*
TP	0 40000000	0 00000000	11111010	37777777	OF

FIGURE DIV 2 A/M = 37777777 40000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000002	00000000	37777777	
T8	0 00000002	0 00000000	00000000	40000000	
T5	0 00000005	1 40000000	00000001	40000000	
T2	0 00000012	0 37777771	00000010	37777777	
TR	0 00000024	0 26666771	11111010	37777777	*
T7	0 00000050	0 26666771	11111010	37777777	*
T4	0 00000120	0 26666771	11111010	37777777	*
T1	0 00000240	0 26666771	11111010	37777777	*
TP	0 00000500	0 26666771	11111010	37777777	*
T6	0 00001200	0 26666771	11111010	37777777	*
T3	0 00002400	0 26666771	11111010	37777777	*
T0	0 00005000	0 26666771	11111010	37777777	*
T8	0 00012000	0 26666771	11111010	37777777	*
T5	0 00024000	0 26666771	11111010	37777777	*
T2	0 00050000	0 26666771	11111010	37777777	*
TR	0 00120000	0 26666771	11111010	37777777	*
T7	0 00240000	0 26666771	11111010	37777777	*
T4	0 00500000	0 26666771	11111010	37777777	*
T1	0 01200000	0 26666771	11111010	37777777	*
TP	0 02400000	0 26666771	11111010	37777777	*
T6	0 05000000	0 26666771	11111010	37777777	*
T3	0 12000000	0 26666771	11111010	37777777	*
T0	0 24000000	0 26666771	11111010	37777777	*
T8	0 50000000	0 26666771	11111010	37777777	*
T5	0 20000000	0 26666772	11111010	37777777	*
T2	0 40000000	0 77777704	00000100	37777777	*
TP	0 40000000	0 00000001	11111010	37777777	OF

FIGURE DIV 3   A/M = 40000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000000	00000000	40000001	
T8	0 00000000	0 00000000	00000000	40000001	
T5	0 00000001	1 40000001	00000000	40000001	
T2	0 00000002	0 37777770	00000011	37777776	
TR	0 00000004	0 26666770	11111011	37777776	*
T7	0 00000010	0 26666770	11111011	37777776	*
T4	0 00000020	0 26666770	11111011	37777776	*
T1	0 00000040	0 26666770	11111011	37777776	*
TP	0 00000100	0 26666770	11111011	37777776	*
T6	0 00000200	0 26666770	11111011	37777776	*
T3	0 00000400	0 26666770	11111011	37777776	*
T0	0 00001000	0 26666770	11111011	37777776	*
T8	0 00002000	0 26666770	11111011	37777776	*
T5	0 00004000	0 26666770	11111011	37777776	*
T2	0 00010000	0 26666770	11111011	37777776	*
TR	0 00020000	0 26666770	11111011	37777776	*
T7	0 00040000	0 26666770	11111011	37777776	*
T4	0 00100000	0 26666770	11111011	37777776	*
T1	0 00200000	0 26666770	11111011	37777776	*
TP	0 00400000	0 26666770	11111011	37777776	*
T6	0 01000000	0 26666770	11111011	37777776	*
T3	0 02000000	0 26666770	11111011	37777776	*
T0	0 04000000	0 26666770	11111011	37777776	*
T8	0 10000000	0 26666770	11111011	37777776	*
T5	0 20000000	0 26666770	11111011	37777776	*
T2	0 40000000	0 77777702	00000100	37777776	*
TP	0 40000000	0 00000000	11111001	37777776	

FIGURE DIV 4 A/M = 40000001 40000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000002	00000000	40000001	
T8	0 00000002	0 00000000	00000000	40000001	
T5	0 00000005	1 40000001	00000000	40000001	
T2	0 00000012	0 37777770	00000011	37777776	
TR	0 00000024	0 26666770	11111011	37777776	*
T7	0 00000050	0 26666770	11111011	37777776	*
T4	0 00000120	0 26666770	11111011	37777776	*
T1	0 00000240	0 26666770	11111011	37777776	*
TP	0 00000500	0 26666770	11111011	37777776	*
T6	0 00001200	0 26666770	11111011	37777776	*
T3	0 00002400	0 26666770	11111011	37777776	*
T0	0 00005000	0 26666770	11111011	37777776	*
T8	0 00012000	0 26666770	11111011	37777776	*
T5	0 00024000	0 26666770	11111011	37777776	*
T2	0 00050000	0 26666770	11111011	37777776	*
TR	0 00120000	0 26666770	11111011	37777776	*
T7	0 00240000	0 26666770	11111011	37777776	*
T4	0 00500000	0 26666770	11111011	37777776	*
T1	0 01200000	0 26666770	11111011	37777776	*
TP	0 02400000	0 26666770	11111011	37777776	*
T6	0 05000000	0 26666770	11111011	37777776	*
T3	0 12000000	0 26666770	11111011	37777776	*
T0	0 24000000	0 26666770	11111011	37777776	*
T8	0 50000000	0 26666770	11111011	37777776	*
T5	0 20000000	0 26666771	11111011	37777776	*
T2	0 40000000	0 77777704	00000100	37777776	*
TP	0 40000000	0 00000001	11111011	37777776	OF

FIGURE DIV 6 A/M = 60000000 60000001 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000000	0 00000002	00000000	37777777	
T8	1 77777776	0 00000000	00000000	40000000	
T5	1 77777775	1 40000001	00000001	40000000	
T2	1 77777772	0 37777774	00000010	37777777	
TR	1 77777764	0 26666700	11111110	37777777	*
T7	1 77777750	0 26667710	11110110	37777777	*
T4	1 77777720	0 26667730	11110110	37777777	*
T1	1 77777640	0 26667070	11111010	37777777	*
TP	1 77777500	0 26677170	11101010	37777777	*
T6	1 77777200	0 26677370	11101010	37777777	*
T3	1 77776400	0 26677770	11101010	37777777	*
TC	1 77775000	0 26670770	11111010	37777777	*
T8	1 77772000	0 26772770	11011010	37777777	*
T5	1 77764000	0 26776770	11011010	37777777	*
T2	1 77750000	0 26706770	11111010	37777777	*
TR	1 77720000	0 27726770	10111010	37777777	*
T7	1 77640000	0 27766770	10111010	37777777	*
T4	1 77500000	0 27066770	11111010	37777777	*
T1	1 77200000	0 37266770	01111010	37777777	*
TP	1 76400000	0 37666770	01111010	37777777	*
T6	1 75000000	0 30666770	11111010	37777777	*
T3	1 72000000	0 32666770	11111010	37777777	*
TC	1 64000000	0 36666770	11111010	37777777	*
T8	1 50000000	0 46666770	11111010	37777777	*
T5	1 20000000	0 66666770	11111010	37777777	
T2	1 40000001	1 77777700	00000100	40000000	
TP	1 37777777	1 00000000	00000100	40000000	OF

FIGURE DIV 6 A/M = 20000000 60000001 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000000	0 00000002	00000000	40000001	
T8	1 77777776	0 00000000	00000000	40000001	
T5	1 77777775	1 40000002	00000000	40000001	
T2	1 77777772	0 37777773	00000011	37777776	
TR	1 77777764	0 26666707	11111101	37777776	*
T7	1 77777750	0 26667717	11110101	37777776	*
T4	1 77777720	0 26667737	11110101	37777776	*
T1	1 77777640	0 26667777	11110101	37777776	*
TP	1 77777500	0 26667077	11111101	37777776	*
T6	1 77777200	0 26677277	11101101	37777776	*
T3	1 77776400	0 26677677	11101101	37777776	*
TC	1 77775000	0 26670677	11111101	37777776	*
T8	1 77772000	0 26772677	11011101	37777776	*
T5	1 77764000	0 26776677	11011101	37777776	*
T2	1 77750000	0 26706677	11111101	37777776	*
TR	1 77720000	0 27726677	10111101	37777776	*
T7	1 77640000	0 27766677	10111101	37777776	*
T4	1 77500000	0 27066677	11111101	37777776	*
T1	1 77200000	0 37266677	01111101	37777776	*
TP	1 76400000	0 37666677	01111101	37777776	*
T6	1 75000000	0 30666677	11111101	37777776	*
T3	1 72000000	0 32666677	11111101	37777776	*
TC	1 64000000	0 36666677	11111101	37777776	*
T8	1 60000000	0 46666677	11111101	37777776	*
T5	1 20000000	0 66666677	11111101	37777776	
T2	1 40000001	1 77777770	00000010	40000001	
TP	1 40000001	1 00000000	00000010	40000001	OF

FIGURE DIV 7 A/M = 40000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000000	0 00000000	00000000	00000000	
T8	1 00000000	0 37777777	00000001	77777777	*
T5	1 00000000	0 77777700	00000100	00000000	*
T2	1 00000001	1 66667777	11110001	77777777	*
TR	1 00000003	1 66666677	11111101	77777777	*
T7	1 00000007	1 66666677	11111101	77777777	*
T4	1 00000017	1 66666677	11111101	77777777	*
T1	1 00000037	1 66666677	11111101	77777777	*
TP	1 00000077	1 66666677	11111101	77777777	*
T6	1 00000177	1 66666677	11111101	77777777	*
T3	1 00000377	1 66666677	11111101	77777777	*
T0	1 00000777	1 66666677	11111101	77777777	*
T8	1 00001777	1 66666677	11111101	77777777	*
T5	1 00003777	1 66666677	11111101	77777777	*
T2	1 00007777	1 66666677	11111101	77777777	*
TR	1 00017777	1 66666677	11111101	77777777	*
T7	1 00037777	1 66666677	11111101	77777777	*
T4	1 00077777	1 66666677	11111101	77777777	*
T1	1 00177777	1 66666677	11111101	77777777	*
TP	1 00377777	1 66666677	11111101	77777777	*
T6	1 00777777	1 66666677	11111101	77777777	*
T3	1 01777777	1 66666677	11111101	77777777	*
T0	1 03777777	1 66666677	11111101	77777777	*
T8	1 07777777	1 66666677	11111101	77777777	*
T5	1 17777777	1 66666677	11111101	77777777	*
T2	1 37777777	1 77777770	00000010	77777777	*
TP	1 40000001	1 00000000	00000010	77777777	0F

FIGURE DIV 8 A/M = 37777777 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	0 37777776	0 00000002	00000000	37777777
T8	0 00000002	0 77777777	00000000	40000000
T5	0 00000004	0 26666665	11111110	37777777
T2	0 00000011	1 37777772	00000001	40000000
TR	0 00000023	1 37777766	00000001	40000000
T7	0 00000047	1 37777756	00000001	40000000
T4	0 00000117	1 37777736	00000001	40000000
T!	0 00000237	1 37777676	00000001	40000000
TP	0 00000477	1 37777576	00000001	40000000
T6	0 00001177	1 37777376	00000001	40000000
T3	0 00002377	1 37776776	00000001	40000000
T0	0 00004777	1 37775776	00000001	40000000
T8	0 00011777	1 37773776	00000001	40000000
T5	0 00023777	1 37767776	00000001	40000000
T2	0 00047777	1 37757776	00000001	40000000
TR	0 00117777	1 37737776	00000001	40000000
T7	0 00237777	1 37677776	00000001	40000000
T4	0 00477777	1 37577776	00000001	40000000
T!	0 01177777	1 37377776	00000001	40000000
TP	0 02377777	1 36777776	00000001	40000000
T6	0 04777777	1 35777776	00000001	40000000
T3	0 11777777	1 33777776	00000001	40000000
T0	0 23777777	1 27777776	00000001	40000000
T8	0 47777777	1 17777776	00000001	40000000
T5	0 17777777	1 77777777	00000001	40000000
T2	0 37777777	1 77777770	00000010	40000000
TP	0 37777777	1 00000000	00000010	40000000

FIGURE DIV 9 A/M = 40000001 00000001 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	0 37777777	0 00000002	00000000	40000000
T8	0 00000002	0 77777777	00000000	40000000
T5	0 00000004	0 26666665	11111111	37777777
T2	0 00000011	1 37777774	00000000	40000000
TR	0 00000023	1 37777770	00000000	40000000
T7	0 00000047	1 37777760	00000000	40000000
T4	0 00000117	1 37777740	00000000	40000000
T1	0 00000237	1 37777700	00000000	40000000
TP	0 00000477	1 37777600	00000000	40000000
T6	0 00001177	1 37777400	00000000	40000000
T3	0 00002377	1 37777000	00000000	40000000
TC	0 00004777	1 37776000	00000000	40000000
T8	0 00011777	1 37774000	00000000	40000000
T5	0 00023777	1 37770000	00000000	40000000
T2	0 00047777	1 37760000	00000000	40000000
TR	0 00117777	1 37740000	00000000	40000000
T7	0 00237777	1 37700000	00000000	40000000
T4	0 00477777	1 37600000	00000000	40000000
T1	0 01177777	1 37400000	00000000	40000000
TP	0 02377777	1 37000000	00000000	40000000
T6	0 04777777	1 36000000	00000000	40000000
T3	0 11777777	1 34000000	00000000	40000000
TC	0 23777777	1 30000000	00000000	40000000
T8	0 47777777	1 20000000	00000000	40000000
T5	0 17777777	1 00000001	00000000	40000000
T2	0 37777777	0 00000002	00000000	40000000
TP	0 40000001	0 00000001	00000000	40000000

FIGURE DIV10 A/M = 40000001 40000002 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 40000001	0 00000002	00000000	37777777
T8	1 77777776	0 77777777	00000000	40000000
T5	1 77777774	0 26666666	11111110	37777777
T2	1 77777771	1 37777775	00000001	40000000
TR	1 77777763	1 37777775	00000001	40000000
T7	1 77777747	1 37777775	00000001	40000000
T4	1 77777717	1 37777775	00000001	40000000
T1	1 77777637	1 37777775	00000001	40000000
TP	1 77777477	1 37777775	00000001	40000000
T6	1 77777177	1 37777775	00000001	40000000
T3	1 77776377	1 37777775	00000001	40000000
T0	1 77774777	1 37777775	00000001	40000000
T8	1 77771777	1 37777775	00000001	40000000
T5	1 77763777	1 37777775	00000001	40000000
T2	1 77747777	1 37777775	00000001	40000000
TR	1 77717777	1 37777775	00000001	40000000
T7	1 77637777	1 37777775	00000001	40000000
T4	1 77477777	1 37777775	00000001	40000000
T1	1 77177777	1 37777775	00000001	40000000
TP	1 76377777	1 37777775	00000001	40000000
T6	1 74777777	1 37777775	00000001	40000000
T3	1 71777777	1 37777775	00000001	40000000
T0	1 63777777	1 37777775	00000001	40000000
T8	1 47777777	1 37777775	00000001	40000000
T5	1 17777777	1 37777775	00000001	40000000
T2	1 37777777	0 77777774	00000000	40000000
TP	1 40000001	0 40000002	00000000	40000000

FIGURE DIV11 A/M = 37777777 40000002 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 40000001	0 00000002	00000000	40000001
T8	1 77777776	0 77777777	00000000	40000001
T5	1 77777774	0 26666665	11111111	37777776
T2	1 77777771	1 37777776	00000000	40000001
TR	1 77777763	1 37777776	00000000	40000001
T7	1 77777747	1 37777776	00000000	40000001
T4	1 77777717	1 37777776	00000000	40000001
T1	1 77777637	1 37777776	00000000	40000001
TP	1 77777477	1 37777776	00000000	40000001
T6	1 77777177	1 37777776	00000000	40000001
T3	1 77776377	1 37777776	00000000	40000001
TC	1 77774777	1 37777776	00000000	40000001
T8	1 77771777	1 37777776	00000000	40000001
T5	1 77763777	1 37777776	00000000	40000001
T2	1 77747777	1 37777776	00000000	40000001
TR	1 77717777	1 37777776	00000000	40000001
T7	1 77637777	1 37777776	00000000	40000001
T4	1 77477777	1 37777776	00000000	40000001
T1	1 77177777	1 37777776	00000000	40000001
TP	1 76377777	1 37777776	00000000	40000001
T6	1 74777777	1 37777776	00000000	40000001
T3	1 71777777	1 37777776	00000000	40000001
TC	1 63777777	1 37777776	00000000	40000001
T8	1 47777777	1 37777776	00000000	40000001
T5	1 17777777	1 37777776	00000000	40000001
T2	1 37777777	0 77777774	00000000	40000001
TP	1 37777777	0 40000002	00000000	40000001

FIGURE DIV12

A/M = 40000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000001	0 00000000	00000000	37777777	
T8	1 00000000	0 77777777	00000001	40000000	*
T5	1 00000001	1 37777700	00000101	40000000	*
T2	1 00000002	0 26667771	11110010	37777777	*
TR	1 00000004	0 26666771	11111010	37777777	*
T7	1 00000010	0 26666771	11111010	37777777	*
T4	1 00000020	0 26666771	11111010	37777777	*
T1	1 00000040	0 26666771	11111010	37777777	*
TP	1 00000100	0 26666771	11111010	37777777	*
T6	1 00000200	0 26666771	11111010	37777777	*
T3	1 00000400	0 26666771	11111010	37777777	*
T0	1 00001000	0 26666771	11111010	37777777	*
T8	1 00002000	0 26666771	11111010	37777777	*
T5	1 00004000	0 26666771	11111010	37777777	*
T2	1 00010000	0 26666771	11111010	37777777	*
TR	1 00020000	0 26666771	11111010	37777777	*
T7	1 00040000	0 26666771	11111010	37777777	*
T4	1 00100000	0 26666771	11111010	37777777	*
T1	1 00200000	0 26666771	11111010	37777777	*
TP	1 00400000	0 26666771	11111010	37777777	*
T6	1 01000000	0 26666771	11111010	37777777	*
T3	1 02000000	0 26666771	11111010	37777777	*
T0	1 04000000	0 26666771	11111010	37777777	*
T8	1 10000000	0 26666771	11111010	37777777	*
T5	1 20000000	0 26666771	11111010	37777777	*
T2	1 40000000	0 77777702	00000100	37777777	*
TP	1 40000000	0 00000000	11111010	37777777	

FIGURE DIV13 A/M = 40000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000001	0 00000000	00000000	40000001	
T8	1 00000000	0 77777777	00000001	40000001	*
T5	1 00000001	1 37777701	00000100	40000001	*
T2	1 00000002	0 26667770	11110011	37777776	*
TR	1 00000004	0 26666770	11111011	37777776	*
T7	1 00000010	0 26666770	11111011	37777776	*
T4	1 00000020	0 26666770	11111011	37777776	*
T1	1 00000040	0 26666770	11111011	37777776	*
TP	1 00000100	0 26666770	11111011	37777776	*
T6	1 00000200	0 26666770	11111011	37777776	*
T3	1 00000400	0 26666770	11111011	37777776	*
T0	1 00001000	0 26666770	11111011	37777776	*
T8	1 00002000	0 26666770	11111011	37777776	*
T5	1 00004000	0 26666770	11111011	37777776	*
T2	1 00010000	0 26666770	11111011	37777776	*
TR	1 00020000	0 26666770	11111011	37777776	*
T7	1 00040000	0 26666770	11111011	37777776	*
T4	1 00100000	0 26666770	11111011	37777776	*
T1	1 00200000	0 26666770	11111011	37777776	*
TP	1 00400000	0 26666770	11111011	37777776	*
T6	1 01000000	0 26666770	11111011	37777776	*
T3	1 02000000	0 26666770	11111011	37777776	*
T0	1 04000000	0 26666770	11111011	37777776	*
T8	1 10000000	0 26666770	11111011	37777776	*
T5	1 20000000	0 26666770	11111011	37777776	*
T2	1 40000000	0 77777702	00000100	37777776	*
TP	1 40000000	0 00000000	11111001	37777776	OF

FIGURE DIV14 A/M = 60000000 60000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 40000000	0 00000001	00000000	37777777	
T8	1 00000000	0 00000000	00000001	40000000	
T5	1 00000001	1 40000002	00000001	40000000	
T2	1 00000002	0 37777775	00000010	37777777	
TR	1 00000004	0 26666701	11111110	37777777	*
T7	1 00000010	0 26667711	11110110	37777777	*
T4	1 00000020	0 26667731	11110110	37777777	*
T1	1 00000040	0 26667071	11111010	37777777	*
TP	1 00000100	0 26677171	11101010	37777777	*
T6	1 00000200	0 26677371	11101010	37777777	*
T3	1 00000400	0 26677771	11101010	37777777	*
T0	1 00001000	0 26670771	11111010	37777777	*
T8	1 00002000	0 26772771	11011010	37777777	*
T5	1 00004000	0 26776771	11011010	37777777	*
T2	1 00010000	0 26706771	11111010	37777777	*
TR	1 00020000	0 27726771	10111010	37777777	*
T7	1 00040000	0 27766771	10111010	37777777	*
T4	1 00100000	0 27066771	11111010	37777777	*
T1	1 00200000	0 37266771	01111010	37777777	*
TP	1 00400000	0 37666771	01111010	37777777	*
T6	1 01000000	0 30666771	11111010	37777777	*
T3	1 02000000	0 32666771	11111010	37777777	*
T0	1 04000000	0 36666771	11111010	37777777	*
T8	1 10000000	0 46666771	11111010	37777777	*
T5	1 20000000	0 66666771	11111010	37777777	
T2	1 40000001	1 77777702	00000100	40000000	
TP	1 37777777	1 77777777	00000100	40000000	OF

FIGURE DIV15 A/M = 44000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777771	0 00000000	00000000	40000010	
T8	0 00000000	0 00000001	00000000	40000010	
T5	0 00000001	1 40000012	00000000	40000010	
T2	0 00000002	0 37777703	00000111	37777767	
TR	0 00000004	0 26667717	11110101	37777767	*
T7	0 00000010	0 26667727	11110101	37777767	*
T4	0 00000020	0 26667747	11110101	37777767	*
T1	0 00000040	0 26667007	11111101	37777767	*
TP	0 00000100	0 26677107	11101101	37777767	*
T6	0 00000200	0 26677307	11101101	37777767	*
T3	0 00000400	0 26670707	11110101	37777767	*
TC	0 00001000	0 26771707	11010101	37777767	*
T8	0 00002000	0 26773707	11010101	37777767	*
T5	0 00004000	0 26777707	11010101	37777767	*
T2	0 00010000	0 26707707	11110101	37777767	*
TR	0 00020000	0 27727707	10110101	37777767	*
T7	0 00040000	0 27767707	10110101	37777767	*
T4	0 00100000	0 27087707	11110101	37777767	*
T1	0 00200000	0 37267707	01110101	37777767	*
TP	0 00400000	0 37667707	01110101	37777767	*
T6	0 01000000	0 30667707	11110101	37777767	*
T3	0 02000000	0 32667707	11110101	37777767	*
TC	0 04000000	0 36667707	11110101	37777767	*
T8	0 10000000	0 46667707	11110101	37777767	*
T5	0 20000000	0 66667707	11110101	37777767	
T2	0 40000001	1 77777010	00001010	40000010	
TP	0 37777777	1 00000010	00001010	40000010	OF

FIGURE DIV16 A/M = 50000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777720	0 00000000	00000000	40000100	
T8	0 00000000	0 00000020	00000000	40000100	
T5	0 00000001	1 40000140	00000000	40000100	
T2	0 00000002	0 37777177	00001001	37777677	
TR	0 00000004	0 26677177	11101101	37777677	*
T7	0 00000010	0 26677377	11101101	37777677	*
T4	0 00000020	0 26670777	11110101	37777677	*
T1	0 00000040	0 26771777	11010101	37777677	*
TP	0 00000100	0 26773777	11010101	37777677	*
T6	0 00000200	0 26777777	11010101	37777677	*
T3	0 00000400	0 26707777	11110101	37777677	*
TC	0 00001000	0 27727777	10110101	37777677	*
T8	0 00002000	0 27767777	10110101	37777677	*
T5	0 00004000	0 27067777	11110101	37777677	*
T2	0 00010000	0 37267777	01110101	37777677	*
TR	0 00020000	0 37667777	01110101	37777677	*
T7	0 00040000	0 30667777	11110101	37777677	*
T4	0 00100000	0 32667777	11110101	37777677	*
T1	0 00200000	0 36667777	11110101	37777677	*
TP	0 00400000	0 46667777	11110101	37777677	*
T6	0 01000000	0 66667777	11110101	37777677	
T3	0 02000001	1 37770200	00010100	40000100	*
TC	0 04000002	0 26777477	11001001	37777677	*
T8	0 10000004	0 26670777	11110101	37777677	*
T5	0 20000010	0 26771777	11010101	37777677	*
T2	0 40000020	0 77703170	00101010	37777677	*
TP	0 37777760	0 00002000	11010101	37777677	OF

FIGURE DIV17 A/M = 60000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777400	0 00000000	00000000	40001000	
T8	0 00000000	0 00000400	00000000	40001000	
T5	0 00000001	1 40002000	00000000	40001000	
T2	0 00000002	0 37772777	00010001	37776777	
TR	0 00000004	0 36773677	11011101	37776777	*
T7	0 00000010	0 26707677	11101101	37776777	*
T4	0 00000020	0 27717677	10101101	37776777	*
T1	0 00000040	0 27737677	10101101	37776777	*
TP	0 00000100	0 27777677	10101101	37776777	*
T6	0 00000200	0 27077677	11101101	37776777	*
T3	0 00000400	0 37277677	01101101	37776777	*
TC	0 00001000	0 37677677	01101101	37776777	*
T8	0 00002000	0 30677677	11101101	37776777	*
T5	0 00004000	0 32677677	11101101	37776777	*
T2	0 00010000	0 36677677	11101101	37776777	*
TR	0 00020000	0 46677677	11101101	37776777	*
T7	0 00040000	0 66677677	11101101	37776777	
T4	0 00100001	1 37702700	00100100	40001000	*
T1	0 00200002	0 27774777	10010001	37776777	*
TP	0 00400004	0 26707677	11101101	37776777	*
T6	0 01000010	0 27717677	10101101	37776777	*
T3	0 02000020	0 27737677	10101101	37776777	*
TC	0 04000040	0 27777677	10101101	37776777	*
T8	0 10000100	0 27077677	11101101	37776777	*
T5	0 20000200	0 37277677	01101101	37776777	*
T2	0 40000400	0 70771770	10010010	37776777	*
TP	0 37777400	0 00400000	01101101	37776777	0F

FIGURE DIV18 A/M = 40000000 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000001	00000000	40000001	
T8	0 00000001	0 00000000	00000000	40000001	
T5	0 00000003	1 40000001	00000000	40000001	
T2	0 00000006	0 37777770	00000011	37777776	
TR	0 00000014	0 26666770	11111011	37777776	*
T7	0 00000030	0 26666770	11111011	37777776	*
T4	0 00000060	0 26666770	11111011	37777776	*
T1	0 00000140	0 26666770	11111011	37777776	*
TP	0 00000300	0 26666770	11111011	37777776	*
T6	0 00000600	0 26666770	11111011	37777776	*
T3	0 00001400	0 26666770	11111011	37777776	*
T0	0 00003000	0 26666770	11111011	37777776	*
T8	0 00006000	0 26666770	11111011	37777776	*
T5	0 00014000	0 26666770	11111011	37777776	*
T2	0 00030000	0 26666770	11111011	37777776	*
TR	0 00060000	0 26666770	11111011	37777776	*
T7	0 00140000	0 26666770	11111011	37777776	*
T4	0 00300000	0 26666770	11111011	37777776	*
T1	0 00600000	0 26666770	11111011	37777776	*
TP	0 01400000	0 26666770	11111011	37777776	*
T6	0 03000000	0 26666770	11111011	37777776	*
T3	0 06000000	0 26666770	11111011	37777776	*
T0	0 14000000	0 26666770	11111011	37777776	*
T8	0 30000000	0 26666770	11111011	37777776	*
T5	0 60000000	0 26666770	11111011	37777776	*
T2	0 40000000	0 77777703	00000100	37777776	*
TP	0 40000000	0 00000000	11111001	37777776	OF

FIGURE DIV19 A/M = 40000002 00000000 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37777777	0 00000004	00000000	40000001	
T8	0 00000004	0 00000000	00000000	40000001	
T5	0 00000011	1 40000001	00000000	40000001	
T2	0 00000022	0 37777770	00000011	37777776	
TR	0 00000044	0 26666770	11111011	37777776	*
T7	0 00000110	0 26666770	11111011	37777776	*
T4	0 00000220	0 26666770	11111011	37777776	*
T1	0 00000440	0 26666770	11111011	37777776	*
TP	0 00001100	0 26666770	11111011	37777776	*
T6	0 00002200	0 26666770	11111011	37777776	*
T3	0 00004400	0 26666770	11111011	37777776	*
TC	0 00011000	0 26666770	11111011	37777776	*
T8	0 00022000	0 26666770	11111011	37777776	*
T5	0 00044000	0 26666770	11111011	37777776	*
T2	0 00110000	0 26666770	11111011	37777776	*
TR	0 00220000	0 26666770	11111011	37777776	*
T7	0 00440000	0 26666770	11111011	37777776	*
T4	0 01100000	0 26666770	11111011	37777776	*
T1	0 02200000	0 26666770	11111011	37777776	*
TP	0 04400000	0 26666770	11111011	37777776	*
T6	0 11000000	0 26666770	11111011	37777776	*
T3	0 22000000	0 26666770	11111011	37777776	*
TC	0 44000000	0 26666770	11111011	37777776	*
T8	0 10000000	0 26666771	11111011	37777776	*
T5	0 20000000	0 26666772	11111011	37777776	*
T2	0 40000000	0 77777706	00000100	37777776	*
TP	0 40000000	0 00000002	11111011	37777776	OF

FIGURE DIV20 A/M = 40000002 00000001 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	0 17777777	0 00000002	00000000	60000000
T8	0 00000002	0 77777777	00000000	60000000
T5	0 00000004	0 06666665	11111111	17777777
T2	0 00000011	1 17777774	00000000	60000000
TR	0 00000023	1 17777770	00000000	60000000
T7	0 00000047	1 17777760	00000000	60000000
T4	0 00000117	1 17777740	00000000	60000000
T1	0 00000237	1 17777700	00000000	60000000
TP	0 00000477	1 17777600	00000000	60000000
T6	0 00001177	1 17777400	00000000	60000000
T3	0 00002377	1 17777000	00000000	60000000
T0	0 00004777	1 17776000	00000000	60000000
T8	0 00011777	1 17774000	00000000	60000000
T5	0 00023777	1 17770000	00000000	60000000
T2	0 00047777	1 17760000	00000000	60000000
TR	0 00117777	1 17740000	00000000	60000000
T7	0 00237777	1 17700000	00000000	60000000
T4	0 00477777	1 17600000	00000000	60000000
T1	0 01177777	1 17400000	00000000	60000000
TP	0 02377777	1 17000000	00000000	60000000
T6	0 04777777	1 16000000	00000000	60000000
T3	0 11777777	1 14000000	00000000	60000000
T0	0 23777777	1 10000000	00000000	60000000
T8	0 47777777	1 00000000	00000000	60000000
T5	0 17777777	1 60000001	00000000	60000000
T2	0 37777776	1 40000002	00000000	17777777
TP	0 40000002	0 00000001	00000011	17777777

FIGURE DIV21

A/M = 37777776 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000001	0 00000000	00000000	60000000
T8	1 00000000	0 77777776	00000001	60000000
T5	1 00000000	0 06666665	11111111	17777777
T2	1 00000001	1 17777774	00000000	60000000
TR	1 00000003	1 17777770	00000000	60000000
T7	1 00000007	1 17777760	00000000	60000000
T4	1 00000017	1 17777740	00000000	60000000
T1	1 00000037	1 17777700	00000000	60000000
TP	1 00000077	1 17777600	00000000	60000000
T6	1 00000177	1 17777400	00000000	60000000
T3	1 00000377	1 17777000	00000000	60000000
TC	1 00000777	1 17776000	00000000	60000000
T8	1 00001777	1 17774000	00000000	60000000
T5	1 00003777	1 17770000	00000000	60000000
T2	1 00007777	1 17760000	00000000	60000000
TR	1 00017777	1 17740000	00000000	60000000
T7	1 00037777	1 17700000	00000000	60000000
T4	1 00077777	1 17600000	00000000	60000000
T1	1 00177777	1 17400000	00000000	60000000
TP	1 00377777	1 17000000	00000000	60000000
T6	1 00777777	1 16000000	00000000	60000000
T3	1 01777777	1 14000000	00000000	60000000
TC	1 03777777	1 10000000	00000000	60000000
T8	1 07777777	1 00000000	00000000	60000000
T5	1 17777777	1 60000000	00000000	60000000
T2	1 37777776	1 40000000	00000000	17777777
TP	1 37777776	0 00000000	00000001	17777777

FIGURE DIV22 A/M = 37777774 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000002	0 00000000	00000000	60000000
T8	1 00000000	0 77777775	00000001	60000000
T5	1 00000000	0 06666663	11111111	17777777
T2	1 00000001	1 17777770	00000000	60000000
TR	1 00000003	1 17777760	00000000	60000000
T7	1 00000007	1 17777740	00000000	60000000
T4	1 00000017	1 17777700	00000000	60000000
T1	1 00000037	1 17777600	00000000	60000000
TP	1 00000077	1 17777400	00000000	60000000
T6	1 00000177	1 17777000	00000000	60000000
T3	1 00000377	1 17776000	00000000	60000000
T0	1 00000777	1 17774000	00000000	60000000
T8	1 00001777	1 17770000	00000000	60000000
T5	1 00003777	1 17760000	00000000	60000000
T2	1 00007777	1 17740000	00000000	60000000
TR	1 00017777	1 17700000	00000000	60000000
T7	1 00037777	1 17600000	00000000	60000000
T4	1 00077777	1 17400000	00000000	60000000
T1	1 00177777	1 17000000	00000000	60000000
TP	1 00377777	1 16000000	00000000	60000000
T6	1 00777777	1 14000000	00000000	60000000
T3	1 01777777	1 10000000	00000000	60000000
T0	1 03777777	1 00000000	00000000	60000000
T8	1 07777777	1 60000000	00000000	60000000
T5	1 17777776	0 57777777	00000001	17777777
T2	1 37777774	1 37777770	00000010	17777777
TP	1 37777774	0 00000000	11111101	17777777

FIGURE DIV23 A/M = 37777770 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000004	0 00000000	00000000	60000000
T8	1 00000000	0 77777773	00000001	60000000
T5	1 00000000	0 06666667	11111101	17777777
T2	1 00000001	1 17777760	00000000	60000000
TR	1 00000003	1 17777740	00000000	60000000
T7	1 00000007	1 17777700	00000000	60000000
T4	1 00000017	1 17777600	00000000	60000000
T1	1 00000037	1 17777400	00000000	60000000
TP	1 00000077	1 17777000	00000000	60000000
T6	1 00000177	1 17776000	00000000	60000000
T3	1 00000377	1 17774000	00000000	60000000
T0	1 00000777	1 17770000	00000000	60000000
T8	1 00001777	1 17760000	00000000	60000000
T5	1 00003777	1 17740000	00000000	60000000
T2	1 00007777	1 17700000	00000000	60000000
TR	1 00017777	1 17600000	00000000	60000000
T7	1 00037777	1 17400000	00000000	60000000
T4	1 00077777	1 17000000	00000000	60000000
T1	1 00177777	1 16000000	00000000	60000000
TP	1 00377777	1 14000000	00000000	60000000
T6	1 00777777	1 10000000	00000000	60000000
T3	1 01777777	1 00000000	00000000	60000000
T0	1 03777777	1 60000000	00000000	60000000
T8	1 07777776	0 57777777	00000001	17777777
T5	1 17777774	0 46666677	11111101	17777777
T2	1 37777770	1 37777770	00000010	17777777
TP	1 37777770	0 00000000	11111101	17777777

FIGURE DIV24 A/M = 37777760 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000010	0 00000000	00000000	60000000
T8	1 00000000	0 77777767	00000001	60000000
T5	1 00000000	0 06666657	11111101	17777777
T2	1 00000001	1 17777740	00000000	60000000
TR	1 00000003	1 17777700	00000000	60000000
T7	1 00000007	1 17777600	00000000	60000000
T4	1 00000017	1 17777400	00000000	60000000
T1	1 00000037	1 17777000	00000000	60000000
TP	1 00000077	1 17776000	00000000	60000000
T6	1 00000177	1 17774000	00000000	60000000
T3	1 00000377	1 17770000	00000000	60000000
T0	1 00000777	1 17760000	00000000	60000000
T8	1 00001777	1 17740000	00000000	60000000
T5	1 00003777	1 17700000	00000000	60000000
T2	1 00007777	1 17600000	00000000	60000000
TR	1 00017777	1 17400000	00000000	60000000
T7	1 00037777	1 17000000	00000000	60000000
T4	1 00077777	1 16000000	00000000	60000000
T1	1 00177777	1 14000000	00000000	60000000
TP	1 00377777	1 10000000	00000000	60000000
T6	1 00777777	1 00000000	00000000	60000000
T3	1 01777777	1 60000000	00000000	60000000
T0	1 03777776	0 57777777	00000001	17777777
T8	1 07777774	0 46666677	11111101	17777777
T5	1 17777770	0 46666677	11111101	17777777
T2	1 37777760	1 37777770	00000010	17777777
TP	1 37777760	0 00000000	11111101	17777777

FIGURE DIV25 A/M = 37777740 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000020	0 00000000	00000000	60000000
T8	1 00000000	0 77777757	00000001	60000000
T5	1 00000000	0 06666637	11111101	17777777
T2	1 00000001	1 17777600	00000100	60000000
TR	1 00000003	1 17777600	00000000	60000000
T7	1 00000007	1 17777400	00000000	60000000
T4	1 00000017	1 17777000	00000000	60000000
T1	1 00000037	1 17776000	00000000	60000000
TP	1 00000077	1 17774000	00000000	60000000
T6	1 00000177	1 17770000	00000000	60000000
T3	1 00000377	1 17760000	00000000	60000000
T0	1 00000777	1 17740000	00000000	60000000
T8	1 00001777	1 17700000	00000000	60000000
T5	1 00003777	1 17600000	00000000	60000000
T2	1 00007777	1 17400000	00000000	60000000
TR	1 00017777	1 17000000	00000000	60000000
T7	1 00037777	1 16000000	00000000	60000000
T4	1 00077777	1 14000000	00000000	60000000
T1	1 00177777	1 10000000	00000000	60000000
TP	1 00377777	1 00000000	00000000	60000000
T6	1 00777777	1 60000000	00000000	60000000
T3	1 01777776	0 57777777	00000001	17777777
TC	1 03777774	0 46666677	11111101	17777777
T8	1 07777770	0 46666677	11111101	17777777
T5	1 17777760	0 46666677	11111101	17777777
T2	1 37777740	1 37777770	00000010	17777777
TP	1 37777740	0 00000000	11111101	17777777

FIGURE DIV26 A/M = 37777700 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000040	0 00000000	00000000	60000000
T8	1 00000000	0 77777737	00000001	60000000
T5	1 00000000	0 06666577	11111101	17777777
T2	1 00000001	1 17777500	00000100	60000000
TR	1 00000003	1 17777400	00000000	60000000
T7	1 00000007	1 17777000	00000000	60000000
T4	1 00000017	1 17776000	00000000	60000000
T1	1 00000037	1 17774000	00000000	60000000
TP	1 00000077	1 17770000	00000000	60000000
T6	1 00000177	1 17760000	00000000	60000000
T3	1 00000377	1 17740000	00000000	60000000
TC	1 00000777	1 17700000	00000000	60000000
T8	1 00001777	1 17600000	00000000	60000000
T5	1 00003777	1 17400000	00000000	60000000
T2	1 00007777	1 17000000	00000000	60000000
TR	1 00017777	1 16000000	00000000	60000000
T7	1 00037777	1 14000000	00000000	60000000
T4	1 00077777	1 10000000	00000000	60000000
T1	1 00177777	1 00000000	00000000	60000000
TP	1 00377777	1 60000000	00000000	60000000
T6	1 00777776	0 57777777	00000001	17777777
T3	1 01777774	0 46666677	11111101	17777777
TC	1 03777770	0 46666677	11111101	17777777
T8	1 07777760	0 46666677	11111101	17777777
T5	1 17777740	0 46666677	11111101	17777777
T2	1 37777700	1 37777770	00000010	17777777
TP	1 37777700	0 00000000	11111101	17777777

FIGURE DIV27 A/M = 37777600 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000100	0 00000000	00000000	60000000
T8	1 00000000	0 77777677	00000001	60000000
T5	1 00000000	0 06666477	11111101	17777777
T2	1 00000001	1 17777300	00000100	60000000
TR	1 00000003	1 17777000	00000000	60000000
T7	1 00000007	1 17776000	00000000	60000000
T4	1 00000017	1 17774000	00000000	60000000
T1	1 00000037	1 17770000	00000000	60000000
TP	1 00000077	1 17760000	00000000	60000000
T6	1 00000177	1 17740000	00000000	60000000
T3	1 00000377	1 17700000	00000000	60000000
T0	1 00000777	1 17600000	00000000	60000000
T8	1 00001777	1 17400000	00000000	60000000
T5	1 00003777	1 17000000	00000000	60000000
T2	1 00007777	1 16000000	00000000	60000000
TR	1 00017777	1 14000000	00000000	60000000
T7	1 00037777	1 10000000	00000000	60000000
T4	1 00077777	1 00000000	00000000	60000000
T1	1 00177777	1 60000000	00000000	60000000
TP	1 00377776	0 57777777	00000001	17777777
T6	1 00777774	0 46666677	11111101	17777777
T3	1 01777770	0 46666677	11111101	17777777
T0	1 03777760	0 46666677	11111101	17777777
T8	1 07777740	0 46666677	11111101	17777777
T5	1 17777700	0 46666677	11111101	17777777
T2	1 37777600	1 37777770	00000010	17777777
TP	1 37777600	0 00000000	11111101	17777777

FIGURE DIV28 A/M = 37777400 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000200	0 00000000	00000000	60000000
T8	1 00000000	0 77777577	00000001	60000000
T5	1 00000000	0 06666277	11111101	17777777
T2	1 00000001	1 17776700	00000100	60000000
TR	1 00000003	1 17776000	00000000	60000000
T7	1 00000007	1 17774000	00000000	60000000
T4	1 00000017	1 17770000	00000000	60000000
T1	1 00000037	1 17760000	00000000	60000000
TP	1 00000077	1 17740000	00000000	60000000
T6	1 00000177	1 17700000	00000000	60000000
T3	1 00000377	1 17600000	00000000	60000000
T0	1 00000777	1 17400000	00000000	60000000
T8	1 00001777	1 17000000	00000000	60000000
T5	1 00003777	1 16000000	00000000	60000000
T2	1 00007777	1 14000000	00000000	60000000
TR	1 00017777	1 10000000	00000000	60000000
T7	1 00037777	1 00000000	00000000	60000000
T4	1 00077777	1 60000000	00000000	60000000
T1	1 00177776	0 57777777	00000001	17777777
TP	1 00377774	0 46666677	11111101	17777777
T6	1 00777770	0 46666677	11111101	17777777
T3	1 01777760	0 46666677	11111101	17777777
T0	1 03777740	0 46666677	11111101	17777777
T8	1 07777700	0 46666677	11111101	17777777
T5	1 17777600	0 46666677	11111101	17777777
T2	1 37777400	1 37777770	00000010	17777777
TP	1 37777400	0 00000000	11111101	17777777

FIGURE DIV29

A/M = 37777000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60000400	0 00000000	00000000	60000000
T8	1 00000000	0 77777377	00000001	60000000
T5	1 00000000	0 06665677	11111101	17777777
T2	1 00000001	1 17775700	00000100	60000000
TR	1 00000003	1 17774000	00000000	60000000
T7	1 00000007	1 17770000	00000000	60000000
T4	1 00000017	1 17760000	00000000	60000000
T1	1 00000037	1 17740000	00000000	60000000
TP	1 00000077	1 17700000	00000000	60000000
T6	1 00000177	1 17600000	00000000	60000000
T3	1 00000377	1 17400000	00000000	60000000
T0	1 00000777	1 17000000	00000000	60000000
T8	1 00001777	1 16000000	00000000	60000000
T5	1 00003777	1 14000000	00000000	60000000
T2	1 00007777	1 10000000	00000000	60000000
TR	1 00017777	1 00000000	00000000	60000000
T7	1 00037777	1 60000000	00000000	60000000
T4	1 00077776	0 67777777	00000001	17777777
T1	1 00177774	0 46666677	11111101	17777777
TP	1 00377770	0 46666677	11111101	17777777
T6	1 00777760	0 46666677	11111101	17777777
T3	1 01777740	0 46666677	11111101	17777777
T0	1 03777700	0 46666677	11111101	17777777
T8	1 07777600	0 46666677	11111101	17777777
T5	1 17777400	0 46666677	11111101	17777777
T2	1 37777000	1 37777770	00000010	17777777
TP	1 37777000	0 00000000	11111101	17777777

FIGURE DIV30 A/M = 37776000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60001000	0 00000000	00000000	60000000
T8	1 00000000	0 77776777	00000001	60000000
T5	1 00000000	0 06664677	11111101	17777777
T2	1 00000001	1 17773700	00000100	60000000
TR	1 00000003	1 17760000	00010000	60000000
T7	1 00000007	1 17760000	00000000	60000000
T4	1 00000017	1 17740000	00000000	60000000
T1	1 00000037	1 17700000	00000000	60000000
TP	1 00000077	1 17600000	00000000	60000000
T6	1 00000177	1 17400000	00000000	60000000
T3	1 00000377	1 17000000	00000000	60000000
T0	1 00000777	1 16000000	00000000	60000000
T8	1 00001777	1 14000000	00000000	60000000
T5	1 00003777	1 10000000	00000000	60000000
T2	1 00007777	1 00000000	00000000	60000000
TR	1 00017777	1 60000000	00000000	60000000
T7	1 00037776	0 57777777	00000001	17777777
T4	1 00077774	0 46666677	11111101	17777777
T1	1 00177770	0 46666677	11111101	17777777
TP	1 00377760	0 46666677	11111101	17777777
T6	1 00777740	0 46666677	11111101	17777777
T3	1 01777700	0 46666677	11111101	17777777
T0	1 03777600	0 46666677	11111101	17777777
T8	1 07777400	0 46666677	11111101	17777777
T5	1 17777000	0 46666677	11111101	17777777
T2	1 37776000	1 37777770	00000010	17777777
TP	1 37776000	0 00000000	11111101	17777777

FIGURE DIV31

A/M = 37774000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60002000	0 00000000	00000000	60000000
T8	1 00000000	0 37775777	00000001	60000000
T5	1 00000000	0 06662677	11111101	17777777
T2	1 00000001	1 17767700	00000100	60000000
TR	1 00000003	1 17750000	00010000	60000000
T7	1 00000007	1 17740000	00000000	60000000
T4	1 00000017	1 17700000	00000000	60000000
T1	1 00000037	1 17600000	00000000	60000000
TP	1 00000077	1 17400000	00000000	60000000
T6	1 00000177	1 17000000	00000000	60000000
T3	1 00000377	1 16000000	00000000	60000000
T0	1 00000777	1 14000000	00000000	60000000
T8	1 00001777	1 10000000	00000000	60000000
T5	1 00003777	1 00000000	00000000	60000000
T2	1 00007777	1 60000000	00000000	60000000
TR	1 00017776	0 57777777	00000001	17777777
T7	1 00037774	0 46666677	11111101	17777777
T4	1 00077770	0 46666677	11111101	17777777
T1	1 00177760	0 46666677	11111101	17777777
TP	1 00377740	0 46666677	11111101	17777777
T6	1 00777700	0 46666677	11111101	17777777
T3	1 01777600	0 46666677	11111101	17777777
TC	1 03777400	0 46666677	11111101	17777777
T8	1 07777000	0 46666677	11111101	17777777
T5	1 17776000	0 46666677	11111101	17777777
T2	1 37774000	1 37777770	00000010	17777777
TP	1 37774000	0 00000000	11111101	17777777

FIGURE DIV32 A/M = 37770000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60004000	0 00000000	00000000	60000000
T8	1 00000000	0 77773777	00000001	60000000
T5	1 00000000	0 06656677	11111101	17777777
T2	1 00000001	1 17757700	00000100	60000000
TR	1 00000003	1 17730000	00010000	60000000
T7	1 00000007	1 17700000	00000000	60000000
T4	1 00000017	1 17600000	00000000	60000000
T1	1 00000037	1 17400000	00000000	60000000
TP	1 00000077	1 17000000	00000000	60000000
T6	1 00000177	1 16000000	00000000	60000000
T3	1 00000377	1 14000000	00000000	60000000
T0	1 00000777	1 10000000	00000000	60000000
T8	1 00001777	1 00000000	00000000	60000000
T5	1 00003777	1 60000000	00000000	60000000
T2	1 00007776	0 57777777	00000001	17777777
TR	1 00017774	0 46666677	11111101	17777777
T7	1 00037770	0 46666677	11111101	17777777
T4	1 00077760	0 46666677	11111101	17777777
T1	1 00177740	0 46666677	11111101	17777777
TP	1 00377700	0 46666677	11111101	17777777
T6	1 00777600	0 46666677	11111101	17777777
T3	1 01777400	0 46666677	11111101	17777777
T0	1 03777000	0 46666677	11111101	17777777
T8	1 07776000	0 46666677	11111101	17777777
T5	1 17774000	0 46666677	11111101	17777777
T2	1 37770000	1 37777770	00000010	17777777
TP	1 37770000	0 00000000	11111101	17777777

FIGURE DIV33 A/M = 37760000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60010000	0 00000000	00000000	60000000
T8	1 00000000	0 27767777	00000001	60000000
T5	1 00000000	0 06646677	11111101	17777777
T2	1 00000001	1 17737700	00000100	60000000
TR	1 00000003	1 17670000	00010000	60000000
T7	1 00000007	1 17600000	00000000	60000000
T4	1 00000017	1 17400000	00000000	60000000
T1	1 00000037	1 17000000	00000000	60000000
TP	1 00000077	1 16000000	00000000	60000000
T6	1 00000177	1 14000000	00000000	60000000
T3	1 00000377	1 10000000	00000000	60000000
TC	1 00000777	1 00000000	00000000	60000000
T8	1 00001777	1 60000000	00000000	60000000
T5	1 00003776	0 57777777	00000001	17777777
T2	1 00007774	0 46666677	11111101	17777777
TR	1 00017770	0 46666677	11111101	17777777
T7	1 00037760	0 46666677	11111101	17777777
T4	1 00077740	0 46666677	11111101	17777777
T1	1 00177700	0 46666677	11111101	17777777
TP	1 00377600	0 46666677	11111101	17777777
T6	1 00777400	0 46666677	11111101	17777777
T3	1 01777000	0 46666677	11111101	17777777
TC	1 03776000	0 46666677	11111101	17777777
T8	1 07774000	0 46666677	11111101	17777777
T5	1 17770000	0 46666677	11111101	17777777
T2	1 37760000	1 37777770	00000010	17777777
TP	1 37760000	0 00000000	11111101	17777777

FIGURE DIV34 A/M = 37740000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60020000	0 00000000	00000000	60000000
T8	1 00000000	0 77757777	00000001	60000000
T5	1 00000000	0 06626677	11111101	17777777
T2	1 00000001	1 17677700	00000100	60000000
TR	1 00000003	1 17670000	00010000	60000000
T7	1 00000007	1 17400000	00000000	60000000
T4	1 00000017	1 17000000	00000000	60000000
T1	1 00000037	1 16000000	00000000	60000000
TP	1 00000077	1 14000000	00000000	60000000
T6	1 00000177	1 10000000	00000000	60000000
T3	1 00000377	1 00000000	00000000	60000000
TC	1 00000777	1 60000000	00000000	60000000
T8	1 00001776	0 67777777	00000001	17777777
T5	1 00003774	0 46666677	11111101	17777777
T2	1 00007770	0 46666677	11111101	17777777
TR	1 00017760	0 46666677	11111101	17777777
T7	1 00037740	0 46666677	11111101	17777777
T4	1 00077700	0 46666677	11111101	17777777
T1	1 00177600	0 46666677	11111101	17777777
TP	1 00377400	0 46666677	11111101	17777777
T6	1 00777000	0 46666677	11111101	17777777
T3	1 01776000	0 46666677	11111101	17777777
TC	1 03774000	0 46666677	11111101	17777777
T8	1 07770000	0 46666677	11111101	17777777
T5	1 17760000	0 46666677	11111101	17777777
T2	1 37740000	1 37777770	00000010	17777777
TP	1 37740000	0 00000000	11111101	17777777

FIGURE DIV35

A/M = 37700000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60040000	0 00000000	00000000	60000000
T8	1 00000000	0 77737777	00000001	60000000
T5	1 00000000	0 06566677	11111101	17777777
T2	1 00000001	1 17577700	00000100	60000000
TR	1 00000003	1 17370000	00010000	60000000
T7	1 00000007	1 16000000	01000000	60000000
T4	1 00000017	1 16000000	00000000	60000000
T1	1 00000037	1 14000000	00000000	60000000
TP	1 00000077	1 10000000	00000000	60000000
T6	1 00000177	1 00000000	00000000	60000000
T3	1 00000377	1 60000000	00000000	60000000
TC	1 00000776	0 57777777	00000001	17777777
T8	1 00001774	0 46666677	11111101	17777777
T5	1 00003770	0 46666677	11111101	17777777
T2	1 00007760	0 46666677	11111101	17777777
TR	1 00017740	0 46666677	11111101	17777777
T7	1 00037700	0 46666677	11111101	17777777
T4	1 00077600	0 46666677	11111101	17777777
T1	1 00177400	0 46666677	11111101	17777777
TP	1 00377000	0 46666677	11111101	17777777
T6	1 00776000	0 46666677	11111101	17777777
T3	1 01774000	0 46666677	11111101	17777777
TC	1 03770000	0 46666677	11111101	17777777
T8	1 07760000	0 46666677	11111101	17777777
T5	1 17740000	0 46666677	11111101	17777777
T2	1 37700000	1 37777770	00000010	17777777
TP	1 37700000	0 00000000	11111101	17777777

FIGURE DIV36 A/M = 37600000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60100000	0 00000000	00000000	60000000
T8	1 00000000	0 77677777	00000001	60000000
T5	1 00000000	0 06466677	11111101	17777777
T2	1 00000001	1 17377700	00000100	60000000
TR	1 00000003	1 16770000	00010000	60000000
T7	1 00000007	1 15000000	01000000	60000000
T4	1 00000017	1 14000000	00000000	60000000
T1	1 00000037	1 10000000	00000000	60000000
TP	1 00000077	1 00000000	00000000	60000000
T6	1 00000177	1 60000000	00000000	60000000
T3	1 00000376	0 57777777	00000001	17777777
TC	1 00000774	0 46666677	11111101	17777777
T8	1 00001770	0 46666677	11111101	17777777
T5	1 00003760	0 46666677	11111101	17777777
T2	1 00007740	0 46666677	11111101	17777777
TR	1 00017700	0 46666677	11111101	17777777
T7	1 00037600	0 46666677	11111101	17777777
T4	1 00077400	0 46666677	11111101	17777777
T1	1 00177000	0 46666677	11111101	17777777
TP	1 00376000	0 46666677	11111101	17777777
T6	1 00774000	0 46666677	11111101	17777777
T3	1 01770000	0 46666677	11111101	17777777
TC	1 03760000	0 46666677	11111101	17777777
T8	1 07740000	0 46666677	11111101	17777777
T5	1 17700000	0 46666677	11111101	17777777
T2	1 37600000	1 37777770	00000010	17777777
TP	1 37600000	0 00000000	11111101	17777777

FIGURE DIV37 A/M = 37400000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60200000	0 00000000	00000000	60000000
T8	1 00000000	0 77677777	00000001	60000000
T5	1 00000000	0 06266677	11111101	17777777
T2	1 00000001	1 16777700	00000100	60000000
TR	1 00000003	1 15770000	00010000	60000000
T7	1 00000007	1 13000000	01000000	60000000
T4	1 00000017	1 10000000	00000000	60000000
T1	1 00000037	1 00000000	00000000	60000000
TP	1 00000077	1 60000000	00000000	60000000
T6	1 00000176	0 57777777	00000001	17777777
T3	1 00000374	0 46666677	11111101	17777777
T0	1 00000770	0 46666677	11111101	17777777
T8	1 00001760	0 46666677	11111101	17777777
T5	1 00003740	0 46666677	11111101	17777777
T2	1 00007700	0 46666677	11111101	17777777
TR	1 00017600	0 46666677	11111101	17777777
T7	1 00037400	0 46666677	11111101	17777777
T4	1 00077000	0 46666677	11111101	17777777
T1	1 00176000	0 46666677	11111101	17777777
TP	1 00374000	0 46666677	11111101	17777777
T6	1 00770000	0 46666677	11111101	17777777
T3	1 01760000	0 46666677	11111101	17777777
TC	1 03740000	0 46666677	11111101	17777777
T8	1 07700000	0 46666677	11111101	17777777
T5	1 17600000	0 46666677	11111101	17777777
T2	1 37400000	1 37777770	00000010	17777777
TP	1 37400000	0 00000000	11111101	17777777

FIGURE DIV38 A/M = 37000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 60400000	0 00000000	00000000	60000000
T8	1 00000000	0 77377777	00000001	60000000
T5	1 00000000	0 05666677	11111101	17777777
T2	1 00000001	1 15777700	00000100	60000000
TR	1 00000003	1 13770000	00010000	60000000
T7	1 00000007	1 07000000	01000000	60000000
T4	1 00000017	1 00000000	00000000	60000000
T1	1 00000037	1 60000000	00000000	60000000
TP	1 00000076	0 57777777	00000001	17777777
T6	1 00000174	0 46666677	11111101	17777777
T3	1 00000370	0 46666677	11111101	17777777
T0	1 00000760	0 46666677	11111101	17777777
T8	1 00001740	0 46666677	11111101	17777777
T5	1 00003700	0 46666677	11111101	17777777
T2	1 00007600	0 46666677	11111101	17777777
TR	1 00017400	0 46666677	11111101	17777777
T7	1 00037000	0 46666677	11111101	17777777
T4	1 00076000	0 46666677	11111101	17777777
T1	1 00174000	0 46666677	11111101	17777777
TP	1 00370000	0 46666677	11111101	17777777
T6	1 00760000	0 46666677	11111101	17777777
T3	1 01740000	0 46666677	11111101	17777777
T0	1 03700000	0 46666677	11111101	17777777
T8	1 07600000	0 46666677	11111101	17777777
T5	1 17400000	0 46666677	11111101	17777777
T2	1 37000000	1 37777770	00000010	17777777
TP	1 37000000	0 00000000	11111101	17777777

FIGURE DIV39 A/M = 36000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 61000000	0 00000000	00000000	60000000	
T8	1 00000000	0 76777777	00000001	60000000	
T5	1 00000000	0 04666677	11111101	17777777	
T2	1 00000001	1 13777700	00000100	60000000	
TR	1 00000003	1 07770000	00010000	60000000	
T7	1 00000007	1 77000000	01000000	60000000	
T4	1 00000017	1 60000000	00000000	60000000	*
T1	1 00000036	0 57777777	00000001	17777777	
TP	1 00000074	0 46666677	11111101	17777777	
T6	1 00000170	0 46666677	11111101	17777777	
T3	1 00000360	0 46666677	11111101	17777777	
TC	1 00000740	0 46666677	11111101	17777777	
T8	1 00001700	0 46666677	11111101	17777777	
T5	1 00003600	0 46666677	11111101	17777777	
T2	1 00007400	0 46666677	11111101	17777777	
TR	1 00017000	0 46666677	11111101	17777777	
T7	1 00036000	0 46666677	11111101	17777777	
T4	1 00074000	0 46666677	11111101	17777777	
T1	1 00170000	0 46666677	11111101	17777777	
TP	1 00360000	0 46666677	11111101	17777777	
T6	1 00740000	0 46666677	11111101	17777777	
T3	1 01700000	0 46666677	11111101	17777777	
TC	1 03600000	0 46666677	11111101	17777777	
T8	1 07400000	0 46666677	11111101	17777777	
T5	1 17000000	0 46666677	11111101	17777777	
T2	1 36000000	1 37777770	00000010	17777777	
TP	1 36000000	0 00000000	11111101	17777777	

FIGURE DIV40 A/M = 34000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62000000	0 00000000	00000000	60000000	
T8	1 00000000	0 75777777	00000001	60000000	
T5	1 00000000	0 02666677	11111101	17777777	
T2	1 00000001	1 07777700	00000100	60000000	
TR	1 00000003	1 77770000	00010000	60000000	
T7	1 00000007	1 67000000	01000000	60000000	*
T4	1 00000016	0 57777777	00000001	17777777	
T1	1 00000034	0 46666677	11111101	17777777	
TP	1 00000070	0 46666677	11111101	17777777	
T6	1 00000160	0 46666677	11111101	17777777	
T3	1 00000340	0 46666677	11111101	17777777	
TC	1 00000700	0 46666677	11111101	17777777	
T8	1 00001600	0 46666677	11111101	17777777	
T5	1 00003400	0 46666677	11111101	17777777	
T2	1 00007000	0 46666677	11111101	17777777	
TR	1 00016000	0 46666677	11111101	17777777	
T7	1 00034000	0 46666677	11111101	17777777	
T4	1 00070000	0 46666677	11111101	17777777	
T1	1 00160000	0 46666677	11111101	17777777	
TP	1 00340000	0 46666677	11111101	17777777	
T6	1 00700000	0 46666677	11111101	17777777	
T3	1 01600000	0 46666677	11111101	17777777	
TC	1 03400000	0 46666677	11111101	17777777	
T8	1 07000000	0 46666677	11111101	17777777	
T5	1 16000000	0 46666677	11111101	17777777	
T2	1 34000000	1 37777770	00000010	17777777	
TP	1 34000000	0 00000000	11111101	17777777	

FIGURE DIV41

A/M = 30000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 64000000	0 00000000	00000000	60000000	
T8	1 00000000	0 73777777	00000001	60000000	
T5	1 00000000	0 76666677	11111101	17777777	
T2	1 00000001	1 77777700	00000100	60000000	*
TR	1 00000003	1 57770000	00010000	60000000	*
T7	1 00000006	0 46777777	11000001	17777777	
T4	1 00000014	0 46666677	11111101	17777777	
T1	1 00000030	0 46666677	11111101	17777777	
TP	1 00000060	0 46666677	11111101	17777777	
T6	1 00000140	0 46666677	11111101	17777777	
T3	1 00000300	0 46666677	11111101	17777777	
T0	1 00000600	0 46666677	11111101	17777777	
T8	1 00001400	0 46666677	11111101	17777777	
T5	1 00003000	0 46666677	11111101	17777777	
T2	1 00006000	0 46666677	11111101	17777777	
TR	1 00014000	0 46666677	11111101	17777777	
T7	1 00030000	0 46666677	11111101	17777777	
T4	1 00060000	0 46666677	11111101	17777777	
T1	1 00140000	0 46666677	11111101	17777777	
TP	1 00300000	0 46666677	11111101	17777777	
T6	1 00600000	0 46666677	11111101	17777777	
T3	1 01400000	0 46666677	11111101	17777777	
T0	1 03000000	0 46666677	11111101	17777777	
T8	1 06000000	0 46666677	11111101	17777777	
T5	1 14000000	0 46666677	11111101	17777777	
T2	1 30000000	1 37777770	00000010	17777777	
TP	1 30000000	0 00000000	11111101	17777777	

FIGURE DIV42

A/M = 20000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 70000000	0 00000000	00000000	60000000	
T8	1 00000000	0 67777777	00000001	60000000	
T5	1 00000000	0 66666677	11111101	17777777	
T2	1 00000001	1 57777700	00000100	60000000	*
TR	1 00000002	0 46667777	11110001	17777777	
T7	1 00000004	0 46666677	11111101	17777777	
T4	1 00000010	0 46666677	11111101	17777777	
T1	1 00000020	0 46666677	11111101	17777777	
TP	1 00000040	0 46666677	11111101	17777777	
T6	1 00000100	0 46666677	11111101	17777777	
T3	1 00000200	0 46666677	11111101	17777777	
T0	1 00000400	0 46666677	11111101	17777777	
T8	1 00001000	0 46666677	11111101	17777777	
T5	1 00002000	0 46666677	11111101	17777777	
T2	1 00004000	0 46666677	11111101	17777777	
TR	1 00010000	0 46666677	11111101	17777777	
T7	1 00020000	0 46666677	11111101	17777777	
T4	1 00040000	0 46666677	11111101	17777777	
T1	1 00100000	0 46666677	11111101	17777777	
TP	1 00200000	0 46666677	11111101	17777777	
T6	1 00400000	0 46666677	11111101	17777777	
T3	1 01000000	0 46666677	11111101	17777777	
T0	1 02000000	0 46666677	11111101	17777777	
T8	1 04000000	0 46666677	11111101	17777777	
T5	1 10000000	0 46666677	11111101	17777777	
T2	1 20000000	1 37777770	00000010	17777777	
TP	1 20000000	0 00000000	11111101	17777777	

FIGURE DIV43 A/M = 20000000 00000000 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 00000000	0 00000000	00000000	40000000	
T8	1 00000000	0 37777777	00000001	40000000	
T5	1 00000000	0 66666677	11111101	37777777	
T2	1 00000001	1 37777700	00000100	40000000	*
TR	1 00000002	0 26667777	11110001	37777777	*
T7	1 00000004	0 26666677	11111101	37777777	*
T4	1 00000010	0 26666677	11111101	37777777	*
T1	1 00000020	0 26666677	11111101	37777777	*
TP	1 00000040	0 26666677	11111101	37777777	*
T6	1 00000100	0 26666677	11111101	37777777	*
T3	1 00000200	0 26666677	11111101	37777777	*
T0	1 00000400	0 26666677	11111101	37777777	*
T8	1 00001000	0 26666677	11111101	37777777	*
T5	1 00002000	0 26666677	11111101	37777777	*
T2	1 00004000	0 26666677	11111101	37777777	*
TR	1 00010000	0 26666677	11111101	37777777	*
T7	1 00020000	0 26666677	11111101	37777777	*
T4	1 00040000	0 26666677	11111101	37777777	*
T1	1 00100000	0 26666677	11111101	37777777	*
TP	1 00200000	0 26666677	11111101	37777777	*
T6	1 00400000	0 26666677	11111101	37777777	*
T3	1 01000000	0 26666677	11111101	37777777	*
T0	1 02000000	0 26666677	11111101	37777777	*
T8	1 04000000	0 26666677	11111101	37777777	*
T5	1 10000000	0 26666677	11111101	37777777	*
T2	1 20000000	0 77777770	00000010	37777777	*
TP	1 20000000	0 00000000	11111101	37777777	

FIGURE DIV44

A/M = 51437240 67122325 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 44542115	1 71160352	00000000	20100711	
T8	1 06617426	1 13134751	00000000	57677066	
T5	1 15437065	1 75060700	11110111	57677066	
T2	1 33076133	1 63151000	11110111	57677066	*
TR	1 66174266	0 10642133	00001000	20100711	
T7	1 54370555	1 70375245	11010111	57677066	
T4	1 30761333	1 51681713	11110111	57677066	*
T1	1 61742666	0 65724761	00000000	20100711	
TP	1 43705554	0 73751654	00001000	20100711	
T6	1 07613330	0 07025442	01001000	20100711	
T3	1 17426661	1 77644162	00110011	57677066	
T0	1 37055542	0 17030277	01001000	20100711	
T8	1 76133305	1 17651554	00110111	57677066	
T5	1 74266613	1 06532531	11110111	57677066	
T2	1 70555427	1 06374563	11010011	57677066	
TR	1 61333057	1 05600357	11110101	57677066	
T7	1 42666137	1 04410127	11110101	57677066	
T4	1 05554277	1 62037447	11100101	57677066	
T1	1 13330577	1 75066306	11110101	57677066	
TP	1 26661377	1 63164074	11110011	57677066	*
T6	1 55542776	0 10670123	00001000	20100711	
T3	1 33305775	1 70351225	11110111	57677066	
T0	1 66613773	1 52731652	10110111	57677066	*
T8	1 55427766	0 66203650	00001010	20100711	
T5	1 33057754	0 74511452	00001000	20100711	
T2	1 66137730	1 71225124	00000000	20100711	
TP	1 11640050	0 63164415	00001000	20100711	OF

FIGURE DIV45 A/M = 40241062 72603715 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 55440464	1 51761317	00000000	12021725	
T8	1 26016462	1 10315366	00000000	65756052	
T5	1 54035145	1 05500726	01110101	65756052	
T2	1 30072313	1 72370021	11010111	65756052	
TR	1 60164627	1 63756236	11000101	65756052	*
T7	1 40351456	0 03755514	00001110	12021725	
T4	1 00723135	1 64603423	11110101	65756052	
T1	1 01646272	0 05640175	00011000	12021725	
TP	1 03514565	1 70470344	11010101	65756052	
T6	1 07231353	1 60157164	11000001	65756052	*
T3	1 16462726	0 74357277	00001000	12021725	
TC	1 35145654	0 02761413	00001110	12021725	
T8	1 72313531	1 62613210	11110111	65756052	
T5	1 64627262	0 71667560	10001010	12021725	
T2	1 51456545	1 60437333	11100101	65756052	*
TR	1 23135312	0 75210716	00110100	12021725	
T7	1 46272624	0 64663751	00000010	12021725	
T4	1 14565451	1 76415715	01110101	65756052	
T1	1 31353122	0 01275751	10000010	12021725	
TP	1 62726245	1 07441714	01110101	65756052	
T6	1 45654513	1 06272005	01010101	65756052	
T3	1 13531227	1 73552267	11010001	65756052	
TC	1 27262457	1 76212532	01110101	65756052	*
T8	1 56545136	0 00666303	10001110	12021725	
T5	1 35312275	1 06425071	01110011	65756052	
T2	1 72624573	0 17272104	00000100	65756052	
TP	1 05153205	0 70242676	00000100	65756052	OF

FIGURE DIV46 A/M = 40000002 77464736 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 56763275	0 02710031	00000000	00727427	
T8	1 75067750	0 20065053	00000000	77050350	
T5	1 72157721	1 37122477	00100001	77050350	
T2	1 64337643	1 65415451	10100101	77050350	
TR	1 50677506	0 73152744	01010010	00727427	
T7	1 21577214	0 70264350	01011010	00727427	
T4	1 43376430	0 62411367	01111000	00727427	
T1	1 06775060	0 47163376	01011010	00727427	
TP	1 15772140	0 20217433	01101010	00727427	
T6	1 33764301	1 31611456	10100001	77050350	
T3	1 67750602	0 03542555	01010010	00727427	*
T0	1 57721405	1 00375623	10000101	77050350	
T8	1 37643013	1 17743121	00101101	77050350	
T5	1 77506027	1 27160714	10000101	77050350	
T2	1 77214057	1 65312303	10100101	77050350	
TR	1 76430136	0 73744430	00010010	00727427	
T7	1 75060274	0 67650520	01010010	00727427	
T4	1 72140570	0 61460600	01010110	00727427	
T1	1 64301360	0 45000240	01111010	00727427	
TP	1 50602740	0 14141140	01011010	00727427	
T6	1 21405701	1 21374671	10000001	77050350	
T3	1 43013603	1 51741034	10101101	77050350	
T0	1 06027406	0 44023612	01010110	00727427	
T8	1 14057014	0 12706263	00111010	00727427	
T5	1 30136031	1 15006036	10101101	77050350	
T2	1 60274063	0 52216276	00000000	77050350	
TP	1 17503715	0 52670641	00000000	77050350	0F

FIGURE DIV47

A/M = 77047436 24263720 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 00704547	0 35534246	00000000	41254457	
T8	0 35534245	0 42161226	00000000	41254457	
T5	0 73270512	0 32065774	11000001	36523320	
T2	0 66561224	0 34676213	10001101	36523320	*
TR	0 55342450	0 47211151	01111001	36523320	*
T7	0 32705120	0 56367645	01000001	36523320	
T4	0 65512240	0 64472034	11011001	36523320	
T1	0 53424500	0 21721313	10010101	36523320	
TP	0 27051201	1 24037400	01100110	41254457	
T6	0 56122403	1 13443677	00110000	41254457	
T3	0 34245007	1 60573146	10011110	41254457	
TC	0 70512016	0 17133054	01001001	36523320	
T8	0 61224035	1 01444502	00100110	41254457	
T5	0 42450073	1 44565074	00001010	41254457	
T2	0 05120166	0 47077531	01000001	36523320	
TR	0 12240354	0 56612504	00110101	36523320	
T7	0 24500730	0 63360732	11010001	36523320	
T4	0 51201660	0 26404106	01101101	36523320	
T1	0 22403541	1 10466066	10001010	41254457	
TP	0 45007303	1 02322543	00110110	41254457	
T6	0 12016607	1 46231176	00111010	41254457	
T3	0 24035416	0 42427734	11000001	36523320	
TC	0 50073034	0 55572112	10011101	36523320	
T8	0 20166070	0 01121747	11010001	36523320	
T5	0 40354161	1 65437377	00101000	41254457	
T2	0 00730342	1 53270777	00010000	36523320	
TP	0 77047436	0 24263720	11000101	36523320	

FIGURE DIV48 A/M = 60202004 17350247 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	0 16657605	1 72043476	00000000	41653276
T8	0 72043476	1 60533103	00000000	41653276
T5	0 64107174	0 77302700	00110011	36124501
T2	0 50216370	0 25141324	10011001	36124501
TR	0 20434761	1 33177041	01001110	41653276
T7	0 41071743	1 31153510	01100110	41653276
T4	0 02163707	1 25372637	01010100	41653276
T1	0 04347617	1 15650064	01011110	41653276
TP	0 10717437	1 76315566	01100100	41653276
T6	0 21637076	0 25157255	10001001	36124501
T3	0 43476175	1 33113722	01100110	41653276
TC	0 07174373	1 31273253	01010110	41653276
T8	0 16370767	1 25451134	01011110	41653276
T5	0 34761757	1 16717706	00100100	41653276
T2	0 71743737	1 77603202	00110110	41653276
TR	0 63707676	0 25742326	10011001	36124501
T7	0 47617575	1 34571045	01011110	41653276
T4	0 17437373	1 34257521	01000110	41653276
T1	0 37076767	1 33302650	01110110	41653276
TP	0 76175757	1 32670136	00011100	41653276
T6	0 74373737	1 26455663	01000110	41653276
T3	0 70767677	1 10706155	10101110	41653276
TC	0 61757577	1 03661741	00010110	41653276
T8	0 43737377	1 40337311	11100110	41653276
T5	0 07676776	0 51113544	10110001	36124501
T2	0 17575774	1 42447312	00000000	36124501
TP	0 60202004	0 17350247	00001001	36124501

FIGURE DIV49 A/M = 35222771 02103136 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 21002075	0 00010113	00000000	16152677	
T8	0 00010113	0 02627176	00000000	61625100	
T5	0 00020227	1 66273474	01010001	61625100	
T2	0 00040456	0 64751761	10010110	16152677	
TR	0 00101134	0 07016751	01100110	16152677	
T7	0 00202271	1 01063242	01010001	61625100	
T4	0 00404563	1 65763606	00010001	61625100	
T1	0 01011346	0 61031205	11111110	16152677	
TP	0 02022714	0 72456421	10001110	16152677	
T6	0 04045631	1 66754362	00010001	61625100	*
T3	0 10113462	0 63022535	11101110	16152677	
TC	0 20227144	0 06311301	00111110	16152677	
T8	0 40456311	1 76662122	00010001	61625100	
T5	0 01134622	0 03636036	10101110	16152677	
T2	0 02271445	1 00413414	11110001	61625100	
TR	0 04563113	1 04064132	01010001	61625100	
T7	0 11346227	1 73715366	00100001	61625100	
T4	0 22714466	0 76104545	10101110	16152677	
T1	0 45631135	1 75230632	01010001	61625100	*
TP	0 13462272	0 02653256	10001110	16152677	
T6	0 27144565	1 06355054	01001001	61625100	
T3	0 56311353	1 77551232	01010001	61625100	
TC	0 34622727	1 62167567	01000001	61625100	*
T8	0 71445656	0 54421157	10111100	16152677	
T5	0 63113534	0 67336326	00101110	16152677	
T2	0 46227270	1 56076075	01001000	16152677	
TP	0 46227270	0 05612335	10111110	16152677	OF

FIGURE DIV50 A/M = 63420721 13535047 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 11412441	1 75316651	00000000	47440035	
T8	0 75316651	1 61052476	00000000	47440035	
T5	0 72635522	0 72454037	00011101	30337742	
T2	0 65473244	0 15401143	00111101	30337742	
TR	0 53166511	1 72664546	10000000	47440035	
T7	0 26355223	1 44111342	11100010	47440035	*
T4	0 54732446	0 62751666	00011001	30337742	
T1	0 31665114	0 75274411	01011111	30337742	
TP	0 63552230	0 24142106	01011101	30337742	
T6	0 47324461	1 11766444	10000010	47440035	
T3	0 16651143	1 02315166	11100000	47440035	
TC	0 35522307	1 66472301	10000110	47440035	
T8	0 73244616	0 25414764	00110001	30337742	
T5	0 66511435	1 12611700	10100110	47440035	
T2	0 55223073	1 04264056	11000000	47440035	
TR	0 32446167	1 71110162	11100010	47440035	
T7	0 65114357	1 43060311	11000110	47440035	*
T4	0 52230736	0 60470705	00010101	30337742	
T1	0 24461674	0 71531757	00010001	30337742	
TP	0 51143570	0 13632602	00011101	30337742	
T6	0 22307361	1 66147634	11000010	47440035	
T3	0 44616742	0 26646452	00011001	30337742	
TC	0 11435705	1 14177154	11000010	47440035	
T8	0 23073613	1 11736305	10100100	47440035	
T5	0 46167427	1 03435047	10100000	47440035	
T2	0 14357057	0 27272117	00000000	47440035	
TP	0 63420721	0 13535047	00000000	47440035	

FIGURE DIV51 A/M = 76072766 74455603 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 77471243	0 15235633	00000000	06455137	
T8	1 62542146	0 71631375	00000000	71322640	
T5	1 45304314	0 61037022	11101110	06455137	
T2	1 12610630	0 62655314	10100110	06455137	
TR	1 25421460	0 63300107	11110100	06455137	
T7	1 53043140	0 77475545	00000010	06455137	
T4	1 26106300	0 75540462	10110010	06455137	
T1	1 54214601	1 64043724	01000101	71322640	*
TP	1 30431403	1 03422613	00010101	71322640	
T6	1 61063007	1 70307470	10101001	71322640	
T3	1 42146017	1 72343723	00000101	71322640	*
T0	1 04314036	0 62355100	11010110	06455137	
T8	1 10630074	0 64327557	11100000	06455137	
T5	1 21460170	0 71424465	10110010	06455137	
T2	1 43140361	1 74513732	00100101	71322640	*
TR	1 06300742	0 77075116	01010110	06455137	
T7	1 14601705	1 71534214	00001101	71322640	*
T4	1 31403612	0 61647761	10100010	06455137	
T1	1 63007424	0 61364011	11011110	06455137	
TP	1 46017050	0 62347372	11100010	06455137	
T6	1 14036120	0 65563134	10011010	06455137	
T3	1 30074240	0 71745447	10100000	06455137	
T0	1 60170501	1 75435756	00000001	71322640	*
T8	1 40361202	0 71440066	10111010	06455137	
T5	1 00742405	1 74644735	00000101	71322640	*
T2	1 01705012	1 71611074	00001000	06455137	*
TP	1 76072766	0 74455603	11110010	06455137	

FIGURE DIV52 A/M = 56002562 61556456 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62577376	0 14460725	00000000	27440716	
T8	1 63317054	0 65537463	00000000	50337061	
T5	1 46636130	0 72637055	10101010	27440716	
T2	1 15474261	1 55137134	01100101	50337061	*
TR	1 33170542	0 53036300	11101110	27440716	
T7	1 66361304	0 77637736	00100000	27440716	
T4	1 54742610	0 16237503	11101110	27440716	
T1	1 31705421	1 26130400	01110111	50337061	
TP	1 63613043	1 26030203	01010101	50337061	
T6	1 47426107	1 26437672	00000001	50337061	
T3	1 17054217	1 25326540	00110111	50337061	
T0	1 36130437	1 23324503	00110101	50337061	
T8	1 74261077	1 17320471	00110001	50337061	
T5	1 70542177	1 07310146	00110101	50337061	
T2	1 61304377	1 67377570	00000011	50337061	
TR	1 42610776	0 75337311	11101010	27440716	
T7	1 05421775	1 65337624	00100101	50337061	*
T4	1 13043772	0 72437560	10101010	27440716	
T1	1 26107765	1 55530341	00110101	50337061	*
TP	1 54217752	0 52141012	11001010	27440716	
T6	1 30437724	0 65744753	10000010	27440716	
T3	1 61077650	0 12351654	11001010	27440716	
T0	1 42177521	1 16254532	01010101	50337061	
T8	1 04377243	1 06000440	01110111	50337061	
T5	1 10776507	1 66550303	00010101	50337061	
T2	1 21775216	1 55340000	00001010	27440716	
TP	1 56002562	0 61556456	11101010	27440716	

FIGURE DIV53 A/M = 04363677 30736776 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 03423023	0 22540776	00000000	31222477	
T8	0 22540776	0 52200324	00000000	46555300	
T5	0 45301774	0 55622247	00001100	31222477	
T2	0 12603770	0 64660306	00011110	31222477	
TR	0 25407760	0 02705423	00100110	31222477	
T7	0 53017741	1 44560566	10010001	46555300	
T4	0 26037702	0 62503746	00100110	31222477	
T1	0 54077604	0 76622523	00010110	31222477	
TP	0 30177410	0 26607056	00101110	31222477	
T6	0 60377021	1 13375654	11000001	46555300	
T3	0 40776043	1 16440033	01111001	46555300	
T0	0 01774107	1 75077371	11000001	46555300	
T8	0 03270217	1 52643264	10111001	46555300	*
T5	0 07760436	0 76152141	01001110	31222477	
T2	0 17741074	0 27540111	00011110	31222477	
TR	0 37702171	1 15077742	11000001	46555300	
T7	0 77604363	1 12644206	10111001	46555300	
T4	0 77410747	1 03407717	11100001	46555300	
T1	0 77021717	1 77764141	00011001	46555300	
TP	0 76043636	0 20114774	11100010	31222477	
T6	0 74107475	1 20106311	11101001	46555300	
T3	0 70217173	1 20163125	11011001	46555300	
T0	0 60436367	1 20045555	11100001	46555300	
T8	0 41074757	1 20060635	11010001	46555300	
T5	0 02171737	1 20686775	10100001	46555300	
T2	0 04363677	0 61675774	00000000	46555300	
TP	0 04363677	0 30736776	00000000	46555300	

FIGURE DIV54

A/M = 74520707 47016161 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER
T8	1 74576514	1 52404733	00000000	37111644
T8	1 25373046	1 44067417	00000000	40666133
T5	1 52766114	0 47260672	00010010	37111644
T2	1 25754230	0 45672441	10001010	37111644
TR	1 53730460	0 62670766	10010000	37111644
T7	1 27661140	0 14712511	10001110	37111644
T4	1 57542301	1 11405575	01110001	40666133
T1	1 37304603	1 65011420	01110111	40666133
TP	1 76611406	0 03354026	10001100	37111644
T6	1 75423015	1 66510300	01110111	40666133
T3	1 73046032	0 06552667	10000000	37111644
T0	1 66114065	1 75103602	01110111	40666133
T8	1 54230152	0 23530463	10011010	37111644
T5	1 30460325	1 27161212	01010111	40666133
T2	1 61140653	1 20140771	01110011	40666133
TR	1 42301527	1 02307130	01101011	40666133
T7	1 04603257	1 47606436	00100001	40666133
T4	1 11406536	0 46726632	10000110	37111644
T1	1 23015274	0 64766540	10001010	37111644
TP	1 46032570	0 20060164	11011000	37111644
T6	1 14065361	1 22740404	00110101	40666133
T3	1 30152743	1 06707345	00100001	40666133
T0	1 60325707	1 56674047	00011001	40666133
T8	1 40653616	0 64723765	10000000	37111644
T5	1 01527434	0 20750517	10011100	37111644
T2	1 03257071	0 61743436	00000000	40666133
TP	1 74520707	0 47016161	00000000	40666133

FIGURE DIV55 A/M = 41676363 31043510 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 33130202	0 14073355	00000000	43107232	
T8	0 14073355	0 76237434	00000000	43107232	
T5	0 30166732	0 20267535	11100101	34670545	
T2	0 60355665	1 25056626	01010100	43107232	
TR	0 40733553	1 17254007	00011100	43107232	
T7	0 01667327	1 71651441	10010010	43107232	
T4	0 03556657	1 46642354	00010000	43107232	*
T1	0 07335536	0 41314475	11101001	34670545	
TP	0 16673274	0 51623631	10100111	34670545	
T6	0 35566570	0 77437441	01101011	34670545	
T3	0 73355361	1 44300356	00110000	43107232	*
T0	0 66732742	0 45610472	00101011	34670545	
T8	0 55665704	0 47413744	01100011	34670545	
T5	0 33553610	0 55017470	01101011	34670545	
T2	0 67327420	0 60031747	11100001	34670545	
TR	0 56657040	0 16053466	01101001	34670545	
T7	0 35536101	1 71430301	10010110	43107232	
T4	0 73274203	1 46200254	00010000	43107232	*
T1	0 66570406	0 40210276	11101001	34670545	
TP	0 55361014	0 56412234	01101111	34670545	
T6	0 32742030	0 63016450	11101011	34670545	
T3	0 65704060	0 24027607	01100101	34670545	
TC	0 53610141	1 15356053	00011000	43107232	
T8	0 27420303	1 75055351	01010010	43107232	
T5	0 57040606	0 20742407	10101101	34670545	
T2	0 36101415	0 61107211	01000010	43107232	
TP	0 41676363	0 31043510	01000010	43107232	

FIGURE DIV56 A/M = 72014004 13437770 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 04417370	1 71706720	00000000	47574674	
T8	0 71706720	1 54214264	00000000	47574674	
T5	0 63615640	0 60633654	00000001	30203103	
T2	0 47433500	0 71662636	00010001	30203103	
TR	0 17067200	0 13760572	00010011	30203103	
T7	0 36166401	1 66455272	11101010	47574674	
T4	0 74335002	0 27537707	00000001	30203103	
T1	0 70672005	1 16763415	10111100	47574674	
TP	0 61564013	1 15665027	10101100	47574674	
T6	0 43350027	1 12440043	11111110	47574674	
T3	0 06720057	1 06016113	11101110	47574674	
T0	0 15640137	1 75722232	10111110	47574674	
T8	0 33500277	1 53552570	10111010	47574674	*
T5	0 67200576	0 77742503	00010001	30203103	
T2	0 56401374	0 27120314	01010001	30203103	
TR	0 35002771	1 27754427	00101100	47574674	
T7	0 72005763	1 17627042	10101110	47574674	
T4	0 64013747	1 16354111	11101110	47574674	
T1	0 50027717	1 16626237	10101100	47574674	
TP	0 20057637	1 14342463	11111110	47574674	
T6	0 40137477	1 12613152	10111110	47574674	
T3	0 00277177	1 04344331	11101110	47574674	
T0	0 00576377	1 72607776	10100000	47574674	
T8	0 01374777	1 44303560	11111110	47574674	*
T5	0 02771776	0 63234663	00000001	30203103	
T2	0 05763774	1 46471540	00000010	30203103	
TP	0 72014004	0 13437770	00000001	30203103	

FIGURE DIVB7 A/M = 23676546 10403014 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 12257535	1 23604361	00000000	20517362	
T8	0 73604361	1 21540153	00000000	57260415	
T5	0 67410742	0 03017601	01000110	20517362	
T2	0 57021705	1 67217230	00101011	57260415	
TR	0 36043612	0 76357065	01001000	20517362	
T7	0 74107424	0 16457434	01000100	20517362	
T4	0 70217051	1 06317606	10100101	57260415	
T1	0 60435123	1 04217224	10101011	57260415	
TP	0 41074247	1 07020000	01101111	57260415	
T6	0 02170517	1 77422630	00100011	57260415	
T3	0 04361236	0 17754064	00011000	20517362	
T0	0 10742475	1 06132565	11100001	57260415	
T8	0 21705173	1 05645761	10100011	57260415	
T5	0 43612367	1 02173371	11001011	57260415	
T2	0 07424757	1 75550412	10101011	57260415	
TR	0 17051737	1 62703453	10100011	57260415	*
T7	0 36123676	0 06516432	00010100	20517362	
T4	0 74247575	1 64435671	10100011	57260415	
T1	0 70517372	0 11702167	00111000	20517362	
TP	0 61236765	1 73206764	10100011	57260415	
T6	0 42475753	1 55075300	11001111	57260415	*
T3	0 05173726	0 74604305	00110100	20517362	
TC	0 12367654	0 11340374	01010000	20517362	
T8	0 24757531	1 73100305	11101101	57260415	
T5	0 51737263	1 67663421	00000011	57260415	*
T2	0 23676546	1 57547065	00000000	20517362	*
TP	0 23676546	0 10403014	01010100	20517362	

TIME	A REGISTER	B REGISTER	FIGURE DIV58 BC REGISTER	C REGISTER	A/M = 26077601 60157300 1
T8	1 42262332	0 30122704	00000000	54202442	
T8	1 47655074	0 11720107	00000000	54202442	
T5	1 17532171	1 27042651	01000010	54202442	
T2	1 37264363	1 54307104	00001100	54202442	*
TR	1 76550746	0 53315745	01100001	23575335	
T7	1 75321714	0 54520242	00111011	23575335	
T4	1 72643630	0 54157054	01101011	23575335	
T1	1 65507460	0 55235400	01100111	23575335	
TP	1 53217140	0 57360550	01110011	23575335	
T6	1 26436300	0 64656670	00100011	23575335	
T3	1 55074600	0 74422137	01111001	23575335	
T0	1 32171400	0 16753536	00110101	23575335	
T8	1 64363001	1 02341030	10011110	54202442	
T5	1 50746003	1 20126743	11000000	54202442	
T2	1 21714007	1 56457341	00001010	54202442	*
TR	1 43630016	0 50635257	10101001	23575335	
T7	1 07460034	0 64361076	01111001	23575335	
T4	1 17140070	0 26651423	00110111	23575335	
T1	1 36300160	0 10430525	11110101	23575335	
TP	1 74600341	1 17503016	00001100	54202442	
T6	1 71400703	1 03412677	10000000	54202442	
T3	1 63001607	1 73227131	10001110	54202442	
T0	1 46003417	1 52662145	10001000	54202442	*
T8	1 14007036	0 60233640	11110011	23575335	
T5	1 30016074	0 05474057	01011001	23575335	
T2	1 60034171	0 15112130	00100010	54202442	
TP	1 60034171	0 71272720	00100010	54202442	OF

FIGURE DIV59 A/M = 40000215 77473026 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 66562553	1 54177240	00000000	03070242	
T8	1 23600540	1 06124762	00000000	74707535	
T5	1 47401301	1 00150471	11011011	74707535	
T2	1 17002603	1 16242732	01010011	74707535	
TR	1 36005407	1 22424433	11011011	74707535	
T7	1 74013017	1 53772645	10010001	74707535	
T4	1 70026036	0 42075757	11000000	03070242	
T1	1 60054074	0 21163171	10101010	03070242	
TP	1 40130171	1 56457130	01001011	74707535	
T6	1 00260362	0 32130545	10100000	03070242	
T3	1 00540744	0 07451554	00100000	03070242	*
T0	1 01301711	1 03222065	11011001	74707535	
T8	1 02603623	1 24365601	01010111	74707535	
T5	1 05407447	1 46672351	01011011	74707535	
T2	1 13017116	0 12676106	10001100	03070242	
TR	1 26036235	1 31475141	11011011	74707535	
T7	1 54074472	0 00204566	10100000	03070242	*
T4	1 30171165	1 14610002	01011111	74707535	
T1	1 60362353	1 27141753	01010011	74707535	
TP	1 40744727	1 54222406	01011101	74707535	
T6	1 01711656	0 35457451	00100010	03070242	
T3	1 03623535	1 67236677	01010001	74707535	
T0	1 07447272	0 63505742	00100100	03070242	
T8	1 17116564	0 42404346	10100000	03070242	
T5	1 36235350	0 20200156	10101000	03070242	
T2	1 74472721	0 60602334	00000000	74707535	
TP	1 03305057	0 47476622	00000000	74707535	OF

FIGURE DIV60 A/M = 74660235 21636425 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 02377223	1 45265627	00000000	46523256	
T8	0 45265627	1 51122501	00000000	46523256	
T5	0 12553456	0 53411724	00110001	31254521	
T2	0 25327134	0 60417373	10101001	31254521	
TR	0 52656270	0 72414411	00101101	31254521	
T7	0 25534560	0 16407746	00100001	31254521	
T4	0 53271341	1 73732164	10011010	46523256	
T1	0 26562703	1 45421547	11010100	46523256	*
TP	0 55346606	0 66237237	00101001	31254521	
T6	0 32713414	0 05054122	01101101	31254521	
T3	0 65627031	1 52055614	11000110	46523256	
TC	0 53456062	0 77307372	00101001	31254521	
T8	0 27134144	0 20274400	10001111	31254521	
T5	0 56270311	1 26216470	01100010	46523256	
T2	0 34560623	1 14350457	11010000	46523256	
TR	0 71341447	1 10464304	11000110	46523256	
T7	0 62703117	1 01674207	10000100	46523256	
T4	0 45606237	1 61213065	11101010	46523256	
T1	0 13414476	0 15004614	01100101	31254521	
TP	0 27031175	1 72734000	10001110	46523256	
T6	0 56062373	1 43315476	11100000	46523256	*
T3	0 34144766	0 62207616	00100101	31254521	
TC	0 70311754	0 75063357	01011001	31254521	
T8	0 60623730	0 25344462	00101001	31254521	
T5	0 41447661	1 11636315	10000110	46523256	
T2	0 03117543	0 43474063	00001000	46523256	
TP	0 74660235	0 21636425	00000000	46523256	

FIGURE DIV61 A/M = 36063112 21326047 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 32272650	1 71430106	00000000	31716246	
T8	0 71430106	1 00354402	00000000	46061531	
T5	0 63060215	1 46712536	00100001	46061531	
T2	0 46140432	0 47733435	00010110	31716246	
TR	0 14301064	0 40615551	11010010	31716246	
T7	0 30602150	0 54361500	01010110	31716246	
T4	0 61404320	0 63671666	01010000	31716246	
T1	0 43010640	0 02511713	01010110	31716246	
TP	0 06021501	1 45225570	10100011	46061531	
T6	0 14043202	0 63561640	01010010	31716246	
T3	0 30106404	0 02271766	01010000	31716246	
T0	0 60215011	1 44664405	10001101	46061531	
T8	0 40432022	0 62461453	01010010	31716246	
T5	0 01064044	0 70071305	11010110	31716246	
T2	0 02150111	1 40264563	10000001	46061531	*
TR	0 04320222	0 51467516	01000100	31716246	
T7	0 10640444	0 56065672	01010010	31716246	
T4	0 21501110	0 67001042	01111010	31716246	
T1	0 43202220	0 11132362	01010010	31716246	
TP	0 06404441	1 62365416	10001101	46061531	
T6	0 15011102	0 15663474	01010010	31716246	
T3	0 32022205	1 73540741	10110001	46061531	
T0	0 64044413	1 45502435	10101001	46061531	*
T8	0 50111026	0 64315233	01010110	31716246	
T5	0 20222054	0 03560145	01011010	31716246	
T2	0 40444131	0 01362332	10000000	46061531	
TP	0 40444131	0 04571155	00000000	46061531	OF

FIGURE DIV62 A/M = 24741103 63724254 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 45641207	1 70677650	00000000	60336670	
T8	1 07100130	1 12475460	00000000	60336670	
T5	1 16200261	1 05421730	00110100	60336670	
T2	1 34400543	1 73512650	00110100	60336670	
TR	1 71001306	0 76006027	11101001	17441107	
T7	1 62002615	1 76544651	00010100	60336670	*
T4	1 44005432	0 04772022	10001011	17441107	
T1	1 10013065	1 11214657	01110100	60336670	
TP	1 20026153	1 04107526	01101100	60336670	
T6	1 40054327	1 72757244	00001100	60336670	
T3	1 00130656	0 74301010	11101011	17441107	
TC	1 00261535	1 73332632	00010100	60336670	*
T8	1 00543272	0 75346763	11000011	17441107	
T5	1 01306565	1 74243560	01011100	60336670	*
T2	1 02615352	0 01172647	11000001	17441107	
TR	1 05432725	1 04613310	00111100	60336670	
T7	1 13065653	1 71107610	01100100	60336670	
T4	1 26153526	0 73050727	11010001	17441107	
T1	1 54327255	1 70477450	00001100	60336670	*
TP	1 30656532	0 70542420	10100011	17441107	
T6	1 61535265	1 61033652	01010100	60336670	*
T3	1 43272552	0 53540024	10011011	17441107	
TC	1 06565324	0 56763172	10000011	17441107	
T8	1 15352650	0 64307505	11100011	17441107	
T5	1 32725520	0 02450333	10010011	17441107	
T2	1 65653241	0 25140600	00000110	60336670	
TP	1 65653241	0 65317434	00000000	60336670	0F

FIGURE DIV63 A/M = 40000001 74520140 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 62311602	1 23170133	00000000	00753715	
T8	1 04607646	1 14512260	00000000	77024062	
T5	1 11417515	1 20240522	10010101	77024062	
T2	1 23037233	1 57545420	00000111	77024062	
TR	1 46076466	0 37266177	01001000	00753715	
T7	1 14175164	0 00421204	01111110	00753715	*
T4	1 30372331	1 72200612	10110101	77024062	
T1	1 60764663	1 23645600	10000111	77024062	*
TP	1 41751547	1 76537605	10000101	77024062	*
T6	1 03723317	1 04213677	10110001	77024062	*
T3	1 07646637	1 27663562	00010101	77024062	
T0	1 17[illegible]15477	1 46503520	10110111	77024062	
T8	1 37233176	0 35372377	01011000	00753715	
T5	1 75466375	1 63022750	11010111	77024062	
T2	1 75154772	0 70731050	00111010	00753715	
TR	1 72331764	0 62057056	01001000	00753715	
T7	1 64663750	0 46013042	01101010	00753715	
T4	1 51547720	0 16103032	01101010	00753715	
T1	1 23317641	1 25434167	10000001	77024062	
TP	1 46637503	1 62014342	10100101	77024062	
T6	1 15477206	0 64104014	01101010	00753715	
T3	1 33176414	0 52365765	01000000	00753715	
TC	1 66375030	0 26626667	01101000	00753715	
T8	1 54772061	1 46673531	10010111	77024062	
T5	1 31764142	0 35552412	01011010	00753715	
T2	1 63750305	0 75347044	00000000	77024062	
TP	1 14027473	0 41214356	00000000	77024062	OF

FIGURE DIV64 A/M = 12231200 64527135 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 67745772	0 01721272	00000000	46770716	
T8	1 76056506	0 57022723	00000000	46770716	
T5	1 74135214	0 67044620	00010111	31007061	
T2	1 70272430	0 07130644	00010101	31007061	
TR	1 60565061	1 54271621	11001010	46770716	
T7	1 41352142	0 03564444	00010101	31007061	
T4	1 02724305	1 45261221	11101010	46770716	
T1	1 05650612	0 65743443	00010101	31007061	
TP	1 13521424	0 04726371	00010001	31007061	
T6	1 27243051	1 57564672	01101010	46770716	
T3	1 56506122	0 72552665	00010001	31007061	
T0	1 35214244	0 16344536	00010101	31007061	
T8	1 72430511	1 72621304	11101110	46770716	
T5	1 65061223	1 45535747	11100000	46770716	*
T2	1 52142446	0 66472770	00010011	31007061	
TR	1 24305114	0 06204064	00011001	31007061	
T7	1 50612231	1 52322060	11101010	46770716	
T4	1 21424462	0 70045142	10010101	31007061	
T1	1 43051145	1 46022414	01101010	46770716	*
TP	1 06122312	0 47246032	00010101	31007061	
T6	1 14244624	0 47533247	00010101	31007061	
T3	1 30511450	0 40305701	10010101	31007061	
TC	1 61223120	0 51633065	00010001	31007061	
T8	1 42446240	0 54505136	00010101	31007061	
T5	1 05114500	0 62231450	00010111	31007061	
T2	1 12231200	1 44403342	00100000	31007061	
TP	1 12231200	0 64527135	00010101	31007061	

FIGURE DIV65 A/M = 40000001 64375304 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 50253200	1 53333102	00000000	01160135	
T8	1 24444676	1 26344442	00000000	76617642	
T5	1 51111575	1 42620746	11010001	76617642	
T2	1 22223372	0 20341044	10101010	01160135	
TR	1 44446765	1 57713772	00010001	76617642	
T7	1 11115752	0 30727014	10101110	01160135	
T4	1 22233724	0 03140305	00100100	01160135	*
T1	1 44467651	1 74310654	11010001	76617642	
TP	1 11157523	1 00460375	11011001	76617642	*
T6	1 22337247	1 11751536	10011101	76617642	
T3	1 44676517	1 31654230	11011111	76617642	
T0	1 11575236	0 06432030	00101010	01160135	*
T8	1 23372475	1 03075742	11010001	76617642	
T5	1 46765173	1 26022540	01011011	76617642	
T2	1 15752367	1 44676165	10011001	76617642	
TR	1 33724756	0 32776401	00000110	01160135	
T7	1 67651735	1 53504064	11111001	76617642	
T4	1 57523672	0 52512200	00100110	01160135	
T1	1 37247564	0 26504756	00100000	01160135	
TP	1 75517351	1 43220576	11011001	76617642	
T6	1 75236722	0 31543424	00100110	01160135	
T3	1 72475645	1 51316033	11011101	76617642	
T0	1 65173512	0 45736326	00100100	01160135	
T8	1 52367224	0 15155102	00100110	01160135	
T5	1 24756451	1 20341267	11011001	76617642	
T2	1 51735123	0 62724550	00000010	76617642	
TP	1 26042655	0 46425510	00000010	76617642	OF

FIGURE DIV66 A/M = 76212415 00414240 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 00312075	0 33376625	00000000	70544122	
T8	0 33376625	0 71056217	00000000	70544122	
T5	0 66775452	0 61367203	10001111	07233655	
T2	0 55773124	0 61113406	11101101	07233655	
TR	0 33766250	0 63664064	10001011	07233655	
T7	0 67754520	0 66705747	10100101	07233655	
T4	0 57731240	0 75247776	10000001	07233655	
T1	0 37662501	1 02253011	01011110	70544122	*
TP	0 77546203	1 76214364	01100000	70544122	
T6	0 77312406	0 75063526	11001101	07233655	
T3	0 76625015	1 04615571	00100010	70544122	*
T0	0 75452033	1 01377525	01000000	70544122	
T8	0 73124067	1 74433375	01110000	70544122	
T5	0 66250156	0 72531540	10011111	07233655	
T2	0 54520335	1 75041645	01010000	70544122	*
TR	0 31240672	0 73356360	10001011	07233655	
T7	0 62501565	1 76472004	01011100	70544122	*
T4	0 45203352	0 76431066	10011001	07233655	
T1	0 12406725	1 05640271	00010010	70544122	*
TP	0 25015653	1 03264724	01000000	70544122	
T6	0 52033527	1 00215772	01100000	70544122	
T3	0 24067257	1 72377007	01001100	70544122	
T0	0 50156536	0 65133063	11101011	07233655	
T8	0 20335274	0 73623746	10100101	07233655	
T5	0 40672571	1 77304230	01110010	70544122	*
T2	0 01565363	1 70030401	11000100	70544122	*
TP	0 76212415	1 00414240	10000000	70544122	

FIGURE DIV67 A/M = 37716231 06621435 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 34230727	0 24317475	00000000	14254051	
T8	0 24317475	0 17754656	00000000	63523726	
T5	0 50637173	1 12454452	11001011	63523726	
T2	0 21476367	1 22656065	10001011	63523726	
TR	0 43174757	1 40251012	11011111	63523726	
T7	0 06371736	0 36710210	00110110	14254051	
T4	0 14763675	1 51563566	10001001	63523726	
T1	0 31747572	0 57525327	00100100	14254051	
TP	0 63717364	0 43627027	10100100	14254051	
T6	0 47636750	0 33022220	11110110	14254051	
T3	0 17475720	0 24430632	00110100	14254051	*
T0	0 37173641	1 34024502	01001111	63523726	
T8	0 76367503	1 55576344	00001011	63523726	
T5	0 74757206	0 47542704	00110100	14254051	
T2	0 71736414	0 23772062	10010000	14254051	
TR	0 63575031	1 42527063	11001011	63523726	
T7	0 47572062	0 33434142	10100100	14254051	
T4	0 17364144	0 13444556	10100000	14254051	*
T1	0 36750311	1 22034252	11001011	63523726	
TP	0 75720623	1 41515464	10101011	63523726	
T6	0 73641446	0 37601144	00110100	14254051	
T3	0 67503115	1 52345437	11001001	63523726	
T0	0 57206232	0 52161052	11010100	14254051	
T8	0 36414464	0 52536376	10100000	14254051	
T5	0 75031150	0 51640745	10110100	14254051	
T2	0 72062320	1 43721113	00001000	14254051	
TP	0 72062320	0 76225116	01100100	14254051	OF

FIGURE DIV68 A/M = 11565411 20731660 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 06662624	0 32035730	00000000	26415534	
T8	0 32035730	0 60245020	00000000	51362243	
T5	0 64073660	0 66127614	01000100	26415534	
T2	0 50167540	0 74664365	10011000	26415534	
TR	0 20357301	1 62054116	01101101	51362243	*
T7	0 40736602	0 64747162	10001010	26415534	
T4	0 01675404	0 17335011	01001110	26415534	
T1	0 03573011	1 11156405	01100101	51362243	
TP	0 07366023	1 75017457	01100001	51362243	
T6	0 16754046	0 12644674	10010000	26415534	
T3	0 35730115	1 16013733	01100101	51362243	
T0	0 73660233	1 07502233	00110101	51362243	
T8	0 67540467	1 60706034	10101101	51362243	
T5	0 57301156	0 00423727	10010100	26415534	
T2	0 36602335	1 72342212	00110111	51362243	
TR	0 75404672	0 03531372	10011010	26415534	
T7	0 73011565	1 70666240	10001011	51362243	
T4	0 66023353	1 52030766	01110001	51362243	*
T1	0 54046726	0 44716403	10001110	26415534	
TP	0 30115654	0 57244753	01010010	26415534	
T6	0 60233530	0 66146472	01001010	26415534	
T3	0 40467260	0 74724731	10010010	26415534	
T0	0 01156541	1 62153146	01101001	51362243	*
T8	0 02335302	0 64145042	11001010	26415534	
T5	0 04672604	0 10721650	10010010	26415534	
T2	0 11565411	0 41663540	00000000	51362243	
TP	0 11565411	0 20731660	00000000	51362243	

FIGURE DIV69 A/M = 75175170 05337450 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 02425672	0 02176161	00000000	42360576	
T8	0 02176161	0 45006470	00000000	42360576	
T5	0 04374342	0 47424361	00010001	35417201	
T2	0 10770704	0 44407145	10101001	35417201	
TR	0 21761610	0 66637515	00000001	35417201	
T7	0 43743420	0 02006435	11110001	35417201	
T4	0 07707041	1 60515563	10100110	42360576	
T1	0 17616102	0 16042767	01010001	35417201	
TP	0 37434205	1 70405456	10101100	42360576	
T6	0 77070413	1 43575022	00001110	42360576	*
T3	0 76161026	0 34703466	10110001	35417201	
T0	0 74342054	0 47436350	00010011	35417201	*
T8	0 70204130	0 44523144	10011001	35417201	
T5	0 61610260	0 66607514	00100001	35417201	
T2	0 43420540	0 02226434	11010001	35417201	
TR	0 07041301	1 60755561	10100110	42360576	
T7	0 16102602	0 17542763	00010001	35417201	
T4	0 34205405	1 71605446	10101100	42360576	
T1	0 70413013	1 45175002	01001110	42360576	*
TP	0 61026026	0 41703426	10110001	35417201	
T6	0 42054054	0 51436250	10010011	35417201	
T3	0 04130130	0 70523744	10010001	35417201	
TC	0 10260261	1 43547400	00101110	42360576	*
T8	0 20540542	0 34130421	11010001	35417201	
T5	0 41301304	0 47710245	00010001	35417201	*
T2	0 02602610	1 17640515	00000000	35417201	*
TP	0 75175170	0 05337450	11000001	35417201	

FIGURE DIV70 A/M = 15626505 12777521 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 11631731	0 22222044	00000000	26570515	
T8	0 22222044	0 63041214	00000000	51207262	
T5	0 44444110	0 64672145	10001000	26570515	
T2	0 11110220	0 17256020	01101010	26570515	
TR	0 22220441	1 11135242	01010101	51207262	
T7	0 44441103	1 75511160	00011011	51207262	
T4	0 11102206	0 11734070	10101010	26570515	
T1	0 22204415	1 15271462	00010001	51207262	
TP	0 44411033	1 03002320	01010111	51207262	
T6	0 11022067	1 61224245	00010101	51207262	
T3	0 22044156	0 60160421	11101010	26570515	
T0	0 44110334	0 00233577	11100000	26570515	
T8	0 10220671	1 73066551	01010111	51207262	
T5	0 20441562	0 16665251	00101010	26570515	
T2	0 41103345	1 06153724	01010101	51207262	
TR	0 02206713	1 67547235	00010101	51207262	
T7	0 04415626	0 75026301	11101110	26570515	
T4	0 11033455	1 65456204	00010101	51207262	*
T1	0 22067132	0 71644327	10101000	26570515	
TP	0 44156265	1 54111030	01011111	51207262	*
T6	0 10334552	0 50734010	10101010	26570515	
T3	0 20671324	0 60562555	10100000	26570515	
T0	0 41562650	0 07235047	01101000	26570515	
T8	0 03345521	1 71073371	01010011	51207262	
T5	0 06713242	0 02776511	10001010	26570515	
T2	0 15626505	0 25777242	00000000	51207262	
TP	0 15626505	0 12777521	00000000	51207262	

FIGURE DIV71 A/M = 32021413 77402621 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 67352527	1 55464166	00000000	77253076	
T8	1 22313612	1 07700346	00000000	77253076	
T5	1 44627425	1 06053702	11000110	77253076	
T2	1 11457058	1 25373013	10010110	77253076	
TR	1 23136127	1 61251234	11010110	77253076	
T7	1 46274256	0 64266611	01001001	00524701	
T4	1 14570534	0 52273326	01011001	00524701	
T1	1 31361270	0 26334557	01001001	00524701	
TP	1 62742561	1 45046336	11100100	77253076	
T6	1 45705342	0 34031776	01010001	00524701	
T3	1 13612705	1 61356765	10000110	77253076	
T0	1 27425612	0 62452073	01011001	00524701	
T8	1 57053424	0 47662071	00011001	00524701	
T5	1 36127050	0 17202066	01111001	00524701	
T2	1 74256121	1 37071144	01010110	77253076	
TR	1 70534242	0 00726432	00001001	00524701	*
T7	1 61270505	1 70022055	11110110	77253076	
T4	1 42561213	1 71487341	10100110	77253076	*
T1	1 05342427	1 72442111	10110110	77253076	*
TP	1 12705057	1 74577430	10000110	77253076	*
T6	1 25612137	1 00542376	10110000	77253076	*
T3	1 53424277	1 10777762	10000110	77253076	
T0	1 27050577	1 30143153	11110110	77253076	
T8	1 56121376	0 02242447	01011001	00524701	*
T5	1 34242775	1 75772117	10010100	77253076	
T2	1 70505773	0 13704436	00100000	77253076	
TP	1 70505773	0 71775561	00100000	77253076	0F

FIGURE DIV72 A/M = 77421646 01563506 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 00156137	1 76543640	00000000	61144207	
T8	0 76543640	1 61322346	00000000	61144207	
T5	0 75307500	0 50477405	11001101	16633570	
T2	0 72617200	0 51734705	10100101	16633570	
TR	0 65436400	0 52625505	10100101	16633570	
T7	0 53075000	0 53406105	11101101	16633570	
T4	0 26172000	0 57041105	01011101	16633570	
T1	0 54364000	0 76757104	00001101	16633570	
TP	0 30750000	0 03573103	11001101	16633570	
T6	0 61720001	1 12434617	00100000	61144207	
T3	0 43640003	1 06335636	00100010	61144207	
T0	0 07500007	1 76137714	00100010	61144207	
T8	0 17200016	0 02222340	11111101	16633570	
T5	0 36400035	1 07723301	00110010	61144207	
T2	0 75000073	1 71122021	10111010	61144207	
TR	0 72000167	1 63522262	00110010	61144207	*
T7	0 64000356	0 55217255	11101101	16633570	
T4	0 50000734	0 62473425	11001101	16633570	
T1	0 20001670	0 75724745	10100101	16633570	
TP	0 40003561	1 75116313	00100010	61144207	*
T6	0 00007342	0 00267337	11001101	16633570	
T3	0 00016705	1 03625207	00100100	61144207	
TC	0 00035613	1 60716015	10101010	61144207	
T8	0 00073426	0 70661542	10010101	16633570	
T5	0 00167055	1 62627705	00100010	61144207	*
T2	0 00356132	1 45657632	00000000	16633570	*
TP	0 77421646	0 01563506	11011101	16633570	

FIGURE DIV73 A/M = 36762602 74640044 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 56100526	1 41641171	00000000	55427614	
T8	1 36136610	1 77321065	00000000	55427614	
T5	1 74275420	0 10112235	11100101	22350163	
T2	1 70573041	1 17043401	01011110	55427614	
TR	1 61366103	1 15550037	00011000	55427614	
T7	1 42754207	1 00761603	10010110	55427614	
T4	1 05730417	1 76302433	01111010	55427614	
T1	1 13661036	0 11377261	10000011	22350163	
TP	1 27542075	1 17315370	01111010	55427614	
T6	1 57304173	1 16473614	00011000	55427614	
T3	1 36610367	1 02630245	10011000	55427614	
T0	1 75420757	1 72121326	11011000	55427614	
T8	1 73041737	1 53603461	10111010	55427614	*
T5	1 66103676	0 61101246	11100101	22350163	
T2	1 54207574	0 06752072	00001011	22350163	
TR	1 30417371	1 62345013	11011010	55427614	
T7	1 61036762	0 00204121	11100111	22350163	
T4	1 42075745	1 77237271	01001010	55427614	
T1	1 04173713	1 55027417	01101000	55427614	*
TP	1 10367626	0 56531111	00100111	22350163	
T6	1 20757454	0 57732527	00100101	22350163	
T3	1 41737130	0 52335543	10100101	22350163	
TC	1 03576260	0 67343674	00100001	22350163	
T8	1 07574540	0 71457655	10000101	22350163	
T5	1 17371301	1 50556540	10011010	55427614	*
T2	1 36762602	1 41357320	00000000	22350163	*
TP	1 36762602	0 74640044	01100101	22350163	

FIGURE DIV74 A/M = 74170301 57266103 0

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	1 74711330	0 12011202	00000000	32317176	
T8	1 65766576	0 50547251	00000000	45460601	
T5	1 53755374	0 53625611	00010110	32317176	
T2	1 27732770	0 51702731	10110110	32317176	
TR	1 57665760	0 76335270	00010010	32317176	
T7	1 37553740	0 26221777	01010000	32317176	
T4	1 77327701	1 13043577	11101001	45460601	
T1	1 76652603	1 15771102	00001101	45460601	
TP	1 75537407	1 70344200	11101011	45460601	
T6	1 73277017	1 40572224	10001001	45460601	*
T3	1 66576036	0 53675541	00010110	32317176	
T0	1 55374074	0 51122611	11010110	32317176	
T8	1 32770170	0 26574731	00010110	32317176	
T5	1 65760360	0 27721270	00010010	32317176	
T2	1 53740741	1 14242402	11101001	45460601	
TR	1 27701703	1 17367600	01000011	45460601	
T7	1 57603607	1 05337223	01101001	45460601	
T4	1 37407417	1 52460252	10101001	45460601	
T1	1 77017036	0 77651614	00010110	32317176	
TP	1 76036074	0 21052747	11010100	32317176	
T6	1 74074171	1 21526710	10100011	45460601	
T3	1 70170363	1 20035444	11101001	45460601	
T0	1 60360747	1 27655714	00100001	45460601	
T8	1 40741717	1 14313434	11101001	45460601	
T5	1 01703637	1 10411674	10100001	45460601	
T2	1 03607477	0 41223572	00000000	45460601	
TP	1 74170301	0 57266103	00000000	45460601	

FIGURE DIV75 A/M = 37775613 00354702 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 37404050	0 31755437	00000000	06174407	
T8	0 31755437	0 31207441	00000000	71603370	
T5	0 63733077	1 53212472	01010001	71603370	
T2	0 47666176	0 36531566	00110010	06174407	
TR	0 17554375	1 66306665	01000101	71603370	
T7	0 37330772	0 54701353	10111010	06174407	
T4	0 75661764	0 50110345	10111010	06174407	
T1	0 75543750	0 46536332	00101010	06174407	
TP	0 73307720	0 13562304	10111010	06174407	
T6	0 66617641	1 30361121	11011101	71603370	
T3	0 55437502	0 00050044	11111010	06174407	*
TC	0 33077205	1 13145421	01000101	71603370	
T8	0 66176413	1 21116534	01000101	71603370	
T5	0 54375027	1 35030763	01010101	71603370	
T2	0 30772057	1 65705441	00000101	71603370	
TR	0 61764136	0 51707703	10100010	06174407	
T7	0 43750274	0 42103226	10111010	06174407	
T4	0 07720570	0 22524074	10101010	06174407	
T1	0 17641361	1 56255500	01000101	71603370	
TP	0 37502742	0 34627001	10101010	06174407	
T6	0 77205704	0 17754421	00100010	06174407	*
T3	0 76413611	1 21734353	10000101	71603370	
TC	0 75027423	1 54474421	01000101	71603370	
T8	0 72057046	0 31265644	10100010	06174407	
T5	0 64136114	0 00047130	11101010	06174407	*
T2	0 50274231	0 22310201	00010100	71603370	*
TP	0 50274231	0 11150140	00010000	71603370	6F

FIGURE DIV76 A/M = 33747715 02416455 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 16141140	1 70702140	00000000	07054017	
T8	0 70702140	1 07065121	00000000	70723760	
T5	0 61604301	1 06075123	01001101	70723760	
T2	0 43410603	1 06017331	01101101	70723760	
TR	0 07021407	1 06164745	01000101	70723760	
T7	0 16043017	1 06275074	01001001	70723760	
T4	0 34106037	1 06417052	01101101	70723760	
T1	0 70214077	1 07163206	01001101	70723760	
TP	0 60430177	1 10273577	01001001	70723760	
T6	0 41060377	1 12414261	01101101	70723760	
T3	0 02140777	1 17155625	01001101	70723760	
T0	0 04301777	1 30250534	01011101	70723760	
T8	0 10603777	1 52466352	01001101	70723760	
T5	0 21407776	0 26123135	10110010	07054017	
T2	0 43017775	1 64301272	01111001	70723760	
TR	0 06037772	0 52070576	10010010	07054017	
T7	0 14077764	0 43255423	10000010	07054017	
T4	0 30177750	0 25507075	10100010	07054017	
T1	0 60377721	1 63331172	01011001	70723760	
TP	0 40777642	0 57750376	00010010	07054017	
T6	0 01777604	0 36714024	10101010	07054017	
T3	0 03777210	0 15006007	10100100	07054017	*
T0	0 07776421	1 42137176	01001001	70723760	
T8	0 17775042	0 05254305	10100110	07054017	
T5	0 37772105	1 22654712	01000101	70723760	
T2	0 77764213	0 47531026	00001000	70723760	
TP	0 77764213	0 23655013	00001000	70723760	OF

FIGURE DIV77 A/M = 40607402 05140070 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 22130502	1 64012150	00000000	71531776	
T8	0 64012150	1 13662500	00000000	71531776	
T5	0 50024321	1 10276177	11001000	71531776	
T2	0 20050643	1 33227265	01101110	71531776	
TR	0 40121507	1 61311660	01101110	71531776	
T7	0 00243216	0 43263762	10010001	06246001	
T4	0 00506434	0 24725747	10110001	06246001	
T1	0 01215071	1 63524616	00101100	71531776	
TP	0 02432162	0 45611635	10110001	06246001	
T6	0 05064344	0 31001475	11110001	06246001	
T3	0 12150710	0 02461175	10010001	06246001	*
T0	0 24321621	1 16613262	00101110	71531776	
T8	0 50643443	1 26361652	01001110	71531776	
T5	0 21507107	1 47476633	01001110	71531776	
T2	0 43216216	0 27335707	00110001	06246001	
TR	0 06434435	1 40644517	10001100	71531776	
T7	0 15071072	0 27751437	00010001	06246001	
T4	0 32162165	1 40474776	11000100	71531776	
T1	0 64344352	0 21330175	10110001	06246001	
TP	0 50710725	1 54631263	00001110	71531776	
T6	0 21521652	0 37622760	00110011	06246001	
T3	0 43443525	1 60316650	11101110	71531776	
T0	0 07107252	0 61275742	10010001	06246001	
T8	0 15216524	0 60051707	11010001	06246001	
T5	0 34435250	0 60301621	10110001	06246001	
T2	0 71072520	1 60023444	01000000	06246001	
TP	0 06705260	0 76657624	00000001	06246001	OF

FIGURE DIV78 A/M = 40000261 01341144 1

TIME	A REGISTER	B REGISTER	BC REGISTER	C REGISTER	
T8	0 32777271	1 57217524	00000000	72121446	
T8	0 57217524	1 25120737	00000000	72121446	
T5	0 36437251	1 44362235	00001110	72121446	
T2	0 75076522	0 15514243	01111001	05656331	
TR	0 72175245	1 27573157	00001000	72121446	
T7	0 64372513	1 41411775	10100010	72121446	
T4	0 50765226	0 20071344	11011001	05656331	
T1	0 21752455	1 54325351	00001010	72121446	
TP	0 43725132	0 35422273	01111001	05656331	
T6	0 07652265	1 67307137	00101100	72121446	
T3	0 17524552	0 54676727	10000101	05656331	
TC	0 37251324	0 56344311	01110101	05656331	
T8	0 76522650	0 34707355	10100001	05656331	
T5	0 75245520	0 17664266	00011001	05656331	*
T2	0 72513241	1 21613115	10101110	72121446	
TR	0 65226503	1 55751011	00001110	72121446	
T7	0 52455206	0 30572574	11010001	05656331	
T4	0 25132414	0 00253624	11010101	05656331	*
T1	0 52265031	1 14660210	00011110	72121446	
TP	0 24552063	1 23603207	00101100	72121446	
T6	0 51324147	1 31731254	10001010	72121446	
T3	0 22650316	0 00432002	11111101	05656331	*
TC	0 45520635	1 15427644	00000010	72121446	
T8	0 13241473	1 25170177	00011000	72121446	
T5	0 26503167	1 44523734	00000110	72121446	
T2	0 55206356	1 11247010	00001100	05656331	
TP	0 22571422	0 52402376	11010001	05656331	0F

TIME	A REGISTER	B REGISTER	FIGURE DIV79 BC REGISTER	C REGISTER	A/M = 53203760 70027323 1
T8	1 40751345	1 44374146	00000000	27153034	
T8	1 33403632	1 07653376	00000000	50624743	
T5	1 67007465	1 67342637	01011101	50624743	
T2	1 56017152	0 07072635	01010100	27153034	
TR	1 34036325	1 70021536	01011101	50624743	
T7	1 70074652	0 01130422	10110110	27153034	
T4	1 60171525	1 72325120	01001111	50624743	
T1	1 40363252	0 05727417	10100100	27153034	
TP	1 00746525	1 04673172	00011011	50624743	
T6	1 01715253	1 61434341	01001011	50624743	
T3	1 03632526	0 64145750	10100010	27153034	
T0	1 07465254	0 17666774	00000000	27153034	
T8	1 17152531	1 07371633	01011101	50624743	
T5	1 36325263	1 70621533	01011101	50624743	
T2	1 74652546	0 02530424	10110100	27153034	
TR	1 71525315	1 75325114	01001101	50624743	
T7	1 63252632	0 13727467	10100000	27153034	
T4	1 46525465	1 10673012	10011111	50624743	
T1	1 15253153	1 11434102	01001111	50624743	
TP	1 32526327	1 75616361	00101011	50624743	
T6	1 65254656	0 12701710	10110110	27153034	
T3	1 52531535	1 16640704	00010101	50624743	
T0	1 25263273	1 05346756	01000001	50624743	
T8	1 52546567	1 64531671	01011011	50624743	
T5	1 25315356	0 72450631	10010010	27153034	
T2	1 52632735	0 05141402	00000100	50624743	
TP	1 25145043	0 75317137	00000000	50624743	OF

PUNCHED PAPER
TAPE VERSION

SDS SCIENTIFIC DATA SYSTEMS

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 304003

IDENTIFICATION: P and S Register Tester

AUTHOR: M. Mulholland, SDS

ACCEPTED: August 28, 1964

COMPUTER CONFIGURATION: SDS 930

PURPOSE: To exercise the P and S Registers by storing and executing BRM's throughout memory. By comparing the "mark" of the BRM with an expected value, the program checks whether the computer stored the correct location. Therefore, the test checks whether the P and S registers functioned properly.

STORAGE: 0 - 221_8, 146 locations

SOURCE LANGUAGE: SYMBOL 8

LOADING PROCEDURE: Load the P and S Register Tester into memory at the completion of the Automatic Instruction Diagnostic program using the normal fill procedure.

1. Set Bpt 3 if only one pass is desired through the memory. Set the mode switch to RUN.

2. The program stores BRM's throughout memory, starting at location 100_8.

3. The computer executes the BRM's.

4. In case of an error, the computer halts; move mode switch to IDLE.

   The registers contain the following information.

   A = Expected location

   B = Actual location designated by "mark" of the BRM instruction just executed

   X = Which 4K bank, X = 0 through 7

   P = 0037

   C = 07100053

LOADING PROCEDURE: (cont.)

5. Set Bpt 1 if an error occurs. The computer will execute repeatedly the BRM just executed.
6. To inhibit the error halt, set Bpt 2.
7. Move the mode switch to RUN.
8. If Bpt 3 is set, the computer halts when it has executed all of the BRM's.
9. If the mode switch is set to RUN, the program reexecutes the set of BRM's.
10. If Bpt 3 is reset, the computer will loop through the program repeatedly until Bpt 3 is set.

```
                        *
                        *      LOADER FOR P AND S REGISTERS EXERCISER
                        *
           00002               ORG    2
00002  2 32 00012              WIM    10,2
00003  0 41 00002              BRX    2
00004  0 32 00176              WIM    126
00005  0 32 00177              WIM    127
00006  0 71 00010              LDX    8
00007  0 01 00176              BRU    126
00010  00077756                OCT    77756
00011  00000000                OCT    0
00012        00164             BSS    116
```

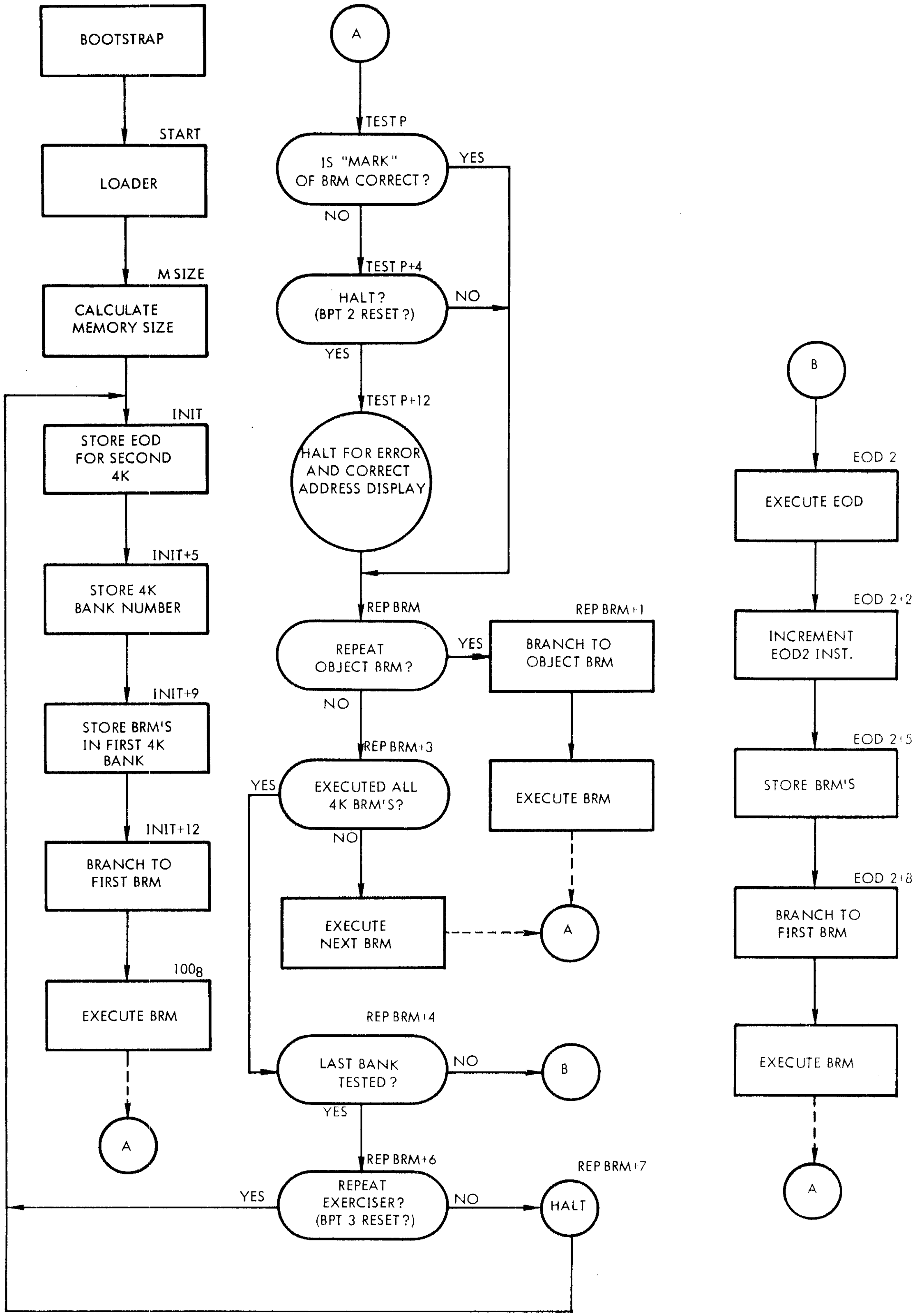
```
                                *
                                *     LOADER FOR P AND S REGISTERS EXERCISER
                                *
                   00002                    ORG     2
00002   2 32 00012                          WIM     10,2
00003   0 41 00002                          BRX     *-1
00004   0 71 00011                          LDX     9
00005   2 32 04000                          WIM     2048,2
00006   0 40 21000                          BRTW
00007   0 41 00005                          BRX     *-2
00010   0 01 04000                          BRU     2048
00011   0 00 40000                          HLT*    0
                                *
                                *                               CARD READER LOADER
                                *
                   04000                    ORG     2048
04000   0 71 00011              LOAD        LDX     9
04001   0 02 03606                          RCBW    1,4             READ CARDS BINARY
04002   2 32 00000                          WIM     0,2
04003   0 40 21000                          BRTW                    BUFFER READY TEST
04004   0 41 04002                          BRX     *-2
04005   0 40 11006              FILE        CFTW    1               END OF FILE TEST
04006   0 01 04012                          BRU     MSIZE
04007   0 40 12006                          CRTW    1               CARD READER READY
04010   0 01 04005                          BRU     FILE            NO
04011   0 01 04001                          BRU     LOAD+1
                                            PAGE
04012   0 76 04012              MSIZE       LDA     *
04013   0 61 04014                          MIN     EOD1
04014   0 06 20100              EOD1        EOD     64
04015   0 35 20000                          STA     8192
04016   0 72 20000                          SKA     8192
04017   0 01 04013                          BRU     MSIZE+1
04020   0 76 04014                          LDA     EOD1
04021   0 54 00063                          SUB     =0620101
04022   0 35 00077                          STA     63
04023   0 76 04025                          LDA     ZERO
04024   0 01 00000                          BRU     0
```

```
                               PAGE
04025  00000000         ZERO   OCT    0
                        EOD    OPD    620000
04026  0 00 00000              PZE
04027  0 00 00000              PZE
04030  0 00 00000              PZE
04031  0 00 00000              PZE
04032  0 00 00000              PZE
04033  0 00 00000              PZE
04034  0 00 00000              PZE
04035  0 00 00000              PZE
04036  0 00 00000              PZE
04037  0 00 00000              PZE
                               PAGE
            00000              ORG    0
00000  0 75 00064       INIT   LDB    =O7777              MASK
00001  0 35 00060              STA    WK2
00002  0 76 00065              LDA    =O620210            EOD FOR OVER 4K
00003  0 35 00015              STA    EOD2
00004  0 76 00077              LDA    63
00005  0 35 00057              STA    WK1
00006  0 06 20200              EOD    128
00007  0 71 00066              LDX    =O70100
00010  0 76 00061              LDA    BRM                 STORE BRM INSTRUCTIONS
00011  2 35 10000              STA    4096,2              IN LOW 4K OF MEMORY
00012  0 41 00011              BRX    *-1
00013  0 71 00067              LDX    =O70101
00014  0 01 00100              BRU    64
                               PAGE
00015  0 00 00000       EOD2   PZE                        GREAT THAN 4K
00016  0 76 00070              LDA    =O10
00017  0 63 00015              ADM    EOD2                INCREMENT BANK EOD
00020  0 71 00071              LDX    =O70000
00021  0 76 00061              LDA    BRM                 STORE BRM INSTRUCTION
00022  2 35 00000              STA    0,2                 IN OVER 4K MEMORY
00023  0 41 00022              BRX    *-1
00024  0 71 00071              LDX    =O70000
00025  2 41 00000              BRX    0,2
```

```
                                 PAGE
00026  0 00 00000     TESTP     PZE
00027  0 46 00200               CXA                         TEST REGISTERS
00030  0 54 00072               SUB     =1
00031  0 70 00026               SKM     TESTP               CORRECT
00032  0 40 20200               BPT     2                   NO
00033  0 01 00045               BRU     REPBRM              YES
00034  0 75 00026               LDB     TESTP               B CONTAINS ERROR ADDRESS
00035  0 37 00062               STX     SAVEX
00036  0 71 00060               LDX     WK2
00037  0 66 20014               RCY     12                  MERGE
00040  0 46 00340               RCH     340                  BANK
00041  0 66 20014               RCY     12                  LOCATION
00042  0 00 00000               HLT                         A CONTAINS CORRECT ADDRESS
00043  0 71 00062               LDX     SAVEX
00044  0 75 00064               LDB     =07777              MASK
00045  0 40 20400     REPBRM    BPT     1                   REPEAT BRM
00046  2 01 37777               BRU     16383,2
00047  2 41 00000               BRX     0,2
00050  0 61 00060               MIN     WK2
00051  0 60 00057               SKR     WK1                 LAST BANK
00052  0 01 00015               BRU     EOD2                NO
00053  0 40 20100               BPT     3                   YES-REPEAT ROUTINE
00054  0 00 00000               HLT                         NO
00055  0 76 00042               LDA     TESTP+12
00056  0 01 00000               BRU     INIT                YES
                                PAGE
00057  0 00 00000     WK1       PZE
00060  0 00 00000     WK2       PZE
00061  0 43 00026     BRM       BRM     TESTP
00062  0 00 00000     SAVEX     PZE
                      EOD       OPD     620000
            00000               END
00063  00620101
00064  00007777
00065  00620210
00066  00070100
00067  00070101
00070  00000010
00071  00070000
00072  00000001
```

REPBRM	00045	MSIZE	04012	SAVEX	00062	TESTP	00026
EØD1	04014	EØD2	00015	FILE	04005	INIT	00000
LØAD	04000	ZERØ	C4025	BRM	00061	EØD	04067
EØD	04026	WK1	00057	WK2	00060		

SDS 930 P & S REGISTER TESTER



SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

 Catalog No. 594003B

IDENTIFICATION:	SDS 925/930 Rapid Access Data (RAD) Files Apocalyptic Diagnostic Program
AUTHOR:	SDS
DATE:	June 16, 1966
COMPUTER CONFIGURATION:	Any SDS 925 or 930 Computer with one typewriter attached to the W-buffer, and one or more RAD files attached to any channel using interlace
PURPOSE:	To provide the operator with a means of testing and checking out the SDS rapid access data files
PROGRAMMED OPERATORS:	None
SUBROUTINES REQUIRED:	None
STORAGE:	The program nominally occupies 3708_{10} locations from 0000_8 to 7174_8. When operated, a minimum of 4K locations are used
SOURCE LANGUAGE:	METASYMBOL
LOADING PROCEDURES:	Standard fill from paper tape or cards

SECTION I

METHOD OF OPERATION

1.1 INITIATING THE PROGRAM

1.1.1 The program is supplied on paper tape or cards with a self-loading bootstrap.

1.1.2 To load the program, place the program medium (cards or paper tape) into the appropriate reader, reset all BREAKPOINT switches, and follow the standard fill procedure.

1.1.3 When the program is loaded, a message RAD APOCALYPTIC DIAGNOSTIC is typed to indicate that the program is now under operator control.

1.2 OPERATING THE PROGRAM

1.2.1 By the input of the appropriate directives at the keyboard, the operator can initiate procedures that will check out the RAD file under test.

1.2.2 When the program is loaded and started normally, an initializer routine sets up certain run-time parameters to initial values. The initializer is not operated again when a restart is made; all run-time parameters retain the values they had before restart.

1.3 REGAINING OPERATOR CONTROL

1.3.1 To regain operator control after the test program is activated, the operator sets BREAKPOINT switch 1 or moves the RUN-IDLE-STEP switch to IDLE, presses RESET or START, moves the switch to STEP and then to RUN. If the instruction in location 0001 is destroyed, he executes a BRU to location 0300. This will return control to the keyboard.

1.3.2 When a restart at 0300 is made, all parameters retain the values they had initially; the initializer is not operated again.

1.3.3 Recovery of operator control via any method other than described in paragraph 1.3.1 is not recommended.

1.4 BREAKPOINT SWITCHES

1.4.1 BREAKPOINT switches exercise the following functions:

BREAKPOINT 1	RESET	Continues sequence of operations
	SET	Returns control to the typewriter in the immediate mode after the control line has been executed.

BREAKPOINT 2	RESET	In the interrupt test, keeps the sector address constant.
	SET	In the interrupt test, advances the sector address by 1. (After BREAKPOINT 2 is placed in a SET position, it is returned to the RESET position.)
BREAKPOINT 3	RESET	In the compare function, allows bit summary typeout. In the interrupt test, continues sequence of operations. In the octal card read function, continues card read operation.
	SET	During the execution of the compare function, suppresses bit summary typeout. In the interrupt test, aborts testing and returns control to keyboard. In the octal card read function, aborts reading and returns control to the keyboard.
BREAKPOINT 4	RESET	Allows all status and error messages to be output at the keyboard.
	SET	Suppresses all message output except in those certain cases (specifically named in the directive descriptions) where typeout is not suppressible.

SECTION II

PROGRAM DIRECTIVES

2.1 The following discussion assumes familiarity with the SDS 925/930 Diagnostic Control Program (catalog No. 004009) and the test language described therein.

2.2 The operator directs the operation of this program through the input of directives, each named by a unique mnemonic. These directives are entered at the keyboard or by card reader. Most directives have optional or required parameters that follow the mnemonic call, and optional iteration count or place marker identifiers that precede the mnemonic directive.

2.3 Functional directions fall into two categories:

a. Environmental directives, which permit the user to initialize and modify certain environmental parameters that affect subsequent operational directives.

b. Operational directives, which validate data transfer operations. Some of these directives directly perform data manipulation and comparison; others facilitate the task.

2.4 The directives associated with the program are described in paragraph 2.6 of this section.

2.5 Table 2-1 lists the directives of this RAD apocalyptic diagnostic program. The directives are arranged alphabetically by mnemonic name. The subparagraph reference after each mnemonic indicates the location of the detailed description.

Table 2-1. RAD Directives

Directive	Class*	Function	Mode**	Paragraph
ADR	E	Declare Device Starting Address	I	2.6.6
BAK	E	Return To Starting Address	I or C	2.6.7
C	O	Compare Data	I or C	2.6.23
CHN	E	Declare I/O Channel	I	2.6.12
CTR	O	Counter Control	I or C	2.6.2
DLY	O	Time Delay	I or C	2.6.26
ERR	O	Error Report	I or C	2.6.4
INT	O	Priority Interrupt Test	I	2.6.16

* **See footnotes, page 5.

Table 2-1. RAD Directives (Cont.)

Directive	Class*	Function	Mode**	Paragraph
JMP	U	Jump To Location	I or C	2.6.27
LEN	E	Set Record Length	I or C	2.6.11
MARK	O	Mark Identify	I or C	2.6.28
OCT	U	Octal Card Read	I	2.6.22
PAT	E	Declare Test Pattern	I or C	2.6.14
Q	O	Query Tests	I or C	2.6.25
R	O	Read Record	I or C	2.6.20
RNON	O	Read Record, Inhibit Band Incrementation	I or C	2.6.21
S	E	Move Device Address	I or C	2.6.8
STOP	O	Stop Identify	I or C	2.6.29
TDA	O	Type Device Address	I or C	2.6.10
TRY	E	Declare Retry Count	I	2.6.5
TYC	O	Type Counters	I or C	2.6.3
TYP	O	Type I/O Buffer	I or C	2.6.13
UNT	E	Declare Unit Environment	I	2.6.1
UPA	D	Update Device Address	I or C	2.6.9
W	O	Write Record	I or C	2.6.17
WNON	O	Write Record, Inhibit Band Incrementation	I or C	2.6.18
WSEQ	O	Write Sequentially	I or C	2.6.19
XC	O	Declare Compare Typeout Mode	I or C	2.6.24
XPAT	O	Rotate Pattern	I or C	2.6.15

* E = Environmental; O = Operational; U = Utility
** I = Immediate; C = Control Line

2.6 DIRECTIVES USED WITH THE RAD APOCALYPTIC DIAGNOSTIC PROGRAM

In the following discussion, trailing parameters are referenced as P_1, P_2, P_3, etc.; and leading parameters are iteration counts, i, or place markers, m.

2.6.1 Directive: Declare Unit Environment (UNT)

Purpose: To establish the hardware environment to be tested and to specify the unit sequence in multiple-unit installations

Parameters, Trailing:
P_1 (octal), where $0 \leq P_1 \leq 7$
P_2 (octal), where $0 \leq P_2 \leq 4$
P_3 (octal), where $0 \leq P_3 \leq 1$

Parameters, Leading: None allowed

Mode: Immediate only

Format: $UNT, P_1, P_2, P_3

Explanation: P_1 specifies the unit assignment number (0 through 7), P_2 specifies the number of basic (.5 million character) storage modules contained in the device (that is, the device capacity). This number is the last digit of the dash portion of the model number; -X4 shows 4 modules or a two-million character device. If a two-million character device is specified, two entries are made in the unit list (namely, two one-million character units). Eight is the maximum number of unit entries that can be made. When this limit is exceeded, a typeout occurs that indicates that the unit list is full. Any nonzero value for parameter P_3 purges any previous entries in the unit list.

2.6.2 Directive: Counter Control (CTR)

Purpose: To provide the capability of setting, resetting, incrementing, decrementing, and testing the fifteen counters maintained within the RAD program. These counters may be tested individually or in combination. The first nine counters (1 through 9) are general-purpose counters that can only be modified by this directive. The last six counters (10 through 15) are modified internally by other functions in response to various conditions which occur. Table 2-2 contains a detailed description of each counter and its use.

Table 2-3 indicates the logical indicators (bits) used as entrance parameter P_1 to specify which counters are operated upon.

Parameters, Trailing:

P_1 (octal), where $0 \leq P_1 \leq 77777$

P_2 (decimal), where $0 \leq P_2 \leq 9999$

P_3 (decimal), where $0 \leq P_3 \leq 2$

P_4 (decimal), where $0 \leq P_4 \leq 9999$ (maximum)

P_5 is alphabetic word TYPE or is not present

Parameters, Leading: Place mark (m) allowed

Mode: Immediate or control line

Format: mCTR, P_1, P_2, P_3, P_4, P_5

Explanation: P_1 specifies which counters are to be operated upon, P_2 is the value of the operation, P_3 indicates the operation, and P_4 is the test value. Each bit of the 15-bit parameter P_1 represents a counter (see table 2-2). In P_3, 0 equals set value, 1 equals increment by value, and 2 equals decrement by value. A counter tests true if the counter value is greater than P_4 (the test value) or the counter is less than zero (negative). In both cases, the test takes place after the counter has been modified by P_2 (the value of the operation), using the operation specified by P_3 (set, increment, decrement). Each time a counter tests true, the counter's name and value are output on the console typewriter if parameter P_5 contains the word TYPE.

The CTR directive may be preceded by a place mark parameter to provide conditional branching within the control line. If any counter specified by P_1 tests true, the place mark branch is taken; if not, the control line continues in sequence.

Example: ..+ 4CTR,01011,1,1,12,TYPE . . . + 4.STOP,0001

This indicates that counters 1, 4, and 10 are to be incremented by 1 and tested for value greater than 12. If any counter tests true, its name and value are typed, and control continues at place mark 4. Otherwise, the next directive in line is executed.

Table 2-2. RAD-Maintained Counters

Counter Name	Counter Number	Use
Counter 1	1	General purpose
Counter 2	2	General purpose
Counter 3	3	General purpose
Counter 4	4	General purpose
Counter 5	5	General purpose
Counter 6	6	General purpose
Counter 7	7	General purpose
Counter 8	8	General purpose
Counter 9	9	General purpose
Record Counter	10	Shows the number of records that have passed since the last address declaration. This counter is unconditionally set to 1 when an ADR or UPA directive is executed, and is incremented by 1 each time the S directive is executed
Channel Errors-Read	11	Shows the number of channel errors that have occurred during read operations
Channel Errors-Write	12	Shows the number of channel errors that have occurred during write operations
Coupler Errors-Read	13	Shows the number of coupler errors that have occurred during read operations
Coupler Errors-Write	14	Shows the number of coupler errors that have occurred during write operations
Data Errors	15	Incremented by 1 for each data word that does not compare with the test pattern (see C directive).

Table 2-3. Counter Logical Indicators (Bits)

Octal Value	Counter
00001	Counter 1
00002	Counter 2
00004	Counter 3
00010	Counter 4
00020	Counter 5
00040	Counter 6
00100	Counter 7
00200	Counter 8
00400	Counter 9
01000	Record Counter
02000	Channel Errors-Read
04000	Channel Errors-Write
10000	Coupler Errors-Read
20000	Coupler Errors-Write
40000	Data Errors

Note: Indicators are logically merged to operate on more than one counter. For example, $P_1 = 77777$ would specify all counters.

2.6.3 Directive: Type Counters (TYC)

Purpose: To interrogate the 15 counters maintained within the program

Parameters, Trailing: P_1 (octal), where $0 \leq P_2 \leq 77777$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: TYC, P_1

Explanation: Parameter P_1 indicates the counters to be typed by placing a one bit into the bit position corresponding to the desired counter.

2.6.4 Directive: Error Report (ERR)

Purpose: To typeout the value of the five error counters

Parameters, Trailing: P_1 (decimal), where $0 \leq P_1 \leq 9999$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: ERR, P_1

Explanation: Parameter P_1 is the error report frequency. Every time the ERR directive is executed, an internal counter is incremented by 1 and then compared with the value specified by P_1. If this counter is greater than P_1, the value of the five error counters are output on the typewriter and the internal error frequency counter is reset to zero. If P_1 is zero, the five error counters are set to zero.

2.6.5 Directive: Declare Retry Count (TRY)

Function: To cause an internal retry counter to be set at the value specified by parameter P_1. This counter is referenced when an error occurs during data transfers (see W and R directives, 2.6.17 and 2.6.20, respectively).

Parameters, Trailing: P_1 (decimal), where $0 \leq P_1 \leq 9999$

Parameters, Leading: None allowed

Mode: Immediate only

Format: $TRY, P_1

2.6.6 Directive: Declare Device Starting Address (ADR)

Function: To establish the initial operating points in the RAD environment

Parameters, Trailing: P_1 (octal), where $0 \leq P_1 \leq 7$
P_2 (octal), where $0 \leq P_2 \leq 7777$

Parameters, Leading: None allowed

Mode: Immediate only

Format: $ADR, P_1, P_2

Explanation: P_1 specifies the unit and P_2 specifies the band and sector address on the unit. The P_2 parameter initializes the two address counters maintained by this program. One counter is called starting device address; it can only be changed by this directive (ADR) and the UPA (Update Device Address) directive (see paragraph 2.6.9). The other counter is called the current device address. It is initialized by this directive (ADR) and by the Return to Starting Address (BAK) directive (paragraph 2.6.7); and incremented by the S (Move Device Address) directive (paragraph 2.6.8). All data transfer operations start at the address specified by the current device address counter. The program is initialized with unit 0, band 00 sector 00 as the starting device address.

Example: $ADR ,1,3125 CR

Immediate mode ($)

Directive (ADR)

Band 31_8, sector 25_8 starting address of unit 1 (,1,3125)

Carriage return (field separator) (CR)

Error Message: If the unit specified by P_1 does not exist in the RAD environment as defined by the UNT directive, an error message is typed. If the band address exceeds the capacity of the unit specified by P_1, an error message is typed and a band and sector address of 00 00 is used.

2.6.7 Directive: Return to Starting Address (BAK)

Function: To set the current device address equal to the starting device address

Parameters, Trailing: None allowed

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: BAK

2.6.8 Directive: Move Device Address (S)

Function: To cause the current device address to be incremented by an amount equal to the number of sectors required to contain the current record length as set by LEN (Set Record Length) directive (see paragraph 2.6.11). If the current device address is incremented past a unit address boundary as set by the UNT directive, the next unit in the unit list is selected as the current unit, and the band-sector address is set to 00 00.

Parameters, Trailing: None allowed

Parameters, Leading: Iteration counts (i) allowed

Mode: Immediate or control line

Format: iS

Explanation: The iteration count effectively moves the current device address by i (n) sectors, where (n) is the number of sectors needed to contain the current record length.

2.6.9 Directive: Update Device Address (UPA)

Function:	To cause the current device address to be initialized as the starting device address. When this directive is executed, the record counter is unconditionally set at 1.
Parameters, Trailing:	None allowed
Parameters, Leading:	None allowed
Mode:	Immediate or control line
Format:	UPA

2.6.10 Directive: Type Device Address (TDA)

Function:	To cause a typeout RECORD STARTING ADDRESS UBBSS where U is the current unit number, BB is the current band address, and SS is the current sector address.
Parameters, Trailing:	None allowed
Parameters, Leading:	None allowed
Mode:	Immediate or control line
Format:	TDA

2.6.11 Directive: Set Record Length (LEN)

Function:	To establish the length of logical records for all data transfers and compare operations
Parameters, Trailing:	P_1 (decimal), where $0 \leq P_1 \leq 4096$
Parameters, Leading:	None allowed
Mode:	Immediate or control line

Format: LEN, P_1

Explanation: Parameter P_1 is the desired length, expressed in words, where $1 \leq P_1 \leq 4096$. If P_1 has a value of zero, the record length is set to 4096 words (1 band). If the program is to be run on a minimum configuration system (that is, 4K memory), then the limits are $1 \leq P_1 \leq 128$ (namely, the record length is limited to 2 sectors). The program is initialized with the record length set to 64 words (1 sector).

2.6.12 Directive: Declare I/O Channel (CHN)

Function: To define the I/O channel associated with the RAD

Parameters, Trailing: P_1 (alphabetic), where $A \leq P_1 \leq H$

Parameters, Leading: None allowed

Mode: Immediate only

Format: \$CHN, P_1

Explanation: P_1 is the mnemonic identifier of this particular channel and is input in the form of a single alphabetic character, in the range A through H. Channels W and Y are represented by letters A and B, respectively. The program is initialized with channel A being assigned.

2.6.13 Directive: Type I/O Buffer (TYP)

Function: To cause the contents of the I/O buffer area to be typed at the console typewriter

Parameters, Trailing: P_1 (decimal), where $0 \leq P_1 \leq 4096$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: TYP, P_1

Explanation: Parameter P_1 determines the relative location within the buffer where the typing is to start. The relative location of the first word of the buffer is 1. If the parameter is 0 or blank, the standard value is set to 1. Words typed are preceded by their relative location within the buffer. The number of words typed out is equivalent to the length of the record read length. The unit-band-sector address at which the last data transfer was initiated is typed prior to the output of the data.

2.6.14 Directive: Set Test Pattern (PAT)

Function: To set the test pattern for the write and compare operations

Parameters, Trailing: P_1 (octal), where $0 \leq P_1 \leq 77777777$

P_2 (decimal), where $1 \leq P_2 \leq 8$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: PAT, P_1, P_2

Explanation: The desired test pattern is specified by parameter P_1 as 1 to 8 octal digits. Parameter P_2 specifies the number of octal digits input in P_1. The pattern generated for write and compare operations is based on the number of octal digits input in P_1. If the number of digits, N, is an even divisor of 8 (the number of octal digits in a 24-bit word), the pattern generated is cyclic every word; that is, every word in the pattern is the same. If N is not an even divisor of 8, however, the pattern is cyclic every Nth word.

Example: PATTERN GENERATED

		Word	Pattern
Example:	PAT,52,2	1	52525252
		2	52525252
		.	.
		.	.
		.	.
	PAT,543,3	1	54354354
		2	35435435
		3	43543543

PATTERN GENERATED

Word	Pattern
4	54354354
⋮	⋮

The program is initialized with a pattern of zeros.

2.6.15 Directive: Rotate Pattern (XPAT)

Function: To cause the first word of the existing pattern to be rotated right the number of bit positions specified by parameter P_1. The word obtained as a result of this shift is used in conjunction with the pattern digit count declared by the last PAT directive to define a new pattern.

Parameters, Trailing: P_1 (decimal), where $0 \leq P_1 \leq 48$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: XPAT, P_1

2.6.16 Directive: Priority Interrupt Test (INT)

Function: To test the optional priority interrupt available with the RAD for interrupt interval timing, multiple interrupt occurrence, and correct sector reporting.

Parameters, Trailing: P_1 (octal), where $0 \leq P_1 \leq 7$

P_2 (octal), where $200 \leq P_2 \leq 277$

P_3 (octal), where P_3 is any zero or nonzero value

Parameters, Leading: None allowed

Mode: Immediate only

Format: \$INT, P_1, P_2, P_3

Explanation: Parameter P_1 specifies the unit to be tested (0 through 7). Parameter P_2 specifies the interrupt link address, 200_8 through 277_8. Parameter P_3 is used to indicate the installation of the optional arming feature: zero indicates that it is not installed, any nonzero value indicates that it is installed.

Interrupt testing starts with sector 00 and advances sequentially through sector 77_8.

The testing can only be done in the immediate mode. To abort testing set BREAKPOINT 3.

Error Messages: If P_1 unit is not in the RAD environment as previously defined by the UNT function, an error message is typed and control returns to the keyboard. If an error occurs in interrupt testing, it is reported on the typewriter and that sector is tested again. This continues until the sector interrupt is received correctly or BREAKPOINT 2 is set and reset, at which time the next sector in sequence is tested. A number of error typeouts can occur:

a. INTERRUPT TIMING ERROR XX. XX indicates that an interrupt was received but was outside the allowable time limits (1 disc revolution per 35 ms). XX. XX represents the elapsed time in milliseconds from address POT to the interrupt occurrence.

b. COUPLER SECTOR XXXX YYYY indicates that the sector address that was input in parallel (PIN) does not equal the sector address that was output in parallel (POT). XXXX is the expected sector address and YYYY is the sector address received.

c. MULTIPLE INTERRUPTS XXXX occurs when more than one interrupt was received in two disc revolutions (70 milliseconds). XXXX is the number of interrupts received during the 70 ms after the disc-address POT operation.

d. INTERRUPT NOT RECEIVED occurs when an interrupt is not received within 70 milliseconds after the address POT operation. If any sector fails to give an interrupt signal, the program does not loop on that sector but selects the next sector in sequence.

2.6.17 Directive: Write Record (W)

Function: To cause one record to be written on the RAD, starting at the current device address.

Parameters, Trailing: P_1 (octal)

Parameters, Leading: Iteration counts (i) allowed

Mode: Immediate or control line

Format: iW, P_1

Explanation: If Parameter P_1 is zero or not present, an attempt is made to write over bands that are write-protected. If Parameter P_1 is nonzero, any write-protected bands are skipped; when a write-protected band is encountered, an error message is typed.

Error Message: Errors are detected in the channel and the coupler and reported at the typewriter (if BREAKPOINT 4 is reset) before the retry counter is checked to determine whether to repeat the write operation. Iteration counts are not performed if the retry count is exhausted before the write is successfully completed.

If an error is detected, one or more of the following messages is typed:

WRITE ERROR CHANNEL ERROR

WRITE ERROR COUPLER ERROR

WRITE ERROR RECORD LENGTH ER NNNN XXXX

WRITE ERROR COUPLER SECTOR YYYY ZZZZ

In the typeouts above, NNNN is the expected word transfer count (the write record length), XXXX is the actual transfer count, YYYY is the expected sector address at completion of the data transfer, and ZZZZ is the received sector address. The error report is followed by a typeout of the current device address.

2.6.18 Directive: Write Record, Inhibit Band Incrementation (WNON)

Function: To serve as an alternative write function, identical to W, except that nonband incrementing EOM's are used.

Parameters, Trailing: P_1 (octal)

Parameters, Leading: Iteration counts (i) allowed

Mode: Immediate or control line

Format: iWNON, P_1

Explanation: (same as W directive, paragraph 2.6.17).

2.6.19 Directive: Write Sequentially (WSEQ)

Function: To test the disc file's ability to perform gather write operations between adjacent sectors without the attendant loss of access time (1 disc revolution).

Due to the special nature of this test, certain restrictions are placed on its use:

a. At least 8K words of memory are required.

b. The test cannot operate in write-protected areas.

c. If the disc file has the optional priority interrupt, and if the interrupt link address is not in the first special system interrupt group, then the priority interrupt module should be removed.

d. Care should be exercised in choosing the initial starting address so that unit address boundaries will not be crossed.

Parameters, Trailing: None allowed

Parameters, Leading: Place mark (m) allowed

Mode: Immediate or control line

Format: mWSEQ

Explanation: When the test is run, the following sequence of operations will occur:

The current device address at the time the directive is executed is declared at the starting device address; starting at this address 4096 words of information (one band or 64 sectors) will be transferred to the disc. This information (consisting of the unit-band-sector address of the sector being written) is transferred, one sector at a time (that is, each sector gets its own address written). Extended mode I/O operation is used, with end-of-record interrupts occurring at the completion of each sector transfer. At this time, channel and coupler error indicators are checked and the next sector transfer operation is initiated.

After the write operation is completed (whether normal or aborted), the entire 4096 words are read back into the I/O buffer area for subsequent verification. (No provision is made for automatic verification; it is recommended that this directive be followed by a TYP directive to allow visual verification.)

The place mark branch will be taken if any errors are detected in the write operation.

Upon exit, the record length will be set to 4096 words.

Error Messages: The following typeouts, all suppressible with BREAKPOINT 4, may occur:

a. INTERRUPT TIMING ERROR –

   RECORD STARTING ADDRESS UBBSS

This occurs if the preceding sector failed to transfer in less than 20 milliseconds (that is, 1 disc revolution was lost). UBBSS is the unit-band-sector address of the sector following the one that failed, numerically one greater. (See timing chart, figure 2-1.)

b. CHANNEL ERROR

   RECORD STARTING ADDRESS UBBSS

This typeout occurs if a channel error is detected during any transfer operation. UBBSS is the same as in step a.

c. COUPLER ERROR

   RECORD STARTING ADDRESS UBBSS

This typeout occurs if a coupler error is detected during any transfer operation. UBBSS is the same as in step a.

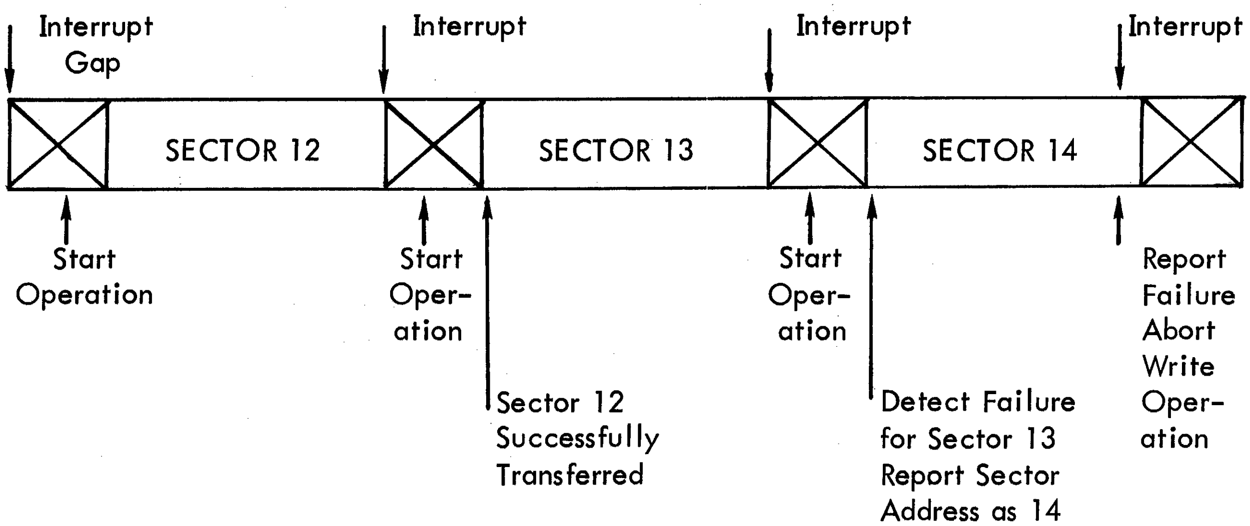


Figure 2-1. Timing Chart

If any of the errors are detected, the ongoing operation is completed (that is, the sector after the one that failed is transferred), the error is reported, and all further write operations are aborted.

2.6.20 Directive: Read Record (R)

Function: To cause one record to be read from the RAD, starting at the current device address.

Parameters, Trailing: P_1 (octal)

Parameters, Leading: Iteration counts (i) allowed

Mode: Immediate or control line

Format: iR, P_1

Explanation: If P_1 is zero or not present, write-protected bands are read. If parameter P_1 is nonzero, write-protected bands are not read and an error message is typed to indicate a write-protected band is encountered.

Error Messages: Errors are detected in the channel and the coupler and are reported at the typewriter (if BREAKPOINT 4 is reset) before the try count is checked to determine whether or not to repeat the attempt to read correctly. Iteration counts are not performed if the retry count is exhausted before the read is successfully completed.

If an error is detected, the following messages may be typed:

READ ERRORS CHANNEL ERROR

READ ERRORS COUPLER ERROR

READ ERRORS RECORD LENGTH ER NNNN XXXX

READ ERRORS COUPLER SECTOR YYYY ZZZZ

In the above, NNNN is the expected word transfer count (read record length), XXXX is the actual transfer count, YYYY is the expected sector address at the completion of the data transfer, and ZZZZ is the received sector address. The error report is followed by a typeout of the current device address.

2.6.21 Directive: Read Record, Inhibit Band Incrementing (RNON)

Function:	To serve as an alternate read operation, identical to the R directive, except that nonband-incrementing EOM's are used.
Parameters, Trailing:	P_1 (octal)
Parameters, Leading:	Iteration count (i) allowed
Mode:	Immediate or control line
Format:	iRNON, P_1

Explanation: (same as the R directive, paragraph 2.6.20).

2.6.22 Directive: Octal Card Read (OCT)

Function:	To read cards from card reader 1 on the channel indicated by the CHN directive.
Parameters, Trailing:	None allowed
Parameters, Leading:	None allowed
Mode:	Immediate only

Format: $OCT

Explanation: The card format is as follows:

```
0      0 0      1 1   1 1      2 2     3       7       8
1      8 9      4 5   6 7      4 5     2 . . . 3       0
unused address        1st word 2nd word        8th word
```

Up to eight locations may be altered by each card. These locations may be contiguous with the address named in columns 9 through 14. If a blank field or column 80 is passed, the program alters the named locations, then reads the next card. Leading zeros in each field must be punched. Cards will be read until the hopper is empty.

Another function of this directive is for patching purposes utilizing the JMP function (see paragraph 2.6.27).

Error Messages: An alphabetic or nonoctal digit in the address or data word fields or a channel error will result in an error typeout.

When the card reader is not ready, a message is typed. In order to resume the control line sequence of operations or to make the card reader ready to read more cards, BREAKPOINT 3 must be set and reset.

2.6.23 Directive: Compare Data (C)

Function: To perform a word-for-word comparison of the I/O buffer area with the current test pattern as defined by the last PAT directive executed.

Parameters, Trailing: None allowed

Parameters, Leading: Place mark (m) allowed

Mode: Immediate or control line

Format: mC

Explanation: This comparison takes place on N words, where N is defined as the record read length expressed in words. In the event that the record read length is greater than the record write length (namely, the record write length is not modulo sector length), then a word of zeros is used as the pattern compare word to verify that zeros were written past the record write length to the next sector boundary.

When the directive C is used in the control line, it may be preceded by a place mark parameter. If the place mark is present, the branch occurs when compare errors are detected; otherwise, control line execution continues in sequence.

Example: . . .+ 5C . . . 5.STOP,0

If any compare errors are detected, the place mark branch is taken and the next directive executed is the STOP directive (see paragraph 2.6.29) appearing at place mark 5. If all comparisons test all right, the next directive after 5C + is executed.

Error Messages: If a compare error is detected, the following message is typed (except for special case noted in paragraph 2.6.24):

COMPARE ERRORS @ AAAAA

N EEEEEEEE RRRRRRRR

In the above, AAAAA represents the unit-band-sector address at which the last data transfer took place, N is the relative location of the word in the I/O buffer, EEEEEEEE is the expected word pattern, and RRRRRRRR is the word received.

If any compare errors are detected after all comparisons are made, a bit summary typeout occurs in the following format:

SUMMARY AAAAA

BIT POS.	DROPPED	GAINED
0	X	X
.	.	.
.	.	.
.	.	.
23	X	X

In the above, AAAAA represents the unit-band-sector address at which the last data transfer took place and X is the number of bits dropped or gained in a particular bit position.

Setting BREAKPOINT 4 suppresses the compare error typeout; setting BREAKPOINT 3 suppresses the bit summary typeout.

2.6.24 Directive: Declare Compare Typeout Mode (XC)

Function: To determine in which mode the compare typeout is to take place.

Parameters, Trailing: P_1 (octal), where $0 \leq P_1 \leq 1$

Parameters, Leading: None allowed

Mode: Immediate or control line

Format: XC, P_1

Explanation: P_1 specifies the compare error report made. If P_1 = 0, the normal mode is reported; if P_1 = 1, the inhibit mode prevails. The first compare error is reported in the normal manner but all subsequent compare error typeouts are suppressed. The breakpoint options as indicated in paragraphs 1.4.1 and 2.6.23 remain in effect.

2.6.25 Directive: Query Tests (Q)

Function: To request that certain tests be performed.

Parameters, Trailing: P_1 (octal), where $0 \leq P_1 \leq 77$

Parameters, Leading: Place mark (m) allowed

Mode: Immediate or control line

Format: mQ, P_1

Explanation: P_1 indicates which SKS (Skip If Signal Not Set) test or tests are to be performed. If any of the test results are true, the branch to place marker is taken. If a not-true condition prevails on all tests requested, the branch does not occur; the next directive in the control line sequence is executed instead.

Tests may be requested using the following values for P_1:

	Parameter P_1	SKS Test	Typeout
NOT SET FOR ERROR	1	Coupler Error	CPLR. ERROR[1]
	2	Write Protected	WRITE PROTECTED[1]
SET	4	RAD Ready	RAD READY[2]
FOR NOT ACTIVE	10	Channel Active	CHANNEL ACTIVE[2]
SET FOR	20	Word Count = 0	WD. COUNT=ZERO[2]
	40	Channel Interrecord	CH INTERRECORD[2]

A zero or blank is used for P_1 if a channel error test is desired. If an error exists, CHN. ERR AAAAA is typed and the place mark branch is taken. AAAAA is the unit-band-sector address at which the last data transfer took place.

A combination of tests may be requested by merging respective parameter values. For example, $P_1 = 11$ would test for both channel error and channel active.

2.6.26 Directive: Time Delay (DLY)

Function: To cause a time delay for the number of milliseconds specified by the trailing parameter.

Parameters, Trailing: P_1 (decimal), where $0 \leq P_1 \leq 9999$

Parameters, Leading: Iteration count (i) allowed

Mode: Immediate or control line

Format: iDLY, P_1

Explanation: P_1 specifies the number of milliseconds delay this directive provides. The maximum time delay allowed is 9.9999 seconds (namely, when $P_1 = 9999$). If P_1 is zero, a standard value of 35 ms (one disc revolution) is used. With a preceding iteration count, the delay is i (P_1) milliseconds.

---

[1] Typeout suppressed in not-true condition

[2] Typeout preceded by a "not" in a not-true condition

2.6.27 Directive: Jump to Location (JMP)

Function: To interface the control program with user-supplied subroutines.

Parameters, Trailing: P_1 (octal), memory address

P_2 through P_{10} (octal)

Parameters, Leading: Iteration counts (i) allowed

Mode: Immediate or control line

Format: iJMP, P_1, P_2 through P_{10}

Explanation: Parameter P_1 is the octal address of the user-supplied routine. This directive will execute a BRM instruction to this location. Parameters P_2 through P_{10} may be used by the subroutine as entrance parameters.

Two returns are provided. The normal return is a BRR instruction directed to the location specified in P_1. This return allows any iteration count specified to be executed.

If, however, the contents of the location specified by P_1 are increased by 1 (MIN) before the BRR instruction is executed, iteration is not allowed.

2.6.28 Directive: Mark Identify (MARK)

Function: To cause the word MARK XXXX to be typed at the console typewriter.

Parameters, Trailing: P_1 (four alphanumeric characters)

Parameters, Leading: Not allowed

Mode: Immediate or control line

Format: MARK, P_1

Explanation: P_1 specifies the four alphanumeric characters, XXXX, typed out on the typewriter. The typeout MARK XXXX is not suppressible.

2.6.29 Directive: Stop Identify (STOP)

Function: To cause a typeout STOP XXXX, and to cause the computer to halt. When the user clears the halt, the operation resumes at the next directive in the control line sequence.

Parameters, Trailing: P_1 (any four legal alphanumeric characters)

Parameters, Leading: None allowed

Format: STOP, P_1

Explanation: P_1 specifies the four alphanumeric characters in the typeout STOP XXXX; XXXX are these characters. This typeout is not suppressible by any BREAKPOINT switch.

SDS SCIENTIFIC DATA SYSTEMS

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

IDENTIFICATION: 900 Series Interrupt Arm-Disarm Feature Test Program

AUTHOR: A. Kliszewski

ACCEPTED: December 14, 1964

COMPUTER CONFIGURATION: Any 900 series computer with typewriter, 1 to 896 channels of system interrupts and the arm-disarm feature. Also required to perform the test is special module card #109745. When this card's input is connected to computer signal $\overline{Rti}$, any PIN command should set all armed interrupts.

PURPOSE: To check out, thoroughly, the operation of the arm-disarm feature.

PROGRAMMED OPERATORS: None

STORAGE: Octal locations 001, 023-027, 031, 033, 100, 200-3163.

TIMING: About one minute per set of 32 channels when all are present and operating correctly.

USE:

1.0 Loading

Normal FILL procedure. The program is available on an absolute binary paper tape.

2.0 Control

The program may be restarted by moving the RUN-IDLE-STEP switch to IDLE, pressing START, then STEPping once and RUNing. Also, Breakpoint Switch #1 will, when set, return control to the input stage (restart all tests), from any test. During tests #3 and 4, Breakpoint Switch #2 will, if set, cause control to be transferred to the next test. Starting with Breakpoint Switch #4 set will transfer control to the diagnostic loop.

USE: (Cont)

3.0 Input

3.1 As soon as the line "READY TO START TESTS..." is typed out, the computer is ready for input which consists of the interrupt group number N which is set into the A register, and M, the number of channels in the group to be tested, if M < 32, set into the B register, both in binary form. The groups consist of 32 interrupt channels each and are numbered sequentially 0-27 (0-33 octal).

3.2 Before proceeding with the tests, the special test cable must be connected to the Nth group of interrupts.

3.3 If the system tested comprises interrupt channels wired for single instruction execution, a simple modification to the program may be made from the console at this point. This consists of changing the BRM instruction in location numbered (177 + m) octal where m is the single instruction channel number (1 < m < 896) to a MIN (1777 + m) instruction. If m = 1, test #7 in the program will not work without some modification.

3.4 To start the tests, clear the HALT.

4.0 Output

4.1 N and M are typed out immediately after the HALT has been cleared.

4.2 If no failures are detected during the tests, the only subsequent type out is a series of TEST i COMPLETE, i = 1, 2, ..., 7.

4.3 Whenever a test has failed, typeout will identify the interrupt channel number (1 through 32) with the offending "COUNT" and the "PASS" number. The COUNT corresponds to the number of times the corresponding channel has been interrupted. The PASS number has meaning only in tests three and four which consist of M passes each.

METHOD: The program generates that coding which constitutes the linkage and service for the interrupt channels:

METHOD: (Cont)

Locations 200-1777 linkage
2000-2037 32 counters
2040-2177 32 counter incrementing interrupt service routines.

This is done immediately after loading.

The tests are as follows:

Test 1:
A. Disarm all 32 interrupts
B. Generate interrupts on all 32 channels
C. Check that all counters contain zero

Test 2:
A. Arm all 32 interrupts
B. Generate interrupts
C. Check that all counters contain ones

Test 3:
A. Selectively disarm ith channel
B. Generate interrupts
C. Check that counters 1 through i contain zeros, counters i + 1 through 32 contain ones
D. Clear counters
E. Do above for i = 1, 2, ..., 32 (i = pass number)

Test 4:
A. Selectively arm ith channel
B. Generate interrupts
C. Check that counters 1 through i contain ones and counters i + 1 through 32 contain zeros
D. Clear counters
E. Do above for i = 1, 2, ..., 32

Test 5:
A. Arm odd channels, disarm even ones (simultaneously)
B. Generate interrupts
C. Check that odd counters contain ones, even counters contain zeros

Test 6:
A. Same as Test 5, but odd and even channels have reverse roles

METHOD: (Cont)

Test 7: Modify the first interrupt service routine to disarm channels 2 -32

A. Arm all channels
B. Generate interrupts
C. Check that all counters contain ones (they should, since the disarm instruction was given after the interrupts attained waiting status)

The Diagnostic Loop: This is a loop consisting of AIR, POT (word selected by user), and PIN (to initiate interrupts).

The selected word is keyed into the A register when the program types = DIAGNOSTIC LOOP and halts.

Note: Counters are cleared at the beginning of all tests.

BPI SET FOR BINARY

SDS SCIENTIFIC DATA SYSTEMS MTØ - MT1.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

 Catalog No. 000017C*

IDENTIFICATION: Media Conversion Routine

AUTHOR: William Birney, SDS

ACCEPTED: April 29, 1965

COMPUTER CONFIGURATION: Any 4K SDS 910, 920, 925, or 930 with one or more peripheral devices attached to any buffer or channel and a console typewriter on the W buffer.

PURPOSE: To copy variable length records on binary or BCD cards, paper or magnetic tape, or typed input, to cards, paper or magnetic tape, typewriter or line printer.

PROGRAMMED OPERATORS: N/A

SUBROUTINES REQUIRED: Depending on the peripherals and channels involved in the conversion; one or more of the standard I/O handlers is required:

		Buffered	Extended	Peripheral
PTYIO	Cat. No.	020019	503001	Typewriter/Paper Tape
CDRP	"	030005	533002	Card Read/Punch
MTAPE	"	040004	540001	Magnetic Tape
PRINT	"	060005	563001	Line Printer

STORAGE: Relocatable; requires 1140 (02164_8) locations (not including I/O handlers), plus temporary record storage.

TIMING: N/A

SOURCE LANGUAGE: META-SYMBOL

*This version of the routine can be used to perform conversions involving peripherals attached to a channel which is restricted to interlaced I/O, in addition to channels permitting buffered I/O (W and Y). Also, three more breakpoint control options (see 2.3.11), and an error halt when writing is attempted on a file-protected tape, have been added. Binary records can now be listed on a typewriter or line printer.

LOADING PROCEDURE:

The I/O handlers needed must be loaded with the routine itself. The version (buffered or extended mode) of each handler used depends on which computer is being used and which channel the peripheral device involved is attached to. When using a 910 or 920 computer, the buffered handlers must be used, regardless of whether either buffer is interlaced. When using a 925 or 930 computer, the extended mode handlers must be used for peripherals on channels C through H, and may be used for peripherals attached to interlaced channels A and B.

For example, if the media conversion involves a line printer on Channel D, the extended mode version of PRINT must be loaded. If the conversion involves a paper tape reader on the W buffer (925 or 930 only), the buffered version of PTYIO must be loaded unless that buffer has interlace capability, in which case either version of PTYIO will suffice. If the conversion does not involve, for example, cards, then neither version of CDRP need be loaded. Note that only one version of a handler may be loaded, so that it is not possible to perform both interlaced and non-interlaced I/O with peripherals using the same handler. For example, a conversion may not involve a typewriter on channel E and a paper tape unit on the Y buffer or a non-interlaced channel B.

If Media Conversion is loaded by the Monarch loader, the (buffered) handlers will normally be loaded automatically from the Monarch Library. Therefore the versions of the handlers required for the conversion(s) to be performed should be in the Monarch Library.

If Media Conversion is loaded by the Universal Binary Loader (UBL) instead of the Monarch Loader, the versions of the handlers required for the conversion(s) may be loaded prior to loading Media Conversion or immediately afterward. In the latter case, the UBL will expect to find the handlers in the Library provided (for details, see the UBL Program Description).

USE:

1.0 MEDIA SPECIFICATION LIST

The program accepts an input list to the typewriter (unit 1, W buffer) and performs the specified conversion(s).

USE: (Cont.)

The list is of the form

$$D_1U_1C_1 - D_2U_2C_2, D_3U_3C_3, \ldots, D_nU_nC_n.$$

D_1 is input device MTØ - MT1.

U_1 is input unit number / CANCELS

C_1 is input channel SET BP1 FOR BINARY

D_i is output device IDLE, RUN AFTER EOF

U_i is output unit number $\Big\}$ $2 \le i \le n$ SET BP2 TO STOP ON READ ERROR

C_i is output channel

D_i is a two-letter word:

Input	Output
CR - card reader	card punch - CP
PR - paper tape reader	paper tape punch - PP
TY - typewriter	typewriter - TY
MT - magnetic tape	magnetic tape - MT
	line printer - LP

When D_i (device) specifies CR, CP, PR, PP or LP, then U_i (unit) must be 1 or 2. When D_i specifies TY, U_i must be 1, 2 or 3. In either case, if U_i is omitted, it is assumed to be 1. U_i is either 0, 1, ..., 6 or 7 if MT is used, and may not be omitted.

C_i, if present, is one of W or A, Y or B, C, D, E, F, G, H. C_i is assumed to be W if omitted.

Example of a legal list: MT1Y - CP2W, LP.

The program converts data from the form designated in $D_1U_1C_1$ to the forms specified successively in $D_2U_2C_2, \ldots,$ and $D_nU_nC_n$. Each output triplet $D_iU_iC_i$ $(i > 2)$ must be separated from the preceding one by a comma. The last must be followed by a period or a carriage return. Blanks in the list are ignored. Except for a slash (/), any character in the list which cannot logically follow the preceding ones causes an error message to be output to the console typewriter and cancellation of the entire list. A slash cancels the input list. In both cases the typewriter requests a new list.

USE: (Cont.)

Examples: (a) PR3W-MT2, LPY
(b) PRW-MT2, LPY

Example (a) will not cause any conversion to take place since only units 1 and 2 of the paper-tape reader can exist. A list-error message will be printed by the typewriter.

Example (b) will cause a variable length record or records on paper tape unit 1 to be transmitted by means of the W Buffer to the computer and then to be recorded on magnetic tape unit 2 (W Buffer) and printed on line printer unit 1 (Y Buffer).

2.0 TRANSMISSION

2.1 Transmission Mode

All input/output data is assumed to be BCD (decimal) unless breakpoint switch 1 is set, in which case media conversion will take place in binary mode. The switch is tested before each paper tape, magnetic tape or card input or output operation, and therefore should not normally be moved until the entire list is executed or a halt for EOF (see below) occurs.

All media conversion is done in four characters/word mode. When the number of characters input is not a multiple of four, the following will take place:

(a) Paper tape (BCD) or typewriter: the last word transmitted is padded with "60" codes. (On BCD paper tape or typewriter output, each "60" is converted to "12".)
(b) Paper tape (binary): "0" codes are used for padding.
(c) Magnetic tape: "0" codes are used for padding.

In BCD mode, "12" (blank) codes input from paper tape or typewriter are converted to "60" (ƀ) codes, and "77" (ǂǂ) codes are ignored. Each "60" is converted to a "12" on output from BCD paper tape or typewriter.

2.2 Transmission Errors

2.2.1 An error message is typed if any of the following errors occurs:

USE: (Cont.)

(a) Magnetic tape - character or longitudinal parity error which remains after 11 reread attempts, or after 6 write attempts (2 attempts each on 3 sections of tape).
(b) Typewriter, paper tape - parity error flip-flop set in buffer.
(c) Line printer error, or card reader or card punch error.

The message is "Halt due to error in transmission".

After the message is printed, a halt occurs. If the error occurs during output of a record, clearing the halt resumes conversion. If the error occurs during input, clearing the halt has one of two effects depending on the setting of breakpoint 2. If breakpoint 2 is reset, the input information is ignored (no output takes place), and a new record is input. If breakpoint 2 is set, conversion resumes and the contents of the record in the input buffer are used for output. Thus if a card reader error halt occurs, breakpoint 2 should be reset and the card causing the difficulty should be discarded or replaced in the hopper, before the halt is cleared.

2.2.2 "Buffer error during typewriter input of list" is printed if such an error occurs during list input. In this case, a new list is requested.

2.2.3 It is recommended that records read from paper tape or written on magnetic tape (except end-of-file) contain in excess of two words, to prevent malfunction.

2.3 Details of Transmission

Input:

2.3.1 Input from cards or magnetic tape: records are transmitted until EOF or end-of-tape is reached. When EOF or end-of-tape is reached, an appropriate message is typed and one of two actions is taken by the operator when the machine halts:

(a) The RUN switch is moved to IDLE, START button depressed, RUN switch moved to STEP, back to IDLE and to RUN, causing the program to request a new input list.
(b) The RUN switch is moved to IDLE and back to RUN, causing the program to continue transmitting records until the next EOF is reached.

USE: (Cont.) 2.3.2 Input from typewriter: records (separated by CR, i.e., carriage return) are transmitted until breakpoint 4 is set. Code delete ("77" or ⧻) characters are ignored.

2.3.4 Input from paper tape:

(a) BCD: As for magnetic tape; CR serves as EOR (end-of-record) and stop code plus gap as EOF. Records in a file should not be separated by gap. Code delete characters are ignored. Successive records are transmitted until EOF occurs or breakpoint 4 is set.

(b) Binary: As for magnetic tape; gap serves as EOR. Successive records are transmitted until breakpoint 4 is set, or until an end-of-library record is punched on paper tape or cards. (An end-of-library record consists of the octal word 017170000.)

Output:

2.3.5 Magnetic tape: If input source is paper tape (BCD) or typewriter, the CR at the end of each record is suppressed, and "12" codes are converted to "60" codes. When any output magnetic tape is at end-of-tape following completed transmission of a record, an end-of-tape message is typed and a halt occurs. Action is then taken as in 2.3.1.

2.3.6 Cards: If input source is paper tape (BCD) or typewriter, the CR at the end of each record is suppressed, and "12" codes are converted to "60" codes. If a BCD (binary) input record exceeds 20 (40) words in length, it is divided into 20 (40) word segments for output on successive cards.

2.3.7 Typewriter: A CR is inserted at the end of each record input from magnetic tape or cards. If breakpoint 1 is set, binary input is assumed, and the record is unpacked. The output is in effect an octal listing of the binary record. If breakpoint 1 is reset, BCD input is assumed, and trailing blank words (four "60" characters) are suppressed, and "60" codes are converted to "12" codes. If a record (BCD or unpacked binary) exceeds 21 words in length, it is divided into 21 word segments for output on successive lines.

USE: (Cont.) 2.3.8 Line printer: If input source is paper tape (BCD) or typewriter the CR at the end of each record is suppressed, and "12" codes are converted to "60" codes. If input was binary (breakpoint 1 set), the record is unpacked, and the output is in effect an octal listing of its contents. If an EOF occurs, the printer will upspace and print a blank line, which is to be disregarded. This enables the program to determine any error in printing the last line in the file.

2.3.9 Paper (BCD): A CR is inserted at the end of each record input from magnetic tape or cards. Trailing blank words (four "60" characters) are suppressed. Conversion of "60" codes to "12" codes takes place. A stop character is inserted after the last record in a file, and the first record in a file is preceded by gap.

2.3.10 Paper tape (binary): If breakpoint 3 is set, it is assumed that the input is in META-SYMBOL format. If input is from cards or magnetic tape, only the first logical record on each card or card image is output. The rest is deleted. (The length of a META-SYMBOL logical record is specified in bits 4-8 of the first word of the record.)

2.3.11 Caution: If a binary record input is listed on the typewriter or line printer, it is unpacked and therefore cannot subsequently be duplicated, for example, on magnetic tape.

e.g., 1. CR - LP, CP with breakpoint 1 set. The contents of each card read will be unpacked and listed on the printer, and this unpacked record will be punched on binary cards. Thus the binary cards will contain garbage.

e.g., 2. CR - CP, LP. with breakpoint 1 set. Each binary card will be duplicated, and then its contents will be unpacked and listed. Thus the conversion will be successful.

2.3.12 The accompanying tables show in detail the functions of the breakpoints and the mechanics of each possible conversion.

USE: (Cont.)

Breakpoint	Function when Reset	Function when Set
1	BCD input/output	Binary input/output
2	Delete output of record when input error occurs	Do not delete output of record when input error occurs
3	Normal mode	Delete data trailing each logical record on card or magnetic tape in META-SYMBOL format, when punching paper tape.
4	Normal mode	Halt before input of each record (paper tape or typewriter only).

	Device		End of Record		End of File		Line Length (Wds)	1	2	3	4
Mode	Input	Output	In	Out	In	Out					
BCD	MT	TY	gap	carr. ret.	EOF record		21	X			X
BIN	MT	TY	gap	CR	EOF record		21				
BIN	MT	PP	gap	gap	EOF record						
BCD	MT	PP	gap	CR	EOF record	⧻ & gap		X			X
BCD	MT	LP	gap	upspace	EOF record	blank line	33				
BIN	MT	LP	gap	upspace	EOF record	blank line	33				
BIN	MT	CP	gap	next card	EOF record		40				
BCD	MT	CP	gap	next card	EOF record		20				
either	MT	MT	gap	gap	EOF record	EOF rec.					
(BCD)	TY	MT	CR	gap					X	X	
(BCD)	TY	PP	CR	CR				X	X		X
(BCD)	TY	LP	CR	upspace			33		X	X	
(BCD)	TY	CP	CR	next card			20		X	X	
(BCD)	TY	TY	CR	CR			(21)	X	X		X
BIN	PR	MT	gap	gap							
BCD	PR	MT	CR	gap	⧻ & gap	EOF rec.			X	X	
BIN	PR	CP	gap	next card			40				
BCD	PR	CP	CR	next card	⧻ & gap		20		X	X	
BIN	PR	PP	gap	gap							
BCD	PR	PP	CR	CR	⧻ & gap	⧻ & gap		X	X		X
BCD	PR	LP	CR	upspace	⧻ & gap	blank line	33		X	X	
BIN	PR	LP	gap	upspace			33				
BCD	PR	TY	CR	CR	⧻ & gap		21	X	X		X

	Device		End of Record		End of File		Line Length (Wds)	1	2	3	4
Mode	Input	Output	In	Out	In	Out					
BIN	PR	TY	gap	CR			21				
either	CR	MT	end of card	gap	EOF sig.	EOF rec.					
BIN	CR	PP	end of card	gap	EOF sig.						
BCD	CR	PP	end of card	CR	EOF sig.	⧻ & gap		X			X
BIN	CR	CP	end of card	next card	EOF sig.		40				
BCD	CR	CP	end of card	next card	EOF sig.		20				
BCD	CR	TY	end of card	CR	EOF sig.		21	X			X
BIN	CR	TY	end of card	CR	EOF sig.		21				
BCD	CR	LP	end of card	upspace	EOF sig.	blank line	33				
BIN	CR	LP	end of card	upspace	EOF sig.	blank line	33				
BCD	X	TY, PP	blank record causes CR output								

EOF to LP causes upspace, blank line.

1 Trailing blank words (60606060) suppressed.

2 "77" codes deleted

3 "12" → "60".

4 "60" → "12".

XDS | Xerox Data Systems

BPT 2
A=25

XDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No: 851107-11A00
524001B

RESET ALL BPT
TYPE ØIRS

IDENTIFICATION: Extended Mode I/O Test Program

AUTHOR: England/Vorgitch, XDS

ACCEPTED: September 30, 1965

COMPUTER CONFIGURATION: Any 925/930 computer with a typewriter attached to the W channel and a paper tape punch and reader on any interlaced communication channel. The W channel need not be interlaced for the typewriter.

PURPOSE: To test as many of the extended I/O operations as possible with paper tape. Given a communication channel that is known to be good then the program serves as a paper tape tester.

PROGRAMMED OPERATORS: None

SUBROUTINES REQUIRED: None

STORAGE: The program occupies 843_{10} locations from 0177 to 01721 octal.

TIMING: N/A

SOURCE LANGUAGE: META-SYMBOL

LOADING PROCEDURE: The program is supplied on paper tape or cards with a self-loading bootstrap. Use the standard fill procedure.

USE: The program will load and address the typewriter for control. The program may be returned to the typewriter control at any time by using the START; STEP, RUN procedure.

USE: (Cont)

To select the reader and punch units to be used enter:

'CU' P to select punch and

'CU' R to select reader.

Where C is the channel number and may be any digits 0-7 and U is the unit, either 1 or 2. The reader and punch need not be on the same channel.

The test is started by entering the letter 'S.' Control of the test operation is then a function of the Breakpoint Switches. The following table summarizes the switch functions.

Switch	Reset	Set
1	Run in the normal mode as determined by the other switches.	Stop and return to keyboard control at the end of the current pass (either punch or read).
2	Continue to run the test selected by switch 3.	Cycle test runs from punch to read to punch, etc.
3	Selects punch mode.	Selects read mode.
	(Used when switch 2 is reset or when starting the test)	
4	Stop and type diagnostic messages whenever an error occurs.	Do not stop and type on errors but continue to run.

When running cyclic tests from punch to read, etc., the tape from the punch should be inserted into the reader.

METHOD:

Punching

The program punches 4 blocks of 64 characters each in one pass. The characters form a counting sequence from 00 to 77_8. The first block is started with leader and output with an IOSP. All punching is done in the one character per word mode. When the word count reaches zero an IOSD is loaded to punch a second block of 64 characters. No leader

METHOD: (Cont)

is punched between the first and second blocks. This results in one physical block 128 characters long. Starting with leader two additional blocks of 64 characters are then punched with an IOSD.

At the conclusion of each output operation the channel address register is stored and compared with the expected value. If they do not agree the program types the expected and actual values.

The program tests the channel during the output operation to see if the channel should erroneously disconnect before the word count reaches zero.

Reading

Each of the four blocks is read with a different set of commands and counts so as to test as many operations as possible. After reading a block a general subroutine checks for input parity errors, channel end address agreement with expected, and the data read character by character. Error messages with block numbers will be typed in the event of any one of these tests failing.

The handling of each block is described below:

Block 1

The first block is one half of a 128 character physical block. Reading 1 character per word an IOSD with a count of 64 is used to read this block. The program checks to see if the count reaches zero and the channel becomes inactive at the same time.

Block 2

This is the second half of the first physical block and is read with an IOSP with a count of 65. The read should terminate because of the end of record. The program checks to see that the word count does not reach zero and the channel remains active after the CIT (interrecord test) instruction skips. The tape is finally stopped with a disconnect before the data is checked.

METHOD: (Cont)

Block 3

Block 3 is a 64 character physical block. It is read with 2 channel commands. The first is an ISOP with a count of 32. If the count goes to zero before the channel disconnects then an IORP with a count of 33 is loaded. This should cause the interrecord indicator to be turned on at the end of the record. The count should not reach zero and the channel should remain active. The tape is again stopped with a disconnect before the data is checked.

Block 4

This is the third physical block of 64 characters and is read with an IORD with a count of 56. The program waits for the channel to be inactive then checks to see if the channel ignored the last eight characters. If the tape was erroneously stopped after the 56th character this will be visually discernible to the operator or will show up as a failure on the first block of the next read pass.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

 Catalog No. 012013C*

IDENTIFICATION:	Utility and Debug Package (AID)
AUTHOR:	J.A. Barnett, SDS
ACCEPTED:	October 12, 1965
COMPUTER CONFIGURATION:	Any 900 Series SDS computer with a console typewriter.
PURPOSE:	Provide various utility routine and debugging aids for the programmer's use during on-line program checkout.
PROGRAMMED OPERATORS:	None
SUBROUTINE REQUIRED:	None
STORAGE:	5030_8 - Relocatable; 0-116 during loading.
TIMING:	N/A
SOURCE LANGUAGE:	META-SYMBOL
LOADING PROCEDURE:	AID is available on either self-loading cards or paper tape.

1. Paper Tape:

   Paper tape copies of the package are loaded with Binary Input - Relocatable Paper Tape Loader, Catalog No. 000019. Use the standard FILL procedure. When the computer halts after reading a few inches of tape, place the address at which the package is to load in the A Register and clear the halt.

2. Cards:

   Card copies of the package are loaded with Binary Input - 2 Card Relocating Loader, Catalog No. 030007. To load the package into the 925/930, place the load address in the A Register and use the LOAD-FROM-CARDS switch. On the 910/920 place 03200002 in location 1, the load address

*Updated to provide DUMP output using the unbuffered Model 9372 Line Printer.

LOADING PROCEDURE (Cont.):

in the A Register, press START, 0203606 in the C Register and the RUN-IDLE-STEP switch in RUN.

When loading is complete the typewriter will be addressed for input of the first line. The loading process destroys the contents of locations 0 through 0116. The package itself occupies 5030_8 contiguous words starting at the location specified in the A Register.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

PART I

GENERAL INFORMATION

INTRODUCTION

AID is a package that embodies a group of commonly used utility programs and debugging aids. The package is intended as a helper for the programmer or engineer who is attempting to do on-line checkout of his program. Request for various functions are made by the user in the form of input lines. As an example, dumping of locations 100 through 200 would occur after the input of the following line: DUMP 100 TO 200. In general, an input line will consist of an action word (DUMP, LOAD, etc.) followed by a list of parameters. (See APPENDIX II for a Backus Normal Form description of the syntax of legal lines.)

Lines may be input from the typewriter, card reader of photo reader (see Assignment of the Control Media, page 9). An input line is terminated by one of the following conditions:

1. Encountering a carriage return (052 code '!'). If input is from cards, the remainder of the card is skipped; if input is from the photo reader, the reader is immediately disconnected. This allows the next character to be read without loss of data.

2. Reading 80 characters. Input is immediately disconnected and a carriage return is used as the 81st character (no data is lost on paper tape).

3. Encountering gap on paper tape.

4. End-of-file conditions on card reader causes control to return immediately to the typewriter.

5. If a channel error is encountered, the line being input is disregarded and control returns to the typewriter.

6. If a delete character (077 code '++') is input, the line being input is ignored. On cards the remainder of the card is skipped, on paper tape the reader is disconnected. Except with card input, the next line is assumed to start with the next character input. With cards, input starts with the next card. Use of the delete character is not considered an error, but a convenience for the user to inhibit error messages.

7. Null lines (i.e., those consisting of blanks and carriage returns) are permissable. No action will be taken; they may be used as desired to space input. In a like manner, blank cards are ignored.

After a line has been input, it is checked for correctness. If an error is found, a message is typed out. No processing occurs until the entire line has been checked. Therefore, when an error message is typed, it is only necessary to re-enter the offending line in a corrected form. (See On-line Translator description for exception to this rule.)

I/O DEVICES

Provisions are made for use of the following devices:

Mnemonic	Name	Comment
TY	typewriter	
PR	photo reader	
PP	paper tape punch	
CR	card reader	
CP	card punch	either buffered or unbuffered
LP	line printer	

The devices are all assumed to be unit 1 on the W buffer and are addressed in a non-interlaced fashion. (The channel and unit assignments are assembly parameters and thus may be easily altered to fit the requirements of a particular installation.)

FORMATS

All numerics entered are considered to be octal unless terminated by a decimal point (period).

Example:

$$100 = 100_8 = 64_{10}$$

$$100. = 144_8 = 100_{10}$$

Any numeric may be preceded by a + or - sign. If a numeric is used to specify a location it is computed, and its value mod 2^{14} is used.

In the subsequent description, the following conventions will be used:

num	a 24 bit numeric quantity
loc	a 14 bit address quantity

In several instances it is necessary to specify bounded regions of memory, i.e., DUMP 100 TO 200. The generic loc TO loc will be used to represent this type of entry. In all cases, the second location must be strictly greater than the first location; if not, an error message will be issued. A blank must separate the symbol 'TO' and both numerics enclosing it.

Certain sections of the package will accept Hollerith strings as input. A Hollerith string is a set of characters enclosed by single quote marks ('). The Hollerith information is packed 4 characters per word and left justifies in successively higher locations of memory. Blanks embedded in the Hollerith string are made into 060_8 codes.

Examples:

'A' = 21606060_8

'ƀƀƀA' = 60606021_8

'ABCDE' = 21222324_8 25606060_8

Caution: Neither a carriage return (052), delete character (077) nor a quote mark (014) are legitimate in a Hollerith field.

In all cases (other than the on-line translation or in Hollerith strings) blanks, commas, tabs and equal signs are treated identically. Further if one character of the four (say a blank) is sufficient to delimit a field, then several will work equally well. In other words multiple blanks, etc., are not significant when used to delimit a field.

Examples:

A, B $\equiv$ A, ƀ, = B

A=B $\equiv$ A, B $\equiv$ A ƀ B

however

12. $\not\equiv$ 12,.   A = B $\not\equiv$ AB

i.e., delimiters may not be embedded in a field.

COMMUNICATIONS LINKAGE

After loading is complete, a BRM to the package is put in cell 2. The user's program may communicate with the package via a BRM *2. This will generate the necessary return linkage between the package and the user's program. The package may also be reached by placing a BRU 2 in the C Register. Further, the package may be entered by executing a BRU to the location used as the load address. If this latter method of entry is used, disregard the next paragraph.

Upon entry to the package the contents of all registers, the status of overflow indication and the conditions of the interrupt system (enabled, disabled) are saved. The W buffer status is then checked. If it is active, 1 second of time is counted off. After this, the buffer is unconditionally disconnected and the interrupt system disabled. If the buffereoriginally tested ready, then the interrupt system is immediately disabled. The stored values of the registers may be displayed or altered (see TYPE and STORE directive descriptions). To return control to the user's program, follow the procedure described as the GO directive. The package does not make use of any POP's.

PACKAGE TABLE AREAS

When either the INSERT or SNAPSHOT directive is used, an entry is made in the ALTER-POINTS table. This is a table which is internal to the package and allows for twenty (20) such entries to be made. When an insert or snapshot is removed, the table entry assigned to it is opened for another entry to be made. The table overflow error message is issued when an attempt is made to have more than 20 inserts and snapshots in core simultaneously.

As snapshots and inserts are entered, information pertinent to them is put in the Package Table area. The current location of this area may be displayed using the TYPE directive or altered using the STORE directive. It is the user's responsibility to make sure he doesen't destroy his own program. If the ORG directive is used when entering an insert, the next available Package Table Locations is located somewhat after the ORGed location. See description of the REMOVE, INSERT, and SNAPSHOT directive for further information.

PART II

OPTIONS

ASSIGNMENT OF THE CONTROL MEDIA

Input lines are read from the control media. Upon the completion of loading and after the occurrence of any error, the typewriter is assigned as the control media. To assign the control media, one of the following lines should be entered:

C = TY	Read the next and subsequent lines from the typewriter
C = PR	Read the next and subsequent lines from the photo reader
C = CR	Read the next and subsequent lines from the card reader

ASSIGNMENT OF PERIPHERAL DEVICES

Several of the options provided with the package will operate on more than one I/O device. The selection of the device is made by the user when a line of the following form is entered:

ASSIGN F_1, F_2 . . .

where F_i is one of the following:

BI = PR	subsequent loading takes place from the photo reader
BI = CR	subsequent loading takes place from the card reader
BO = PP	subsequent binary output is on the paper tape punch
BO = CP	subsequent binary output is on the card punch
LO = TY	subsequent dumps are made on the typewriter
LO = LP	subsequent dumps are made on the line printer

Example:

ASSIGN LO = LP, BI = PR

The above line causes subsequent dumps to occur on the line printer and subsequent loading to take place from the photo reader.

Point of Interest

After loading, the following assignments are automatically made and no ASSIGN line need be entered unless a different option is desired.

ASSIGN BI = PR, BO = PP, LO = TY

RESUME COMPUTATION DIRECTIVE

To restore the registers and indicators to their stored values, one of the following types of lines should be entered:

GO

GO loc

After the registers and indicators have been restored, computation will resume at the location specified by the stored value of the P register. If a numeric is used (as on the second line above), its value will be used instead as the transfer address.

Example:

GO 200

The above line will restore the A, B and X register, the overflow indicator and interrupt system, and then transfer to location 200.

LOAD DIRECTIVE

The package provides a resident loader for use with either absolute or relocatable programs that contain no POP's or External References or Definitions. The decks are input from the device specified as BI on the last ASSIGN line. To initiate loading, a line of the following type should be entered:

LOAD loc, GO

If the numeric (loc) is entered, it is used as the relocation bias. If not, the ending address + 1 of the last relocatable deck entered is used. If the symbol GO is entered, decks will be loaded until an end transfer card is input; if not, upon encountering any end card, control is immediately returned to user input. In any case, if an end transfer record is encountered during loading, the transfer address replaces the stored value of the P register and control returns to user input, i.e., the transfer is not taken until the GO directive is used.

Examples:

LOAD

Load one deck and return control to user input.

LOAD 200

Load one deck and use 200 as the relocation bias.

LOAD GO

Load until an end transfer record is encountered.

LOAD 200, GO

Use 200 as the relocation bias and load decks until an end transfer record is met.

Point of Interest

Encountering an End of File on the card reader immediately terminates loading. The relocation bias is ignored when loading absolute decks.

DUMP DIRECTIVE

Selected portions of memory may be dumped in either octal or instruction format. Instruction format is a line containing the octal contents of 8 locations with the mnemonic op-code printed directly beneath the numeric. The output device is the one specified as LO in the previous ASSIGN line. In all cases, if an output line would be all 0's then it is not output.

To produce an octal dump, a line of one of the following types should be entered:

DUMP loc TO loc, loc TO loc, ...

or

DUMPO loc TO loc, loc TO loc, ...

To produce an instruction dump, a line of the following type should be entered:

DUMPI loc TO loc, loc TO loc, ...

Example:

DUMPI 1000 TO 3777, 500 TO 577

Will cause locations 1000 through 3777 and locations 500 through 577 to be dumped in instruction format.

Point of Interest

No matter what locations are given for a dump, the beginning address will be a multiple of 8 and the ending address will be a multiple of 8 plus 7, i.e., DUMP 5002 to 5010 will dump locations 5000 through 5017 inclusive.

BINARY PUNCH DIRECTIVE

Selected portions of memory and end transfer records may be punched out in the standard, absolute, binary format. The output will occur on the device specified as BO in the previous ASSIGN line. A line of the following type will cause this action:

PUNCH $F_1, F_2, \ldots$

Where F_i is one of the following

loc to loc	Punches out all locations between the specified addresses.
loc	Punches out an end transfer record to the specified location. If this does not appear in the PUNCH statement an end record without a transfer address is punched. Further, a field of this type, if used, must be the last one entered.

Examples:

PUNCH 107 TO 210, 300 TO 577, 120

The above line would cause location 107 through 210 inclusive, location 300 through 577 inclusive and an end transfer record to location 120 to be punched on the device specified as BO.

PUNCH 107 TO 210, 300 TO 577

Same as the above example except an end record without a transfer address is punched.

PUNCH BOOTSTRAP DIRECTIVE

A bootstrap to load programs in the standard, absolute, binary format from paper tape may be punched out on paper tape A line of the following form will cause this action:

BOOT

The above line will output the bootstrap on the paper tape punch. See Appendix IV for information on how to use the bootstrap.

ALTER DIRECTIVE

The stored value of the registers and selected areas of memory may be set to values specified by the user. A line of the following type will cause this action:

STORE F_1, F_2, F_3, . . .

where F_i is one of the following

A = num	the numeric replaces the stored value of the A register
B = num	the numeric replaces the stored value of the B register
P = loc	the numeric replaces the stored value of the P register
X = num	the numeric replaces the stored value of the X register
I = 1 or 0	if the numeric equals 1, the stored interrupt flag is set to enable; the flag is set to disable if the numeric is 0
O = 1 or 0	if the numeric is 1, the stored overflow indicator is turned on; if 0, the indicator is turned off

T = loc	the numeric becomes the value used as the Package Table Area location
loc = num	the numeric replaces the contents of the specified location
loc TO loc = num	the numeric replaces the contents of all memory locations specified between the two loc's

Example:

```
STORE  200 = 5, A = -1,  1000 TO 1400 = 0
```

The above line sets the stored value of the A register to -1, location 200 to 5 and zeros cells 1000 through 1400 inclusive.

DISPLAY DIRECTIVE

The stored values of the registers along with selected areas of memory may be output on the typewriter. A line of the following form will cause this action:

TYPE F_1, F_2, F_3, . . .

where F_i is one of the following

P	type out the value of the stored P register
A	type out the value of the stored A register
B	type out the value of the stored B register
X	type out the value of the stored X register
I	type 1 if the stored interrupt indicator is enabled, 0 if disabled
O	type 1 if the stored overflow indicator is on, 0 if off

ALL	type out the stored value of all the registers and indicators
T	type out the Package Table Area location
loc	type out the contents of the location specified by this number
loc TO loc	type out the contents of all locations which lie between the first and second locations
ALT	type out the locations at which there is either a snapshot or insert

When memory locations are specified, the op-code will be typed along with the octal contents.

Example:

TYPE 200, A, B, P, 300 TO 310

The above line types out the stored values of the A, B and P registers, contents of location 200 and contents of location 300 through 310 inclusive.

DISPLAY IN BCD

Selected portions of memory may be typed out in BCD. A line of the following form causes this action:

TBCD loc TO loc, etc.

The locations specified are typed out, not more than 20 words per line. If several areas are specified, the typewriter is double spaced between them.

Example:

TBCD 105 TO 114, 306 TO 340

If the above line were entered, locations 105 through 114 and locations 306 through 340 inclusive would be output on the typewriter in BCD.

SEARCH DIRECTIVE

The package makes provisions for searching specified areas of core and typing out locations within those areas whose contents lie in a particular value range.

Input of a line in one of the following forms causes this action:

IN loc TO loc, F1, F2

The area searched is specified as 'loc TO loc'.

F1 is one of the following:

F1 = num — look for all locations in the specified area whose contents are C = num

F1 = num_1 TO num_2 — look for all locations in the specified area whose contents, C, satisfy $num_1 \leq C \leq num_2$

F_2, either a numeric or nothing, is entered. If no F_2 is given, it is assumed to be 77777777. F2 is used to mask the contents of the location as well as the value ranges before comparisons are made. As an example:

IN 100 TO 200, 1000, 37777

If the above line were input, locations 100 through 200 inclusive would be searched for all locations whose address portion equalled 1000.

OUT loc TO loc, F1, F2

See above for use of loc TO loc and F2.

F1 is one of the following:

F1 = num — look for all locations in the specified area whose contents, C, $\neq$ num.

F1 = num_1 TO num_2 — look for all locations in the specified area whose contents, C, satisfy $C < num_1$ or $C > num_2$

OUT 200 TO 300, 0 TO 100000, 7700000

If the above line were input, locations 200 through 300 inclusive would be searched for all locations whose OP codes were neither HLT's nor BRU's.

ON-LINE TRANSLATOR DIRECTIVE

An on-line translator is provided in the package. In contradistinction to other operations in thepackage, commas, equal signs, blanks and tabs are not interchangeable. Blanks are disregarded as before but commas are significant. Therefore, it is recommended that lines be punctuated exactly as described below.

To initiate on-line translation, a line of the following type should be entered:

ORG loc

The location specified on the ORG line is the location at which the subsequent instruction is to be entered. As each line is entered the present value of the location counter is incremented and typed out (if input is from the typewriter).

If a line of the following type is entered:

ORG

Then the last value of the location counter is used. This is a convenience to expedite error recovery with the On-Line Translator.

The following directives are recognized:

1.	RES num	the numeric in the address field is added to the location counter.
2.	ORG loc	the location counter is set to the specified address.
3.	DAT F_1, F_2, F_3	where F_i is either a numeric or a Hollerith string. The numbers and Hollerith information are put into successively higher locations of memory and the location counter is incremented accordingly.

4. END or END loc — when an END directive is encountered, control is transferred back to the main body of the AID Package. If a location is specified, it replaces the stored value of the P register.

5. See Appendix 1 for the list of op-codes and their octal equivalent. With the mnemonics, relative and indirect addressing are done as in META-SYMBOL, e.g., $±num means the present value of the location counter ± the numeric. *num or *$ means that this instruction is indirectly addressed.

Example:

	ORG	1000			
1000	LDX	1100		1001	07101100
1001	LDA	1101		1001	07601101
1002	BRU	$+2	would generate	1002	00101004
1003	ADD	1102		1003	05501102
1004	STA	1200, 2		1004	23501200
1005	BRX	$-2		1005	04101003
	ORG	1100			
1100	DAT	-30, 0, 1		1100	77777750
				1101	00000000
				1102	00000001
	ORG	2000			
2000	DAT	'THIS IS A TEST', -4		2000	63343162
				2001	60316260
				2002	21606325
				2003	62636060
				2004	77777774
2005	EOM	2641		2005	00202641
2006	LDX	2005		2006	07102005
2007	MIW	2005, 2		2007	21202005
2010	BRX	$-1		2010	04102007
2011	EOM	14000		2011	00214000
2012	LDA	*100.		2012	07640144
2013	STA	*$+1		2013	03542013
2014	HLT	1		2014	00000001
	END	2005		PREG = 2005	

INSERT DIRECTIVE

The package makes provision for patching a program by 'inserting' a sequence of code as described below. A line of the following type will cause this action:

```
INSERT    loc
```

Example:

```
location      code

1000          LDA     100       memory before patch
1001          STA     101
1002          SKN     102
1003          BRU     $+2
```

these lines are input:

```
INSERT        1002
              BRU     $+2
              DAT     1
              ADD     $-1
              STA     101
              MIN     102
              END
```

the following is the resultant code in memory

```
1000          LDA     100
1001          STA     101
1002          BRU     T
1003          BRU     $+2
 .
 .
 .
T             BRU     T+2        T = first available locations in Package
                                     Table Area (see page 8)
T+1           DAT     1
T+2           ADD     T+1        inserted instructions
T+3           STA     101
T+4           MIN     102
T+5           SKN     102        original instruction
T+6           BRU     1003       linkage back to original program
T+7           BRU     1004       linkage back to original program
```

This example illustrates several important features of the insert action.

1. The insertions are placed in the first and successively higher Package Table Areas locations.

2. Relative addressing refers to the INSERT area.

3. The inserted instructions are executed before the original instruction.

4. An insertion may be made at any location including a transfer or skip. Note that two BRU's are placed after the original instruction.

5. The on-line translator is used.

6. If an ORG directive is used.

   a. The user must provide the linkage (BRU) to the ORGed region himself.

   b. The two BRUs back to the program will be generated immediately after the last instruction entered.

   c. The locations after the second BRU will now be considered to be the first available locations for the next snapshot or insert. This is true even if ORG is the first instruction of the patch.

7. If an error occurs during the translator input, the original instruction will not be disturbed and no entry will be made in the Alter Points Table. Therefore, there is no need to reset the insert. If it is desired to continue making the insert, type I followed by a carriage return. This will save re-entering all instructions entered before the error.

See the description of the RESET directive for further information, also the section on use of the On-Line Translator.

SNAPSHOT DIRECTIVE

Dynamic snapshots can be made from any place in memory. To initiate this action a line of the following type should be entered.

    SNAP   loc, num, loc TO loc, loc TO loc, etc.

The first numeric specifies the location at which the snapshot is to take place. The second numeric specifies the number of times that the snapshot is to be done. The bounded areas are dumped on the typewriter.

The snapshot is made by replacing the original instruction with a BRU into the Package Table Area. The following sequence of events then takes place each time program control passes to the BRU:

1. The present values of the registers are saved.
2. The stored register values are typed out.
3. The specified regions of core are dumped on the typewriter.
4. The saved values of the registers are copied back to the register.
5. The original instruction is executed.
6. Control is transferred back to the user program.

The above steps are repeated each time that program control is transferred to the snapshot location. Upon entry a life counter is decremented. When the snapshot has been taken a number of times that equals the specified life count, the original instruction is put back in line and no further snapshots will be made at this location. Further, the Alter Points entry is erased. Program control is then transferred to the snapshot location which now contains the original instruction.

Example:

```
SNAP    300, 2, 360  TO  377
```

The above line will cause the registers to be typed along with a dump of locations 360 through 377 the next two times that program control passes to location 300.

Point of Interest

Read the description of the INSERT and REMOVE directives for further information on the handling of table space.

RESET DIRECTIVE

To remove inserts or snapshots from core a line of the following type should be entered:

REMOVE F_1, F_2, ...

where F_i is one of the following

S remove all snapshot entries

I remove all insertions

loc remove the snapshot or insert that was made at this location

The corresponding Alter Point Entries are removed and made available for other uses and the original instruction is placed back in line.

Caution: Because of the way that snapshots and inserts are performed and removed, it is not advisable to have more than one of them active at any particular location.

APPENDIX I

OP CODES AND THEIR OCTAL EQUIVALENTS

OP Code	Octal
HLT	0
BRU	1
EOM	2
EOD	6
MIY	10
BPO	11
MIN	12
POT	13
ETR	14
MRG	16
EOR	17
NOP	20
EXU	23
YIM	30
BPI	31
WIM	32
PIN	33
STA	35
STB	36
STX	37
SKS	40
BRX	41
BRM	43
RCH	46
SKE	50
BRR	51
SKB	52
SKN	53
SUB	54
ADD	55
SUC	56
ADC	57
SKR	60
MDE	60
MIN	61

APPENDIX I (Continued)

OP Code	Octal
XMA	62
ADM	63
MUL	64
MUS	64
DIV	65
DIS	65
SKM	70
LDX	71
SKA	72
SKG	73
SKD	74
LDB	75
LDA	76
EAX	77

APPENDIX I (Continued)

Shift Mnemonic	Octal
RSH	06600000
RCY	06620000
LRSH	06624000
LSH	06700000
LCY	06720000
NOD	06710000

Register Change Mnemonics

CLA	04600001
CLB	04600002
CLR	04630003
CAB	04600004
CBA	04600010
XAB	04600014
BAC	04610012
ABC	04620005
CLX	24600000
CXA	04600200
CAX	04600400
XXA	04600600
CBX	04600020
CXB	04600040
XXB	04600060
STE	04600122
LDE	04600140
XEE	04600160
CNA	04601000

Alternate Forms of the Directive

E may be used instead of END
O may be used instead of ORG

APPENDIX II

BACKUS NORMAL FORM DESCRIPTIONS OF LEGAL INPUT LINES

```
<NULL>        :=
<ALPHA>       :=   A/B/C/D/---------/X/Y/Z
<NUMBER>      :=   0/1/2/3/4/5/6/7/8/9
<DECMAL>      :=   .
<PLUS>        :=   +
<MINUS>       :=   -
<HOLRTH>      :=   '
<DONE>        :=   !
<DLMIN>       :=   $/)/*/Δ/(
<DLMOUT>      :=   / /=/,/≠
<ILEGAL>      :=   √/:/>/?/[/</ /1/;///ო/ǂ

<SIGN>        :=   <PLUS>/<MINUS>
<NUMERIC>     :=   <NUMBER>/<NUMBER> <NUMERIC>
<DEC>         :=   <NUMERIC> <DECMAL>
<BCD>         :=   <DLM>/<SIGN>/<ILEGAL>/<NUMBER>/<DECMAL>
                   /<ALPHA>/<BCD> <BCD>

<SYM>         :=   <ALPHA>/<ALPHA> <SYM>
<DLM>         :=   <DLMIN>/<DLMOUT>
<NUM>         :=   <NUMERIC>/<DEC>/<SIGN> <NUMERIC>
                   /<SIGN> <DEC>
<HOL>         :=   <HOLRTH> <BCD> <HOLRTH>
<TRM>         :=   <DONE>

<TERM>        :=   <NULL>/<NUM>/<SYM>/<HOL>
<STRING>      :=   <TERM>/<TERM> <DLM> <STRING>
<LINE>        :=   <STRING> <TRM>
```

APPENDIX III

SUMMARY OF LINE FORMATS

$$\text{C} = \left\{\begin{array}{c}\text{TY}\\ \text{PR}\\ \text{CR}\end{array}\right\}$$

$$\text{ASSIGN BI} = \left\{\begin{array}{c}\text{PR}\\ \text{CR}\end{array}\right\},\quad \text{BO} = \left\{\begin{array}{c}\text{PP}\\ \text{CP}\end{array}\right\},\quad \text{LO} = \left\{\begin{array}{c}\text{TY}\\ \text{LP}\end{array}\right\}$$

$$\text{GO}\ \left\{\text{LOC}\right\}$$

$$\text{LOAD}\ \left\{\text{RELOCATION BIAS}\right\},\ \left\{\text{GO}\right\}$$

$$\left\{\begin{array}{l}\text{DUMP}\\ \text{DUMPO}\\ \text{DUMPI}\end{array}\right\}\ \text{loc TO loc, ETC}$$

$$\text{PUNCH}\ \left\{\text{loc TO loc}\right\},\ \left\{\text{TRANSFER ADDRESS}\right\},\ \text{ETC}$$

$$\text{STORE}\ \left\{\left\{\begin{array}{c}\text{T}\\ \text{A}\\ \text{B}\\ \text{P}\\ \text{X}\\ \text{I}\\ \text{O}\end{array}\right\} = \text{num}\right\},\ \left\{\text{loc} = \text{num}\right\},\ \left\{\text{loc TO loc} = \text{num}\right\},\ \text{ETC}$$

$$\text{TYPE}\ \left\{\begin{array}{c}\text{T}\\ \text{A}\\ \text{B}\\ \text{X}\\ \text{P}\\ \text{I}\\ \text{O}\\ \text{ALL}\\ \text{ALT}\end{array}\right\},\ \left\{\text{loc}\right\},\ \left\{\text{loc TO loc}\right\},\ \text{ETC}$$

$$\text{ORG}\ \left\{\text{loc}\right\}$$

APPENDIX III (Continued)

Op Code $\left\{ * \right\} \left\{ \begin{array}{l} \$\pm \text{num} \\ \text{loc} \end{array} \right\} \quad \left\{ , \text{num} \right\}$

INSERT loc

I

SNAP loc, life, $\left\{ \text{loc TO loc} \right\}$, ETC

REMOVE $\left\{ \text{loc} \right\}$, $\left\{ \text{S,} \right\}$, $\left\{ \text{I} \right\}$

$\left\{ \begin{array}{c} \text{IN} \\ \text{OUT} \end{array} \right\}$ loc TO loc , $\left\{ \begin{array}{c} \text{num} \\ \text{num To num} \end{array} \right\}$, $\left\{ \text{mask} \right\}$

BOOT

TBCD loc TO loc , ETC

APPENDIX IV

USE OF PAPER TAPE BOOTSTRAP OUTPUT BY THE BOOT DIRECTIVE

The bootstrap occupies locations 0, $2\text{-}53_8$ and is used to load programs in the standard absolute binary format. The bootstrap is loaded with the FILL procedure. The breakpoints are interrogated during loading and cause the following action:

BP2

Breakpoint 2 set will cause the bootstrap to halt after it has loaded itself. This option is useful when the bootstrap and the object program are on physically different tapes. Clear the halt to continue loading.

BP1

Breakpoint 1 set will cause the bootstrap to halt each time that an end record is encountered. If the end record specifies a transfer address, the transfer will be taken when the halt is cleared. If no transfer address is on the end record, the bootstrap will attempt to load the next program when the halt is cleared.

HALTS

P = 00007	BP 2 Halt
P = 00014	Buffer Error Halt
P = 00043	Checksum Error
P = 0005Ø	BP1 Halt

The bootstrap will not load programs that contain POP definitions or external references and definitions.

IDENTIFICATION: Card Reader Test Program

AUTHOR: S. Vorgitch, SDS

ACCEPTED: May 26, 1964

COMPUTER CONFIGURATION: Any SDS 920 or SDS 910 with a typewriter, and SDS Model 9151 card reader.

PURPOSE: To provide an acceptance test for the SDS 9151 card reader.

PROGRAMMED OPERATORS: None

STORAGE: Octal locations 200-613 (268 words).

TIMING: Not applicable.

USE: The Card Reader Test Program consists of a self-loading paper tape and a special 64-card test deck. The card deck is sequentially numbered and must be in correct order to run the test.

To perform the card reader test, proceed as follows:

1. Load the program by normal Fill procedure. When the program is loaded the computer will halt.

2. Check the test deck for correct sequencing of cards.

3. Select either the binary or Hollerith test by means of Breakpoint Switch 1.

   BP 1 set: Hollerith read test

   BP 1 reset: Binary read test

4. Load the test deck and turn on the EOF ON indicator.

5. Clear the computer halt to start the test.

Normal Run

If the test deck reads through successfully one of the following messages will be typed out, depending on the mode selected: "Binary test complete" or "Hollerith

USE: (cont.)

test complete". The computer will then halt. At this time the test deck may be reloaded and the program repeated by clearing the halt.

Note: During the Hollerith read test, the VALIDITY CHECK light will be on continuously.

Error Indication

If an error occurs during the test, a message will be typed and the computer will halt. If an error halt occurs, the entire program must be restarted.

The following error messages are possible:

1. BIN check error:

   The binary check character (52522525) read from columns 1 and 2 of the last card is not correct. The A register contains the pattern read from the card while the B register contains the value which should have been read.

2. HOL check error:

   Either the Hollerith character (T) read from column 3 is not correct or a validity check has occurred while reading it. The A register contains the binary-coded value of the Hollerith character read from the card while the B register contains the value which should have been read.

3. IDN check error:

   The identification number read from the last card does not sequentially follow that of the preceding card. The A register contains the sequence number read from the card while the B register contains the expected sequence number.

4. Error in column N: ($8 \leq N \geq 71$)

   The information read from the particular card column is not correct. For the binary test, the A register contains the binary pattern read from the card, while the B register contains the pattern which should have been read. For the Hollerith test, the A register contains the Hollerith value read from the card, while the B register contains the binary pattern that should be in that card column.

USE: (cont.)

5. Error in column M: (M = 80* or 81*)

   On each odd numbered card (columns 4 and 5) columns 72 through 80 are skipped using EOM 12006. Even numbered cards are read until an end of record interrupt (I2) is received. If the I2 occurs before column 80 or an I1 is received after column 80, the appropriate error note is printed.

6. Signal not present:

   At certain places, the program tests for the presence of the following signals:

   Card Reader ready to feed (SKS 12006)

   Card Reader ready to read one column (SKS 14006)

   The program will loop before continuing until the particular signal is received. Upon terminating the test, the Card Reader end-of-file signal (SKS 11006) is tested.

METHOD:

Each card in the test deck contains the following fields:

1. Binary check character (columns 1 and 2).
2. Hollerith check character (column 3).
3. Identification number (columns 4 and 5).
4. Test data (columns 8 through 71).

Columns 6, 7, and 72-80 are blank.

The test data contains every possible columnwise combination of punches arranged in ascending binary order. Column binary information is read from top to bottom, where the 12-row is the most significant bit and the 9-row is the least significant bit.

Each field on the card is read and interpreted for correctness as follows:

1. Columns 1 and 2 are read in the binary mode and form a check character whose octal value is 52522525.
2. Column 3 is read in the Hollerith mode and is the letter "T". The octal value is 63.

METHOD: (cont.)

3. Columns 4 and 5 are read in Hollerith and converted to binary by the program to form the card identification number. This number is then checked for correct sequence.

4. Binary test: Columns 8 through 71 are read in the binary mode, one column at a time. Each column is checked against an internal binary counter. Between columns, the W buffer is disconnected and the program waits for Card Reader ready (SKS 12006) before reading the next column.

   Hollerith test: Columns 8 through 71 are read in the Hollerith mode, one column at a time. The octal value read from a particular column is used to select a binary pattern from a table. This value is compared with the internal binary counter pattern. The two patterns should match for non-valid Hollerith characters. The validity error signal should be set for non-valid Hollerith characters.

XDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

IDENTIFICATION: Memory Lock-out and Power Fail-safe Test

AUTHOR: S. J. Vorgitch, Xerox Data Systems

ACCEPTED: December 2, 1965

COMPUTER CONFIGURATION: Any XDS 930 with Memory lock-out (manual or program controlled) / Power fail-safe

PURPOSE: To verify the operation of the memory lock-out/Power fail-safe options

PROGRAMMED OPERATORS: None

SUBROUTINES REQUIRED: None

STORAGE: 369 locations from 0100_8 to 0660_8

TIMING: N/A

SOURCE LANGUAGE: META-SYMBOL

LOADING PROCEDURE: The program is supplied on self loading cards or paper tape. The Memory lock-out register must have been previously reset (start button or reset manual lock-out switches). After loading without error, the program will half with 0107_8 in the P register. At this time, the A register can be initialized with a starting random number and the Breakpoint Switches may be set. Location one is used for program recovery.

USE: (Cont)

Breakpoint Switches have the following meaning:

1.0 SWITCHES

SWITCHES	SET	RESET
1	Do not use random numbers to fill memory. Use A register input at loading halt ($p=0107_8$)	Use random numbers to fill memory
2	This machine has the Memory lock-out option.	This machine does not have the Memory lock-out option.
3	Perform Memory lock-out test.	Perform Power fail-safe test.
4	Memory lock-out is manual.	Memory lock-out is under program control.

2.0 ERROR INDICATION

For Memory lock-out (programmed and manual) the typewriter output is the only verification of proper operation.

For Power-fail save descrepancies are indicated by a type-out of the memory address that did not compare after power up. The program will then halt with P = 0572. The A register contains the data expected and the B register contains the data read out of memory. Clear the halt to continue comparing.

METHOD:

After loading, memory size is computed. If required (B. P. 4 set) a note is printed requesting the operator to CLEAR ALL MANUAL SWITCHES. Location 37_8 is then loaded with a BRM to a power off subroutine and the program will halt to accept a random number seed and allow Breakpoint setting. After the idle-run sequence memory is filled with random number from 01000_8 to the end of memory. These numbers are then compared and B. P. 3 is tested. If reset, the sequence of filling and comparing is cycled until power off occurs. If B. P. 3 is set, a mask is generated with one bits corresponding to the maximum number of memory lock-out areas according to the memory size.

USE: (Cont)

If lock-out is manual (B.P. 4 set) the actual locked-out areas will be typed as the program attempts to write (STA) into all of memory. The switches can be changed at any time. Locations $0\text{-}511_{10}$ (manual memory lock-out switch 1) should never be locked out.

In both the power fail-safe and the memory lock-out programs, B.P. 3 is interrogated. This permits going back and forth between the test programs.

METHOD:

Each functional routine is essentially independent of the others and attempts to accomplish its operation in as straight forward a manner as possible. The program attempts to prevent the tape from running off the reels in either direction. By using the write end of file operation, sections of the tape can be set off for further test without having to work from the beginning of the reel on every pass.

At the start of every routine, the status of the tape is tested and if the operation obviously cannot be accomplished the program does not attempt it. No reverse operation will be started if the tape is at the load point. No forward operation except write end of file will be started if the tape is on the end of reel marker. No write operation is attempted if the file protect is on.

All reading, spacing and searching operations will be unconditionally terminated when an end of file is encountered.

IDENTIFICATION: 42 KC Magnetic Tape Test Program, W Buffer

AUTHOR: A. W. England, SDS

ACCEPTED: May 4, 1964

COMPUTER CONFIGURATION: Any SDS 900 series with a typewriter and one or more magnetic tape units of any type except 9145 attached to the W buffer. The buffer must be interlaced.

PURPOSE: To provide a simple and easy means for initial checkout and testing of 42 KC magnetic tape units.

PROGRAMMED OPERATORS: None.

STORAGE: The program occupies 587 words from 400_8 to 1512_8. It uses the HELP Word Output Subroutine located at 200_8. The area from the end of the program to the end of memory may be used as input and interlaced output record image.

TIMING: The program is sufficiently fast to keep the tape operating at full speed for all operations.

USE:

1.0 LOADING

To load the program, insert the paper tape in the paper tape reader and follow the normal FILL procedure. When it is loaded, the light on the typewriter will indicate that the program is now under operator control.

1.1 REGAINING OPERATOR CONTROL

If at anytime the operator should lose control of the program he may return it to the keyboard mode by moving the RUN-IDLE-STEP switch to IDLE, pressing START, moving the switch to STEP and then to RUN. If for some reason location 0001 is destroyed he may execute a BRU to location 400 to return control to the keyboard.

2.0 CONTROL FUNCTIONS

The following list contains a call letter for the various control functions the program will perform. These may be typed anytime the light on the typewriter is lit.

USE: (cont.) 2.1 PARAMETER CONTROL

These functions cause the program to set up the various parameters for the tests which will follow.

2.1.1 P, Set Pattern

The previously typed 8 octal digits are set up as the pattern for writing operations.

2.1.2 L, Set Record Length

The previously typed octal number will be established as the record length for all subsequent test operations. For write operations it determines the number of words to be written. For read it determines the maximum number of words which will be stored in memory. The maximum length record is 7777_8 (4095). If a larger number is entered it will be reduced to this maximum.

2.1.3 U, Set Unit Number

The previously typed octal digit is used to identify the logical tape unit number which is to be tested. The program adjusts all tape unit addressing instructions accordingly.

2.1.4 C, Set Record Count

The record counter is incremented by one every time the program passes a record in the forward direction, and decremented by one for the reverse direction. After a rewind it is cleared to zero. This control function with the letter C is provided so that the user can reset this counter whenever he wishes to start a series of operations for which a count is needed. The previously typed 8 octal digits will be saved as the new record count.

2.1.5 Z, Set Parity

If the preceding digit is even the program converts all read and write EOM's to the BCD even parity mode. If the digit is odd it converts all read and write EOM's to the binary odd parity mode.

2.2 TAPE OPERATION CONTROL

2.2.1 Breakpoint Functions

Breakpoints 1, 2 and 4 apply generally to all tape operations. Breakpoint 3 is used when writing.

USE: (cont.)

BP 1	RESET:	Continuous operation. The operation will continue as long as this Breakpoint is RESET or until the end of tape is reached.
	SET:	Stop continuous operation. If initially SET do only one operation.
BP 2	RESET:	Perform all operations without stopping between records.
	SET:	Stop after each record.
BP 3	RESET:	Write normally.
	SET:	Write a continuous record as long as this Breakpoint is set.
BP 4	RESET:	Stop if a read or write error occurs.
	SET:	Do not stop on a read or write error.

2.2.2 W, Write

The previously set pattern will be written as a record of length indicated by the L function. Records will be continuously written as long as Breakpoint 1 is RESET. If Breakpoint 3 is SET one long continuous record will be written. If Breakpoint 4 is RESET and a write error occurs, the program will halt and print WRITE ERROR. If Breakpoint 4 is SET the error will not cause a stop. If the tape is situated on the end of tape marker the write routine will write a Tape Mark to signify end of file, then type END OF REEL, and return to keyboard.

2.2.3 I, Write with Identification

This function is the same as Write except that the record count number is written as the first word of the record. This provides a unique identification word for each record. The record count word is inserted as the first word of the image so no additional words are added to the record.

2.2.4 R, Read

The next record on the selected tape unit will be read into memory. If the record is longer than the preset record length, the program will skip the extra words. If the skip remainder of record operation is not functioning and additional I1 interrupts occur, the program will count these and print the count at the end

USE: (cont.)

of the record. As long as Breakpoint 1 is RESET the program will read records sequentially until an end of file or the end of tape is encountered. If Breakpoint 2 is SET, the program will stop the tape after each record; otherwise the tape will run without stopping.

If a read error occurs and Breakpoint 4 is RESET, the program will stop and type READ ERROR. If BP 4 is SET, the program continues.

2.2.5 B, Backspace

If the input number previous to the B is cleared to zero by a carriage return then the selected tape is backspaced one record at a time as long as Breakpoint 1 is RESET or until the load point or an end of file is encountered. If the input number previous to the B is non zero then it is decremented by one after each backspaced record and the backspacing operation is terminated when it reached zero. The counted backspace may also be stopped on Breakpoint 1, the load point, or an end of file. If Breakpoint 2 is reset, the spacing will proceed without stopping between records. If Breakpoint 2 is set, the tape will be stopped after each record spaced and then restarted to continue over the next one.

2.2.6 S, Space Forward

Space forward is the same as backspace except that it will also stop when the end of tape is reached.

2.2.7 X, Search-Forward

The selected tape is searched forward using the read-scan mode until a record is found whose first word is equal to the previously typed octal number. When the record is found it is read into memory. If an end of file or the end of tape is encountered before the record is found the tape is stopped.

2.2.8 Y, Search Reverse

The selected tape is scanned backward until a record is found whose first word (last word scanned over in reverse) is equal to the previously typed octal number. If an end of file or the load point is encountered before the record is found, the tape is stopped.

2.2.9 D, Rewind

The selected tape is started in a rewind and the program returns to keyboard control.

USE: (cont.)

2.2.10 E, Erase

This function is similar to write except that instead of writing information, it erases tape for a distance equal to the specified record length.

2.2.11 F, Write End of File

A tape mark is written on tape to indicate End of File. The record counter is not incremented by the operation. The operation may be executed even though the tape is on the end of tape conductive leader.

2.3 OUTPUT CONTROL

After the tape has been read or moved by some other operation, the results may be inspected using the following control characters.

2.3.1 N, Tape Record Count Number

The program will type the current contents of the record counter.

2.3.2 T, Type Record Read

If the number of words read was less than or equal to the preset length, the program will type: RECORD LENGTH < OR = LLLL, where LLLL represents the record length. If the number of words read exceeds the present record length, the program will type: RECORD LENGTH > LLLL.

After typing one of these messages the program will begin to type the record image in octal numbering each eighth word in octal. This output will continue until either Breakpoint 1 is SET or until the entire record is typed.

If Breakpoint 1 is SET when the T key is struck only the record length will be typed.

3.0 STATUS AND ERROR MESSAGES

The following messages will be typed by the program to inform the operator of the status of the tape operation:

USE: (cont.)

3.1 WRITE ERROR

This indicates that the program detected a write error and Breakpoint 4 was RESET. The tape is stopped and the program returns to keyboard control after typing.

3.2 READ ERROR

This indicates that the program detected a read error and Breakpoint 4 was RESET. The tape is stopped and the program returns to keyboard control after typing.

3.3 SKIP REMAINDER OF RECORD ERRORS: nnnnnnnn

This indicates that the record read was longer than the preset record length and that when the program attempted to ignore the remainder of the record, it still received I1 interrupts. The number of interrupts is indicated by the octal number nnnnnnnn. Control returns to the keyboard after typing. This error stop may not be disabled by Breakpoint 4.

3.4 FILE PROTECT ON

This is typed whenever the user asks for a write or erase operation on a tape which has the file protect ring removed. After typing the program returns to keyboard control.

3.5 END OF REEL

This indicates that a forward operation has reached the end of tape.

3.6 LOAD POINT

This indicates that a reverse operation has reached the beginning of tape.

3.7 TAPE MARK

This indicates that a read, scan, or space operation has encountered an End of File record as indicated by the reading of a Tape Mark.

4.0 FUNCTION SUMMARY

In the calling sequence the small letter d is used to denote an octal digit.

USE: (cont.) 4. FUNCTION SUMMARY (cont.)

Function Description		
Clear digit accumulator		Carriage Return
Set test pattern word		ddddddddP
Set record length		ddddL
Set logical tape unit number		dU
Set record count number		ddddddddC
Set to even parity (BCD)		0 Z
Set to odd parity (Binary)		1 Z
Write		W
Write with record count ID		I
Read		R
Backspace		B
Backspace by count	(octal count)	B
Space forward		S
Space forward by count	(octal count)	S
Search forward and read		dddddddd X
Search reverse		dddddddd Y
Rewind		D
Erase		E
Write end of file		F
Type record count number		N
Type record read		T

IDENTIFICATION: Multi-Magnetic Tape System Exerciser

AUTHOR: A. W. England, SDS

ACCEPTED: 24 June 1964

COMPUTER CONFIGURATION: All 920, 925, and 930 systems, or any 910 systems with a typewriter, which have one to sixteen tape units attached to the W and/or Y buffers. No interlace is required and the tapes may be of any density and speed within the limitations of the buffer to which they are attached.

PURPOSE: This program is designed to exercise from one to sixteen tape units by first writing random numbers in random length records on all tapes under test and then reading these records back and comparing them with the numbers written. An attempt is made to tabulate and output all useful information concerning the errors made, if any, the mode of operation of each unit, and the number of passes over the tape.

STORAGE: The program occupies the first 1155 words of memory. The remaining memory may be used for test record storage.

TIMING: The program requires approximately 20 minutes to write or read a full reel (2400 feet) of tape.

USE:

1.0 LOADING

Place tape in reader and FILL. When loading is complete the light on the typewriter will light if no loading error occurred.

2.0 KEYBOARD CONTROL

When the keyboard light is on, the operator has control over the program. By actuating various keys he may set the test parameters, inspect results or start the exerciser test running.

2.1 REGAINING KEYBOARD CONTROL

Control may be returned to the keyboard mode at any time by moving the RUN-IDLE-STEP switch to IDLE, pressing the START button, and moving the switch first to STEP then to RUN.

*Supersedes Program Description Catalog No. 044004B

USE: (Cont)

3.0 CONTROL FUNCTIONS

The following list contains the call letters for the various functions which the program will perform. These may be typed anytime the typewriter light is lit.

3.1 SELECT UNITS, "U" U φ,

The units to be exercised are selected by first typing the letter "U" followed by the several unit numbers separated by commas and finally a carriage return. After the last unit number followed by a comma is entered a carriage return must be given to terminate the unit select operation. Units on the W buffer are numbered 00-07 and on the Y buffer $10\text{-}17_8$.

3.2 SET STARTING RANDOM NUMBER, "N" 7777 N

The initial random number is set by first typing the octal number desired (up to 8 digits) and then the letter N. The number being typed can be set to zero by typing a carriage return.

3.3 SET MAXIMUM FILE LENGTH, "M" 1000 M

The maximum number of records in the test file is set by typing the desired number of records in octal followed by the letter M. If the entire 2400 foot reel is to be written a maximum count of 10000_8 or greater should be sufficient.

3.4 SET MAXIMUM RECORD LENGTH, "L" 40 L

The maximum number of words in a record is set by typing the limit in octal followed by the letter L. If the specified maximum is less than or equal to 8 or greater than the maximum memory available then the maximum length is set equal to the memory available.

3.5 MODE SELECT

The recording mode, either BCD or Binary is selected by typing the appropriate letter.

3.5.1 Select Binary Mode, "B"

Typing the letter B will cause the appropriate EOM instructions to be converted to the binary mode of operation.

USE: (Cont)

3.5.2 Select BCD Mode, "D"

Typing the letter D will cause the EOM instructions to be set for BCD operation.

3.6 SELECT OUTPUT MEDIA

The output of the various messages and counters during the operation of the program can be on either the on-line typewriter or on paper tape for off-line listing. This is controlled by typing the appropriate letter before starting.

3.6.1 Select Typewriter Output, "T"

The typewriter is selected by typing the letter T.

3.6.2 Select Punch Output, "P"

The punch is selected by typing the letter P.

3.7 INITIATE TAPE OPERATION

After the appropriate parameters have been set the tape exercise operation may be initiated. There are three ways in which this may be done. If nothing has been recorded then the exercise must be begun with a START WRITE. However, once a file of information is written on tape and the program is stopped the other two starts can be used.

3.7.1 Start Write, "S"

To begin the exercise operation, type the letter S. The program will rewind all units and start to write a random number test file on the selected units.

3.7.2 Continue Operation, "C"

Once the exercise operation has been stopped with Breakpoint 1 (see section 4.1) it can be resumed from the point at which it was stopped by typing the letter C.

3.7.3 Restart Read, "R"

If during a read pass the program is stopped and the operator would like to reread the file from the beginning he can type the letter R to restart the read pass.

3.8 OUTPUT OPERATIONAL STATUS, "O"

The operator can inspect the status of the operation at anytime by stopping the program with Breakpoint 1 (see section 4.1) and typing the letter O. The program will then type out the status of the exercise operation as follows:

USE: (Cont)

3.8.1 Type of Pass

It types READ or WRITE depending on the type of pass in progress.

3.8.2 Mode of Operation

It then types the mode of operation, either BINARY or BCD.

3.8.3 Unit

The UNIT NO. of the tape unit currently being addressed is typed.

3.8.4 Program Counters

After this information the program will type a table of 17 counters each identified by a three or four character symbol. These symbols and their definitions follow:

MRC Maximum Record Count. This is the octal number entered with the M key at the start of the exercise operation.

MRL Maximum record length currently being used.

WRC Write Record Count. If in a write pass this indicates the number of records written. In a read pass it indicates the total number written in the previous write pass.

RRC Read Record Count. This indicates the number of records read during a read pass.

WPC Write Pass Count. The number of write passes completed.

RPC Read Pass Count. The number of read passes completed.

WEC Write Error Count. The number of write errors that have occurred.

RWEC Rewrite Error Count. This number of rewrite errors.

PREC Permanent Read Error Count. The records that were read bad 10 times.

CPEC Character Parity Error Count. The number of character parity errors that have occurred since the start of the exercise.

USE: (Cont)

LPEC Longitudinal Parity Error Count. The number of longitudinal parity errors that have occurred. For each read try only one character or longitudinal parity can be counted and character parity has priority.

WCEC Word Count Error Counts. The number of word count errors that have occurred. A word count error occurs if the record read is longer or shorter than the record expected.

CH1, CH2, CH3, CH4, CH5, CH6 Errors in Channels 1-6. Channel 1 is the most significant bit, channel 6 the least. These counters are also output whenever a read error occurs if Breakpoint 2 is RESET. After a read error output they are cleared.

4.0 BREAKPOINT SWITCHES

The four Breakpoint switches are used to change the status of the program while it is running. These functions are as follows:

4.1 BREAKPOINT 1

RESET: Normal

SET: Stop operation. After almost every tape operation there is a STOP point. If Breakpoint 1 is set the program will mark its place and return to the keyboard control mode. Operation can be continued by typing the letter C.

4.2 BREAKPOINT 2

RESET: Output counters and messages whenever the normal output situation occurs.

SET: Skip the output of messages and counters. This will inhibit all output except the OUT OF SYNC message and the FILE PROTECT ON message.

4.3 BREAKPOINT 3

RESET: At the end of a read pass go on to another write with new random numbers.

SET: At the end of a read pass go back and reread the same file again.

4.4 BREAKPOINT 4

RESET: Run without halts.

SET: Halt on a write error or at the end of a read pass. Clearing these halts will allow the program to continue.

USE: (cont.)

5.0 MESSAGES

The program will type or punch status messages at various times in the operation of the exercise. These are described below:

5.1 END OF PASS

At the end of a write or read pass the output will be either WRITE or READ, PASS DONE. This is followed by a carriage return and the following two lines:

```
 WRITES    READS   WRITE ERR REWRITES BAD READS
aaaaaaaa bbbbbbbb cccccccc dddddddd eeeeeeee
```

where the a's represent the number of write passes in octal, the b's the number of read passes, the c's the number of write errors which have occurred, the d's the number rewrite errors, and the c's the number of records which were read erroneously 10 times.

5.2 REWRITE ERROR

If a write error is detected the program erases backward over the record and attempts to rewrite it. If this second attempt is also in error the program outputs the following counter titles:

WRITE PASS RECORD NO. WRITE ERRS REWRITE ERRS

This is followed on the same line by the mode of operation (Binary or BCD) and the unit number. On the next line below the appropriate title it outputs the write pass count, the write record number count, the write error count and the rewrite error count. All counts are in octal.

5.3 READ ERROR

If a read error occurs, the program rereads the record nine more times and then outputs the read pass, record number, mode, and unit number. This is followed by a carriage return, the message, READ ERROR another carriage return and then nine, eight-octal-digit counters which represent the following quantities (from left to right): character parity error count, longitudinal parity error count, word count error count, errors in channel 1, channel 2, etc., to channel 6. On the next line the program outputs a good or bad message for each of the 10 reads. This consists of the letter G if the read was correct or B if the read was incorrect.

USE: (cont.)

For example:

B G G G B G G G G G

Indicates that the first and fifth reads were bad and all others were good.

5.4 READ PASS OUT OF SYNC

The first word of every record is the number of records preceding it on the tape. When each record is read, the program compares this first word with the read record count. If they disagree the program backspaces and rereads the record a second time, if they still disagree then the difference between them is computed and the program spaces over as many records as necessary to position itself in front of the correct record. If the first word of this record does not agree with the read record count after two attempts the program ends the read pass and outputs the following. As in a read error it outputs the read pass count, read record number, mode, density and unit number. This is followed by this message:

```
READ PASS ABORT, OUT OF SYNC.
    aaaaaaaa       bbbbbbbb
```

where the a's represent the first word of the first record read that did not agree with the read record count, and the b's represent the first word of the record read after spacing to what should have been the correct record. The program then goes to the end of read pass section where the end of pass output will be produced and from there on to another write or reread pass.

If a tape mark or the load point was encountered when spacing, the program terminates the read pass and outputs TAPE MARK before the other outputs. If it was the load point which was encountered it also outputs LOAD POINT. In either case the two words a and b will be the same since only one record was read.

5.5 FILE PROTECT ON

Before the program attempts to write on a tape it tests the file protect for that unit. If the file protect should be on, the program outputs: FILE PROTECT ON (Mode) UNIT NO. n. and returns to the keyboard mode.

METHOD: 1.0 WRITING

At the start of the write pass all units are rewound. The program then sets the tape control table for the lowest numbered unit and waits for it to be ready. As soon as this unit is ready a check is made to see if the tape is at

METHOD: (cont.)

the loadpoint. If it is not,another rewind is given and the program waits until it is ready and at the load point. A three inch section of tape is erased before the first random number record is written. After writing this record on the first unit the control table is set to the next higher numbered unit and the record is written again. This continues until a record has been written on all units under test. The program then generates a new record of random numbers and starts writing this on all units.

1.1 WRITE ERROR

If a write error occurs the program erases backward to the front of this record and attempts to rewrite it. If this second attempt is also in error then the program outputs the rewrite error message. It then erases backward over the record again, erases it forward and attempts to write the record again on a new section of tape. An error here is considered a new write error and the process continues until a correct write is made.

1.2 END OF PASS

The write pass is concluded if one of two conditions occurs: Either the write record count reaches the maximum record count or an end of reel is encountered on any tape under test. When one of these occurs the program writes an end of file on all units and rewinds them. It then outputs the end of pass message and proceeds to the read pass.

2.0 READING

A read pass is similar to a write except that the program reads each record into memory and compares it with the random numbers which it regenerates for each read. The first record must be read starting from the load point. This insures that the tape is always positioned properly for the start of the pass.

2.1 READ ERRORS

When a read error occurs the program will always reread the record nine more times for a total of ten attempts regardless of whether or not a subsequent read was correct. It then outputs the results of these reads. Several conditions can cause a read error.

2.1.1 Character Parity Errors

The program counts a character parity error as any buffer error which occurs before the gap is reached.

METHOD: (Cont)

2.1.2 Longitudinal Parity Error

If no character parity errors have occurred before the gap is reached and the buffer error is on after the tape stops, the program counts a longitudinal parity error.

2.1.3 Word Count Error

A word count error is defined as a record which was not of the length expected. The program tests for this in three ways. If more words than expected were read the read routine falls out of the loop too soon. The program also checks to see if the read loop should detect the end of record before expected. The third test is based on the fact that the program always writes records that consist of a multiple of four characters. Therefore, if the buffer contains anything other than zero at the end of the read an error has occurred.

2.2 READ SYNCHRONIZATION

When each record is read the first word is compared against the program record count. If they disagree it means that the program and tape are no longer synchronized. To guard against a read error causing the disagreement, the program backspaces and reads the record again. If they still disagree then the program computes the number of records to be spaced over in order to reach the desired record and moves to that point. It reads the new record and again checks the first word. If this word disagrees with the record count and a second read attempt does not correct the disagreement then the program aborts the read pass and outputs the appropriate message. If a tape mark or the load point is encountered while spacing to the correct position the pass is aborted without further read attempts.

2.3 END OF FILE

If the program should detect the buffer ready after the first word is read then a check for end of file is made. If the ready condition was caused by the reading of a tape mark then the read pass is complete and appropriate messages are output. If there is no tape mark character then the program assumes that the tape mark was read erroneously and terminates the read pass anyway and outputs an END OF FILE READ ERROR message.

METHOD: (Cont)

3.0 BCD MODE

In the BCD mode random numbers are generated and written the same as in binary. However, on the read pass all non compares between the generated number and the number from tape are checked to see if they are caused by the 12 to 00 conversion. This occurs because both the character 00 and the character 12 will be written on tape as a 12 but this character will always be read into memory as a 00.

A

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 544001B

IDENTIFICATION: Extended Mode Multi-Magnetic Tape Exerciser

AUTHOR: S. J. Vorgitch, SDS

ACCEPTED: May 20, 1965

COMPUTER CONFIGURATION: Any SDS 925/930 system with 1 to 64 tape units attached to interlaced channels A through H. The typewriter on channel A(W) is used for control.

PURPOSE: The program is designed to exercise 1 to 64 magnetic tapes on channels A through H. The exercise operates under interrupt control in the extended mode using all four function codes and all SKS's and EOM's associated with the cahnnel and magnetic tape.

PROGRAMMED OPERATORS: None

STORAGE: The program occupies 1976 locations from 100_8 through 3770. The remainder of memory may be used for test record storage.

TIMING: The program requires approximately 10 to 15 minutes to write or read a full reel (2400 feet) or tape.

SOURCE LANGUAGE: META-SYMBOL

LOADING PROCEDURE: Self-loading

USE:

1.0 LOADING

The program is loaded using the standard fill procedure. The typewriter will be addressed if no loading error occurred.

2.0 KEYBOARD CONTROL

When the keyboard light is on, the operator has control over the program. By actuating various keys he may set the test parameters, inspect results or start the exerciser test running.

USE: (Cont)

2.1 REGAINING KEYBOARD CONTROL

Control may be returned to the keyboard mode at any time by moving the RUN-IDLE-STEP switch to IDLE, pressing the START button, and moving the switch first to STEP, then to RUN.

3.0 CONTROL FUNCTIONS

The following list contains the call letters for the various functions which the program will perform. These may be typed anytime the typewriter light is lit.

3.1 Select Units, "U" u ϕ,

u ϕ, ϕ FOR SINGLE UNIT TEST

The units to be exercised are selected by first typing the letter "U" followed by up to sixty-four unit numbers separated by comma's. After the last unit number is input with a comma, a carriage return will terminate unit selection. Units are numbered 0-7 for channel A, 10-17 for channel B, ... 70-77 for channel H.

3.2 Set Starting Random Number, "N"

The initial random number is set by typing the letter "N" followed by an octal number and entered with a carriage return.

3.3 Set Maximum File Length, "M"

The maximum file length is set by typing the letter "M" followed by an octal number and entered with a carriage return. Due to the writing technique used, the minimum file length is three records and the input maximum may be exceeded by three records.

3.4 Set Maximum Record Length, "L"

The maximum number of words in a record is set by typing the letter "L" followed by an octal number and entered with a carriage return. The minimum record length allowed by the program is four words. The maximum record length is determined within the program by the memory available.

USE: (Cont)

3.5 Select Mode, "B" or "D"

The recording mode can be set by typing a "B" for binary or a "D" for BCD.

3.6 Initiate Tape Operation

After the appropriate parameters have been set the tape exercise operation may be initiated. There are four ways in which this may be done. If nothing has been recorded then the exercise must begin with a START WRITE or if the tape unit(s) is in the test mode and not at load point the WRITE TEST may be used. However, once a file of information is written on tape and the program is stopped the other two starts can be used.

3.6.1 Start Write, "S"

To begin the exercise operation, type the letter "S". The program will rewind all units and start to write a random number test file on the selected units.

3.6.2 Continue Operation, "C"

Once the exercise operation has been stopped with Breakpoint 1 (see section 4.1) it can be resumed from the point at which it was stopped by typing the letter "C".

3.6.3 Restart Read, "R"

If during a read pass the program is stopped and the operator would like to reread the file from the beginning he can type the letter R to restart the read pass.

3.6.4 Write Test "K"

To perform in this mode the tape unit specified must be in the test mode and not at load point. The program will write random length, random number records until stopped with Breakpoint 1.

USE: (Cont)

3.7 Add or Remove a Unit From Test, "P"

If during a read or write pass the program is stopped with Breakpoint 1, a unit can be removed from the exercise by typing the letter "P" followed by a minus (-) and the unit number terminated with a carriage return. A unit can only be added to exercise prior to a write pass. This is accomplished by stopping the program after the read pass (B.P. 4) then setting B.P. 1 and clearing the halt. After the rewind the typewriter will be addressed. Type the letter P followed by a plus (+ or &) and the unit number and terminated with a carriage return.

3.8 Output Operational Status, "O"

The operator can inspect the status of the operation at anytime by stopping the program with Breakpoint 1 (see section 4.1) and typing the letter O. The program will then type out the status of the exercise operation indicated by section 3.8.1 - 3.8.4. Set Breakpoint 1 to terminate output and return control to the typewriter.

3.8.1 Type of Pass

It types READ or WRITE depending on the type of pass in progress.

3.8.2 Mode and Density

It then types the mode of operation, either BINARY or BCD and the tape density.

3.8.3 Unit

The UNIT NO. of the tape unit currently being addressed is typed.

3.8.4 Program Counters

After this information the program will type a table of 18 counters each identified by a three or four character symbol. These symbols and their definitions follow:

USE: (Cont)

MRC	Maximum Record Count. This is the octal number entered with the M key at the start of the exercise operation.
CRN	Current Record Number. Indicates record number just completed.
MRL	Maximum record length currently being used.
WPC	Write Pass Count. The number of write passes completed.
RPC	Read Pass Count. The number of read passes completed.
WEC	Write Error Count. The number of write errors that have occurred.
RWEC	Rewrite Error Count. This number of rewrite errors.
PREC	Permanent Read Error Count. The records that were read bad 10 times.
CPEC	Character Parity Error Count. The number of character parity errors that have occurred since the start of the exercise.
LPEC	Longitudinal Parity Error Count. The number of longitudinal parity errors that have occurred. For each read try only one character or longitudinal parity can be counted and character parity has priority.
WCEC	Write Count Error Count. The number of word count errors that have occurred. This is accomplished by "PINning" the channel and comparing with the expected record length.

USE: (Cont)

RCEC	Read Count Error Count. The number of word count errors that have occurred. This is accomplished by "PINning" the channel and comparing with the expected record length.
CH1 CH2 CH3 CH4 CH5 CH6	Errors in Channels 1-6. Channel 1 is the most significant bit, channel 6 the least. These counters are also output whenever a read error occurs if Breakpoint 2 is RESET. After a read error output they are cleared.

3.9 Write Error Message Option

3.9.1 Type "X" to inhibit the write error message (5.2.1).

3.9.2 Type "Y" to restore the write error message.

4.0 BREAKPOINT SWITCHES

The breakpoints are used to change the status of the program while it is running. These functions are as follows:

4.1 Breakpoint 1

Reset: Normal

Set: Stop operation. After almost every tape operation there is a STOP point. If Breakpoint 1 is set the program will mark its place and return to the keyboard control mode. Operation can be continued by typing the letter C.

4.2 Breakpoint 2

Reset: Output counters and messages whenever the normal output situation occurs.

Set: Skip the output of messages and counters. This will inhibit all output except messages described in section 5.4 - 5.9.

USE: (Cont)

4.3 Breakpoint 3

Reset: At the end of a read pass go on to another write with new random numbers.

Set: At the end of a read pass go back and reread the same file again.

4.4 Breakpoint 4

Reset: Run without halts.

Set: Halt on a write error or at the end of a read or write pass. Clearing these halts will allow the program to continue.

5.0 MESSAGES

The program will type status messages at various times in the operation of the exercise. These are described below:

5.1 End of Pass

At the end of a write or read pass the output will be either WRITE or READ, PASS DONE. This is followed by a carriage return and the following two lines:

WRITES	READS	WRITE ERR	REWRITES	BAD READS	READ ERRS
aaaaaaaa	bbbbbbbb	cccccccc	dddddddd	eeeeeeee	ffffffff

where the a's represent the number of write passes in octal, the b's the number of read passes, the c's the number of write errors which have occurred, the d's the number rewrite errors, the e's the number of records which were read erroneously 10 times, and the f's the number of erroneous reads.

5.2 Writing Errors

5.2.1 Write Error

If a write error occurs and the message is not inhibited (3.9), the following line will be output

WRITE ERROR mode density uuuuuuuu

where the u's represent the unit number.

USE: (Cont)

5.2.2 Rewrite Error

If a write error is detected the program erases backward over the record and attempts to rewrite it. If this second attempt is also in error the program outputs the write pass count, the write record number count, the write error count, the rewrite error count, and the number of parity errors. All counts are in octal.

5.3 Read Error

If a read error occurs, the record is re-read nine more times. During the reread, a "B" or "G" will be typed to indicate that the re-read was either Bad or Good. After nine re-reads a read error message is typed indicating on the first line the read pass number, the record number, the mode, density, and unit followed by a carriage return. Nine octal numbers are then typed which represent the number of character parity errors, the number of longitudinal parity errors, the number of read count errors and the number of tape channel errors. (CH1-CH6)

5.4 Unit N Aborted Record Sync Error

The first word of every record is a record number identification. This number is compared with the record number expected. If they disagree the tape is backspaced one record and re-read. If the record number is still in error, the tape is backspaced a maximum of 8 records looking for the last record read correctly. If successful the tape is spaced forward one record and the process is repeated. If the previous good record was not found or the recovery procedure failed a second time, that unit is declared out of synchronization and removed from the exercise.

5.5 Unit N Aborted Density Test Failure

If, while attempting to erase forward off the load point prior to the write pass, the program cannot ascertain the density of a unit the unit is removed from the test.

USE: (Cont)

5.6 Unit N Aborted Non-Ready Condition

If during the course of program operation a unit does not come ready after approximately 5 minutes it is removed from the exercise.

5.7 Unit N Aborted Write Count Error

If, after a write operation, the results of PINning the channel do not compare with the expected address after two attempts, the unit is removed from the test.

5.8 Unit N Aborted Excessive Errors

Each unit under test is initialized to permit the typing of 200 error messages. If any unit exceeds this limit it is removed from the test. Type an "S" to reset the error counter and re-start the test.

5.9 All Units Removed From Test

If during program operation, the unit count is found to be zero due to program aborts or console removal (section 3.7) a message is typed and control is returned to the console.

METHOD:

1.0 WRITING

At the beginning of the write pass all units under test are rewound and erased forward approximately 3.5 inches. A random length record of random numbers is generated within the program and is used to write 3 records on the first tape in the unit list. The whole record is first written using the function codes IOSP and IOSD. The first half of the generated record is then written to tape with the IORP function code. The third record is the last half of the generated record and is written with an IORD function code. This process is repeated for all units under test, after which, another record is generated and writing returns to the first tape in the unit list.

The IOSP arms both the zero word count and end of record interrupts, whereas the IOSD, IORP, and IORD only arm the end of record interrupt.

METHOD: (Cont)

1.1 Write Error

If a write error occurs the program erases backward to the front of this record and attempts to rewrite it. If this second attempt is also in error then the program outputs the rewrite error message. It then erases backward over the record again, erases it forward and attempts to write the record again on a new section of tape. An error here is considered a new write error and the process continues until a correct write is made.

1.2 Write Count Error

After each write operation the channel is PINned and compared with the expected terminating address. If they differ an error note is typed indicating the mode, density, unit, channel and two octal numbers which represent the number of words written (result of PINning the channel) and the number of words attempted. After the note a space reverse is performed and a re-write is attempted. If unsuccessful the error note will again be printed followed by an abort message (see 5.7).

1.3 End of Pass

The write pass is concluded if one of two conditions occurs: Either the write record count reaches the maximum record count or an end of reel is encountered on any tape under test. When one of these occurs the program writes an end of file on all units and outputs the end of pass message. It then rewinds all units and proceeds to the read pass.

2.0 READING

A read pass is similar to a write except that the program reads each record into memory and compares it with the random numbers which it regenerates for each read. In addition, the proceed mode of reading is inhibited (IORF) such that the tape will stop and test ready before proceeding to the next read operation. The first record must be read starting from the load point. This insures that the tape is always positioned properly for the start of the pass.

METHOD:
(Cont)

2.1 Read Errors

When a read error occurs the program will always reread the record nine more times for a total of 10 attempts regardless of whether or not a subsequent read was correct. It concurrently outputs the results of these reads. Several conditions can cause a read error.

2.1.1 Character Parity Errors

The program counts a character parity error as any channel error which occurs before zero word count.

2.1.2 Longitudinal Parity Error

If no channel errors have occurred before the zero word count is reached and the channel error is on after the tape stops, the program counts a longitudinal parity error.

2.1.3 Data Compare Error

Any data word (except identifier) which does not compare with the word expected is considered cause for nine re-reads.

2.1.4 Read Count Error

After each read operation the channel is PINned and the results compared with the expected terminating address. If they differ the record is re-read nine times.

3.0 BCD MODE

In the BCD mode random numbers are generated and written the same as in binary. However, on the read pass all non compares between the generated number and the number from tape are checked to see if they are caused by the 12 to 00 conversion. This occurs because both the character 00 and the character 12 will be written on tape as a 12 but this character will always be read into memory as a 00.

Catalog No. 020012

NO INTERLACE

IDENTIFICATION: Paper Tape Reproducer Program

AUTHOR: A. W. England, SDS

ACCEPTED: 15 August 1963

COMPUTER CONFIGURATION: Any 910 or 920 with punch, reader, and typewriter.

PURPOSE: To reproduce paper tape.

PROGRAMMED OPERATORS: N/A

STORAGE: The program occupies 270_{10} locations from 200_8 to 616_8. The next 512 words are reserved for record table storage. The remainder of memory is used to hold the records of the tape to be reproduced.

TIMING: All operations proceed at the maximum rate of either the punch (60 characters per second) or the reader (300 characters per second).

USE:

I. TO LOAD PROGRAM

A. Insert tape in reader.

B. With COMPUTE switch in IDLE press START button.

C. Move COMPUTE switch to RUN.

D. Set BP 1.

E. Raise and lower FILL switch.

II. TO REPRODUCE A TAPE

A tape may be reproduced by first reading it, then verifying it, punching one or several copies of it, and finally verifying the copies. Each of these functions of the program will be described below. Upon

USE: (Cont)

completion of each function, control will be returned to the operator via the typewriter, which is signified by the illuminated typewriter light and the presence of 000001 in the I/O address lights.

A function is initiated by typing one control letter; R, V, or P. The letter P may be preceded by a number to indicate the number of copies to be made.

Below is the procedure for reproducing tapes:

A. Read

1. Place the tape to be reproduced in the reader.

2. RESET BP 1.

3. Type R.

4. When the tape runs out of the reader, SET BP 1.

B. Verify

1. Place the tape to be verified in the reader.

2. Reset BP 1.

3. If there is only one copy to be verified or if there are several copies to be verified one at a time, RESET BP 2.

4. Type V. The tape will be read and verified. If an error is found, the reader will stop at the end of the copy and the program will type VERIFY ERROR.

5. If several copies are to be verified at once, SET BP 2 and type V.

   When the tape runs out of reader or onto the the last length of trailer, SET BP 1.

C. Punch

After the tape has been read and verified, it may be punched as follows:

USE: (Cont)

1. If the tape is to have a short leader RESET BP 3. If it is to have a long leader (for mounting on a reel) SET BP 3.

2. If the tape is to have a short trailer, RESET BP 4. If it is to have a long trailer (for mounting on a reel), SET BP 4.

3. If only one copy is to be made, RESET BP 2 and type P. When the copy has been punched the light will come on.

4. If more than one copy is to be punched, SET BP 2 and type a carriage return, then the number of copies to be made, and the letter P. The program will punch the desired number of copies and stop. If it appears that there is not sufficient paper tape to punch all the copies, BP 2 may be RESET and the program will stop after the copy it is then punching.

D. Verify New Tape

Follow procedure as outlined in B above.

III. ERRORS

A. Reading

1. If a Read error occurs, the program will stop the tape and type READ ERROR.

2. If the tape to be reproduced exceeds the capacity of memory, the program will stop the tape and type STORAGE FULL.

3. If the tape to be reproduced has more than 512 blocks on it, the program will stop the tape and type TABLE FULL.

B. Verifying

If a copy fails verification, the program will stop the tape after that copy and type VERIFY ERROR.

IV. BREAKPOINTS

BP NO.	RESET	SET
1	Normal	Stop
2	One	Many
3	Short Leader	Long Leader
4	Short Trailer	Long Trailer

METHOD:

When a tape is read, the program records the gap length preceding the punched information and stores this in the record table. It also maintains the starting addresses of each record read. When Breakpoint 1 is set, the read is stopped and an end indicator is inserted in the table.

Verify is similar to read except that no attempt is made to verify the length of gaps. Each word of the record read from tape is compared with the corresponding word in memory. Any disagreement is indicated.

Punching utilizes the gap count generated during read to reproduce the proper length gaps between records. Each record is punched from memory with gaps as required. At the beginning and end of the tape the program punches either short (3 feet) or long (10 feet) leader depending on the settings of Breakpoints 3 and 4.

```
                                   LIST                          PAPER TAPE REPRODUCER
00200  0 71 00027  BEGIN  LDX   SC5
00201  0 76 00026         LDA   SC4
00202  2 35 00000         STA   0,2
00203  2 72 00000         SKA   0,2
00204  0 01 00207         BRU   *+3
00205  2 77 34000         EAX   NEG2K,2
00206  0 01 00201         BRU   *-5
00207  0 37 00612         STX   LIMIT
00210  0 76 00244         LDA   R8
00211  0 35 00001         STA   1
00212  0 02 20004  ENTER  DIR
00213  0 02 00001         EOM   RKBW1
00214  0 32 00012         WIM   T
00215  0 76 00012         LDA   T
00216  0 75 00255         LDB   C4                  77
00217  0 70 00252         SKM   R
00220  0 01 00222         BRU   *+2                 NOT R
00221  0 01 00256         BRU   READ
00222  0 70 00253         SKM   P
00223  0 01 00225         BRU   *+2                 NOT P
00224  0 01 00373         BRU   PUNCH
00225  0 70 00254         SKM   V
00226  0 01 00230         BRU   *+2                 NOT V
00227  0 01 00471         BRU   VERIFY
00230  0 70 00247         SKM   CR
00231  0 01 00233         BRU   *+2                 NOT CR
00232  0 01 00245         BRU   CLEAR
00233  0 14 00255  DIGIT  ETR   C4                  ACCUMULATE DIGITS
00234  0 35 00012         STA   T
00235  0 46 30003         RCH   30003
00236  0 76 00250         LDA   PCNT
00237  0 67 00001         LSH   1
00240  0 55 00250         ADD   PCNT
00241  0 67 00001         LSH   1
00242  0 55 00012         ADD   T
00243  0 35 00250         STA   PCNT
00244  0 01 00212  R8     BRU   ENTER
00245  0 46 30003  CLEAR  RCH   30003               CLEAR PCNT
00246  0 01 00243         BRU   *-3
00247  00000052    CR     OCT   52
00250  0 00 00000  PCNT   PZE
00251  0 00 00000  RPCNT  PZE
00252  00000051    R      OCT   51
00253  00000047    P      OCT   47
00254  00000065    V      OCT   65
00255  00000077    C4     OCT   77
00256  0 76 00575  READ   LDA   R1                  SET INTERRUPTS
00257  0 75 00576         LDB   R2                  /
00260  0 35 00031         STA   I1W                 /BRU CONB
```

```
00261  0 36 00033            STB     I2W                       /BRU I2RD
00262  0 76 00611            LDA     START                     STARTING ADDRESS 0F ST0R
00263  0 35 00610            STA     ADDR
00264  0 35 00616            STA     TBL
00265  0 71 00574            LDX     TBLS                      TABLE SIZE T0 TBLC
00266  0 02 20002            EIR                               ENABLE INTERRUPT
00267  0 02 03604            E0M     RPTW4                     START READER
00270  0 00 00000   WAITI1   HLT                               WAIT I1
00271  0 32 40610   C0NB     WIM*    ADDR
00272  0 61 00610            MIN     ADDR
00273  0 76 00612            LDA     LIMIT
00274  0 73 00610            SKG     ADDR
00275  0 01 40323            BRU*    E3
00276  0 01 40277            BRU*    *+1
00277  0 00 00270            PZE     WAITI1
00300  0 40 20010   I2RD     SKS     SBEW
00301  0 01 40321            BRU*    E1
00302  0 32 00014            WIM     T+2
00303  0 76 00014            LDA     T+2
00304  0 72 00026            SKA     SC4
00305  0 01 40321            BRU*    E1
00306  0 41 00310            BRX     *+2                       TBLC+1 T0 TBLC
00307  0 01 40322            BRU*    E2
00310  0 76 00610            LDA     ADDR
00311  2 35 01616            STA     TBLE,2
00312  0 76 00577            LDA     R3                        BRU C0NA
00313  0 35 00031            STA     I1W
00314  0 76 00320            LDA     BIAS
00315  0 02 03604            E0M     RPTW4
00316  0 01 40317            BRU*    *+1
00317  0 00 00324            PZE     C0NT
00320  77700000     BIAS     0CT     77700000
00321  0 00 00343   E1       PZE     ERR1
00322  0 00 00355   E2       PZE     ERR2
00323  0 00 00364   E3       PZE     ERR3
00324  0 55 00557   C0NT     ADD     C1                        00000200
00325  0 40 20400            SKS     BP1
00326  0 01 00336            BRU     ST0P
00327  0 01 00324            BRU     *-3
00330  0 14 00560   C0NA     ETR     C2                        37700000
00331  2 16 01616            MRG     TBLE,2
00332  2 35 01616            STA     TBLE,2
00333  0 76 00575            LDA     R1
00334  0 35 00031            STA     I1W
00335  0 01 00271            BRU     C0NB
00336  2 76 01616   ST0P     LDA     TBLE,2
00337  0 16 00025            MRG     SC3
00340  2 35 01616            STA     TBLE,2
00341  0 02 00000            E0M     0
00342  0 01 00212            BRU     ENTER
```

```
00343  0 02 00000  ERR1   EØM    0
00344  0 02 20004         DIR
00345  0 02 03641         EØM    TYPW4
00346  0 12 00562         MIW    EM1              CR REA
00347  0 12 00563         MIW    EM1+1            D SP ER
00350  0 12 00564         MIW    EM1+2            RØR CR
00351  0 02 14000  TØP    EØM    TØPW
00352  0 40 21000         SKS    SBRW
00353  0 01 00352         BRU    *-1
00354  0 01 00212         BRU    ENTER
00355  0 02 00000  ERR2   EØM    0
00356  0 02 20004         DIR
00357  0 02 03641         EØM    TYPW4
00360  0 12 00565         MIW    EM2              CR TAB
00361  0 12 00566         MIW    EM2+1            LE SP F
00362  0 12 00567         MIW    EM2+2            U LL CR
00363  0 01 00351         BRU    TØP
00364  0 02 20004  ERR3   DIR
00365  0 02 03641         EØM    TYPW4
00366  0 12 00570         MIW    EM3              CR STØ
00367  0 12 00571         MIW    EM3+1            R AGE
00370  0 12 00572         MIW    EM3+2            SP FUL
00371  0 12 00573         MIW    EM3+3            L . . CR
00372  0 01 00351         BRU    TØP
00373  0 76 00250  PUNCH  LDA    PCNT
00374  0 54 00024         SUB    SC2
00375  0 35 00251         STA    RPCNT
00376  0 76 00600         LDA    R4
00377  0 35 00424         STA    SW1              /
00400  0 71 00574         LDX    TBLS
00401  2 76 01616         LDA    TBLE,2
00402  0 14 00027         ETR    SC5
00403  0 35 00610         STA    ADDR
00404  0 76 00614         LDA    SLC              SHØRT LEADER CØUNT
00405  0 40 20100         SKS    BP3
00406  0 76 00613         LDA    LLC              LØNG LEADER CØUNT
00407  0 02 20004         DIR
00410  0 14 00560  CØNE   ETR    C2
00411  0 37 00012         STX    T
00412  0 02 01644  LØØP   EØM    PPTW4
00413  0 54 00561         SUB    C3
00414  0 72 00025         SKA    SC3
00415  0 01 00423         BRU    SW1-1
00416  0 71 00422         LDX    PWC
00417  0 35 00013         STA    T+1              KILL TIME
00420  0 41 00417         BRX    *-1
00421  0 01 00412         BRU    LØØP
00422  00060000    PWC    ØCT    60000
00423  0 71 00012         LDX    T
00424  0 20 00000  SW1    NØP
```

```
00425  2 76 01617         LDA     TBLE+1,2
00426  0 14 00027         ETR     SC5
00427  0 12 40610  OUT    MIW*    ADDR
00430  0 61 00610         MIN     ADDR
00431  0 73 00610         SKG     ADDR
00432  0 01 00434         BRU     *+2
00433  0 01 00427         BRU     OUT
00434  0 02 14000         EOM     TOPW
00435  0 41 00436         BRX     *+1
00436  0 40 21000         SKS     SBRW
00437  0 01 00436         BRU     *-1
00440  2 53 01616         SKN     TBLE,2
00441  0 01 00450         BRU     CONF                  NOT DONE
00442  0 76 00601  CONC   LDA     R5                    DONE
00443  0 35 00424         STA     SW1                   S(SW1) BRU COND
00444  0 76 00614         LDA     SLC
00445  0 40 20040         SKS     BP4
00446  0 76 00613         LDA     LLC
00447  0 01 00410         BRU     CONE
00450  2 76 01616  CONF   LDA     TBLE,2
00451  0 14 00027         ETR     SC5
00452  0 35 00610         STA     ADDR
00453  2 76 01616         LDA     TBLE,2
00454  0 01 00410         BRU     CONE
00455  0 02 14000  COND   EOM     TOPW
00456  0 40 21000         SKS     SBRW
00457  0 01 00456         BRU     *-1
00460  0 40 20200         SKS     BP2
00461  0 01 00463         BRU     *+2
00462  0 01 00212         BRU     ENTER
00463  0 76 00251         LDA     RPCNT
00464  0 54 00024         SUB     SC2
00465  0 35 00251         STA     RPCNT
00466  0 72 00025         SKA     SC3
00467  0 01 00212         BRU     ENTER
00470  0 01 00376         BRU     PUNCH+3
00471  0 76 00602  VERIFY LDA     R6                    BRU V1
00472  0 75 00603         LDB     R7                    BRU V2
00473  0 35 00031         STA     I1W
00474  0 36 00033         STB     I2W
00475  0 71 00574  CONTV  LDX     TBLS
00476  0 46 30003         RCH     30003
00477  0 35 00615         STA     VFLG                  R(VFLG)
00500  2 76 01616         LDA     TBLE,2
00501  0 72 00025         SKA     SC3
00502  0 01 00543         BRU     VDONE
00503  0 14 00027         ETR     SC5
00504  0 35 00610         STA     ADDR
00505  0 75 00026  VLOOP  LDB     SC4
00506  0 02 03604         EOM     RPTW4
```

```
00507  0 02 20002           EIR
00510  0 40 20400  PAUSE   SKS    BP1
00511  0 01 00212          BRU    ENTER
00512  0 01 00510          BRU    *-2
00513  0 32 00016  V1      WIM    T+4
00514  0 76 00016          LDA    T+4
00515  0 70 40610          SKM*   ADDR
00516  0 36 00615          STB    VFLG
00517  0 61 00610          MIN    ADDR
00520  0 01 40521          BRU*   *+1
00521  0 00 00510          PZE    PAUSE
00522  0 32 00016  V2      WIM    T+4
00523  0 40 20010          SKS    SBEW
00524  0 36 00615          STB    VFLG
00525  0 76 00016          LDA    T+4
00526  0 72 00026          SKA    SC4
00527  0 36 00615          STB    VFLG
00530  0 76 00610          LDA    ADDR
00531  0 75 00027          LDB    SC5
00532  0 01 40533          BRU*   *+1
00533  0 00 00534          PZE    *+1
00534  2 70 01617          SKM    TBLE+1,2
00535  0 01 00540          BRU    V2A
00536  0 41 00500          BRX    CØNTV+3
00537  0 01 00545          BRU    ERR4
00540  0 76 00026  V2A     LDA    SC4
00541  0 35 00615          STA    VFLG
00542  0 01 00536          BRU    *-4
00543  0 53 00615  VDØNE   SKN    VFLG
00544  0 01 00554          BRU    VCØNT
00545  0 02 20004  ERR4    DIR
00546  0 02 03641          EØM    TYPW4
00547  0 12 00604          MIW    EM4                   CR VER
00550  0 12 00605          MIW    EM4+1                 IFY SP
00551  0 12 00606          MIW    EM4+2                 ERRØ
00552  0 12 00607          MIW    EM4+3                 R.. CR
00553  0 01 00351          BRU    TØP
00554  0 40 20200  VCØNT   SKS    BP2
00555  0 01 00475          BRU    CØNTV
00556  0 01 00212          BRU    ENTER
00557  00000100    C1      ØCT    100
00560  37700000    C2      ØCT    37700000
00561  00100000    C3      ØCT    00100000
00562  52512521    EM1     ØCT    52512521,24122551,51465152
00565  52632122    EM2     ØCT    52632122,43251226,64434352
00570  52626346    EM3     ØCT    52626346,51212725,12266443,43333352
00574  77777000    TØLS    DEC    -512
00575  0 01 00271  R1      BRU    CØNB
00576  0 01 00300  R2      BRU    I2RD
00577  0 01 00330  R3      BRU    CØNA
```

```
00600   0 20 00000   R4      NOP
00601   0 01 00455   R5      BRU     COND
00602   0 01 00513   R6      BRU     V1
00603   0 01 00522   R7      BRU     V2
00604   52652551     EM4     OCT     52652551,31267012,25515146,51333352
00610   0 00 00000   ADDR    PZE
00611   0 00 01616   START   PZE     TBLE
00612   0 00 00000   LIMIT   PZE
00613   16000000     LLC     OCT     16000000
00614   04400000     SLC     OCT     04400000
00615   0 00 00000   VFLG    PZE
00616   0 00 00000   TBL     PZE
             01616   TBLE    EQU     TBL+512
             03604   RPTW4   BOOL    03604
             20010   SBEW    BOOL    20010
             20400   BP1     BOOL    20400
             20200   BP2     BOOL    20200
             20100   BP3     BOOL    20100
             20040   BP4     BOOL    20040
             00023   SC1     BOOL    23
             00024   SC2     BOOL    24
             00025   SC3     BOOL    25
             00026   SC4     BOOL    26
             00027   SC5     BOOL    27
             03641   TYPW4   BOOL    03641
             14000   TAPW    BOOL    14000
             21000   SBRW    BOOL    21000
             00012   T       BOOL    12
             01644   PPTW4   BOOL    01644
             00001   RKBW1   BOOL    00001
             34000   NEG2K   BOOL    34000
             00031   I1W     BOOL    31
             00033   I2W     BOOL    33
             00200           END     BEGIN
```

PUNCHED PPT LOADER
NO INTERLACE

Catalog No. 040003

REC REGION Q735

IDENTIFICATION: Paper Tape and Magnetic Tape Copier

AUTHOR: Reider D. Andersen and Philip H. Hartman, SDS

ACCEPTED: March 24, 1964

COMPUTER CONFIGURATION: Any 900 series computer with 4K memory and one magnetic tape unit

PURPOSE: To copy paper tape to magnetic tape and magnetic tape to paper tape.

PROGRAMMED OPERATORS: None

STORAGE: Locations 200_8 to 733_8 and 733_8 to end of memory for buffer area

TIMING: N/A

USE:

I. LOADING

The program is loaded via the standard FILL procedure. Set the necessary Breakpoints before loading.

II. PROGRAM RESTART

Move the RUN-IDLE-STEP switch to IDLE, press START, move the switch to step and then to RUN. Or, BRU to location 200_8.

III. OPERATION

	Breakpoints Set	Operation
A.	1	Paper tape to magnetic tape unit 1 (continued until halted).
B.	2	Magnetic tape unit 1 to paper tape (continues until halted or an End-of-File is encountered).

USE: (Cont)

	Breakpoints Set	Operation
C.	1, 2	Verify Magnetic tape unit 1 and paper tape input (continues until halted or an End-of-File is encountered).
D.	1, 4	Same as A, but halts after each record.
E.	1, 2, 4	Same as C, but halts after each record.
F.	2, 4	Same as B, but halts after each read and each punch output.
G.	1, 3, 4	Backspace one record on Magnetic tape unit 1.
H.	1, 3	Backspace one file on Magnetic tape unit 1.
I.	2, 3, 4	Skip one record on Magnetic tape unit 1.
J.	2, 3	Skip one file on Magnetic tape unit 1.
K.	1, 2, 3	Write an END-OF-FILE mark on Magnetic tape unit 1.
L.	1, 2, 3, 4	Rewind Magnetic tape unit 1.

IDENTIFICATION: Power Fail-Safe Interrupt Tester

AUTHOR: J. Gaines, SDS

ACCEPTED: 13 April 1964

COMPUTER CONFIGURATION: Any 910, 920, or 930

PURPOSE: To provide a program to test the Power Fail-Safe Interrupt System

PROGRAMMED OPERATORS: None

STORAGE: 323_8 locations

TIMING: N/A

USE:

1.0 This program gives the user the ability to go through a power OFF/ON cycle and provides a positive indication when an error occurs. The operation is as follows:

1.1 Set Breakpoint 2.

1.2 Load the paper tape by the normal FILL procedure.

1.3 The operator can proceed with an OFF/ON cycle at this point (i.e., turn computer OFF then ON).* As far as the routine is concerned, there is no limit to the speed with which power can be turned off and on.

1.4 If an error that disrupts the program occurs during an OFF/ON cycle, the computer halts.

Clear the halt to return to the program which then awaits another OFF/ON cycle (paragraph 1.3).

*Halting of the computer after setting Breakpoint 2 but prior to the OFF/ON cycle implies a memory failure independent of the Fail fe interrupts. If such an error occurs, go to paragraph 1.5 a couple c ...s.

If the user suspects memory failure, he ... the Memory Tester Routine, SDS Catalog No. 004007.

A slow off-on alternation of the most significant active bit in the P Register gives the user a simple visual indicator of whether the program is continuing to run properly.

1.5 To re-initialize, if it is necessary, branch to location 00001, set Breakpoint 2 and return to paragraph 1.3.

METHOD:

The program contains three parts: a power off routine, a power on routine, and a larger routine which forms the interrupted program during the OFF/ON cycle. This latter routine generates and tests random numbers endlessly.

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

Catalog No. 394001

IDENTIFICATION: Model EJ-30 System Junction Box Digital Checkout Program

AUTHOR: A. Kliszewski, SDS

ACCEPTED: November 11, 1964

COMPUTER CONFIGURATION: SDS 930 computer with EJ-30 junction box, and an A/D converter.

Also required: special test cable (Part No. 110047).

PURPOSE: To checkout the digital portion of the EJ-30 junction box.

PROGRAMMED OPERATORS: BID

STORAGE: Octal locations: 12-15, 23-26, 100, 101, 172, 200-577, 640-777

TIMING: Function of use.

USE: Loading is by FILL procedure.

Restart is by IDLE, START, STEP, RUN.

Breakpoint Control

BP1 and BP2 are interpreted together (binary coded) as the PIN input connector to which the test cable is connected.

One or both of them must be set at all times.

BP3 set causes the program to type out the number of successful loops completed and halt.

BP4 set causes the POT-PIN diagnostic routine to be entered.

USE: (Cont)

Error Indications

Two kinds of errors are detected:

a. SKS errors (SKIP when no SKIP was expected or vice versa).

b. PIN-POT errors (the word POTted out differs from the word PINned in).

Whenever an SKS error is detected, the program halts in the SKSERR subroutine. The SKS instruction which caused the error is in the A register. To continue as though no error had been detected, clear the HALT.

Whenever a PIN-POT error is detected, the program halts in the PIN ERR subroutine. The A register then contains the word POTted out, while the word PINned in is in the B register. To continue as though no error has been detected, clear the HALT.

PIN-POT Diagnostic Routine

This routine is entered when restarting the program with BP4 set. It halts to allow the user to set the A, B and X registers. It then performs the following:

POT out contents of A register to EJ-30 POT register,

PIN in the EJ-30 POT register through the appropriate connector,

Stall (based on contents of X register),

POT out contents of B register to EJ-30 POT register,

PIN in the EJ-30 POT register through the appropriate connector,

Stall,

Repeat the sequence indefinitely.

METHOD: If BP4 is set, only the PIN-POT diagnostic routine is executed.

Otherwise the program performs all the tests possible with the given connection. These include:

SKS tests if cable is connected to PIN input No. 1 or No. 2.

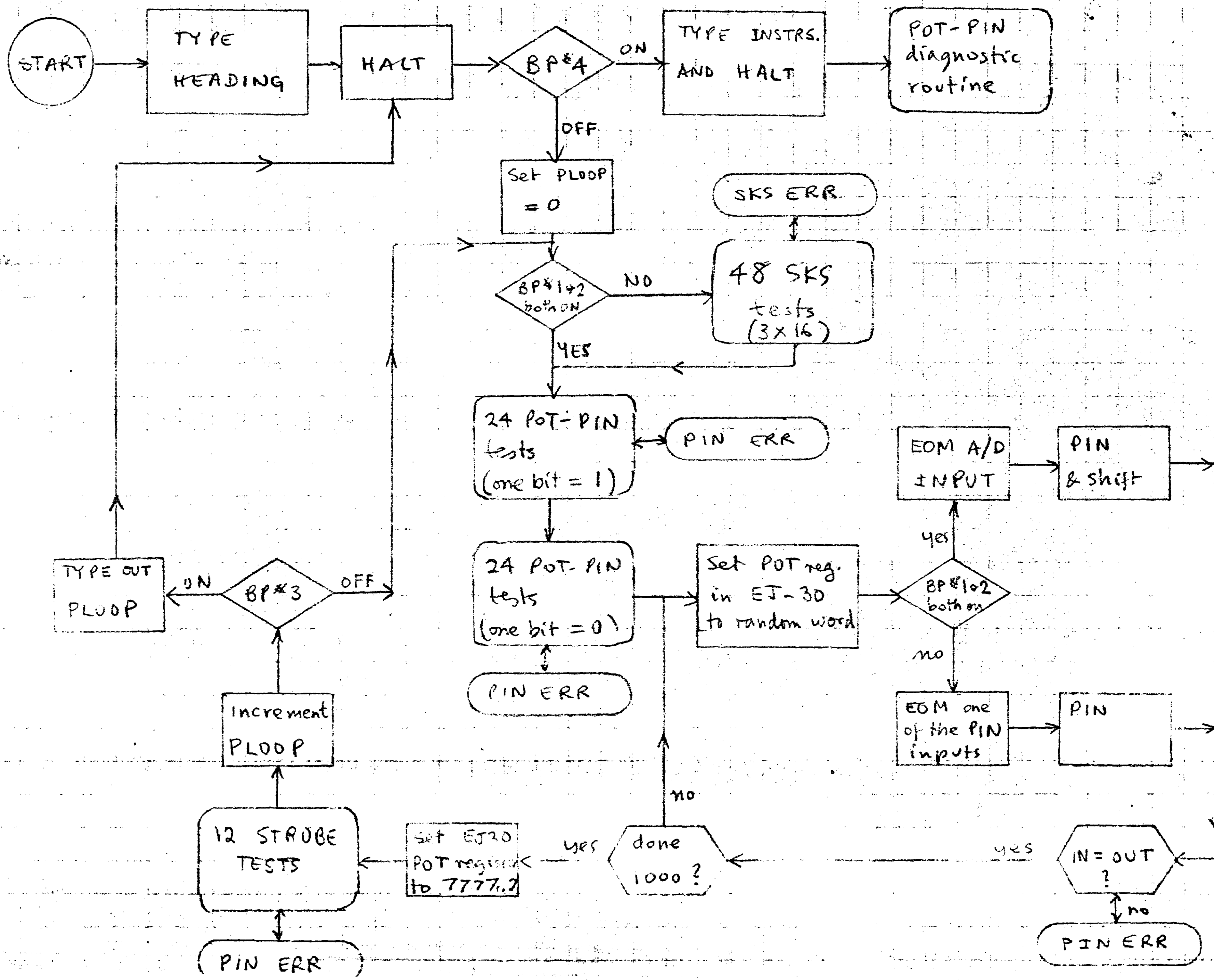
POT-PIN tests

POT-PIN test through ADC if cable is connected to PIN input No. 3

"Strobe" pulse tests

In the absence of errors, the program will keep repeating these tests until BP3 is set. It will then halt to allow the user to change connections.

General Flow in the EJ-30 Test Program



L. ISAAC
11-2-64

EJ30 CHANGES SO TEST WILL WORK OK

		A	J	E
DELETE	OP014	04	24	04
	OP015	4	24	04
	1F060	4	24	14
	2F060	4	24	15
	OALRT	4	24	17
	OP014	3	14	25
	OP015	3	14	22
ADD	2F060	4	24	17
	1AD14	4	18	34
	1AD15	4	18	35
	1AD14	3	14	25
	1AD15	3	14	22
DELETE	OT002	4	06	43
	GROUND	3	17	39
ADD	1T001	4	06	43
	OPOT1	3	17	39

L. Israel
26 Oct 1964

EJ30 CLOSED-LOOP TEST

Part No. 110047 is a cable for use in a closed-loop test of the EJ30. Pins 20-43 at the "input" end of the cable are connected directly to pin 20-43 at the "output" end of the cable. Also, pins 1-12 at the "output" end are connected respectively to pins 20-31 on that same cable card, and pins 1-16 at the "input" end are connected respectively to pins 20-35 on that same cable card.

1. The "output" end of the cable should be plugged to A04J05 of the EJ30 (POT output register connector).

2. When the "input" end of the cable is plugged to A04J20 of the EJ30 (PIN input # 1), the POT register is fed back to the # 1 PIN inputs, and the high order 16 bits of the POT register are fed back (respectively) to SKS test lines 00-15.

3. When the "input" end of the cable is plugged to A04 J22 of the EJ30 (PIN input # 2), the POT register is fed back to the # 2 PIN inputs, and the high order 16 bits of the POT register are fed back (respectively) to SKS test lines 16-31.

4. When the "input" end of the cable is plugged to A04J24 of the EJ30 (PIN input # 3), the POT register is fed back to the # 3 PIN inputs, and bits 05-12 of the POT register are fed back to bit positions 16-23 of the ADC input word. The remainder of the bits in the ADC input word will float at "1" in the absence of any grounding.

5. Activating the EOM strobe pulses will respectively reset to "0" the high-order 12 bits of the POT register, without regard to the location to which the "input" end is plugged.

This cable, and the test with which it is used, are designed to operate properly with the interim version of the EJ30 used for Autonetics, NBS, Ae[illegible]tron. However, the design specifications for the final version of the EJ30 should be such that no changes in either the cable or the test are required.

The EJ30 closed-loop test will have the following broad characteristics:

1. Only the digital I/O portion of the EJ30 will be checked.

2. Breakpoint switches 1 and 2 will represent the PIN input connector to which the test cable is plugged (binary coded).

3. In order to make the test attractive as an acceptance test, the number of successful loops made on the entire test will be printed out (typewriter) after each pass, whenever breakpoint 3 is set.

4. For test purposes only, breakpoint 4 may be set, causing the following sequence of actions: set POT register to contents of A, PIN, stall based on positive-valued contents of index register (large positive is longest stall), POT contents of B, PIN, stall based on X, and repeat this sequence indefinitely.

5. The test will perform the following sequence of actions:

   a. Reset POT register, check that no SKS tests are met.
   b. Set POT register, check that all SKS tests are met.
   c. Repeat a.
   d. In turn, set POT register to each configuration which has only a single bit set, and check PIN input.
   e. In turn, set POT register to each configuration which has only a single bit reset, and check PIN input.
   f. Set POT register to a large number of pseudo-random configurations in sequence, and check PIN input each time.
   g. Read in ADC input and check that data is as per specs; this should be in conjunction with test f.
   h. Set POT register to ones and issue the proper commands for the EOM strobe pulses out of the EJ30; check that bits 0-11 of the POT register were reset.

Tests a, b, and c should be bypassed when breakpoints 1 and 2 are in state 11. Test g should be included only when breakpoints 1 and 2 are in state 01 or 10.

The tests should be repeated indefinitely, subject to previous qualifications, tallying the number of loops. If any error is detected, the program should halt, and the contents of A, B and X should contain sufficient diagnostic information to specifically pinpoint the error. For instance, the sign bit of A could specify PIN error or SKS error, while B and X could contain the expected and actual test patterns involved (obviously different).