Return to
Ken Barkley

9000A-910
Utility Programs

User Manual

P/N 661447
MAY 1983

FLUKE®

Table of Contents

Section 1
Introduction

Many of the programs discussed in this manual are binary programs. A binary program is one that is compiled into Z80 machine code for execution in the 9010A/9005A. A binary program is compiled on a host computer and then downloaded to the 9010A/9005A, using the AUX I/F function. A binary program can be read/written from and to the cassette tape just like other programs. It can be uploaded/downloaded to or from a host computer and is executed in the same manner as other programs (including use as a subroutine of another program).

A binary program differs from a program written in the 9010A high-level language in that it cannot be modified or listed by the 9010A, because a binary program does not correspond to the 9010A high-level language test statements. If a listing is attempted (by pressing PROGM nn ENTER), the 9010A will display PROGRAM nn IS BINARY - ss BYTES, where nn is the program number and ss is the size of the binary program in bytes.

Binary programs are useful in that they provide detailed access to the internal workings of the 9010A/9005A. In many cases this allows us to add to the capabilities of the basic instrument, as is demonstrated by the programs described in this manual.

Section 2
Merge Tape Program
(2419 Bytes)

WHY MERGE TAPE?

Until now, there has been no easy way to move a single 9010A program from one tape to another. Once a program has been saved onto a 9010A cassette tape, that program is essentially "stuck" with the company of the other programs on that tape. It can be liberated from those other programs by deleting the programs, but then the new group of programs to reside with it on the tape must be entered manually.

There also has been no easy way to renumber the programs on a tape. To renumber a program, the 9010A user must manually reenter the program with the new number.

THE MERGE TAPE PROGRAM

A simple method is needed for reading specific programs from a tape and allowing the user to renumber them without the time-consuming and error-prone process of manual program reentry.

The 9010A Merge Tape program provides the means to do this. This program allows the user to read specific programs from a cassette tape and renumber them as desired without destroying the programs already in the 9010A memory.

PROGRAM RESTRICTIONS

There are, of course, a few restrictions imposed upon the user of the Merge Tape program. These restrictions are listed below.

1. The Merge Tape program occupies 2419 bytes of 9010A user memory, thereby reducing the memory available for the programs to be merged.

2. The user must know which programs need to be read from the source tape and which programs already exist in the 9010A memory.

3. The Merge Tape program has no provisions for helping the user to keep track of program numbers currently assigned, so the user must keep careful account of the renumbering of programs. It is possible to "lose" a program by renumbering two programs to the same number. (Only the last program assigned to the number will be copied.)

4. The Merge Tape program must be executed as program 0, so no user program can be copied as program 0.

NOTE

If program 0 on the source tape is the main executive program in a program set, this conflict can be circumvented by renumbering it to another available program number. Then, after deleting the Merge Tape program, a new one-line program 0 can be created which simply executes the renumbered main executive program.

LOADING THE MERGE TAPE PROGRAM

Loading the Merge Tape program is done with the usual READ TAPE operation. If there are unwanted programs residing on the tape with the Merge Tape program, they must be deleted from the 9010A memory after loading. (Side B of the Utility Programs tape contains the Merge Tape program alone to make this first step as quick and painless as possible.)

USING THE MERGE TAPE PROGRAM

First, let's define some terms that will be used in this discussion.

- Source Tape — The tape from which programs are being read.
- Source Program Number — The number of a program as stored on a source tape.
- Object Tape — The tape to which programs are being written.
- Object Program Number — The number of a program as stored on an object tape.
- Number Conflict — This occurs when the user tries to assign a number that is already in use in 9010A memory or has already been specified during the Merge Tape session.
- Insufficient Memory — This condition occurs when there is not sufficient available memory to add the indicated program to those already specified.

PROGRAM EXECUTION

To run the Merge Tape program, press EXEC 0 ENTER on the 9010A. The 9010A will then display (C)FLUKE VER-n.n LOAD CASSETTE. To determine the current program state, refer to Figure 2-1.

At this point the 9010A is waiting for the user to load a source tape. The source tape loaded at this point must remain in the 9010A tape drive until completion of the Read-Source-Tape Phase. After the source tape is placed in the 9010A tape drive, press CONT. A STOP or other Immediate Mode 9010A command (WRITE, LEARN, etc.) at this point will terminate execution of the Merge Tape program.

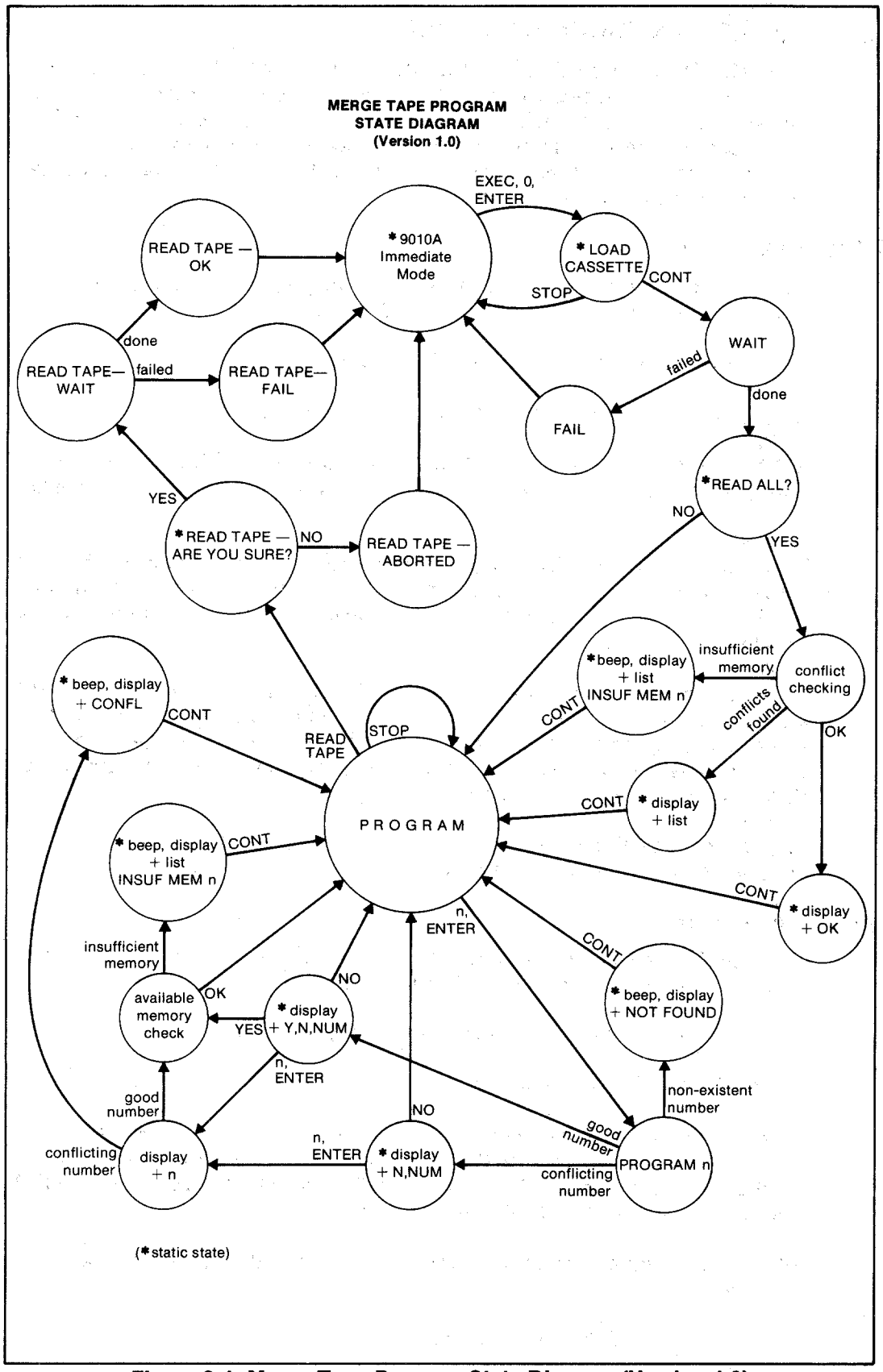

Figure 2-1. Merge Tape Program State Diagram (Version 1.0)

READ-SOURCE-TAPE-HEADER PHASE

When CONT has been detected, the 9010A will display WAIT and read the source tape header, which contains the program numbers and sizes. This takes only about 6 seconds, after which the 9010A proceeds to the Read-All Phase if the tape header was read correctly. (If an error was detected due to a bad tape, no data on the tape, or no tape in the tape drive, the 9010A will display FAIL and program execution will be terminated.)

READ-ALL PHASE

When the 9010A enters the Read-All Phase, it displays READ ALL? There are two correct responses to the READ ALL? question:

1. YES (to perform the Read-All operation)

2. NO (to proceed to the Specification Phase)

A YES response to the READ ALL? question tells the 9010A to specify all programs that are on the source tape. If there are no errors, the 9010A will display OK.

If there are number conflicts, the 9010A lists the conflicting program numbers (or the first 11 conflicting program numbers, if there are many conflicts). Number conflicts are not resolved at this point. If desired, they must be renumbered in the Specification Phase.

If insufficient memory is available, the 9010A beeps and displays INSUF MEM and the number of the program causing the condition. That program and all subsequent programs on the source tape will be ignored and not specified. (The operator may choose to enter some of those programs individually during the Specification Phase of the Merge Tape program.)

At the end of the Read-All Phase, the 9010A waits for a CONT response before continuing to the Specification Phase.

SPECIFICATION PHASE

When the 9010A enters the Specification Phase, it displays PROGRAM. The two correct responses to the PROGRAM prompt are:

1. A source program number (to specify a program)

2. READ TAPE (to proceed to the Read-Source-Tape Phase)

When a program number is entered, the 9010A first checks that there is indeed a program with that number on the source tape. If there is no such program, the 9010A beeps and displays NOT FOUND. The 9010A then waits for a CONT before returning to the PROGRAM prompt.

When a valid source program number is entered, the 9010A displays one of two messages:

1. N,NUM, if there is a number conflict

2. Y,N,NUM, if there is no conflict

Y indicates that YES is a legal response, N indicates a NO response, and NUM indicates an object program number.

A YES response, when valid, specifies that the given source program be entered with the same number as on the source tape. The 9010A checks for an insufficient memory condition, and enters the program specification if no error is encountered.

A NO response does not specify the given source program number, but it removes any specification previously entered for that number.

If an object program number is entered, then the 9010A checks its validity. If there is a number conflict, the 9010A displays CONFL. If an insufficient memory condition is detected, the 9010A beeps and displays INSUF MEM. After either of these error messages, the 9010A waits for a CONT response before returning to the PROGRAM prompt.

When the object tape contents have been specified as desired, a READ TAPE response to the PROGRAM prompt will cause the 9010A to proceed to the Read-Source-Tape Phase.

READ-SOURCE-TAPE PHASE

Upon entering the Read-Source-Tape Phase, the 9010A displays the familiar READ TAPE - ARE YOU SURE? message.

A NO response is the operator's opportunity to abort. The 9010A will display READ TAPE ABORTED, and the 9010A memory will be returned to its condition before the Merge Tape program was executed.

A YES response will cause the 9010A to read only the specified programs from the source tape. If the READ TAPE operation encounters an error, the 9010A will display READ TAPE FAIL and will return to Immediate Mode. If the operation is performed successfully, the 9010A will display READ TAPE OK and will return to Immediate Mode.

CAUTION

It is important that the source tape (which was loaded for the Read-Tape-Header Phase) be loaded into the 9010A tape drive before the YES response. If any other tape is in the drive, the 9010A memory will be corrupted, with unpredictably catastrophic results. If this happens, turn the 9010A power OFF and then ON and start over. The source tape will not be damaged.

WRITE-OBJECT-TAPE PHASE

This phase is performed independently of the Merge Tape program. When the 9010A program memory contents appear as desired for the object tape, insert the object tape into the 9010A tape drive. In 9010A Immediate Mode, perform a WRITE TAPE operation to transfer all programs currently in 9010A memory onto the object tape.

Section 3
Register Arithmetic Programs

INTRODUCTION

The Register Arithmetic programs provide the means for performing simple addition and subtraction of 9010A register contents. The operations are performed the same as those on unsigned 32-bit integers (which is consistent with normal usage of the registers in the 9010A). This means that the results of the addition or subtraction are carried through zero without regard to overflow or borrow conditions, respectively.

The Register Arithmetic programs operate on the contents of the registers specifed by the programmer, according to the value of register B. If Reg B = 362 (hex), then register 3 will be set to the result of performing the addition or subtraction operation on registers 6 and 2 (r3 = r6 ± r2). The registers used as input to the Register Arithmetic programs are unchanged with one exception. If desired, one of the input registers can be used as the output register also (i.e., REGB = 662 or REGB = 262). In that case, its value will be changed to the result value.

The two Register Arithmetic programs are described individually below. Execution time for either program is 31 ms.

32-BIT UNSIGNED ADD (102 BYTES)

This program will perform an addition of the contents of two 9010A registers and will place the sum in a third register. For example, if the Unsigned Add program is program 10, then

```
REG6 = FF
REG2 = 1
REGB = 362
EXECUTE PROGRAM 10
```

will set register 3 equal to register 6 (value 1). Thus, register 3 will be set to 100. The register numbers (i.e., REG6, REG2, and REG3) are examples only. Any register numbers can be used.

Program 1 demonstrates the use of the 32-Bit Unsigned Add binary program. It asks for two hexadecimal numbers and displays their sum. (Remember that overflow is ignored.)

Section 4
Frequency Counter Program
(207 Bytes)

INTRODUCTION

The Frequency Counter program provides a means of measuring the frequency of a signal with the 9010A probe. Its frequency range is from 10 Hz to 6 MHz. The program is executed as any other 9010A program. Each time the program is executed the frequency reading is returned in register 8 and the annunciator value (0 = Hz, 1 = kHz) in register 9. For example, if the Frequency Counter program returns register 8 = 950 and register 9 = 1, then the frequency reading is 950 kHz.

HOW IT WORKS

The Frequency Counter program reads a count from the event counter and then converts that value to a frequency. The counter is implemented by generating a software gate which first resets the event counter and then waits a specified number of machine cycles before reading the event counter. The following software gate times are tested in the order shown:

10 μs, 100 μs, 1 ms, 10 ms, 100 ms, and 1 sec

The gate time starts at 10 μs, and, if the count is 9 or less, the gate time is increased by a factor of 10 and another reading is taken. This process continues until either a count of 10 or greater is read or the 1-second gate time is reached. If the count is 10 or greater, the frequency is computed from the following formula:

frequency = N/(gate time), where N = number of counts

The result is a resolution of from 100 kHz down to 1 Hz. (Any frequency less than 10 Hz will produce a reading of 0 Hz.)

CROSSING GATE BOUNDARIES

When using the Frequency Counter program at frequencies that cause the gate to change ranges (counts which alternate between 9 and 10), there is a possibility that the gate will toggle between the two ranges. The following example will help to clarify this.

Assume you are trying to measure a frequency of 950 kHz. A 10 μs gate time will generate a count of the closest integer to

$N = (950)*(10*10^{-6}) = 9.5$

The count will be either 9 or 10, depending on the phase of the input signal with respect to the microprocessor clock. If the count is 10, it is not necessary to increment the gate time, and the corresponding frequency is calculated to be 1000 kHz. If the count is 9, however, the gate time will be incremented (to 100 μs) and a count of 95 will be obtained, yielding a frequency of 950 kHz.

As a result of this, as the phase of the input signal changes (as it will unless it is actually synchronized to the microprocessor clock), the gate time will toggle between 10 μs and 100 μs. The frequency reading alternates between 950 kHz and 1000 kHz, indicating an apparent jitter of 50 kHz, but all that is really happening is a resolution shift between 10 kHz and 100 kHz.

DEMO PROGRAM

Program 3 demonstrates the use of the Frequency Counter binary program. It repetitively displays frequency readings, providing a real-time frequency readout.

Program 3: Frequency Counter Demo (51 bytes)

```
1 LABEL 1
  EXECUTE PROGRAM 30          ;take frequency reading
  DPY - FREQUENCY = @8~
  IF REG9 = 0 GOTO 2          ;Hz?

  DPY - +K

2 LABEL 2
  DPY - +HZ
  GOTO 1
```

Section 5
Setup Program
(248 Bytes)

INTRODUCTION

The Setup program allows the user to change certain 9010A setup parameters under program control. These parameters are Enable Mask, Trap Mask, Pod Timeout, Exercise-Errors Flag, and Beep-On-Error Flag.

The program is executed as any other 9010A program, with register 9 indicating the parameter to be changed and register 8 indicating the desired value of the parameter. See Table 5-1 for function codes.

Table 5-1. Setup Functions

REG 9 VALUE	FUNCTION
0	Set Enable Mask to value in register 8 (see Table 5-3)
1	Set Trap Mask to value in register 8 (see Table 5-2)
2	Set Pod Timeout to value of register 8 (max. = 65535 dec.)
3	Set Exercise-Errors *
4	Set Beep-On-Error *
*Register 8 = 0 means NO; any other value means YES	

Table 5-2. Trap Mask

REG 8 BIT	FUNCTION (1 = yes, 0 = no)	REG 8 BIT	FUNCTION (1 = yes, 0 = no)
0	Set Trap on Data Error	4	Set Trap on Active Interrupt
1	Set Trap on Address Error	5	Set Trap on Illegal Address
2	Set Trap on Control Error	6	(not used)
3	Set Trap on Active Force Line	7	Set Trap on Bad Power Supply

DEMO PROGRAM

Program 4 demonstrates the use of the Setup binary program. After each setup operation, the user can use the 9010A SETUP key to verify that the parameter of interest has indeed been changed.

For example, if a Z80 Interface Pod is being used, the following procedure will enable the WAIT line only. Press EXEC 4 ENTER to execute the Setup demonstration program. The 9010A will respond with SETUP PROGRAM -- REG9 =. From Table 5-1, we see that to set the enable mask, the register 9 value should be 0. Press 0 ENTER and the 9010A displays ENABLE MASK = and waits for the mask value for register 8. From Table 5-3, we find that the WAIT bit is bit 5 of the Z80 enable mask (register 8). Therefore, the mask value to enable WAIT only is 20 (hex). Press 2 0 ENTER. The program then terminates and the 9010A returns to Immediate Mode. The setup parameters can now be checked to verify that the WAIT line was indeed enabled.

Program 4: Setup Demo (191 bytes)

```
  DPY - #SETUP PROGRAM -- REG9 = /9
  IF REG9 = 1 GOTO 1      ;set trap mask?

  IF REG9 = 2 GOTO 2      ;set pod timeout?

  IF REG9 = 3 GOTO 3      ;set exercise-errors flag?

  IF REG9 = 4 GOTO 4      ;set beep-on-error flag?

  DPY - #ENABLE MASK = /8
  GOTO F

1 LABEL 1
  DPY - #TRAP MASK = /8
  GOTO F

2 LABEL 2
  DPY - #POD TIMEOUT = /8
  GOTO F

3 LABEL 3
  DPY - EXERCISE ERRORS ?8
  GOTO F

4 LABEL 4
  DPY - #BEEP-ON-ERROR ?8

F LABEL F
  EXECUTE PROGRAM 40      ;change setup
```

Section 6
Probe Pulser Program
(144 Bytes)

INTRODUCTION

The Probe Pulser program provides a means for controlling the probe stimulus while under program control. The value of register 8 dictates what probe pulser condition is to be set as shown in Table 6-1. For example, if the Probe Pulser program is program 50, then

```
REG8 = 2
EXECUTE PROGRAM 50
```

will cause the probe to pulse high only.

Table 6-1. Register 8 Bit Assignments

BIT	FUNCTION
0	Pulse low
1	Pulse high
(0 = no, 1 = yes)	

SPECIAL RESTRICTIONS

There are two special restrictions to remember:

1. The HIGH and LOW keys on the 9010A cannot be disabled, and they will affect pulser condition whenever they are used. The HIGH and LOW keys must not be pressed while the Probe Pulser program is executing.

2. The programmer should insure that the Probe Pulser program disables the pulser before returning to Immediate Mode.

If these restrictions are violated, the 9010A pulser status may, as a result, "disagree" with the HIGH and LOW keys (e.g., the 9010A pulsing high with the HIGH key released). This condition can be encountered either while executing a program (situation 1 above) or in Immediate Mode (situation 2 above).

NOTE

If the 9010A is synchronized Address or Data, the pulser will not actually generate a pulse while the 9010A is idle. To check for disagreement between the pulser output and the HIGH and LOW keys, the 9010A must be put into FREE-RUN mode.

To correct the problem while executing a program, simply force the desired pulsing mode using the Probe Pulser program. The situation will be corrected automatically at the next execution of the Probe Pulser program. If, however, immediate recovery is required and the program being executed does not allow interactive execution of the Probe Pulser program, then program execution will have to be terminated and recovery must be accomplished according to the following Immediate Mode procedure.

To correct the problem while in Immediate Mode, simply toggle the HIGH and LOW keys as required. For example, if the HIGH and LOW keys are released but the 9010A is toggling low, the recovery procedure would be as follows. First, set the LOW key (pulsing condition will be unchanged). Then release the LOW key, at which time the low pulsing will cease. Setting and then releasing both the HIGH and LOW keys will be sufficient to recover from any state.

DEMO PROGRAM

Program 5 demonstrates the use of the Probe Pulser binary program. To use this program, connect a probe to the 9010A, release the HIGH and LOW keys, and press SYNC F to put the 9010A into Free-Run mode. Then press EXEC 5 ENTER. The 9010A will display PULSER MODE? (F=EXIT). At this point the 9010A is waiting for you to enter a value to be sent to the Probe Pulser program, which will check bits 0 and 1 to determine the pulser mode (as shown in Table 6-2).

Table 6-2. Probe Pulser Demo Inputs

PRESS	PULSER STATUS
0 ENTER	No pulsing
1 ENTER	Pulsing low
2 ENTER	Pulsing high
3 ENTER	Pulsing low and high

To exit from the program, press F ENTER. The 9010A will then disable the pulser and terminate the program.

Program 5: Probe Pulser Demo (60 bytes)

```
0 LABEL 0
  DPY - #PULSER MODE? (F=EXIT) /8
  IF REG8 = F GOTO F                ;exit?

  EXECUTE PROGRAM 50                ;set pulser mode
  GOTO 0

F LABEL F
  REG8 = 0
  EXECUTE PROGRAM 50                ;disable pulser
```

Section 7
Pod Self-Test Program
(2646 Bytes)

INTRODUCTION

The Pod Self-Test program provides a complete self-test of the particular interface pod connected to the 9010A. The tests performed are described in Table 7-1. They are executed in the order shown until either of the following occurs:

1. A failure is detected.

2. The pod is removed from the self-test mode.

When the pod is removed from the self-test mode, the instrument software enters the READY mode, as indicated by FLUKE 9000 READY on the 9010A display.

ERROR REPORTING

When a failure is detected, it is reported on the 9010A display. Immediately after a failure, testing is begun again at Test 0. This means that the error message on the 9010A display is constantly being updated.

When the fault byte is reported, active force lines, active interrupts, and illegal addresses are ignored. The error codes are as shown in Table 7-1.

NOTE

There is a special consideration to be remembered when using the Pod Self-Test program with the Z80 Interface Pod. The I register, available at special address 20000, is set to value FF when this program is run. Therefore, if your Unit Under Test uses the I register, you may need to reinitialize it to the appropriate value.

Table 7-1. Error Codes

ERROR CODE	ATTEMPTED ACTION	POSSIBLE FAULT(S)
000	Reset and initialize pod. Read @ 0FF00FF0. (Expected data = 0FF0F00F .)	A,B,C,D
001	Write @ 0FF00F0 = F00F0FF0 .	A,B,C,D
002	Test all control lines.	B
003	Disable all lines that can be enabled.	F
004	Set address sync mode.	A,F
005	Read @ 0000F00F. (Expected data = 00000FF0 .)	A,B,C,D
006	Write @ 00000001 = FFFFFFFE .	A,B,C,E
007	Write @ 00000002 = FFFFFFFD .	A,B,C,E
.	.	.
.	.	.
.	.	.
.	.	.
037	Write @ 80000000 = 7FFFFFFF .	A,B,C,E
038	Enable first pod enable line.	F
039	Enable second pod enable line.	F
.	.	.
.	.	.
.	.	.
.	.	.
037 + n	Enable last pod enable line.	F

NOTE: The data values actually depend on the number of data lines on the interface pod, (i.e., an 8-bit pod only reads or writes the 8 least-significant bits, a 16-bit pod the 16 least-significant bits, and a 32-bit pod all 32 bits).

Fault Codes:
- A — UUT power-sensing circuit failure.
- B — Control line(s) cannot be driven.
- C — Address line(s) cannot be driven.
- D — Wrong data read.
- E — Data line(s) cannot be driven.
- F — Failure in forcing- or interrupt-line buffer(s) or associated logic.

Appendix A
Program List

SIDE A

Program 0:	Merge Tape (binary program)
Program 1:	Unsigned Add Demo This program demonstrates the use of the Unsigned Add program (program 10).
Program 2:	Unsigned Subtract Demo This program demonstrates the use of the Unsigned Subtract program (program 20).
Program 3:	Frequency Counter Demo This program demonstrates the use of the Frequency Counter program (program 30).
Program 4:	Setup Demo This program demonstrates the use of the Setup program (program 40).
Program 5:	Probe Pulser Demo This program demonstrates the use of the Probe Pulser program (program 50).
Program 10:	Unsigned 32-Bit Add (binary program)
Program 20:	Unsigned 32-Bit Subtract (binary program)
Program 30:	Frequency Counter (binary program)
Program 40:	Setup (binary program)
Program 50:	Probe Pulser (binary program)
Program 60:	Pod Self-Test (binary program)
Program 99:	Content Listing This program contains a list of all programs on the 9010A Utility Programs tape, side A.

SIDE B

Program 0:	Merge Tape (binary program)