FLUKE.

TroubleShooter

For Digital Board Test Applications

Volume 7, number 2
Winter 1990

Inside the Troubleshooter

Create an Exclusive-OR Binary Program

By Kirk Schuetz and Gary Miles
Planning Research Corporation

Although the Fluke 9010A Micro-System Troubleshooter is a very flexible piece of test equipment, you can expand its capabilities by creating your own binary programs. In the Fall 1989 issue of the Troubleshooter, I described a procedure for creating binary programs. In this issue, I'll show how to use binary programs to calculate the output from an exclusive-OR of two registers. Applying the technique described in the first article, use the hexadecimal values for the machine instructions in the first column of Listing 1 to create the exclusive-OR binary program—Program 15 from the first article. (If you need a copy of the Fall 1989 issue, contact Fluke at 1-800-44-FLUKE.)

Notice that three digits in Local Register 0 are used to specify the destination and source registers: REG0 = dest src src. The program uses Register 0 because it is the first register pointed to, and the easiest to find. Notice also that local registers may be used as source and destination.

The following 9010A programs illustrate several practical uses for the exclusive-OR binary program (Program 15). In Program 16 below, the results of the XOR calculation should be:

12345678 XOR 87654321 = 95511559.

```
PROGRAM 16                  ! example
REG1 = 12345678             ! initialize values
REG2 = 87654321E
REG0 = 321                  ! destination, source, source
EXECUTE PROGRAM 15          ! xor binary
DPY-$1 XOR $2 = $3
```

Using the exclusive-OR program, you can easily compare expected and actual results. If the results are the same, the destination register will hold a zero; if the results are different, the failing bit in the destination register will be set to one.

This method uses about the same amount of memory as alternate methods, but has two advantages: it determines which bit failed, and it is not limited by the number of labels available. One alternate method is to use two IF statements per comparison, one to see if the actual data is greater than the expected, and the other to see if it is less. The other alternate method is to use a label for each comparison, with each IF statement branching to the next label. See Program 17.

```
PROGRAM 17
0: LABEL 0
REG2 = 6D                   ! expected data
READ @ 1004                 ! test address
REG0 = 12E                  ! dest=1, source=2,E
EXECUTE PROGRAM 15          ! xor binary
IF REG1 > 0 GO TO E         ! test to see if it passed
REG2 = 4A                   ! expected data
READ @ 1005                 ! test address
EXECUTE PROGRAM 15          ! xor binary
IF REG1 > 0 GOTO E          ! test to see if it passed
GOTO F                      ! passed
E:LABEL E                   ! error
DPY-FAIL#
STOP
GOTO 0                      ! repeat test
F:LABEL                     ! finish
```

As Program 18 shows, the XOR binary program can be used to change the state of a bit even if the state of the bit is not known.

```
PROGRAM 18                  ! change state of bit
READ @ 1000                 ! get data
REG1 = 80                   ! want to change the state of bit 7
REG0 = E1E                  ! dest=E, source-1,E
EXECUTE PROGRAM 15          ! xor binary
WRITE @ REGF = REGE         ! write new data
```

Continued on page 2

Cover Story

(Continued from Cover)

As Program 19 shows, the XOR binary program can also be used to swap the contents of two registers without using a third register. This technique is useful if you have no more registers to use.

```
PROGRAM 19                ! swap two registers
REG1 = 55                 ! initialize values
REG2 = AA
DPY=REG1=$1 REG2=$2
REG0 = 212
EXECUTE PROGRAM 15        ! xor binary
REG0 = 112
EXECUTE PROGRAM 15        ! xor binary
REG0 = 212
EXECUTE PROGRAM 15        ! xor binary
DPY-+ REG1=$1 REG2=$2
```

The uses of binary programs are almost limitless. Because the designers had the foresight to include this capability in the 9010A, they expanded the range of an already good instrument.

```
LISTING 1
                          ! XOR
!THIS PROGRAM PROVIDES A WAY TO XOR 2 FLUKE REGISTERS

This                      ! REGISTER 0 IS USED TO SPECIFY THE
column                    ! DESTINATION AND BOTH SOURCE REGISTERS
contains
hexadecimal               ! BITS 0 THRU 3 ARE A SOURCE REGISTER
machine                   ! BITS 4 THRU 7 ARE A SOURCE REGISTER
instructions              ! BITS 8 THRU 11 ARE THE DESTINATION REGISTER
```

```
Listing 1 continued

514F50 BYTE 51H,4FH,50H   ! NECESSARY FOR THE FLUKE
2A2B00    LD  HL,(002BH)  ! BASE ADDR FOR FLUKE REGISTERS
54        LD  D,H
5D        LD  E,L
23        INC HL          ! POINT HL AT REG 0 (SOURCE REGISTERS)
23        INC HL
AF        XOR A
0602      LD  B,02H       ! B = LOOP COUNT FOR LOOP 1
0E02 LOOP1 LD C,02H       ! C = LOOP COUNT FOR LOOP 2
ED6F LOOP2 RLD            ! MOVE UPPER 4 BITS OF (HL) TO LOWER 4 BITS
                          ! OF REG A, LOWER 4 BITS OF (HL) TO UPPER
                          ! 4 BITS OF (HL), LOWER 4 BITS OF REG A TO
                          ! LOWER 4 BITS OF (HL)
E5        PUSH HL
6F        LD  L,A         ! A = REGISTER NUMBER
2600      LD  H,00H
29        ADD HL,HL       ! REGISTER NUMBER X 4
29        ADD HL,HL
19        ADD HL,DE       ! HL = ADDR OF 2ND MSB OF REGISTER
E3        EX  HL,(SP)     ! REG ADDR ON STACK
0D        DEC C           ! DECREMENT LOOP 2 COUNT
20F3      JR  NZ,LOOP2
ED6F      RLD             ! RESTORE REG 0 BYTE TO ORIGINAL STATE
23        INC HL          ! HL POINTS TO DEST REG NUMBER IN REG 0
10EC      DJNZ LOOP1      ! DECREMENT LOOP 1 COUNT
D1        POP DE          ! DE = DEST REGISTER ADDR
E1        POP HL          ! POP ADDR CREATED BY BITS 12 THRU 15 OF REG 0
E1        POP HL          ! HL = SOURCE REGISTER ADDR
0604      LD  B,04H       ! B = LOOP COUNT FOR LOOP 3
7E  LOOP3 LD  A,(HL)      ! LOAD A AT ONE SOURCE XOR WITH THE OTHER
23        INC HL          ! SOURCE AND STORE AT IN THE DEST REGISTER
E3        EX  HL,(SP)
AE        XOR (HL)
23        INC HL
12        LD  (DE),A
13        INC DE
10F7      DJNZ LOOP3
E1        POP HL          ! POP SOURCE ADDR FROM STACK
C9        RET
```

Winter Course Schedule

Training Pays Off in Productivity

To gain maximum performance from your Fluke equipment, spend just a few days training with experts from the company that designed and built the equipment. Enrolling in a Fluke training course is as easy as calling one of the Training Coordinators or the 800 number listed below. They can help you determine your training needs and select the appropriate course. Or ask for a course planner if you need more information on course objectives or prerequisites.

Course	December	January	February	March
9010A Troubleshooting Catalog #TAF9000 2 Days, $595	5 San Francisco, CA	8,9 Dallas, TX 15,16 Orlando, FL 22,23 Los Angeles, CA 22,23 Calgary, AB	5,6 New York, NY 19,20 Seattle, WA	5,6 Chicago, IL 12,13 San Francisco, CA 19,20 Atlanta, GA 26,27 Phoenix, AZ
9010A Programming Catalog #TAF9001 2 Days, $595	7 San Francisco, CA	10,11 Dallas, TX 17,18 Orlando, FL 24,25 Los Angeles, CA 24,25 Calgary, AB	7,8 New York, NY 21,22 Seattle, WA	7,8 Chicago, IL 14,15 San Francisco, CA 21,22 Atlanta, GA 28,29 Phoenix, AZ
9100 Part I Catalog #TAF9102 4 Days, $1195	11-14 Orlando, FL	8-11 Los Angeles, CA 22-25 Chicago, IL 29-1 Seattle, WA	5-8 Orlando, FL 19-22 Los Angeles, CA	5-8 Chicago, IL 19-22 Orlando, FL
9100A Part II Catalog #TAF9103 5 Days, $1495	3-7 Chicago, IL 17-21 Orlando, FL	14-18 Los Angeles, CA 28-1 Chicago, IL	11-15 Orlando, FL 25-1 Los Angeles, CA 4-8 Seattle, WA	11-15 Chicago, IL 25-29 Orlando, FL
900 Troubleshooting Catalog #TGF900 2 Days, $595	11,12 Orlando, FL	15,16 San Francisco	19,20 Dallas, TX	19,20 Los Angeles, CA

Prices are U.S. List.

Training
1-800-44-FLUKE
ext. 73

Training Coordinator
Everett, WA
(206) 356-6330

Training Coordinator
Palatine, IL
(708) 705-0500

Canadian Classes
Toronto, ON
(416) 890-7600

9100A Adds Support for RISC Chips And Two Other Chip Families

Users of boards incorporating the Reduced Instruction Set Computing (RISC) architecture can now use the Fluke 9100A Digital Test System to test such boards. The 9132A-R2000 Processor Support Package adapts the 9100A for full emulative testing of boards based on the R2000 RISC microprocessor. The package tests both the 12 MIPS and 16 MIPS versions of the R2000, which is used primarily in engineering workstations.

Other new Processor Support Packages include: the 9132A-68HC11, which adapts the 9100A to boards containing the 68HC11A0, 68HC11A1, 68HC11E0, and 68HC11E1; and the 9132A-80C186, which adapts the 9100A to boards containing the Intel 80C186, 80186, 80C188, and 80188. The 68HC11 family chips are found primarily in automotive applications, and the Intel family chips are used as embedded controllers in a wide variety of applications.

In all configurations, the 9100A connects to the UUT through the 9132A Memory Interface Pod, and the Processor Support Package configures the 9132A for the testing requirements of the specific microprocessor. In addition to standard test routines, each package includes a micro-floppy containing advanced diagnostics for detecting kernel hardware faults.

The new Processor Support Packages are available on an eight-week delivery schedule. The packages are available separately, or as part of a test configuration that also includes one 9132A Memory Interface Pod, two ROM modules, and two RAM modules.

Learn and Auto-Test Software Available for the 9100A

Some previous users of the 9010A who are currently using the 9100A have asked whether the 9100A can perform the 9010A's Learn feature on unknown UUTs. Yes, it can. Now you can not only learn RAM and ROM descriptors in blocks of 16 bytes up to 4 Mbytes, but you can execute BUS, RAM, and ROM tests in a single keystroke—called Auto-Test—for easier kernel testing in the Immediate Mode.

And it's free! Just call 1-800-44-FLUKE and ask for the 9100A Learn and Auto-Test software. You'll receive a 9100A user disk that includes the software and a README text file that explains how to use it.

New CAD Translators for the 9100A

A 9100A application engineer in Fluke's Atlanta office has developed CAD translation software for SCHEMA, CALAY, TELESIS, and RACAL-REDAC. You perform translations on a PC by using a program written in QuickBASIC. After translating your netlist, you can download it to the 9100A. To receive a free MS-DOS disk that contains the translation programs, sample netlists, and a README text file, call 1-800-44-FLUKE and ask for the 9100A CAD Translator disk. QuickBASIC source listings are also available on request, should you need to modify the programs for your application.

Software Allows Control of the Fluke 900 From a PC

A new software package allows the Fluke 900 Dynamic Troubleshooter to be controlled from an MS-DOS personal computer. With the 900 PC Software Package, users can organize and edit both data and program files from the PC keyboard, using the PC's disk space for storage. The 900 PC Software simplifies test development by allowing users to choose their own text editor; this feature makes operation easier and reduces development times. The software also offers a friendly user interface that includes pull-down menus and pop-up windows. After developing a test, the operator can either execute the test right from the PC Software or download it to the 900 for stand-alone execution. The 900 PC Software Package is available on an eight-week delivery schedule.

Applications

Making Pulse Width Measurements With the Fluke 9100A

Robert S. Murphy, ISI Services, Inc.
Anthony H. Phan, Hughes Aircraft Co.

While developing tests for the Fluke 9100A Test System, you may want to measure the pulse width of a particular signal. Sometimes these signals are generated by an asynchronous source such as a one-shot. A transition count or level history will tell you that the signal has toggled, but it won't tell you much more. A signature is useful only if you have a signal that is synchronous to the clocks in the UUT. However, by using the triggering capabilities of the 9100A-003 I/O Module, a known clock source, and an external gate, you can determine the signal's pulse width to within two clock periods. This technique can be useful for testing watch-dog timers, one-shot outputs, power-on reset pulses, and even low-duty-cycle synchronous signal pulse widths if pulse duration is your measurement of interest.

Measuring the Pulse

To make a pulse measurement, you can use the leading and trailing edges of a pulse to gate a clock of a known frequency, and then count the number of clock transitions to determine the period of the pulse. See Figure 1. The more clock periods that occur during the gate period, the more accurate your measurement will be.

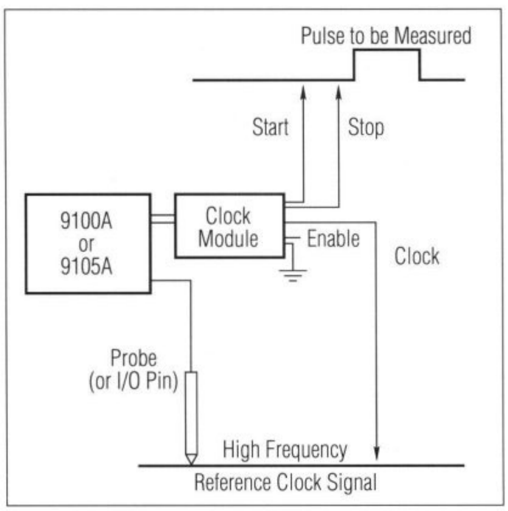


Figure 1. Pulse width can be measured when reference to a clock of a known frequency

To make pulse width measurements with the 9100A, use External Sync and connect the Start and Stop lines to the pulse to be measured; this forms the gate period. Next, connect a clock of a known frequency to the probe or to a pin of the I/O module to count the transitions of the reference clock. If you have a Vector Output I/O module, you can use it to generate the reference frequency as well as count the transitions. Simply connect one of the I/O pins to the INT CLK pin on the side of the I/O module. The pulse width is calculated by multiplying the transition count by the clock period.

If you use the internal clock of the Vector Output I/O module as your reference, selecting 1 MHz will allow you to display the transition count directly in microseconds. Or, for greater accuracy (as the example below illustrates), select 10 MHz and divide the count by ten for a reading in microseconds. If you need greater resolution, use a faster clock with a 40 MHz probe, and you'll improve the resolution to better than 25 nanoseconds. Remember that the accuracy of this measurement depends on the accuracy of the reference clock, plus two counts (actually somewhat less than one clock period at both ends of the gate period). In time units the counts would equal two times the clock period. So be sure the number of clock periods during the gate period (the more the better) is high enough so that losing two counts will not significantly affect your measurement. I think you'll find this pulse measurement technique useful in a variety of applications. You'll also eliminate the need for another piece of test equipment on your bench.

```
global numeric c
global numeric start
global numeric finish
global numeric time
end declare

t1 = open device "/term1", as "output", mode "unbuffered"
m1 = "/mod1"
m2 = "/mod2"


reset (m1)
reset (m2)

clockfreq device m1, freq "10MHz"
sync device m1, mode "ext"
counter device m1, mode "transition"
enable device m1, mode "always"
edge device m1, start "-", stop "+"


arm device m1
writepin device m2, pin 1, level "0", mode "latch"
wait time 500
writepin device m2, pin 1, level "1", mode "latch"

start = system ()
loop until c = $C or time = 5
c = checkstatus device m1
finish = systime ()
time = finish - start
end loop

readout device m1


width = (count device m1, pin 1) /10
print on t1, "Pulse Width = ", width, " usecs"
wait time 2500

end program
```

Fluke Products

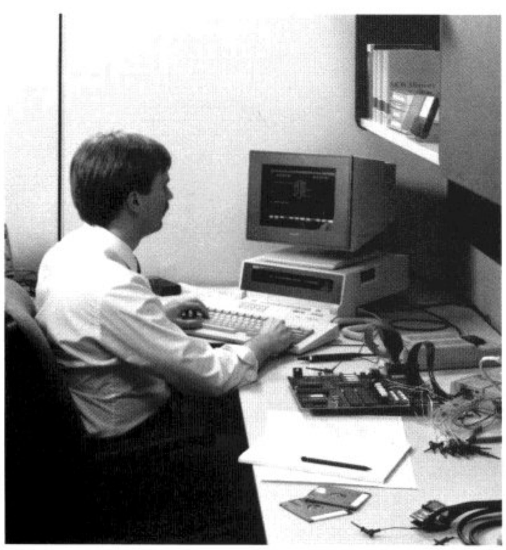
Increase Your Proficiency In 9100A Programming

An east coast contract-manufacturing firm recently had the chance to test its experience in, and philosophy of, functional testing. In just two months, Atlantic Design (Binghamton, NY) used the Fluke 9100A to develop functional tests for 23 circuit boards used in an aircraft simulation system by General Electric Corporation. As part of Atlantic Design's turnkey contract-manufacturing services, it provides both in-circuit and functional testing of new boards for its customers. Rick Travis, Atlantic Design's Test Engineering Manager, says that the company's test development methods are "really straightforward." There's no magic solution, he says, for developing comprehensive functional tests for major circuit boards—even at a rate of one every two or three days. Instead, he says, Atlantic Design approaches the challenge logically, using four principles to guide them:

- They make all of their test routines as generic as possible.
- They partition circuit boards for ease of testing.
- They divide the task logically and assign two people to it.
- They adapt the 9100A to test non-microprocessor-based boards and use the 9100A as a controller when using GPIB or other test equipment.

Generic Routines

As Travis says, microprocessor-based boards incorporate many of the same functions. "Because the same functions and components have to be tested again and again, we write a generic test for them. Then, in testing a new board, generally only a minor modification to the test is required. If you already have good memory routines, why would you create new ones?" Intelligent partitioning of the board is a key element, says Travis. "To test any micro-based board you have to look at data lines and address lines," he says. "Most of these boards have memory and some kind of clock module. If you partition the board by these common elements, it's much easier to write a generic test for each functional element." Similarly, he says, a large test file would require a lot of editing; by using smaller files for smaller partitions, they have created a library of modular files that can be transferred from one application to another will little modification.



Divide and Conquer

At least two distinct tasks are required to create a 9100A program, says Travis, and that's why he typically assigns two people to the task. One task is to create GFI test routines for the board and to input all the data pertinent to the board—interconnect diagrams, parts libraries, and so on. The other task is to create the stimulus routines that actually operate the board. "To us it makes more sense to use two people. While the first one is writing stimulus routines, the second is creating the GFI routines and inputting preliminary data." The process, he says, makes for efficient test generation. "It allows us to build a complete GFI (guided fault isolation) routine for a board in very little time."

Testing the Imagination

According to Travis, all it takes is a little imagination to provide more efficient functional testing. "Part of our effectiveness has come simply from adapting the 9100A to non-microprocessor-based boards," he says. "For example, we built an adapter for an 80286-based memory board—a board with several megabytes of memory but no processor. It took us a day. Then we tested it out; that took one more day. Now we use the set-up, along with the 9100A's HyperRAM testing algorithm, for high-speed testing of these boards. "We also have used the 9100A for dual purposes in a single application—as both a tester and a controller for part-analog boards. Functional testing of the digital portions is handled by the 9100A. Then, to test the analog circuitry, we use the 9100A as a controller for a GPIB instrument; for the analog tests, the 9100A passes the baton to the analog tester while maintaining control of the process." The 9100 remains in command, says Travis. It makes the go/no-go evaluation of the test results, based on the data passed to it by the GPIB tester. Together, the two testers provide rapid, comprehensive testing of mixed analog and digital boards. Atlantic Design is a member of the Fluke Cooperative Services (FCS) team, a group of independent firms that help users develop and implement new applications of Fluke equipment. Its test services include development of in-circuit and functional tests for their customers' systems.

Fluke Cooperative Service Members

The Fluke Cooperative Services team is a network of independent specialists who know hardware, software, and your industry.

Atlantic Design
120 Plaza Drive
Binghamton, N.Y. 13903
607-729-9251
Contact: Rick Travis

Manufacturing, engineering, design, and drafting services for industrial and military accounts. Experienced in 9100A applications, including the design and implementation of analog and digital test systems.

Phoenix Test Engineering, Inc.
2501 W. Berhend Drive, Suite 77
Phoenix, AZ 85027
602-581-9100
Contact: Vince Richards

Design and fabrication of custom systems, application programming, training, and program development for the Fluke 9100A. Over fifty man-years experience in the ATE field.

Raytheon Service Corp.
22 Cotton Road
Nashua, NH 03063
603-889-1500
Contact: Ben Spencer

Test program software for 3050 and most testers. Experience in military documentation. Trained in 9100A programming. Contract testing.

Ro-mar, Ltd.
2268 Executive Circle
Colorado Springs, CO 80906-4136
719-540-0600
Contact: Dick Stein

Design and fabrication of custom rack & stack board test systems. Applications programming for 9000A, 9100A, and custom board test systems. PCB and module testing and troubleshooting. Has commercial and military experience.

SofTest Designs
4242 Medical Drive
San Antonio, TX 78229
512-558-7171
Contact: Paul Bernhard

Design and fabrication of custom systems, application programming and test fixtures for Fluke 9100A Series Test Systems. Application review and program design for PCB testing. Custom software packages for automatic probing and test of PCB's. Has commercial and military experience.

FLUKE®
TroubleShooter

John Fluke Mfg. Co., Inc.
Everett, WA 98206-9090
Call 1-800-44-FLUKE
or in Washington state
(206) 347-6100

Philips International B.V.
T&M Dept., Building TQIII-4
5600 MD Eindhoven,
The Netherlands
Telex: 35000 PHTC NL/
NLFEVSI

P0021A-02U9011/SE EN

Get a Free Fluke 87 Multimeter

The author of each story in this issue of the Troubleshooter receives a free Fluke multimeter for sharing a unique application of Fluke board testers. And we're making the same offer to you.

If you use Fluke board test instruments, and you've learned some new ways to get the most out of your system or have found a solution to a difficult problem, we'd like to hear from you.

Send your stories to:
Editor, Fluke Troubleshooter
M/S 246-F
John Fluke Mfg. Co., Inc.
P.O. Box 9090
Everett, WA 98206-9090

If we publish your story, we'll send you a Fluke 87 multimeter free. Published articles will carry your byline and your company's name.

Fluke reports the information provided by customers in the Troubleshooter without complete engineering testing or review. We provide the information "as is" and without express or implied warranties. The reader should be aware that modification of a Fluke instrument may void the Fluke warranty.

All prices are U.S. list. Some products may not be available in certain countries.