

6-30-2004
Bob Shepard

"Experiment -- A Logical Riddle" Last problem in the Digi-Comp 1 Instruction Manual.

When Tom, Dick, and Jim were asked who caught the fish, they replied...

1. If Tom didn't, then Dick did.
2. "Tom did" and "Jim didn't" are not both true statements.
3. If Dick did, then Tom did and Jim did.

Q: Is it true that Jim did?

```
***** DISCLAIMER *****  
*                               *  
* The following approach has not *  
* been tested, may not make sense, *  
* and probably is not interpreted *  
* per the original authors. I had *  
* surgery yesterday and my pain *  
* medication alters reality. *  
*                               *  
*****
```

It is a confusing problem. The second statement is worded oddly and may have multiple interpretations.

One way to solve the problem is to program the machine to test for logic conditions when clocked. Combinations of fishermen with fish (states) would be entered and then tested.

If there are violations of any of the three statements for all states where Jim might have fish, the answer is NO -- Jim didn't have fish. Otherwise, Jim did have fish and we know from the valid states what combination of friends can have fish. Only four states need to be tested because Jim is assumed to have fish.

Because the test will be for violations of the statements, the tests can be worded as follows:

1. Tom didn't and Dick didn't.
2. Tom did and Jim didn't.
3. Dick did and (Tom didn't or Jim didn't)

The third test can be broken down as follows:

- 3a. Dick did and Tom didn't.
- 3b. Dick did and Jim didn't.

Note that tests 2 and 3b include "Jim didn't." Because the states being tested all assume "Jim did," these two tests can be omitted.

Assign Digi-Comp's three flip-flops to represent the three fishermen. Let "1" indicate he did have fish and "0" indicate he didn't.

At this point you might notice that there are no flip-flops left to indicate whether the logic test passes or not. The trick is to set Jim's flip-flop for each test state and have your program reset it if the test indicates a logic violation. Before the clock it's a state input and after the clock it's a test result.

For a totally different approach, use the "if-then" statements to program "previous state - next state" in the machine. The challenge will be to create a set of "if-then" type rules for the second statement. It may be possible to start the machine from each possible state and run the program until a valid (stable) end state is achieved. It may be necessary to rerun the test with three separate programs to account for parts of the second statement.

-Bob S.