Output Logic Testing

by Chris Werstiuk
Valence Engineering, Inc.

Introduction

One of the most confusing aspects of modern relay testing is deciding what to test. Testing earlier relay generations was relatively straightforward, as these devices had limited scope and fixed configurations. Pickup and timing tests were performed to make sure the device was mechanically sound and all the parts operated correctly. Today's relays are infinitely more complex, and our testing philosophy should be modified accordingly. The complexity of these relays is due to the flexibility of their outputs. LEDs, output relays, virtual outputs to remote devices, and displays on the relay's front panel can customize to communicate nearly every aspect of the electrical system. These outputs are controlled by their internal logic which actually defines the relay's functionality.

Output logic errors are the most prevalent problems I discover during relay testing and occur in 80 percent of the relays. Because of this experience, I feel that output logic testing is, by far, the most important part of relay testing. This article discusses output logic testing philosophies to help discover these problems before it's too late.

Think of the relay's output logic as an electrical schematic that must be commissioned with the same attention to detail that one would apply to any other schematic. The first step to output testing is to translate the output logic into a checklist, electrical schematic, operating description, or any other representation that you understand. The logic should be broken down into its base components or combinations, and each combination should be checked off as it is proven. Following are some rules for output logic testing.

1. Always use the as-left relay settings when testing output logic. Why test something you are not going to use?
2. All setting changes must be completed before performing output logic tests. Why would you test something that you're going to change in the future? One-character mistakes in changed relay logic can be disastrous.
3. Use end devices (breaker/disconnect switch status) whenever possible. Your goal is to simulate real life conditions to ensure the logic operates correctly while in-service.
4. Question the design engineer if the logic doesn't make sense to you.
5. Use the operating description instead of interpreting the relay's settings when testing complex logic. The relay is going to do what it's programmed to do. You want to make sure that it's programmed correctly. If an operating description doesn't exist, ask for one.

Simple Output Logic Testing Example

The output logic in Figure 1 is from a typical feeder relay and is shown in the most common logic formats and translated into electrical schematics. This logic was compared to the dc schematic and performs the following functions.

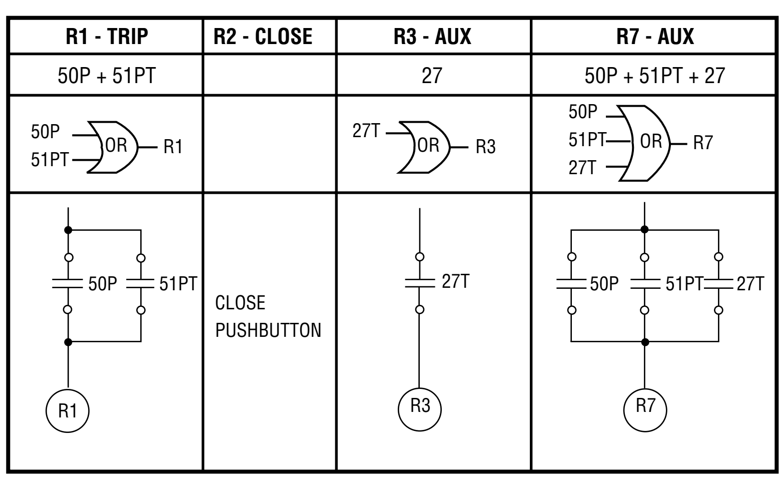

Figure 1 — Example Output Logic

- Output R1 trips a lockout relay that trips the circuit breaker.
- Output R2 closes the circuit breaker.
- Output R3 trips the circuit breaker directly.
- Output R7 provides trip annunciation to the SCADA/DCS/alarm panel.

Testing this logic is relatively straightforward. After all pickup and timing tests are completed, perform the following steps:

1. Verify that the rules discussed above are followed and energize as much of the circuit as possible. If the station is in operation and there are not sufficient test switches to isolate all the desired trips, record the reasons for untested logic. (See rule #3.)
2. Connect relay test set inputs to Output R1 and inject a 50P timing test. Verify:
   - Output R1 operates within the correct time.
   - The relay annunciates the correct trip and phase.
   - The lockout relay trips.
3. Reset the lockout and inject a 51 timing test and verify everything in step 2.

FINAL OUTPUT CHECKS						
SOLID STATE TRIP	R1	R2	R3	R5	R6	R7
NA	51PT	CLOSEPB	27T	NA	NA	27T
	50P					51PT
						50P
COMMENTS: ☑ YES ❑ NO ❑ SEE NOTES						

Figure 2 — Example Test Sheet for Output Logic

4. Connect the test set input across Output R2 and push the close button. Verify the contact operates and closes the circuit breaker..
5. Connect the relay test set input to Output R3 and inject a 27 timing test. Verify:
   - Output R3 operates within the correct time.
   - The relay annunciates the correct trip and phase.
   - The circuit breaker trips.
6. Repeat steps 2, 3, and 5 with the test set inputs connected across R7 and verify DCS/SCADA/Annunciator operation.

One way to document these tests is to create a spreadsheet with each output contact in a column as shown in Figure 2. Each row under the output designations represents an OR gate. Remove the highlight as each step is performed.

Complex Relay logic Example

Complex logic schemes require more thought. The breaker fail protection commonly installed on SEL relays is a great example of a slightly more complex logic scheme. The output logic for this example will be OUT101=SV1T. In order to test this output, we need to find out how SV1T and its related settings are defined. SV1T is derived from the settings in Figure 3.

Relay Settings	Application Review
OUT101 = SV1T SV1 = (SV1 + IN201) * (50P1 + 50G1) SV1PU = 10 SV1DO = 10 50P1P = 0.25 50G1P = 0.25	IN201 = TRIPS FROM OTHER RELAYS

Figure 3 — Example #2 Settings

Break down all of the possible combinations required for OUT101 to operate before testing. The first step is to expand the equation until all of the brackets have been removed and we are left with a series of OR functions. SV1 becomes SV1 * 50P1 + SV1 * 50G1 + IN201 * 50P1 + IN201 * 50G1. Figure 4 displays the different logic schemes and Figure 5 is an example test sheet.

MATH LOGIC

SV1 = (SV1 + IN201) * (50P1 + 50G1)
SV1 = (SV1 * 50P1) + (5SV1 * 50G1) + (IN201) * (50P1 + 50G1)
SV1 = (SV1 * 50P1) + (5SV1 * 50G1) + (IN201 * 50P1) + (IN201 * 50G1)

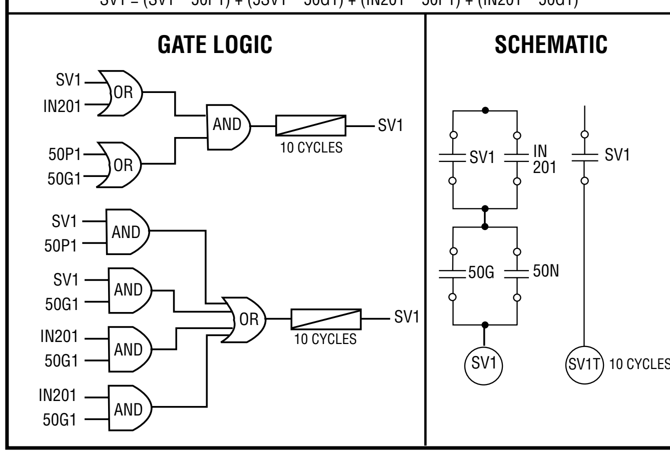

Figure 4 — Example Breaker Failure Logic

FINAL OUTPUT CHECKS						
OUT 101	OUT 102	OUT 103	OUT 104	OUT 105	OUT 106	OUT 107
SV1 * 50P1	NA	NA	NA	NA	NA	NA
SV1 * 50G1						
IN201 * 50P1						
IN201 * 50G1						
COMMENTS: ☑ YES ❑ NO ❑ SEE NOTES						

Figure 5 — Example Test Sheet for Breaker Failure Logic

After the logic has been reduced to its simplest components, output testing is a breeze using the following steps.

- Apply IN201 and nothing should happen.
- You could apply 0.3 ampere in any phase to operate the relay but this will operate both the 50P and 50G elements. First apply three-phase current higher than 0.25A with IN201 applied. OUT101 should close after 10 cycles has passed. You can now mark 50P1*IN201 as completed.
- Remove IN201 and if the output remains closed, mark 50P1*SV1 as completed because the SV1 seal-in function has operated correctly.
- Turn the current off and the output should open.
- Reapply current and nothing should happen.
- Apply 0.20 amperes at 0º in each phase (equals 0.6 amperes ground fault) and nothing should happen until you apply IN201. If the output operates 10 cycles after applying IN201, mark 50G1*IN201 as completed.
- Remove IN201 and you can mark 50P1*SV1 as completed if the output remains closed.

Output Logic Testing Methods

There are many ways to prove the output logic settings. The tester can perform all other tests and wait until the end before performing a timing test for every element on every output contact as shown in the example. To reduce test set changes, connect the test set contact sensing to output R1 and perform a 51PT timing test, then move the contact sensing to R7 and repeat the test. Then perform a 50P test, move to R1 and perform the test again. If the test set has multiple inputs and/or event reporting, connect all relay outputs to the test set inputs and watch to make sure all the correct relay outputs operate. After proving that the contacts actually operate at least once, monitor their status via the software or open the event records to make sure the correct outputs operate.

Modern test equipment allows one to simulate almost any real-life fault and be able to perform a timing test on nearly every element without interference from other elements. If one is unable to obtain a timing test for an element due to interference from another element, the settings are redundant (except 87-differential elements, of course) and should be noted on the test sheets and brought to the design engineer's attention. Waiting until the end and proving all of the output logic at once is the least confusing method, but it can be inefficient due to the need to reconfigure the test set between each test, a relatively minor inconvenience for computer operated testers. This method should always be used, if setting changes are performed during testing, to ensure the relay will operate when required.

Another method incorporates the element testing into the output testing. This method is more efficient and should only be used if setting changes during testing are nonexistent or carefully planned. With this method, perform each timing test on another output until all of the assigned outputs are used. In the example, the first 51PT timing test (A-phase, for example) can be performed using R1 and the next timing test (B-phase, for example) is performed on R7. After all of the 51PT timing tests are completed, the first 50-element timing test can be performed on R7 and the next test will be performed on R1. This method is more efficient but requires careful attention to make sure that no element/output combinations are missed.

Conclusion

A common mistake for relay testers is to use spare outputs, displays, and/or LEDs for their pickup and timing tests and ignore the in-service output logic, believing that they are using the same elements in their test equations as the final logic. Following are two real life examples of mistakes made when testing the output logic last.

Example 1

After testing the differential element using a spare contact on a GE T60 relay using "XFMR PCNT DIFF OP" as the test logic, I went back to test each of the outputs. Everything went fine until I came to the third output and decided to apply a B-phase differential fault and nothing happened! I reviewed the setting and it was set to operate when "XFMR PCNT DIFF OP A" operated. One character made the difference between an A-phase differential trip and a differential trip on all phases.

Lesson learned: One character mistakes are easy to miss and potentially disastrous!!

Example 2

I finished testing all of the elements on an SEL-300G relay and everything was fine. I started testing the output logic and came across the highlighted bit of logic of this equation: 64G1T + (64G2T * !27PP1 * 60LOP) + 87R + 87U + (81D2T+ 81D3T) * 52A + 81D5T + 81D6T. 64G2T is 3rd harmonic undervoltage protection, 27PP1 is undervoltage protection, and 60LOP is fuse failure protection. 64G2T was disabled by this equation because "!27PP1 * 60LOP" is an impossible condition and can never happen. The setting was changed to "!27PP1 * !60LOP" and operates correctly. These settings were supplied by a major generator manufacturer from a boilerplate template. How many generators out there have 64G2T protection disabled and no one knows it?

Lesson learned: Always perform output logic tests and question logic that doesn't make sense!

Chris Werstiuk graduated from the Electrical Engineering Technology program at the Northern Alberta Institute of Technology and is a Registered Engineering Technologist and Journeyman Power System Electrician. He has performed low-, medium-, and high-voltage acceptance testing, commissioning, and maintenance testing at various locations throughout the Americas. His experience is varied, and he is comfortable in mining, utility, generating, distribution, production, institutional, plant, pipeline, and commercial environments. Chris is currently a Project Manager for Valence USA, a full service field service and electrical training organization.