Development of a Built-in Automatic Testing Method for Digital Protection Logic

Tae Young Jhee [a], Tae Ryoun Kim [a], Jonghyun Kim [a*]
[a] *Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*
[*] *Corresponding author: jonghyun.kim@kaist.ac.kr*

***Keywords :** Built-in Automatic Testing, State transition diagram, Plant Protection System (PPS)*

1. Introduction

The Plant Protection System (PPS) is a safety-related system in nuclear power plants (NPPs) that initiates a reactor trip when monitored process variables exceed or fall below predefined trip setpoints under abnormal or accident conditions. To ensure the continued capability of this safety function, regulatory requirements mandate periodic surveillance testing (PST) to verify that the PPS and its subsystems can perform their design-basis functions upon demand [1, 2]. In conventional NPPs, PST has largely been conducted as a procedure-driven activity by operators or maintenance personnel.

Digital safety I&C (Instrumentation and Control) systems in NPPs have been transitioning from software-based Programmable Logic Controller **(**PLC) platforms to deterministic Programmable Logic Device (PLD)-based architectures. For safety-critical functions such as the Reactor Protection System (RPS), PLC-based designs rely on high-integrity software [3]. In contrast, PLD-based designs implement protection functions as hardware-configured logic, enabling deterministic execution with reduced dependence on operating systems and complex runtime software [4]. This transition is motivated by the reduced software qualification burden as well as the deterministic and parallel processing characteristics of Field-Programmable Gate Array **(**FPGA) logic [4]. The practical feasibility of FPGA-based safety I&C has also been demonstrated in safety-related applications, including isolation functions and shutdown system implementations [5, 6].

Despite these advances in digital safety I&C, PST in many NPPs remains largely procedure-driven and manually executed. In conventional large NPPs, the PPS typically employs a multi-channel architecture, and surveillance testing therefore requires repeated channel-level verification of protection logic functions. Because these test activities must be performed repeatedly for individual channels while maintaining the required protection function, manual PST imposes a substantial procedural and operational burden [7]. Moreover, channel-by-channel manual testing can increase the duration for which individual protection channels are placed in a test or bypass condition, potentially reducing overall protection system availability [8]. Therefore, an automated testing framework for PST is needed.

To address these challenges, built-in test capabilities have been incorporated within the PLC-based protection framework in commercial NPPs, where periodic testing is performed through coordinated use of the ATIP (Automatic Test and Interface Processor) and the associated bypass function. For example, the IDiPS-RPS (Integrated Digital Protection System–Reactor Protection System), developed under the KNICS (Korea Nuclear Instrumentation and Control System) R&D program, incorporated automated test signal generation via the ATIP to support periodic verification of protection logic during normal operation [9]. Jeon et al. proposed an automated testing approach for PLC-based protection systems using a dedicated signal processing unit capable of injecting test stimuli and acquiring output responses [10].

However, existing automated testing approaches remain functionally limited. Test initiation is often operator-dependent, and key actions—such as protection channel bypassing and test stimulus injection—are not performed autonomously. These limitations stem in part from the characteristics of PLC-based platforms, in which test execution is typically implemented as an external, procedure-driven sequence rather than as a controller-integrated function. Consequently, the expected benefits of automation—including shorter test duration, reduced operator involvement, and improved protection system availability—have not been fully achieved. These limitations motivate further automation of PST procedures, which can be achieved by integrating the test sequence directly within the FPGA-based logic processor.

This study presents a built-in automatic testing method for digital PPS at the logic-model level. In the proposed method, the test output is confined within the processor test logic and is not propagated to the downstream safety logic, thereby ensuring that the test function does not interfere with the safety function. As a representative case study, the fixed-setpoint bistable logic test is modeled as a state transition diagram and implemented in MATLAB/Simulink using Stateflow. State and transition coverage analyses confirmed that all defined states and transitions were exercised during simulation. The proposed approach can be applied to the development of controller-integrated automated PST for FPGA-based digital protection systems.

2. Methods

2.1 State transition diagram

A state transition diagram provides an abstract, graphical representation of sequential logic that can be

synthesized and implemented in programmable logic devices (PLDs), such as FPGAs. In its conventional form, it is expressed as a directed graph composed of states, input conditions, and output responses. Each state is depicted as a circle, while directed arcs indicate transitions between states.

Each transition arc is annotated with the input conditions that enable the state change. The associated outputs are indicated within the state symbol. If a transition arc is shown without an explicit condition, it denotes an unconditional transition, meaning that the system advances to the next state regardless of the input values.

For correctness and deterministic behavior, all possible combinations of input conditions must be defined for each state. Moreover, no identical input combination should be assigned to more than one outgoing transition from the same state.

2.2 Built-in Automatic Testing Method

The Built-in Automatic Testing Method is an automated periodic surveillance testing approach in which, at predefined test intervals, the test logic generates either (i) a test stimulus corresponding to an abnormal trip condition or (ii) a test stimulus representing a setpoint exceedance based on the initial setpoint and the current process variable value. The generated test stimuli are injected into FPGA-based processor logic to periodically verify the functional performance of the protection logic under the processor test logic.

In this architecture, as shown in Figure 1, the test logic implemented within the FPGA-based processor applies the generated test stimuli to the protection logic and acquires the corresponding outputs for assessment against the expected results. To ensure independence from the safety function, the test stimuli and the resulting test outputs are tagged with a test identifier. The tagged test output is captured exclusively by the processor test logic for pass/fail evaluation and is not forwarded to the downstream safety logic that performs the subsequent safety function. This design ensures that the built-in test function remains functionally isolated from the trip execution path, thereby preventing any unintended actuation of safety functions during testing.

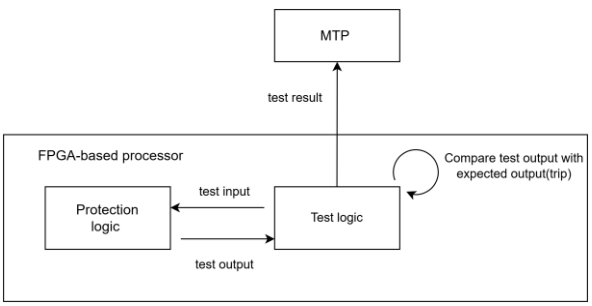

Fig. 1 Concept of Built-in Automatic Testing

3. Results

3.1 Case Study: Built-in Automatic Testing Method for Fixed Setpoint Bistable logic

The fixed-setpoint bistable logic test is intended to verify the functional integrity of bistable logic applied to process variables with fixed setpoints, such as high pressurizer pressure and low steam generator pressure in the APR1400 PPS [11]. The test stimulus represents an abnormal trip condition, corresponding to a high trip or low trip of the setpoint. During the test, the response is evaluated by confirming (i) generation of the expected bistable output and (ii) consistency between the observed output and the expected output (trip) for the applied stimulus. The proposed built-in automatic periodic testing method is executed according to the following procedure:

(1) Generate a test stimulus corresponding to either the high condition or the low condition within the built-in test logic.
(2) Apply the generated stimulus to the bistable logic in the processor in accordance with the predefined test period.
(3) Produce the corresponding bistable output (trip/non-trip) in response to the applied stimulus.
(4) The processor test logic acquires the bistable output from the bistable logic.
(5) Determine the test result (pass/fail) by comparing the acquired output with the expected trip decision for the applied stimulus.
(6) Transfer the resulting test status to the MTP (Maintenance and Test Panel). The tagged test output is confined to the test logic and is not propagated to the coincidence logic processor.

3.2 State Transition diagram of the Built-in Automatic Testing Method for Fixed Setpoint Bistable Logic

The state transition diagram of the built-in automatic testing method for the fixed-setpoint bistable logic is shown in Figure 2. The logic consists of ten states: WAIT, READ_PV_ALL, SELECT_PV, HIGH_TRIP, LOW_TRIP, APPLY_TEST, CAPTURE_OUT, PASS, FAIL, and SEND. These states can be divided into two principal functional groups: (1) test stimulus generation (READ_PV_ALL, SELECT_PV, HIGH_TRIP, LOW_TRIP, and APPLY_TEST) and (2) output verification and result transmission (CAPTURE_OUT, PASS, FAIL, and SEND).

In the WAIT state, the logic remains in a standby condition until the predefined periodic test interval is satisfied (Time $\geq$ Test Period), at which point a new automatic test sequence is initiated.

In READ_PV_ALL, all fixed trip setpoints required for the current test cycle are retrieved. Upon completion (All setpoints loaded), the logic proceeds to SELECT_PV.

In SELECT_PV, the process variable (PV) to be tested is selected and the corresponding trip setpoint

(SP) is latched. The trip polarity of the selected channel is then determined: Trip Polarity = HIGH denotes a rising-trip configuration in which a trip is initiated when the PV exceeds SP, whereas Trip Polarity = LOW denotes a falling-trip configuration in which a trip is initiated when the PV falls below SP. Based on this determination, the logic transitions to either HIGH_TRIP or LOW_TRIP.

In the HIGH_TRIP state, a high-side test stimulus is generated by assigning a test input corresponding to a high-trip condition. Similarly, in the LOW_TRIP state, a low-side test stimulus is generated for a falling-trip condition. Once the stimulus is prepared (HIGH/LOW test input ready), the logic transitions to APPLY_TEST.

In APPLY_TEST, the prepared stimulus is applied to the bistable logic (Test input valid = 1).

In CAPTURE_OUT, the bistable output corresponding to the applied test stimulus is captured and qualified. The transition condition (Test_Output_ID = Selected_PV & Test_Output_ID ≠ Previous_Test_Output_ID) ensures that the received output corresponds to the currently selected PV and represents a newly generated response rather than a previously latched output.

The logic then transitions to PASS if the captured output matches the expected trip decision (Test_Output = Expected_Output), or to FAIL if a discrepancy is detected (Test_Output ≠ Expected_Output).

In the SEND state, the test result (PASS or FAIL) is transmitted to the external monitoring interface after being maintained for a required hold duration (Hold_Time ≥ Required_Hold). The PV index comparison (Current_PV_Index < Last_PV_Index or Current_PV_Index = Last_PV_Index) determines whether the logic proceeds to the next PV for continued testing or returns to the WAIT state for the subsequent test cycle.

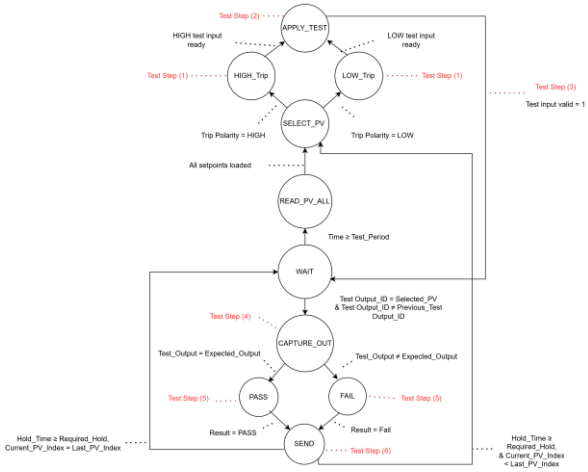

Fig. 2 State Transition Diagram of the Built-in Automatic Testing Method for Fixed Setpoint Bistable Logic

3.3 Implementation and Coverage Analysis of the Proposed State Transition Model

To confirm the operability of the proposed state transition model, the model was implemented in MATLAB/Simulink using Stateflow based on the state transition diagram defined in Figure 2. For simulation, an initial process variable (PV) value and its corresponding trip setpoint were specified, together with the periodic test initiation condition. Based on these specified values, the rising-trip and falling-trip test inputs were calculated and applied during simulation. Coverage analysis using the MATLAB Simulink Coverage tool was performed to confirm that the implemented state transition model executes without errors and follows the intended operational sequence, including state transitions, test stimulus generation, response capture, and PASS/FAIL decision logic. The analysis confirmed that all defined states achieved 100% block execution coverage. The results indicate that the implemented model operates consistently with the designed state transition model.

4. Conclusions

This study developed and modeled a Built-in Automatic Testing Method as an automated periodic surveillance testing (PST) scheme for FPGA-based reactor protection logic. As a representative case study, the fixed setpoint bistable logic test—conventionally performed as a manual PST procedure—was reformulated as an on-line built-in test sequence. The proposed method was modeled using a state transition diagram and implemented in MATLAB/Simulink with Stateflow to demonstrate its functional feasibility at the logic-model level. HDL synthesis, FPGA-board implementation, and hardware interface verification with external test equipment were not addressed in this study. Further research is required to verify the functional isolation of the test function from the protection function and to evaluate applicability to commercial safety-critical protection logic.

Future work will extend the proposed methodology to additional PST test cases, including (i) rate-limited variable-setpoint bistable logic, (ii) manual-reset variable-setpoint bistable logic, and (iii) coincidence logic. The proposed state transition model will be described in a hardware description language (HDL), synthesized, and implemented on an FPGA development board to verify that the test logic executes correctly and maintains functional isolation from the protection logic. Quantitative performance criteria will be established, and V&V activities will be conducted to evaluate fault detection coverage, timing characteristics (e.g., response time), and applicability to safety-related protection systems.

ACKNOWLEDGEMENT

This work was supported by the Innovative Small Modular Reactor Development Agency grant funded by

the Korea Government (MCEE) (No. RS-2024-00408005).

REFERENCES

[1] Institute of Electrical and Electronics Engineers (IEEE), IEEE Standard Criteria for the Periodic Surveillance Testing of Nuclear Power Generating Station Safety Systems, IEEE Std 338-2012, 2012.
[2] U.S. Nuclear Regulatory Commission (U.S. NRC), Periodic Testing of Electric Power and Protection Systems, Regulatory Guide 1.118, Rev. 3, 2007.
[3] IEEE, IEEE Standard for System, Software, and Hardware Verification and Validation, IEEE Std 1012-2024, 2024.
[4] J.-J. Lu, T.-C. Hsu, and H.-P. Chou, System assessment of an FPGA-based RPS for ABWR nuclear power plant, Progress in Nuclear Energy, Vol. 85, pp. 44–55, 2015.
[5] U.S. Nuclear Regulatory Commission, Wolf Creek Generating Station, Issuance of Amendment No. 181: Modification of the Main Steam and Feedwater Isolation System Controls, Mar. 31, 2009, ADAMS Accession No. ML090610317.
[6] J. She and J. Jiang, On the speed of response of an FPGA-based shutdown system in CANDU nuclear power plants, Nuclear Engineering and Design, Vol. 241, No. 6, pp. 2280–2287, 2011.
[7] V. Agarwal et al., Technical Specification Surveillance Interval Extension of Digital Equipment in Nuclear Power Plants: Review and Research, INL/EXT-19-54251, Idaho National Laboratory, 2019.
[8] Y. Lee et al., The Fault-Tolerant Evaluation Model Due to the Periodic Automatic Fault Detection Function of the Safety-Critical I&C Systems in Nuclear Power Plants, Nuclear Engineering and Technology, Vol. 53, No. 1, pp. 229–239, 2021.
[9] J. Lee et al., The Automatic Test Features of the IDiPS Reactor Protection System, Nuclear Engineering and Technology, Vol. 39, No. 3, pp. 299–308, 2007.
[10] H. Jeon et al., Development of Automatic Testing System for PLC-based Reactor Protection Systems, Nuclear Engineering and Technology, Vol. 43, No. 2, pp. 113–122, 2011.
[11] Korea Hydro & Nuclear Power Co., Ltd. (KHNP), APR1400 Design Control Document Tier 2, Chapter 7: Instrumentation and Controls, Rev. 3, 2018.