Methodology for 100% Combinatorial Testing of the U.S. EPR™ Priority Module

ANP-10310NP
Revision 1

Technical Report

March 2011

AREVA NP Inc.

(c) 2011 AREVA NP Inc.

Copyright © 2011

AREVA NP Inc.
All Rights Reserved

Nature of Changes

Item	Section(s) or Page(s)	Description and Justification
1	Section 2	Updated Section 2 to reflect Figure 2-1
2	Figure 2-1	Updated Figure 2-1 to incorporate changes based on new I&C architectural design
3	Section 6	Updated Section 6 to conform to changes as described in the Response to RAI 373, Questions 7.1-23 and 7.1-24

Contents

Page

1.0 INTRODUCTION 1-1
1.1 Purpose and Scope 1-1
1.2 Overview 1-1
2.0 PRIORITY MODULE USED IN PACS 2-1
3.0 D&IC-ISG-04 GUIDANCE FOR PRIORITY MODULE TESTING 3-1
4.0 DESCRIPTION OF 100% COMBINATORIAL TESTING 4-1
4.1 Testing of Actuation Signals 4-1
4.1.1 Testing of Nonlatched Actuation Signals 4-1
4.1.2 Testing of Latched Actuation Signals 4-1
4.1.3 Testing of Delayed Actuation Signals 4-2
4.2 Testing of Infrastructure Signals 4-2
4.3 Testing for Invalid Signals 4-3
4.4 Test Apparatus 4-3
5.0 DESCRIPTION OF THE SPLM1 AND SPLM1-PC10 5-1
5.1 The SPLM1 module 5-1
5.2 The SPLM1-PC10 module 5-1
5.3 Distribution of Functions Between PLD A and PLD B 5-2
5.4 Details of Subsystem A 5-2
6.0 TEST METHODOLOGY EXAMPLE USING THE SPLM1-PC10 6-1
6.1 Test Environment 6-1
6.2 Test Case Implementation for SPLM1-PC10 6-3
6.3 Expanding the Existing Test Environment 6-4
7.0 MANUAL VERIFICATION 7-1
7.1 Approach 7-1
7.2 Rule-Based Sorting of the Test Cases 7-1
7.3 Tool Support and Plausibility Checks 7-3
8.0 EXCLUSION OF INFRASTRUCTURE SIGNALS 8-1
9.0 REFERENCES 9-1
APPENDIX A RULE-BASED SORTING OF TEST CASES A-1

APPENDIX B EXAMPLE DEBOUNCING FUNCTION B-1

List of Figures

Figure 2-1—PACS Communication-Priority Pair (CoPP) 2-2
Figure 4-1—Overall Test Concept 4-5
Figure 5-1—Hardware Structure of the SPLM1 Module 5-5
Figure 5-2—Overview of SPLM1-PC10 Subsystem A 5-6
Figure 5-3—Overview of SPLM1-PC10 Subsystem B 5-7
Figure 5-4—Subfunction 1: Acquisition and Prioritization of Safety I&C Commands 5-8
Figure 5-5—Subfunction 2: Acquisition and Prioritization of Operational I&C and Control Room Commands 5-8
Figure 5-6—Subfunction 3: Travel Limit Switch Responded, and Control Room Test 5-9
Figure 6-1—Test Environment for the SPLM1 Programmed Variants 6-5
Figure 6-2—Time Curve with Inputs and Expected Outputs 6-5
Figure 6-3—Time Curve with Expected and Observed Outputs 6-6
Figure 6-4—Time Curves, Indicating Differences Between Expected and Observed Outputs 6-6
Figure 6-5—Composing Test Cases: Each Case Consists of Four Steps 6-7
Figure 6-6—Composition of the Three Basic Types of Test Cases into Elementary Steps. 6-7
Figure 6-7—Specification of Test Cases Using a Spreadsheet 6-8
Figure 6-8—Logic Involving a Timer 6-8
Figure 6-9—Specifying Test Cases Involving the Timer 6-9
Figure 6-10—Logic Involving a Latched Actuation Signal 6-9
Figure 6-11—Specifying a Test Case Involving a Latched Actuation Signal 6-10
Figure 8-1—Example of Processing of Infrastructure Signals 8-2
Figure A-1—Result After Application of Rule 0 ..... A-1
Figure A-2—Result After Application of Rule 1 ..... A-2
Figure A-3—Result After Application of Rule 2 ..... A-2
Figure A-4—Result After Application of Rule 3 ..... A-2

Figure A-5—Result After Application of Rule 4 ..... A-3

Figure A-6—Result After Application of Rule 5 ..... A-3

Figure A-7—Result After Application of Rule 6 ..... A-3

Figure B-1—Example of Debouncing Logic in the PLD B-2

Nomenclature

Acronym	Definition
CCF	Common-Cause Failure
CoPP	Communication - Priority Pair
DAS	Diverse Actuation System
ISG	Interim Staff Guidance
PACS	Priority Actuation and Control System
PAS	Process Automation System
PLD	Programmable Logic Device
PS	Protection System
SAI&C	Severe Accident Instrumentation and Control
SAS	Safety Automation System
SPLM	TELEPERM XS Programmable Logic Module
SPLM1-PC	(SPLM1 module used for) Priority Control
TÜV	Technischer Überwachsungsverein (*Technical Inspection Association*)

Glossary

Term	Definition
actuation signal	A signal received by the priority module device that requests initiation or termination of action of the final actuated device. There are three types of actuation input signal: latched, nonlatched, and delayed.
communication-priority pair	The term used to describe a PACS communication module and priority module pair. These modules are separate devices but are always paired to carry out the non-safety and safety functions, in support of the single actuated device they support.
delayed actuation signal	An actuation signal that must remain at a new valid logic value for a pre-defined period of time, before the new value is used in processing. For the special case of time-limited delay, the value used in processing can be different from the value of the input, following a transition from valid logic 0 to valid logic 1 and expiration of the time-limited delay. In this case, the value used in processing would be 0 and the input value could remain 1. *Note: This special case is anticipated to be used to control the overlapping test between protection system and priority module, ensuring that the test mode would be cancelled even in case of a permanently erroneously frozen input.*
infrastructure signal	A signal received by the priority module that indicates the status of elements that support the priority module (e.g., power supply status, output driver status, specific test modes). An infrastructure signal does not request an action of the final actuated device. It is used to set the output of the priority module to a predefined value, in case of a fault in an element supporting the priority module. Infrastructure signals are generated based only on signals originating in the module or the module's division.
latched actuation signal	A priority module input that functions as follows. Following an actuation input signal transition from a valid logic "1" to a valid logic "0", the logic "1" continues to be used in processing (i.e., it is latched). When a different (pre-designated) actuation input signal (e.g., an actuation signal in counter-direction) transitions from a valid logic "0" value to a valid logic "1", the latched value returns to a logic "0" for use in processing.
minimum stability time	The amount of time an input signal must remain stable at a priority module input terminal before it may be used in processing. The minimum stability time may be specific to individual signals, in accordance with the characteristics of the signal sources. The minimum stability time considers effects from synchronized sampling of inputs (all inputs from I&C and field signals pass through a D-flip-flop, to ensure stable inputs; this implies a delay between 0 and 1 clock cycle) and/or due to input debouncing which excludes short input spikes from processing. This implies a time delay, to manage the bouncing of limit switch contacts.
nonlatched actuation signal	A priority module input whose logic value present as a valid input is the value used in processing.

priority module	The portion of the PACS that receives actuation signals from multiple sources and provides commands to an actuated device.
valid signal	An input signal that has remained stable at a priority module device input terminal for at least the minimum stability time.

1.0 INTRODUCTION

1.1 *Purpose and Scope*

This report describes the AREVA NP Inc. (AREVA NP) methodology for 100% combinatorial testing that will be applied to priority modules in the U.S. EPR™ I&C priority actuation and control system (PACS). As described in NRC Interim Staff Guidance D&IC-ISG-04, 100% combinatorial testing of a single, archetypical PACS priority module is a method acceptable to NRC staff as a means for proof-of-design testing.

This report presents an example of 100% combinatorial testing of an SPLM1-PC10 priority module. The example is illustrative of the test methodology. However, the SPLM1-PC10 is not planned for use as the U.S. EPR PACS priority module, and the example does not represent a formal test of the SPLM1-PC10.

The report demonstrates that the methodology for 100% combinatorial testign conforms with the applicable guidance in D&IC-ISG-04 for proof-of-concept testing.

1.2 *Overview*

This report describes:

1. The role of the electronic priority module in PACS.
2. The priority module's basic functions.
3. The guidance of D&IC-ISG-04 for proof-of-design testing.
4. The 100% combinatorial test methodology, including manual verification of automatically generated test cases.

The test methodology explanation includes an illustrative example involving a priority module, SPLM1-PC10, similar to the priority module to be used in the U.S. EPR PACS. The priority module for U.S. EPR will have a level of complexity no greater than the PC10 module presented here.

The example: describes the required functions of the test machine employed for 100% combinatorial testing and the overall testing process; illustrates the creation of the test vectors based on a spread-sheet implementation of the PC10's logic; and, provides examples of outputs from the existing test machine. The existing test machine is expected to be extended to implement the 100% combinatorial testing.

2.0 PRIORITY MODULE USED IN PACS

The U.S. EPR PACS prioritizes actuation signals from safety I&C systems and non-safety I&C systems. It performs actuator control and actuator monitoring by using electronic equipment.

To achieve independence between PACS safety functions and non-safety functions, these functions are allocated to two separate modules: a safety-related priority module, and a non-safety-related communication module. Together, these modules are referred to as a PACS communication-priority pair (CoPP).

PACS comprises numerous CoPPs that perform the following functions:

- Priority management of commands from I&C systems of different safety classes.
- Acquisition and distribution of check-back signals (e.g., from limit switches, torque switches, fault signals from switchgear,).
- Command termination, based on check-back signals.
- Actuator monitoring (e.g., run-time monitoring, discrepancy monitoring).

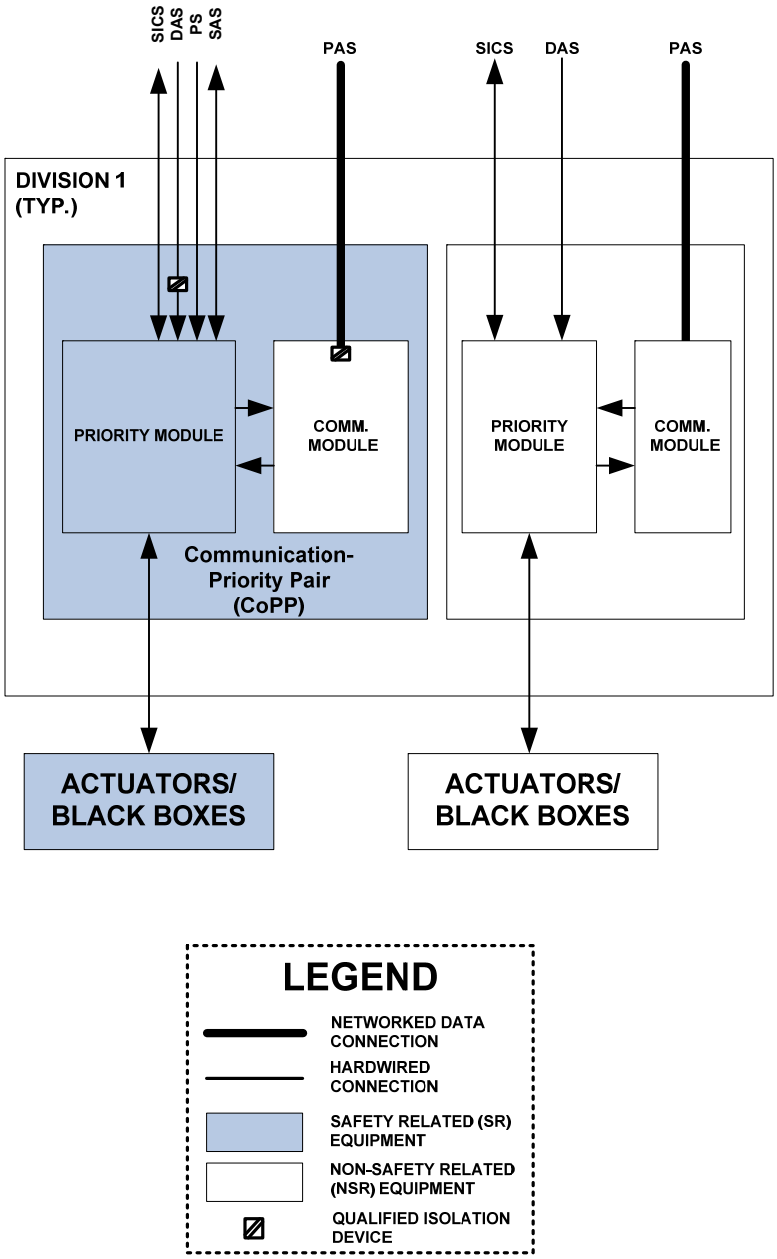
Every safety-related actuator (and small number of non-safety related actuators) has a dedicated CoPP.

Figure 2-1 provides a diagram of a communication-priority pair, showing the communication module, AV42(E), and the priority module, SPLM1-PC1x. (Note that the “x” here denotes various programmed instances of the SPLM1 as a priority control module, e.g., SPLM1-PC10, SPLM1-PC11, SPLM1-PC12)

Although the communication module performs non-safety functions, its hardware is qualified to the same requirements as safety-related modules.

The safety-related portion of PACS utilizes an electronic module where the logic functions performing priority control and command termination are implemented by using programmable logic devices (PLD). Adaptation between the module-internal signal levels (e.g., 3.3V) to the external signal levels (e.g., 24V) and module internal functionality (e.g., power supply to the PLD) are implemented using discrete electronics. Some self-monitoring features (e.g., switching outputs off in case of overload during start-up of the module or in case of loss of power supply) may also be implemented in the PLD. Their processing in the PLD is referred to as “processing of infrastructure signals.”

Figure 2-1—PACS Communication-Priority Pair (CoPP)



3.0 D&IC-ISG-04 GUIDANCE FOR PRIORITY MODULE TESTING

D&IC-ISG-04 establishes guidance for command prioritization. Two key subsections discuss the use of a PLD for the priority function.

Subsection 2.6 discusses software tools used to design and program a PLD (emphasis added):

> 2.6. Software used in the design, testing, maintenance, etc. of a priority module is subject to all of the applicable guidance in Regulatory Guide 1.152, which endorses IEEE Standard 7-4.3.2-2003 (with comments). This includes software applicable to any programmable device used in support of the safety function of a prioritization module, such as programmable logic devices (PLDs), programmable gate arrays, or other such devices. Section 5.3.2 of IEEE 7-4.3.2-2003 is particularly applicable to this subject. **Validation of design tools used for programming a priority module or a component of a priority module is not necessary if the device directly affected by those tools is 100 percent tested before being released for service. 100 percent testing means that every possible combination of inputs and every possible sequence of device states is tested, and all outputs are verified for every case.** The testing should not involve the use of the design tool itself. Software-based prioritization must meet all requirements (quality requirements, V&V, documentation, etc.) applicable to safety-related software.

Subsection 2.8 discusses proof-of-design testing (emphases added):

> 2.8. To minimize the probability of failures due to common software, **the priority module design should be fully tested (This refers to proof-of-design testing, not to individual testing of each module and not to surveillance testing.).** If the tests are generated by any automatic test generation program then all the test sequences and test results should be manually verified. **Testing should include the application of every possible combination of inputs and the evaluation of all of the outputs that result from each combination of inputs. If a module includes state-based logic (that is, if the response to a particular set of inputs depends upon past conditions), then all possible sequences of input sets should also be tested.** If testing of all possible sequences of input sets is not considered practical by an applicant, then the applicant should identify the testing that is excluded and justify that exclusion. The applicant should show that the testing planned or performed provides adequate

assurance of proper operation under all conditions and sequences of conditions. Note that it is possible that logic devices within the priority module include unused inputs: assuming those inputs are forced by the module circuitry to a particular known state, those inputs can be excluded from the “all possible combinations” criterion. For example, a priority module may include logic executed in a gate array that has more inputs than are necessary. The unused inputs should be forced to either “TRUE” or “FALSE” and then can be ignored in the “all possible combinations” testing.

Programming a PLD involves the use of design tools, typically provided by the PLD manufacturer. It is not practical to submit such tools to a verification and validation program, as required by IEEE 7-4.3.2 (cited in D&IC-ISG-04 section 2.6).

D&IC-ISG-04 section 2.8 recommends testing all possible combinations of inputs and sequences of input sets. Therefore, the AREVA NP approach to validating the PLD design tool is through 100% combinatorial testing of a single archetypical programmed PLD, along with full manual verification of the test results.

The combinatorial testing of the PLDs in the PACS priority module is a significant part of the design and testing of the programming of that module. Because the priority module for the U.S. EPR is currently in the design phase, the testing methodology is presented here using the existing PC10 module as an example. The priority module for U.S. EPR will have a level of complexity no greater than the PC10 module presented here.

4.0 DESCRIPTION OF 100% COMBINATORIAL TESTING

4.1 *Testing of Actuation Signals*

4.1.1 *Testing of Nonlatched Actuation Signals*

4.1.2 *Testing of Latched Actuation Signals*

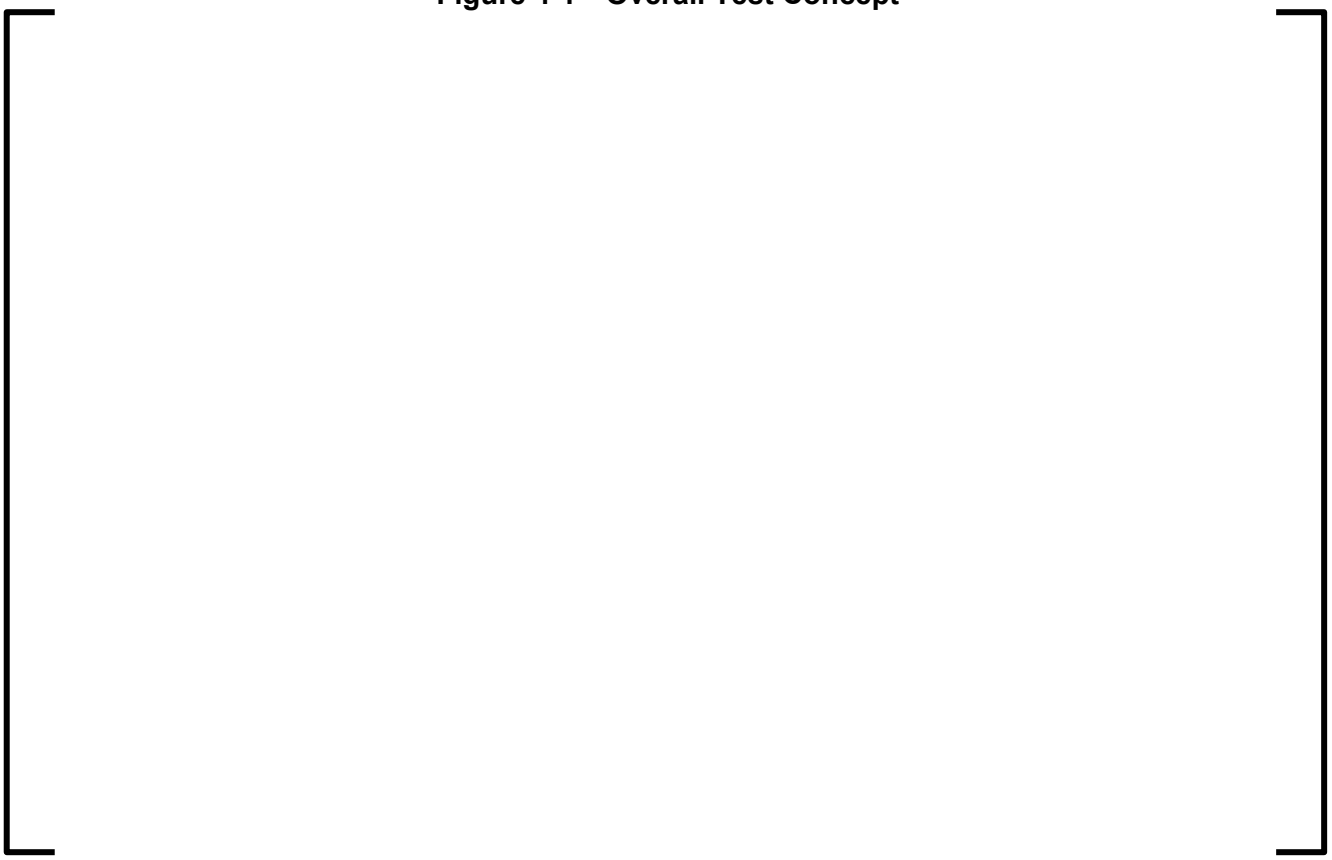
4.1.3 *Testing of Delayed Actuation Signals*

4.2 *Testing of Infrastructure Signals*

4.3 *Testing for Invalid Signals*

4.4 *Test Apparatus*

Figure 4-1—Overall Test Concept



5.0 DESCRIPTION OF THE SPLM1 AND SPLM1-PC10

5.1 *The SPLM1 module*

5.2 *The SPLM1-PC10 module*

5.3 *Distribution of Functions Between PLD A and PLD B*

5.4 *Details of Subsystem A*

Figure 5-1—Hardware Structure of the SPLM1 Module

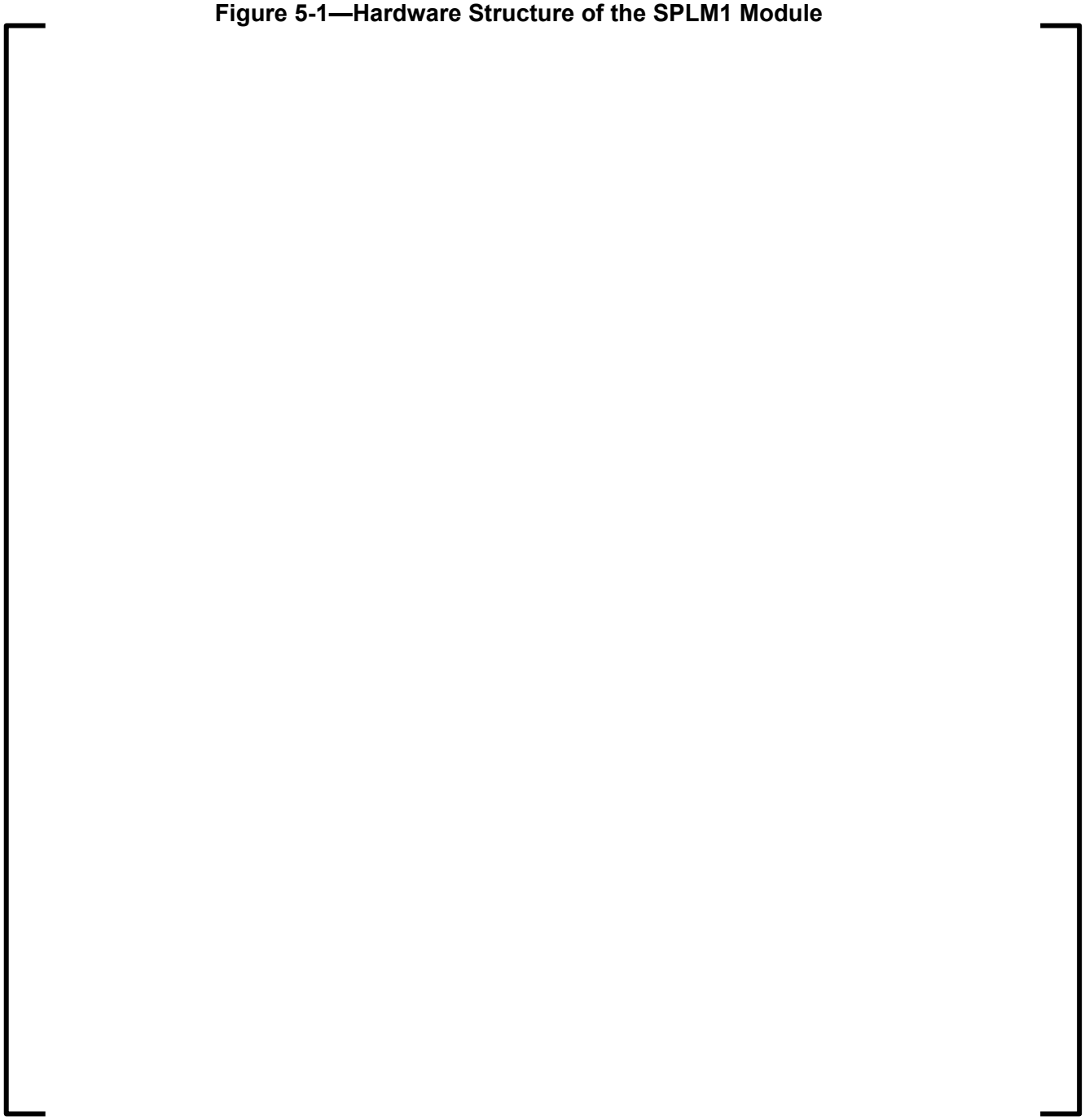


Figure 5-2—Overview of SPLM1-PC10 Subsystem A

Figure 5-3—Overview of SPLM1-PC10 Subsystem B

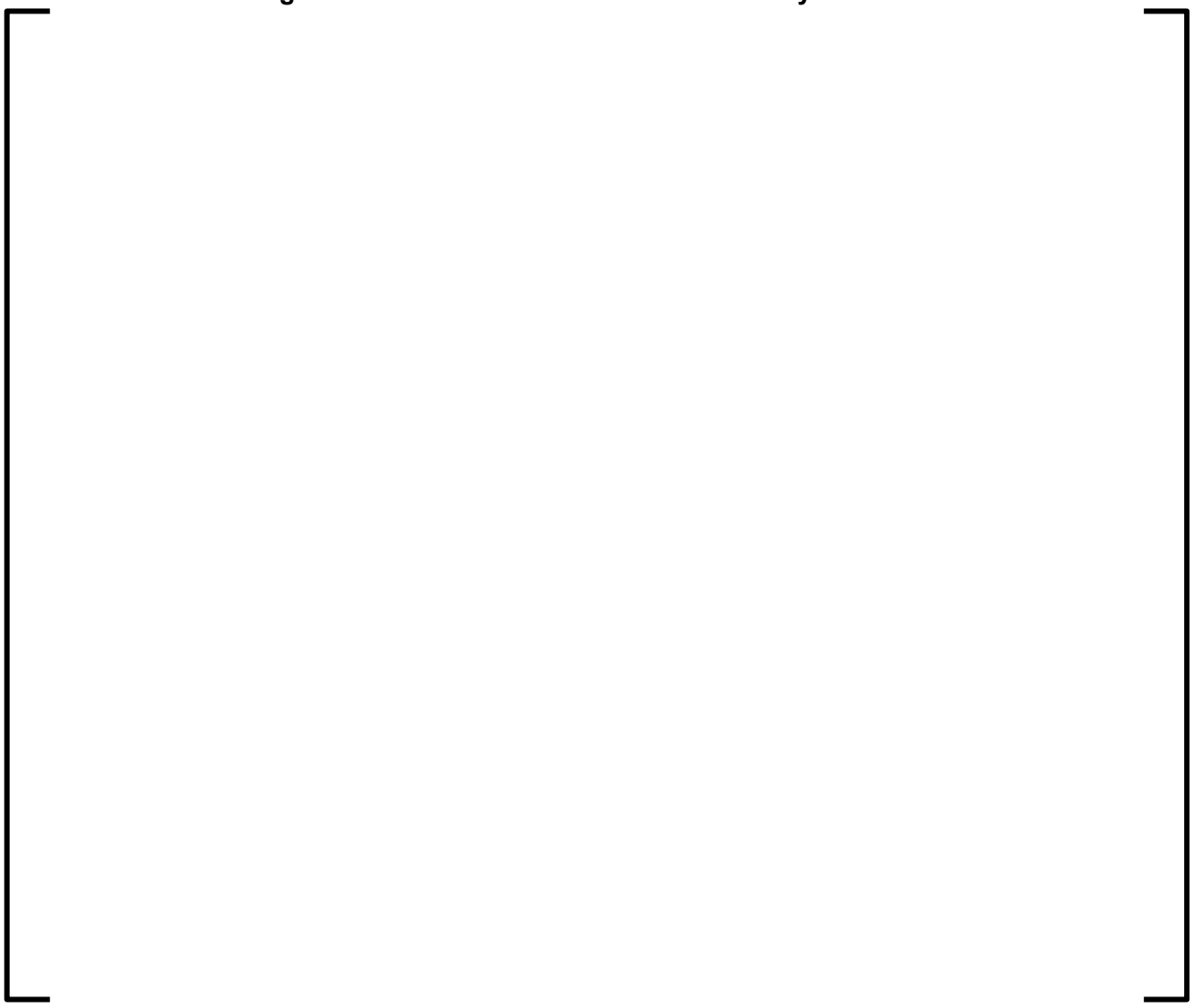


Figure 5-4—Subfunction 1: Acquisition and Prioritization of Safety I&C Commands

Figure 5-5—Subfunction 2: Acquisition and Prioritization of Operational I&C and Control Room Commands

Figure 5-6—Subfunction 3: Travel Limit Switch Responded, and Control Room Test

6.0 TEST METHODOLOGY EXAMPLE USING THE SPLM1-PC10

6.1 *Test Environment*

6.2 *Test Case Implementation for SPLM1-PC10*

6.3 *Expanding the Existing Test Environment*

Figure 6-1—Test Environment for the SPLM1 Programmed Variants

Figure 6-2—Time Curve with Inputs and Expected Outputs

Figure 6-3—Time Curve with Expected and Observed Outputs

Figure 6-4—Time Curves, Indicating Differences Between Expected and Observed Outputs

Figure 6-5—Composing Test Cases: Each Case Consists of Four Steps

Figure 6-6—Composition of the Three Basic Types of Test Cases into Elementary Steps.

Figure 6-7—Specification of Test Cases Using a Spreadsheet

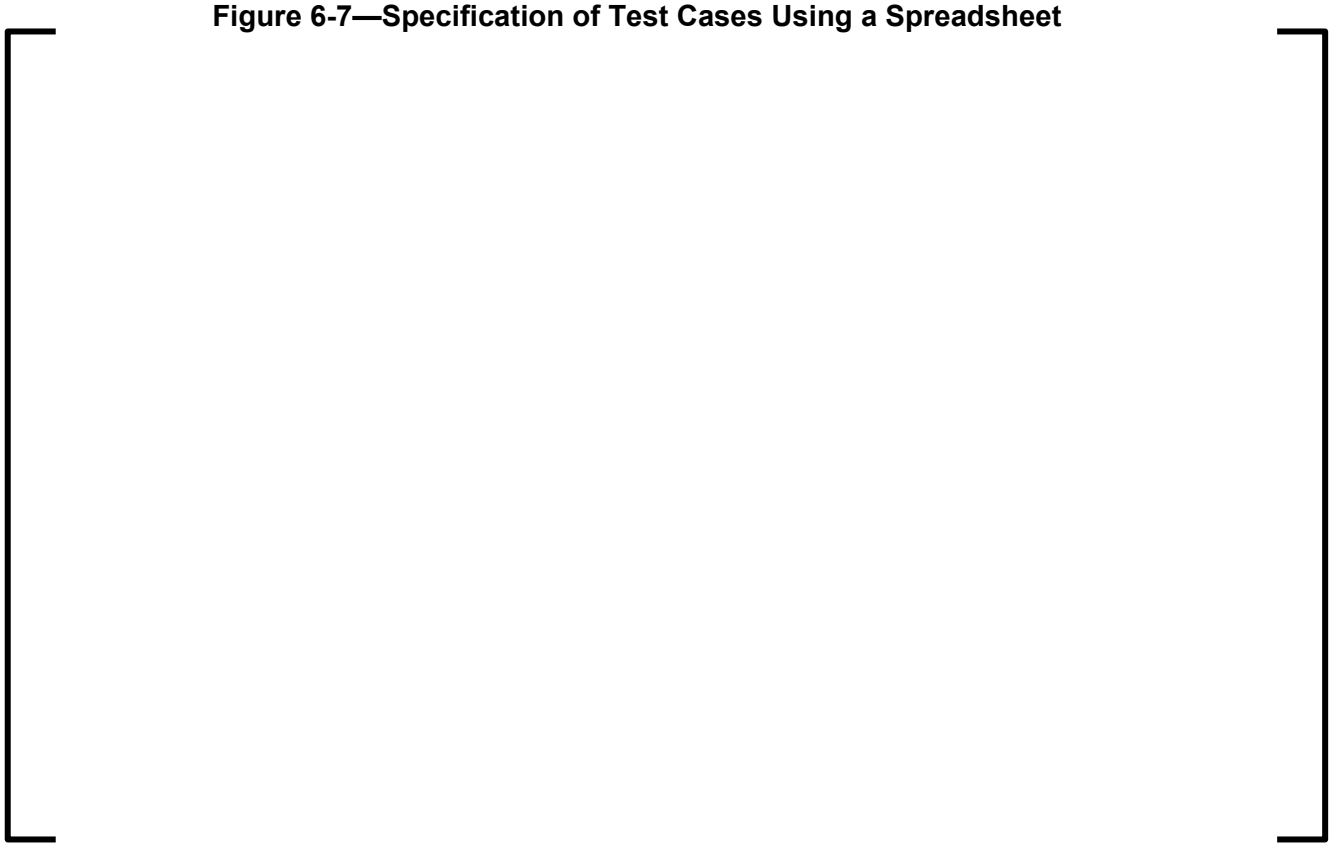


Figure 6-8—Logic Involving a Timer

Figure 6-9—Specifying Test Cases Involving the Timer

Figure 6-10—Logic Involving a Latched Actuation Signal

Figure 6-11—Specifying a Test Case Involving a Latched Actuation Signal

7.0 MANUAL VERIFICATION

7.1 *Approach*

7.2 *Rule-Based Sorting of the Test Cases*

7.3 *Tool Support and Plausibility Checks*

8.0 EXCLUSION OF INFRASTRUCTURE SIGNALS

Figure 8-1—Example of Processing of Infrastructure Signals

9.0 REFERENCES

1. NRC Interim Staff Guidance D&IC-ISG-04, “Digital Instrumentation and Controls - Task Working Group #4 - Highly-Integrated Control Rooms - Communications Issues,” March 2009.

2. Regulatory Guide 1.152, “Criteria for Digital Computers in Safety Systems of Nuclear Power Plants,” January 1996.

3. IEEE Std 7-4.3.2-2003, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,” 2003.

APPENDIX A
RULE-BASED SORTING OF TEST CASES

Figure A-1—Result After Application of Rule 0

Figure A-2—Result After Application of Rule 1

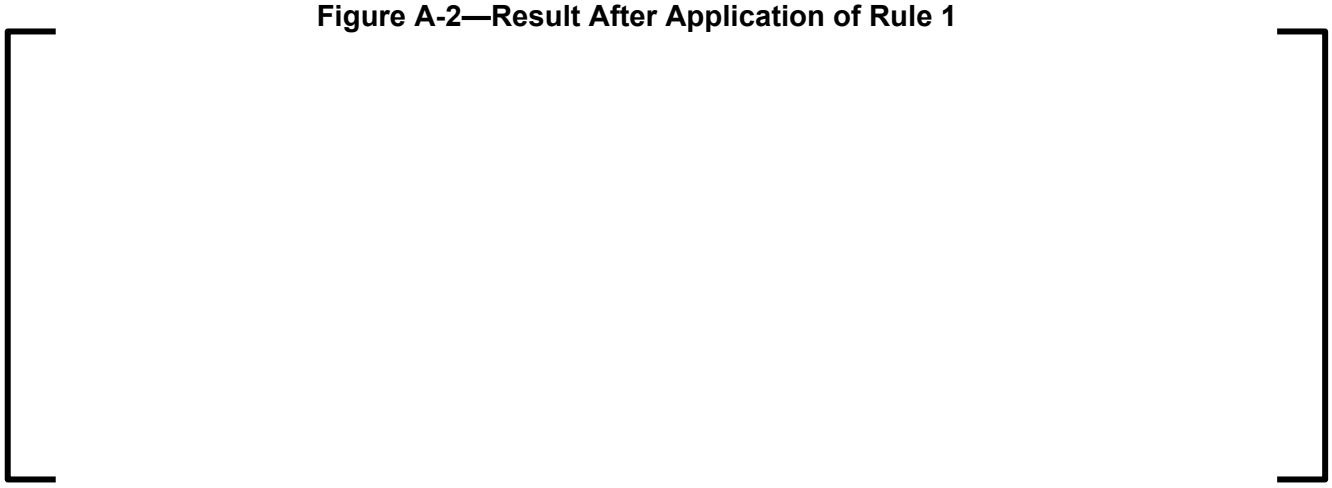


Figure A-3—Result After Application of Rule 2

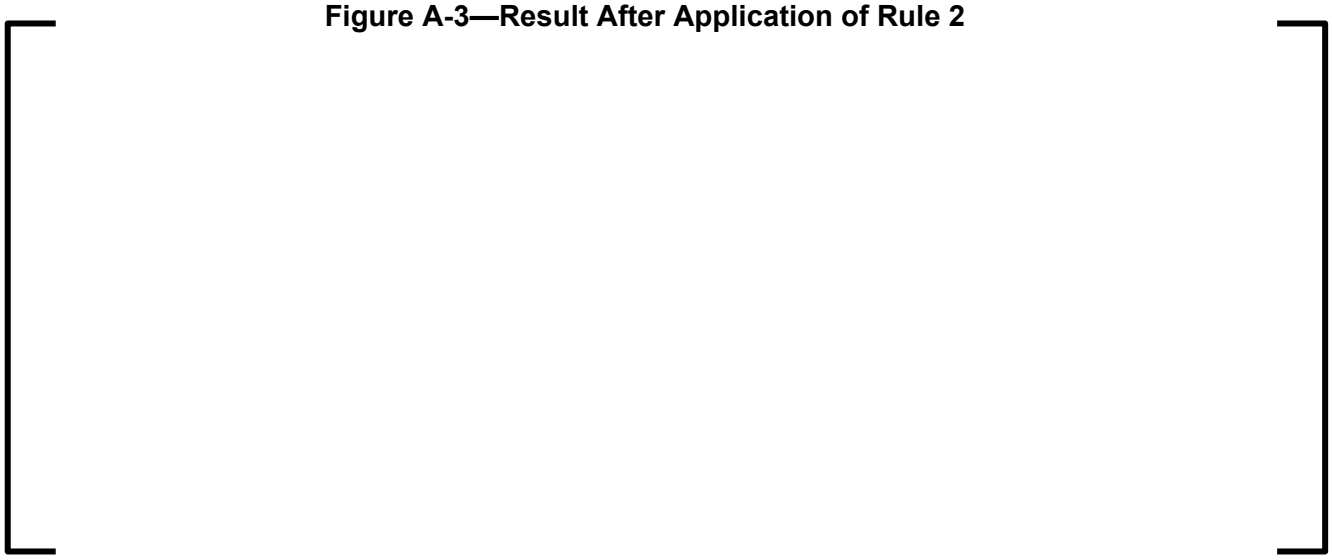


Figure A-4—Result After Application of Rule 3

Figure A-5—Result After Application of Rule 4

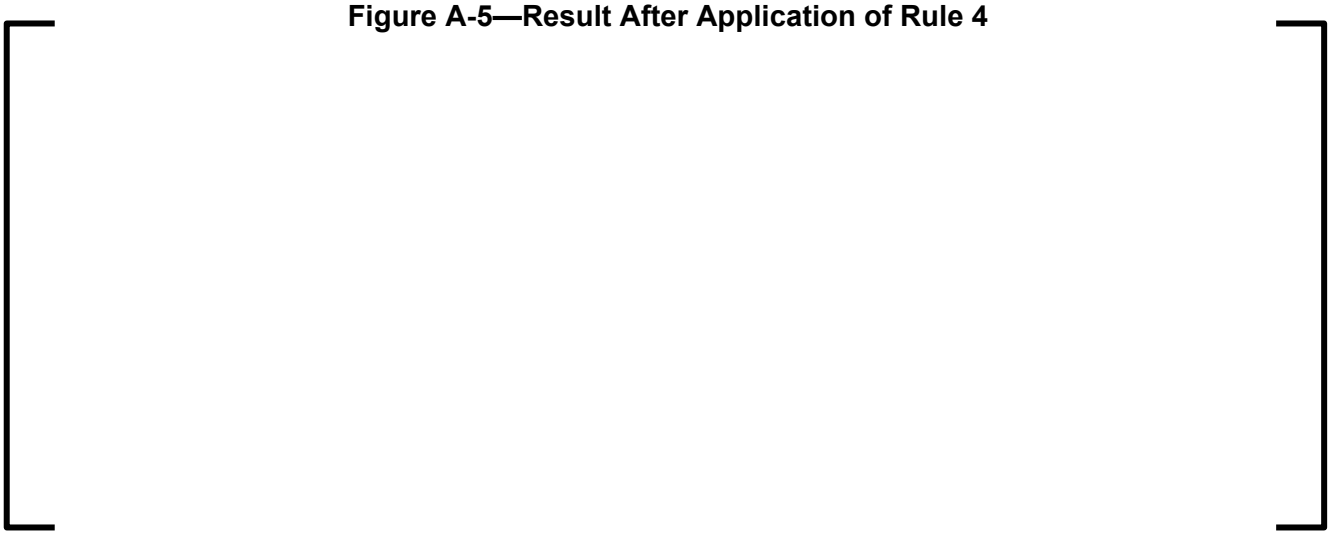
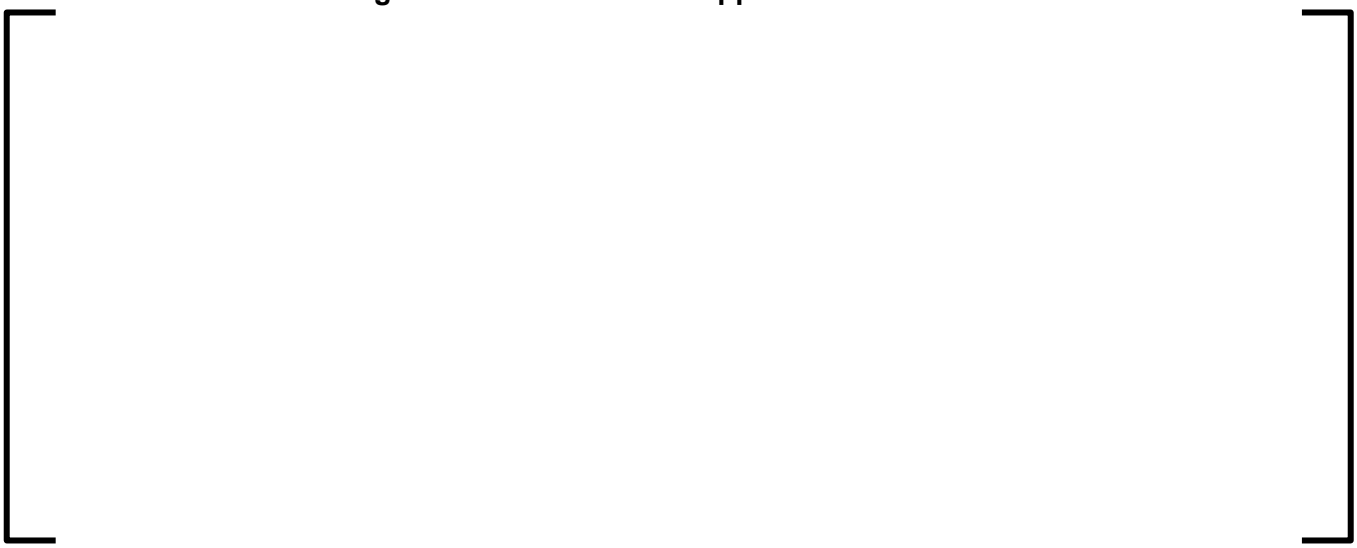


Figure A-6—Result After Application of Rule 5

Figure A-7—Result After Application of Rule 6



APPENDIX B
EXAMPLE DEBOUNCING FUNCTION

[
]

Figure B-1—Example of Debouncing Logic in the PLD