Modelling of FPGA-based Reactor Protection Systems of an Experimental Power Reactor

T J Suryono, Sudarno, S Santoso, R Maerani

Center for Nuclear Reactor Technology and Safety, National Nuclear Energy of Indonesia (BATAN), Gd. 80, Kawasan Puspiptek Serpong, Tangerang Selatan, Banten, Indonesia

E-mail: suryono@batan.go.id

Abstract. The reactor protection system of nuclear power plants including an experimental power reactor which will be built by Indonesia is a safety system that actuates the control rods to be inserted in the reactor core to absorb the neutron to stop the fission reaction and then shut down the reactor (reactor trip). The reactor protection system (RPS) is actuated when the level of signals from the sensors of important components in the reactors deviates from the setpoint determined in the bi-stable processor of the RPS. RPS for the experimental power reactor has 3 redundant channels for reliability and to minimize fake signals from the sensors due to electrical noise. It can be done by selecting the channels in local coincidence logic in the RPS by voting 2 of 3 channels which are eligible to generate actuation signals to trip the reactor. Recently, the RPSs are based on the programmable logic controller (PLC). However, now the trend changes to FPGA-based RPS because of its simplicity and reliability. This paper investigates the model of the FPGA-based RPS for an experimental power reactor and the functionality of each component of the model. The results show that the model can represent the functionality of RPS for the experimental power reactor.

1. Introduction

Nuclear Energy Agency of Indonesia (BATAN) as a technical support organization, has functions to conduct research and development of reactor safety and technology. For the first project and to increase the public acceptance, BATAN plants to build a non-commercial experimental power reactor (RDE) of 10 MWth to provide electricity around Puspiptek area [1], [2]. The design of RDE is based on the 10 MW High Temperature Gas-cooled Test Reactor (HTR-10) which is a modular pebble-bed type reactor. Unlike light water reactors, which use water as a cooler and a moderator, the HTR uses graphite for moderator and helium gas for cooling the reactor. One of the important components of the experimental power reactor is instrumentation and controls. The instrumentations including some main components such as pumps, valves, and motors. All the components should be monitored and controlled to make sure that the reactor is operated safely. The controls can be conducted by automatic systems or by human actions.

Reactor protection system (RPS), an example of an automatic system, will break the current source to the control rod drive mechanism and drop the control rods into the reactor core to stop the fission reaction and trip the reactor if some monitored process variables exceeded the safety operation criteria
of the plant. Then, the human actions are needed to cool the reactor by removing the residual heat and
to prevent the release of radioactive material to the environment.

Currently, most of the RPSs are based on the computer and programmable logic controller (PLC) [3][4][5]. However, because of the complexity of new design of RPSs that difficult to handle by the PLC, and the high maintenance cost of old RPSs [5], field programable gate array (FPGA)-based RPS are developed. FPGA-based RPS system offers some advantages compared with PLC-based systems. FPGA is easy to verify and validate, and also retain to cyberattacks because it is not based on an operating system (OS) [8][4]. Although FPGA has not been widely used in nuclear industries, it offers reliability especially for handling complex systems [8].

An HTR-10-based RDE, a new generation of NPP, will implement FPGA-based RPS. Therefore, in
order to design the RPS, a good understanding of the principle and the functionality of FPGA-based
RPS is important. This paper discusses the modeling of the FPGA-based RPS for the RDE as one of the
methods to achieve that goal. The model is based on the functional analysis of the RPS and the
requirement analysis of designing the FPGA and RPS. In addition, the model uses simple components
that can represent the functionality of the RPS including the inputs (initiation level), the FPGA-based
components (logic level), and the outputs (actuation level). Moreover, the functional analysis of the
model is also discussed.

2. Overview of RPS of an Experimental Power Reactor

The RDE is equipped with the RPSs that are used to monitor critical plant process variables throughout
the operation of the plant and to initiate a reactor trip if a limiting safety system setting (a trip setpoint)
of those process variables is exceeded. Table 1 shows the example of process variables of the RDE and
their trip setpoints. It can be seen from Table 1, for example, if the helium hot temperature is more than
740°C, the initiating and actuating signal to trip the reactor is generated in the RPS.

| No | Process variable   | Description                                                                 | Trip setpoints          |
|----|--------------------|------------------------------------------------------------------------------|-------------------------|
| 1  | Neutron flux       | Indication of spatial and temporal levels and variations of neutron flux      | ≥ 120% (power measuring range) |
|    |                    | Guidance for operators to the critical approach                             | ≥ 200% (middle measuring range) |
|    |                    | Reactor power measurement                                                   |                         |
| 2  | Helium temperature | Measurements of cold and hot helium temperatures                           | ≥ 740°C (hot)           |
|    |                    |                                                                            | ≥ 290°C (cold)          |
| 3  | Pressure sliding   | Measurements of differential pressure in the primary system and secondary system | ≥ 0.03 MPa/min          |
|    | rate               | Indication of system components or structures failure                       |                         |
| 4  | Flow rate          | Calculation of the amount of heat removed from the core                     | ≤ 0.75 or ≥ 1.3         |

Figure 1 shows the block diagram of the RPS of an RDE. It consists of 3 redundancy channels, each channel has bistable processors (BP), local coincidence logic (LCL) processors, and initiation logic. The BPs and LCLs are connected with a cross channel communication both in the same channel and to other channels. The inputs of the RPS are the conditioned signals of the process variable from the transmitter and measured in the sensors. The signals are converted from analog to digital (represented by the voltage signal with the value 0 V to 5 V) so that they can be processed in the digital RPS.

The digital signal is processed and then compared with the trip setpoint in the BP in each channel. If the value is exceeded the safety limit setting, the signal is considered as an initiating signal. The
initiating signal, together with those from other channels are forwarded to the LCL processor through the cross-channel communication link. The LCL processors vote 2 out of 3 (2oo3) signals that indicate trip status to generate an actuation signal to trip the reactor. This process is conducted parallelly in each LCL processor in each channel (reactor trip A, reactor trip B, and reactor trip C).

Figure 1. Block diagram of RPS of the RDE [7]

3. Overview of FPGA

Nowadays, nuclear power plants are equipped with computer-based instrumentation and control (I&C) systems and digital-based RPS [9]. Most of them used a microcontroller-based system such as a programmable logic controller (PLC) [8][5][4]. However, due to the complexity of new RPSs that needs higher integrity level and higher demands of verification and validation activities that are difficult to conduct in PLC, and high maintenance cost of PLC-based RPS [10], the FPGA-based RPS is developed.

Compare with the PLC, FPGA is a less complex system, easy to be verified and validated, and resistant to cyberattack because it is not based on the operating system (OS) and other applications. Literature [8] summarizes the comparison between PLC and FPGA in terms of strength, weakness, opportunity, and threat as can be seen in Table 2. The design of FPGA is based on the system engineering approach: need/system analysis, requirement analysis, functional analysis, design synthesis, and design verification and validation [8]. The need analysis is based on the strength, weakness, opportunity, and threat of the FPGA as in Table 2. Although the FPGA still has weakness compare with the PLC, it can handle the complexity of the new RPS, increase plan reliability, and resistance to cyberattacks. The requirement analysis includes regulatory and performance requirements. In the design process of FPGA, some standards and regulatory guides should be referred such as IEC 62566 (Development of hardware development language HDL-programmed integrated circuits for systems performing category A functions). The performance requirements include the identification of the performance parameters such as technical, operation, and response time of the RPS. The functional analysis is related to defining the allocation of hardware and software including some algorithms and logic.

The design should be synthesis by elaborating and optimizing the algorithms and the datapath of the design. The FPGA is designed using VHDL or Verilog code and should be verified by using VHDL or Verilog simulator and validated by mapping it into the FPGA board and test the design. VHDL (very-high-speed integrated circuit hardware description language), that will be used in this research, is a hardware description language that is used to design an FPGA-based system. The guidance for using the VHDL is provided in IEEE 1076 (IEEE Standard VHDL Language reference manual) [12].
Table 2. Comparison between PLC and FPGA [8]

| Items          | PLC                                           | FPGA                                      |
|----------------|-----------------------------------------------|-------------------------------------------|
| Strength       | Experienced (it has been used for more than 3 years) | Less verification and validation load, long term supportability |
| Weakness       | Difficult to verify, quick obsolescence       | Relatively new in the nuclear industry, less experience |
| Opportunity    | Performance and computational capability with large data handling | Less complex, increase plant availability and reliability, resistance to cyber attack |
| Threat         | Common cause failure to software error, complexity, cyberattack vulnerability | Vulnerability to radiation (if static RAM-based FPGA is used) |

Figure 2 shows the FPGA design flow. The design starts with the design entry by writing code VHDL/Verilog into the design. It also includes making the schematic diagram of the design. Then, the design is synthesized using VHDL/Verilog simulator to analyze the behavior of the design. In the mapping place and route phase, the map, place, and route of the design are validated using simulation and timing analysis. The last phase is generating bitstream and mapping into the FPGA board for testing.

![FPGA Design Flow](image)

**Figure 2.** FPGA design flow [11]

4. Methodology
The modeling of FPGA-based RPS includes some processes such as functional analysis, requirement analysis, modeling, and feasibility analysis. The functional analysis involves the identification of the components of RPS of the RDE and their functions. In addition, the classifications of RPS based on the signal/information generated that are initiation level, logic level, and actuation level is conducted. The next step is the components analysis that includes identification of hardware and software to model the RPS that meets the RPS design criteria. Then, the model of FPGA-based RPS is created using the components identified from the previous step. Finally, the model should be analyzed using a case study to investigate the feasibility so that it can represent the functional of FPGA-based RPS of the RDE.
5. Results and Discussions

5.1. Functional analysis
The RPS can be classified into three functional blocks: input (initiation level), logic process (logic level), and outputs (actuation level).

Initiation level
The initiation level includes the generation of initiation signals that indicate a trip state. It involves the data acquisition and the signal conditioning of the process variable measured by the sensor and transducer, the conversion of the signal into a digital value, and the comparison of the value with the predefined safety limit (trip setpoint) in the bistable processor.

Logic level
The initiation signals from the bistable processor are sent to the all LCL processor in the redundancy channels. Then the LCL processors conduct the voting and gating logic of the 2 out of the 3 coincidence initiation signals (2oo3) to generate the actuation signals.

Actuation level
It includes the conversion of the actuation signals into trip signals to activate the protective action. In the case of the reactor trip, it will break the electric current flowing through the control rod drive mechanism that will cause the control rods dropped into the reactor core and trip the reactor.

5.2. Component analysis
The components analysis includes the analysis of hardware and software of the RPS model. The hardware includes the inputs (sensor, transducer, and analog to digital converter/ADC), logic process (bistable processor and LCL processor), and outputs (relay or LEDs). While the software is the hardware description language (HDL) to design the FPGA.

5.3. Hardware analysis
5.3.1. Inputs
Sensors and transducers
In industrial applications, the process variables are generated by the components of the plant and measured in sensors and converted into voltage signals. Then, the transmitter (transducer) transforms the signals into electrical current 4-20 mA and delivers them to the receiver (controller, ADC, or display unit) through the wire for further processing, as can be seen in Figure 3. The reasons are that the electrical current can be transmitted through the media over a distance, and resistant to the noise [13]. Therefore, this model implements the 4-20mA current source generator as the source of the process variable signals.

The value of the process variable is proportional to the range of current 4-20 mA. A 4 mA represents 0% of the scale of the process variable, while the 20 mA represents 100% of scale, as can be seen in Figure 4. The formula for converting the process variable to the current 4-20 mA follows the linear correlation

\[ y = mx + c \]  \hspace{1cm} (1)

where \( y \) = output from instrument, \( m \) = slope, \( x \) = input of instrument, \( c \) = \( y \)-intercept point.

\[ y = mx + c \]

Figure 3. Electrical current loop 4-20 mA
The value of \( m \) can be calculated as follows:

\[
m = \frac{(20 - 4)}{(100 - 0)} = \frac{16}{100}
\]

\[
y = \frac{16}{100} x + c
\]

For the value of scale, \( x = 0, y = 100, c = 4 \), therefore

\[
y = \left( \frac{16}{100} \right) x + 4
\] (2)

The scale means that if the measurement range of hot helium gas temperature of the EPR is from 0 to 850°C, as mentioned in [15], the percentage value of 100°C will be equal to

\[
\frac{100}{(850 - 0)} \times 100\% = 11.76\%
\]

and the current related to the scale is

\[
y = \left( \frac{16}{100} \right) 11.76 + 4 = 5.88 \, mA
\]

\textit{Current to voltage converter}

As in Figure 3, the 24 VDC power supply is used to supply the current loop 4-20 mA and delivered through the wire for a distance. In order to be processed in the receiver (output side), such as display unit and other controllers (usually digital systems), the current should be converted into voltage value (1-5 V or 1-10 V). It can be conducted by connecting a resistance to the loop, as can be seen in Figure 3. The value of \( R \) can be determined by applying formula \( V = I \times R \). For the output voltage 1-5 V, the resistance value is 250 ohms.

\textit{Analog to digital converter}

Figure 5 shows the simple concepts of 16 bits ADC. Based on [16] 16 bits ADC means that the ADC can detect \( 2^{16} \) or 65536 discrete analog levels. In other words, the resolution of the ADC is 65535 (the last value of the 16 bits). This ADC value can be related to the voltage input using the following formula:

\[
\frac{\text{resolution of } \text{ADC}}{\text{system voltage}} = \frac{\text{ADC reading}}{\text{analog voltage measured}}
\]
In the case of the model system of this paper, the system voltage is 5V and the resolution of ADC is 65535 (5V is related to 65535).

Figure 5. A simple 16 bits ADC

| Process variable | Current value | Voltage value | ADC value          |
|------------------|---------------|---------------|--------------------|
| 740°C            | 17.926 mA     | 4.48 V        | 59647 (dec)        |
|                  |               |               | 1110100011111111   (bin) |
| 290°C            | 17.26 mA      | 4.315 V       | 56557 (dec)        |
|                  |               |               | 110111001101101    (bin) |

As an example, let consider the value of the process variable (trip setpoint) of hot helium temperature and cold helium temperature provided in Table 1, which are 740°C and 290°C, respectively. First, the value should be converted into the current 4-20mA using eq.1 and eq.2. Then, the resulted current value is converted into a voltage value by multiplying it with resistance 250 ohms. Finally, the voltage is correlated with the ADC value. The results are shown in Table 3. The ADC values of the process variables will be processed by the bistable processor and compared with the trip setpoints to generate the initiating signals.

5.3.2. Logic process

Bistable processor
The bistable processor stores predefined safety limits of process variables that cause a reactor trip if the value is violated (trip setpoint). The trip setpoint is set based on the design basis accident analysis. For the EPR, the setpoints are provided in Table 1. In order to be processed by the bistable processor, the trip setpoints should be converted into a digital value (the results of calculation of section V.A) and the examples are provided in Table 3. The process variable from the sensor is compared with the trip setpoint in the bistable processor. If the value reaches or beyond the threshold, the signal is considered as an initiating signal. Therefore, the functionality of the bistable processor is similar to the logic comparator.

Figure 7 shows the logic diagram of a 1-bit comparator. The diagram consists of an inverter, AND, and OR gates and there are 3 possible outputs. It is considered that A is the measured process variable and B is the trip setpoint. In the case of cold helium temperature, the sensor measured that the value is 290°C or 56557 (from Table 3). It will be an input A of the comparator. Then, the value is compared with the input B (trip setpoint = 290°C or 56557). It can be seen that the value of A is equal to B (A=B). In other words, the value of the process variable reaches the trip setpoint value. Therefore, the process variable (cold helium temperature) is set as an initiating signal for the reactor trip.
The RPS of the EPR has 3 redundancy channels, each channel has a bistable processor and an LCL processor. The initiating signal generated by the bistable processor will be delivered to all LCL processors (3 LCL processors) through cross channel communication. However, not all the 3 signals are used to produce the actuation signal, only 2 coincidence signals are needed. This process is called the 2 out of 3 (2oo3) method and conducted by the LCL processor.

The functionality of the LCL processor can be represented by the logic 2 out of 3, as can be seen in Figure 7. The diagram consists of 3 AND gates and one OR gate. The logic AB, AC, and BC represent the coincidence logic of process variable from the sensor of channel A&B, channel A&C, and channel B&C. Table 4 shows the truth table of the logic. It can be seen that output 1 is achieved if the 2 input values are 1.

| A | B | C | AB | AC | BC | AC+AC+BC |
|---|---|---|----|----|----|-----------|
| 0 | 0 | 0 | 0  | 0  | 0  | 0         |
| 0 | 0 | 1 | 0  | 0  | 0  | 0         |
| 0 | 1 | 0 | 0  | 0  | 0  | 0         |
| 0 | 1 | 1 | 0  | 0  | 0  | 1         |
| 1 | 0 | 0 | 0  | 0  | 0  | 0         |
| 1 | 0 | 1 | 0  | 1  | 0  | 1         |
| 1 | 1 | 0 | 1  | 0  | 0  | 1         |
| 1 | 1 | 1 | 1  | 1  | 1  | 1         |

Figure 8, based on the method mentioned in [17], shows the process of voting the 2 out of 3 initiating signals from the bistable processor. It can be seen that there are 3 possible coincidence logics: AB, AC, and BC. The bistable processor delivers the resulted initiating signals to all LCL processors in the RPS. Therefore, in each LCL there is a matrix trip: [AB1, AB2, AB3], [BC1, BC2, BC3], or [AC1, AC2, AC3] that will generate a reactor trip path that consists of the series of matrix output. For example, trip path 1 consists of a series connection of [AB1, BC1, AC1].
Based on the description of the bistable processor and LCL processor in the previous section and the research on the hardware to design the FPGA, the suitable FPGA board for this model is Digilent Basys 3, as can be seen in Figure 9. Although the FPGA board is simple and for beginner, it is reliable and has some features such as switches to represent the inputs, and LEDs and seven-segment displays to represent the outputs.

**Figure 8.** 2oo3 LCL outputs and trip path

### FPGA hardware

Outputs

As the outputs of the model, the relay that is connected to the FPGA board that functions as the LCL processor. Another way is by using the output ports of the FPGA board such as LEDs and seven-segment displays to represent the value of the process variable. Regarding the relay or switch, it will change the state (from OFF to ON) when the actuation signal is sent to the relay. On the other hand, for the same case, the LEDs on the FPGA board will be ON and the seven segments will display the value of the process variable. The relay or the LED can represent the controller of the control rod that will be triggered to drop the control rods if the actuation signal is sent to the controller.

5.4. Software analysis

There are some VDL that are commonly used in designing the FPGA-based system such as VHDL (Very High Speed Integrated Circuit Hardware Description Language) and Verilog. Based on the research, the most commonly used HDL for the chosen FPGA-board is VHDL. Therefore, this research implements VHDL. However, this paper only discusses the model of the FPGA-based system and investigates the feasibility of the model to represent the functionality of the FPGA-based RPS. Therefore, the implementation of the design including the VHDL code writing and applied to the FPGA board is not discussed. It will be the future works of this research.
5.5. Model
The proposed model of the FPGA-based RPS is shown in Fig 10. Below are the descriptions of the components of the model and the assumptions used to design and investigate the model:

1. Inputs that provide initiation signals (sensors/transducers, signal conditioners, analog to digital converter) are represented by the current generator, current to voltage generator, and ADC.
2. A comparator to model the bistable processor using FPGA board Digilent Basys 3.
3. A logic 2 out of 3 to represent the LCL processor using FPGA board Digilent Basys 3.
4. Relay or LEDs to represent the output (actuated device).
5. This model is for a single RPS channel. For the purpose of analysis, the two other identical models are used.

![Figure 10. Model of the FPGA-based RPS](image)

5.6. Case study
As a case study, a simple simulation is used to investigate the functionality of the model. It is considered that sensors in each channel measured the process variables such as hot helium temperature (channel A and B) and cold helium temperature (channel C) as can be seen in Table 5.

| Channel | Process variable        | Value °C | Current mA | Voltage V | ADC value |
|---------|-------------------------|----------|------------|-----------|-----------|
| A       | Hot helium temp         | 760      | 18.3056    | 4.5764    | 59983     |
| B       | Hot helium temp         | 750      | 18.1184    | 4.5296    | 59369     |
| C       | Cold helium temp        | 300      | 17.7136    | 4.4284    | 58043     |

The signals are converted into the electrical current value by the transducer. In this case, the signal generator is set to 18.3056 mA (A), 18.1184 (B), and 17.7136 9C. At the receiver of the input, the currents are converted into voltage value and then to a digital value by the ADC. The bistable processor examines the digital value of the process variable with the predetermined trip setpoint value stored in a bistable processor. The results are provided in Table 6. It can be seen that the process variable value is more than the trip setpoint. Therefore, the process variables are set as the initiation signals.

| Channel | Process variable        | Value | Trip setpoint |
|---------|-------------------------|-------|---------------|
| A       | Hot helium temp         | 59983 | 59647         |
| B       | Hot helium temp         | 59369 | 59647         |
| C       | Cold helium temp        | 58043 | 56557         |

The resulted initiating signals from the bistable processor are sent to all LCL processors (2oo3 logic processor) and produced 3 matrix trips ac as can be seen in Table 7. It can be seen that there is one matrix trip that indicates the coincidence logic that is AB. Therefore, this signal is set as an actuation signal.
Table 7. Matrix trips and actuation signal

| Matrix strip | Process variable                  | Actuation signal |
|--------------|-----------------------------------|------------------|
| AB           | Hot helium temp, hot helium temp  | yes              |
| AC           | Hot helium temp, cold helium temp | No               |
| BC           | Hot helium temp, cold helium temp | No               |

The last step is, in the actuation level, the actuation signal is converted to a trip signal that in the model is represented by a signal that change the state of the relay from “OFF” to “ON” or the LEDs from “OFF” to “ON”.

6. Conclusions
The modeling of FPGA-based RPS using simple equipment has been conducted. The results of the preliminary investigation of the model show that the model can represent the functionality of the RPS. The process variables are converted into current loop 4-20 mA and transformed into digital value before processing in the bistable processor. The bistable processor acts as a comparator that examines the process variable value with the trip setpoint. Then, amongst the 3 initiating signals from the bistable processor, in the LCL processor, the two coincidence signals are set as an actuated signal to turn on the relay or a LED that represents the reactor trip. Future work is related to the implementation of the model and some experiments to verify the model.

Acknowledgments
This work is funded by DIPA of PTKRN-BATAN and Insinas Ristekbrin of Ministry of Research and Technology of the 2021 fiscal year.

References
[1] M. Subekti, S. Bakhri, and G. R. Sunaryo, “The Simulator Development for RDE Reactor,” J. Phys. Conf. Ser., vol. 962, no. 1, 2018.
[2] BATAN, “Experimental Power Reactor (RDE).”
[3] J. G. Choi and D. Y. Lee, “Development of RPS trip logic based on PLD technology,” Nucl. Eng. Technol., vol. 44, no. 6, pp. 697–708, 2012.
[4] I. Ahmed, J. Jung, and G. Heo, “Design verification enhancement of field programmable gate array-based safety-critical I& C system of nuclear power plant,” Nucl. Eng. Des., vol. 317, pp. 232–241, 2017.
[5] J. Yoo, J. H. Lee, and J. S. Lee, “A research on seamless platform change of reactor protection system from PLC to FPGA,” Nucl. Eng. Technol., vol. 45, no. 4, pp. 477–488, 2013.
[6] G. Zuying and S. Lei, “Thermal hydraulic transient analysis of the HTR-10,” Nucl. Eng. Des., vol. 218, no. 1–3, pp. 65–80, 2002.
[7] S. Sudarno, A. Cahyono, D. F. Atmoko, and S. Santoso, “Detail Engineering Design Reaktor Daya Eksperimental: Sistem Proteksi Reaktor,” Badan Tenaga Nuklir Nasional (BATAN), [in Indonesian], 2018.
[8] J. Jung and I. Ahmed, “Development of Field Programmable Gate Array-based Reactor Trip Functions Using Systems Engineering Approach,” Nucl. Eng. Technol., vol. 48, no. 4, pp. 1047–1057, 2016.
[9] C. A. Verrastro et al., “IGORR Conference 2014 FPGA Based Reactor Protection System Architecture Instrumentación y Control - Centro Atómico Ezeiza - Comisión Nacional de Energía Atómica systems to guarantee that design specifications are not exceeded. The RPS has priority over T,” no. May, 2015.
[10] J. Yoo, J. Lee, and J. Lee, “A RESEARCH ON SEAMLESS PLATFORM CHANGE OF REACTOR PROTECTION SYSTEM FROM PLC TO FPGA,” Nucl. Eng. Technol., vol. 45, no. 4, pp. 477–488, 2013.
[11] C. Tanougast, A. Dandache, M. Salah, and S. Sadoudi, “Hardware Design of Embedded Systems for Security Applications,” *Embed. Syst. - High Perform. Syst. Appl. Proj.*, no. May 2014, 2012.

[12] I. C. Society, *IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-2008*, vol. 2008, no. January, 2009.

[13] T. Witt, R. Mena, and E. Cornell, “Single chip, 2-wire, 4-20mA current loop RTD temperature transmitter design,” *IECON Proc. (Industrial Electron. Conf.)*, vol. 10, pp. 2380–2383, 2014.

[14] O. Barrière, A. Echelard, and J. L. Véhel, *Lessons In Industrial Instrumentation*, vol. 17, no. January. 2012.

[15] Sudarno, S. Santoso, K. Santosa, R. Maerani, and Deswandri, “Assessment of Input Parameters and Architecture of RDE Reactor Protection System,” *J. Phys. Conf. Ser.*, vol. 1198, no. 5, 2019.

[16] SparkFun, “Analog to Digital Conversion.” [Online]. Available: https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/all. [Accessed: 25-Aug-2020].

[17] USNRC, “Plant Protection System,” *Usn. Tech. Train. Center, Westinghouse Technol. Syst. Manual. Usn. HRTD. Rev 10/08.*, p. https://www.nrc.gov/docs/ML1125/ML11251A044.pdf.