Complexity Analysis of an FPGA-Based ESF-CCS

Joyce MAYAKA[1], Jae Cheon JUNG [2]

1. Department of NPP Engineering, KEPCO International Nuclear Graduate School, Ulsan, Korea
(Tel: +82-10-9389-1610, E-mail: joyce.k.mayaka@gmail.com)
2. Department of NPP Engineering, KEPCO International Nuclear Graduate School, Ulsan, Korea
(Tel: +82-52-712-7309, E-mail: jcjung@mail.kings.ac.kr)

Abstract: In the recent past, there has been a lot of research into the use of Field Programmable Gate Arrays (FPGA) in the safety systems of Nuclear Power Plants. This interest is driven by, among other factors, the search for a platform that is relatively easy to verify, validate, and ultimately license. Current Instrumentation and Control (I&C) Systems are based on Programmable Logic Controllers (PLC), which due to their software structure, have inherent complexity. Complexity results in difficult and costly verification and validation (V&V), safety justification and system maintenance. The difficulty experienced in V&V is due to the possibility of latent errors in the software, as well as the inability to separate support functions from the primary control functions. The Engineered Safety Features Component Control System (ESF-CCS) serves to actuate the safety components in the event of an accident. The primary purpose of the ESF-CCS is to prevent the release of radioactive material, as well as to prevent damage to the core. This study proposes a design for the ESF-CCS based on the FPGA architecture and is aimed at a reduction in system complexity. The reduction in complexity is achieved by the use of flat hardware logic and the separation of logically independent functions. Following this, complexity analysis of the developed system is carried out in order to facilitate the comparison of the FPGA-based system to the PLC-based system

Keyword: Field Programmable Gate Arrays, APR1400, Engineered Safety Features Component Control System, Complexity Metrics.

1 Introduction

The safety critical instrumentation and control systems in the Nuclear Power Plant are responsible for shutting down the reactor and actuating the Engineered Safety Features in the event of an accident, to prevent damage to the plant and the release of radioactive material. As such, these systems must be developed with a high level of reliability.

Currently, most digital I&C systems are implemented on Programmable Logic Controller (PLCs) hardware. PLCs provide numerous advantages over traditional analog systems such as the capability for self-monitoring, drift free operation and advanced Human-System Interfaces. However, a major drawback of PLC-based systems is the difficulty to verify these systems due to their being software based [1]. The presence of an operating system causes the I&C system to have a large complexity overhead [2] and complexity is an indicator of system reliability [3]. The higher a systems complexity, the more difficult it is to verify, validate and maintain.

To deal with this concern, Field programmable Gate Arrays have been proposed as an alternative hardware platform. The FPGAs architecture promises a reduction in the complexity of the developed system as the final product, like traditional analog electronics, can be designed to consist of only hardware [4]. This paper is aimed at demonstrating that by adopting FPGAs in the safety I&C systems, the complexity of these systems can be reduced.

2 Complexity and Complexity Metrics

Complexity is the degree to which a process is difficult to analyze, understand or explain [5]. When selecting evaluating a system design, from a safety perspective, the simpler design options are those that accomplish the function and address potential hazards with the most confidence and clarity. The IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations [6] requires that measures should be taken to avoid unnecessary complexity. Further, the NRC considers simplicity "as supporting all fundamental design principles for developing safety systems with high reliability" [7]. By evaluating the complexity of a system, design choices can be justified or altered if the system

complexity is found to be too great.
Detection and correction of complexity problems can be achieved if the system architecture is evaluated before going on to the next phase of development. Empirical studies referred to in [8] show that there is a positive correlation between measured complexity and the number of errors found in the implemented system. Therefore, the simpler a system is, the easier it is to implement, verify, maintain, and the less likely it is to contain latent errors. The prediction and subsequent reduction of complexity has been found to result in large savings in maintenance costs and efforts [9] [10].
Complexity metrics provide a means for assessing the complexity of a system. There are numerous complexity metrics, each measuring different facets of complexity. For instance, Halstead's [11] calculation is based on operators and operands of a software program while Henry and Kafura [12] define complexity as a function of the data flow between modules. For real-time safety systems in nuclear plants, NUREG/CR-6083 [13] recommends using either Halstead's, McCabe's or Henry's metrics as they call for fewer subjective judgments. Different metrics are also suitable for use at different stages of the development lifecycle. Smidts and Li [14] ranked different metrics for their applicability in different lifecycle phases, based on expert opinion. Cyclomatic complexity was ranked in the top 3, for both the design and implementation phases. Additionally, in [15], Cyclomatic complexity was identified as a measure for use during the Design, Coding, Testing, and Operation phases to predict the reliability of software in nuclear safety-critical applications. Thus, as this study is aimed at reducing and computing complexity at the design phase, Cyclomatic complexity was chosen as a suitable metric.

In order to compute the design complexity of a given system, the complexity of the individual modules must be assessed. This cyclomatic complexity of the system modules is mathematically computed using graph theory. In the graph, each node corresponds to a section of code in which the flow is sequential. The arcs are representative of branches in the program. The Cyclomatic complexity of a graph, with n vertices, e edges, and P connected components is calculated as:

$$v\,(G) = e - n + P \qquad (1)$$

Once the module design complexity of the individual components of a design has been computed, the design complexity, S_O, of a module can be determined from its structure chart. The design complexity of a module M is evaluated as:

$$S_o = \sum_{i \in D} iv(G_i)$$

Where D is the set of descendants of M unioned with M.

3 FPGAs and Complexity

FPGAs have been used in other industries, such as aviation, and are gaining increased attention as an alternative digital platform to PLCs in NPP I&C systems, particularly for safety applications [5]. One of the principle advantages of FPGA systems is that they can result in a reduction in system complexity. This is achieved by the reduction of the hardware complexity, reduction of the logic complexity, and the separation of ancillary functions.

The implementation of FPGA-based systems can reduce hardware complexity by minimizing the number of hardware components required for a given I&C function. The FPGA circuits are configurable logic blocks that can implement the required application logic as well as the input/output functions and data communication, on a single circuit. The result of this integration is a system-on-chip architecture in which a complete I&C function is implemented on a single circuit board with fewer electronic components than has been the case with other implementations, including older analog and newer microprocessor-based implementations [4]. The reduction in hardware complexity is therefore evident in the reduction of the number of hardware components and the number of interconnections and interface points.

Compared to possible FPGA-based solutions, microprocessor-based systems include a large amount of software, such as the operating system in addition to the application software that implements the required I&C functions. This software introduces a high complexity overhead [4]. In contrast, in FPGA solutions based on 'flat-hardware logic' where the complexity is solely a function of the complexity of the I&C function to be performed.

Lastly, FPGA-based systems result in complexity reduction by their ability to process

logically independent functions separately and in parallel on the same integrated circuit. As a result, functions such as self-monitoring or configuration functions can be separated from the main I&C safety function. The capability to separate functions ensures any failure of the support functions do not interfere with the main I&C application or vice versa, thus limiting the scope of a failure within a circuit, maintaining separation and independence between functions.

4 PLC Based ESF-CCS System

The structure of a single channel of the ESF-CCS system is illustrated in Figure *1*. The system is composed of the Control Channel Gateway (CCG), Group Controller (GC), and Loop Controller (LC). The GC process system level demands from the PPS and the MI switches at the safety console. This sub-system carries out selective 2 out of 4 logic on the signals from 4 channels. In addition, it processes signals from the redundant group controller, and transmits the resolved signal to the all the LCs that control components in the system being actuated.

For the control of individual components, the CCG receives soft control commands from the ESCM as well as manual commands from the Minimum Inventory switches, via the MCR Control Panel Multiplexor (CPM). The CCG validates the signal received from the ESCM modules, and resolves the signals from 13 ESCMs and MI switches into a single command. This signal is then transmitted to the GC. The GC processes signals from redundant CCGs, and then transmits the control command to the LC responsible for the particular component.

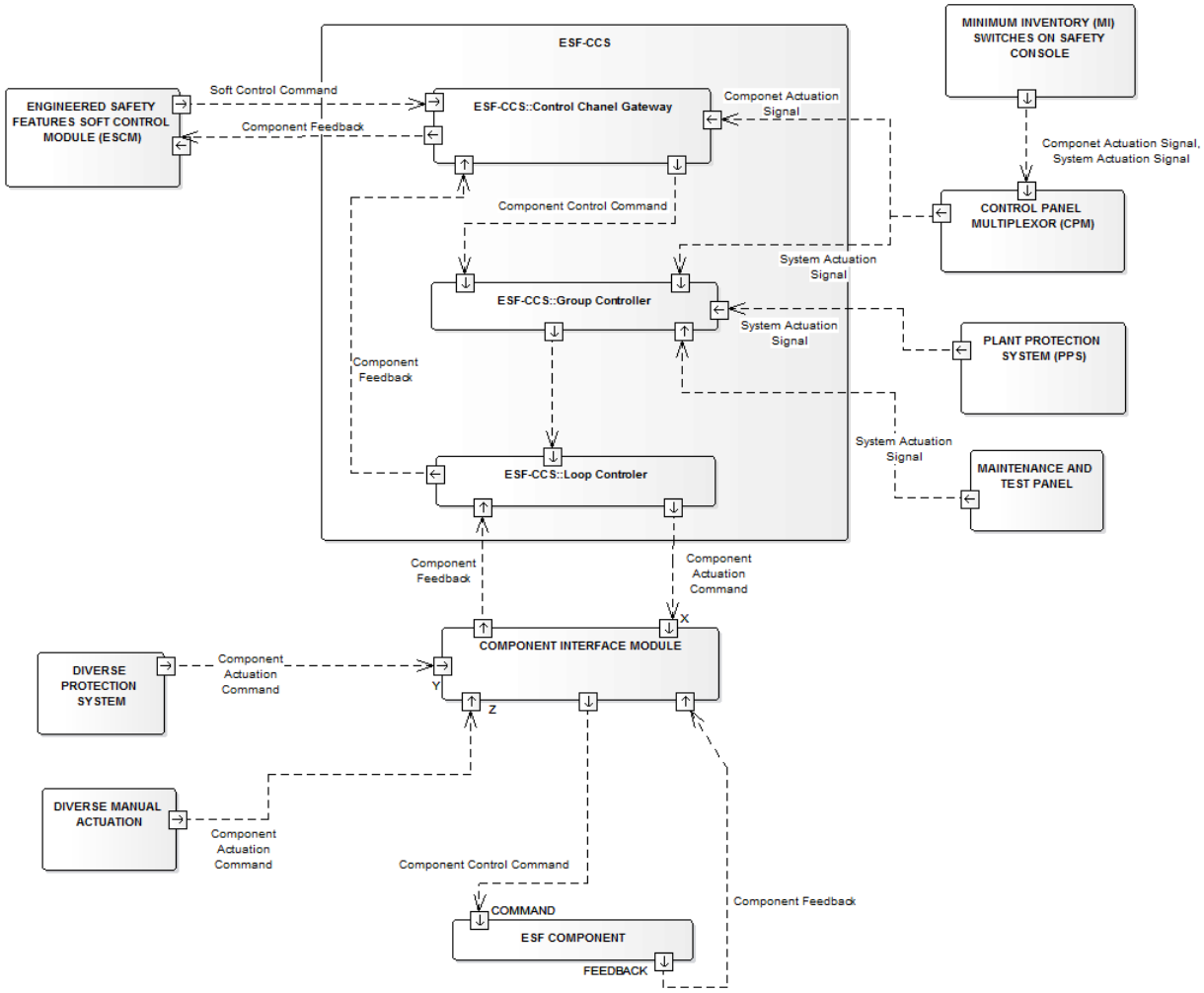

Figure 1: Interface Diagram for a single Channel of the ESF-CCS system

An analysis of the ESF-CCS system established that it performs several functions:

- System Level actuation
- Component Level actuation
- Transmission of signals (other than component actuation and feedback) to other subsystems
- Testing
- System health monitoring

Of these, the core safety functions of the ESF-CCS are system and component level actuation with the remaining three being ancillary functions. In spite of the fact that the first three functions are logically independent, their separation is difficult due to the shared operating system as well as hardware resources. The combination of these functions in a sequential thread results in software with high complexity.

In order to quantify the system complexity, the software architecture was analyzed using McCabe's Cyclomatic Complexity. The primary function of the ESF-CCS is to actuate the ESF systems at both the system level and the individual component level. System level actuation is carried out by all the processor modules of the GC while component level actuation is fulfilled by the CCG and select processor modules of the GC. The complexity of these two functions was computed separately. The structure charts [16] for the system-level actuation and component level actuation are illustrated in Figure *2*.For each functional module, the, module complexity, *iv* is indicated.

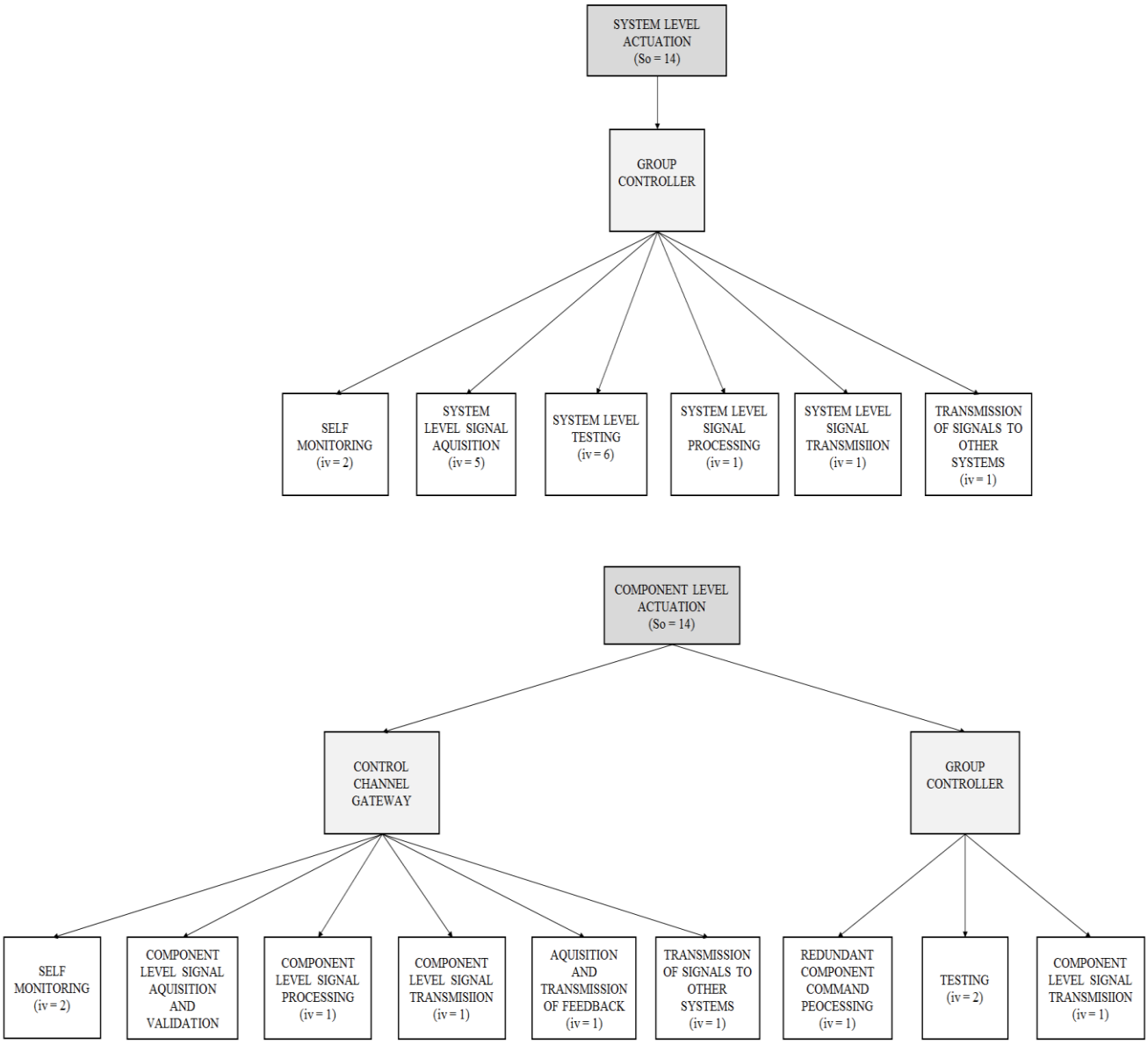

Figure 2: Structure Charts for PLC-based System Level and Component level actuation functions

5 FPGA Based ESF-CCS System

In order to reduce the system complexity, two approaches were taken: the adaptation of a System on Chip architecture and the separation of logically independent functions.

The SoC architecture is a hardware configuration in which all the functional components of the system are implemented on a single integrated circuit. In this architecture, the control logic as well as supporting operations such as communication interfaces, analog to digital converters and digital signal processing modules are contained on a single circuit board. The result of this integration is a reduction in the number of hardware components. The result of this integration is a system composed of two hardware components one for the system level actuation and the other for the component level actuation. The primary advantage of a reduction in hardware complexity is a reduction in the number of random hardware failures, [4]. In addition, simplification of the hardware structure results in ease in repair and maintenance, which ultimately improves the system reliability.

In the system analysis, several logically independent functions were identified: System level actuation, component level actuation, system health monitoring and transmission of signals to other systems. The FPGA architecture allows the system to be developed such multiple functions can be completely segregated from input, through the control logic. Ancillary functions, such as self-monitoring, were separated from the main control logic as well. The benefit of functional segregation is twofold. First, as has been stated previously, it results in a reduction in system complexity. The simplified design can enhance reliability and testability. Furthermore, the segregation of functions provides defense against the propagation of faults. It ensures that a failure in a support function, has no impact on the principle control logic, thus limiting the scope of failure within a system.

The interfaces for the resultant system is depicted in Figure 3. The system-level actuation is carried out solely by the Group Controller while the component level actuation is performed by the Control Channel Gateway. As with the PLC-based design, a flow-graph was developed and the system complexity computed.

Table 1 details the computed complexity of both the PLC-based and FPGA-based systems. McCabe in [12] and [22] proposes 10 as the upper limit for Cyclomatic complexity. Any module exhibiting a greater complexity is less manageable and testable. From this result, the complexity of the PLC based system functions is seen to be above the threshold of 10.

The complexity of both functions is seen to be reduced below the threshold. Thus by, exploiting the capabilities of the FPGA architecture, the complexity of the I&C system can be significantly reduced, resulting in an increase in system reliability.

6. Conclusion

In this work, an architecture for the ESF-CCS based on FPGA hardware, aimed at reducing the system complexity was developed. An analysis of the PLC based ESF-CCS demonstrated that the system complexity is a function of the hardware architecture. In PLC based systems it is difficult to completely separate logically independent functions due to the shared operating system. The FPGA platform can significantly reduce the system complexity due to its capability of parallel processing and the reduction in the number of hardware components required. By adopting these measures an architecture for the FPGA platform was developed and shown and its complexity measured using McCabe's cyclomatic complexity metric. The complexity of the developed system was found to be below the threshold defined for complex systems.

These results indicate that by adopting an FPGA architecture, the control logic can be reduced. This reduction in complexity can result in systems with higher reliability, which are easier to test and maintain

Table 1: Complexity of FPGA and PLC-based Systems

Function	PLC-Based System	FPGA-Based System
System Level Actuation	15	8
Component Level Actuation	14	9

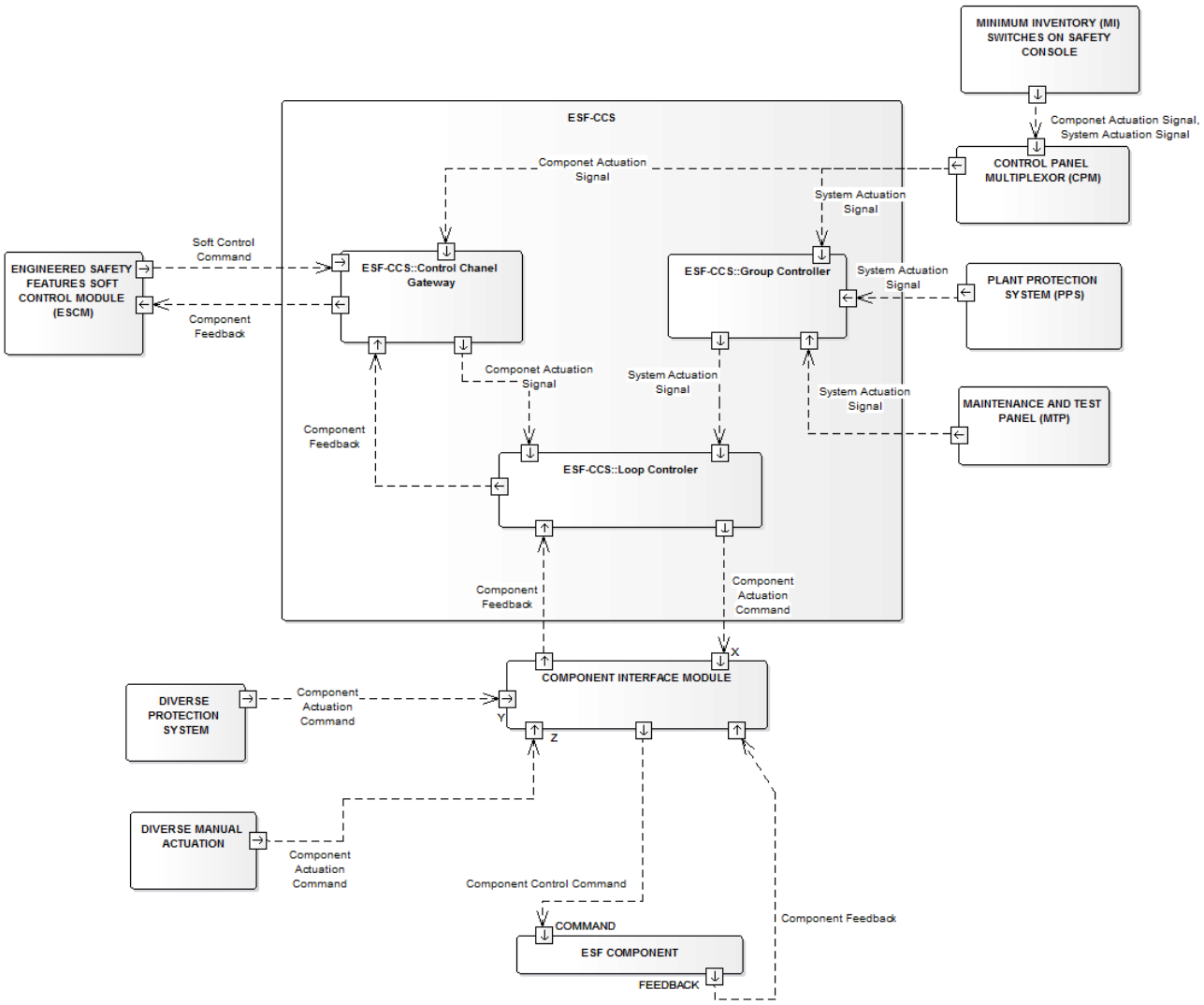

Figure 3: Interface diagram for FPGA-based system

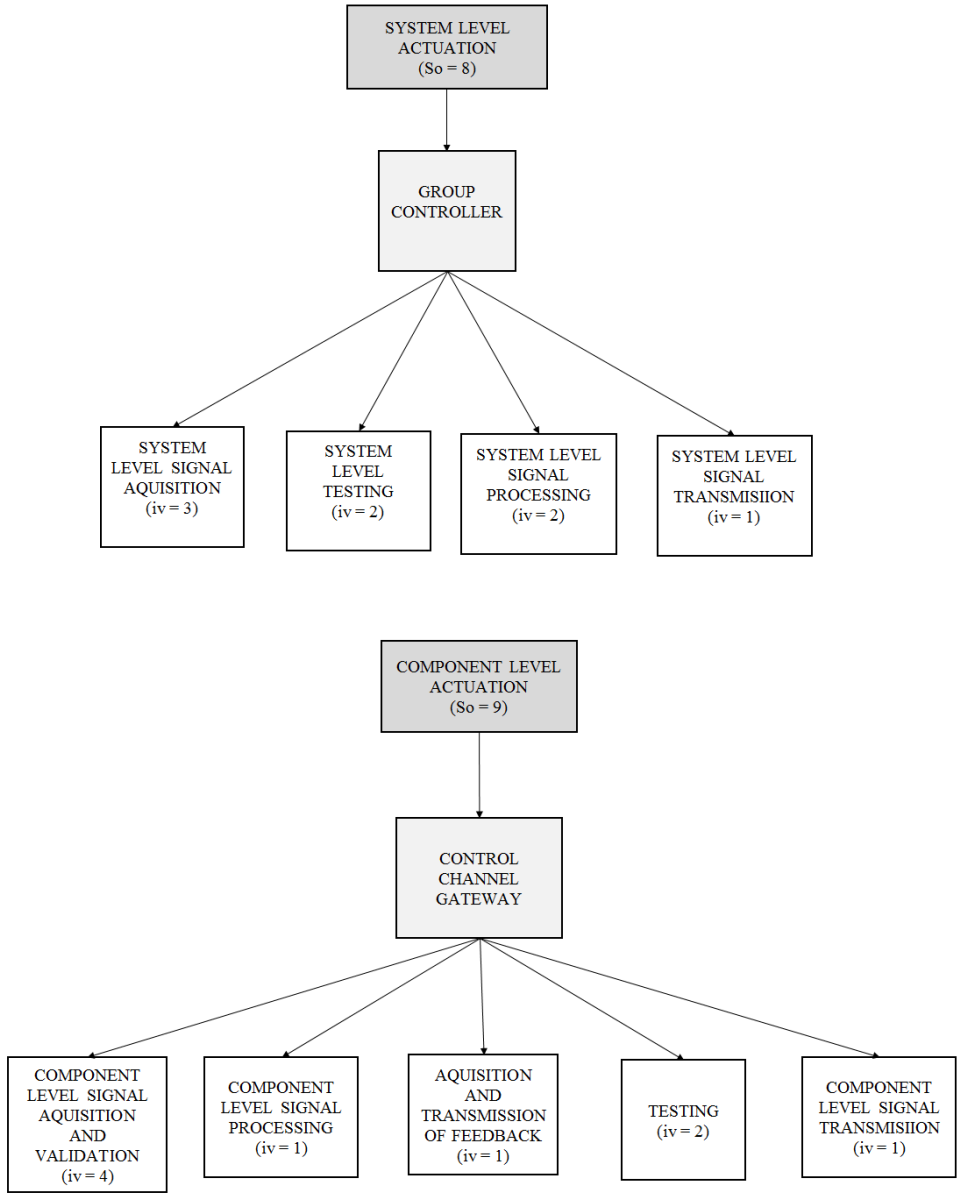

Figure 4: Structure Charts for FPGA-based System Level and Component level actuation functions

Acknowledgement

This work was supported by the 2017 Research fund of the KEPCO International Nuclear Graduate School (KINGS), Republic of Korea and the International Atomic Energy Agency.

References

[1] B. Fink, C. Killian, T. Nguyen, A. Druilhe, and Frédéric Daumas, "Guidelines on the Use of Field Programmable Gate Arrays (FPGAs) in Nuclear Power Plant I&C Systems," EPRI, Technical Report 1019181, 2009.

[2] International Atomic Energy Agency, "Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of Nuclear Power Plants," IAEA, Vienna, Technical Report NP-T-3.17, 2016.

[3] "Software Quality Metrics," U.S. Department of Transportation Federal Aviation Administration, DOT/FAA/CT-91/1, 1991.

[4] R. Fink, C. Killian, and T. Nguyen, "Recommended Approaches and Design Criteria for Application of Field Programmable Gate Arrays in Nuclear Power Plant Instrumentation and Control Systems," EPRI, Technical Report 1022983, 2011.

[5] IEEE, "Standard Glossary of Software Engineering Terrninology," IEEE Std 610.12-1990.

[6] IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations, IEEE Std 7-4.3.2, 2016.

[7] U.S Nuclear Regulatory Commission. (2013, May) Design Specific Review Standard for mPower iPWR Design. [Online]. https://www.nrc.gov/docs/ML1231/ML12318A204.pdf

[8] Thomas J. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering* , vol. SE-2, no. 4, pp. 308 - 320, December 1976.

[9] G. Stark, R.C. Durst, and C.W. Vowell, "Using Metrics in Management Decision Making," *IEEE Computer*, vol. 27, pp. 42 - 48, September 1994.

[10] S. R. Chidamber and C.F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, pp. 476- 493, June 1994.

[11] M. H. Halstead, *Elements of Software Science*. New York: Elsevier North-Holland, 1977.

[12] D. Kafura and S. Henry, "Software Structure Measures Based on Information Flow," *IEEE Transactions on Software Engineering*, vol. 5, pp. 510-518, 1981.

[13] G. G Preckshot, "Reviewing Real-Time Performance of Nuclear Reactor Safety Systems," U.S Nuclear Regulatory Commission, Washington, DC, NUREG/CR-6083, 1993.

[14] C. S. Smidts and M. Li, "Software Engineering Measures for Predicting Software Reliability in Safety Critical Digital Systems," U.S Nuclear Regulatory Commission, Washington, DC, NUREG/GR-0019, 2000.

[15] C. S. Smidts, Y. Shi, M. Li, W. Kong, and J. Dai, "A Large Scale Validation of a Methodology for Assessing Software Reliability," U. S. Nuclear Regulatory Commission, Washington, DC, NUREG/CR-7042, 2011.

[16] KEPCO and KHNP. (2014, December) APR1400 Design Control Document Tier 2. [Online]. https://www.nrc.gov/docs/ML1500/ML15006A046.pdf

[17] Hayashi Toshifumi et al., "Application of FPGA to nuclear power plant I&C systems," *Nuclear Safety and Simulation*, vol. 3, no. 1, pp. 51 - 58, March 2012.

[18] Sharad Sinha and Thambipillai Srikanthan, "Hardware Complexity Metrics for High Level Synthesis of Software Functions," in *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, 2011.

[19] M. Burth R Kazman, "Assessing Architechtural Complexity," in *Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering*, Florence, Italy, March 1988, pp. 104-112.

[20] Thomas J. McCabe and Charles W. Butler, "Design complexity measurement and testing," *Communications of the ACM*, vol. 32, no. 12, pp. 1415-1425, December 1989.

[21] M. Bobrekl et al., "Review Guidelines for Field-Programmable Gate Arrays in Nuclear Power Plant Safety Systems," U. S. Nuclear Regulatory Comission, Washington, DC, NUREG/CR-7006, 2010.

[22] U.S Nuclear Regulatory Commission, "Review Guidelines for Field Programmable Gate Arrays in Nuclear Power Plant Safety Systems," U.S Nuclear Regulatory Commission, Washington, DC, NUREG/CR-7006 2009.