KNS 2016 Autumn
Meeting 10.27~28

Hazard Analysis of Software Requirements Specification for Process Module of FPGA-based Controllers in NPP

Sejin Jung, Eui-Sub Kim, Junbeom Yoo*, Jong Yong Keum and Jang-Soo Lee

Dependable Software laboratory
Konkuk University

Contents

- Introduction

- Software hazard analysis with two approaches

- Discussion of the results

- Conclusion

Introduction

- Safety systems like nuclear I&C should be identified that hazard or risk in systems are acceptably safe

- Also, software in these systems should be analyzed before used

Software hazard analysis *"… eliminates or controls software hazards and hazards related to interfaces between the software and the system (including hardware and human components). It includes analyzing the requirements, design, code, user interfaces and changes (NIST 1993)*

- NUREG/CR-6430 proposes the method for performing software hazard analysis
  - It proposes applicable methods and guide phrases
  - HAZOP is introduced in NUREG/CR-6430 to apply guide phrases

Software Hazard Analysis

- Analysis method in NUREG/CR-6430 had been used in Korea reactor protection systems for PLC development

  - Appropriate guide phrases and analysis process are selected and applied
  - NUREG/CR-6430 provides useful methods is able to be identified

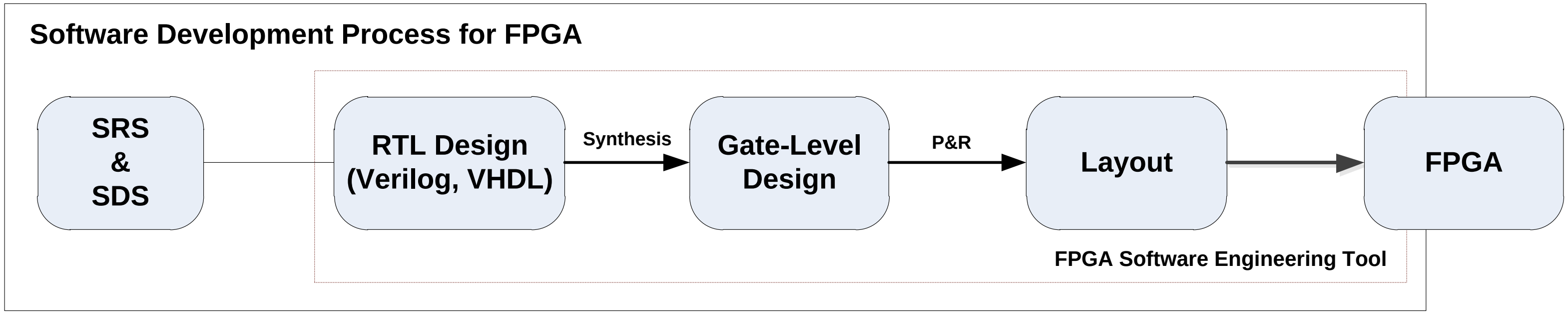
- FPGA has received much attention from nuclear industry as an alternative platform of PLC to digital I&C system

  - FPGA software also should be analyzed before used
  - Using NUREG/CR-6430 methods may be applicable choice

Software Hazard Analysis

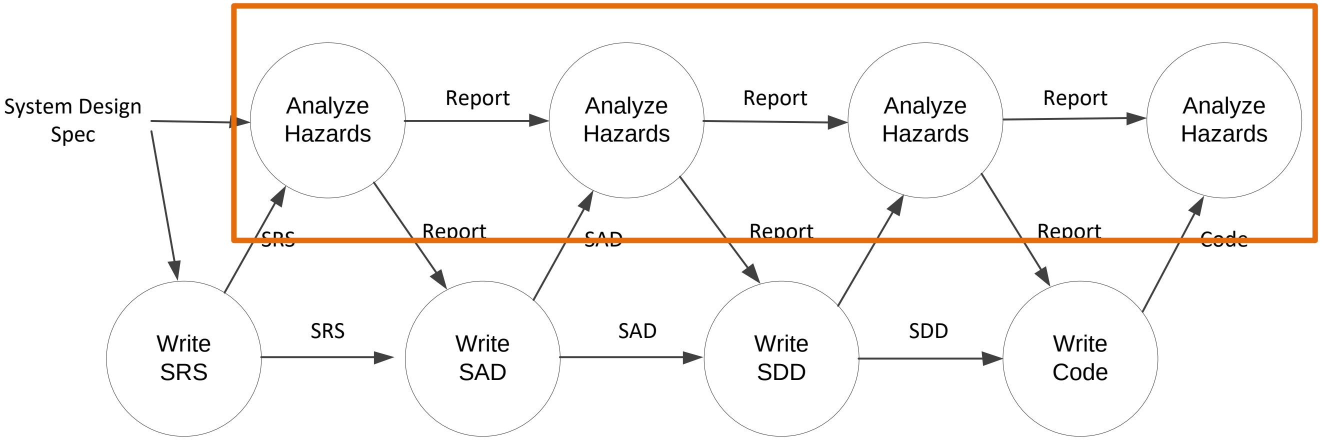
- However, FPGA has a different development process PLC, since it is a hardware-based platform
  - So, software hazard analysis with NUREG/CR-6430 need to consider the applicability of methods

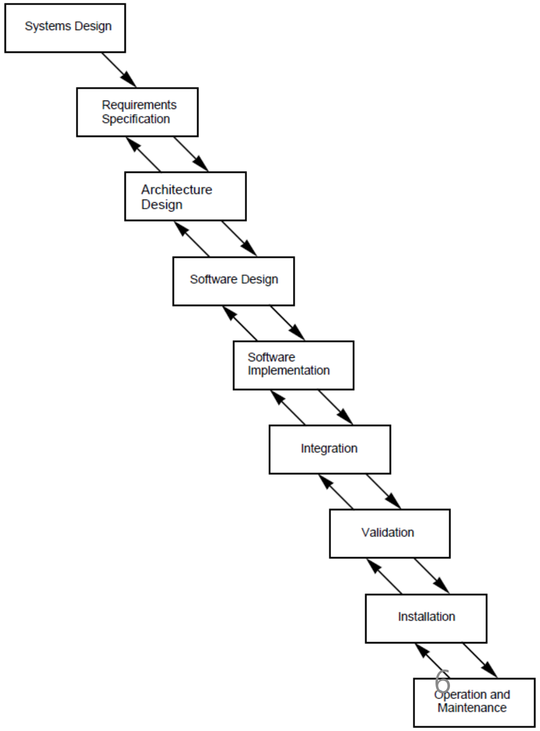
- We perform the hazard analysis methods of NUREG/CR-6430
  - Target is prototype version of SW requirements specification of module in FPGA-based controllers
  - We perform comparing analysis with analysis results of HAZOP which is applied normal methods


Software Development Process for FPGA
SRS & SDS
RTL Design (Verilog, VHDL)
Synthesis
Gate-Level Design
P&R
Layout
FPGA
FPGA Software Engineering Tool

NUREG/CR-6430: Software Hazard Analysis

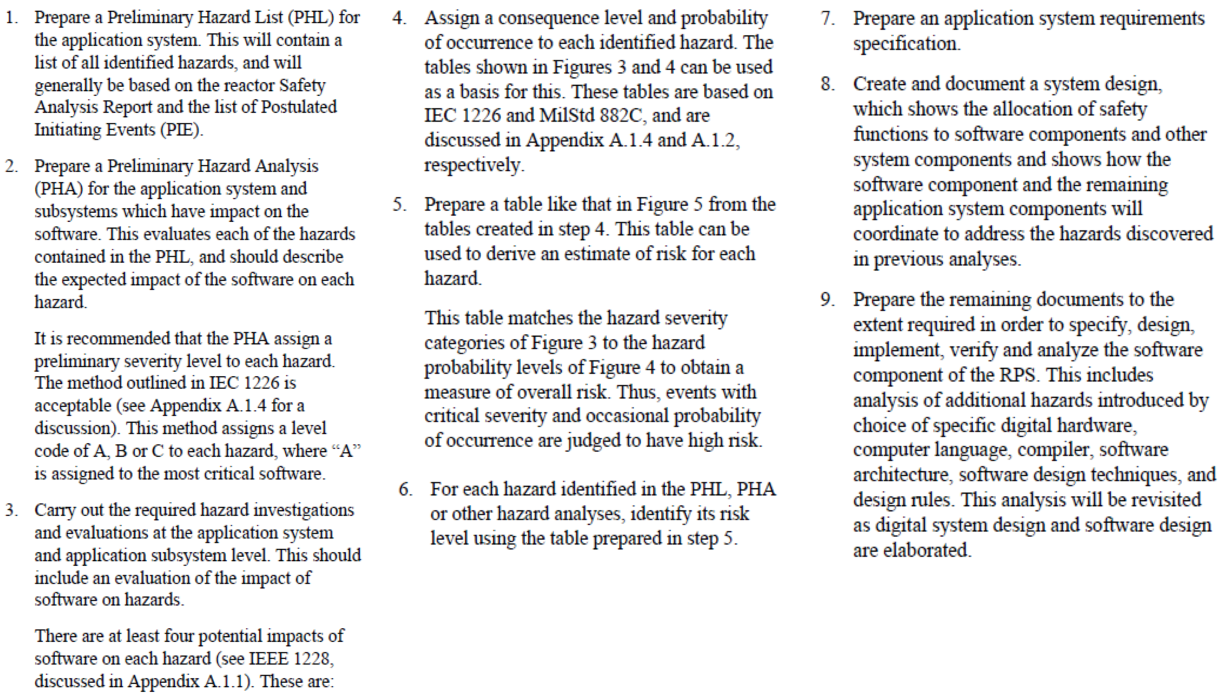
- NUREG/CR-6430 proposes the software hazard analysis methods
  - The method consists of considering software life cycle
  - It does not fix the analysis techniques
  - It also provides guide phrases to apply software





NUREG/CR-6430: Hazard Analysis of Requirements

- Prerequisites to software hazard analysis
  - Consisting of eight step of process
    - Preparing PHL
    - Performing PHA
    - Assigning consequence level and probability
    - Identifying risk of hazards
    - Identifying requirements specification of system and safety function



1. Prepare a Preliminary Hazard List (PHL) for the application system. This will contain a list of all identified hazards, and will generally be based on the reactor Safety Analysis Report and the list of Postulated Initiating Events (PIE).

2. Prepare a Preliminary Hazard Analysis (PHA) for the application system and subsystems which have impact on the software. This evaluates each of the hazards contained in the PHL, and should describe the expected impact of the software on each hazard.

   It is recommended that the PHA assign a preliminary severity level to each hazard. The method outlined in IEC 1226 is acceptable (see Appendix A.1.4 for a discussion). This method assigns a level code of A, B or C to each hazard, where "A" is assigned to the most critical software.

3. Carry out the required hazard investigations and evaluations at the application system and application subsystem level. This should include an evaluation of the impact of software on hazards.

   There are at least four potential impacts of software on each hazard (see IEEE 1228, discussed in Appendix A.1.1). These are:

4. Assign a consequence level and probability of occurrence to each identified hazard. The tables shown in Figures 3 and 4 can be used as a basis for this. These tables are based on IEC 1226 and MilStd 882C, and are discussed in Appendix A.1.4 and A.1.2, respectively.

5. Prepare a table like that in Figure 5 from the tables created in step 4. This table can be used to derive an estimate of risk for each hazard.

   This table matches the hazard severity categories of Figure 3 to the hazard probability levels of Figure 4 to obtain a measure of overall risk. Thus, events with critical severity and occasional probability of occurrence are judged to have high risk.

6. For each hazard identified in the PHL, PHA or other hazard analyses, identify its risk level using the table prepared in step 5.

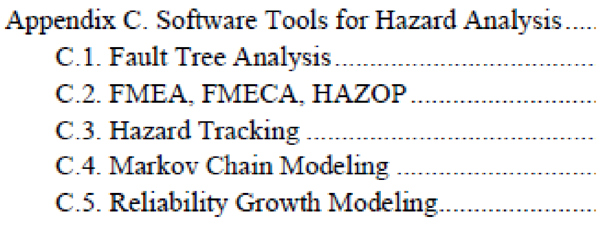
7. Prepare an application system requirements specification.

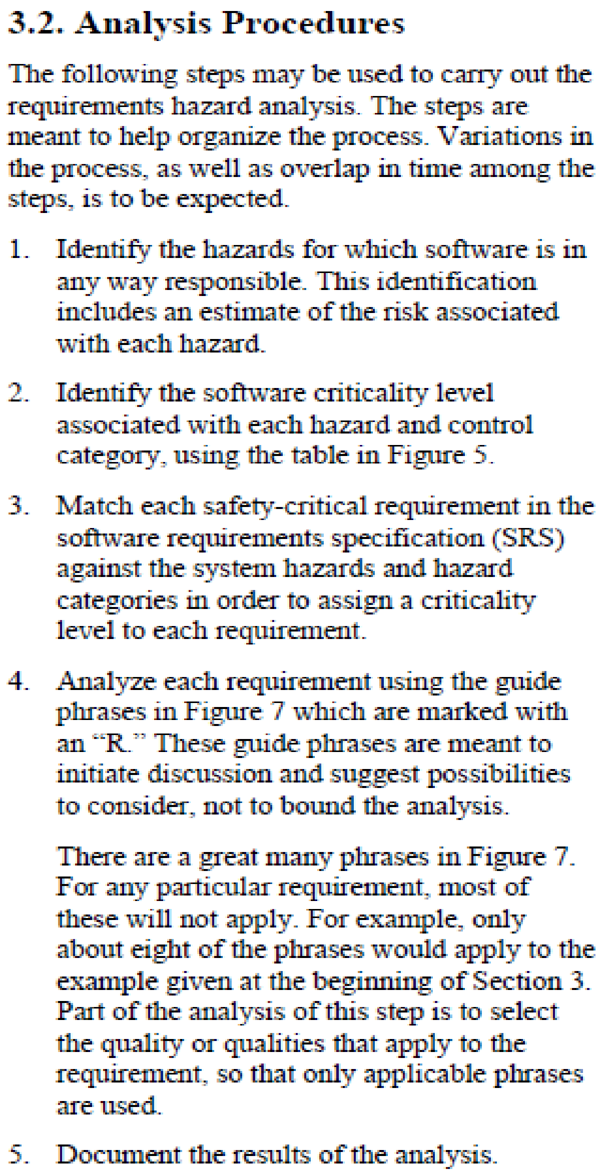
8. Create and document a system design, which shows the allocation of safety functions to software components and other system components and shows how the software component and the remaining application system components will coordinate to address the hazards discovered in previous analyses.

9. Prepare the remaining documents to the extent required in order to specify, design, implement, verify and analyze the software component of the RPS. This includes analysis of additional hazards introduced by choice of specific digital hardware, computer language, compiler, software architecture, software design techniques, and design rules. This analysis will be revisited as digital system design and software design are elaborated.

NUREG/CR-6430: Hazard Analysis of Requirements

- Hazard analysis of software requirements specification
  - It consists of 5 steps
    - Identifying the hazards for software responsible
    - Identifying the critical level
    - Matching each safety-critical requirements in the SRS
    - Analyzing each requirements using the guide phrases
    - Document the results
- Analyzing methods with guide phrases
  - Methods are not fixed




3.2. Analysis Procedures

The following steps may be used to carry out the requirements hazard analysis. The steps are meant to help organize the process. Variations in the process, as well as overlap in time among the steps, is to be expected.

1. Identify the hazards for which software is in any way responsible. This identification includes an estimate of the risk associated with each hazard.
2. Identify the software criticality level associated with each hazard and control category, using the table in Figure 5.
3. Match each safety-critical requirement in the software requirements specification (SRS) against the system hazards and hazard categories in order to assign a criticality level to each requirement.
4. Analyze each requirement using the guide phrases in Figure 7 which are marked with an "R." These guide phrases are meant to initiate discussion and suggest possibilities to consider, not to bound the analysis.

   There are a great many phrases in Figure 7. For any particular requirement, most of these will not apply. For example, only about eight of the phrases would apply to the example given at the beginning of Section 3. Part of the analysis of this step is to select the quality or qualities that apply to the requirement, so that only applicable phrases are used.
5. Document the results of the analysis.

NUREG/CR-6430: Guide Phrases

- NUREG/CR-6430 provides guide phrases for applying analysis
  - It is able to support analyzing the hazard analysis of SW
  - Guide phrases consists of 'quality,' 'aspect,' 'phase' and 'guide phrases'

Quality	Aspect	Phase	Guide Phrases
Accuracy	Sensor	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
		RADC	Below minimum range
		RADC	Above maximum range
		RADC	Within range, but wrong
		RADC	Physical units are incorrect
		RADC	Wrong data type or data size
	Actuator	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
		RADC	Below minimum range
		RADC	Above maximum range
		RADC	Physical units are incorrect
		RADC	Wrong data type or data size

Aspects of Guide Phrases

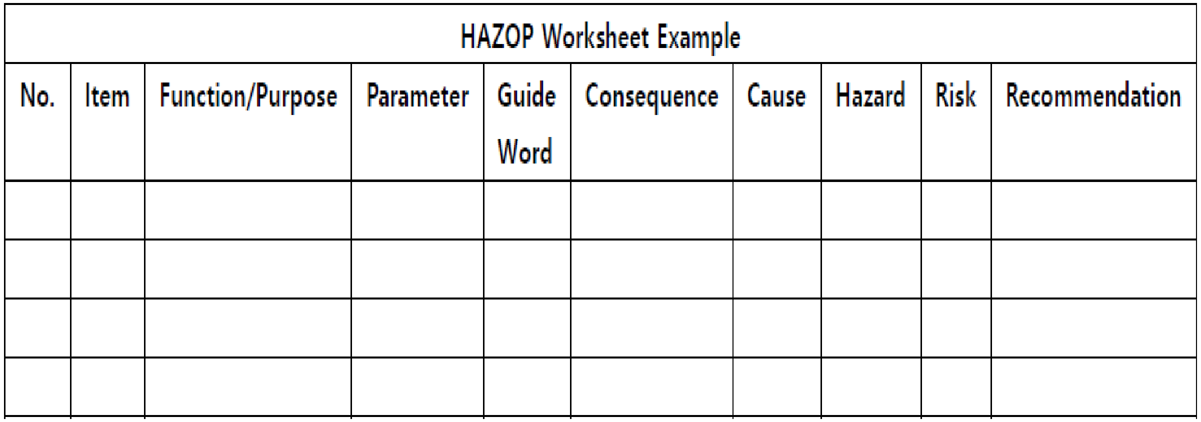
- Sensor
- Actuator
- Operator input/output
- Calculation
- Message
- Timing
- Functionality
- …

DEPENDABLE SOFTWARE LABORATORY

KU KONKUK UNIVERSITY

HAZOP

- HAZOP is used to identify and analyze hazards and operational concerns of a system
  - It utilizes key guide words and system diagrams
  - Generally, HAZOP uses worksheet table to analyze
  - There are several guide words which are used to analyze



HAZOP Worksheet Example

No.	Item	Function/Purpose	Parameter	Guide Word	Consequence	Cause	Hazard	Risk	Recommendation

HAZOP guide words

- No
- Reverse
- Also
- Early
- Late
- Part of
- Before/After
- Inadvertent

Software hazard analysis with two approaches to DFLC-N PM SW req.

SOFTWARE HAZARD ANALYSIS WITH TWO APPROACHES

Hazard Analysis of FPGA SW requirements

- We use software requirements of DFLC-N PM to analyze
  - SW requirement of DFLC-N PM is the prototype version of FPGA-based controllers in NPPs
  - It consists of 16 component and control software

- Hazard analysis of DFLC-N PM is performed with two approaches
  - HAZOP with general worksheet and guide words
  - HAZOP with the process of NUREG/CR-6430 and guide phrases
  - We identify the usability of NUREG guides by through the analysis

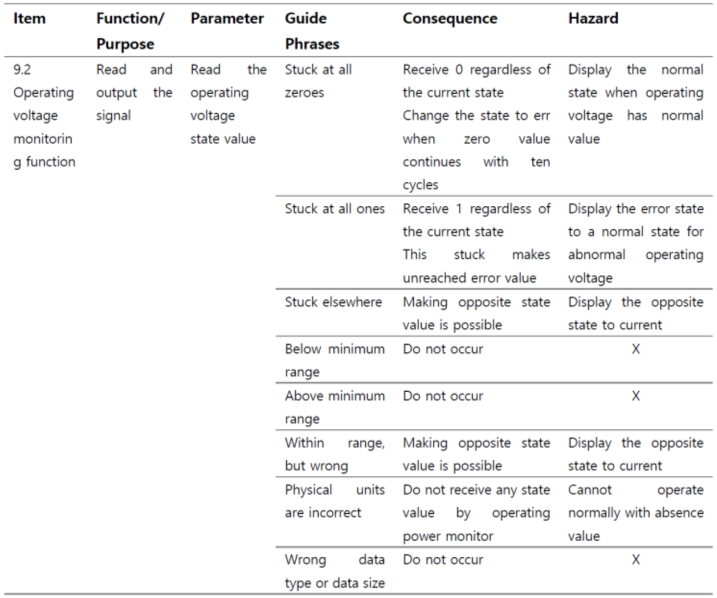
Preliminary Hazard List

- We first identify the preliminary hazard lists of DFLC-N
  - It reflects the characteristics of HW component
  - Consisting of 4 main subjects

No.	Preliminary Hazard List – Process Module
1	**Power supply** a. Loss of operating power b. Over current c. Overvoltage
2	**Physical effects of internal/external** a. Fire occurrence b. Physical impact c. Radioactivity
3	**Operation error** a. Operation error of application b. Memory error/failure c. Response time error(timing error, scan time) d. Error diagnosis function failure e. Lack of transmit capacity f. LED failure g. Disability of network
4	**Operation failure** a. Operation failure by operator (bypass)

Software Hazard Analysis with NUREG/CR-6430 Guides

- We apply analysis methods of requirements analysis in NUREG/CR-6430 process and guide phrases
  - HAZOP is used to apply guide phrases and analyze
  - Guide phrases are chosen to reflect the characteristics of FPGA
    - Several guide phrases are not used to analyze
  - Perform analyzing relations between PHL and hazards
    - Because, it is able to analyze the effects of higher level of design or design process in software life cycle

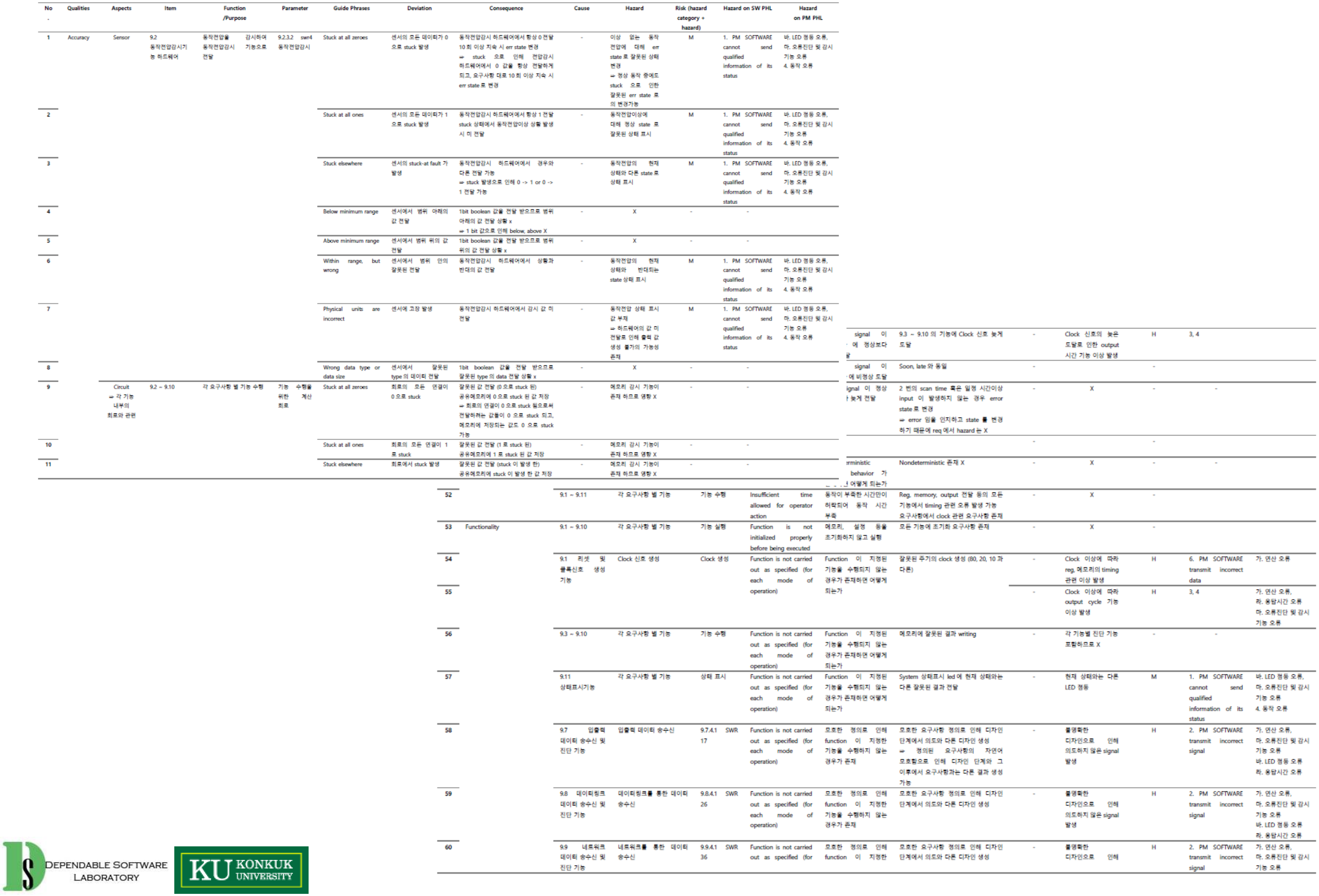


Item	Function/ Purpose	Parameter	Guide Phrases	Consequence	Hazard
9.2 Operating voltage monitoring function	Read and output the signal	Read the operating voltage state value	Stuck at all zeroes	Receive 0 regardless of the current state Change the state to err when zero value continues with ten cycles	Display the normal state when operating voltage has normal value
			Stuck at all ones	Receive 1 regardless of the current state This stuck makes unreached error value	Display the error state to a normal state for abnormal operating voltage
			Stuck elsewhere	Making opposite state value is possible	Display the opposite state to current
			Below minimum range	Do not occur	X
			Above minimum range	Do not occur	X
			Within range, but wrong	Making opposite state value is possible	Display the opposite state to current
			Physical units are incorrect	Do not receive any state value by operating power monitor	Cannot operate normally with absence value
			Wrong data type or data size	Do not occur	X

Dependable Software Laboratory

KU KONKUK UNIVERSITY

Software Hazard Analysis with NUREG/CR-6430 Guides - 2



No.	Qualities	Aspects	Item	Function /Purpose	Parameter	Guide Phrases	Deviation	Consequence	Cause	Hazard	Risk (hazard category + hazard)	Hazard on SW PHL	Hazard on PM PHL
1	Accuracy	Sensor	9.2 동작전압감시기능 하드웨어	동작전압을 감시하여 동작전압감시 기능으로 전달	9.2.3.2 swr4 동작전압감시	Stuck at all zeroes	센서의 모든 데이터가 0 으로 stuck 발생	동작전압감시 하드웨어에서 항상 0 전달 10 회 이상 지속 시 err state 변경 ⇒ stuck 으로 인해 전압감시 하드웨어에서 0 값을 항상 전달하게 되고, 요구사항 대로 10 회 이상 지속 시 err state 로 변경	-	이상 없는 동작 전압에 대해 err state 로 잘못된 상태 변경 ⇒ 정상 동작 중에도 stuck 으로 인한 잘못된 err state 로의 변경가능	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
2						Stuck at all ones	센서의 모든 데이터가 1 으로 stuck 발생	동작전압감시 하드웨어에서 항상 1 전달 stuck 상태에서 동작전압이상 상황 발생 시 미 전달	-	동작전압이상에 대해 정상 state 로 잘못된 상태 표시	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
3						Stuck elsewhere	센서의 stuck-at fault 가 발생	동작전압감시 하드웨어에서 경우와 다른 전달 가능 ⇒ stuck 발생으로 인해 0 -> 1 or 0 -> 1 전달 가능	-	동작전압의 현재 상태와 다른 state 로 상태 표시	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
4						Below minimum range	센서에서 범위 아래의 값 전달	1bit boolean 값을 전달 받으므로 범위 아래의 값 전달 상황 x ⇒ 1 bit 값으로 인해 below, above X	-	X	-	-	
5						Above minimum range	센서에서 범위 위의 값 전달	1bit boolean 값을 전달 받으므로 범위 위의 값 전달 상황 x	-	X	-	-	
6						Within range, but wrong	센서에서 범위 안의 잘못된 전달	동작전압감시 하드웨어에서 상황과 반대의 값 전달	-	동작전압의 현재 상태와 반대되는 state 상태 표시	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
7						Physical units are incorrect	센서에 고장 발생	동작전압감시 하드웨어에서 감시 값 미 전달	-	동작전압 상태 표시 값 부재 ⇒ 하드웨어의 값 미 전달로 인해 출력 값 생성 불가의 가능성 존재	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
8						Wrong data type or data size	센서에서 잘못된 type 의 데이터 전달	1bit boolean 값을 전달 받으므로 잘못된 type 의 data 전달 상황 x	-	X	-	-	
9		Circuit ⇒ 각 기능 내부의 회로와 관련	9.2 ~ 9.10	각 요구사항 별 기능 수행	기능 수행을 위한 계산 회로	Stuck at all zeroes	회로의 모든 연결이 0 으로 stuck	잘못된 값 전달 (0 으로 stuck 된) 공유메모리에 0 으로 stuck 된 값 저장 ⇒ 회로의 연결이 0 으로 stuck 됨으로써 전달하려는 값들이 0 으로 stuck 되고, 메모리에 저장되는 값도 0 으로 stuck 가능	-	메모리 감시 기능이 존재 하므로 영향 X	-	-	
10						Stuck at all ones	회로의 모든 연결이 1 로 stuck	잘못된 값 전달 (1 로 stuck 된) 공유메모리에 1 로 stuck 된 값 저장	-	메모리 감시 기능이 존재 하므로 영향 X	-	-	
11						Stuck elsewhere	회로에서 stuck 발생	잘못된 값 전달 (stuck 이 발생 한) 공유메모리에 stuck 이 발생 한 값 저장	-	메모리 감시 기능이 존재 하므로 영향 X	-	-	

No.	Qualities	Aspects	Item	Function /Purpose	Parameter	Guide Phrases	Deviation	Consequence	Cause	Hazard	Risk (hazard category + hazard)	Hazard on SW PHL	Hazard on PM PHL
							signal 이 … 에 정상보다 … 달	9.3 ~ 9.10 의 기능에 Clock 신호 늦게 도달	-	Clock 신호의 늦은 도달로 인한 output 시간 기능 이상 발생	H	3, 4	
							signal 이 … 에 비정상 도달	Soon, late 와 동일	-		-		
							ignal 이 정상 … 늦게 전달	2 번의 scan time 혹은 일정 시간이상 input 이 발생하지 않는 경우 error state 로 변경 ⇒ error 임을 인지하고 state 를 변경하기 때문에 req 에서 hazard 는 X	-	X	-	-	
									-		-		
							erministic … behavior 가 … 어떻게 되는가	Nondeterministic 존재 X	-	X	-	-	
52			9.1 ~ 9.11	각 요구사항 별 기능	기능 수행	Insufficient time allowed for operator action	동작이 부족한 시간만이 허락되어 동작 시간 부족	Reg, memory, output 전달 등의 모든 기능에서 timing 관련 오류 발생 가능 요구사항에서 clock 관련 요구사항 존재	-	X	-		
53	Functionality		9.1 ~ 9.10	각 요구사항 별 기능	기능 실행	Function is not initialized properly before being executed	메모리, 설정 등을 초기화하지 않고 실행	모든 기능에 초기화 요구사항 존재	-	X	-		
54			9.1 리셋 및 클록신호 생성 기능	Clock 신호 생성	Clock 생성	Function is not carried out as specified (for each mode of operation)	Function 이 지정된 기능을 수행되지 않는 경우가 존재하면 어떻게 되는가	잘못된 주기의 clock 생성 (80, 20, 10 과 다른)	-	Clock 이상에 따라 reg, 메모리의 timing 관련 이상 발생	H	6. PM SOFTWARE transmit incorrect data	가. 연산 오류
55									-	Clock 이상에 따라 output cycle 기능 이상 발생	H	3, 4	가. 연산 오류, 라. 응답시간 오류 마. 오류진단 및 감시 기능 오류
56			9.3 ~ 9.10	각 요구사항 별 기능	기능 수행	Function is not carried out as specified (for each mode of operation)	Function 이 지정된 기능을 수행되지 않는 경우가 존재하면 어떻게 되는가	메모리에 잘못된 결과 writing	-	각 기능별 진단 기능 포함하므로 X	-	-	
57			9.11 상태표시기능	각 요구사항 별 기능	상태 표시	Function is not carried out as specified (for each mode of operation)	Function 이 지정된 기능을 수행되지 않는 경우가 존재하면 어떻게 되는가	System 상태표시 led 에 현재 상태와는 다른 잘못된 결과 전달	-	현재 상태와는 다른 LED 점등	M	1. PM SOFTWARE cannot send qualified information of its status	바. LED 점등 오류, 마. 오류진단 및 감시 기능 오류 4. 동작 오류
58			9.7 입출력 데이터 송수신 및 진단 기능	입출력 데이터 송수신	9.7.4.1 SWR 17	Function is not carried out as specified (for each mode of operation)	모호한 정의로 인해 function 이 지정한 기능을 수행하지 않는 경우가 존재	모호한 요구사항 정의로 인해 디자인 단계에서 의도와 다른 디자인 생성 ⇒ 정의된 요구사항의 자연어 모호함으로 인해 디자인 단계와 그 이후에서 요구사항과는 다른 결과 생성 가능	-	불명확한 디자인으로 인해 의도하지 않은 signal 발생	H	2. PM SOFTWARE transmit incorrect signal	가. 연산 오류, 마. 오류진단 및 감시 기능 오류 바. LED 점등 오류 라. 응답시간 오류
59			9.8 데이터링크 데이터 송수신 및 진단 기능	데이터링크를 통한 데이터 송수신	9.8.4.1 SWR 26	Function is not carried out as specified (for each mode of operation)	모호한 정의로 인해 function 이 지정한 기능을 수행하지 않는 경우가 존재	모호한 요구사항 정의로 인해 디자인 단계에서 의도와 다른 디자인 생성	-	불명확한 디자인으로 인해 의도하지 않은 signal 발생	H	2. PM SOFTWARE transmit incorrect signal	가. 연산 오류, 마. 오류진단 및 감시 기능 오류 바. LED 점등 오류 라. 응답시간 오류
60			9.9 네트워크 데이터 송수신 및 진단 기능	네트워크를 통한 데이터 송수신	9.9.4.1 SWR 36	Function is not carried out as specified (for	모호한 정의로 인해 function 이 지정한	모호한 요구사항 정의로 인해 디자인 단계에서 의도와 다른 디자인 생성	-	불명확한 디자인으로 인해	H	2. PM SOFTWARE transmit incorrect signal	가. 연산 오류, 마. 오류진단 및 감시 기능 오류

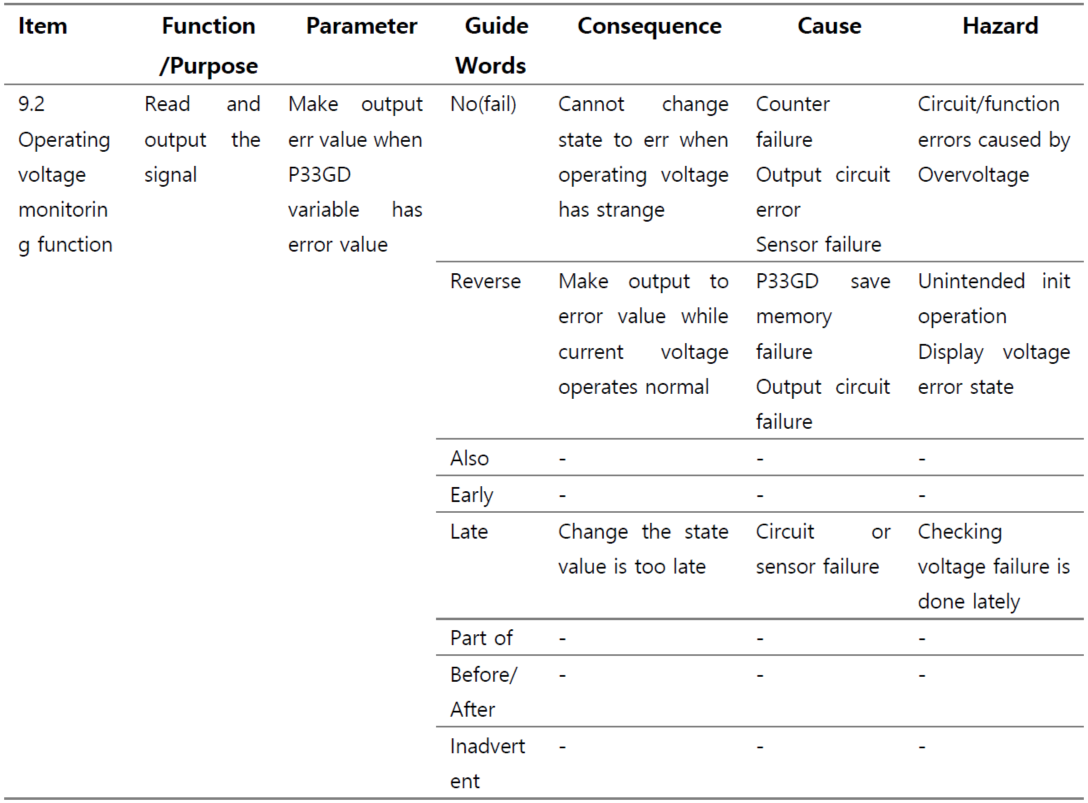
DEPENDABLE SOFTWARE LABORATORY

KU KONKUK UNIVERSITY

Software Hazard Analysis with HAZOP

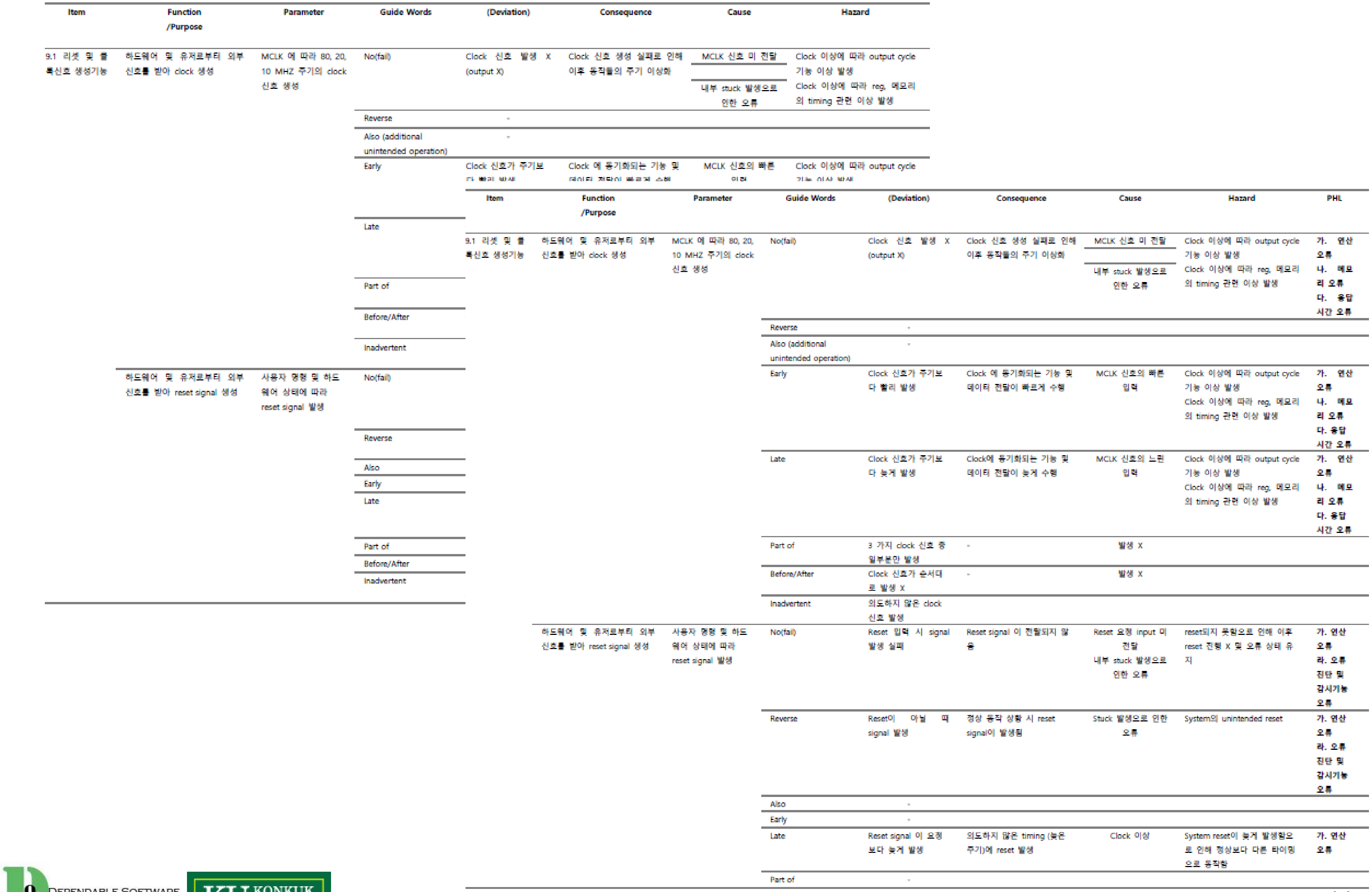
- Parts of the results about hazard analysis with HAZOP and guide words with generally used

- All of items and function in requirements are analyzed(matching) with guide words



Item	Function /Purpose	Parameter	Guide Words	Consequence	Cause	Hazard
9.2 Operating voltage monitoring function	Read and output the signal	Make output err value when P33GD variable has error value	No(fail)	Cannot change state to err when operating voltage has strange	Counter failure Output circuit error Sensor failure	Circuit/function errors caused by Overvoltage
			Reverse	Make output to error value while current voltage operates normal	P33GD save memory failure Output circuit failure	Unintended init operation Display voltage error state
			Also	-	-	-
			Early	-	-	-
			Late	Change the state value is too late	Circuit or sensor failure	Checking voltage failure is done lately
			Part of	-	-	-
			Before/ After	-	-	-
			Inadvertent	-	-	-

Software Hazard Analysis with HAZOP - 2



Item	Function /Purpose	Parameter	Guide Words	(Deviation)	Consequence	Cause	Hazard	PHL
9.1 리셋 및 클록신호 생성기능	하드웨어 및 유저로부터 외부 신호를 받아 clock 생성	MCLK 에 따라 80, 20, 10 MHZ 주기의 clock 신호 생성	No(fail)	Clock 신호 발생 X (output X)	Clock 신호 생성 실패로 인해 이후 동작들의 주기 이상화	MCLK 신호 미 전달 내부 stuck 발생으로 인한 오류	Clock 이상에 따라 output cycle 기능 이상 발생 Clock 이상에 따라 reg, 메모리의 timing 관련 이상 발생	가. 연산 오류 나. 메모리 오류 다. 응답 시간 오류
			Reverse	-				
			Also (additional unintended operation)	-				
			Early	Clock 신호가 주기보다 빨리 발생	Clock 에 동기화되는 기능 및 데이터 전달이 빠르게 수행	MCLK 신호의 빠른 입력	Clock 이상에 따라 output cycle 기능 이상 발생 Clock 이상에 따라 reg, 메모리의 timing 관련 이상 발생	가. 연산 오류 나. 메모리 오류 다. 응답 시간 오류
			Late	Clock 신호가 주기보다 늦게 발생	Clock에 동기화되는 기능 및 데이터 전달이 늦게 수행	MCLK 신호의 느린 입력	Clock 이상에 따라 output cycle 기능 이상 발생 Clock 이상에 따라 reg, 메모리의 timing 관련 이상 발생	가. 연산 오류 나. 메모리 오류 다. 응답 시간 오류
			Part of	3 가지 clock 신호 중 일부분만 발생	-	발생 X		
			Before/After	Clock 신호가 순서대로 발생 X	-	발생 X		
			Inadvertent	의도하지 않은 clock 신호 발생				
	하드웨어 및 유저로부터 외부 신호를 받아 reset signal 생성	사용자 명령 및 하드웨어 상태에 따라 reset signal 발생	No(fail)	Reset 입력 시 signal 발생 실패	Reset signal 이 전달되지 않음	Reset 요청 input 미 전달 내부 stuck 발생으로 인한 오류	reset되지 못함으로 인해 이후 reset 진행 X 및 오류 상태 유지	가. 연산 오류 라. 오류 진단 및 감시기능 오류
			Reverse	Reset이 아닐 때 signal 발생	정상 동작 상황 시 reset signal이 발생됨	Stuck 발생으로 인한 오류	System의 unintended reset	가. 연산 오류 라. 오류 진단 및 감시기능 오류
			Also	-				
			Early	-				
			Late	Reset signal 이 요청보다 늦게 발생	의도하지 않은 timing (늦은 주기)에 reset 발생	Clock 이상	System reset이 늦게 발생함으로 인해 정상보다 다른 타이밍으로 동작함	가. 연산 오류
			Part of	-				

Item	Function /Purpose	Parameter	Guide Words	(Deviation)	Consequence	Cause	Hazard
9.1 리셋 및 클록신호 생성기능	하드웨어 및 유저로부터 외부 신호를 받아 clock 생성	MCLK 에 따라 80, 20, 10 MHZ 주기의 clock 신호 생성	No(fail)	Clock 신호 발생 X (output X)	Clock 신호 생성 실패로 인해 이후 동작들의 주기 이상화	MCLK 신호 미 전달 내부 stuck 발생으로 인한 오류	Clock 이상에 따라 output cycle 기능 이상 발생 Clock 이상에 따라 reg, 메모리의 timing 관련 이상 발생
			Reverse	-			
			Also (additional unintended operation)	-			
			Early	Clock 신호가 주기보	Clock 에 동기화되는 기능 및	MCLK 신호의 빠른	Clock 이상에 따라 output cycle
			Late				
			Part of				
			Before/After				
			Inadvertent				
	하드웨어 및 유저로부터 외부 신호를 받아 reset signal 생성	사용자 명령 및 하드웨어 상태에 따라 reset signal 발생	No(fail)				
			Reverse				
			Also				
			Early				
			Late				
			Part of				
			Before/After				
			Inadvertent				

DEPENDABLE SOFTWARE LABORATORY

KU KONKUK UNIVERSITY

Discussion of the results

DISCUSSION OF THE RESULTS

Discussions of the Results and Process with comparison

- Difference points of the analysis results about two approaches

- Guide phrases and perspective makes the differences
- We perform comparing analysis about the results with two approaches

- Differences appears in the analyzing aspects of requirement elements and analysis results
  - Results(related PHL) aspects
  - Analysis aspects of each elements in requirements
  - Especially, differences about applying methods are presented about guide phrases

- Usability of NUREG/CR-6430 about applying FPGA SW is also checked

Differences of Analysis aspects

- Differences of analysis aspects
  - Analysis aspects of requirements points is different with each approaches
  - Comparing results are appeared ‘cause’ or ‘analysis of deviation’

Requirements Point	Analysis Aspects	
	NUREG/CR-6430	HAZOP (GW)
Sensor	Analysis of deviation	Cause
Input/output	Analysis of deviation	Cause
Timing	Analysis of deviation	Cause Analysis of deviation
Function	Analysis of deviation	Analysis of deviation
Circuit	Analysis of deviation	Cause
Security	Analysis of deviation	-
Memory	Cause	Cause Analysis of deviation
Data bus	(Analysis of deviation)	Cause Analysis of deviation
Network	(Analysis of deviation)	Cause Analysis of deviation

Differences of PHL aspects

- **Differences of PHL aspects**
  - **We compare connected PHL in the analysis results**
  - **Potential hazards which are analyzed in SW requirement have some different list**

PHL	**NUREG/CR-6430**	**HAZOP (General GW)**
Operation error		
a. Operation error of application	O	O
b. Memory error/failure	N/A	O
c. Response time error	O	O
d. Error diagnosis function failure	O	O
e. Lack of transmit capacity	N/A	N/A
f. LED failure	O	O
g. Disability of network	N/A	N/A
Operation failure		
a. Operation failure by operator (bypass)	O	O

Discussions of the Results and Process with comparison

- Two approaches has different point of view to analyze about each elements of software requirements spec.
  - It is appeared by cause and analysis of deviation
- Differences in comparison of PHL do not means usefulness directly
  - We think it caused by extension of difference about analysis aspects
    - Guide phrases about memory is not contained in NUREG/CR-6430

PHL	NUREG/CR-6430	HAZOP (General GW)
Operation error		
a. Operation error of application	O	O
b. Memory error/failure	N/A	O
c. Response time error	O	O
d. Error diagnosis function failure	O	O
e. Lack of transmit capacity	N/A	N/A
f. LED failure	O	O
g. Disability of network	N/A	N/A
Operation failure		
a. Operation failure by operator (bypass)	O	O

Discussions of the Results and Process with comparison

- **Two approaches has different point of view to analyze about each elements of software requirements spec.**
  - **It is appeared by cause and analysis of deviation**
- **Differences in comparison of PHL do not means usefulness directly**
  - **We think**
    aspects
    - **Guid**

PHL
a. Operation error of applica
b. Memory error/failure
c. Response time error
d. Error diagnosis function f
e. Lack of transmit capacity
f. LED failure
g. Disability of network
a. Operation failure by oper

Requirements Point	Analysis Aspects	
	NUREG/CR-6430	HAZOP (GW)
Sensor	Analysis of deviation	Cause
Input/output	Analysis of deviation	Cause
Timing	Analysis of deviation	Cause Analysis of deviation
Function	Analysis of deviation	Analysis of deviation
Circuit	Analysis of deviation	Cause
Security	Analysis of deviation	-
Memory	Cause	Cause Analysis of deviation
Data bus	(Analysis of deviation)	Cause Analysis of deviation
Network	(Analysis of deviation)	Cause Analysis of deviation

Discussions of the Results and Process with comparison

- Additionally, NUREG/CR-6430 provides guide phrases about security, safety and so on

  - These guide phrases make possible to identify whether requirement spec considers about these contents
    - It also can help to analyze non-functional view accordance with these guide phrases

  - Providing guide phrases also makes easy to apply
    - Because, identifying deviation of guide phrases about req. elements is simple

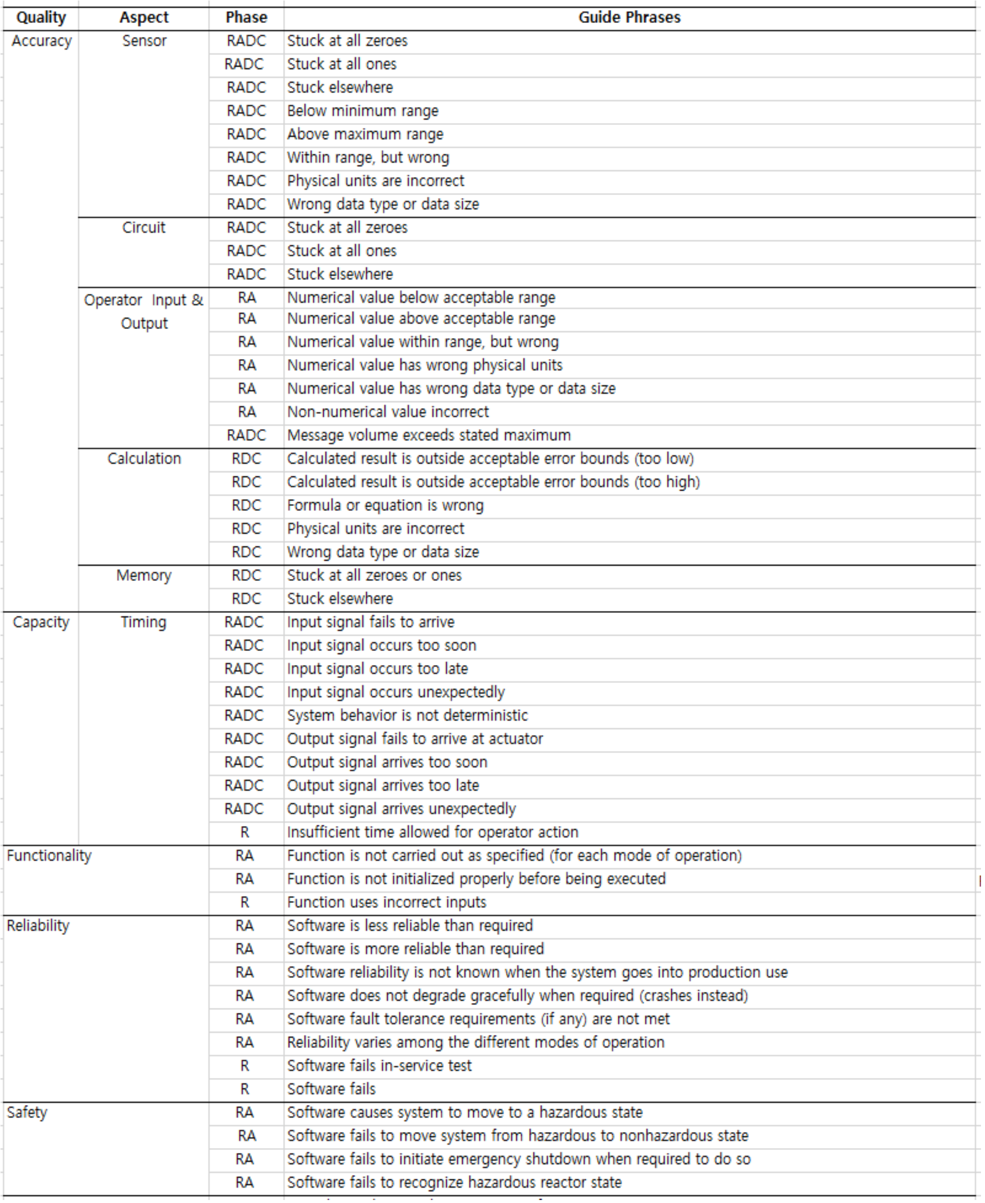
Conclusion

- We perform software hazard analysis of FPGA SW requirement
  - Using two approaches
    - HAZOP
    - NUREG/CR-6430 guides
- We also perform comparing analysis with these approaches
  - Perspective of PHL and analysis aspects
  - Identifying the usability of NUREG/CR-6430 guides for hazard analysis of FPGA SW requirements specification
    - Some insufficiency points also exists
- We are now planning to supplement the guide phrases to apply efficiently

Q&A

THANK YOU

Guide Phrases



Quality	Aspect	Phase	Guide Phrases
Accuracy	Sensor	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
		RADC	Below minimum range
		RADC	Above maximum range
		RADC	Within range, but wrong
		RADC	Physical units are incorrect
		RADC	Wrong data type or data size
	Circuit	RADC	Stuck at all zeroes
		RADC	Stuck at all ones
		RADC	Stuck elsewhere
	Operator Input & Output	RA	Numerical value below acceptable range
		RA	Numerical value above acceptable range
		RA	Numerical value within range, but wrong
		RA	Numerical value has wrong physical units
		RA	Numerical value has wrong data type or data size
		RA	Non-numerical value incorrect
		RADC	Message volume exceeds stated maximum
	Calculation	RDC	Calculated result is outside acceptable error bounds (too low)
		RDC	Calculated result is outside acceptable error bounds (too high)
		RDC	Formula or equation is wrong
		RDC	Physical units are incorrect
		RDC	Wrong data type or data size
	Memory	RDC	Stuck at all zeroes or ones
		RDC	Stuck elsewhere
Capacity	Timing	RADC	Input signal fails to arrive
		RADC	Input signal occurs too soon
		RADC	Input signal occurs too late
		RADC	Input signal occurs unexpectedly
		RADC	System behavior is not deterministic
		RADC	Output signal fails to arrive at actuator
		RADC	Output signal arrives too soon
		RADC	Output signal arrives too late
		RADC	Output signal arrives unexpectedly
		R	Insufficient time allowed for operator action
Functionality		RA	Function is not carried out as specified (for each mode of operation)
		RA	Function is not initialized properly before being executed
		R	Function uses incorrect inputs
Reliability		RA	Software is less reliable than required
		RA	Software is more reliable than required
		RA	Software reliability is not known when the system goes into production use
		RA	Software does not degrade gracefully when required (crashes instead)
		RA	Software fault tolerance requirements (if any) are not met
		RA	Reliability varies among the different modes of operation
		R	Software fails in-service test
		R	Software fails
Safety		RA	Software causes system to move to a hazardous state
		RA	Software fails to move system from hazardous to nonhazardous state
		RA	Software fails to initiate emergency shutdown when required to do so
		RA	Software fails to recognize hazardous reactor state