

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

DESIGN OF FAULT TOLERANT
FPGA SYSTEM-ON-MODULE
FOR MISSION CRITICAL RAILWAY APPLICATIONS

OĞUZHAN UĞUZ

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF ELECTRONICS ENGINEERING

ADVISOR: ASSIST. PROF. DR. İHSAN ÇİÇEK

JULY 2025

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

DESIGN OF FAULT TOLERANT
FPGA SYSTEM-ON-MODULE
FOR MISSION CRITICAL RAILWAY APPLICATIONS

OĞUZHAN UĞUZ

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF ELECTRONICS ENGINEERING

ADVISOR: ASSIST. PROF. DR. İHSAN ÇİÇEK

JULY 2025

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GÖREV KRİTİK DEMİRYOLU UYGULAMALARI İÇİN
HATA TOLERANSLI FPGA
MODÜL ÜZERİ SİSTEM TASARIMI

OĞUZHAN UĞUZ

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DR. ÖĞR. ÜYESİ İHSAN ÇİÇEK

TEMMUZ 2025



MASTER of SCIENCE JURY APPROVAL FORM

A thesis submitted by Oğuzhan Uğuz defended on 14/07/2025 before the jury formed with the 30/06/2025 date and 2025/34 numbered decision of the GTU Graduate Administration Board, has been accepted as a MASTER of SCIENCE thesis in the Department of Electronics Engineering.

JURY

MEMBER

(THESIS ADVISOR) : Assist. Prof. Dr. İhsan ÇİÇEK

MEMBER

: Prof. Dr. Hüseyin Arda ÜLKÜ

MEMBER

: Assoc. Prof. Dr. Engin AFACAN

APPROVAL

Gebze Technical University Graduate Administration Board
...../...../..... date and/..... numbered decision.

SIGNATURE/SEAL



Dedicated to my wife

ABSTRACT

The railway industry is undergoing a digital transformation, which results in more safety-critical functions being carried out by engineered systems. Safety-related system development, as standardized in generic and industry-specific standards such as IEC 61508, EN 50126, and ISO 26262, presents unique challenges, requirements, and opportunities. The systems engineering approach to holistic systems development processes is key to identifying, analyzing, and mitigating hazards related to systems of interest.

In this thesis, the basics of systems engineering approach to systems development, how systems engineering methods and practices can be applied to safety-related systems development, and finally, the development process of a generic FPGA system-on-module for use in safety-related specific applications are presented. The overall system development life cycle concepts and the life cycle model used in the development of the FPGA system-on-module are presented. System environment, boundaries, behavior, missions, and functions are identified, forming the basis of the operational concepts and the system definition. Risk analysis tools, such as Failure Modes and Effects Analysis and Fault Tree Analysis are presented, followed by the analysis of risks related to system operation based on the developed system definition.

Requirement engineering basics, including the role of requirements in systems development and requirement writing guidelines are introduced, and following that, the system requirements and functional safety requirements are developed. The functional and physical architectures of the system are developed employing trade studies. System design is developed and presented with the help of block definition diagrams and sequence diagrams. Failure Modes, Effects and Diagnostic Analysis (FMEDA) study is done in the as-designed system to assess system risks. The results show that the as-designed system meets the technical requirements for failure rates and diagnostic capabilities for certification up to SIL 3. The actual certification requires further work to develop the as-implemented system compliant with the engineering management requirements of functional safety equipment development standards.

Keywords: FPGA, Functional Safety, System Development Life Cycle, RAMS, Railway

ÖZET

Demiryolu endüstrisi dijital bir dönüşüm sürecinden geçmektedir ve bu durum, emniyet kritik işlevlerin mühendislik sistemleri tarafından yerine getirilmesini giderek daha yaygın hale getirmektedir. IEC 61508, EN 50126 ve ISO 26262 gibi jenerik ve sektöre özel standartlarla standartlaştırılmış olan emniyetle ilgili sistem geliştirme süreci, kendine özgü zorluklar, gereksinimler ve fırsatlar barındırmaktadır. Sistem mühendisliği yaklaşımı, sistemlerin bütünsel geliştirme süreçlerine dair tehlikelerin tanımlanması, analiz edilmesi ve azaltılması açısından kritik bir rol oynamaktadır.

Bu tezde, sistem geliştirmeye yönelik sistem mühendisliği yaklaşımının temelleri, bu yaklaşımın emniyetle ilgili sistem geliştirmeye nasıl uygulanabileceği ve son olarak, emniyetle ilgili özel uygulamalarda kullanılmak üzere geliştirilen genel amaçlı bir FPGA sistem-modülünün geliştirme süreci sunulmaktadır. Geliştirilen FPGA sistem-modülünün yaşam döngüsü modeli ve genel sistem geliştirme yaşam döngüsü kavramları ele alınmaktadır. Sistem çevresi, sınırları, davranışı, görevleri ve işlevleri tanımlanmakta; bu unsurlar işletim konsepti ve sistem tanımının temelini oluşturmaktadır. Arıza Türleri ve Etkileri Analizi (FMEA) ve Hata Ağacı Analizi (FTA) gibi risk analiz araçları tanıtılmakta, ardından geliştirilen sistem tanımına dayanarak sistem işletimine ilişkin risklerin analizi yapılmaktadır.

Gereksinim mühendisliğinin temelleri, gereksinimlerin sistem geliştirme sürecindeki rolü ve gereksinim yazma yönergeleriyle birlikte sunulmakta, ardından sistem gereksinimleri ile işlevsel güvenlik gereksinimleri geliştirilmektedir. Sistemin işlevsel ve fiziksel mimarileri, tasarım alternatiflerinin değerlendirilmesiyle (trade study) geliştirilmiştir. Sistem tasarımı, blok tanım diyagramları ve sıralama diyagramları yardımıyla geliştirilmiş ve sunulmuştur. Tasarlanan sistem üzerinde Arıza Türleri, Etkileri ve Tanılama Analizi (FMEDA) çalışması yapılmış ve sistem riskleri değerlendirilmiştir. Sonuçlar, tasarlanan sistemin arıza oranları ve tanılama yetenekleri açısından SIL 3 seviyesine kadar olan sertifikasyon için teknik gereksinimleri karşıladığını göstermektedir. Ancak, gerçek sertifikasyon süreci için, sistemin uygulanmış haliyle işlevsel güvenlik ekipmanı geliştirme standartlarının mühendislik yönetimi gerekliliklerine uygun olarak daha fazla çalışmanın yapılması gerekmektedir.

Anahtar Kelimeler: FPGA, Fonksiyonel Emniyet, Sistem Geliştirme Yaşam Döngüsü, RAMS, Demiryolu

ACKNOWLEDGEMENTS

I would like to thank Dr. İhsan Çiçek for his sincerity, the time he dedicated to me, the push he gave me to achieve the better. I feel very fortunate for having had the opportunity of working with such a cheerful advisor as him.

I would like to thank all my current and former colleagues who supported me in reaching my aspirations.

Finally, I am grateful to my lovely wife who has always been there for me, for her patience and support throughout my master's degree education.



TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	vi
ÖZET	vii
ACKNOWLEDGEMENTS	viii
TABLE OF CONTENTS	ix
LIST OF SYMBOLS AND ABBREVIATIONS	x
LIST OF FIGURES	xi
LIST OF TABLES	xiii
1. INTRODUCTION	1
1.1. Scope, Structure and Contributions of This Work	5
2. DEPENDABILITY OF SYSTEMS	7
2.1. System Quality Attributes	7
2.2. Safety-Instrumented Systems	9
2.3. The System Life Cycle	11
2.3.1. System Life Cycle Models	12
3. DEVELOPMENT METHODOLOGY	15
3.1. Iterative Risk Reduction	15
4. SYSTEM CONCEPT	20
4.1. Business Need	20
4.2. Intended Application and System Context	21
5. SYSTEM DEFINITION	23
5.1. Use Cases	26
5.2. Functional Breakdown Structure	28
5.3. System Interfaces	30
6. RISK MANAGEMENT	32
6.1. Top-Down Risk Management	34
6.2. Bottom-Up Risk Management	39
6.3. Hazard Analysis	39
6.3.1. Fault Tree Analysis	40
6.3.2. Failure Modes and Effects Analysis	43
7. SYSTEM REQUIREMENTS	49
7.1. System Safety Requirements	52
8. SYSTEM ARCHITECTURE	56
8.1. FPGA SoM Measures of Performance	56
8.2. Architecture Space Exploration	57
8.3. Functional Architecture	58
8.4. Physical Architecture	67
9. DESIGN AND IMPLEMENTATION	69
9.1. As-Designed Risk Analysis	75
10. DISCUSSION AND FUTURE WORKS	77
REFERENCES	78
BIOGRAPHY	83
PUBLICATIONS AND PRESENTATIONS FROM THE THESIS	84

LIST OF SYMBOLS AND ABBREVIATIONS

ASIC	: Application-specific integrated-circuit
BOM	: Bill of materials
COTS	: Commercial off the shelf
DC	: Diagnostic coverage
EARS	: Easy Approach to Requirements Engineering
EN	: Enable
FMEA	: Failure Modes and Effects Analysis
FMECA	: Failure Modes, Effects and Criticality Analysis
FMEDA	: Failure Modes, Effects and Diagnostics Analysis
FPGA	: Field-programmable gate array
FTA	: Fault Tree Analysis
HW	: Hardware
I/O	: Input/output
JTAG	: Joint Test Action Group
MCU	: Microcontroller
MoP	: Measure of performance
NVM	: Non-volatile memory
PCB	: Printed circuit board
PG	: Power-good
RAMS	: Reliability, Availability, Maintainability and Safety
SFF	: Safe failure fraction
SIF	: Safety-instrumented function
SIS	: Safety-instrumented system
SMT	: Surface mount technology
SoC	: System-on-chip
SoI	: System-of-interest
SoM	: System-on-module
SRAM	: Static random access memory
SW	: Software
SysML	: Systems Modeling Language
UML	: Unified Modeling Language

LIST OF FIGURES

	<u>Page</u>
Figure 1.1: Diagram showing the accumulation of risks arising from local failure modes.	2
Figure 1.2: Life cycle costs associated with FPGA-based and ASIC-based development for varying production volumes.	4
Figure 2.1: Example structure of a safety-instrumented system-of-systems.	10
Figure 2.2: Overview of functional safety standards.	10
Figure 2.3: Simplified system life cycle (the V-cycle) model omitting verifications and iterations.	13
Figure 2.4: Recurrency of system life cycle models enables application to lower system levels.	14
Figure 3.1: System life cycle with iterative risk reduction.	17
Figure 3.2: Structural decomposition and analysis of the system failure modes	19
Figure 4.1: FPGA system-on-module in its intended context.	22
Figure 4.2: Context diagram depicting the interactions with the system environment.	22
Figure 5.1: Concept map showing the relationship between systems, behaviors and interactions.	24
Figure 5.2: The relations between project success and the system definition.	24
Figure 5.3: Some life cycle phase iteration scenarios, some of which affect the system definition.	25
Figure 5.4: Use case diagram depicting the power regulation interactions with the carrier board.	27
Figure 5.5: Use case diagram depicting the system operating according to application-specific software.	27
Figure 6.1: The interactions and the iterations between the system definition and the risk assessment.	34
Figure 6.2: Safety measure definitions are informed by the system mission definitions.	35
Figure 6.3: The link between component failures and system operational characteristics, visualized in increasing layers of abstraction.	36
Figure 6.4: Hazard analysis can be approached deductively or inductively.	40
Figure 6.5: Key for Fault Tree Diagrams (FTA)	41
Figure 6.6: FTA diagram inspecting the causes of output signal failure event.	42
Figure 6.7: Simplified process flow model for conducting FMEA study on the as-specified system function.	44
Figure 6.8: Simplified process flow model for using FMECA to develop safety requirements.	45
Figure 7.1: Concept map relating system requirements to other system concepts.	50
Figure 7.2: The relation between the system, hazards, functions, and safety requirements.	54
Figure 8.1: Functional architecture prototype block definition diagram.	59
Figure 8.2: Solution candidates in two-dimensional design space.	60
Figure 9.1: Block definition diagram depicting the overall physical architecture of the system after design.	71
Figure 9.2: Block definition diagram depicting the power conversion circuit.	72

Figure 9.3: Block definition diagram depicting the hardware-description language defined structure implemented in the safety FPGA.	72
Figure 9.4: Sequence diagram depicting the power-up sequencing scenario events and interactions.	74
Figure 9.5: Sequence diagram depicting the safe shut-off scenario events and interactions.	74
Figure 9.6: Sequence diagram depicting the safe shut-off scenario events and interactions.	74



LIST OF TABLES

	<u>Page</u>
Table 1.1: Benchmarking table comparing different digital components.	3
Table 2.1: Overview of system qualitative attributes related to the system-of-interest.	8
Table 2.2: Generic system applications and desirable quality attributes.	9
Table 3.1: Core activities performed in system life cycle phases as suggested by EN 50126.	16
Table 4.1: Benchmarking table comparing different FPGA subsystem form factors.	21
Table 5.1: Excerpt from the Functional Breakdown Structure of the system-of-interest	29
Table 5.2: Preliminary system interface specifications for the system-of-interest.	31
Table 6.1: Problems arising from not employing deductive risk management methods.	37
Table 6.2: An excerpt from the FMECA study done on the as-specified system functions.	46
Table 7.1: Example functional system requirements for the system-of-interest.	51
Table 7.2: Example functional safety requirements for the system-of-interest.	55
Table 8.1: Measure of performance definitions for the system-of-interest.	57
Table 8.2: N2 representation of a system architecture consisting of three system elements shown in bold, and the interfaces between them	58
Table 8.3: Trade study dimensions on system functions.	60
Table 8.4: Key and scores for interpreting Table 8.5.	63
Table 8.5: Effects of functional architecture trade dimensions on measures of performance.	63
Table 8.6: The selected functional architecture solution elements.	66
Table 8.7: Assessment overview of physical architecture candidates.	67
Table 9.1: The results of the FMEDA study done on the design level for most complex components.	76

1. INTRODUCTION

The railway industry is experiencing a digital transformation [1], [2], which include wider adoption of automatic control systems, driver assistance systems, semi- or fully-autonomous driving systems, and more. This transformation increases the prevalence of systems carrying out safety-critical functions, which necessitates the development of systems that are more complex than ever, while also assuring that these systems are safer and more reliable than ever [3]. Systems that carry out safety-critical functions are called safety-instrumented systems (SIS), and their malfunction may result in interruption of service, damage to property, injuries, or even deaths. Therefore, the development of safety-instrumented systems has many challenges on top of the non-safety related system development processes. The development processes for safety-instrumented systems require assurances against design phase errors and assurance for the system safety in face of operational phase errors. The requirements get stricter the higher the risks associated with a system are.

The expanding automatic control systems demand the use of modern digital integrated circuits, such as microcontrollers, FPGAs, and memory units, which are complex elements with a multitude of functions and even more failure modes. Each failure mode of each component of each subsystem poses safety risk, and the amount of risk is cumulative going up to the system level, as shown in **Figure 1.1**. Identifying failure modes requires a bottom-up approach, whereas assuring the safe handling of each failure mode requires a top-down approach. IEC 61508 [4] standard defines a V-cycle system development model and provides guidelines on how the functional safety of the system under development can be assured. While IEC 61508 is industry-independent, there are industry-specific adaptations such as ISO 26262 [5] for automotive applications, EN 50129 [6] for railway hardware, EN 50716 [7] (formerly EN 50128) for railway software, and more.

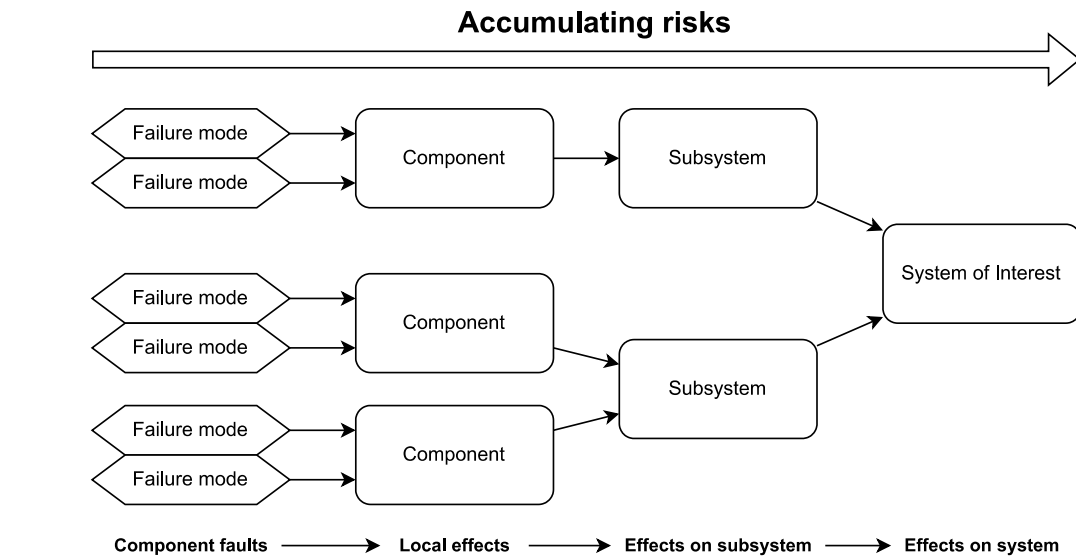


Figure 1.1: Diagram showing the accumulation of risks arising from local failure modes.

Digital subsystems, which are basic building blocks of automatic control systems, may be implemented with one or a combination of available options, such as FPGAs, microcontrollers, application-specific integrated-circuits (ASIC), or discrete digital subcircuits. Each component has context-dependent advantages and limitations. Therefore, there is no constant "right component" as it is a function of the system context.

A benchmarking comparison of digital components is provided in **Table 1.1**. FPGA-based systems pose unique opportunities and challenges for safety-related applications [8]. For starters, FPGAs are massively configurable, even able to implement soft-processors that can emulate microcontrollers [9]. These soft processors occupy more area but provide worse performance when compared to hard processor counterparts. However, this limitation can be offset by the soft processor's customizability. Further, this configurability and reconfigurability enable rapid development iterations, reducing the overall life cycle costs and the time-to-market. These same properties also pose development challenges for safety-related systems, as assuring the correctness of as-designed, as-stored, and as-configured FPGA hardware designs can require dedicated diagnostic capabilities.

Table 1.1: Benchmarking table comparing different digital components.

Property	FPGA	Micro-controller	ASIC	Discrete Digital Subcircuit
Initialization time	$\mathcal{O}(1 \text{ ms})$	$\mathcal{O}(10 \text{ ms})$	$\mathcal{O}(1 \text{ } \mu\text{s})$	$\mathcal{O}(1 \text{ } \mu\text{s})$
Reconfigurability	SW defined	SW defined	Requires semiconductor redevelopment	Requires electronic board redevelopment
Testability and debuggability	Real-time debug with PC	Real-time debug with PC	Limited	Most signals are verifiable with lab. equipment
Functional development complexity	Low	Low	Highest	Medium ³
Safety-integrity development complexity	High	High	Highest	Lowest
Computational capacity	High ²	Medium	Highest ²	Low
Prototyping costs	Low ¹	Low ¹	Highest	Medium ³
Low-volume production costs	Medium	Low	Highest	Low ³
High-volume production costs	Highest	Medium	Low	Medium ³

NOTES:¹ Less prototype iterations are expected due to reconfigurability.² Since the circuit is defined for the specific application, better performance is possible.³ Largely dependent on the implemented functions.

Developing FPGA designs and electronic boards is less resource-intensive than developing ASICs, as ASIC development has extremely high upfront development costs [10]. However, as the production volume increases, the unit cost of ASICs falls dramatically due to scalability. This relation between FPGA and ASIC costs is shown conceptually in **Figure 1.2**. The reconfigurability of FPGAs makes them more favorable over ASICs, as resolving FPGA design errors requires software updates, but resolving ASIC design errors requires repeating the development process almost from the start. After a design matures and proves itself in the application context accompanied by a ramp-up of production quantities, ASICs become favorable in terms of overall life cycle costs.

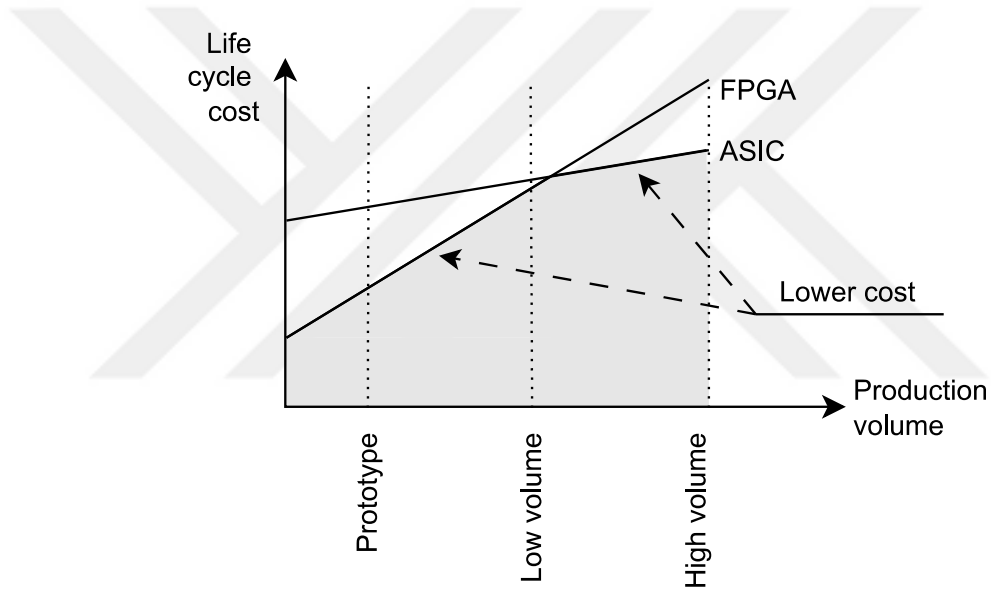


Figure 1.2: Life cycle costs associated with FPGA-based and ASIC-based development for varying production volumes.

Owing to the low development cost and short time to market, FPGAs are increasingly widely employed in safety-related applications [10] despite their challenges. To further reduce the costs related to safety-related FPGA subsystems development and manufacture, a generic FPGA module could be reused in multiple systems, leveraging economies of scale [11] while reducing the repeated safety certification efforts. While there are many commercial off the shelf (COTS) FPGA subsystems, based on a review of vendor offerings as of 2025, no COTS FPGA modules explicitly target functional safety applications. Further, while there is extensive work on FPGA dependability

techniques, little research is available that depict a holistic approach to dependable FPGA system development.

1.1. Scope, Structure and Contributions of This Work

This thesis presents the development process and the resulting design of an FPGA-based embedded-system intended for functional safety applications primarily in railway on-vehicle electronic equipment. The work presented in this thesis is aimed at filling the gap in the literature for a holistic approach in developing dependable FPGA systems. The presented work focuses on the design and implementation half of the V-cycle, therefore, manufacturing and the following phases such as integration, validation, and customer acceptance are out of scope.

The system was developed as part of the efforts towards the development of safety-instrumented systems in TÜBİTAK RUTE. The intent is to modularize the design of the FPGA subsystem to reduce recurrent engineering in multiple subsystems or projects.

Since the system under development is aimed to be used in safety-instrumented systems, qualitative system attributes are central to the development process. To design the system to conform to specific desired system quality attributes, the system life cycle process was adopted. As per [12], railway rolling stock electric/electronic equipment must conform to [13]. This standard also refers to [14], which suggests a V-cycle system life cycle model, which will be discussed in greater detail in the following sections.

The contributions of this thesis are as follows:

- A holistic view into the development processes of functional-safety design life cycle, encompassing the business processes view, the architecture view, and the risk management view is provided.
- The methodology used in the design of the proposed system, which can be used as a guide for future functional-safety capable system development projects is described.
- Safety analyses for FPGA subsystems and diagnostic methods employed to remedy technical risks are presented.

- We provide architecture candidates and evaluation methods for FPGA-based functional-safety capable systems.

The structure of the thesis is as follows: In Section 2, the concepts of system quality attributes, safety-instrumented systems, and the systems engineering approach to system risk management are described. In Section 3, the development methodology that was used in the FPGA module's development is described. The following Sections 4-9 describe the activities and the outcomes of select system life cycle phases. Finally, discussions about the presented work and future works are argued in Section 10.



2. DEPENDABILITY OF SYSTEMS

2.1. System Quality Attributes

A system carries out some functions, and may have numerous measures of performance evaluating quantitative system parameters measuring functional parameters such as its power consumption, maximum speed, weight etc. Aside from these and similar functional measures of performance, systems may have additional qualitative properties that make them superior for certain applications. Therefore, two systems with exactly same functional measures of performance may in fact result in very different outcomes, depending on their qualitative differences. These properties are termed system quality attributes and are also known as system architecture characteristics. An overview of system quality attributes related to the system-of-interest are summarized in **Table 2.1**.

While qualitative attributes do not describe system functions, they still serve stakeholder needs by providing reduction to health risks, property damage risks, and business operational risks. From the descriptions of system qualitative attributes, one can infer that subsets of these attributes can and should be treated as requirements in the development of critical systems. In **Table 2.2** several generic equipment applications ranging from simple to critical are categorized with respect to which system quality attributes are applicable to their respective products.

We see that even systems as simple as TV remotes can benefit from system quality attributes. Also, we observe that critical systems such as an engine control unit can have requirements for a diverse set of system quality attributes.

It should be noted that contrary to the name of the category of system quality attributes, these properties are usually quantifiable.

Table 2.1: Overview of system qualitative attributes related to the system-of-interest.

Attribute	Description	Consequence
Reliability	"Ability to perform as required, without failure, for a given time interval, under given conditions" [15]	Less corrective maintenance Lower operating cost
Availability	"Ability to be in a state to perform as required under given conditions" [15]	Less downtime
Maintainability	"Ability to be retained in, or restored to, a state to perform as required, under given conditions" [15]	Shorter restore times Less downtime
Safety	"Freedom from unacceptable risk" [15]	Less risk to health and property
Supportability	"Ability of an item to be supported with a defined life profile and given resources" [15]	Longer useful life
Dependability	"Ability to perform as and when required" [15]	Less corrective maintenance Lower operating cost Less downtime Less risk to health and property
Resilience	"Ability of an item to return to functional capability, within an acceptable time period, after having been affected by inputs or conditions when subjected to circumstances that are outside those specified or intended" [15]	Shorter restore times Less downtime
Robustness	"Ability of an item to continue to function when subjected to circumstances that are outside those specified or intended" [15]	Less downtime Less risk to health and property
Fault tolerance	"ability to continue functioning with certain faults present" [15]	Less downtime, less risk to health and property

Table 2.2: Generic system applications and desirable quality attributes.

Application	Desirable quality attributes	Remarks
TV remote	Manufacturability	Remotes can be economically mass-produced and discarded when faults occur.
Proof-of-concept system prototype	Reliability, safety	Need not be economical to manufacture. Should not cause physical injury.
Infotainment	Reliability, manufacturability, maintainability.	Does not carry out safety-related functions. Safety is not affected if a unit fails. It should be easy to repair.
Telemetry system	Reliability, availability, maintainability, safety.	Similar to infotainment by physical architecture, however, the telemetry carries out safety-related functions.
Engine Control Unit / Traction Control Unit	Reliability, availability, maintainability, manufacturability, safety, supportability, dependability, ...	Incorrect engine control may lead to a breakdown at best and multiple losses of life at worst.

2.2. Safety-Instrumented Systems

A safety-instrumented system (SIS) may be a system-of-systems, comprising multiple safety-instrumented systems and non-critical systems. A system function that is responsible for reducing risks to people is called a safety-instrumented function (SIF). A safety-instrumented system may implement one or more safety-instrumented functions. The relations described so far are shown in **Figure 2.1**.

Requirements for safety-instrumented systems are internationally standardized in [16] for generic applications. Further, there are industry-specific derivatives of this standard, such as [14] for railway applications, [5] for automotive applications, and more, as shown in **Figure 2.2**. Hata! Başvuru kaynağı bulunamadı..

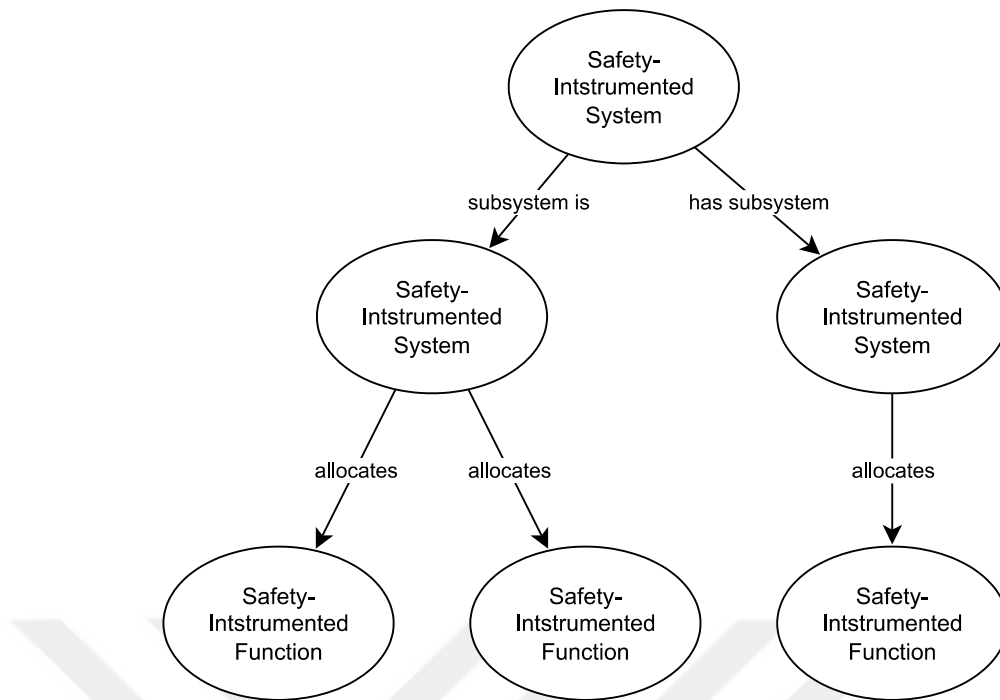


Figure 2.1: Example structure of a safety-instrumented system-of-systems.

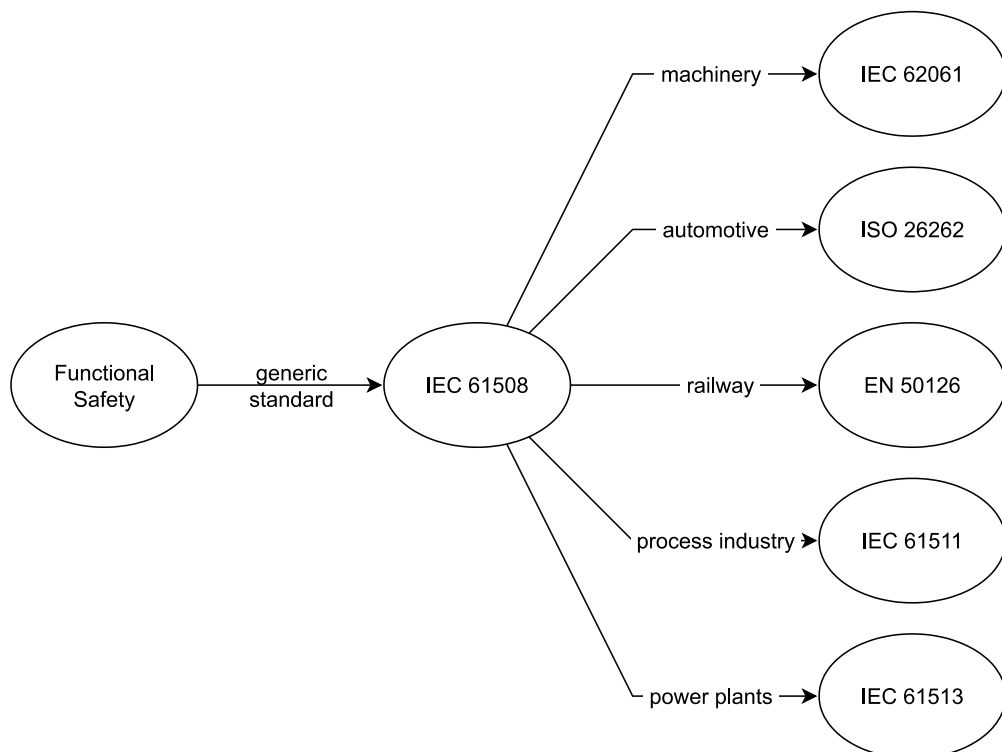


Figure 2.2: Overview of functional safety standards.

Safety-instrumented functions have the goal of reducing certain risks down to acceptable levels. No system is risk-free, and determining acceptable risk levels and the risk acceptance criteria is a difficult process in itself, with ethical issues related to deeming a certain amount of irreversible harm to human health "acceptable" [17], [18].

An example of a safety-instrumented system is an overflow sensor used in the process industry. While the liquid level is normally measured by a primary sensor, an independent and oftentimes simpler secondary "overflow" sensor shuts down the inlets when liquid is sensed near the container's limits. The overflow condition can still occur due to an inlet valve that is stuck-open with no backup valves; however, the risk originating from the primary sensor errors are greatly diminished.

2.3. The System Life Cycle

A single inadequate component can result in not achieving the desired system quality attributes. One could imagine an intrusion-resistant safe with particularly weak hinges. Since the hinges compromise the integrity of the safe, it no longer matters how pick-proof the lock is, nor how thick the steel walls of the safe are.

To overcome coordination problems between subsystems, a holistic approach to the development processes is needed for efficient system development. In this approach, system quality attributes should be specified at the system-of-interest level to align the development efforts. The system quality attribute needs should be integrated into the system requirements and the system architecture to direct the system design. Following the system requirements and the system architecture, necessary capabilities should be implemented into the lower system levels and integrated backup to the system-of-interest level to achieve the desired system quality attributes.

The discipline of systems engineering aids the systems development processes by providing a holistic approach to the system development efforts. [19] reports that systems engineering activities and project success levels have a positive correlation, with the best-performing programs spending 15-20% of the development budgets on system engineering activities.

IEEE 15288 standard [20] defines the systems engineering concepts and principles, and introduces the system life cycle concept. System life cycle is an overarching system development process that encompasses multiple life cycle "phases" that start

with the concept and project kick-off and end with the decommissioning and disposal. However, IEEE 15288 does not present or suggest a specific system life cycle model.

2.3.1. System Life Cycle Models

There are many styles and versions of system life cycle models, such as the waterfall model, the incremental model, the spiral model, the V-cycle model, and more [21]. Each life cycle model representation has certain strengths and shortcomings. Each system life model can work well with certain disciplines, system complexities, and application areas.

An example of a "V-cycle" representation of a system life cycle model, which is suggested by the railway electronics systems applications standard EN 50126 [14] is shown in **Figure 2.3**. This form of V-cycle system life cycle model gets its name from the shape originating from its top-down and bottom-up nature. The first half of the V-cycle, which can be called the "top-down design" half, is concerned with

- understanding the system within its intended application context,
- understanding what the stakeholders (not just customers) need,
- understanding what the system needs to accomplish,
- exploring how the system could be structured, and finally,
- implementing a valid and effective solution to fulfill the stakeholder needs

Followed by the top-down design, the second half of the development life cycle can be called the "bottom-up integration" half, which is concerned with

- making sure the as-manufactured parts and subsystems match the as-designed system
- planning and carrying out the integration of the subsystems and the system-of-interest
- demonstrating that the as-manufactured system fulfills the stakeholder needs
- earning the system acceptance by the customer
- planning and carrying out the commissioning of the system to its intended application context

- planning the decommissioning processes so that the system owner can scrap or recycle the system at the end of its useful life.

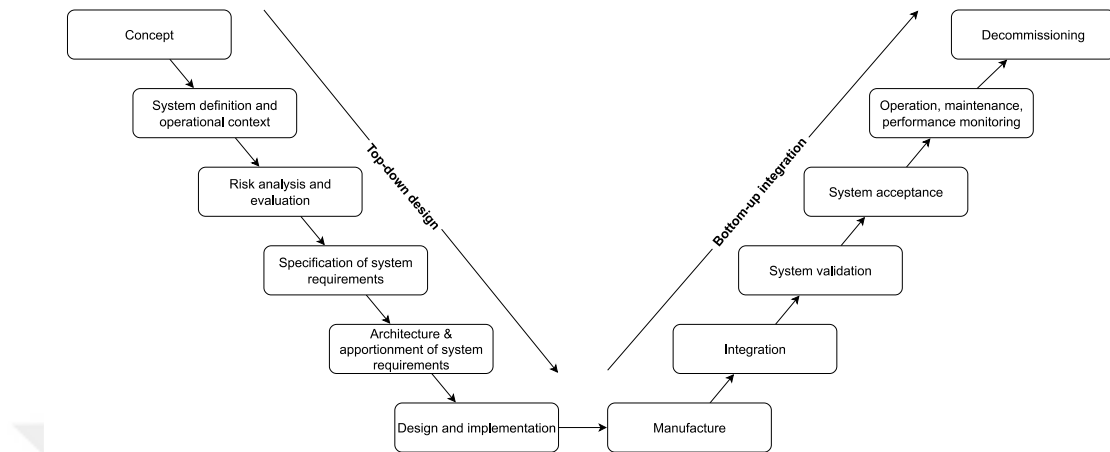


Figure 2.3: Simplified system life cycle (the V-cycle) model omitting verifications and iterations.

An important feature of the V-cycle is that it is recurrent, meaning the same life cycle model can be reapplied to lower system level development processes [22], as shown in **Figure 2.4**. Recursiveness minimizes process complexity while also assuring consistency in the development of all system parts.

Depending on the rules imposed by the top-level system life cycle, system life cycle models employed by lower system levels may use tailored life cycle models or even entirely different development methodologies.

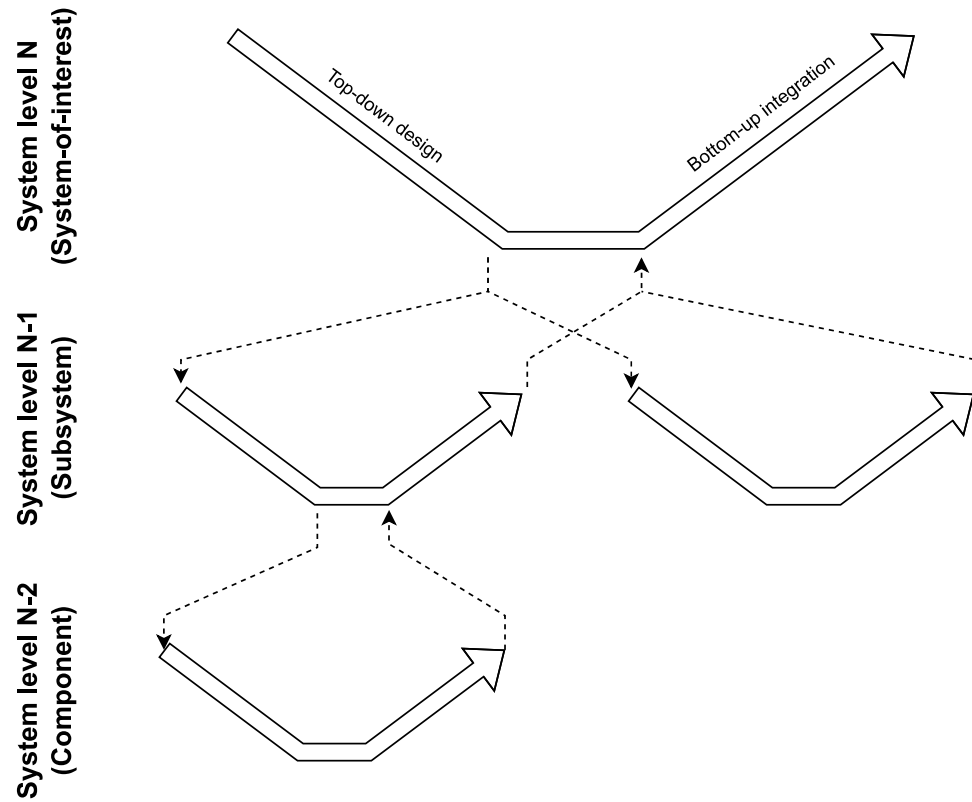


Figure 2.4: Recurrency of system life cycle models enables application to lower system levels.

3. DEVELOPMENT METHODOLOGY

In this section, the methodology used in the development of the system-of-interest is presented. Since technical management processes like quality assurance, configuration management, etc., are generic in nature, they are out of the scope of this work. However, these processes enable the correct functioning of system development life cycle processes; therefore, they are inseparable from the works described in this thesis.

During the development process, the overall life cycle model suggested by EN 50126 [23] was accepted and followed as-is. Summarized core activities performed during each phase of the system life cycle as suggested by EN 50126 are given in **Table 3.1**.

3.1. Iterative Risk Reduction

Each diagnostic capability adds to the system complexity, development efforts and manufacturing costs. Therefore, each diagnostic capability requires justification for inclusion in the system scope. Since the value proposition of diagnostic capabilities is the marginal risk reduction offered, first we need to define the risks related to the system. Following that, we need to include a subset of diagnostic capabilities to the system architecture. Each diagnostic capability set can be realized by numerous different designs, each with their unique strengths and weaknesses. However, each diagnostic capability implementation can generally detect only a subset of potential failure modes, therefore the offered risk reduction may not completely align with the system architecture and system requirements. This problem can be solved with an iterative approach to system design, where a loop consisting of risk analysis, system requirements specification, system architecture development, design and implementation processes are repeated until a prescribed tolerable risk is achieved. This approach is modeled in **Figure 3.1**.

Table 3.1: Core activities performed in system life cycle phases as suggested by EN 50126.

Phase No	Phase Name	Core Activities
1	Concept	- Investigate the purpose, scope, context, and the environment of the system. - Investigate the system from the RAMS perspective.
2	System definition and operational context	- Define the system and the system boundary - Define the mission profile - Define project technical management - Establish the RAM policy and the RAM plan.
3	Risk analysis and evaluation	- Perform risk analysis - Establish hazard log
4	Specification of system requirements	- Specify system requirements - Specify RAMS requirements
5	Architecture and apportionment of system requirements	- Develop and specify system architecture - Allocate system and RAMS requirements
6	Design and implementation	- Specify subsystem and component designs - Perform RAMS analysis of the implemented system - Prepare the safety case
7	Manufacture	- Manufacture the system
8	Integration	- Integrate and verify the system
9	System validation	- Perform system validation
10	System acceptance	- Obtain formal system acceptance from the customer
11	Operation, maintenance, and performance monitoring	- Perform configuration management on manufactured products.
12	Decommissioning	- Decommission and dispose of the system

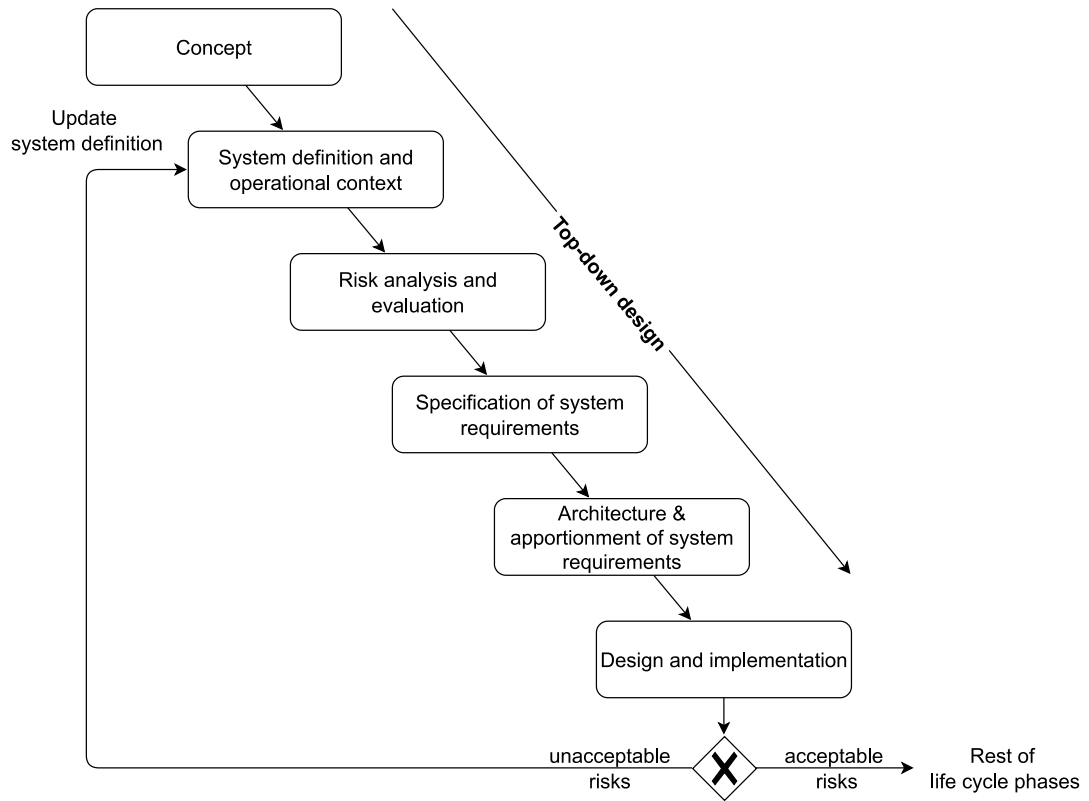


Figure 3.1: System life cycle with iterative risk reduction.

On the first iteration of the risk reduction cycle, a concept level Failure Modes and Effects Analysis (FMEA) study is performed. FMEA is a bottom-up safety analysis tool and is useful for exhaustive and systematic inspection of system failure scenarios. The preparation and performing of FMEA are described in the IEC 60812 standard [24]. The following fields are tabulated during the analysis:

- All components belonging to the system,
- All functionality of components,
- All failure modes of component functions,
- Effect of component failure modes on the system functionality,
- Probability of occurrence of failure modes, denoted with λ in units of $1/h$.

Concept level FMEA study inspects the system functions within the scope of system definition for foreseeable failure modes and their effects on higher level system operations. Following that, a Failure Modes, Effects and Criticality Analysis (FMECA) study, which is a variant of FMEA that inspects and documents whether a

failure mode of the system results in dangerous or safe failures. During the FMECA study, the following fields are added to the FMEA data:

- The criticality of the system failure caused by the component failure mode, classified as safe or dangerous
- The inherent (meaning in isolation) safe failure rate of the component, calculated as in equation (3.1):

$$\lambda_S = \sum_k \lambda_k, \quad \forall k \in \{\text{component safe failure modes}\} \quad (3.1)$$

- The inherent dangerous failure rate of the component, calculated as in equation (3.2):

$$\lambda_D = \sum_k \lambda_k, \quad \forall k \in \{\text{component dangerous failure modes}\} \quad (3.2)$$

Using the calculated risks and criticality classification, an initial set of diagnostic capabilities are introduced to the system definition.

To calculate the impact of the set of diagnostic capabilities within the system definition on the system risks, a Failure Modes, Effects and Diagnostics Analysis (FMEDA) study is performed. FMEDA is built on the foundation of FMECA, with the addition of the following fields:

- the diagnostic capability within the system definition that can partially or fully detect the fault or the error, or the resulting system failure,
- the likelihood of the error or failure being detectable by the diagnostic method,
- the resulting reduced system risks associated with the function, as calculated by the safe failure fraction (SFF) metric [25] is given in equation (3.3):

$$\text{SFF} = (\lambda_S + \lambda_{DD}) / (\lambda_S + \lambda_{DD} + \lambda_{DU}) = 1 - \lambda_{DU} / \lambda \quad (3.3)$$

The decomposition of the system to its failure modes and the risk analysis of failure modes are depicted in **Figure 3.2**. Analyzing the conceptual system functions in terms

of safety helps inform the system requirements, architecture, and design processes, which in turn result in a system design specification that requires fewer risk reduction iterations. If the initial system definition does not satisfy system risk requirements, the second iteration of the risk reduction loop begins with a new set of diagnostic capabilities being introduced to the system definition.

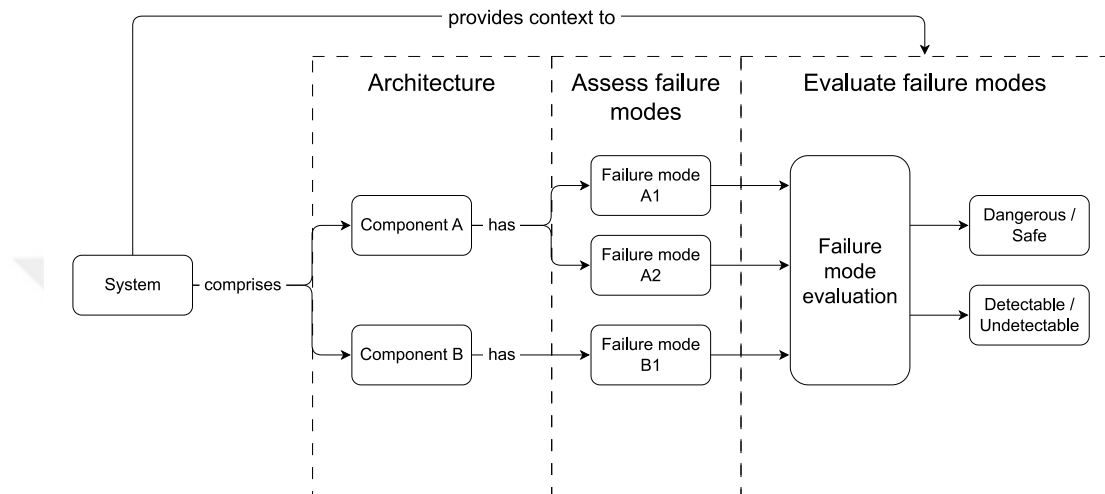


Figure 3.2: Structural decomposition and analysis of the system failure modes

4. SYSTEM CONCEPT

4.1. Business Need

Due to the usual production life cycles of electronic parts, and abnormal events such as the infamous global chip shortage of 2020 [26], hardware subsystems may need to change and adapt to the present market conditions. However, processing units such as microcontrollers and FPGAs can be connected to numerous signals, subsystems and even external systems. This means that the processing units are tightly coupled to the rest of the system, which makes them difficult to substitute. Each new substitute of a processing unit would likely result in complete redesigns of subsystems or the whole system. To fix this issue, we can define a standardized interface between a processing unit and the rest of the system, which would result in loose coupling and less rework down the line.

Another problem is that safety-instrumented system safety assurance and certification processes are lengthy and resource-intensive. Some subsystems share commonalities in form and function; however, normally each must be proven for safety. To minimize the amount of recurrent engineering efforts, we can develop complex subsystems into modules, each of which would be separately certified and reused in present and future projects.

Modularization of processing units, or the FPGA subsystem in particular, for the purposes of this study, solves both aforementioned problems at once. A comparison of different form factors for FPGA subsystems are compared in the benchmarking table given in **Table 4.1**. Further, some key advantages brought on by the modularization of the FPGA subsystem for functional-safety applications are listed as follows:

- Modularization decouples the design, manufacturing, and verification processes of the generic FPGA module and the application-specific carrier boards.
- Carrier boards and FPGA modules can be separately manufactured and stocked, enabling risk reduction strategies against procurement issues.

- The defined interface between the system-of-interest and the carrier board enables changing either the carrier board or the FPGA module without affecting the configuration of the other subsystem.
- Moving high-density interconnect from the carrier board to the much smaller FPGA module leads to better manufacturability and lower manufacturing costs.
- The safety integrity of the designed module can be proven once and reused multiple times in multiple projects.

Table 4.1: Benchmarking table comparing different FPGA subsystem form factors.

Property\Form factor	Direct integration to board	Dedicated FPGA board + application-specific board	Module on carrier board
Design density	Medium	Low	High
Manufacturing scalability	Medium	Low	High
Recurrent design engineering	High	Low	Low
Recurrent safety assurance effort	High	Low	Low
Concurrent development support	No	Yes	Yes

4.2. Intended Application and System Context

The system-of-interest is developed to be used on carrier boards that implement the application-specific hardware, such as I/O ports and signal conditioning circuits, as shown in **Figure 4.1**. The smallest form factor electronic board that is targeted by the FPGA module conforms to 3U 100 mm Eurocard format [27]. The system-of-interest is mechanically and electrically coupled to the carrier board through the edge connector.

To capture the system environment and the high-level interactions of the system with its environment, the context diagram shown in **Figure 4.2** was prepared. A context diagram is a useful tool for improving the high-level system understanding and communication of the envisioned system to stakeholders.

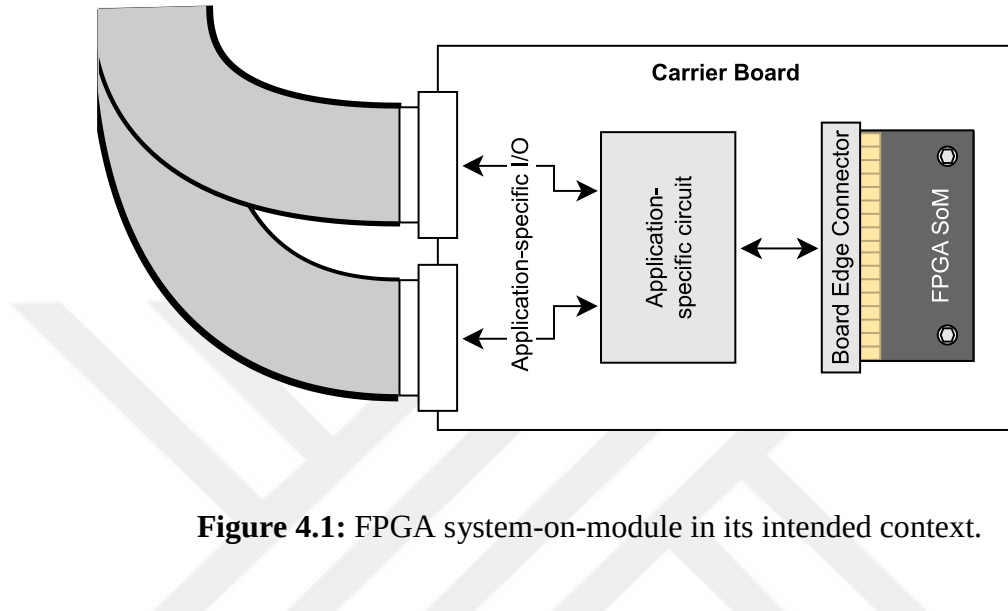


Figure 4.1: FPGA system-on-module in its intended context.

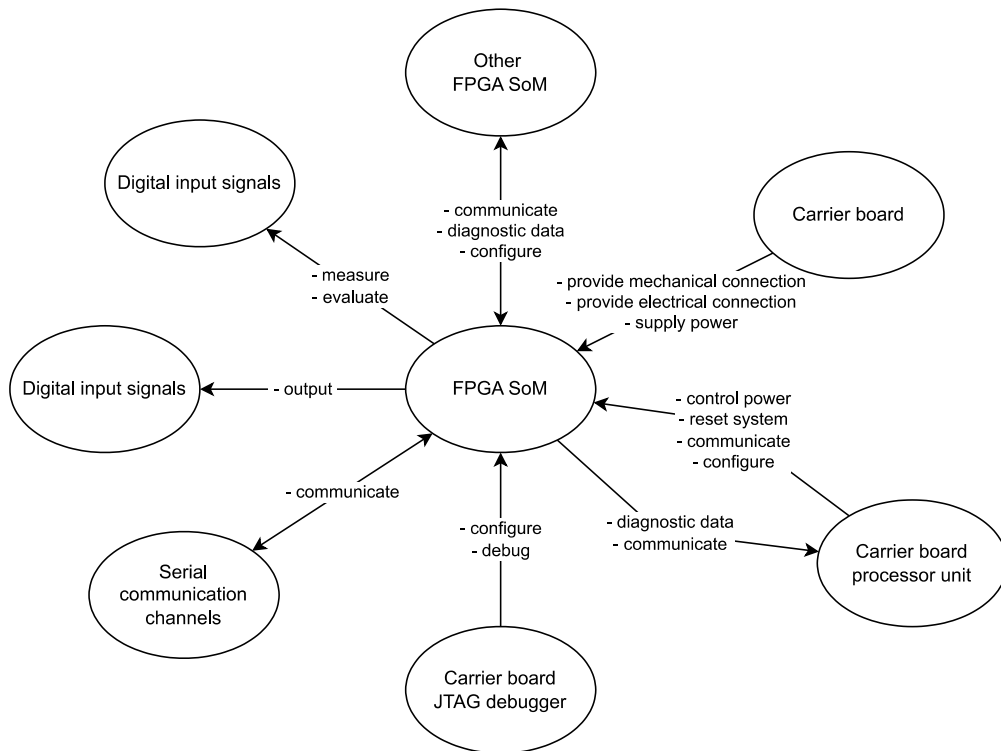


Figure 4.2: Context diagram depicting the interactions with the system environment.

5. SYSTEM DEFINITION

A system includes the parts and their interactions within themselves and with the external elements [28]. Therefore, a system definition cannot be complete without specifying how the system interacts and behaves, as shown in **Figure 5.1**. While modeling and conceptualizing the physical architecture (which includes the modules of software) is usually intuitive to engineers, the same cannot be said for interactions and behaviors [29]. However, as shown in **Figure 5.2** which presents a modified version of the five whys root cause analysis technique, the development of system interactions and behavior is a necessary condition for successful system development. Developing system interaction and behavior definitions presents the best-effort understanding of the system in its working environment, which enables early and accurate stakeholder feedback. The cost of changing design direction increases exponentially as the system development life cycle proceeds [30], which means defining the system interactions and behavior has potential for great return on investment.

Taking **Figure 5.1** as a reference, the following must be defined in enough detail to completely define a system-of-interest:

- The boundaries of the system-of-interest
- The mission profile of the system-of-interest
- The stakeholders, especially those that interact with the system during commissioning, operation and maintenance
- The missions, scenarios and use-cases of the system
- The interfaces of the system-of-interest with external elements
- The interactions through those interfaces
- System behavior during normal operation
- System behavior during failures, including those originating from inappropriate operating conditions

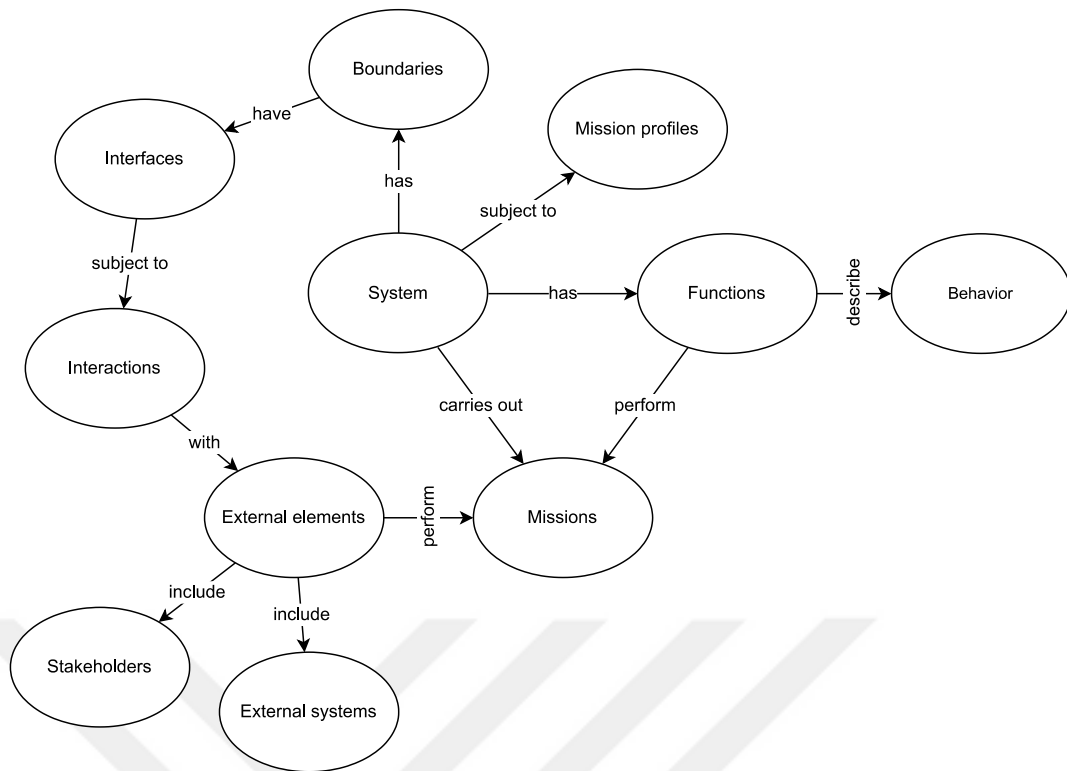


Figure 5.1: Concept map showing the relationship between systems, behaviors and interactions.

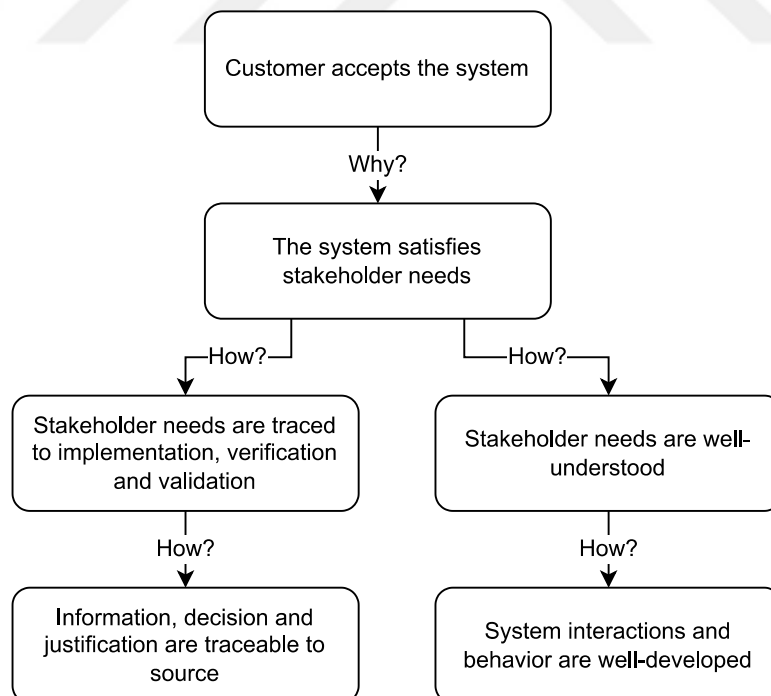


Figure 5.2: The relations between project success and the system definition.

While system definition development is modeled as a single iteration process in the system development life cycle model presented in Section 2.3, system definition can be considered a living document that is updated throughout the system development life cycle. Changing any information regarding the system definition could have larger effects on the overall system due to system interactions and emergence. Therefore, to assess the effects of the modification, many of the subsequent phases should be re-evaluated for applicability to the new version. If inconsistencies or incompatibilities are discovered in the downstream information, related processes should be re-run to update said downstream information to catch up with the updated system definition.

The understanding of the system improves throughout the system development life cycle. Therefore, while the system development life cycle model is shown linear for brevity, it can reasonably be expected to include iterations. Some possible iteration scenarios are shown in **Figure 5.3**.

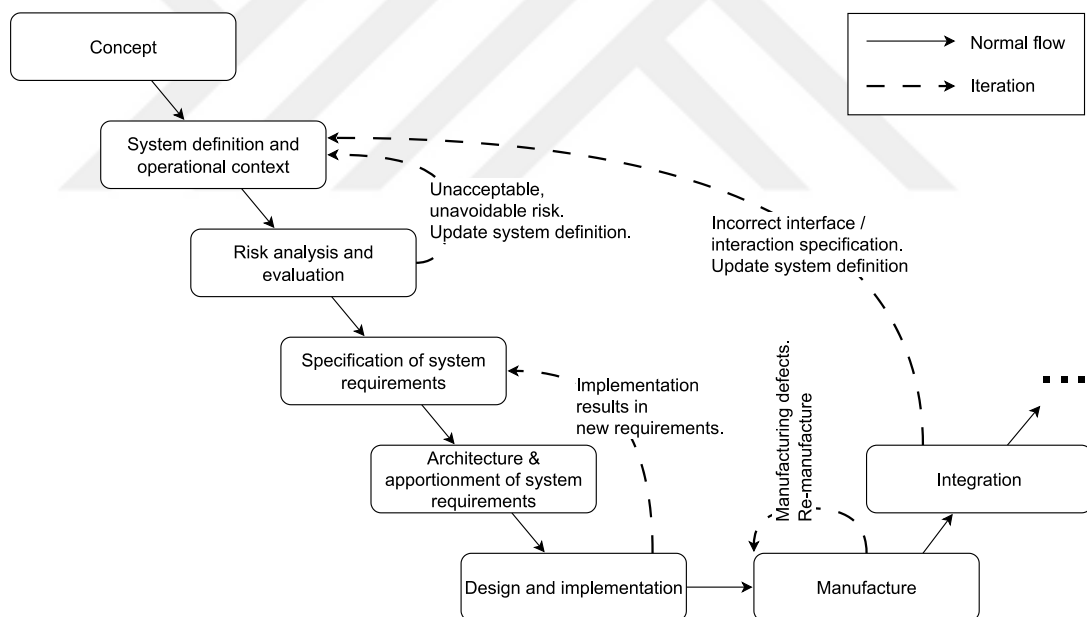


Figure 5.3: Some life cycle phase iteration scenarios, some of which affect the system definition.

5.1. Use Cases

Use cases are used to discover and communicate the functions or services provided by the system, which determine the meaning and the purpose of the system [31]. Use cases refine the mission definitions, which in turn form the basis for system functions.

Use case diagram is an important representation tool that aim to plainly describe the use cases that can be understood and commented on by stakeholders. It consists of the actors, which may be human or non-human, that interact with the system and the functions performed by the system-of-interest.

In **Figure 5.4** and **Figure 5.5**, use case diagrams depicting the system interactions in select scenarios are presented. In **Figure 5.4**, the scenario of power subsystem transforming and managing the input power is presented. In this use case diagram:

- the carrier board supplies the power to system,
- the carrier board controls the power of the module (power enable),
- the carrier board monitors the health of the module power (power good),
- the system regulates its power, which also includes receiving, controlling, transforming and finally monitoring the power.

In **Figure 5.5**, the scenario of application-specific system operation is described. In this use case diagram,

- a system application developer team develops the application logic and the FPGA configuration bitstream,
- an external system (system 1) provides some input to the system,
- the system evaluates the input signals and generates some output signals following the application logic,
- the output signal is connected to multiple external systems,
- information is exchanged between an external system (system 3) over a serial communication network.

These two examples show that a simple diagram can be a great tool for documentation, communication, and stakeholder engagement.

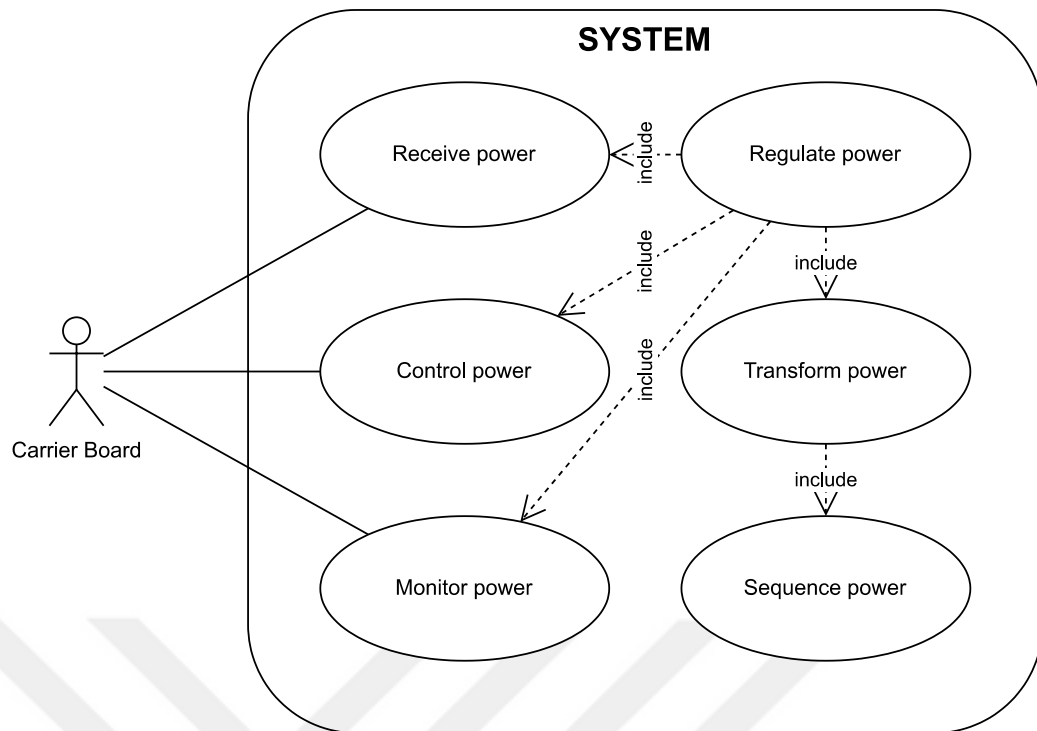


Figure 5.4: Use case diagram depicting the power regulation interactions with the carrier board.

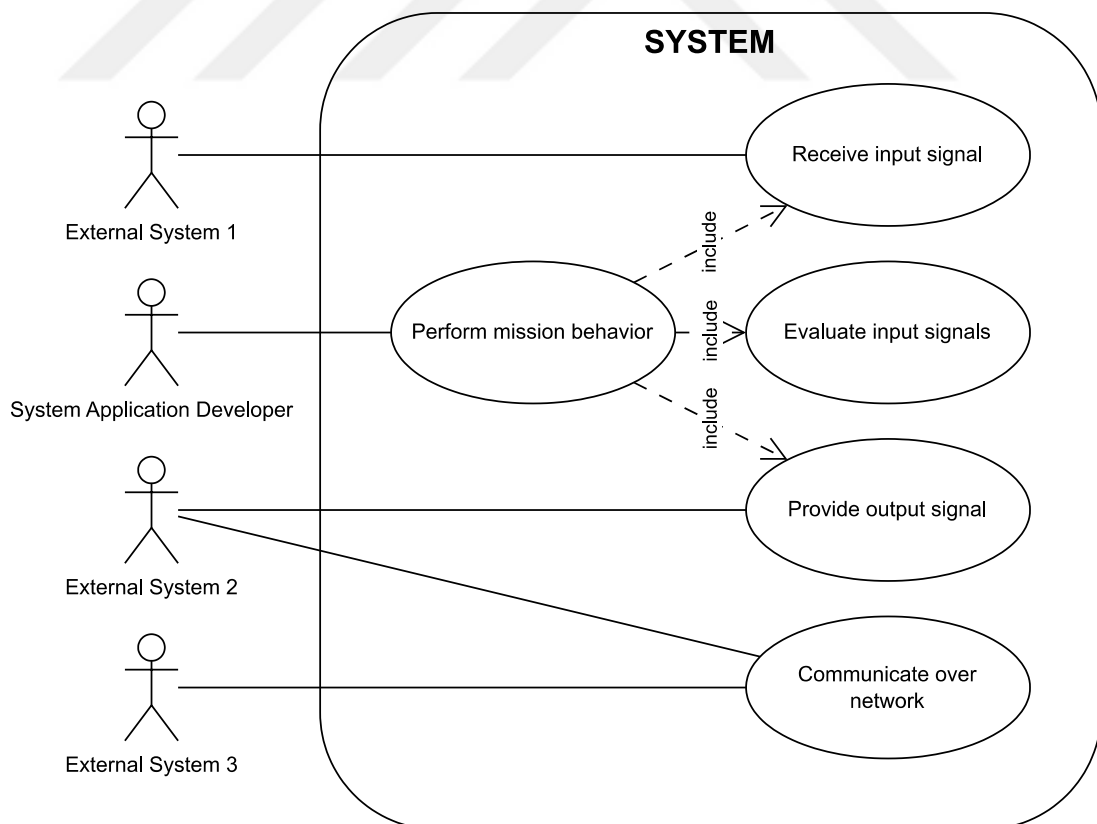


Figure 5.5: Use case diagram depicting the system operating according to application-specific software.

5.2. Functional Breakdown Structure

A system is a complex concept with multiple interrelated aspects. In order to make the problem of system development more tractable, cross-sections of the system in different domains can be analyzed independently and then related to each other afterwards. A system has a "physical" structure, each of which are performing "functions" which are based on "missions" to satisfy stakeholder and system "requirements", where each term in quotes represents a domain of systems. Physical structure and functional structure of systems are interrelated, as functions are allocated to physical components and contrarily physical components exist to carry out functions. However, analyzing these two domains independently of each other and then analyzing the relations between these two domains can enable clearer understanding and communication of the system, which in turn leads to better development outcomes. An example framework for multi-domain analysis is presented in [32].

System functions can be defined in an iterative approach where the mission space and the function space are developed iteratively, each update in one domain informing the other domain and vice versa. Further, the identified functions can be represented in a hierarchical structure, called the Functional Breakdown Structure (FBS). FBS has the added benefit of being compatible with iterative functional specification, starting from the specification of top-level system functions and gradually refining those to specify details. The specification of the functions hierarchy can be accomplished in

- tabular form, where each row represents a function and the hierarchy is represented within the function ID or in dedicated parent/child function fields, or,
- model-based diagrammatic form, where functions are linked to parent and child functions, mission-function allocations, function-system element allocations, and more

Each representation has its advantages. Tabular form can be more lenient for use in simpler systems or by staff not educated about model-based systems engineering. Model-based diagrammatic form has the usual advantages of model-based systems engineering, such as completeness, consistency, and communication of information, along with the high knowledge barrier to entry [33].

In **Table 5.1**, an overview of identified fundamental system functions for the system-of-interest is presented. Further, the first function F1 is refined to lower-level functions to exemplify the use of breakdown structures. The complete specification of the functional breakdown structure is out of the scope of this thesis.

Table 5.1: Excerpt from the Functional Breakdown Structure of the system-of-interest

Function ID	Short Name	Description
F1	Process digital signals	The system provides output signals as prescribed by the application-specific logic and the input signals received from external systems.
F1.1	Measure input signals	The system measures the digital input signals.
F1.1.1	Condition input signals	The system applies signal conditioning to digital input signals.
F1.1.2	Interpret signal	The system translates the digital input signals to symbols.
F2	Evaluate signals	The system evaluates the digital input signals.
F3	Provide output signals	The system provides the output signals as prescribed by the application-specific logic.
F4	Communications	The system participates in network communications, either point-to-point or multidrop, either as host or client, as prescribed in the system interface specifications.
F5	Receive power	The system receives, conditions, and uses power supplied through the edge connector.
F6	Receive control signals	The system receives and conditions digital control signals from the edge connector and uses these to control the system state.
F7	Provide status signals	The system provides status feedback signals to the edge connector.

5.3. System Interfaces

A system provides services, and therefore value to the stakeholder, through its interfaces. Also, since interactions are part of what defines a system, identifying and managing system interfaces starting from the early stages of development leads to a better understanding of system concepts and therefore helps in achieving better project outcomes.

In this stage of the project development, the interfaces of the system can be identified and analyzed as black boxes. In this manner, we first identify system interface ports through the help of a context diagram, which was already developed in the concept phase and shown in **Figure 4.2**. Then, we consider the following matters for each identified interface port:

- Who: Which actors and external systems are interfaced with through this port?
- What: What is exchanged through this interface? Energy, signal, force, or mass? What are the meanings of physical signal magnitudes?
- When: What operational scenarios employ this interface?
- How: What are the expected interactions through this port?

Answering and documenting the answers to these questions, we develop a better understanding of the system in its working environment. It should be noted that at this point in development, a broad understanding of electrical signal magnitudes and corresponding symbols should be documented to be fed into the following life cycle phases: risk assessment and system requirements specifications. To emphasize the importance of specifying the signal magnitudes, we present the following cases:

- An electrical interface may be designed to be fail-safe against power losses or cable breaks through the use of:
 - live-zero signaling [34],
 - active-high signaling, similar to a dead man's switch
- An electrical interface may require special care, such as galvanic isolation, depending on the signal levels.

- An electrical interface may require signal conditioning, which in turn require a specification of signal levels and signal rates.

A preliminary specification of system interfaces based on the context diagram given in **Figure 4.2** and the previous discussions are provided in **Table 5.2**.

Table 5.2: Preliminary system interface specifications for the system-of-interest.

Interface ID	Interfaced element	Interface medium	Interface description
IF1	Carrier board	Force	- The system is mounted by its edge connector and mounting holes on a compatible carrier board.
IF2	Carrier board	Electrical	- The system receives power, control signals from the carrier board. - The system provides status signals to the carrier board.
IF3	Carrier board External elements	Electrical	- The system receives digital input signals at arbitrary levels, transformed over the carrier board to system supply levels. - The system provides digital output signals at arbitrary levels, transformed over the carrier board to system supply levels. - The system communicates with external systems through point-to-point and bus serial communication interfaces.
IF4	Carrier board microcontroller	Electrical	- The system provides diagnostic information to the programmable element on the carrier board, if applicable.
IF5	JTAG	Electrical	- The system provides JTAG connectivity to supporting components for testability and maintainability.

6. RISK MANAGEMENT

Risk management is the set of processes, including risk identification, risk analysis, risk mitigation, and risk tracking, that aim to holistically mitigate the risks associated with any applicable domain. These applicable domains include but are not limited to: systems, missions, projects, processes, and organizations. In the context of safety-instrumented systems, risk management is concerned with all these domains, as the developed system is inherently in interaction with each of these domains. To exemplify:

- Unforeseen interactions within missions may lead to hazards; therefore, mission definitions are a source of risks
- A project or program may consist of multiple interacting systems that are simultaneously in development, which makes it vulnerable to moving target problems, where frequently changing specifications may lead to design errors.
- A process may be inappropriately defined or carried out, leading to adverse outcomes that may be inputs to further system development processes.

Therefore, the holistic approach to risk management considers all aspects of the system development and operations to achieve the desired level of safety.

At this point, the distinction of errors into systematic and random failures is useful and important. Systematic errors of a system include system failures due to:

- incorrect or insufficient specifications resulting from inadequate development processes, quality assurance, lack of traceability and human errors,
- incorrectly programmed software that behaves unexpectedly, especially in edge cases,
- software bugs,
- operation outside the absolute maximum ratings of the equipment, including interfaces and operating temperatures,

- supply chain reliability issues, where some parts are subject to manufacturing defects, which lead to elevated failure rates, even if those failures manifest as random hardware failures,
- conflicts of interest in system development, such as a validator having risk of repercussions if the system is rejected

Adding to the systematic errors, systems are also subject to random failures that can be modeled and predicted using statistical tools and methods that may occur due to:

- random hardware failure governed by the failure rates of components within their useful lifetimes,
- soft errors due to the effects of radiation on semiconductor devices,

Risk analysis is a systematic process that includes discovering, evaluating, and tracking risks related to an aspect of a product or a process. Within the context of functional safety equipment, risk analysis is concerned with the safety impacts of product failures and reasonably foreseeable misuse. A safe system is not necessarily useful on its own unless it is also dependable. Many failure modes and misuse can also affect the dependability characteristics of a system. Therefore, within the context of dependable systems, risk definition also includes concepts that may adversely affect the systems' reliability, availability, maintainability, security, and more.

While the discovery of risks, apportionment of safety (or dependability) requirements, and the apportionment of safety-related functions are top-down processes, said functions are implemented at the lowest level and are integrated up to the system level where they are ultimately put to use in the operational environment. This logical flow between processes mimics that of the overall systems development life cycle. Therefore, where the safety or the dependability of the system is critical, a safety life cycle exists and runs alongside the systems development life cycle. These two life cycles are inherently intertwined - system definition forms the basis of risk analyses, and the results of risk analyses affect the system definition, as shown in **Figure 6.1**.

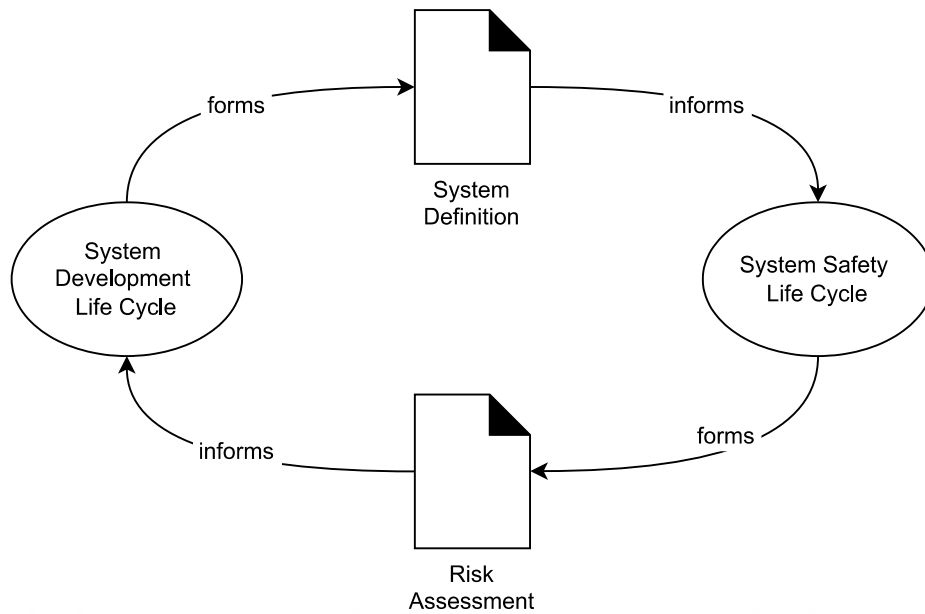


Figure 6.1: The interactions and the iterations between the system definition and the risk assessment.

Just as the systems development life cycle is divided into top-down design and bottom-up integration sections, the safety life cycle is composed of top-down analysis and apportionment and bottom-up implementation and integration sections. In the following sections, the top-down and bottom-up sections of the safety life cycle will be described in order.

6.1. Top-Down Risk Management

The effective management of system risks, including both the safety risks and the dependability risks, necessitates a holistic approach. A system itself does not pose risks on its own (excluding technical risks such as electrocution) - it is the disruptions in the functions provided by the systems that pose risks. Therefore, risks are only meaningful within the scope of missions and the overall operation, as shown in **Figure 6.2** and **Figure 6.3**. Hence, to discover and analyze risk related to the system, the developers must approach the problem from the mission space.

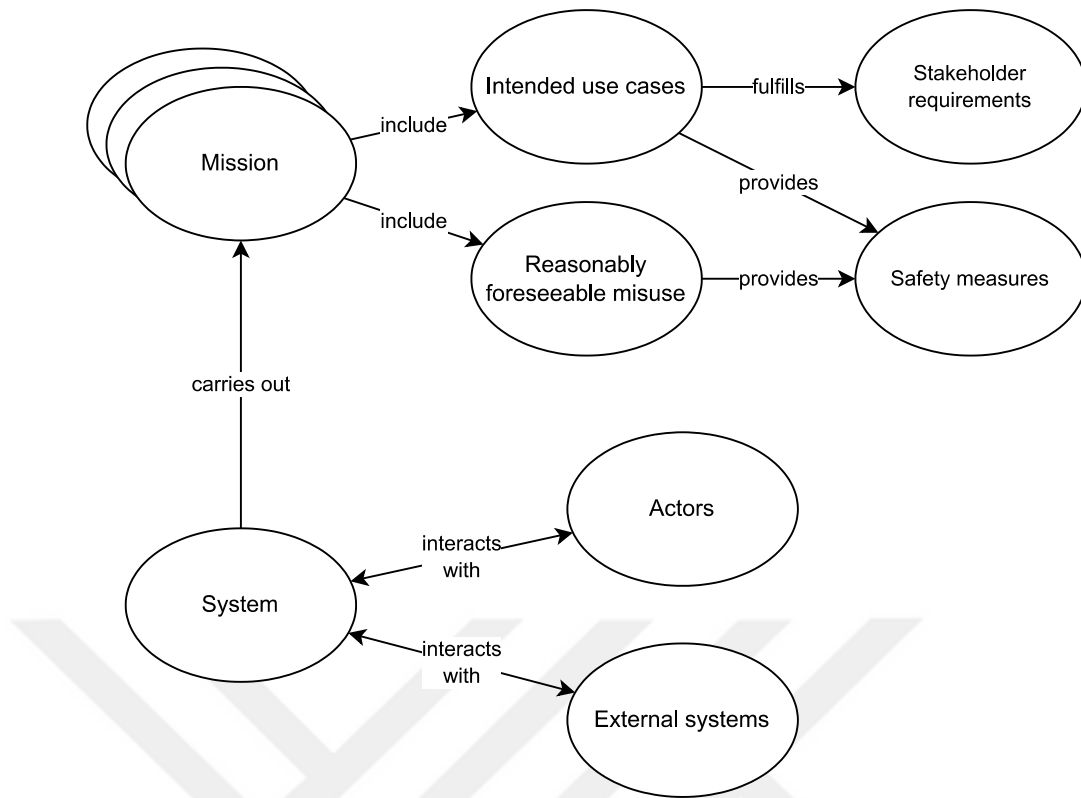


Figure 6.2: Safety measure definitions are informed by the system mission definitions.

Obtaining a prescribed acceptable risk level requires evaluation of the current risk levels and the resulting risk reduction requirements. A desired level of risk reduction can be obtained by various possible solutions in the architecture and implementation space. One could apportion functions differently to subsystems, employ more reliable components, employ hardware redundancy, or use any combination of all possible solutions. If the risk reduction requirement is not apportioned to subsystems, the integration of the developed subsystem will result in uncontrolled dependability characteristics.

The consequences of not approaching the safety aspects of system development in a systems thinking framework can be fatal, both in terms of project performance and especially in terms of public health. Further, management or design errors may be introduced to the system at any point in development since system development is a complex web of tasks. We identify a non-exhaustive list of safety-related management and design risks during system development in **Table 6.1**, which also present the possible impacts on the system development. The table also presents how the holistic, systems approach to system safety development solves these issues.

Inspecting the **Table 6.1**, we see that the resulting system implementation could be overengineered, complex, and expensive at best, and unsafe or unreliable at worst. The systems approach to safety management provides a robust framework on which the development can move on knowing that the combination of all life cycle tasks are necessary and sufficient to develop the right solution. The result is that all safety requirements are efficiently discovered and efficiently solved, resulting in lower system complexity, less development effort waste originating from reworks, shorter time to market, and finally, better public health through reduced risk to people.

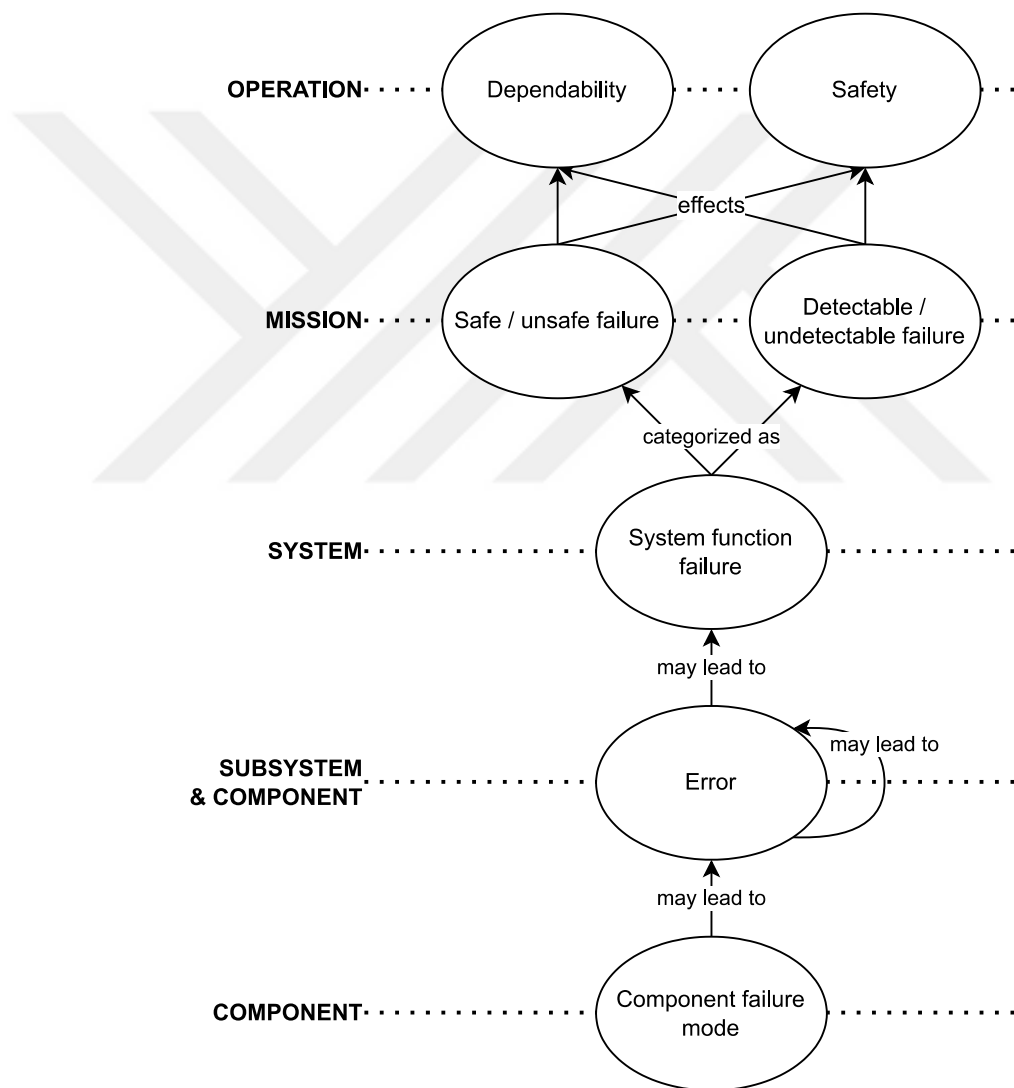


Figure 6.3: The link between component failures and system operational characteristics, visualized in increasing layers of abstraction.

Table 6.1: Problems arising from not employing deductive risk management methods.

Error Introduced During	Error Definition	Impact on the System Development	Holistic Approach Solution
Hazard discovery	Risks are defined in mission space; therefore, not considering the systems' interactions with external elements may result in unidentified hazards.	Unidentified hazards may result in reasonably foreseeable catastrophes. The customer may reject system acceptance.	Clearly define the system, including its environment and their interactions, to inform the hazard discovery task.
Requirements allocation	Boundaries of risk mitigation responsibilities are unclear. Due to the diffusion of responsibility effect (also known as the bystander effect) in psychology [35], each design team may wrongly assume that the other subsystem(s) will handle a particular risk issue, resulting in complete inaction towards the identified risk.	The resulting implementation will be rejected in the system validation phase. System development processes will have to be iterated beginning from the system definition.	Risks are traceable from risk identification to subsystem allocations and to implementation. The traceability begins from identified risks and includes safety requirements, safety functions, subsystems, and finally to safety requirement verification cases.
	Since boundaries of risk mitigation responsibilities are unclear, multiple subsystems may be developed to fulfill some safety requirements, where one would be enough.	The resulting system is more complex, development effort is higher.	

Table 6.1 (continued)

Error Introduced During	Error Definition	Impact on the System Development	Holistic Approach Solution
Requirements verification and validation	Safety-related functions pose particular difficulty in testing because usually to test these functions one needs to introduce hazardous faults into the system. For example, to test an overcurrent protection function, one may need to short some wires or busses.	Verification and validation of safety-related functions may be done partially or not done at all. This could result in a catastrophe in operation if there are design errors in said functions' implementations.	Systems engineering approach considers and prepares plans for system integration, verification and validation before the implementation begins. This results in introduction of provisions for testability, which helps overcome this problem.
Design and implementation	The design and implementation of the system and the subsystems may pose new hazards. Therefore, these new hazards must be identified and managed, which may require development iteration.	Unidentified hazards originating from the implementation may result in catastrophes.	Risk management is an ongoing process and risk assessments are evaluated and re-evaluated in prescribed points in the development life cycle, ensuring due diligence.
Any point in the development life cycle	With improving understanding of the system operations, new hazards are identified. However, the severity of a new hazard is misjudged to be negligible.	No measures are taken against the newly identified hazard. The system may cause harm during development or operation.	All hazards are tracked in a central hazard log. Each hazard is evaluated within the operational concepts and documented along with the rationale to classification.

6.2. Bottom-Up Risk Management

The bottom-up integration section of the safety life cycle is concerned with the correct implementation and the demonstration of safety-related functions, and also assuring that newly surfaced safety-related information is dealt with.

To demonstrate the correctness and adequacy of the as-designed safety-related functions, the development team prepares a safety case [23] the safety case is also used as a basis for conducting independent safety assessment, if applicable to the system of interest. In this safety case, the following are presented:

- Provides safety assurance to stakeholders.
- Provides evidence that the as-designed system complies with system safety requirements.
- Provides safety-related application conditions, which include assumptions about the system operation and the environment, that must be met in specific applications. For example, system integrator could be instructed to put the system in a tamper-proof enclosure, and the system operator could be instructed to keep the enclosure locked at all times. Failure to do so could jeopardize the system through exposure to sabotage opportunities.

6.3. Hazard Analysis

Similar to the system safety life cycle, identification and evaluation of hazards have two complementary deductive (a.k.a. top-down) and inductive (a.k.a. bottom-up) approaches. Both approaches are visualized in **Figure 6.4**. The top-down analysis approach identifies hazardous events with the context provided by the system definition. Following that, hazardous events are traced back to their root causes. Contrarily, the bottom-up analysis approach begins with the definition of some components or subsystems. From the definition of these parts, possible failure modes are identified and traced up to discover hazardous events within the context provided by the system definition. If all the information were available, both of these approaches could model the failure chains individually. However, since hazards and cause-effect relations are usually not known beforehand, both of these approaches provide the

engineer with different perspectives into the system. This in turn helps with hazard identification and the correct evaluation of hazards.

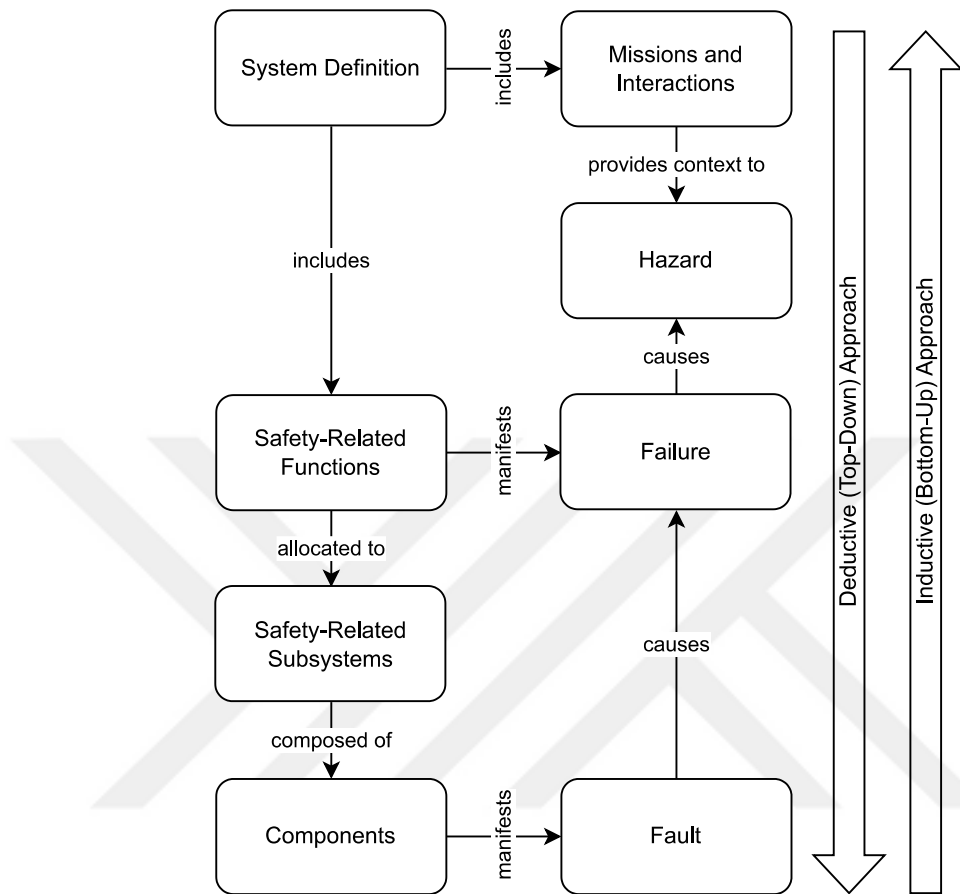


Figure 6.4: Hazard analysis can be approached deductively or inductively.

6.3.1. Fault Tree Analysis

The deductive approach to hazard analysis looks at the overall system operation, missions, and interactions, and identifies foreseeable hazardous events. For example, an emergency brake output not triggering when needed is a hazard. We don't need to know the root cause to come to this conclusion. However, we can use this hazardous event as a starting point to discover all possible causes in increasingly longer interaction chains. There are multiple cause-and-effect relations between events regarding this analysis:

- Direct relation: A prior event may directly result in the hazardous event occurring.

- OR relation: Each of multiple prior events can directly result in the hazardous event occurring.
- AND relation: Multiple prior events need to happen for the hazardous event to occur.
- Combination: Any combination of prior events, combined through OR or AND relations can result in the hazardous event occurring.

When these events are diagrammatically presented the resulting study is called the Fault Tree Analysis (FTA), which was originally used by the U.S. military in 1961 and has since spread widely into many sectors [36]. Fault Tree Analysis is still widely used in many application areas and is an active area of research [37], [38]. The most commonly used building blocks in the Fault Tree Analysis diagrams are presented in the key given in **Figure 6.5**.

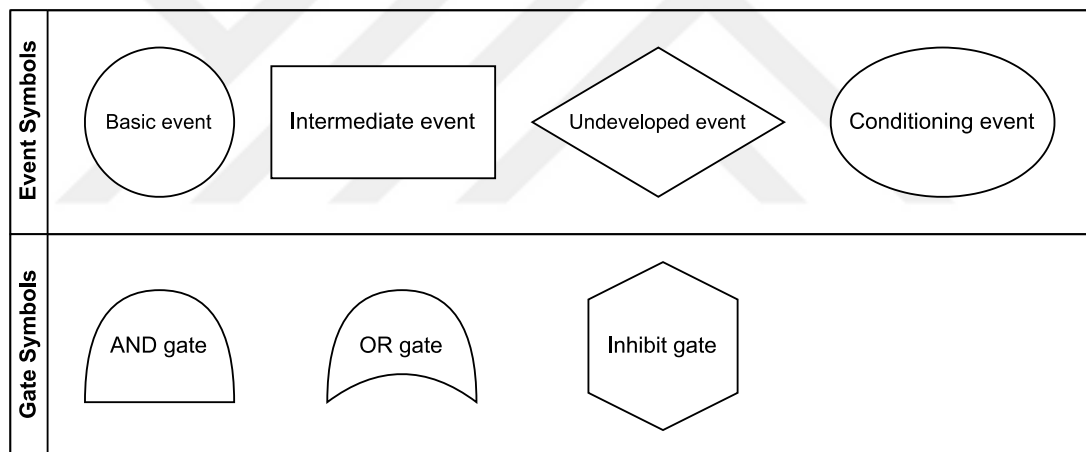


Figure 6.5: Key for Fault Tree Diagrams (FTA)

Fault events that are considered to be root causes as far as the current safety analysis is concerned are modeled using the basic event symbol. It is worth noting that the team carrying out this analysis should ideally lay out some rules regarding the scope of the analysis; otherwise, the root cause of an unintended memory bit flip could be traced all the way to the origin of the universe through the cosmic background radiation. OR gates indicate that multiple antecedent fault events may trigger the attached event, which is undesired in terms of dependability. Contrarily, AND gates indicate that the system has fault tolerance against antecedent faults and that the subsequent event is fault-tolerant. Given that all events are low probability and uncorrelated with each

other, the failure rate of a subsequent event is sum of fault probabilities of all antecedent events in the case of an OR gate as shown in equation (6.1), or the product of fault probabilities of all antecedent events in the case of an AND gate as shown in equation (6.2) [39]:

$$P_{\text{failure,OR}} = \sum P_{\text{fault},i}, \quad \forall i \in \{\text{antecedent events}\} \quad (6.1)$$

$$P_{\text{failure,AND}} = \prod P_{\text{fault},i}, \quad \forall i \in \{\text{antecedent events}\} \quad (6.2)$$

In **Figure 6.6**, the FTA diagram for the hazardous event of a general-purpose output signal failure is presented. Through this analysis we see that the output signal may fail due to multiple reasons, which is an indication that this particular failure mode poses reliability issues as failure rates of all cause events add up.

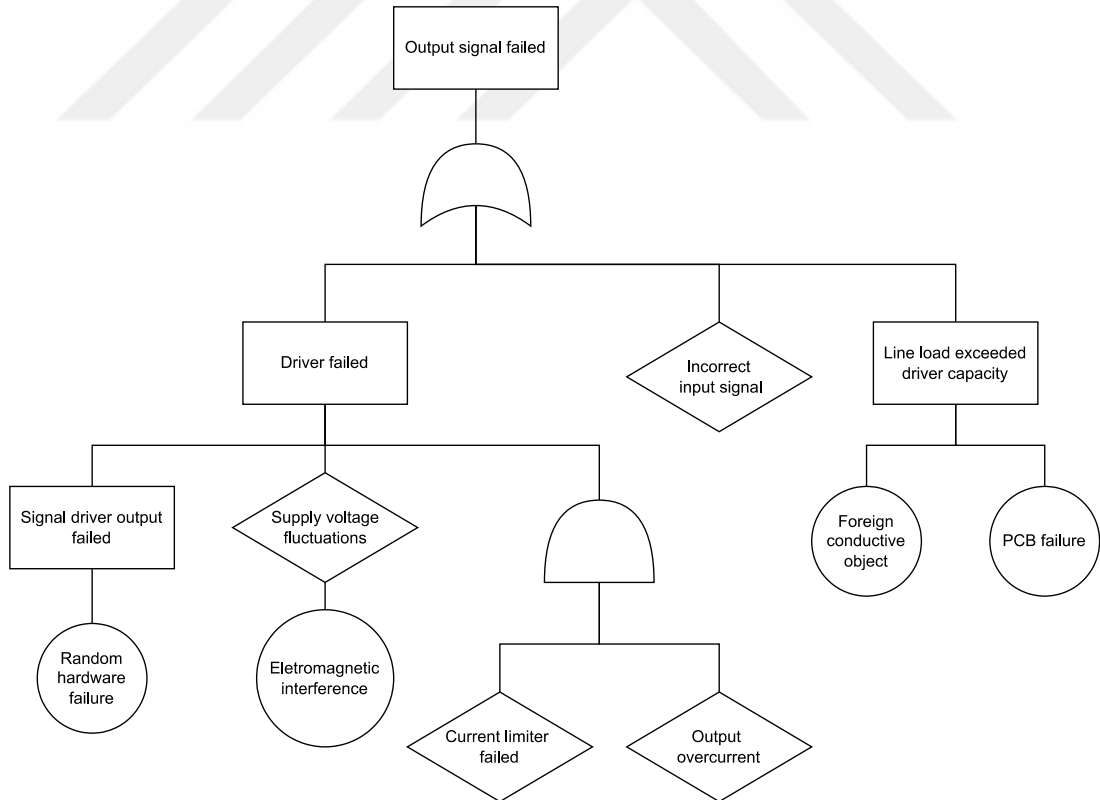


Figure 6.6: FTA diagram inspecting the causes of output signal failure event.

6.3.2. Failure Modes and Effects Analysis

The inductive approach to hazard analysis looks at the individual system elements to identify potential failure modes of all elements. Following that, each identified failure mode's impact on the increasingly higher elements are iteratively identified. Through this iterative process, each identified failure mode is traced up to the system functions, where they are evaluated in terms of safety, detectability, severity, and more.

The most common approach to inductive hazard analysis is the Failure Modes and Effects Analysis (FMEA) standardized in [24] and its derivatives. FMEA is used to identify single point of failure modes of a system. It is prepared and presented in a tabular form, where each row is concerned with a single failure mode. While FMEA is a valuable tool, it has its shortcomings. Some of these shortcomings originate from the fundamental assumptions imposed by the method, which are:

1. Only one failure mode exists at a time.
2. All inputs (including software commands) to the item being analyzed are present and at nominal values.
3. All consumables are present in sufficient quantities.
4. Nominal power is available.

The first assumption in particular poses a big limitation to failure modeling capabilities as it implies that common cause failures are omitted from this analysis. The method is still an active research topic, with a review provided in [40].

Both FTA and FMEA studies can be integrated to run recursively to combine their strengths [41]. One such approach is defined in [42] as the "Bouncing Failure Analysis", where they,

- Identify hazards
- Prepare an FTA diagram to find minimal cut sets of basic events that lead to the hazardous event
- Run an FMEA study to inspect the failure modes' effects on the system operation and the hazardous events
- Update and exploit the FTA diagram to analyze interactions between failure modes, which is normally not possible with FMEA.

FMEA can be applied at any conceptual level, including components, subsystems, systems, system functions, and even system environment and interactions. During the development process of the system-of-interest, a functional FMEA was conducted on the basis of the system definition as shown in the process flow diagram given in **Figure 6.7**. For the specified system functions, the failure modes and the potential root causes of failure modes were identified. Following that, the impact of the failure mode on the system missions as described in the system definition are assessed. Here, the complete definition of missions, scenarios, and use-cases of the system form the basis of impact evaluations. This emphasizes the importance of the systems engineering approach to safety-related equipment development, as the holistic approach to system definition enables the efficient and accurate evaluation of the local interactions.

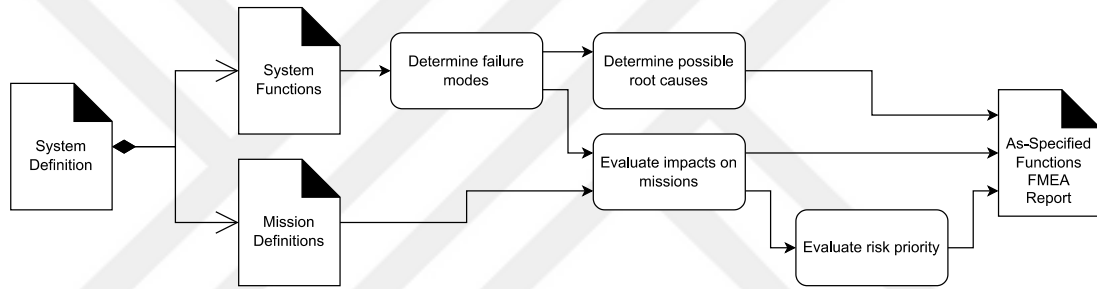


Figure 6.7: Simplified process flow model for conducting FMEA study on the as-specified system function.

Going a step further from the FMEA study, we can include a field for the safety-criticality of failure modes, as shown in **Figure 6.8**. The resulting method is called the Failure Modes, Effects, and Criticality Analysis (FMECA), which is standardized in [24]. In **Table 6.2**, an excerpt from the FMECA study done on the as-specified digital input and output system functions is presented. We find that almost all the failure modes of these functions lead to dangerous situations. This finding justifies the specification of safety-related requirements regarding these functions.

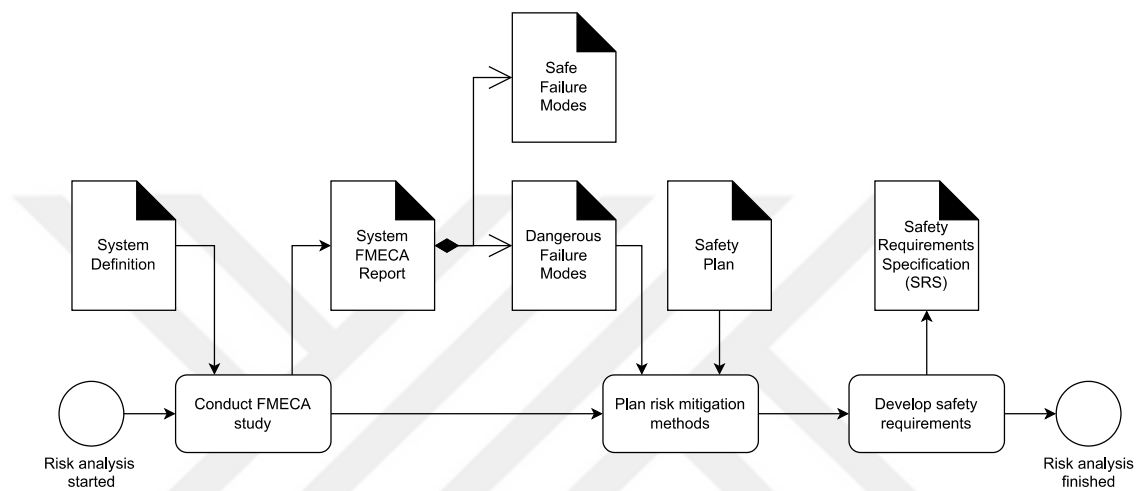


Figure 6.8: Simplified process flow model for using FMECA to develop safety requirements.

Table 6.2: An excerpt from the FMECA study done on the as-specified system functions.

#	System Function	Failure Modes	Root Causes	Impact on Missions	Safe / Dangerous
1.1	Measure the general-purpose digital input pins	Signal is always measured as digital low	- Short-circuit - Measurement circuit faults	External system signals are misinterpreted, resulting in inappropriate reactions.	Dangerous
1.2		Signal is always measured as digital high	- Short-circuit - Measurement circuit faults		Dangerous
1.3		Signal is measured with temporary glitches	- Electromagnetic interference - Supply fluctuations - System power-on glitches - Radiation-induced soft errors on semiconductors		Dangerous
1.4		Measured signal is random noise	- Electromagnetic interference - Loss of conductive path to the signal source - Measurement circuit faults		Dangerous
1.5		Source signal is faster than the measurement bandwidth	- Incorrect interface specification - External system failures	Signal transitions could be missed, resulting in inappropriate or rather a lack of reactions.	Dangerous

Table 6.2 (continued)

#	System Function	Failure Modes	Root Causes	Impact on Missions	Safe / Dangerous
2	Provide digital signals to the general-purpose digital output pins	Signal output is stuck-at digital low	- Short-circuit - Output circuit faults - Signal driving logic faults - Radiation induced soft errors on semiconductors	The signal is always de-energized. Under system interface assumptions, this steers the system under control to a safe state.	Safe ¹
2.1		Signal output is stuck-at digital high	- Short-circuit - Output circuit faults - Signal driving logic faults - Radiation induced soft errors on semiconductors	The signal is always energized. Under system interface assumptions, this compromises the safety integrity.	Dangerous
2.2		Signal output with temporary glitches	- Electromagnetic interference - Supply fluctuations - System power-on glitches - Radiation induced soft errors on semiconductors	Signal could transition without intent, leading to unexpected behavior.	Dangerous
2.3		Signal transition is not monotonic	- Electromagnetic interference - Supply fluctuations		Dangerous
2.4		Output signal is random noise	- Output circuit faults - Signal driving logic faults		Dangerous

Table 6.2 (continued)

#	System Function	Failure Modes	Root Causes	Impact on Missions	Safe / Dangerous
2.5	Provide digital signals to the general-purpose digital output pins	Output signal is limited by the driver bandwidth	- Design error - Output circuit faults	Signal could miss an intended transition, leading to unexpected behavior.	Dangerous
2.6		Signal impedance increase	- Output circuit faults - Copper faults	Signal could transition without intent, or miss an intended transition, leading to unexpected behavior.	Dangerous ²

NOTES

¹ Assuming digital low corresponds to a safe state.

² Safe for small deviations, dangerous for deviations in the range of load impedance, may be safe for very high deviations (high impedance) given that the load pulls bias current.

7. SYSTEM REQUIREMENTS

Requirements are the formal definition statements that are used to describe the acceptability conditions of any piece of engineering artefact [43]. These conditions may originate from different domains, such as from stakeholders, interactions, interfaces, performance, safety, or the system behavior, and more. Requirements can also be interpreted as interfaces that translate needs and targets between different domains [43]. For example, stakeholder requirements map stakeholder needs to system requirements, system requirements map system functions to implementation, and so on.

An individual requirement should conform to a set of characteristics on its own, such as necessity, appropriateness, unambiguity, and verifiability [44], [45]. Further, sets of requirements should conform to an additional set of characteristics as a group, such as completeness, consistency, feasibility, and validity [45].

A system with the same functionality can be implemented in multiple ways. Some functions interact with others or may be realized by the same system element. Therefore, adding or removing a function can have larger effects on the overall system implementation. Hence, to arrive at an efficient implementation, the development team first needs to define a set of definitions that characterize the system behavior and performance, which are then converted to logical solutions and finally to the implemented system. In the context of systems engineering, this set of behavior and performance definitions are called the system requirements.

Stakeholder requirements formally capture the stakeholder needs, which may be lacking in accurate definitions, consistency, and feasibility [23]. System requirements refine and satisfy stakeholder requirements, which connect stakeholder needs and system functions to architecture and the design, and finally to system verification and validation, as shown in **Figure 7.1**.

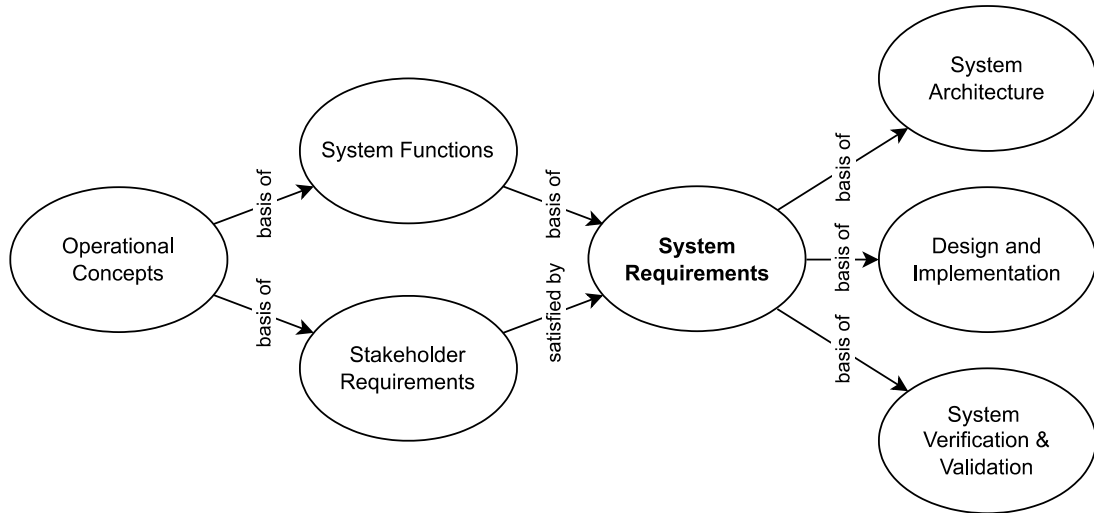


Figure 7.1: Concept map relating system requirements to other system concepts.

In order to prevent communication errors and consequent design errors, system requirement statements follow some predefined set of rules regarding their structure and the information that they convey. There are multiple rulesets, guides, and frameworks for requirements writing [44], [45], one such framework is the EARS ruleset [46] popular due to its simplicity and flexibility [47], [48]. The EARS ruleset presents the following functional requirement sentence template, where the fields in bold are mandatory and the others are optional depending on the structure of the required system behavior:

*While <pre-condition(s)>, when <trigger>, **the <system name> shall <system response>**.*

Within this structure, state-driven behavior is modeled with "*while <pre-condition(s)>*" field, whereas the event-driven behavior is modeled with the "*when <trigger>*" field. Another class of requirements that are always relevant to the system, called the ubiquitous requirements, only use the "*<system> shall <system response>*" part of the template. The system may also have unwanted behavior requirements that are closely related to safety-related requirements. The unwanted behaviors are modeled using the following template:

***If <pre-condition(s)> <trigger>, then the <system name> shall <system response>**.*

While this template is semantically identical to the event-driven requirement template, the authors of the EARS ruleset suggest that the distinction in wording helps with distinguishing between nominal system operations and unwanted or failure mode system operations [46].

In **Table 7.1**, select statements from the system requirements specification of the system-of-interest are presented.

Table 7.1: Example functional system requirements for the system-of-interest.

Reqt. ID	Reqt. Statement	Type	Rationale
SR1	The system shall accommodate the interface requirements as stated in IRS-001.	Interface	
SR2	While the system is powered-on, when carrier board de-asserts power enable, the system shall power-off after $T_{safeshutoff} \pm 10\%$ amount of time.	Functional	Enables the system to carry out shut off operations safely.
SR3	While the system is powered on, when the carrier board de-asserts power enable, the system shall store application-specific information in non-volatile memory within $0.9 \times T_{safeshutoff}$ amount of time.	Functional	Stores cause of power-off for counteractions, investigations, and repairs.
SR4	When the system reset is issued, the system shall store the source of reset in non-volatile memory.	Functional	Stores cause of system reset for counteractions, investigations, and repairs.
SR5	While in application mode, the system shall carry out application-specific logic as described by the bitstream stored on the system's non-volatile memory.	Functional	The system carries out the as-stored "payload" application logic.
SR6	While the system is powering on, the system shall apply power sequencing.	Functional	Required to overcome glitches and hardware faults.

Table 7.1 (continued)

Req. ID	Req. Statement	Type	Rationale
SR6.1	The system shall monitor internal supply voltages.	Functional	Enables feedback for power sequencing.
SR6.2	When an internal supply voltage is in the nominal range for T_PG amount of time, the system shall enable the next supply generator in the power sequence.	Functional	If a supply does not reach the nominal range, the power sequence halts.
SR7	The system shall be compatible with MXM 2.0 edge connector.	Constraint	Design constraint
SR7.1	The system shall have mounting holes on the PCB as defined in the technical drawing TD-001.	Constraint	Design constraint
SR8	The system shall operate with 5V $\pm$ 10% supply voltage.	Performance	Recommended application conditions
SR9	The system performance shall not be permanently affected by supply voltages up to 7.5V.	Performance	Absolute maximum conditions
SR10	The system shall have a propagation delay of at most Tpd from the digital input ports to the digital outputs ports.	Performance	Required for analysis of systems employing the system-of-interest.

7.1. System Safety Requirements

System safety requirements, commonly referred to as safety requirements, are used to identify, document, convey, and track the resolutions to identified system hazards.

Safety requirements can be categorized as follows [49]:

- Functional safety requirements, which are concerned with functional hazards that emerge from the interaction of the system and its environment, which may originate from misuse, failures, and more. An example is brake systems failing.
- Technical safety requirements, which are concerned with technical hazards that are intrinsic to the system, originating from its design and implementation. An

example is the risk of electrical shock if high voltages exist within the system boundaries.

- Contextual safety requirements, which are concerned with hazards originating from operation and maintenance and are in the responsibility of system operators. An example is the preflight checklist used in aviation.

Identification of hazards begins as early as the development of system concepts begins and continues throughout the system development life cycle. The definition of safety requirements begins right after the initial system definition is developed and continues throughout the system development life cycle. As the system definition or specification is updated, new hazards may be identified, which may result in new safety requirements if the associated risks are deemed intolerable. Safety requirements can be documented separately or joined with the system requirements. However, in case they are documented in the same document or model database, safety requirements should be tagged accordingly [49].

Functional safety requirements differ from the non-safety related functional system requirements in their purpose and scope. Any requirement that is concerned with a safety-related function is a functional safety requirement. However, unlike non-safety related system requirements, functional safety requirements also capture the system characteristics and behavior under failure modes of safety-related functions. The overall relations between functions, hazards, and safety requirements are given in the block diagram in **Figure 7.2**. Finally, a subset of functional safety requirements for the system-of-interest are presented in **Table 7.2**.

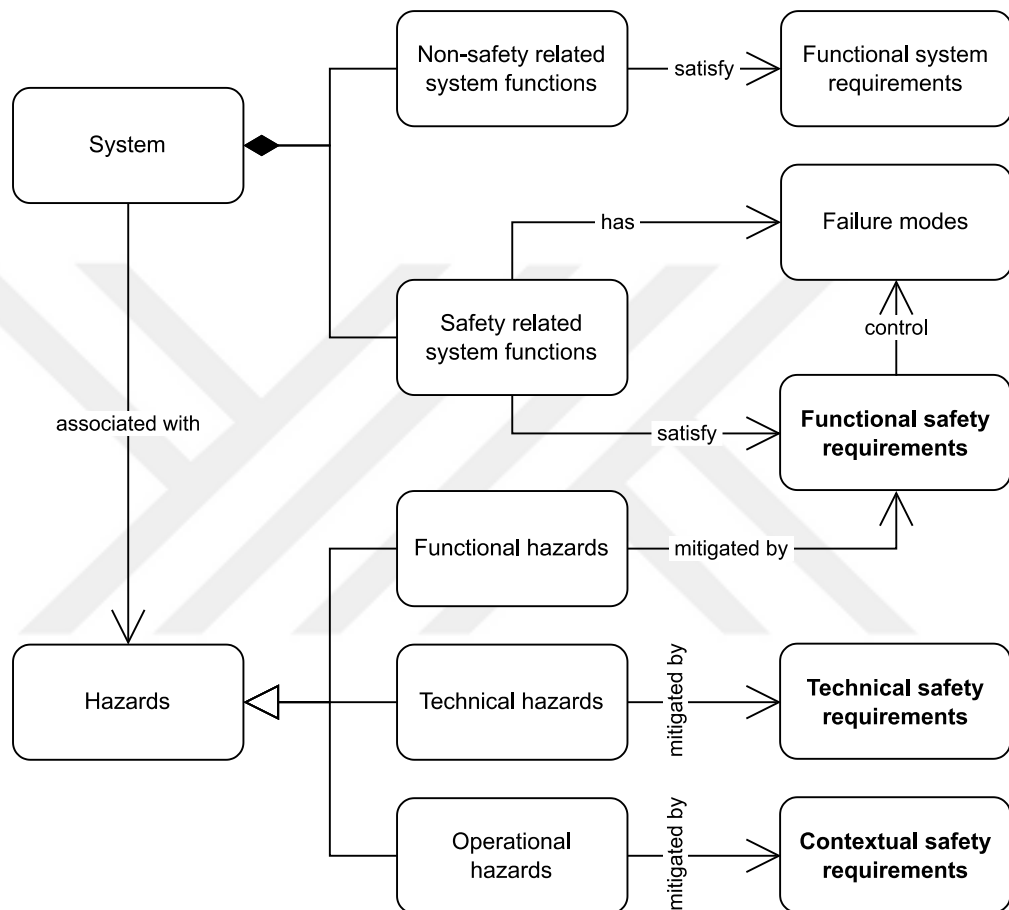


Figure 7.2: The relation between the system, hazards, functions, and safety requirements.

Table 7.2: Example functional safety requirements for the system-of-interest.

Req. ID	Req. Statement	Rationale
FSR1	While the system is powered-off, the system shall provide high-impedance on all digital outputs	It is assumed that the carrier board will steer the system under control to a known safe state.
FSR2	While the application logic is not running the system shall provide high-impedance on all digital outputs	It is assumed that the carrier board will steer the system under control to a known safe state.
FSR3	While a supply has been enabled for at least T_PON duration when a supply voltage is out of nominal range for at least T_VF duration the system shall initiate power failure mode.	An undervoltage or overvoltage condition has occurred, which compromises integrity of digital circuitry operations.
FSR3.1	While the system is in power failure mode the system shall apply power down sequence.	Power down sequence should be controlled to prevent output glitches.
FSR3.2	When the system enters power failure mode the system shall store failure information in non-volatile memory.	Failure information is useful for failure-specific counter-actions and repairability.
FSR3.3	While the system is in power failure mode when all voltages are turned off the system shall enter power failure recovery mode.	Fault conditions may be transient, in which case reapplying power may get the system running without manual operations.
FSR3.3.1	While the system is in power failure recovery mode the system shall attempt to carry out power up sequence once every Thiccup period.	Fault conditions may be transient, in which case reapplying power may get the system running without manual operations.
FSR3.3.2	While the system is in power failure recovery mode when all voltages has stayed in nominal ranges for at least T_PG duration the system shall enter application mode.	The fault condition has been removed, system power is nominal, system can continue operation.

8. SYSTEM ARCHITECTURE

A system architecture is a conceptual solution given a system context and a set of system requirements. A system architecture may have multiple views, such as the structural view, the functional view, the information view, the subsystem or module view, and so on [50]. Each view represents a constrained subset of concerns about the system, which helps in managing system complexity [51].

An architecture may be specified in various ways, ranging from textual descriptions to formal modelling languages such as UML and SysML, and to architecture modeling frameworks such as TOGAF, DoDAF, and so on [52], [53] .

System architecture development process starts with determining the relevant measures of performance (MoP) of the system-of-interest. Multiple candidate architectures are developed and evaluated in terms of system requirements coverage and measures of performance. Finally, a multi-criteria decision analysis method of choice, such as weighted ranking matrix, is used to objectively calculate the best-fit architecture for the current system definition.

The scientific and engineering literature, similar and related systems, subject matter expert opinions, and finally the designers' intuition form inputs to the candidate architecture development process. The resulting architecture must be able to accommodate all system requirements. Architecture elements and system requirements should be traceable in either direction.

8.1. FPGA SoM Measures of Performance

In this subsection, we identify applicable measures of performance for the system under development. These measures of performance will be used as optimization criteria for the evaluation of candidate system architectures. Some measures of performance are qualitative in nature. To enable mathematical operations, we quantify qualitative measures of performance by means of categorization. The identified measures of performance for the system-of-interest are listed and described in **Table 8.1**. The assembly costs were omitted from P1 with the assumption that all components

used in the implementation will be surface-mount technology (SMT) components and that the cost of assembly will be more or less the same. The major items that affect P1 are programmable elements (which cost in the order of \$10s~\$100s) and specialized integrated-circuits (which cost in the order of ~\$10).

Table 8.1: Measure of performance definitions for the system-of-interest.

MoP ID	Short Name	Description	System Performance Implications
P1	BOM cost	The PCB manufacturing and component price per product unit	Lower BOM cost is desired for competitiveness. Lower BOM cost is better.
P2	Startup time	Duration after power-up the system takes to start executing the application program	Lower startup time is required for certain applications. Lower startup time is better.
P3	Configuration vulnerability	Vulnerability of the FPGA program stored in or transferred to configuration registers	Lower configuration vulnerability leads to higher assurance that the FPGA is functioning as-intended. Lower vulnerability is better.
P4	Diagnostic coverage (DC)	The fraction of system failures the system can detect, but not necessarily localize or correct	Higher diagnostic coverage provides better assurance that the system operates as expected. Higher DC is better.
P5	Safe failure fraction (SFF)	Fraction of system element failure modes that lead to detected or undangerous failures.	Higher safe failure fraction means lower rate of dangerous failures. Includes the fraction of undangerous failures on top detectable failures (DC). Higher SFF is better.

8.2. Architecture Space Exploration

In this section, the architecture space will be explored to define and derive feasible candidate system architectures. To systematically examine the architecture space, the system architecture is divided into multiple dimensions and plausible solutions for

each dimension are defined. Here, plausibility of suggested solutions were derived guided by the literature and the designers' expertise. Following that, the combination of compatible solutions in terms of measures of performance described in the previous section were evaluated.

Both the functional and the physical architectures can be modeled visually as block diagrams, or in table form as an N^2 charts. N^2 charts are formed by filling the diagonals of a $N \times N$ matrix with N system elements (which may be physical or functional, depending on the type of the architecture), placing the interface from the element E_i to element E_j in the (i, j) element, as shown in **Table 8.2**. This way, N^2 charts are particularly useful for visualizing strongly cohesive and loosely coupled system elements so that the subsystem boundaries can be designed to have minimal interface complexity [54]. Simpler interfaces result in better maintainability and evolvability, which are desired in system architectures.

Table 8.2: N^2 representation of a system architecture consisting of three system elements shown in bold, and the interfaces between them

E1	$E1 \rightarrow E2$	$E2 \rightarrow E3$
$E1 \leftarrow E2$	E2	$E2 \rightarrow E3$
$E1 \leftarrow E3$	$E2 \leftarrow E3$	E3

8.3. Functional Architecture

The functional architecture describes the functions a system carries out, the interactions between the system functions, and the functional interface of the system with its environment.

The system needs and high-level system requirements were used as inputs to develop the functional architecture prototype model, as shown in **Figure 8.1**. While some functions are required to fulfill system requirements, some functions are not functionally mandatory. However, those functions still serve to improve the system characteristics and therefore play an important role in fulfilling the dependability targets. Further, even the mandatory functions can have one of multiple capability

subsets. For example, power sequence control is a mandatory function; however, power sequence control characteristics can be improved by adding analog voltage monitoring to achieve better system safety at the cost of increased system complexity and possibly worse system availability. Similarly, FPGA operation monitoring can be done with one of various monitoring criteria, with each stricter criterion resulting in better system safety at the cost of worse system availability.

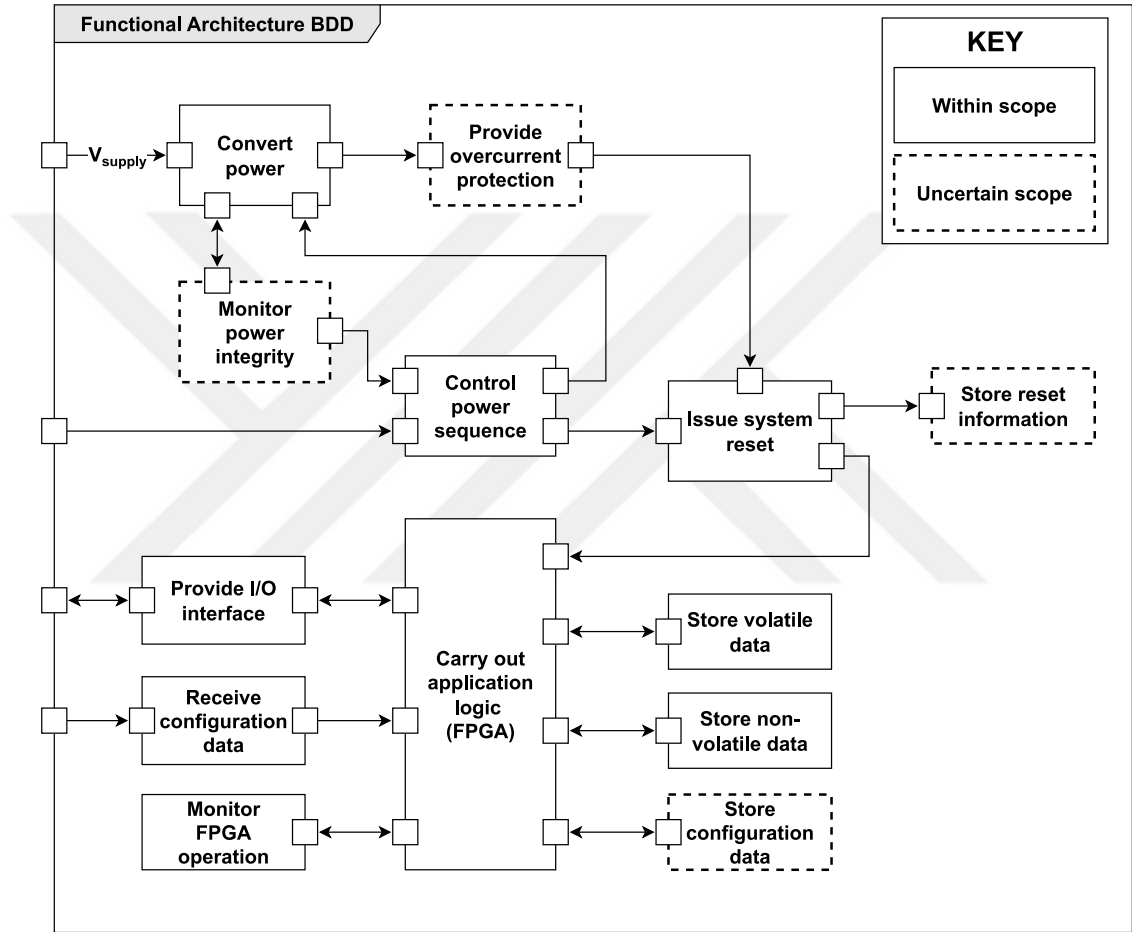


Figure 8.1: Functional architecture prototype block definition diagram.

There can be an intractable number of candidate functional architectures. To make the problem of functional architecture definition problem approachable, we divide the architecture definition into multiple dimensions, as presented in **Table 8.3**. Each dimension represents a choice in the architecture definition, as shown in **Figure 8.2**. Each point on a dimension represents a potentially valid solution to a subset of system requirements. A set constructed as the aggregate of a single point along each architecture dimension, combined with the architecture prototype defined in **Figure**

8.1, represents a candidate architecture. It should be noted that the architecture block diagram model presented in **Figure 8.1** is implementation-free with the exception of the inclusion of an FPGA, which is a design constraint originating from the business case of this project. Functions presented here can be implemented in multiple ways, and the possible implementation techniques will be explored in the following section.

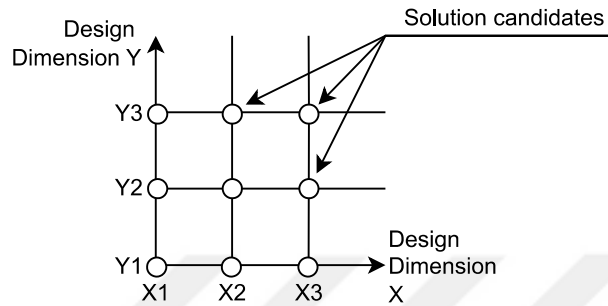


Figure 8.2: Solution candidates in two-dimensional design space.

Table 8.3: Trade study dimensions on system functions.

Arch. Dim.	Solution ID	Solution Description	Remarks
Power supply protection functions	FS1.1	No power protection	No guarantees that the supply voltages will be in the nominal range.
	FS1.2	Under-voltage lockout Overvoltage protection	Better assurance of system operations since supply voltages are verified to be in the nominal range.
	FS1.3	FS1.2 + overcurrent protection	Better availability through protection against temporary shorts and IC latch-ups
	FS1.4	FS1.3 + safe shut-off	Better safety since the system can steer to a known safe state before shut-off
Main FPGA watchdog	FS2.1	Minimum frequency constrained watchdog	Low diagnostic coverage [55]
	FS2.2	Frequency band constrained watchdog	Medium diagnostic coverage [55]

	FS2.3	Challenge-response based watchdog	High diagnostic coverage [55]
--	-------	-----------------------------------	-------------------------------

Table 8.3 (continued)

Arch. Dim.	Solution ID	Solution Description	Remarks
System reset block	FS3.1	Resets issued by the power circuit, the watchdog, and the carrier board	Bare minimum
	FS3.2	FS3.1 + reset issued by the FPGA	Better availability by enabling self-resetting to overcome system soft errors
	FS3.3	FS3.2 + reset issuers saved in non-volatile memory	Better maintainability through enabling easier troubleshooting Better safety through enabling tailored failure counteractions
Power sequencing	FS4.1	Time-programmed power sequencing	Bare minimum
	FS4.2	FS4.1 + 3-level voltage monitoring feedback	The 3 levels are undervoltage, nominal, and overvoltage. Better reliability and safety through power fault handling
	FS4.3	FS4.2 + analog monitoring	Analog monitoring may enable the detection of unwanted oscillations
I/O interface	FS5.1	Provide I/O with no diagnostic or redundancy	Bare minimum
	FS5.2	Provide I/O with online monitoring	Medium diagnostic coverage [55], given that railway applications are high-demand mode per [14]
	FS5.3	Provide I/O with redundancy from the same chip	High diagnostic coverage is limited by common cause failures [55]
	FS5.4	Provide I/O with antivalent signals	High diagnostic coverage [55]

Table 8.3 (continued)

Arch. Dim.	Solution ID	Solution Description	Remarks
Memory subsystem	FS6.1	Provide non-redundant storage	Bare minimum
	FS6.2	Provide dual modular redundant memory chips with shared interfaces	High diagnostic coverage is limited by common cause failures [55]
	FS6.3	Provide dual modular redundant memory chips with independent interfaces	High diagnostic coverage [55]. Very expensive in terms of I/O usage.
	FS6.4	Provide data-redundant storage through the use of error-correcting codes	Medium diagnostic coverage [55]. Increased startup time since memory space needs to be initialized.
	FS6.5	Provide data and address fault-tolerant storage through the use of address-aware error-correcting codes [56]	High diagnostic coverage is achievable depending on the encoding scheme employed [56]. Increased startup time since memory space needs to be initialized.

Following the identification of functional architecture trade dimensions in **Table 8.3**, the effects of solutions candidates on measures of performance are characterized and are presented in **Table 8.5**. The symbols used in **Table 8.5** are described in the key provided in **Table 8.4**. We use the comparative evaluations of effects of design options on measures of performance as the justification for architecture decisions. For each design option presented in **Table 8.3**, points associated with evaluations for effects on measures of performance are summed up to obtain an overall score. Then, the design option with the highest score from each design dimension is selected, as presented in **Table 8.6**. Where the highest scores are tied between multiple solutions, the solution with the lowest system complexity is chosen to reduce the overall life cycle costs. After that, the combination of best-fit design options is evaluated as a system, with the aim to discover interactions between design elements and the resulting emergence originating from the nature of systems.

Table 8.4: Key and scores for interpreting **Table 8.5**.

Symbol	Meaning	Points
--	Strong negative effect on MoP	-2
-	Weak negative effect on MoP	-1
o	No effect on MoP baseline for comparison	+0
b	MoP baseline for comparison	+0
+	Weak positive effect on MoP	+1
++	Strong positive effect on MoP	+2

Table 8.5: Effects of functional architecture trade dimensions on measures of performance.

Solution Dimension	Solution	BOM cost	Startup time	Config. vuln.	DC	SFF	Points
Power supply protection functions	FS1.1 No power protection	b	b	b	--	b	-2
	FS1.2 Under-voltage lockout Overvoltage protection	-	o	++	b	+	+2
	FS1.3 FS1.2 + overcurrent protection	-	o	++	+	+	+3
	FS1.4 FS1.3 + safe shut-off	-	o	++	+	++	+4
Main FPGA watchdog	FS2.1 Minimum frequency constrained watchdog	b	b	b	b	b	+0
	FS2.2 Frequency band constrained watchdog	-	o	o	+	+	+1
	FS2.3 Challenge-response based watchdog	--	o	o	++	++	+2

Table 8.5 (continued)

Solution Dimension	Solution	BOM cost	Startup time	Config. vuln.	DC	SFF	Points
System reset block	FS3.1 Resets issued by the power circuit, the watchdog, and the carrier board	b	b	b	b	b	+0
	FS3.2 FS3.1 + reset issued by the FPGA	o	o	o	o	+	+1
	FS3.3 FS3.2 + reset issuers saved in non-volatile memory	-	o	o	++ ¹	+	+2
Power sequencing	FS4.1 Time-programmed power sequencing	b	b	b	b	b	+0
	FS4.2 FS4.1 + 3-level voltage monitoring feedback	-	+	+	+	+	+3
	FS4.3 FS4.2 + analog monitoring	--	+	+	++	+	+3
I/O interface	FS5.1 Provide I/O with no diagnostic or redundancy	b	b	b	b	b	+0
	FS5.2 Provide I/O with online monitoring	-	o	o	+	+	+1
	FS5.3 Provide I/O with redundancy from same chip	-	o	o	+	+	+1

Table 8.5 (continued)

Solution Dimension	Solution	BOM cost	Startup time	Config. vuln.	DC	SFF	Points
I/O interface	FS5.4 Provide I/O with antivalent signals	-	o	o	+	++	+2
Memory subsystem	FS6.1 Provide non-redundant storage	b	b	b	b	b	+0
	FS6.2 Provide dual modular redundant memory chips with shared interfaces	-	o	o	+	+	+1
	FS6.3 Provide dual modular redundant memory chips with independent interfaces	--	o	o	++	++	+2
	FS6.4 Provide data-redundant storage through use of error-correcting codes	o	-	o	+	+	+1
	FS6.5 Provide data and address fault tolerant storage through use of address-aware error-correcting codes	o	-	o	++	++	+3

NOTES:

1. Specific power failure events can be saved to memory and specific counter-actions are possible after system reset condition is removed.

Table 8.6: The selected functional architecture solution elements.

Architecture Dimension	Select Solution	Possible Interactions and Design Challenges
Power supply protection functions	FS1.4: Under-voltage lockout Overvoltage protection Overcurrent protection Safe shut-off	Safe shut-off only functions if power-off is requested by the carrier board. It does not cover sudden power losses.
Main FPGA watchdog	FS2.3: Challenge-response based watchdog	The watchdog should enable the FPGA to initialize before issuing resets. The watchdog should enable over-the-air programming of the FPGA.
System reset block	FS3.3: Resets issued by the power circuit, the watchdog, the carrier board, and the FPGA Reset issuers saved in non-volatile memory	An independent interface to the non-volatile memory is required to access its data while the system has power or FPGA failures.
Power sequencing	FS4.2: Time-programmed power sequencing 3-level voltage monitoring feedback	Power sequencing should work in tandem with the safe shutoff feature.
I/O interface	FS5.4: Provide I/O with antivalent signals	Could be provided in the carrier board for only safety-related signals.
Memory subsystem	FS6.5: Provide data and address fault-tolerant storage through the use of address-aware error-correcting codes	Software-defined encoding scheme enables tuning for specific application needs.

8.4. Physical Architecture

In order to carry out the functions selected in the previous section and presented in **Table 8.6**, there are multiple physical architecture candidates, each with their own advantages. In this section, feasible and viable physical architecture candidates will be identified, evaluated, and finally, the best-fit physical architecture candidate will be selected to be implemented.

The most promising contenders for the physical architecture are presented in **Table 8.7** where each line represents a set of system elements sufficient to inform the design and implementation processes of the system development life cycle. From these candidates, PA3 was chosen for the basis of implementation due to its unique advantages and the availability of SRAM-based FPGA product alternatives.

Table 8.7: Assessment overview of physical architecture candidates.

#	Physical Architecture Candidate	Advantages	Limitations
PA1	SRAM-based FPGA NVM for configuration data Discrete gates for power and reset logic Challenge-response watchdog IC Sequence controlled by dedicated ICs	Low BOM cost due to SRAM-based FPGA.	Bad supportability due to specialized challenge-response watchdog ICs with sparse substitutes. Low configurability due to discrete gates and specialized ICs. FPGA is prone to soft errors. Power sequence interruptions can be difficult to implement.
PA2	Flash-based FPGA Discrete gates for power and reset logic Challenge-response watchdog IC Sequence controlled by dedicated ICs	Flash-based FPGA is immune to soft errors.	Low configurability due to discrete gates and specialized ICs. Power sequence interruptions can be difficult to implement. Higher BOM cost due to Flash-based FPGA

Table 8.7 (continued)

#	Physical Architecture Candidate	Advantages	Limitations
PA3	SRAM-based FPGA NVM for configuration data Secondary programmable element for power and reset logic, challenge-response watchdog	Low BOM cost due to SRAM-based FPGA. Secondary programmable element results in great reconfigurability and supportability but also higher system complexity.	FPGA is prone to soft errors.
PA4	Flash-based FPGA Secondary programmable element for power and reset logic, challenge-response watchdog	Flash-based FPGA is immune to soft errors. Secondary programmable element results in great reconfigurability and supportability but also higher system complexity.	Higher BOM cost due to Flash-based FPGA

9. DESIGN AND IMPLEMENTATION

On the basis of the system requirements and system architecture specifications as developed earlier, the system elements can be designed and implemented. The design process can include modeling of events and interactions to refine and clarify the requirement and architecture specifications forming the basis of the design. While designing and implementing, new design constraints or hazards can be identified, which could lead to modifications in the system definition, the system requirements, or the system architecture.

In this section, design details of the system under development will be presented and described, which include functional block diagrams, structural block diagrams, sequence diagrams, and risk analysis results. As the risk analysis methods were already described in Section 6, only the results and the remarks about the risk analyses will be presented. Implementation of the system, which includes the complete electronic circuit schematics and the BOM tables, is out of the scope of this thesis, as the main contribution of this thesis is the design process, not the resulting design itself.

The physical architecture developed in Section 8 consists of:

- an SRAM-based FPGA named the “Functional FPGA” to carry out application-specific logic,
- a NVM for storing configuration data in the FPGA, and
- a secondary programmable element named the “safety FPGA” for implementing power and reset logic, challenge-response watchdog

The implementation including these elements also result in the inclusion of new physical elements, such as a temperature sensor to assure that the system is operating within the design limits, and a basic watchdog IC to monitor the operation of the safety FPGA.

Furthermore, the functional architecture developed in Section 8 consists of the following functions, which the system must allocate:

- Provide under-voltage lockout (UVLO)

- Provide overvoltage protection (OVP)
- Provide overcurrent protection
- Provide safe shutoff mechanism where the power cut-off is delayed to enable the system to safely end its operation
- Provide challenge-response based watchdog monitoring to the functional FPGA
- Accept resets issued by the power circuit, the watchdog, the carrier board, the functional FPGA and the safety FPGA
- Store reset information in non-volatile memory
- Control the power-up and power-down sequences using 3-level voltage monitoring feedback
- Provide data and address fault tolerant storage through use of address-aware error-correcting codes

In **Figure 9.1**, the simplified overall physical architecture of the system, including the interfaces, is presented as a block definition diagram.

In **Figure 9.2**, the block definition diagram depicting the power conversion circuit is provided. The power is received from the carrier board. However, due to functional architecture specifications, the supply power should be cut off from the system in case of undervoltages, overvoltages and overcurrents. To enable this, a power switch was designed to be in series with the supply port. However, to control the power switch, the auxiliary supply voltage must always be on. Therefore, the auxiliary supply converter sub-circuit was designed to provide its own protection functions and be always on as long as the supply voltages are within the working range of this converter. The safety FPGA takes power enable signal and the supply monitor status signals as inputs, and generates a power enable signal controlling the power switch, and separate converter enable signals, each controlling a voltage converter. Finally, the regulated supply voltages are connected to respective loads in the system-of-interest.

In **Figure 9.3**, the physical architecture that is implemented within the safety FPGA through the use of hardware description language code is presented. The watchdog is shown to be challenge-response based; however, the mechanism of the watchdog can

be defined for the particular needs of specific applications. The blocks responsible for power control are shown as power controller and power-on reset (POR) generation blocks. The sources of reset are conditioned and combined in the reset generation block. The output of the reset generation block is the system reset, which also triggers a register unit within the safety FPGA. This register unit can comprise volatile, non-volatile, or even both, where the register block could be implemented using volatile buffers that are stored in non-volatile memories for faster response times. Finally, the contents of the reset register block can be accessed through the SMBus port that is only used for safety-related functions.

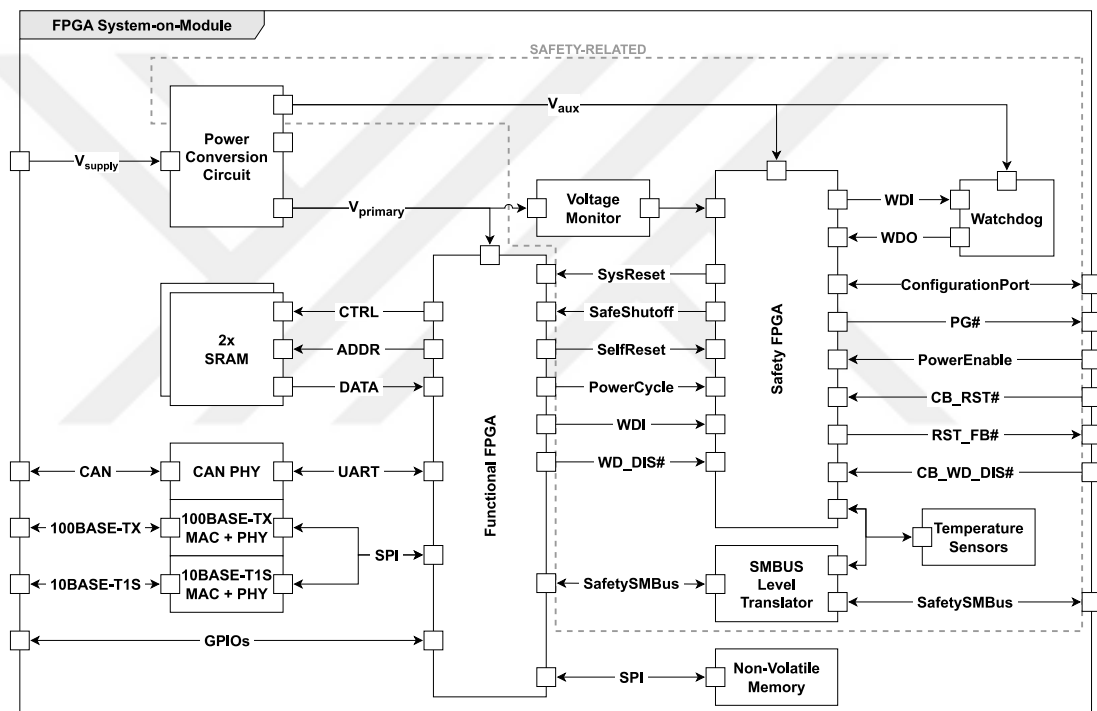


Figure 9.1: Block definition diagram depicting the overall physical architecture of the system after design.

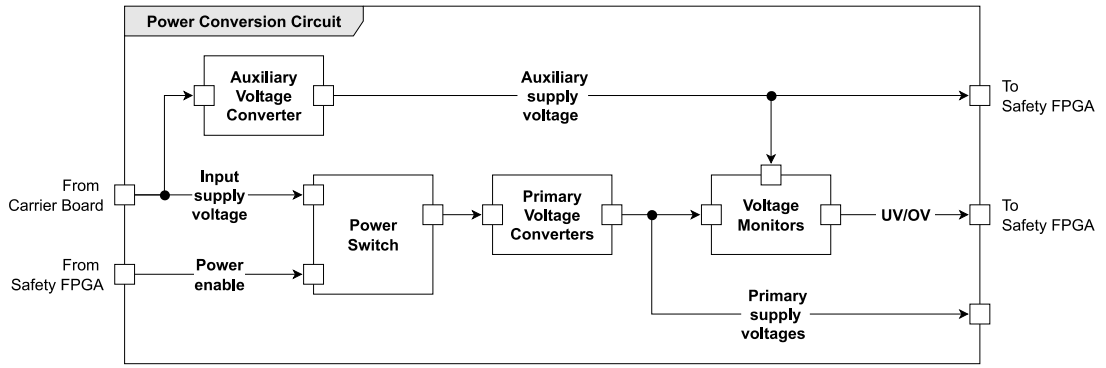


Figure 9.2: Block definition diagram depicting the power conversion circuit.

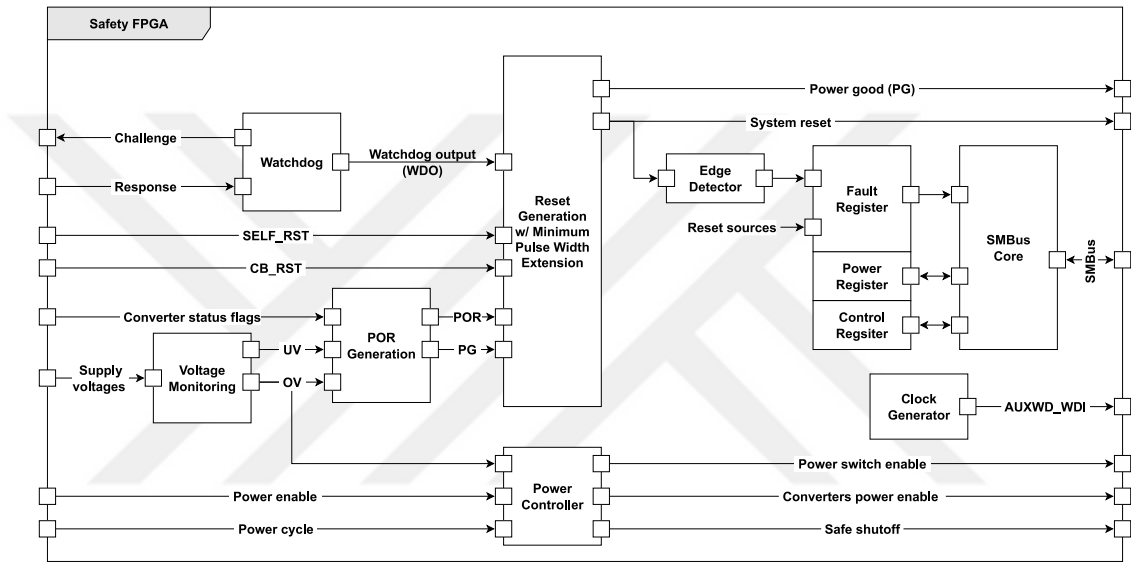


Figure 9.3: Block definition diagram depicting the hardware-description language defined structure implemented in the safety FPGA.

The implementation of the safety FPGA has intricate interactions and edge scenarios. To better understand and communicate the expected operations of the safety FPGA, the sequence diagrams given in **Figure 9.4**, **Figure 9.5**, **Figure 9.6** were developed as a basis for implementation. Sequence diagrams are a part of the Unified Modeling Language used to represent the interactions and information flow between instances of elements through the time dimension [31]. The modeled use case begins at the top of the actor lifelines.

In **Figure 9.4**, the interactions between the power, the converters and the power switch, the safety FPGA and its related watchdog, and finally the functional FPGA during the power-up sequencing scenario are presented. When the module receives power, the auxiliary supply converter powers on the safety FPGA. The safety FPGA initializes

and starts resetting the watchdog to prevent getting reset by it. Then, the safety FPGA enables the power switch and the voltage converters and verifies that the voltages are in the nominal ranges. Finally, the challenge-response transactions between the safety FPGA and the functional FPGA begin following the initialization of the functional FPGA.

In **Figure 9.5**, the interactions between the aforementioned system elements during the safe shut-off scenario are presented. Here, the carrier board deasserts the power enable signal while the system is operational. The safety FPGA immediately signals the functional FPGA that safe shut-off is in progress. After waiting for a prescribed amount of time determined by the application-specific information, the safety FPGA starts disabling converters and the power switch, following the appropriate power-down sequence of events. After issuing each disable request, the safety FPGA verifies that the voltages actually do go out. After all the primary supply voltages are cut, the safe shutoff operation is done, but the safety FPGA keeps resetting its watchdog to prevent getting reset, since the supply to the safety FPGA is always on.

In **Figure 9.6**, the interactions between the carrier board, the auxiliary converter, the safety FPGA, and its watchdog during JTAG programming and debugging scenario are presented. First, the carrier board powers up the module, and the safety FPGA is always on, as usual. Then, a JTAG master device originating from the carrier board connects to the safety FPGA through the IEEE 1149.1 interface. While this connection is underway, the watchdog of the safety FPGA is disabled through the dedicated pin exposed to the edge connector, which may be connected to a JTAG connector on the carrier board. This prevents the watchdog from resetting the safety FPGA while debugging or programming. After the programming or the debugging is done, the watchdog disable signal is released and the system continues operations as usual.

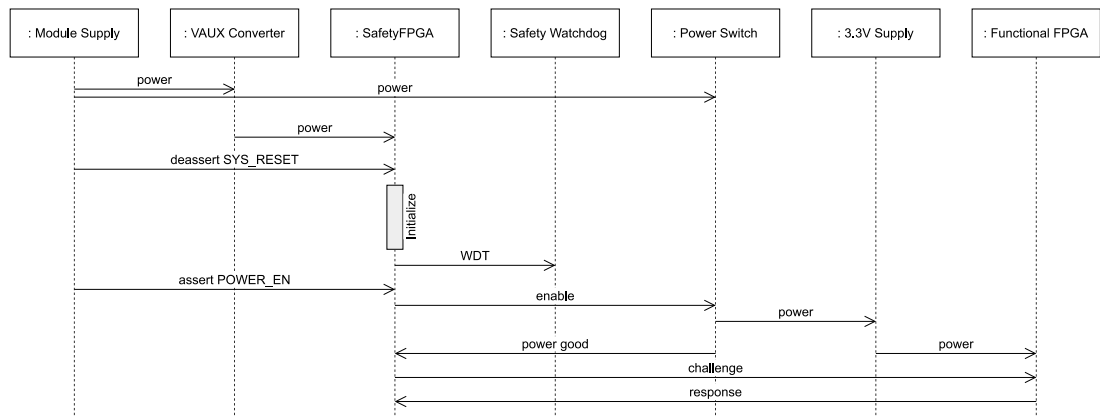


Figure 9.4: Sequence diagram depicting the power-up sequencing scenario events and interactions.

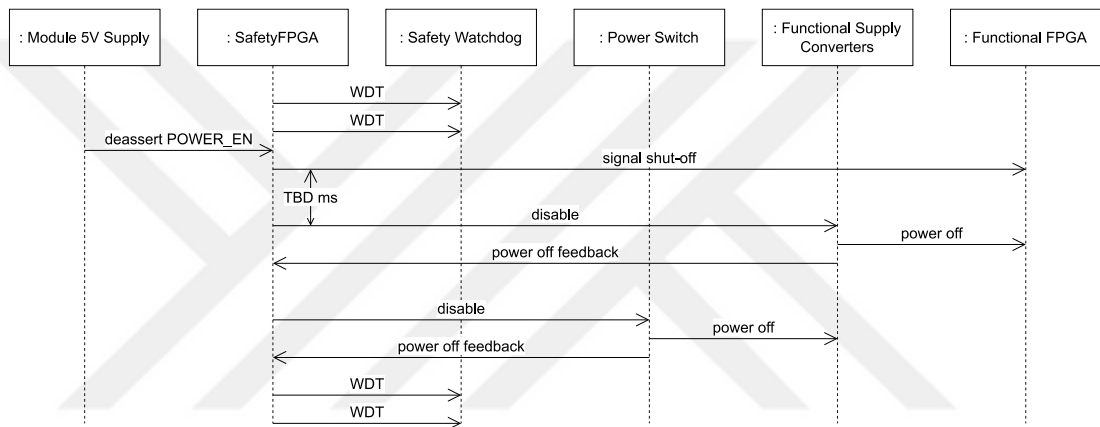


Figure 9.5: Sequence diagram depicting the safe shut-off scenario events and interactions.

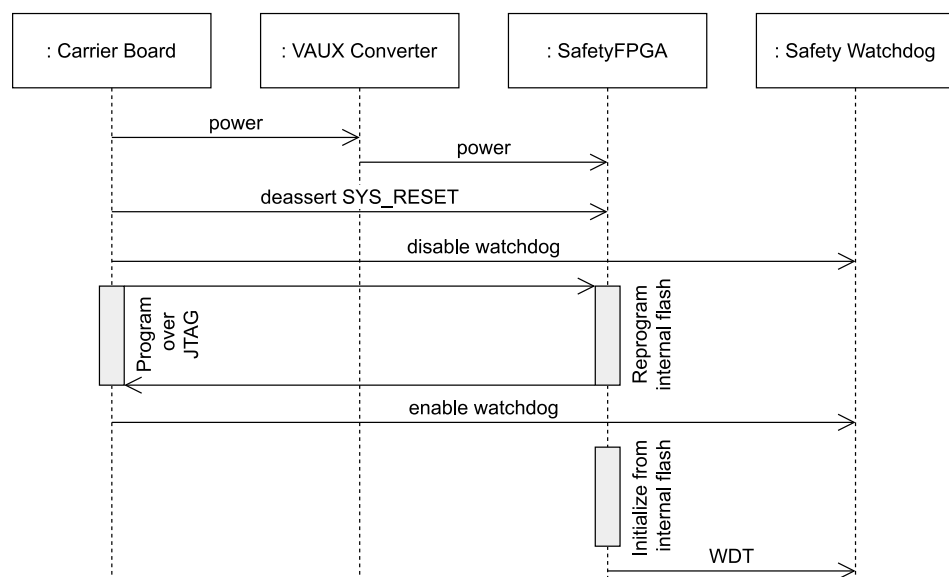


Figure 9.6: Sequence diagram depicting the safe shut-off scenario events and interactions.

9.1. As-Designed Risk Analysis

A risk analysis consisting of a Failure Modes, Effects and Diagnostic Analysis (FMEDA) was conducted at the design level before implementation. This analysis primarily includes the system elements that may be expected to lead to dangerous outcomes. This assumption is tolerable at this stage, as the aim of this analysis is to find out the biggest contributors to system risks according to the Pareto rule where generally 20% of causes result in 80% of outcomes [57]. The results of the FMEDA study are given in **Table 9.1**. We see that, as-designed system can detect 91.92% of failures, which combined with the intrinsically safe failure modes, leads to a safe failure fraction of 94.18%. Further, assuming that all the failure modes of the system can be accurately identified, the obtained safe failure fraction combined with the lack of hardware fault tolerance results in the theoretical appraisal limit of SIL 3 for the safety-related system functions as stated in the system definition. Despite the limitation imposed by the hardware fault tolerance and the safe failure fraction, the probability of failure on demand parameter permits appraisals up to SIL 4. However, the actual appraisal is limited by the lowest limit, which is SIL 3 for the system-of-interest.

Therefore, we conclude that the as-designed system enables certification up to SIL 3, given that the rest of the technical and management-related aspects of the project are carried out according to relevant functional safety standards. Finally, two or more instances of the system-of-interest can be operated in parallel to obtain SIL 4 ratings for specific applications, given that the common cause failures between these instances are properly mitigated.

Table 9.1: The results of the FMEDA study done on the design level for most complex components.

Parameter	Symbol	Value	Unit	Certifiable Up To	Remarks
Hardware fault tolerance	HFT	0	–		
Mean time to failure	MTTF	11.6×10^6	h		Higher is better for reliability and availability.
Diagnostics coverage	DC	91.92%	–		Better DC leads to better SFF.
Safe failure fraction	SFF	94.18%	–	SIL 3	SIL rating is given assuming Type A SIS.
Safe failure rate	λ_S	24.2×10^{-9}	$1/h$		
Dangerous failure rate	λ_D	62.2×10^{-9}	$1/h$		
Dangerous detected failure rate	λ_{Dd}	57.2×10^{-9}	$1/h$		Considered safe due to diagnostic capabilities
Probability of Failure on Demand	PFD	5.0×10^{-9}	$1/h$	SIL 4	Dangerous, undetected failures

10. DISCUSSION AND FUTURE WORKS

In this thesis, a comprehensive view into the development processes of safety-related systems was presented. The development concepts are presented from multiple aspects, including but not limited to project management, risk management, systems engineering design, and requirements engineering aspects. Furthermore, the thesis describes that for some aspects, simultaneous and complementary approaches with top-down and bottom-up analysis techniques are necessary for efficiently arriving at the correct systems. The intention of this thesis is to provide a reference or guide into conducting the preliminary design iterations of safety-related, or even non-safety-related systems. For developing safety-related systems, organizations need to provide assurance against systematic errors, which also necessitate that different roles with sufficient amount of independence from other developers and managers should be defined and carried out.

Regardless of the systematic capability requirements of safety-related systems that pose organizational constraints, a safety-related project can be kicked off with a tailored set of processes, such as those described in this thesis. The result of these analyses will inevitably require iterations on the system development processes, as was emphasized in Section 3.

The resulting system architecture was described in Section 8, which was transformed into designs in Section 9. While describing the obtained design, several block diagrams and sequence diagrams explaining the system elements and interactions were presented. Finally, the results of a Failure Modes, Effects and Diagnostic Analysis (FMEDA) were presented and along with the discussion of implications.

The system presented in this thesis is planned to be used in various projects in TUBITAK RUTE. Before that, the deliverables and the outcomes of this work should be fed into the formal safety-related system development life cycle processes, leading to certification according to [14]. The deliverables of this study should provide a basis for more accurate system definition, system requirements, and system architecture and should finally lead to an elegant solution.

REFERENCES

- [1] P. Li, R. Xue, S. Shao, Y. Zhu, and Y. Liu, "Current state and predicted technological trends in global railway intelligent digital transformation," *RS*, vol. 2, no. 4, pp. 397–412, Nov. 2023, doi: 10.1108/RS-10-2023-0036.
- [2] D. Möller *et al.*, "Emerging Technologies in the Era of Digital Transformation: State of the Art in the Railway Sector:," in *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2022, pp. 721–728. doi: 10.5220/0011141900003271.
- [3] P. Singh, M. A. Dulebenets, J. Pasha, E. D. R. S. Gonzalez, Y.-Y. Lau, and R. Kampmann, "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges," *IEEE Access*, vol. 9, pp. 91427–91461, 2021, doi: 10.1109/ACCESS.2021.3091550.
- [4] *Functional safety of electrical/ electronic/programmable electronic safety-related systems - part 1: general requirements*, IEC 61508-1:2010, International Electrotechnical Commission, 2010.
- [5] *Road vehicles. Functional safety*, ISO 26262:2018, International Organization for Standardization.
- [6] *Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling*, EN 50129:2018/AC:2019-04, European Committee for Electrotechnical Standardization, Apr. 26, 2019. Accessed: Apr. 02, 2025. [Online]. Available: <https://standards.iteh.ai/catalog/standards/clc/02a923fa-b7a3-42fb-84d3-c2da3936e2dc/en-50129-2018-ac-2019-04>
- [7] *Railway Applications - Requirements for software development*, EN 50716:2023, European Committee for Electrotechnical Standardization, Nov. 17, 2023. Accessed: Apr. 02, 2025. [Online]. Available: <https://standards.iteh.ai/catalog/standards/clc/32d1d924-9bfd-477e-adf4-de6e3358be77/en-50716-2023>
- [8] O. Uğuz and İ. Çiçek, "Design of an FPGA-Based System-on-Module for Safety-Critical Applications," in *2024 5th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Nov. 2024, pp. 1–6. doi: 10.1109/CIEES62939.2024.10811198.
- [9] I. Kuon, R. Tessier, and J. Rose, "FPGA Architecture: Survey and Challenges," *FNT in Electronic Design Automation*, vol. 2, no. 2, pp. 135–253, 2008, doi: 10.1561/10000000005.
- [10] C. Bernardeschi, L. Cassano, and A. Domenici, "SRAM-Based FPGA Systems for Safety-Critical Applications: A Survey on Design Standards and Proposed Methodologies," *J. Comput. Sci. Technol.*, vol. 30, no. 2, pp. 373–390, Mar. 2015, doi: 10.1007/s11390-015-1530-5.

- [11] Y. Sered and Y. Reich, “Standardization and modularization driven by minimizing overall process effort,” *Computer-Aided Design*, vol. 38, no. 5, pp. 405–416, May 2006, doi: 10.1016/j.cad.2005.11.005.
- [12] European Commission, “The Technical Specification for Interoperability - Control Command and Signalling TSI,” *Official Journal of the European Union*, vol. 66, no. Commission Implementing Regulation (EU) 2023/1695, pp. 380–560, Aug. 2023.
- [13] *Railway applications - Rolling stock - Electronic equipment*, EN 50155:2021, European Committee for Electrotechnical Standardization. Accessed: Apr. 02, 2025. [Online]. Available: <https://standards.iteh.ai/catalog/standards/clc/b34c4e1c-23a6-4c2c-91ee-85f2f8ee9e3a/en-50155-2021>
- [14] *Railway applications - the specification and demonstration of reliability, availability, maintainability and safety (rams) - part 1: generic rams process*, EN 50126-1:2017, European Committee for Electrotechnical Standardization, 2017.
- [15] *International Electrotechnical Vocabulary (IEV) - Part 192: Dependability*, IEC 60050-192:2015, International Electrotechnical Commission, Mar. 23, 2016. Accessed: Feb. 16, 2025. [Online]. Available: <https://webstore.iec.ch/en/publication/21886>
- [16] *Functional safety of electrical/ electronic/programmable electronic safety-related systems*, IEC 61508:2010, International Electrotechnical Commission, 2010.
- [17] E. Vanem, “Ethics and fundamental principles of risk acceptance criteria,” *Safety Science*, vol. 50, no. 4, pp. 958–967, Apr. 2012, doi: 10.1016/j.ssci.2011.12.030.
- [18] E. B. Abrahamsen and T. Aven, “Why risk acceptance criteria need to be defined by the authorities and not the industry?,” *Reliability Engineering & System Safety*, vol. 105, pp. 47–50, Sep. 2012, doi: 10.1016/j.res.2011.11.004.
- [19] E. Honour, “11.4.2 Systems Engineering Return on Investment,” *INCOSE International Symposium*, vol. 20, no. 1, pp. 1422–1439, 2010, doi: 10.1002/j.2334-5837.2010.tb01150.x.
- [20] *Systems and software engineering — system life cycle processes*, ISO/IEC/IEEE 15288, Institute of Electrical and Electronics Engineers (IEEE), 2023.
- [21] N. B. Ruparelia, “Software development lifecycle models,” *SIGSOFT Softw. Eng. Notes*, vol. 35, no. 3, pp. 8–13, May 2010, doi: 10.1145/1764810.1764814.
- [22] International Council on Systems Engineering (INCOSE), *SYSTEMS ENGINEERING HANDBOOK - A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES*, 5th Edition. John Wiley & Sons Ltd., 2023.
- [23] *Railway applications. The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). Generic RAMS process*, EN 50126-1:2017+A1:2024, European Committee for Electrotechnical Standardization.

- [24] *Failure modes and effects analysis (FMEA and FMECA)*, IEC 60812:2018, International Electrotechnical Commission. Accessed: Apr. 01, 2025. [Online]. Available: <https://webstore.iec.ch/en/publication/26359>
- [25] *Functional safety of electrical/ electronic/programmable electronic safety-related systems - part 4: definitions and abbreviations*, IEC 61508-4:2010, International Electrotechnical Commission, 2010.
- [26] W. Mohammad, A. Elomri, and L. Kerbache, “The Global Semiconductor Chip Shortage: Causes, Implications, and Potential Remedies,” *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 476–483, Jan. 2022, doi: 10.1016/j.ifacol.2022.09.439.
- [27] *Mechanical structures for electronic equipment — Dimensions of mechanical structures of the 482,6 mm (19 in) series*, EN 60297-3-108:2015, European Committee for Electrotechnical Standardization, Jan. 31, 2015.
- [28] R. D. Arnold and J. P. Wade, “A Definition of Systems Thinking: A Systems Approach,” *Procedia Computer Science*, vol. 44, pp. 669–678, 2015, doi: 10.1016/j.procs.2015.03.050.
- [29] R. Plate and M. Monroe, “A Structure for Assessing Systems Thinking”.
- [30] S. R. Hirshorn, *Expanded Guidance for NASA Systems Engineering. Volume 1: Systems Engineering Practices*. 2016. Accessed: Mar. 29, 2025. [Online]. Available: <https://ntrs.nasa.gov/citations/20170007238>
- [31] T. Weilkiens, *Systems engineering with SysML/UML: Modeling, Analysis, Design*, 1. Aufl. Heidelberg: dpunkt.verl, 2006.
- [32] J. E. Bartolomei, D. E. Hastings, R. De Neufville, and D. H. Rhodes, “Engineering Systems Multiple-Domain Matrix: An organizing framework for modeling large-scale complex systems,” *Systems Engineering*, vol. 15, no. 1, pp. 41–61, Mar. 2012, doi: 10.1002/sys.20193.
- [33] E. Carroll and R. Malins, “Systematic Literature Review: How is Model-Based Systems Engineering Justified?,” SAND--2016-2607, 1561164, 627724, Mar. 2016. doi: 10.2172/1561164.
- [34] B. Kaiser, B. Monajemi, D. Kusche, and H. Schulte, “Systematic design and validation of degradation cascades for safety-relevant systems,” in *Safety and Reliability – Theory and Applications*, High Tatras Mountains, Tatranské Matliare, Slovak Republic: CRC Press, Jun. 2017, pp. 527–527. doi: 10.1201/9781315210469-452.
- [35] P. Fischer *et al.*, “The bystander-effect: a meta-analytic review on bystander intervention in dangerous and non-dangerous emergencies.,” *Psychological bulletin*, vol. 137, no. 4, p. 517, 2011.
- [36] N. J. Bahr, *System safety engineering and risk assessment: a practical approach, second edition*. CRC Press, 2018.

- [37] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Computer Science Review*, vol. 15–16, pp. 29–62, Feb. 2015, doi: 10.1016/j.cosrev.2015.03.001.
- [38] M. Yazdi *et al.*, "Fault tree analysis improvements: A bibliometric analysis and literature review," *Quality and Reliability Engineering International*, vol. 39, no. 5, pp. 1639–1659, 2023, doi: 10.1002/qre.3271.
- [39] W. Vesely, *Fault tree handbook with aerospace applications*. Washington, D. C.: NASA Office of Safety and Mission Assurance, 2002.
- [40] C. Spreafico, D. Russo, and C. Rizzi, "A state-of-the-art review of FMEA/FMECA including patents," *Computer Science Review*, vol. 25, pp. 19–28, Aug. 2017, doi: 10.1016/j.cosrev.2017.05.002.
- [41] J. F. W. Peeters, R. J. I. Basten, and T. Tinga, "Improving failure analysis efficiency by combining FTA and FMEA in a recursive manner," *Reliability Engineering & System Safety*, vol. 172, pp. 36–44, Apr. 2018, doi: 10.1016/j.ress.2017.11.024.
- [42] Z. Bluvband, R. Polak, and P. Grabov, "Bouncing failure analysis (BFA): the unified FTA-FMEA methodology," in *Annual Reliability and Maintainability Symposium, 2005. Proceedings.*, Alexandria, VA, USA: IEEE, 2005, pp. 463–467. doi: 10.1109/RAMS.2005.1408406.
- [43] *Systems and software engineering. Life cycle processes. Requirements engineering*, ISO/IEC/IEEE 29148, International Organization for Standardization, 2018.
- [44] S. R. Hirshorn, L. D. Voss, and L. K. Bromley, *NASA Systems Engineering Handbook*. 2017.
- [45] Michael Ryan and Lou Wheatcraft, "Guide to Writing Requirements," International Council on Systems Engineering, INCOSE-TP-2010-006-04, Jul. 2023. [Online]. Available: https://www.incose.org/docs/default-source/working-groups/requirements-wg/gtwr/incose_rwg_gtwr_v4_040423_final_drafts.pdf
- [46] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy Approach to Requirements Syntax (EARS)," in *2009 17th IEEE International Requirements Engineering Conference*, Atlanta, Georgia, USA: IEEE, Aug. 2009, pp. 317–322. doi: 10.1109/RE.2009.9.
- [47] A. Mavin and P. Wilkinson, "Big Ears (The Return of 'Easy Approach to Requirements Engineering')," in *2010 18th IEEE International Requirements Engineering Conference*, Sydney, Australia: IEEE, Sep. 2010, pp. 277–282. doi: 10.1109/RE.2010.39.
- [48] A. Mavin Mav and P. Wilkinson, "Ten Years of EARS," *IEEE Softw.*, vol. 36, no. 5, pp. 10–14, Sep. 2019, doi: 10.1109/MS.2019.2921164.
- [49] *Railway applications. The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). Systems approach to safety*, EN

50126-2:2017+A1:2024, European Committee for Electrotechnical Standardization, 2024.

[50] M. Ozkaya and F. Erata, “A survey on the practical use of UML for different software architecture viewpoints,” *Information and Software Technology*, vol. 121, p. 106275, May 2020, doi: 10.1016/j.infsof.2020.106275.

[51] L. E. Hart, “Introduction to model-based system engineering (MBSE) and SysML,” in *Delaware valley INCOSE chapter meeting*, Ramblewood Country Club Mount Laurel, New Jersey, 2015.

[52] M. Amissah and H. A. H. Handley, “A process for DoDAF based systems architecting,” in *2016 Annual IEEE Systems Conference (SysCon)*, Apr. 2016, pp. 1–7. doi: 10.1109/SYSCON.2016.7490649.

[53] S. Bondar, J. C. Hsu, A. Pfouga, and J. Stjepandić, “Agile digital transformation of System-of-Systems architecture models using Zachman framework,” *Journal of Industrial Information Integration*, vol. 7, pp. 33–43, Sep. 2017, doi: 10.1016/j.jii.2017.03.001.

[54] O. Becker, J. B.- Asher, and I. Ackerman, “A method for system interface reduction using N2 charts,” *Systems Engineering*, vol. 3, no. 1, pp. 27–37, 2000, doi: 10.1002/(SICI)1520-6858(2000)3:1<27::AID-SYS2>3.0.CO;2-I.

[55] *Functional safety of electrical/ electronic/programmable electronic safety-related systems - part 2: requirements for electrical/electronic/ programmable electronic safety-related systems*, IEC 61508-2:2010, International Electrotechnical Commission, 2010.

[56] O. Uğuz and İ. Çiçek, “Address-Aware Error Correcting Codes: A New Approach to Memory Addressing Fault Tolerance,” *Journal of Supercomputing*, In review.

[57] V. Spasojević-Brkić, M. Perišić, Z. Veljković, M. Misita, and M. Damjanović, “Pareto analysis in bulldozer’s failures and stoppages risk management,” *Industrija*, vol. 50, no. 2, pp. 37–49, 2022, doi: 10.5937/industrija50-41907.

BIOGRAPHY

Oğuzhan Uğuz completed his B.Sc. degree in mechanical engineering in Istanbul Technical University in 2019, and B.Sc. degree in control and automation engineering in Istanbul University in 2020 as part of the double-majors program. He has worked in startups for 3 years before joining TUBITAK Institute of Rail Transportation Technologies (RUTE). He has been working at RUTE for 3 years as an electronics systems engineer.



PUBLICATIONS AND PRESENTATIONS FROM THE THESIS

- [1] O. Uğuz and İ. Çiçek, “Design of an FPGA-Based System-on-Module for Safety-Critical Applications,” in *2024 5th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Nov. 2024, pp. 1–6. doi: 10.1109/CIEES62939.2024.10811198.
- [2] O. Uğuz and İ. Çiçek, “Address-Aware Error Correcting Codes: A New Approach to Memory Addressing Fault Tolerance,” *Journal of Supercomputing*, In review.

