Prof. Dr.-Ing. A. Siggelkow

Spezifikation: How To

Hochschule Ravensburg-Weingarten
University of Applied Sciences

B

Contents

Chapter 1

General

Before we the definitions really start, it should be stated how the **specifications** evolved. The classic approach was to specify the hardware first (figure 1.1 left), then build the hardware and finally specify the software running on the hardware and write it. Then the system should run. This approach was time consuming and because of the time-to-market also a cost factor. So, it changed (figure 1.1 right), at the beginning, the entire system will be specified and the hardware and software will be developed in parallel. It is obvious, that there are hard requirements to the new specification, all aspects of interaction of SW and HW must be considered right at the beginning.

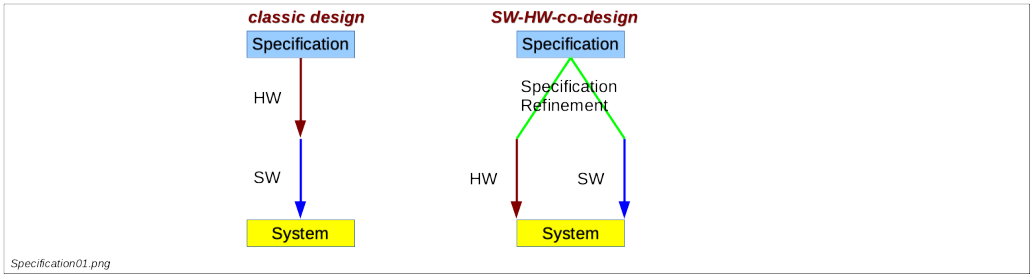


Figure 1.1: Specification: Old and Modern

Figure 1.2 goes a little bit more into the details. What could be seen first is, that before starting with the specification, **requirements** for the product must be defined. The requirements should bring the ideas, being in the brain of the customer only, to a sheet of paper. This paper could then be a part of the contract between the customer and the designer of the product.

After this, the requirements will be expanded to a more detailed specification, a system specification. This can be used to start the parallel design process.

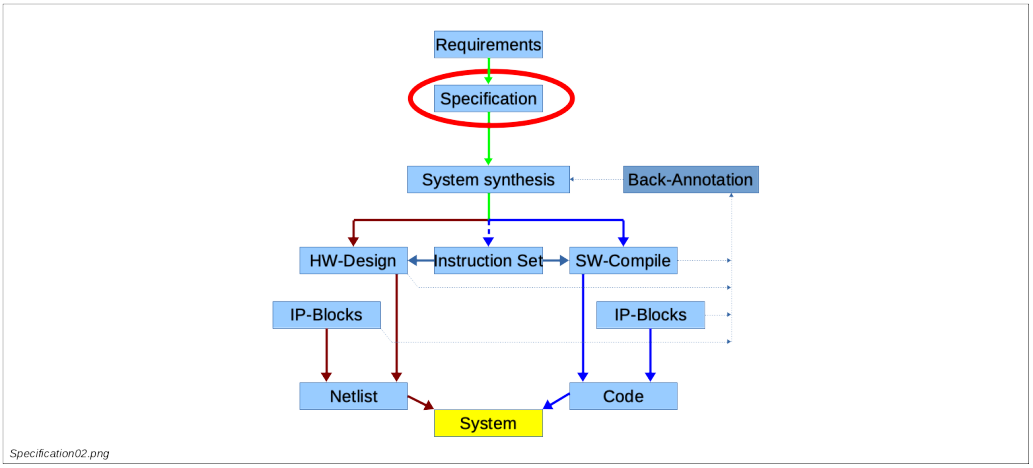


Figure 1.2: Specification: Complete View

Chapter 2

Requirement and Specification

The design of CPS and IoT systems is a complex process. In principle it is the well known divide and conquer approach, like in the classical design flow. It has to be broken down into sub-tasks. Every sub-task must have a well defined interface to the other.

The starting point of a system design is the knowledge in the heads of some people, at least the customer and/or system architects. Those ideas must be written down into a **requirements sheet**, this requirements sheet will be signed by the customer and the design company. After the requirements analysis has been finalized, the **specification** will be derived on base of the requirements.

2.1 Requirements

There is no common definition of **requirements**. There is a requirements analysis according [8]:

- IEEE
- CMMI (Capability Maturity Model Integration)
- Volere
- IIBA (International Institute of Business Analysis)
- and may be more

Because of the electrical and computer science nature of ES, CPS and IoT, I describe the IEEE approach. But nevertheless, in principle, all approaches are similar.

According IEEE, **requirements engineering** consists of:

Requirements elicitation: Collect all possible requirements of the system.

Requirements analysis: The analysis of the whole set of requirements should lead to a common understanding of the product.

Requirements specification: The requirements will be documented and explained - specified.

Requirements validation: The whole set of requirements must be checked against each other to detect holes or conflicts in the description.

Elicitation:- The **elicitation and analysis** of the requirements must follow some criteria:

Complete: The requirements must be described explicit. It should not be the case, that the customer has implicit assumptions of the system.

Clearly defined/bound: Precise definitions help to avoid misunderstandings between the customer and the developer.

Comprehensively described: The customer and the developer should be able to read the requirements with the same understanding.

Atomic: There should be only one requirement described within one section.

Identifiable: Every requirement needs its own ID.

Uniformly documented: The requirements should be documented with the same structure in just one database.

Verifiable: Every requirement needs **acceptance criteria** with which the product can be tested against the requirements. **Test cases** will be derived from the acceptance criteria.

Traceable: It must be traceable, that a requirement has been fulfilled (forwards). Also must be verified, that every implemented functionality has its requirement (backwards).

Consistent: The requirements are not allowed to contradict each other.

Structure and classify:- After the elicitation, the requirements must be **structured and classified**. This increases the clarity. And this increases the understanding of the relations between the requirements. Criteria for this are:

Dependent: Requirements must be tested if one requirements depends on another, or will be independent.

Belonging together: Requirements, which belongs logically or technically to an other requirement, should be grouped.

Role-based: Every user group will have its own sight on the requirements.

Functional requirements: Those explains, what the system should be able to do. Requirements for the function of the system.

Non-functional requirements: Those explains, how the system should bring ist function (Quality, performance, reliability). Requirements which handles standardization, legal, social, medical, etc. aspects.

Quality assurance:- Next, the **quality assurance** of the requirements needs to be done:

Correct: The requirements must be consistent.

Feasible: The requirements must be feasible.

Necessary: Requirements not required by the customer are no requirements.

Prioritized: Identify the importance of the requirements for the product.

Usable: Even having just a port of the system, it should be a productive system.

Relevance for Students Projects:- Important aspects for the requirements of **small projects planned with students** are:

Unitary: The requirement addresses one and only one thing.

Complete: The requirement is fully stated in one place with no missing information.

Consistent: The requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation.

Atomic: The requirement is atomic, i.e., it does not contain conjunctions. E.g., "The postal code field must validate American and Canadian postal codes" should be written as two separate requirements: (1) "The postal code field must validate American postal codes" and (2) "The postal code field must validate Canadian postal codes".

Unambiguous: The requirement is concisely stated without recourse to technical jargon, acronyms (unless defined elsewhere in the Requirements document), or other esoteric verbiage. It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation. Vague subjects, adjectives, prepositions, verbs and subjective phrases are avoided. Negative statements and compound statements are avoided.

Specify Importance (Prioritized): Many requirements represent a stakeholder-defined characteristic the absence of which will result in a major or even fatal deficiency. Others represent features that may be implemented if time and budget permits. The requirement must specify a level of importance.

Verifiable: The implementation of the requirement can be determined through basic possible methods: inspection, demonstration, test (instrumented) or analysis (to include validated modeling & simulation, a Test-Bench element is required for every requirement).

Identifiable: Every requirement needs its own ID.

2.2 Specification and Modeling

2.2.1 Requirements (for the Specification)

Specifications for ES provide **models** of the system under design. A good definition is:

Definition 2.2.1. [9]: “A model is a simplification of another entity, which can be a physical thing or another model. The model contains exactly those characteristics and properties of the modeled entity that are relevant for a given task. A model is minimal with respect to a task if it does not contain any other characteristics than those relevant for the task”.

Models are described in languages, the languages should have some features:

Hierarchy: Physical objects are normally too complex to be understood by human beings. Hierarchy and abstraction are key mechanisms helping to solve this dilemma. There are:

- **Behavioral hierarchies:** Behavioral hierarchies contain objects necessary to describe the system behavior. Such objects are: states, events, output signals and more.
- **Structural hierarchies:** Structural hierarchies describe how systems are composed of physical components. Such components are: processors, memories, actuators, sensors. Processors have: registers, multiplexers, adders. Multiplexers have: gates. Gates have: transistors.

Component-based design: If a system is build of single components with a known behavior, it must be possible to predict the behavior of the system.

Concurrency: Real systems operate concurrently, so the system model needs to reflect this concurrency.

Synchronization and communication: Components must be able to communicate and synchronize.

Timing behavior: Many ES are real-time systems. The timing requirements must be specified. Unfortunately, time is not a simple quantity to be modeled. But some aspects must be considered:

- Measure the **elapsed time** (timer).
- **Delay a process** for a specified time (control over interrupts).
- Specify **timeouts** (again, interrupts are a problem)
- Specify **deadlines** and **schedules** (interrupts, caches, pipelines).

State-oriented behavior: FSM are a good mechanism to model reactive systems. The classical FSM needs to updated, to be able to support model timing and hierarchy.

Event handling: Reactive systems are event driven.

Exception-oriented behavior: Exceptions must be considered, they will be needed but there are problems with it.

Presence of programming elements: The specification of programming elements needs to be merged into the specification of systems.

Executability: Specifications could differ from the ideas in people's heads, so an executable specification has its advantages.

Support for the design of large systems: In SW design, object oriented programming is a means to handle large systems. This is needed in the specification of large HW/SW systems, also.

Domain-specific support: Different application domains need different aspects in the specification.

Readability: People (customer) needs to read and understand the specification, on the other hand, the specification should be executable which means, it must be computer readable. There must be a way, to test the written specification against the executable specification.

Portability and flexibility: The specification should be independent of the HW platform.

Termination: Termination processes needs to be specified.

Support for non-standard I/O devices: It is self explaining.

Non-functional properties: Fault tolerance, size, power consumption, etc.

Support for the design of dependable systems: It is self explaining.

No obstacles to the generation of efficient implementations: It is self explaining.

Appropriate model of computation (MoC): The von-Neumann model of sequential computation has problems with respect to concurrent CPS (timing, multithreading and deadlocks, etc.).

2.2.2 Models of Computation

A system is build of many components. In general, **computation** and **communication** will be differentiated.

- Components: ... and the organization of computations in such components: Procedures, processes, functions, finite state machines, etc.

- Communication protocols: Methods for communication between components (asynchronous message passing, rendezvous based, etc.).

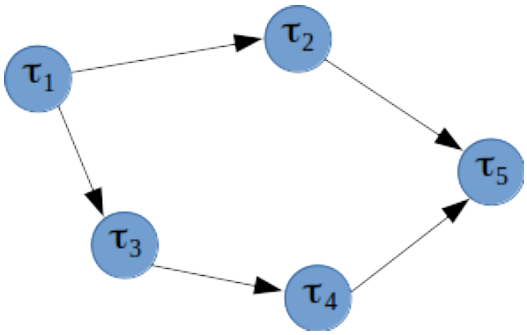


Figure 2.1: Dependence Graph

Components:

The relation between components may be visualized by means of **graphs** (process or task). Nodes in the graph (task graphs or process networks) represent components performing computations. Edges represent relations between components. E.g., one relation is the causality, this results in a **dependence graph** (fig. 2.1).

Definition 2.2.2. [5]: A dependence graph is a directed graph $G = (\tau, E)$ where τ is the set of **vertices** or **nodes** and E is the set of **edges**. $E \subseteq \tau \times \tau$ imposes a relation on τ.
If $(\tau_1, \tau_2) \in E$ with $\tau_1, \tau_2 \in \tau$, then τ_1 is called an **immediate predecessor** of τ_2 and τ_2 is called an **immediate successor** of τ_1. Suppose E^* is the transitive closure of E.
If $(\tau_1, \tau_2) \in E^*$, then τ_1 is called a **predecessor** of τ_2 and τ_2 is called a **successor** of τ_1.

A dependence graph is a special form of a **task graph**. Tasks graphs allow more information:

1. **Timing information:** Arrival times, deadlines, periods, execution times, etc.

2. **Distinction between different types of relations** between computations: Precedence relations, communication described by the edges, I/O nodes (fig. 2.2, partially filled circles are I/O edges).

3. **Exclusive access to resources:** Some computations may require exclusive access to to resources, e.g., I/O, memory space for communication.

4. **Periodic schedules:** Distinguish between a task and ist execution (**job**), such graphs used in signal-processing can be periodic and also infinite (fig. 2.3).

5. **Hierarchical graph nodes:** The granularity depends on the application, but if hierarchy is needed, it can also be shown in the graph (fig. 2.4).

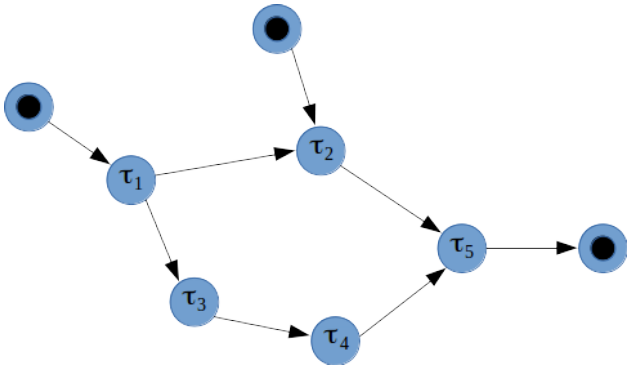


Figure 2.2: Graph including I/O nodes and edges

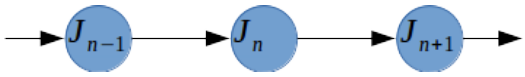


Figure 2.3: Graph including jobs

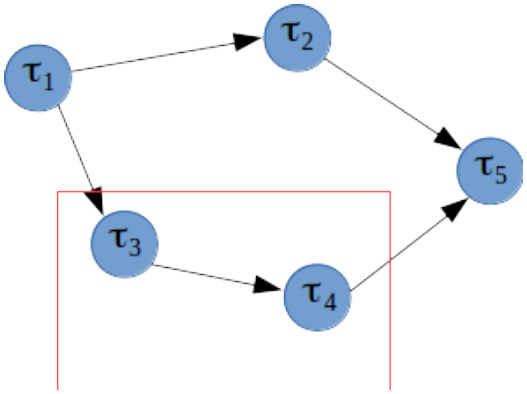


Figure 2.4: Hierarchical task graph

Communication (and computation):

Communication will be reflected by the edges in the task graphs. And the computation in the nodes.

- **Models of communication:** Two different models: **shared memory** and **message passing**.
  - Shared memory: Components can access the same memory. One needs locked-writes, semaphores, conditional critical regions, monitors, spin locks.
  - Message passing: Messages are sent and received (slower than shared memory). There is: asynchronous message passing (non-blocking communication), synchronous message passing (blocking communication, rendezvous based communication), extended rendezvous (remote invocation).
- **Organization of computations within the components:** Differential equations, FSMs, data flow, discrete event model, Von-Neumann model.
- **Combined models:** Actual languages combine some models of communication.

StateCharts: StateMate, StateFlow, BetterState

2.2.3 Early Design Phases

The first ideas could be captured in a very informal way, also natural language.

Use Cases

Use cases (e.g. UML Use Cases; KDE Umbrello) deliver a good understanding of the system in the very early design phase. It shows what the contributors of the system are, what they can do and which functions are needed. It will not show the sequence of the functions.

Example: Figure 2.5 is a simplified use case diagram for an answering machine.

(Message) Sequence Charts and Time/Distance Diagrams

Here, the sequence and timing of the messages could be seen, it is just another perspective of the same story. Example of a Sequence Chart (SC), figure.

In SC, it is not planned to transport timing, for this, time/distance diagrams (TDD) will be used [5].

Example: Figure 2.6 shows an example.

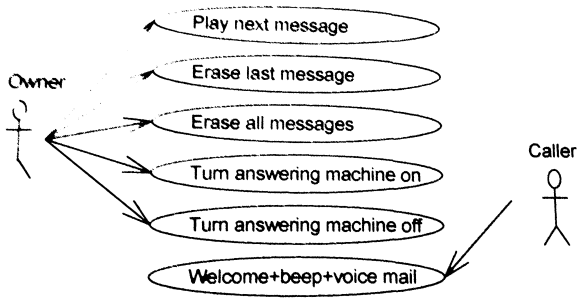


Figure 2.5: Use case example

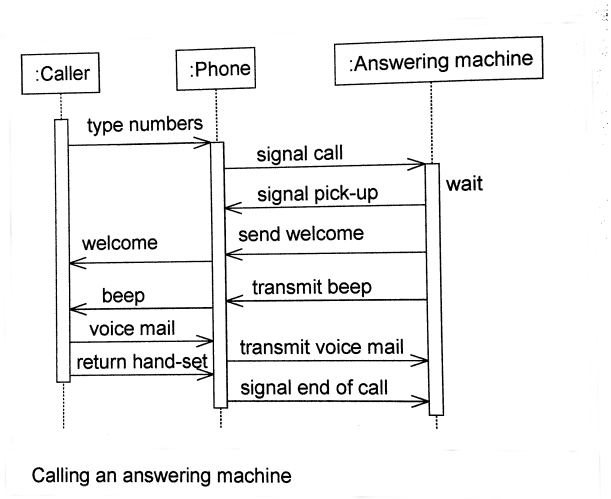


Figure 2.6: Answering machine in UML

Differential Equations

Differential equations are needed to model the real world environment of CPS.

2.2.4 Communicating Finite State Machines (CFSMs)

The importance of Finite State Machines (FSM) has already been mentioned. FSMs, where only one state at a time is active, will be called **deterministic** FSMs. FSMs, which are clocked, are called **synchronous** FSMs. FSMs can be described by state diagrams, fig. 2.7.

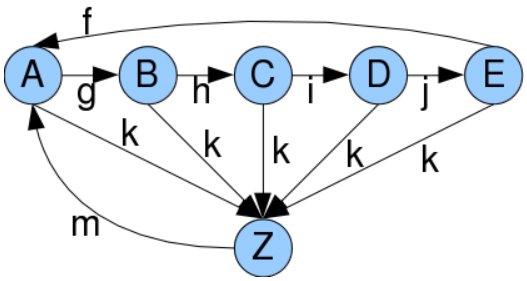


Figure 2.7: State diagram

2.2.5 Executable Specification

see SystemC

Chapter 3

History / Revision / Change Management

Every chapter of the specification (SpecChap) needs its own history table (History / Revision / Change Management). It reflects the changes which have been necessary to refine the document. Every change requires a new version of the document. Figure 3.1 shows a snap-shot of such a history table (from the SpecChap: Document Overview). It is important to see who is the owner of the changes, which versions of the specification (current, previous), which are the changed paragraphs, when has the change been made and tell a few words what has been changed.

1.6 Document List

Open items :

- check and complete list
- mark the attachments and give source description for the other docs.

History		
Target Spec. Current version : 2.2a, 2003-11-24 Previous version : 2.1, 2003-08-21		Author : [redacted]
Paragraph (in previous Version)	Paragraph (in current Version) / Date	Subjects (major changes since last revision)
Current Chapter Version : C0.1 Date :12.9.01		
	11.9.01	First draft
Current Chapter Version : C1.0 Date: 19.11.01		
	19.11.01	First version
Current Chapter Version : C1.1 Date: 17.12.01		
	17.12.01	Documents added
Current Chapter Version : C1.2 Date:		
	29.12.01	Documents added
	11.01.01	EBU and USB docs updated
	11.04.02	Pinlist added
	16.04.02	EBU and USB doc updated
Current Chapter Version : C1.3 Date:		
	3.5.02	Ballout added

Figure 3.1: History: A Snap-Shot of a Table

Chapter 4

SpecChap 1: Document Overview

The Document Overview should contain:

- Table of Contents: obvious
- List of Figures: obvious
- List of Tables: obvious
- Glossary: explain technical words
- Naming Conventions: explain synonyms which has been used
- Document List: all additional documents which will be needed to complete the specification. E.g., if you are using an ARM core, the ARM specification must be mentioned (Figure 4.1).

Table 1-3 Document List

Number	Doc Type	Version	Date	Filename
[1]	Pinlist			
[2]	Ballout			
[3]	ARM engineering specification	A09	-	ARM926EJ-S_EngSpec_A09 (old)
[4]	ARM926 technical reference manual	-	-	DDI0198B_926
[5]	ARM9EJ-S technical reference manual	-	-	DDI0222A_9EJS_R1
[6]	ARM commands and Assembler	-	-	ADS_AssemblerGuide_A
[7]	ARM Errata Sheet	-	-	ARM926EJ-S rev 0.3 Errata List v5
[8]	ARM PrimeCell DMA Controller (PL080)	-	-	PL080-INTM-0000-A (missing in attachments, all)

Figure 4.1: Document List

Chapter 5

SpecChap 2: Product Overview

5.1 Top Level View

5.1.1 Introduction

A few words about the entire system. A system view with a short explanation could help here (figure 5.1).

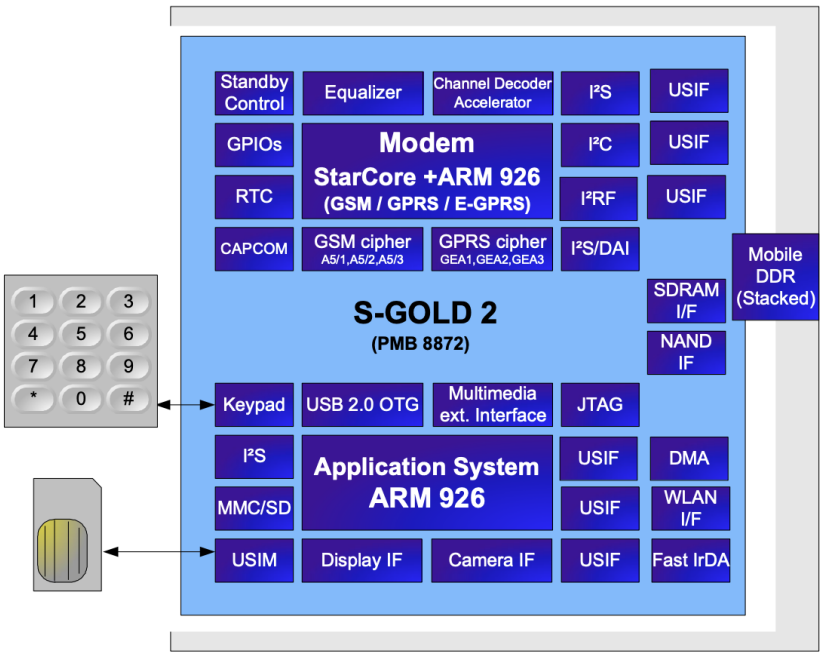


Figure 5.1: System View

5.1.2 Key Features

A list of the key features of the product. Supported standards, used processors, memory, busses, peripherals, etc.

5.1.3 Functional Block Diagram

The block diagram of the product, not the system, will be shown here. The descriptions of the sub-blocks will be explained in the following sections. Figure 5.2 shows an example.

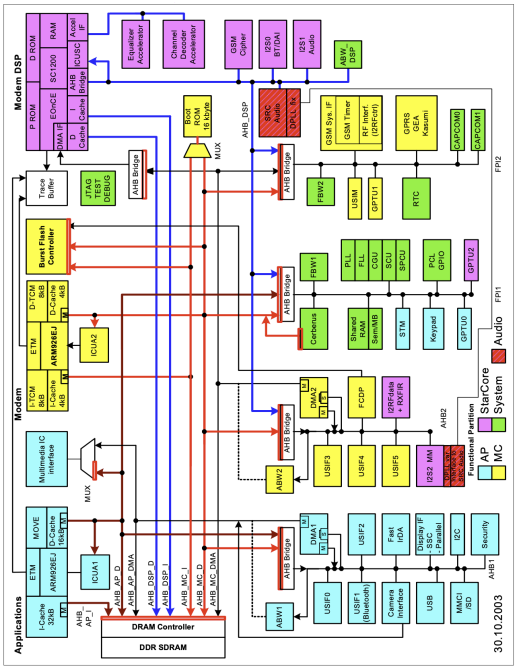


Figure 5.2: Product View

5.1.4 Package

Which kind of a package will be used for the chip (QFP144, P-LFBGA).

5.2 System View

5.2.1 Application System Description and Use Cases

Describe the use cases of the system.

5.2.2 System Description

Short words describing the system.

5.2.3 System Integration and Application Circuit

How is the product/chip be embedded into the system, figure 5.3 shows an example.

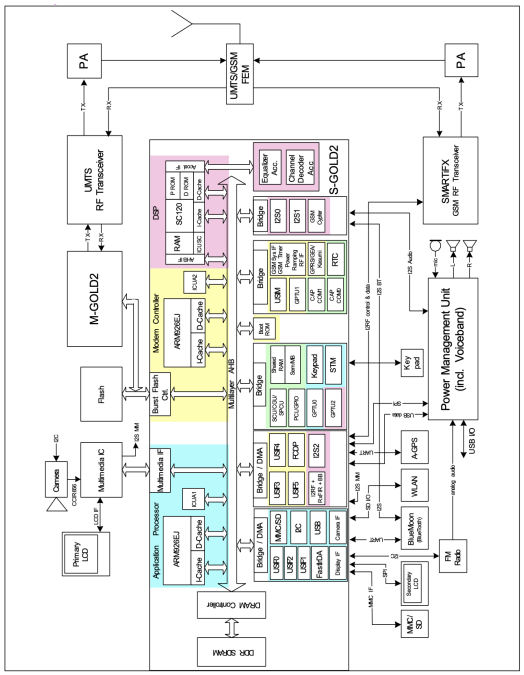


Figure 5.3: System Integration

5.2.4 Compliances

Are there any compliances (UMTS, 5G, IEEE 802.11n (WiFi))?

5.3 Architecture Concepts Overview

5.3.1 Technology

Which technology will be used, e.g. $14nm$.

5.3.2 System Memory Concept

Wow much memory and which kind of memory is planned to be on the chip.

5.3.3 Software Architecture

If drivers will be needed, the key features will be listed here.

5.3.4 Interprocessor Communication Concept

How do the processors communicate with each other (semaphores, simlocks, mailboxes, UART, USB, ...).

5.4 Functional Block Overview

A very short overview of the key features of all sub-blocks. Sub-blocks could be:

- Controller
- Bus peripherals
- Test and Debug
- etc.

5.5 Pin List

The top level pins must be described here. It will be done best as a table.

Chapter 6

SpecChap 3: Architecture Concepts

6.1 Bus Concept

A detailed description of the used busses will be done here. Which peripherals will be connected to which bus system. The arbitration schemes must be explained. Address decodings will be listed.

6.2 Interrupt and DMA Concept

How will the interrupts and the DMA be used and connected in the chip.

6.3 System Control Concept

The system control should be explained here.

6.4 Clock System

Which peripherals and which busses get which clocks. Explain how the clock system is organized in the system.

6.5 Power Management Concept

How could the power consumption be regulated. This is only needed if power is a system issue.

6.6 External Bus Concept

If there is an external bus access, it must be described and explained.

6.7 DSP Concept / Firmware

The features of the DSP subsystem should be explained.

6.8 Debug Concept

If debugging is needed, the key features will be explained here.

6.9 Audio System Concept

The audio concept, the signals and the coding must be described.

6.10 Protection and Security Concept

If the system needs to be protected against non-authorized usage, it should be explained how this will be done.

Chapter 7

SpecChap 4: Description of the Design Elements

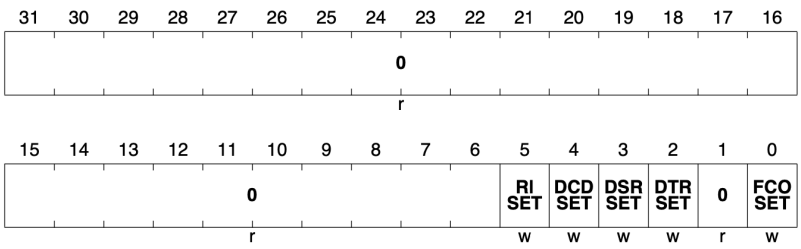
All sub-blocks of the design get its own sub-chapters here. The block must be described in words, referencing figures and tables which support the explanation. The single elements to be described are:

- HW interfaces to other blocks
- Bus interface
- Registers
- Functional overview
- Structural overview
- Interrupts
- Clock and reset
- Voltage class

The use of Flow-Diagramms, State-Charts, State-Diagrams, tables, figures, etc. is highly recommended. But it is required, that these graphics will be explained in the texts. A graphic without explanation cannot be accepted.

A central element of the configuration and function of the blocks are their registers. So, this registers (memory mapped) needs to be explained (figure fig:asSpec17). And of course, there should a memory map with the addresses of the registers.

MSS_SET
Modemstatus Set Register **[Reset value: 0000'0000$_H$]**



Field	Bits	Typ	Description
FCO SET	0	w	Flow Control Output Set 0 ... no effect 1 ... assert FCO (signal is pulled low) *Note: If Hardware Flowcontrol for the FCO signal is enabled, writing this bit has no effect.*

Figure 7.1: Register Example

Chapter 8

SpecChap 5: Test and Debug

The test and debug concept must be described and explained. What must be tested? Which functional tests are needed and how will the design be tested to reach the 100% coverage?

There could be dedicated hardware for the tests (JTAG, TAP, etc.), this hardware must be explained with its function and interfaces.

Chapter 9

SpecChap 6: Memory Maps and Register Lists

List the memory map and all registers with its addresses and reset values (figures 9.1, 9.2).

9.2.2 Controller Memory Map

Table 9-3 Top Level Memory Map without Overlays.

From	To	Subsystem	Submodule	Notes
0x0000 0000	0x0000 7FFF **For ES1, WS7010 16KB ROM:** 0x0000 3FFF	LMU RAM1	RAM	If scu_boot_rom=1 this area is overlayed by ROM (note that the four word locations based at 0x0000,1000 are not accessible to the CPU and will return 0xFFFF,FFFF regardless of the actual ROM content). If INITRAM=1, (VINITHI=0) this area may be overlayed by ITCM which permits boot after SW reset from ITCM
0x0000 4000	0x0000 *7FFF*	LMU RAM1	RAM	Remaining part of RAM1 power partition (ES1 Only - Errata WS7010)
0x0000 8000	0x0000 FFFF	LMU RAM2	RAM	RAM2 power partition
0x0001 0000	0x0001 7FFF	LMU RAM3	RAM	RAM3 power partition
0x0001 8000	0x0001 FFFF	LMU RAM4	RAM	RAM4 power partition

Figure 9.1: Memory Map

0xF430 0000	0xF43F FFFF		GPIO	
0xF440 0000	0xF44F FFFF		SCU	
0xF450 0000	0xF45F FFFF		CGU	
0xF460 0000	0xF46F FFFF		SPCU	
0xF470 0000	0xF47F FFFF		RTC	
0xF480 0000	0xF48F FFFF		Reserved	
0xF490 0000	0xF49F FFFF		GPTU0	
0xF4A0 0000	0xF4AF FFFF		GPTU1	
0xF4B0 0000	0xF4BF FFFF		STM	
0xF4C0 0000	0xF4CF FFFF		TSMU	
0xF4D0 0000	0xF4DF FFFF		KEYPAD	
0xF4E0 0000	0xF5FF FFFF		Reserved	
0xF600 0000	0xF600 0FFF	FPI3	SHARED RAM DSP : Registers	
0xF600 1000	0xF600 27FF		SHARED RAM DSP : RAM	
0xF600 2800	0xF60F FFFF		Reserved	
0xF610 0000	0xF61F FFFF		*USIM*	
0xF620 0000	0xF62F FFFF		GPRS	

Figure 9.2: Memory Map

Chapter 10

SpecChap 7: Electrical Characteristics

The electrical characteristics chapter delivers all data which an engineer needs who wants to use this chip, who wants to solder this chip on a PCB together with other chips. The supply voltages must be defined, the input/output voltages must be know, the driver strength (current) of the output pins must be given, the power consumption is an important issue, the timings of the signal are important and also the absolute maximal values not destroying the device should be known (figure 10.1). There are minimal and maximal values. For bigger SoCs, there can be different supply voltages.

10.2 Absolute Maximum Ratings

Table 10-1 Absolute Maximum Ratings

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Case temperature under bias	T_C	-25	85	°C
Storage temperature	T_{STG}	-55	150	°C
Supply voltage V_{DD_MAIN}	V_{DD_CORE}	-0.15	1.35	V
Supply voltage V_{DDP_EBU}, V_{DDP_ETM} of the digital pins in the I/O supply domains of EBU and ETM interface	V_{DDP_EBU} V_{DDP_ETM}	-0.3	3.60	V
Supply voltages V_{DDP_SIM} of the digital pins in the I/O supply domains of USIM	V_{DDP_SIM}	-0.3	3.60	V
Supply voltages V_{DDP_DIGx}, x= A,B,C,D,E of the digital pins in the I/O supply	V_{DDP_DIGx}	-0.3	3.60	V

Figure 10.1: Absolute Maximum Ratings

Figure 10.2 shows the operating range of the chip. Here, the temperatures, voltages and currents for the recommended operation of the SoC will be given. It

is not recommended to drive the chip with the absolute maximum ratings because this will not guarantee the function of the chip.

10.3 Operating Range

Table 10-2 Recommended Operating Conditions

(-25°C[1] ≤ T_C[2] ≤ 85 °C)

Parameter	Symbol	Limit Values			Unit
		min.	nom.	max.	
Core supply voltage in slow mode	V_{DD_Cslow}	1.00[3]	1.05	1.40[4]	V
Core supply voltage in fast mode	V_{DD_Cfast}	1.25[5]	1.35	1.40[4]	V
Core supply voltage in data retention mode	V_{DD_Cret}	0.87[6]	0.90	1.40	V
PLL[7] supply voltage	V_{DD_PLL}	1.25	1.35	1.40	V
Mixed signal (analog) supply voltage	V_{DD_ANA}	2.25	2.50	2.75	V
Supply voltage V_{DDP_EBU} of the digital pins in the I/O supply domain of the EBU and of the e-fuse block during fusing operation ($I_{DDP\_EBU\_f}$ < 100 µA if no switching activity on I/Os) (level shifters require voltage)	$V_{DDP\_EBU\_f}$	1.70	1.80	1.90	V
Supply voltage V_{DDP_EBU} of the digital pins in the I/O supply domain of the EBU without	V_{DDP_EBU}	1.70	1.80	1.95	V

Figure 10.2: Operating Range

Figure 10.3 shows that also the input and output capacitances and resistors must be specified. This is the major issue for the timing behavior on the PCB.

10.4 Capacitances and Resistors

Table 10-4 Input Capacitances and Resistors

Parameter	Symbol	Limit Values			Unit
		min.	typical	max.	
Input capacitance, f_C = 1 MHz	C_{IN}				pF
Digital I/O; DC value	C_{I/O_DIG}				pF
PLL (F26M)	C_{PLL_I}		22		pF

Figure 10.3: Capacitances and Resistors

Figures 10.4 and 10.5 show, that it is important to show the data of special (analog) interfaces. Here it is the Baseband Receiver Unit.

10.5 Electrical Characteristics of Baseband Receiver Unit

Table 10-6 Baseband Receiver Path (Signal Inputs IR/IRX, QR/QRX)

Parameter	Symbol	Limit Values			Unit	Test Condition/ Remark
		minimum	typical	maximum		
Characteristics						
Input resistance (Single ended)		90 5	150[1)] 10[2)]		kΩ	RXON = 1
150K in Series with diff. OPamp						
Input resistance (Single ended)		1			MΩ	RXON = 0
ADC power consumption (standard mode)			6		mA	
ADC power consumption (enhanced mode)			16		mA	
Input capacitance (Single ended)			5	10	pF	

Figure 10.4: Special Functions Characteristics

typical 2.0 Vpp 5% gain variation -> 1.90 Vpp						
Differential offset introduced by ADC		-70		70	mV	
Signal-to-noise + distortion (standard mode)	S/(N+D)	63	66[3)]		dB	1.0 Vpp desired (@ 20 kHz) (0... 100kHz)
Signal-to-noise + distortion (standard mode)	S/(N+D)	43	46[3)]		dB	0.1 Vpp desired (@ 20 kHz) 1 Vpp interferer (@ 200 kHz)
Signal-to-noise + distortion (enhanced mode)	S/(N+D)	16.5	19.5[4)]		dB	1.0 mVpp desired (@ 20 kHz) (0...100 kHz)

Figure 10.5: Special Functions Characteristics

Figure 10.6 shows the DC characteristics of the pads.

10.6 DC Characteristics

10.6.1 Pads

10.6.1.1 Pad Characteristics

T AS: incl. USIM, RTC and MMC values(16.12.04).

Table 10-7 Pad Characteristics

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Digital Pins in the I/O supply domain of VDDP_EBU and VDDP_ETM						
EBU: L-input voltage for memory Interface	V_{IL_PEBU}	-0.3		0.75	V	
EBU: H-input voltage for memory Interface	V_{IH_PEBU}	1.07		2.25	V	
EBU: L-output voltage for pads of Memory Interface at a load current of 100µA design target 300µA	V_{OL_PEBU}			0.004	V	
EBU: H-output voltage for pads of memory interface at a load current of 100µA design target 300µA	V_{OH_PEBU}	1.685			V	
Input/Output leakage current	I_{IZ}			TBD		

Figure 10.6: DC Characteristics

Figure 10.7 shows the power consumption of the chip parts. Note, that just the current consumption needs to be measured, the power is the product of the voltage and the current.

10.7 Power Supply

For typical current consumption under various operating conditions please refer to the current model available as a separate document.

For the maximum values of currents in the mixed signal (analog) supply domains please refer to the electrical characteristics tables in the respective chapters (**Table 7-38**, **Table 7-39**, **Table 7-57**, **Table 7-58**, **Table 7-59**)

Owner: Mahrla (partially) Completion: see comments

Table 10-11 Current Characteristics

Parameter	Symbol	Limit Values			Unit	Test Condition
		min	typ	max		
Core supply current in fast mode (Mahrla)	I_{Cfast}			400	mA	VDD_C = VDD_Cfastmax Mahrla 041222 : reduction to 350 mA based on power model may be possible)
Core supply current in slow mode (Mahrla)	I_{Cslow}			120	mA	VDD_C = VDD_Cslowmax Mahrla 041222 : 1/3 frequency, -16% voltage
Core supply current f_{osc} = 0 Hz, 32 kHz clock running	I_{C32k}			100	mA	VDD_C = VDD_Cslowmax mainly due to leakage - safe value
PLL domain supply current	I_{PLL}			20[1)]	mA	(Mahrla)
RTC domain supply current	I_{RTC}			t.b.d.		
Core supply power-down current (all core supply domains switched off except VDDstandby)	I_{Cpdn25}		100		µA	T_C = 25 C (Mahrla 041222)
	I_{Cpdn85}			10[2)]	mA	T_C = 85 C (Mahrla 041222)

Figure 10.7: Power Supply

Figures 10.8 and 10.9 show the DC and AC timings of all interfaces.

10.8.2.25 Chipcard-Interface (USIM-Interface)

Responsible: Jennings; last change: 11.4.02 (only checked); 15.04.02, approved by DJ and HRM 10.05.02 ti4, ti7 relaxed to values used in E-GOLD-V2
12.12.04: Responsible Andreas Siggelkow

Table 10-43 Timing Characteristic of Chipcard Interface

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
CCLK clock period (output)	t_{i1}		307		ns
CCLK high time	t_{i2}	110	(307-2x15)/2=138.5	184	ns
CCLK low time	t_{i3}	110	gemessen: 146	184	ns
CCIO (output) valid after CCLK low end	t_{i4}		gemessen: 104	200	ns
CCIO (output) still stable after CCLK low end	t_{i5}	30	gemessen 80		ns

Figure 10.8: Timing Diagrams

Table 10-43 Timing Characteristic of Chipcard Interface

CCIO (input, 50%) setup before CCLK high end	t_{i6}	0			ns
CCIO (input, 50%) hold after CCLK low begin	t_{i7}	200			ns

The PAD CCIO (open drain) must be able to achive a rise and fall time of 1µs with an external pull up resistor of about 20kΩ (normal mode) and 400ns with an external pull up resistor of about 10kΩ (fast mode).

Note: CCIO is an OD pad. The timing values are specified for a pull up current of 1 mA and a capacitance of ***70 pF***. *Current should be calculated with I= VDD/20kOhm. DJ contacts R Heiermann. Current should account for tester load of ~70pF, which is more than 30pF specified in specs*

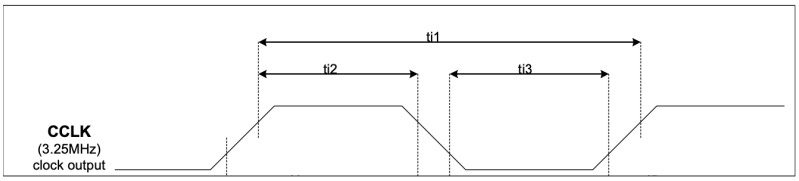


Figure 10-64 Chipcard-Interface Timing

Figure 10.9: Timing Diagrams

Chapter 11

Framework

11.1 Language of the Specification

German or English, ask your supervisor.

11.2 Criteria of Assessment

11.2.1 Content (scientific approach)

- Methodology: approach, theoretical fundamentals
- (Scientific question: clarity, goals)
- Thematic limitation: justification
- (State of research / technology: Description (state-of-the-art))
- (Investigation methods: adequate application)
- Problem analysis and evaluation: empirical data, measured values, models
- Correct and unambiguous content
- Argumentation: conclusive, objective, logical
- (Examples: appropriate)
- Sources: quality, quantity, processing of the topic
- Structure: logical, stringent
- (Criticism: own work)

- (Knowledge progress: is there any?)

11.2.2 Style

- Choice of words: correct, appropriate
- Technical terms: not too much, not too little
- Grammar: should be correct
- Argumentation: logic (premise, premise - conclusion)
- Formulation: conciseness, clarity, comprehensibility

11.2.3 Form

- Structure: continuous
- Quotations: according to the guidelines
- Spelling: grammar, punctuation, separation
- Sources: forwards and backwards complete
- Figures and tables: quality
- Font: sentence, line spacing, margins, etc.
- Overall impression

Chapter 12

Form of the Specification

12.1 Number of Pages

The **Specification** should have a length of 20 to approx. 70 pages. Restriction is an important part of your job and requires you to separate the important from the unimportant. A precise representation is required, not a "gossip". Discuss any deviations with your supervisor. A large number of pages is not a guarantee of a good grade.

12.2 Format

- Page: DIN A4, printed on one side, bound
- Margins: top 2.5 cm, bottom 2.0 cm; left and right 2.5 to 3.0 cm each
- Fonts: Fonts with serifs (e.g. Times New Roman), no more than two different ones
- Font size: 12 point
- Line spacing: text 1.5 lines, bibliography 1 line
- Page numbering: starts with 2 after the title page, the explanation has no page number
- Text: justified, automatic hyphenation
- Page change: Only main chapters force a page change
- Header: Chapter

- Footnotes: smaller font size
- Text: no CAPITALS, SMALL CAPITALS, underline, better: **bold**, *italics*

12.3 Components

The following order can be used:

- Title page
- (Declaration of independence)
- Table of contents / structure
- List of figures and tables
- List of abbreviations
- (Summary, one page)
- Text parts
- Bibliography
- (Acknowledgments)
- (Appendix)

12.3.1 Title Page

Target Specification, Revision 15.2, 2021-03-22, product name, University, study program, **Specification** of Maxi Mustermann, faculty / institute, supervisor

12.3.2 Table of contents / Structure

Please discuss the structure with your supervisor. It reflects the structure, should be structured according to the logic of decision-making, the common thread must be recognizable. Formalities of the structure:

- use Arabic numerals, e.g. 1, 1.1, 1.1.2.4 etc.,
- do not subdivide with A, B, ... or I, II, ...,
- a maximum of five levels, e.g. 1.2.1.3.4, each level must have at least one counterpart, e.g. a 1.1 must be followed by a 1.2,

- self-explanatory bullet points,
- explanations of the sections are not too long.

12.3.3 Tables / Figures

- summarize complex issues and clarify them
- should bring a unique gain in information, not explain the same information in a table and in a figure
- the information content of the table or the figure must be formulated and explained in the text
- there must be references to the figures and tables in the text
- a legend is an integral part of every figure and table, it must be self-explanatory, explain abbreviations, units of measurement, symbols, etc.
- figures and tables also require a reference to the source, unless it is exclusively from you
- no special effects

Design Tables

- Table legend above the table.
- Numbering with Arabic numerals.

Design Figures

- Figure legend below the figure.
- Numbering with Arabic numerals.
- Sufficient resolution (300 dpi) for photos
- Lines $> 0.3mm$ line width, no shading, no fill pattern, gray levels are better
- Indication of source if the illustration is not from you.
- Always quote photos, including your own, with the source / author
- Do not change the original images or data of the test results (data manipulation)

12.3.4 Text of the Specification

The text of the work is the core of your **Specification** . It should:

- be very precise but not too detailed,
- be exclusively themed,
- be understandable without further aids,
- be legible and exciting (the addressee is the educated layman).

Chapter 13

Literature

13.1 Which Literature?

- All literature and information sources for scientific work (from textbooks to specialist journals and internet sources). Specialist journals that contain specific knowledge on certain topics are preferred (IEEE). Reference works, lexicons, encyclopedias, etc. are not suitable. Wikipedia can be used as an introduction to the subject, but must be supported by further literature.
- Published sources can be cited (verifiability).
- "Gray literature" - including unpublished seminar, bachelor, master and doctoral theses - can be mentioned in the text as a short quotation, but must, if necessary, be separated from the list of sources (e.g. under "Non-published sources").
- No popular magazines, daily newspaper articles.
- No private homepages

13.2 Research on Literature

Important work from the start. Opening up the sources:

- Pyramid scheme: a central article leads to others, etc.
- Systematic search: targeted search in magazines and databases
- Forward search: Staring point is an important article on the topic. Use the database, e.g. IEEE, SciHub, to search for authors who have worked with this article.

13.3 How many Sources?

- Depending on the topic, discuss it with the supervisor
- Relevant sources only
- After collecting the sources: read, excerpt, summarize, integrate into the work
- Time required: 2/10 research - 3/10 reading - 5/10 writing
- Use reference management programs, e.g. EndNote, Citavi

13.4 How to quote

siehe I.

- Thoughts must be comprehensible, evidence must be verifiable
- General knowledge of the relevant subject does not have to be proven (Ohm's law)
- literally (directly) or analogously (indirectly)
- The IEEE quotation (see I) is common in the engineering and scientific work.
- Do not copy entire text passages, not even as quotations with the source (exception: standards), not even in the state-of-the-art, discuss exceptions with the supervisor

13.5 Bibliography

- Cite all sources, substantiate all claims
- Only sources in the bibliography which will be used in the text

Chapter 14

Famous Last Words

14.1 Correction Hints

- no redundant information: filler words, empty phrases, etc.
- check again the the logic of the structure and the argumentation
- pay attention to spelling mistakes, use the spelling and grammar correction of the word processing program
- check for correct punctuation
- make corrections not only on the screen, also in a printed version
- prove read it after a time period
- Let it prove read

14.2 Plagiarism

- Plagiarism is the unauthorized exploitation or presumption of authorship, i.e. if you do not mark results, opinions, quotations, used literature, illustrations, etc. as third-party intellectual property - regardless of whether it is unintentional or intentional [28].
- It is not sufficient to cite a source once if it has been used several times in a work, but its relation to its text is no longer recognizable. You would have committed a stylistic plagiarism.

- For example, the copy and paste of Internet content, combined to form a text, is not an independent intellectual achievement. If the exact and comprehensible source is missing, this would be deliberate plagiarism. This also applies to all types of protocols [28].
- Plagiarism is scientific misconduct. In this case, a thesis can be graded insufficient (5.0); It is even possible to be excluded from the study program.

Part I

Literature

Chapter 15

Literature

15.1 Citation style

There are many citation styles possible. The citation style used here is the **IEEE** style. It is used in the electrical engineering community. See https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf

15.1.1 IEEE Style

IEEE is the *Institute of Electrical and Electronical Engineers*. If one needs to publish, it could be in an IEEE journal.

Citations used in a Text

IEEE uses numbers to point to positions in the list of read publications, e.g. **[1]**. This number stands for a monograph, book, paper or internet resource. If multiple publications should be cited, the numbers could be listed, e.g. **[1, 2, 3]** or **[1-5]**.

The Bibliography

The bibliography lists the used publications in the numbered order.

Monographs and Books: The monographs starts with the number, followed by the author's name, the title, the subtitle and the city, publisher and year. E.g.:

- Basic format: [1] J. K. Author, "Title of chapter in the book," in *Title of His Published Book*, xth ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx-xxx.

- [2] I. Schneider, *Archimedes - Ingenieur, Naturwissenschaftler, Mathematiker*. Berlin u.a.: Springer, 2016.
- [3] B. Klaus and P. Horn, *Robot Vision*. Cambridge, MA, USA: MIT Press, 1986.
- [4] A. Histace, "Image restoration-Recent advances and applications," in *Super-Resolution Restoration and Image Reconstruction for Passive Millimeter Wave Imaging*. Rijeka, Croatia: InTech, 2012, pp. 25-45.

Books, Monographs (Online): This is the same as above, but with a link.

- Basic format: [1] J. K. Author, "Title of chapter in the book," in *Title of Published Book*, xth ed. City of Publisher, State, Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx-xxx. [Online]. Available: http://www.web.com

Periodicals: For IEEE Transactions/Journals.

- Basic format: [1] J. K. Author, "Name of paper," *Abbrev. Title of Periodical*, vol. x, no. x, pp. xxx-xxx, Abbrev. Month, year, doi: xxx.
- [2] M. M. Chiampi and L. L. Zilberti, "Induction of electric field in human bodies moving near MRI: An efficient BEM computational procedure," *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 2787-2793, Oct. 2011, doi: 10.1109/TBME.2011.2158315.

Bibliography

[1] Matthias Springer, "What's the difference between safety and security?," TÜV-Nord, Germany. Accessed: 8.8.2017, Available: https://www.tuev-nord.de/explore/en/explains/whats-the-difference-between-safety-and-security/

[2] Swarup Bhunia, Sandip Ray, Susmita Sur-Kolay (Editors), *Fundamentals of IP and SoC Security: Design, Verification, and Debug*. Cham, Switzerland: Springer, 2017, ISBN 978-3-319-50055-3

[3] Wikipedia, "Verification and validation," Accessed: 07.05.2020, Available: https://en.wikipedia.org/wiki/Verification_and_validation

[4] E.A. Lee, "Computing foundations and practice for cyber-physical systems: a preliminary report". Tech. Rep. UCB/EECS-2007-72, EECS Department, University of California, Berkeley, 2007. Accessed: 8.8.2017, Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-72.html

[5] P. Marwedel, *Embedded System Design: Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things*, 3rd ed. Berlin: Springer, 2018

[6] P. Marwedel, *Embedded System Design*. Norwell: Kluver Academic Publishers, 2003

[7] D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), *The Internet of Things, 20th Tyrrhenian Workshop on Digital Communications*, Berlin: Springer, 2010

[8] Wikipedia, "Anforderungsanalyse (Informatik)," Accessed: 13. Sept. 2017. Available: https://de.wikipedia.org/wiki/Anforderungsanalyse_(Informatik)

[9] A. Jantsch, *Modeling Embedded Systems and SoC's: Concurrency and Time in Models of Computation*. San Francisco: Morgan Kaufmann, 2004

[10] C.-F. Kao, H.-M. Chen, I.-J. Huang, "Hardware-Software Approaches to In-Circuit Emulation for Embedded Processors," *IEEE Design & Test of Computers*, vol. 25, no. 5, pp. 462 - 477, Sept. - Oct. 2008

[11] Syed Farid Syed Adnan, Mohd Anuar Mat Isa, Khairul Syazwan Ali Rahman, Mohd Hanif Muhamad, Habibah Hashim (2015): Simulation of RSA and ElGamal Encryption Schemes using RF Simulator. IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2015 IEEE

[12] A. Lopez, K. Vatanparvar, A. Prasad, S. Yang, S. Bhunia, M. A. Al Faruque. "A Security Perspective on the Battery Systems of the Internet of Things". *Springer Journal of Hardware and Systems Security (Springer HASS 17)*, vol. 1, pp. 188-199, 2017.

[13] S. Chhetri, N. Rashid, S. Faezi, M. A. Al Faruque. "Security Trends and Advances in Manufacturing Systems in the Era of Industry 4.0," ACM/IEEE International Conference on Computer-Aided Design (ICCAD 17), Irvine, CA, November, 2017

[14] Heise: https://www.heise.de/security/meldung/Forscher-demontieren-kaputten-Krypto-Standard-P1735-der-IEEE-3883590.html

[15] Animesh Chhotaray, Adib Nahiyan, Thomas Shrimpton, Domenic Forte, Mark Tehranipoor, Standardizing Bad Cryptographic Practice, CCS '17, October 30-November 3, 2017, Dallas

[16] Alexander de Graf, SystemC: an overview, EEMCS/ME/CAS, study script, 10-5-2016, TU Delft

[17] David C. Black, Jack Donovan, Bill Bunton, Anna Keist, *Systemc: From the Ground Up*, 2nd ed., Springer, 2010

[18] Matthias Menge, *Moderne Prozessorarchitekturen: Prinzipien und ihre Realisierungen*. Berlin: Springer 2005

[19] David A. Patterson, John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, RISC-V Edition. Morgan Kaufmann Publishers, 2018

[20] David A. Patterson, John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, ARM Edition. Morgan Kaufmann Publishers, 2018

[21] John L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann Publishers, 2019

[22] Christian Siemers (2014), Embedded Systems Engineering 1, Skript, TU Clausthal

[23] Christoph Krauß, Cyber-Physical Systems: Sicherheit im Internet der Dinge, https://www.sit.fraunhofer.de/de/cyberphysicalsystems/, 02.04.2019

[24] Klaus Wüst, *Mikroprozessortechnik - Grundlagen, Architekturen, Schaltungstechnik und Betrieb von Mikroprozessoren und Mikrocontrollern*, 3rd ed. Wiesbaden: Vieweg+Teubner, 2009

[25] B. Parhami, *Computer Architecture: From Microprocessors to Supercomputers*, New York, Oxford Univ. Press, 2005

[26] Wikipedia, "Trap (computing)," Accessed: 07.07.2020, Available: `https://en.wikipedia.org/wiki/Trap_(computing)`

[27] Wikipedia, "Advanced eXtensible Interface," Accessed: 11. Jan. 2021. Available: `https://en.wikipedia.org/wiki/Advanced_eXtensible_Interface`

[28] Bachelor-Thesis und Protokolle in Biologie an der Uni Würzburg, Mai 2018