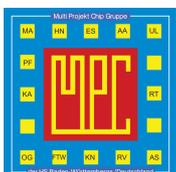


# MPC

MULTI PROJECT CHIP GRUPPE  
BADEN-WÜRTTEMBERG

Herausgeber: Technische Hochschule Ulm    Ausgabe: 63    ISSN: 1868-9221    Workshop: Mannheim Februar 2020

- 1 Implementation of a GPS and GSM Module into a Zynq Z7 SoC Based Emulator Tracking System**  
Gregor Benz, Andreas Siggelkow, Hochschule Ravensburg-Weingarten
- 5 Ein Geschwindigkeitsmesssystem nach dem Ortsfrequenzfilterverfahren im Bereich der funktionalen Sicherheit**  
Frank Wasinski, Werner Bonath, Ubbo Ricklefs, Josef Börcsök, Waldemar Müller, Eike Hahn, Michael Schwarz, Technische Hochschule Mittelhessen
- 13 Hybrid Image Processing MPSoC Architecture for Cloud-based Camera Monitor Systems**  
Jannik Maier, Johannes Schäfer, Anestis Terzis, Oliver Bringmann, Technische Hochschule Ulm
- 19 Entwicklung eines energieautarken Türschildes mit E-Paper-Display und NFC-Konfigurationsschnittstelle**  
Andreas Angermayr, Elke Mackensen, Hochschule Offenburg
- 27 Problemfälle der Stabilität**  
A. Zwick, B. Vettermann, Hochschule Mannheim
- 35 Universal Memory Automaton and Automated Verilog HDL Code Generation for a Cache Coherency Snooping Protocol**  
Matthias W. Fertig, Hochschule Konstanz
- 43 Aufbau von Simulations- und Prognosemodellen energietechnischer Anlagen unter Einsatz von Deep Learning Verfahren**  
Carsten Stephan, Johannes Mast, Stefan Rädle, Joachim Gerlach, Hochschule Albstadt-Sigmaringen



Cooperating Organisation  
Solid-State Circuit Society Chapter  
IEEE German Section





# Inhaltsverzeichnis

<b>Implementation of a GPS and GSM Module into a Zynq Z7 SoC Based Emulator Tracking System</b> .....	<b>1</b>
Gregor Benz, Andreas Siggelkow, Hochschule Ravensburg-Weingarten	
<b>Ein Geschwindigkeitsmesssystem nach dem Ortsfrequenzfilterverfahren im Bereich der funktionalen Sicherheit</b> .....	<b>5</b>
Frank Wasinski, Werner Bonath, Ubbo Ricklefs, Josef Börcsök, Waldemar Müller, Eike Hahn, Michael Schwarz, Technische Hochschule Mittelhessen	
<b>Hybrid Image Processing MPSoC Architecture for Cloud-based Camera Monitor Systems</b> .....	<b>13</b>
Jannik Maier, Johannes Schäfer, Anestis Terzis, Oliver Bringmann, Technische Hochschule Ulm	
<b>Entwicklung eines energieautarken Türschildes mit E-Paper-Display und NFC-Konfigurationsschnittstelle</b> .....	<b>19</b>
Andreas Angermayr, Elke Mackensen, Hochschule Offenburg	
<b>Problemfälle der Stabilität</b> .....	<b>27</b>
A. Zwick, B. Vettermann, Hochschule Mannheim	
<b>Universal Memory Automaton and Automated Verilog HDL Code Generation for a Cache Coherency Snooping Protocol</b> .....	<b>35</b>
Matthias W. Fertig, Hochschule Konstanz	
<b>Aufbau von Simulations- und Prognosemodellen energietechnischer Anlagen unter Einsatz von Deep Learning Verfahren</b> .....	<b>43</b>
Carsten Stephan, Johannes Mast, Stefan Rädle, Joachim Gerlach, Hochschule Albstadt-Sigmaringen	

**Tagungsband zum Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie.

Die Inhalte der einzelnen Beiträge dieses Tagungsbandes liegen in der Verantwortung der jeweiligen Autoren.

Herausgeber:

Lothar Schmidt, Technische Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

Mitherausgeber (Peer Reviewer):

Jürgen Giehl, Hochschule Mannheim, Paul-Wittsack-Straße 10, D-68163 Mannheim

Eckhard Hennig, Hochschule Reutlingen, Alteburgstraße 150, 72762 Reutlingen

Frank Kesel, Hochschule Pforzheim, Tiefenbronnerstraße 65, D-75175 Pforzheim

Elke Mackensen, Hochschule Offenburg, Badstraße 24, D-77652 Offenburg

Andreas Siggelkow, Hochschule Ravensburg-Weingarten, Doggenriedstraße, 88250 Weingarten

Alle Rechte vorbehalten

Diesen Workshopband und alle bisherigen Bände finden Sie im Internet unter:

<http://www.mpc-gruppe.de/de/workshopbaende.html>

# Implementation of a GPS and GSM Module into a Zynq Z7 SoC Based Emulator Tracking System

Gregor Benz, Andreas Siggelkow

**Abstract**—As part of an ever-increasing fleet management system, the location information of the various vehicles is playing an increasingly popular role. For this purpose, an emulator of an outdoor localization system is to be presented as part of this work, which uses a Zynq SoC, a GSM and a GPS module to carry out the satellite time, the longitude, the latitude and a signal strength determination of the adjacent radio masts. The determined data can then be sent to a central evaluation unit as an SMS, via the GSM module.

**Index Terms**—Automotive fleet localization, outdoor navigation, GPS, GSM, Zynq SoC, IoT.

## I. INTRODUCTION

### A. Motivation

Localization is an increasingly popular application of vehicle fleet management [1]. This can result in various advantages, such as increased security or simplified administration. Further savings in operating resources are possible because, among other things, fuel savings and automatic creation of the logbook are possible [2].

GPS-based outdoor localization systems are a suitable solution for this [2]. These systems provide a localization accuracy of up to 2 m and form an economically scalable system [3]. Since GPS-based localization systems are subject to interference such as jamming and spoofing, the implementation of a further satellite-independent approach is recommended [4]. In [5] various technologies, such as mobile radio, interactive sensors or radio-frequency identification (RFID)-based systems, are presented, which allow satellite-independent positioning in the outdoor area. Among other things, each system has a specific accuracy and infrastructure density. The emulator presented here pursues the approach of using a GSM-based position determination in addition to GPS-based localization.

In addition to commercially available GPS-based localization systems [2], [6], there are numerous publications such as [7]–[9] on this topic. Analogously, localization systems [10], [11] present a localization purely based on the use of GSM.

In addition to all of the publications mentioned above, this publication is intended to implement both

positioning approaches in a Zynq SoC emulator. For this purpose, a GPS module should determine the time stamp of the satellite that is to be used as the reference time stamp, as well as the longitude and latitude. In contrast to this, the hard-wired GSM module is to carry out a signal strength measurement (RSSI) of the closest radio station and all adjacent radio stations. The measured coordinates, the signal strength and the time stamp are to be stored in a ring buffer arrangement in the DRAM. The measurement data should be passed on to a central evaluation unit by means of a GSM module. An evaluation to localize the emulator on the central evaluation unit is not part of this work.

### B. Structure of this publication

The remainder of this paper is structured as follows: Section II gives an overview of the localization system that is used to maintain the functions of the emulator. Section III outlines the hard wiring and the implementation of the required functionality modules on the emulator SoC. The functionality of the emulator is also demonstrated here. Finally, Section IV concludes the work and gives perspectives on future topics.

## II. LOCALIZATION SYSTEM OVERVIEW

Almost all digital systems have a debug interface [12] with different possibilities to attack the system [13]. This debug logic connects all sub-blocks in the system by means of a shadow bus system in order to test or debug it. This could act as a back door which is not secured. To equip this back door and all connection points of the debug bus with a lock, is the focus of the demonstrator system introduced in the following. The lock can be a cypher system. Parallel to the debug problem is the update over the air possibility in such systems, especially modern cars (Jeep Cherokee, Heise 2015) and IoT. Also this door can be secured by cyphering.

To enable research in this fields, an application needs to be modeled first on a high abstraction level, SystemC is used, second, an emulator can be used, described in this paper and as a last step, the system can be implemented as a chip with a surrounding environment.

Gregor Benz, Andreas Siggelkow, siggelkow@rwu.de, Hochschule Ravensburg-Weingarten, Doggenriedstraße, D-88250 Weingarten.

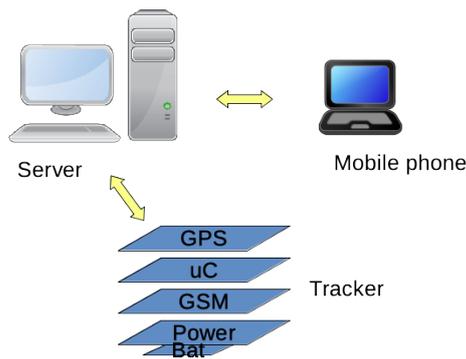


Figure 1. The complete tracking system

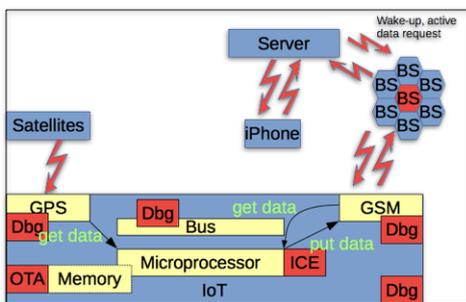


Figure 2. Tracking system with implementation details

The demonstrator which has been chosen is a tracker system for a car fleet, e.g. Uber cars. The idea of the whole tracking system is shown in Figure 1. Here we have the tracker consisting of a GPS receiver, a microcontroller, a power supply and a GSM transceiver. The GPS module receives the data from the satellites and forwards the position data and the time stamp to a memory location. The GSM module is requested to collect the power data and the corresponding base station IDs, which will be also stored in a memory. If a complete set of data has been collected, it will be send by means of the GSM transceiver to a server. The server stores everything in a database. Before storing the data, the server can calculate the position based on the GSM data by means of triangulation. This position is not as accurate as the position got by the GPS system, but the signal is more reliable in case of GPS signal loss. The position of any car can be requested by any mobile device.

Figure 2 shows the idea of the implementation. The microprocessor used in the tracker is equipped with an in-circuit- emulator (ICE) to be able to perform debugging. The memory needs a means to be able to make an SW update over-the-air (OTA). This is also some kind of debugging and a back door. The bus needs a BusWatcher. GPS and modem needs debug capabilities. So, all the blocks indicated with “Dbg”, “OTA” or “ICE” are connected via the debug bus and needs to be secured.

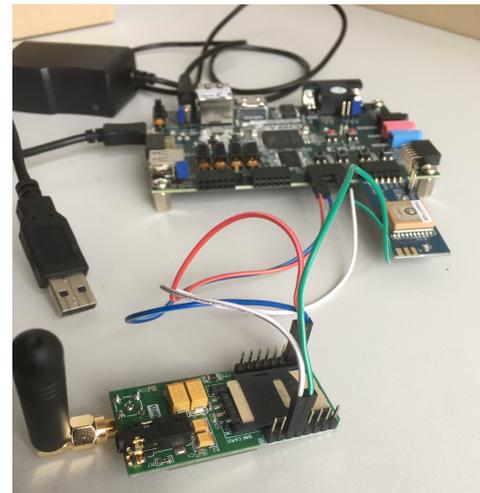


Figure 3. Tracking system on the emulator (Zybo)

The system emulator has been finished, until now without the debug capabilities, and is shown in Figure 3.

With the emulator it is possible to implement different debug elements and the corresponding cypher hardware to evaluate the security strategy. Here, only the functional elements have been implemented.

### III. IMPLEMENTATION AND FUNCTIONALITY TEST

#### A. Zynq SoC

The emulator presented here uses a Zynq-Z7 SoC from Xilinx [14] as the central processing unit, which is available on a Zybo Z7-20 evaluation board [15]. The architecture of the Zynq-Z7 used consists of a programmable logic (PL) and a processing system (PS), which enable data exchange via an AXI interface bus system [14].

The PL is an Artix-7 FPGA, the PS is an ARM-A9 dual-core microcontroller with separate L1- and L2-cache and a 256k on-chip memory (OCM). Various peripheral interfaces such as GBit-Ethernet, CAN, USB, Quad SPI Flash and GPIO are also provided [14, 16]. The system presented here uses only the GPIO, which are configured as two UART interfaces to read out the two evaluation units GPS and GSM. For each UART interface, 2 data lines are configured as transmitter (TX) or receiver (RX), as well as a 3.3V supply voltage and GND.

While the PS transfers control commands for the two evaluation units (GPS and GSM) and evaluates the returned information, the PL is only used for signal transmission between the GPS/GSM module and PS.

Another task that the PS performs is storing the measured information (satellite time, longitude, latitude, signal strengths for the individual radio masts) in the DRAM, which is hard-wired from the Zybo Z7-20 [15]. Two ring buffers are configured for this (one for the data from the GPS module, one for the data

from the GSM module). As soon as a ring buffer is full, old information is overwritten. Using a button (also GPIO, read from the PS) [14, 15], the last recorded measurement can be transmitted via the GSM module to a central evaluation unit via SMS and saved there. A position determination/evaluation based on the data of the GSM and GPS module takes place in the central evaluation unit, which is not presented as part of this work.

Possible future work may involve the migration of the control of the modules and evaluation of the returned information in the PL itself.

### B. GPS module

The GPS Pmod module from digilent [17], which communicates with the Zynq-Z7 SoC using the UART interface, is used to determine the GPS-based data. In [7]–[9] a wide variety of approaches are presented that can be used to locate a position using GPS. The difference between GSM position localization as presented below is that GPS performs an absolute position determination. Due to the high sensitivity (−165 dBm) of the selected receiving chip, an accuracy of up to 3 m can be achieved with an update rate of 10 Hz [17]. An update rate of a localization routine is set to 60 s. The requested information of the GPS location is the satellite time, which is used as a time stamp, and the longitude and latitude.

In the event that GPS is unable to determine the location (e.g. the emulator is located in the interior), which is a valid response from the GPS module, the satellite time is set as 0.0 and the longitude and latitude is set as the letter N in the data transmission. (see Fig 4, in the lower message). Otherwise, the values of satellite time, longitude and latitude can be found after the corresponding title (see Fig 4, in the upper message).

After comparing 10 measurements of longitude and latitude by the GPS module with 10 measurements by a reference measurement system, the specified accuracy of the GPS module [17] can be confirmed.

### C. GSM module

The GSM module used is the GSM-Click product from microelectronics [18]. UART is used as the bus interface between the module and the Zynq-Z7. They are controlled using the AT command set. Three routines/functionality processes are implemented:

- **Unlock the SIM card and basic configuration**
- **RSSI measurement of the adjacent radio cells:**  
In [10, 11] the procedure of GSM location using RSSI measurement of 6 adjacent radio cells and the responsible radio cell is briefly described. Using the necessary AT command, the feedback of the GSM module (RSSI measurement) is saved in the DRAM. Using the measured signal strengths,

```
(a) Satellite Time: 111418.0; GPS Longitude:
47G48'29.4"N; GPS Latitude: 9G38'43.8"E;
GSM-Localization: 5 45 1 0 E-Plus;
#MONI: N1 77 E720 E796 980
-74dbm 25 19; #MONI: N2 70 E720
98A4 685 -75dbm 30 24; #MONI:
N3 74 E720 4B56 982 -79dbm 20
14; #MONI: N4 36 E720 4D2E 669
-81dbm 24 18; #MONI: N5 75 E720
9A7F 690 -82dbm 23 17; #MONI: N6
31 E720 8B39 984 -87dbm 18 12;

(b) Satellite Time: 0.000000; GPS Longitude: N;
GPS Latitude: N; GSM-Localization: #MONI:
S 74 E720 9976 1022 -60dbm 45 0
1 0 E-Plus; #MONI: N1 FF FFFF 0000
980 -76dbm -1 -1; #MONI: N2 FF
FFFF 0000 685 -79dbm -1 -1; #MONI:
N3 FF FFFF 0000 1024 -111dbm -1
-1; #MONI: N4 FF FFFF 0000 1024
-111dbm -1 -1; #MONI: N5 FF FFFF
0000 1024 -111dbm -1 -1; #MONI: N6
FF FFFF 0000 1024 -111dbm -1 -1;
```

Figure 4. Two transmitted localization data packets after transmission via SMS to the central evaluation unit with detectable gps localization data (a) and non-detectable GPS localization data (b). Note: Each line is separated by a semicolon. The satellite time is preceded by the word *Satellite Time*. First/first two characters indicate the hour, next two characters the minutes, next two characters the second, then a point 0 is connected. If no position information is available, a 0.0 is output. The longitude is preceded by the word *GPS Longitude*. The letter G is used instead of a degree sign. If there is no position information, an N is output. The latitude behaves analogously to this, which is preceded by the word *GPS Latitude*. GSM localization is preceded by the word *GSM localization*. Each line has the following structure: #MONI: Cell BSIC LAC CellId ARFCN Power C1 C2 (Ta RxQual PLMN). The expressions in brackets are only available for Cell S.

a relative position determination in the room is possible. This should take place on the central evaluation unit, which should not be presented in the context of this publication.

- **Transmission of the determined data via SMS:**  
For the transmission of the last determined position data (GPS and RSSI measurement), the required information is loaded from the two ring buffers, sent via SMS to the central evaluation unit using the AT command set and the coding form described in the caption of Fig 4. The successful transmission can be verified by feedback from the GSM module.

This routine is started by pressing a button on the evaluation board, which is queried at a frequency of 8 Hz (hand movement expected as soon as possible). Otherwise the position is localized with an update rate of 60 s.

More information about the AT command set can be found in [19].

An evaluation of 10 measurement results from GSM location with researched cell information and their strength show plausible values.

#### IV. CONCLUSION AND FURTHER WORK

In this publication, we have presented an emulator for an outdoor localization system for motor vehicle fleets, which uses GPS and GSM to determine position data and forwards them to a central processing unit via SMS. All the functionalities presented were implemented on a Zynq-Z7 SoC, which was hard-wired with the required components.

The bus system used to control the GPS and GSM peripheral module is UART. Using the AT command set, the signal strength of the adjacent transmitter cells could be measured and the measured data sent. Similarly, a command set was also implemented for the GPS module, which carried out a measurement of the satellite time, the longitude and the latitude. The data determined was then sorted in the DRAM using a ring buffer structure.

As a parallel project, this system will be modeled in SystemC to model the debug and security issues. Later, these models will enhance the presented emulator.

In the future, the functions presented will be integrated on a cheaper hardware platform, such as FPGA. To do this, the peripheral control must be moved from the PS to the PL. Experiments must also be carried out to determine the position information, and the individual functionalities must be transferred from a decentralized circuit board structure to a compact (one circuit board) structure.

#### REFERENCES

- [1] A. Narang, "The Role of Automatic Vehicle Localization (AVL) System in Fleet Management," <https://www.einfochips.com/blog/the-role-of-automatic-vehicle-localization-avl-system-in-fleet-management/>, 2016.
- [2] TrackBase GmbH, "GPS-Tracker-Vorteile," <https://auto-gps-tracker.de/gps-tracker-vorteile/>, 2019.
- [3] M. Bauer, *Vermessung und Ortung mit Atelliten - NAVSTAR-GPS und andere satellitengestützte Navigationssysteme* (4th edition), Herbert Wichmann Verlag, Heidelberg, Germany, 1997.
- [4] R. Karlsson, and F. Gustafsson, "The Future of Automotive Localization Algorithms: Available, reliable, and scalable localization: Anywhere and anytime," *IEEE signal processing magazine*, vol. 34, no. 2, 2017, pp. 60-69.
- [5] T. Strang, F. Schubert, S. Thölert, R. Oberweis, et al., "Lokalisierungsverfahren," Deutsches Zentrum für Luft- und Raumfahrt (DLR), Oberpfaffenhofen-Wessling, Germany, 2008.
- [6] GlobalView, "Fleet Tracking System, GPS Tracking Devices, Vehicle Tracking Applications, & Service," <https://www.global-view.net/>, 2019.
- [7] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, 2000, pp. 28-34.
- [8] N. M. Drawil, H. M. Amar, and O. A. Basir, "GPS Localization Accuracy Classification: A context-Based Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, 2013, pp 262-273.
- [9] B. Cheng, R. Du, B. Yang, W. Yu, C. Chen, and X. Guan, "An Accurate GPS-Based Localization in Wireless Sensor Networks: A GM-WLS Method," in *2011 40th International Conference on Parallel Processing Workshops*, 2011.

- [10] M. Ibrahim, and M. Youssef, "CellSense: An Accurate Energy-Efficient GSM Positioning System," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, 2011, pp. 286-296.
- [11] A. Varshavsky, M. Y. Chen, E. de Lara, J. Froehlich, D. Haehner, et al., "Are GSM Phones THE Solution for Localization?," in *Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06 Supplement)*, 2006.
- [12] Kao, C.-F., Chen, H.-M., Huang, I.-J. (2008): Hardware-Software Approaches to In-Circuit Emulation for Embedded Processors. *IEEE Design and Test*, 25 (5): 462 - 477
- [13] 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
- [14] Xilinx, "Zynq-7000 All Programmable SoC Technical Reference Manual (UG585)," v1.10, 2015.
- [15] Digilent, "Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board," <https://store.digilentinc.com/zybo-z7-zynq-7000-arm-fpga-soc-development-board/>, 2019.
- [16] Xilinx, "Zynq-7000 SoC Data Sheet: Overview (DS190)," v1.11.1, 2018.
- [17] Digilent, "Pmod GPS: GPS Receiver," <https://store.digilentinc.com/pmod-gps-gps-receiver/>, 2019.
- [18] MikroElektronika, "GSM Click," <https://www.mikroe.com/gsm-click>, 2019.
- [19] Telit, "AT Commands Reference Guide," Rev.24, 2016.



**Gregor Benz** received his B.Eng. degree in electrical engineering/physics from the Hochschule Ravensburg-Weingarten in 2018. Since 2018, he is pursuing his Master's degree at Educational University Weingarten.



**Andreas Siggelkow** received the academic degree Dipl.-Ing. degree in 1988 from the University of Karlsruhe. In 1996 he obtained his doctorate at the University of Stuttgart for Dr.Ing. Since 2007 he is professor for ASIC-Design and computer architecture at the Hochschule Ravensburg-Weingarten.

# Ein Geschwindigkeitsmesssystem nach dem Ortsfrequenzfilterverfahren im Bereich der funktionalen Sicherheit

Frank Wasinski, Werner Bonath, Ubbo Ricklefs, Josef Börcsök, Waldemar Müller, Eike Hahn, Michael Schwarz

**Zusammenfassung**—Weiterentwicklungen der Ortsfrequenzfiltervarianten weisen bei niedrigen Ortsfrequenzen eine ausreichende Dämpfung auf. Bei der Verwendung zur Geschwindigkeitsmessung kann damit auf den Einsatz von Tracking-Filtern verzichtet werden. Zur Demonstration der neuen Filtervarianten wird ein Messstand vorgestellt, bei dem sowohl die Sensorelektronik, bestehend aus einem Photodioden-Array mit nachgeschalteten Verstärkern, als auch die Filterelektronik diskret umgesetzt worden sind. Die Funktionsweise des Verfahrens kann damit bei unterschiedlichen Oberflächen und in einem Geschwindigkeitsbereich von 0 bis ca. 60 km/h gezeigt werden. Um das Messverfahren im Bereich der funktionalen Sicherheit verwenden zu können, wird eine sicherheitsgerichtete Systemarchitektur basierend auf einem Safety System on a Chip (Safety SoC) vorgestellt.

**Schlüsselwörter**—Optische Geschwindigkeitsmessung, Ortsfrequenzfilterung, Funktionale Sicherheit.

## I. EINLEITUNG

Optische Ortsfrequenzfilterverfahren kennzeichnen eine Klasse von Sensoren zur berührungslosen vektoriellen Geschwindigkeitsmessung. Sensoren nach diesem optischen Messverfahren können auf fast beliebig zufällig strukturierten Oberflächen eingesetzt werden, weisen keinen Schlupf auf und zeichnen sich durch eine geringe Messunsicherheit aus. Die Messsysteme nutzen hierfür eine spezielle Form der Korrelationsmesstechnik, um den Rechenbedarf und damit die Auswertzeit zu reduzieren. Geschwindigkeiten bis 500 km/h, aber auch sehr niedrige Geschwindigkeiten bis zur sogenannten Nullgeschwindigkeit, lassen sich mit dem Verfahren messen.

Sensoren nach dem Ortsfrequenzfilterverfahren werden seit vielen Jahren untersucht und in vielen Bereichen erfolgreich eingesetzt [1] [2] [3] [4]. Frühe Sensoren dieser Gattung verwendeten Gitter verschiedener Ausführung und Geometrie zur Ortsfrequenzfilterung.

Frank Wasinski, frank.wasinski@ei.thm.de, Werner Bonath, werner.bonath@ei.thm.de, Ubbo Ricklefs, ubbo.ricklefsh@ei.thm.de, Technische Hochschule Mittelhessen, Wiesenstraße 14, D-35390 Gießen.

Josef Börcsök, j.boercoek@uni-kassel.de, Waldemar Müller, w.mueller@uni-kassel.de, Eike Hahn, eike.hahn@uni-kassel.de, Michael Schwarz, m.schwarz@uni-kassel.de, Universität Kassel, Wilhelmshöher Allee 71 - 73, D-34121 Kassel.

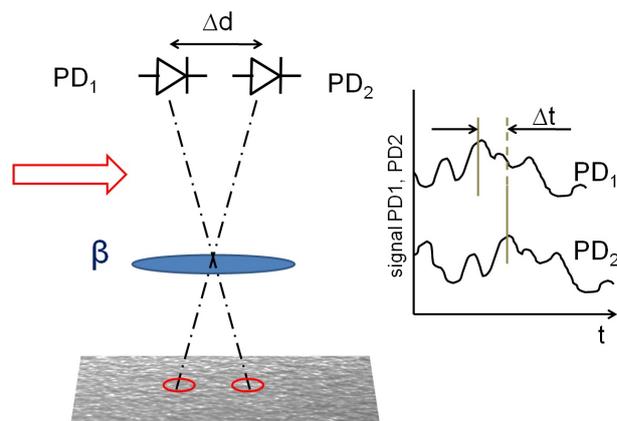


Abbildung 1. Korrelationsmessprinzip mit charakteristischen Signalverlauf.

Mit der einfacher und preiswerter werdenden Kamertechnik werden die Gitter zunehmend durch das Summieren von Pixeln einer Kamera nachgebildet. Wird dieses Summieren flexibel gestaltet, so lassen sich beliebige Ortsfrequenzfilter realisieren. Heute kommen Kamera- und nicht-Kamera-basierte Verfahren zum Einsatz. Aufwendige Kamera-basierte Verfahren korrelieren ganze Bilder aus der Bildfolge miteinander, indem zwei Bilder pixelweise in x- und y-Richtung gegeneinander verschoben, miteinander multipliziert und summiert werden. Dieser Vorgang ist extrem zeitaufwändig [5]. Durch die Ortsfrequenzfilterung kann diese Korrelation auf einen schmalen Ortsfrequenzbereich reduziert werden, was den Rechenaufwand erheblich verringert. Eher Bildverarbeitung-orientierte Sensoren bestimmen im Bild Merkmale, deren Lage anschließend in der Bildfolge verfolgt wird (Optischer Fluss). Diese Vorgehensweise ist schneller als die Korrelation, benötigt aber immer noch viel Rechenzeit [6]. Unabhängig von der Art der Auswertung sind Kamera-basierte Systeme durch die Auslesezyklen der Kamera in der maximal erreichbaren Verarbeitungsgeschwindigkeit begrenzt und deshalb zur Messung hoher Geschwindigkeiten sowie für regelungstechnische Anwendungen ungeeignet.

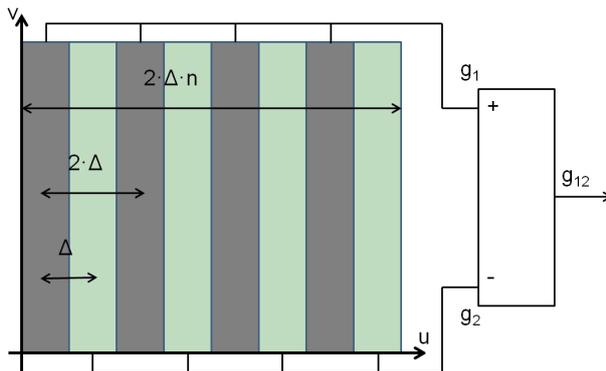


Abbildung 2. Gewichtete Signale  $g_1, g_2$  und das Summensignal  $g_{12}$  des Korrelationssensors.

## II. ORTSFREQUENZFILTERVERFAHREN

In der einfachsten Ausführung der Korrelationssensoren werden zwei Punkte einer Oberfläche über ein Objektiv auf zwei Fotodioden abgebildet. Wird der Sensor in Richtung der Verbindungslinie zwischen diesen Messpunkten bewegt, dann detektieren die beiden Empfänger das gleiche Signal zeitlich versetzt. Gemessen wird diese zeitliche Verschiebung  $\Delta t$ . Aus der Vergrößerung  $\beta$  und dem Abstand  $\Delta d$  der Empfänger, kann dann auf die Geschwindigkeit  $v$  geschlossen werden (Abbildung 1):

$$v = \frac{\Delta d}{\beta \cdot \Delta t} \quad (1)$$

Da die Verschiebung mit der geringsten Messunsicherheit aus hohen Ortsfrequenzen gewonnen werden kann, wird die Korrelation mit einer schmalbandigen Filterung für hohe Ortsfrequenzen gekoppelt. Realisiert wird dies dadurch, dass nicht nur zwei Punkte betrachtet werden, sondern eine größere Fläche, die dann in Streifen unterteilt wird. Die Informationen in diesen Bildstreifen werden dann im einfachsten Fall alternierend mit +1 und -1 gewichtet und summiert. Diese Wichtung und Summierung kann numerisch (oder Hardware basiert) durchgeführt werden, indem als Empfänger eine Flächenkamera Verwendung findet, deren Bildinhalt entsprechend bewertet wird (Abbildung 2). Vorteilhaft dabei ist, dass sich mit der Kamera durch das Zusammenfassen mehrerer Pixelspalten zu einem Streifen unterschiedlichste Ortsfrequenzfilter realisieren lassen. Handelstypische Sensoren zerlegen das Bild über ein Prismenarray in Streifen. In einigen Fällen werden auch Fotodiodenarrays bzw. strukturierte Fotodioden für die Bildbewertung verwendet. Eine Zeilenkamera ermöglicht hierbei einen guten Kompromiss aus Flexibilität und messbarer maximaler Geschwindigkeit.

Durch die schmalbandige Ortsfrequenzfilterung des Bildes erhält man bei der Bewegung des Sensors über eine Fläche als Ausgangsmesswert ein oszillierendes Signal  $g(x)$ , das in Bursts eingehüllt ist. In Abbildung

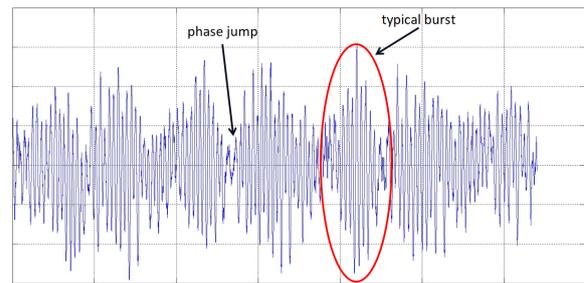


Abbildung 3. Burstartiges Sensorausgangssignal mit Phasensprung.

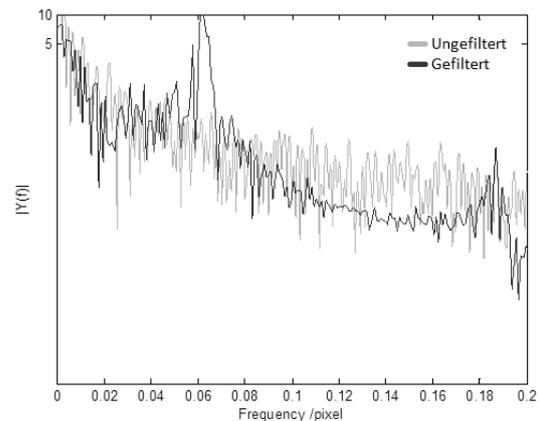


Abbildung 4. Amplitudenspektrum einer typischen Oberfläche (hell) und Filterspektrum der herkömmlichen Filterstruktur (dunkel).

3 ist die typische burstartige Signalform, aber auch die Modulation erkennbar. Ursache für die Bursts sind häufig vorkommende dominante Oberflächenobjekte, die beispielsweise durch eingeschlossene Steine in einer Asphaltoberfläche gebildet werden. Da diese Objekte zufällig im Bild erscheinen, erhält man zwischen den Bursts Phasensprünge, die maßgeblich die Messunsicherheit des Verfahrens beeinflussen. Die Trägerfrequenz des Signals wird zur Geschwindigkeitsmessung ermittelt.

### A. Analyse des Verfahrens

Das helle Spektrum in Abbildung 4 zeigt die Frequenzanalyse einer typischen Oberfläche ohne Filterung, d.h. ohne Einsatz eines Prismen- oder Fotodiodenarrays. Die niedrigen Ortsfrequenzen tragen den größten Anteil der Bildinformation. Die Filterung des Bildes durch einen Sensor - wie in Abbildung 2 - liefert zwar für hohe Ortsfrequenzen einen schmalbandigen Durchlassbereich, unterdrückt aber die niederfrequenten Signalanteile nicht (Dunkler Signalanteil). Die Entfernung dieser störenden niederfrequenten Ortsfrequenzanteile geschieht meist über die Filterung mit einem zusätzlichen elektronischen Bandpass-Filter, sogenannte Tracking-Filter, bei denen die Mittenfrequenz des Durchlassbereichs mit der gemessenen Geschwindigkeit mitgeführt wird. Als Folge dieser zusätzlichen

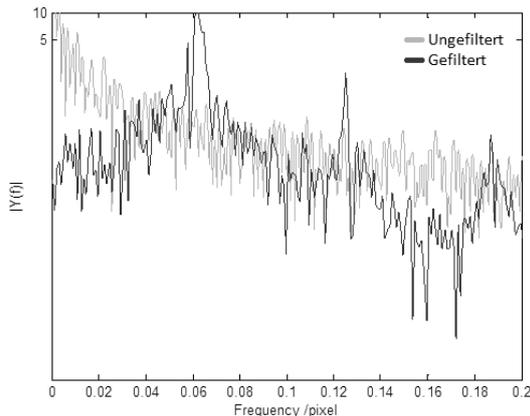


Abbildung 5. Amplitudenspektrum einer typischen Oberfläche (hell) und Filterspektrum einer neuen Filtervariante (dunkel).

zeitlichen Filtertechnik ist eine Messung bei Geschwindigkeit Null nicht mehr möglich. Erschwerend kommt hinzu, dass die schmalbandigen Tracking-Filter stets mit einem Zeitversatz nachgeführt werden, sodass diese Systeme nur eine eingeschränkte Dynamik aufweisen. Des Weiteren können periodische Ortsfrequenzmuster, die vom Ortsfrequenzfilter nicht ausreichend unterdrückt werden, einen hohen Signalanteil liefern, sodass das Tracking-Filter auf diese Frequenz einrasten kann. Eine Fehlmessung wäre die Folge.

### B. Neue Filter

Bisher werden mit strukturierten Fotodioden bzw. Fotodiodenarrays lediglich die bekannten Ortsfrequenzfilter nachgebildet. Die Empfängerfläche des Sensors wird dazu in  $n$  Streifen unterteilt. Die Ortsfrequenzfilterung erfolgt, indem die Streifen alternierend mit  $+1$  und  $-1$  als Wichtung  $w_k$  bewertet werden und dann als Summe den Messwert  $G_{12}$  bilden (Abbildung 2).

$$G_{12} = \sum_{k=1}^n w_k \cdot s_k \quad (2)$$

Durch die Verschaltung und Wichtung der Diodensignale lassen sich neue Ortsfrequenzfilter realisieren, die wesentlich bessere Filtereigenschaften aufweisen [7]. Ein einfacher Wichtungsfaktor besteht z.B. aus der Abfolge:

$$w = \{-1, +2, -1, \dots, -1, +2, -1\} \quad (3)$$

Das Filter unterdrückt wirkungsvoll niederfrequente Ortsfrequenzen (Abbildung 5). Als besonders vorteilhaft haben sich symmetrische Filtervarianten erwiesen, da sich auf diese Weise ein Ausgangssignal ergibt, das nahezu ohne Offset ist. Auf die bisher üblichen Tracking-Filter kann dann verzichtet werden. Damit

ist das Messen der Nullgeschwindigkeit und eine Stillstandserkennung möglich. Gleichzeitig kann eine hohe Dynamik des Systems erzielt werden.

Aus dem Signal eines einzelnen Ortsfrequenzfilters lässt sich der Betrag der Verschiebung, nicht aber das Vorzeichen ermitteln. Durch eine Verschiebung der Wichtungen lassen sich zusätzliche phasenversetzte Signale erzeugen, die zur Bestimmung der Bewegungsrichtung genutzt werden können. Anhand der phasenversetzten Signale lassen sich überdies auch die Phasensprünge erkennen und deren Auswirkung auf die Messung vermeiden.

Zusätzlich ist auch mittels Waveletanalyse eine Erkennung der Bursts und der damit verbundenen Phasensprünge möglich. Bereits das einfache Haar-Wavelet liefert überzeugende Ergebnisse, wie Untersuchungen gezeigt haben [8]. Komplexere Varianten verringern zwar die Messunsicherheit, erhöhen aber auch die benötigte Rechenleistung.

### C. Teststand

Die Entwicklung der Algorithmen und Filtervarianten erfolgte an Hand von Kamerabildern. Die Bilder wurden an unterschiedlichen Oberflächen auf einem Messtisch gewonnen, der als Messreferenz diente. Es sind so Testläufe möglich, bei denen die Messunsicherheit nur durch die Auflösung des Sensors (Fotodiodenarray oder Kamera) und die Optik begrenzt werden. Für Tests bei höheren Geschwindigkeiten wurde ein Messstand aufgebaut, bei dem ein Elektromotor das Hinterrad eines Fahrrads antreibt. Auf dem Hinterrad lassen sich beliebige Testoberflächen aufbringen. Die Kamera wurde durch eine Fotodiodenzeile ersetzt. Um die Gleichzeitigkeit aller Signale sicherzustellen, wurde eine Fotodiodenzeile ausgewählt, bei der alle Fotodioden parallel nach außen geführt sind und nicht seriell, wie es bei CCD/CMOS-Sensoren üblich ist. Verwendet wurde eine Fotodiodenzeile von Hamamatsu, bei der die Gesamtanzahl der Einzelelemente zwar geringer ausfällt, die aber eine größere optisch wirksame Fläche bietet. Die nachgeschalteten Transimpedanzwandler erlauben so, bei noch moderaten Verstärkungsfaktoren, eine erhöhte Bandbreite. Das Leiterplattendesign erwies sich trotzdem als anspruchsvoll, da es durch die diskrete Ausführung jedes Diodenkanals schwierig wird, die gleichen Rahmenbedingungen für alle Kanäle sicherzustellen. Dies betrifft insbesondere die Leiterbahnlänge von den jeweiligen Fotodioden zu den Transimpedanzverstärkern und die damit einhergehende parasitäre Kapazität, sowie die mögliche Störanfälligkeit. Eine weitere nachgeschaltete Spannungsverstärkerstufe erlaubt es über Aufteilung der einzelnen Verstärkungsfaktoren, ein Optimum für die Gesamtverstärkung, den Signal-Rausch-Abstand und die Bandbreite zu finden.

Der optische Aufbau besteht primär aus einem bildseitig telezentrischen Objektiv, das sich durch eine

verzeichnungsarme Abbildung und ein großes Bildfeld auszeichnet. Bedingt durch das verhältnismäßig große Fotodiodenarray, tritt bei dem Aufbau trotzdem ein Vignettierungseffekt auf, der über die jeweiligen Verstärkungsfaktoren der einzelnen Kanäle kompensiert wurde. Somit bestimmen die an den Rändern gelegenen Kanäle, die maximale Bandbreite der gesamten Anordnung. Die Ortsfrequenzfilter sind ebenfalls in Hardware über die Verschaltung und Gewichtung der Fotodiodenkanäle ausgeführt. Es wurden mehrere Filtervarianten und der jeweils dazu passende, um eine Diode versetzte Filtertyp, realisiert.

Der Vorteil der komplett diskreten und ungetakteten Ausführung aller Kanäle liegt in der hohen Dynamik des Systems, die letztlich nur noch über die elektronische Bandbreite der Verstärker begrenzt wird. Bandbreiten bis in den MHz-Bereich sind damit erreichbar. Über die Bildstreifenbreite (Fotodiodenbreite), Vergrößerung und Bandbreite lassen sich Messbereich und Auflösung einstellen. Wählt man eine Bandbreite von 1 MHz, eine Vergrößerung von 1:5 und eine Diodenbreite von  $50 \mu\text{m}$ , so sollten sich Geschwindigkeiten bis ca. 500 km/h bei einer Auflösung von 0,5 mm messen lassen. Getestet wurde der Sensor, aufgrund der mechanischen Limitierung des Teststands, bis ca. 60 km/h. Die Analyse und Auswertung der Daten erfolgte mit MATLAB unter Zuhilfenahme eines passenden AD-Wandlers.

### III. FUNKTIONALE SICHERHEIT

Im Bereich der funktionalen Sicherheit sollen die Gefährdung von Menschen und Umwelt so gering wie möglich gehalten werden. Zur Erreichung dieses Ziels muss die Wahrscheinlichkeit eines unentdeckten Fehlers, der zu einem gefährlichen Ausfall eines Systems führt, unterhalb einer genau definierten Grenze liegen. Effektiv handelt es sich dabei also um eine Risikoreduzierung. Der Sicherheitslevel wird in der Sicherheitsnorm IEC 61508 in vier Stufen, von der niedrigsten Anforderungsstufe Safety Integrity Level 1 (SIL), bis zur höchsten Stufe SIL4 unterschieden. In einem aufwendigen Entwicklungsprozess, muss zur Ermittlung der Gesamtausfallwahrscheinlichkeit jede Komponente des Systems, hinsichtlich seines Fehlerpotentials bewertet werden und eventuelle Maßnahmen zur Reduzierung des Risikopotentials getroffen werden. Dies betrifft die Hard- und Software, sowie die Produktionskette bzw. die damit verbundenen Qualitätssicherungsmaßnahmen. Eine durchgängige Dokumentation aller Schritte des Entwicklungsprozesses, aber auch die Anwendung empfohlener Maßnahmen, wie das V-Modell, Failure Mode and Effects Analysis (FMEA), Fehlerbaumanalyse, etc., helfen, die hohen Anforderungen zur Erreichen, die an ein sicherheitsgerichtetes System gestellt werden [11] [12].

Um diesen Aufwand zu reduzieren, ist auch der Einsatz bereits zertifizierter oder vorbereiteter SIL-

Komponenten möglich. Bei Kombination unterschiedlicher Sicherheitslevel, bestimmt hierbei das kleinste Level innerhalb einer Kette das Gesamtlevel. Eine redundante Anordnung kann hingegen, unter Beachtung geeigneter Architekturmaßnahmen, das Level erhöhen.

Neben den redundanten Strukturen, kommen in diesem Bereich auch häufig integrierte Diagnoseschaltungen zur Anwendung, die Teile des Systems und/oder das Gesamtsystem mit definierten Algorithmen auf seine Funktionsfähigkeit testen. Wird ein Fehler erkannt, muss dieser Zustand entweder einer übergeordneten Instanz angezeigt werden oder er führt zu einer unmittelbaren Abschaltung bzw. Verriegelung des überwachten Systems, sodass das System in keinen gefährlichen Zustand kommen kann.

Besonders kritisch sind Ausfälle in Folge gemeinsamer Ursache (Common Cause Failure) zu werten, da sie auch redundante Strukturen betreffen können. Da sich grundsätzlich für jede Technologie Fälle finden lassen, die zu einem Ausfall des Systems führen können - im schlimmsten Falle sogar zu einem gefährlichen Ausfall -, besteht eine wirksame Gegenmaßnahme im Einsatz diversitärer Technologien, optimalerweise gekoppelt, mit einem hohen Diagnosegrad.

#### A. Sichere Geschwindigkeitsmessung

Eine Komponente zur Geschwindigkeitsmessung ist häufig im Bereich der funktionalen Sicherheit vorzufinden. Sei es im Automotive-Sektor, im Bahn-Sektor oder in der Prozessindustrie bei beweglichen Gütern.

In den beiden erstgenannten Bereichen werden zu diesem Zweck häufig Raddrehzahlsensoren auf magnetischer Basis eingesetzt. Tritt bei diesen Systemen Schlupf an einzelnen Rädern auf, kann über eine Differenzanalyse der Drehzahl-Informationen eine Stabilisierung des Fahrzeugs erreicht werden. Kommt es bei einem überwiegenden Teil der Räder zu Schlupf, kann jedoch keine gesicherte Aussage mehr über die tatsächlich gefahrene Geschwindigkeit getroffen werden. Da das hier beschriebene Ortsfrequenzfilterverfahren die Oberfläche direkt und berührungslos auswertet, kann es auch in dieser Extremsituation eine eindeutige Geschwindigkeitsinformation liefern. An seine Grenzen kann das optische Verfahren zum Beispiel bei ungünstigen Wetterbedingungen kommen (Fliegender Schnee, Wasser oder Laub auf der Straße), bei denen GPS und Radsensoren aber noch zuverlässige Werte liefern. In der Summe lässt sich durch beide diversitären Systeme, ein sichereres Gesamtsystem erstellen.

Ein verhältnismäßig neuer Bereich für die funktionale Sicherheit, erschließt sich seit einigen Jahren durch autonome Fahrssysteme. Da hier die übergeordnete Instanz in Form eines Menschen fehlt, müssen die Fahrssysteme zusätzlich abgesichert sein, um während der Fahrt keine Gefahr darzustellen. Auch hier kommt dem Geschwindigkeitsmesssystem eine hohe Priorität zu, da die derzeit verbauten Steuerungssysteme neben

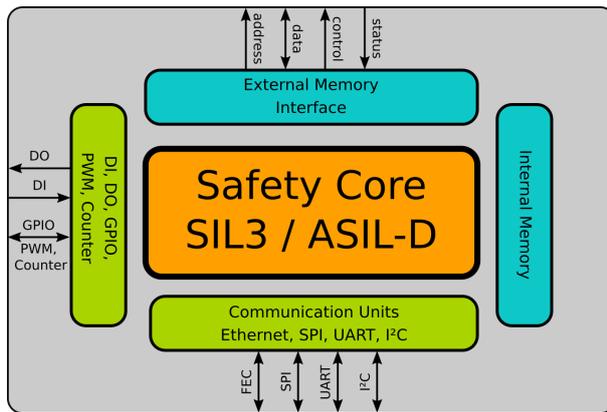


Abbildung 6. Blockschaltbild des Sicherheitsprozessors ReSCUV1.

der Auswertung von meist aufgenommenen Bildinformationen, eine möglichst große Anzahl weiterer Informationen benötigen, um sich sicher zu orientieren. Bewegen sich die autonomen Fahrsysteme in einem von Menschen zugänglichen Bereich, ist eine hohe Positioniergenauigkeit, aber auch eine schnelle Systemreaktion gefordert. Das hier vorgestellte Messverfahren erweist sich in dieser Anwendung als besonders vorteilhaft, da es Nullgeschwindigkeiten mit hoher Präzision und damit auch den Stillstand erfassen kann.

### B. Sicherheitsgerichtete Architektur

Um das Messverfahren im Bereich der funktionalen Sicherheit verwenden zu können, ist eine Messkette erforderlich, deren Komponenten für sicherheitsgerichtete Systeme geeignet sind. Für die digitale Verarbeitung der analogen Informationen des Sensors, wurde daher der bestehende Messaufbau um einen sicherheitsgerichteten Prozessor nebst AD-Wandler ergänzt, der von dem Institut für Computer Architektur und Systemprogrammierung der Universität Kassel zur Verfügung gestellt wurde.

Bei dem Prozessor handelt es sich um das Modell ReSCUV1 (Abbildung 6), der speziell nach den Kriterien der IEC 61508 bzw. ISO 26262 am Fachgebiet entwickelt und bei Europractice gefertigt wurde. Die Systemarchitektur ist als asynchrones Multiprozessor-system ausgelegt und verfügt über ein 1oo2D (one out of two mit Diagnose) Sicherheitskonzept, welches eine Verwendung bis zu SIL3 nach IEC 61508 bzw. ASIL-D (Automotive SIL) nach ISO 26262 ermöglichen kann [9] [10]. Die redundante Architektur, wie im Floorplan in der Abbildung 7 zu sehen, besitzt dabei eine nahezu vollständige Hardwareredundanz, was gegenüber klassischen Lockstep-Architekturen, die lediglich einen redundanten Prozessor-Kern haben, einen wesentlichen sicherheitstechnischen Vorteil darstellt.

In Abbildung 8 ist der Prozessor auf dem aktuellen Evaluation Board erkennbar. Die redundante Architektur drückt sich auch auf der Leiterplatte durch die Anordnung der Peripheriekomponenten aus.

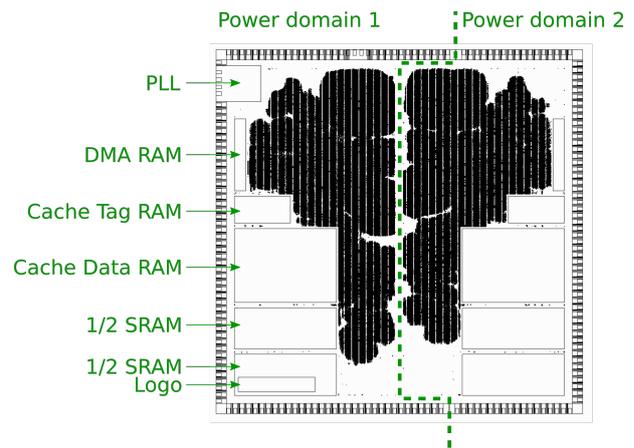


Abbildung 7. Floorplan des Sicherheitsprozessors ReSCUV1.

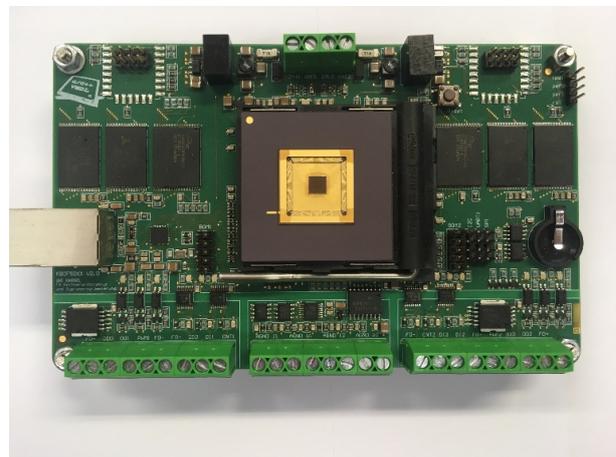


Abbildung 8. Sicherheitsprozessor ReSCUV1 auf Evaluation Board.

Die Auswertung der Geschwindigkeitsinformationen wurde in den Prozessor implementiert und konnte bereits erfolgreich getestet werden. Da es sich bei dem Geschwindigkeitssensor bisher um kein sicherheitsgerichtetes System handelt, bestehen die nächsten Entwicklungsschritte in der Untersuchung der verwendeten Architektur hinsichtlich möglicher Gefahrenpotentiale und um eine Erweiterung sicherheitstechnischer Merkmale. In Frage kommt auch hier ein redundanter Aufbau, der mit einer Diagnoseeinrichtung ergänzt wird. Die rein analoge Struktur mit größtenteils diskreten Bauelementen ermöglicht eine transparente Gefährdungsanalyse, stellt aber auch besondere Anforderungen an die Diagnoseeinrichtung.

## IV. FAZIT

Das in dieser Arbeit vorgestellte Geschwindigkeitsmesssystem nach dem Ortsfrequenzfilterverfahren, eignet sich, aufgrund neu erdachter Filterstrukturen, für eine Geschwindigkeitsmessung über einen besonders weiten Messbereich. Die bei diesem Verfahren üblicherweise verwendeten zeitlichen Tracking-Filter sind nicht mehr notwendig. Somit sind auch Messungen von

Kriechbewegungen, der sogenannten Nullgeschwindigkeit, sowie regelungstechnische Anwendungen möglich, bei denen eine erhöhte Dynamik des Systems gefordert ist. Der beschriebene Messstand demonstriert diese Eigenschaften in besonders ausgeprägter Form, da der gesamte Sensor bis zum Ausgangssignal in analoger Form, ohne getaktete Elemente, realisiert ist. Begrenzender Faktor ist primär nur die elektronische Bandbreite. Erst die anschließende Auswertung bestimmt über den nachfolgenden AD-Wandler und das Analysesystem engere Grenzen. Der Demonstrator ermöglicht eine Messung von 0 bis ca. 60 km/h bei einer Auflösung von 0,5 mm. Für eine angedachte Anwendung des Geschwindigkeitssensors im Bereich der funktionalen Sicherheit, wurde das System um einen sicherheitsgerichteten Prozessor nach IEC 61508 und einen AD-Wandler erweitert, in dem die Auswertung des Ausgangssignals bereits implementiert wurde und erfolgreich getestet werden konnte. Die nächste Phase besteht in der sicherheitstechnischen Bewertung und Erweiterung des bestehenden Sensors.

#### LITERATURVERZEICHNIS

- [1] *Ator, J.T., Image-velocity sensing with parallel-slit reticles, J.Opt.Soc.Am., Vol.53, No. 12, pp. 1416-1422, 1963.*
- [2] *Delingart, E., Berührungslose optische Geschwindigkeits- und Abstandsmessung, Leitz-Mitteilungen aus Wissenschaft und Technik, BD IV, Nr.7, S.249-257, 1976.*
- [3] *Kreutzer, P., Theoretische Betrachtungen zur berührungslosen Geschwindigkeitsmessung mit optischen Gittern, Feinwerktechnik und Messtechnik, Heft 6, S.289-294, 1975.*
- [4] *Aizu, Y., Asakura, T., Spatial Filtering Velocimetry, Springer-Verlag, 2006.*
- [5] *Hardin, L., Nash, L., Optical Velocity Measuring System Using Two Cameras, EP 1 067 388 A1, 2006.*
- [6] *Norskog, A.C., Combined Pointing Device and Bar Code Scanner, US 6 585 158 B2, 2003.*
- [7] *Ricklefs, U., Wasinski, F., Verfahren zur Messung von Wegen und Geschwindigkeiten, DE 10 2014 007 291 A1, 2015.*
- [8] *Ricklefs, U., Wasinski, F., Optische Sensoren zur Messung von Weg und Geschwindigkeit, Photonik Heft 5, S.42-45, 2014.*
- [9] *Hayek A., Börcsök J., 2013, "Safety-Related ASIC-Design in Terms of the Standard IEC 61508", PESARO 2013, The Third International Conference on Performance, Safety and Robustness in Complex Systems and Applications, Venedig, Italy, 21-26 Apr. 2013, ISBN: \*978-1-61208-268-4.*
- [10] *Hayek A. and Börcsök J., 2016: Miniaturized Safety PLC on a Chip for Industrial Control Applications, 13th International Conference of Distributed Computing and Artificial Intelligence (DCAI16), June 01-04. Sevilla, Spain.*
- [11] *Börcsök, J. Electronic Safety Systems - Hardware Concepts, Models and Calculations, Hüthig-Verlag Heidelberg, 2004, ISBN 3-7785-2944-7.*
- [12] *Börcsök, J. Funktionale Sicherheit - Grundzüge sicherheitstechnischer Systeme, VDE-Verlag Berlin - Offenbach, 3. Auflage 2011, ISBN 978-3-8007-3305-7.*



**Frank Wasinski** erhielt 2008 sein Diplom im Fachgebiet Mikroelektronik / Elektronikdesign an der Technischen Hochschule Mittelhessen, sowie 2011 den Masterabschluss an der FernUniversität in Hagen. Seit 2008 war er in mehreren Entwicklungsprojekten tätig. U.a. der Entwicklung eines optischen SIL3-Drehgebers nach dem Moiré-Prinzip für sicherheitskritische Bereiche in Kooperation mit der Wachendorff Automation GmbH & Co KG, sowie der

Bewegungsdetektion basierend auf der Ortsfrequenzfilterung unter Anwendung von Waveletfunktionen. Von 2015 bis 2016 arbeitete er als Entwicklungs-Ingenieur bei Feig Electronic an der Weiterentwicklung eines Lichtgitters. Seit 2016 ist er wissenschaftlicher Mitarbeiter am Fachbereich Elektro- und Informationstechnik der Technischen Hochschule Mittelhessen. Seine Forschungsinteressen liegen im Bereich der Opto- und Mikroelektronik, Bildverarbeitung sowie der funktionalen Sicherheit.



**Waldemar Müller** erhielt 2016 seinen Bachelor und 2018 seinen Master im Fachgebiet Mechatronik an der Universität Kassel. Bereits in der Bachelorarbeit beschäftigte er sich mit Softcore Prozessoren auf FPGA Systemen. Im Rahmen seiner Masterarbeit war er maßgeblich an der Entwicklung eines funktional sicheren SoC beteiligt, welches im Anschluss an die Arbeit über das Europractice Programm in einem 180 nm Prozess gefertigt wurde. Seit 2018 ist er

wissenschaftlicher Mitarbeiter im Fachgebiet Rechnerarchitektur und Systemprogrammierung an der Universität Kassel. Dort forscht er an neuen Strategien und Konzepten für funktional sichere SoC.



**Werner Bonath** studierte Elektrotechnik an der RWTH Aachen und der TH Darmstadt, wo er 1989 am Fachgebiet Mikroelektronische Schaltungen und Systeme promovierte. Anschließend war er bei Boehringer Mannheim in der Geräteentwicklung/Diagnostik tätig. Sei 1993 lehrt und forscht er an der Technischen Hochschule Mittelhessen auf dem Gebiet des mixed Signal IC-Designs. Er ist Senior Member des IEEEs.



**Eike Hahn** erhielt 2016 seinen Bachelor und 2018 seinen Master im Fachgebiet Mechatronik an der Universität Kassel. Bereits in seiner Bachelorarbeit beschäftigte er sich mit dem Thema der funktionalen Sicherheit. Darin evaluierte er verschiedene Möglichkeiten der Anbindung einer singulären Kommunikationsschnittstelle an ein redundantes Prozessorsystem. Im Rahmen seiner Masterarbeit war er maßgeblich an der Entwicklung eines funktional sicheren

SoC beteiligt, welches im Anschluss an die Arbeit über das Europractice Programm in einem 180 nm Prozess produziert wurde. Seit 2018 ist er wissenschaftlicher Mitarbeiter im Fachgebiet Rechnerarchitektur und Systemprogrammierung an der Universität Kassel. Dort forscht er an neuen Strategien und Konzepten für funktional sichere SoC.



**Ubbo Ricklefs** promovierte 1985 am Lichttechnischen Institut der Universität Karlsruhe. Während seiner Industrietätigkeit entwickelte er Flüssigkristallzellen, Wanderfeldröhren und optische Partikelzähler. Von 1985 bis 1990 leitete er bei der Ernst Leitz Wetzlar GmbH die Entwicklung optischer Messmaschinen. Von 1990 bis 2019 lehrte er als Professor an der TH-Mittelhessen im Bereich der Elektrotechnik Technische Mechanik, Elektromechanische

Konstruktionen, Bildverarbeitung und Optoelektronik. Von 2008 bis 2018 war er Leiter des Zentrums für „Nanotechnik und Photonik“ der THM.



**Josef Börcsök** studierte in Darmstadt und Kassel. Er promovierte 1995 und habilitierte 2002. Er war mehr als 20 Jahre lang in führenden Positionen in den Bereichen Avionik, Industriesteuerung und Sicherheitsrechnerarchitektur. Seit 2005 ist er Universitätsprofessor und Leiter des Fachgebiets Rechnerarchitektur und Systemprogrammierung an der Universität Kassel. Die Forschungsschwerpunkte liegen im Bereich Modellierung sicherheitsrelevanter

Rechnertechnologie, innovative Sicherheitsarchitekturen, Sicherheitssysteme auf Chip-Technologie (SSoC), Echtzeitsysteme, sicherheitsbezogene Kommunikation und Protokolle. Er arbeitet seit Jahren in mehreren nationalen und internationalen Normungsausschüssen und ist Mitglied der DKE-Ausschüsse, IEEE, IET und BCS sowie einer der ersten Sicherheitsexperten des TÜV-FSE-Programms. Er hat mehr als 250 Veröffentlichungen in den referierten internationalen Zeitschriften und schrieb 12 Bücher.





# Hybrid Image Processing MPSoC Architecture for Cloud-based Camera Monitor Systems

Jannik Maier, Johannes Schäfer, Anestis Terzis, Oliver Bringmann

**Abstract**—A cloud-based research platform is designed using MPSoC-FPGAs. This platform shall be used to validate the possibilities of cloud-based real-time applications. Therefore, a well-defined scenario is used to implement and validate such a platform exemplarily. The platform is implemented at the example of a Camera-Monitor-System for mirror-replacement. This paper focuses on power consumption and delay of real-time image data processing and transmission.

**Index Terms**—Camera-Monitor-System, cloud, real-time, Multiprocessor System on Chip (MPSoC), Hybrid Image Processing, FPGA, Xilinx Vivado.

## I. INTRODUCTION

Cloud-based data processing has become more important within the last decade. Scalability, distributivity, interoperability, modularity are some of the main advantages of cloud computing [1]. The consideration of new cloud-based architectures is a promising approach because of the increasing processing requirements for advanced automotive use cases [2]. Within this project, a novel hybrid cloud-based processing platform is designed and exemplarily implemented. The main purpose of the platform is to enable the investigation and identification of selected parameter of real-time cloud-based processing architectures for the aforementioned applications. The chosen example architecture is the implementation of a Camera-Monitor-System (CMS) as Mirror replacement [3]. The existence of a defined scenario improves the validation possibilities of the concept. This is due to the fact that there are well defined requirements to this scenario like the ISO 16505 [4] and the UN Regulation Number 46 [5]. Consequently a conclusion can be drawn, which states, whether the requirements can be finally achieved with this modern architecture.

The implementation process starts with the definition of two function sets. One set includes basic and critical functions. The outsourcing of these functions is critical because of their requirement on permanent availability. The second set includes outsourceable advanced functions. These functions could run in a cloud or

Jannik Maier, Jannik.Maier@thu.de, Johannes Schäfer, joschaefer@mail.thu.de, Anestis Terzis, Anestis.Terzis@thu.de, Technische Hochschule Ulm.

Oliver Bringmann, Oliver.Bringmann@uni-tuebingen.de, Eberhard Karls Universität Tübingen.

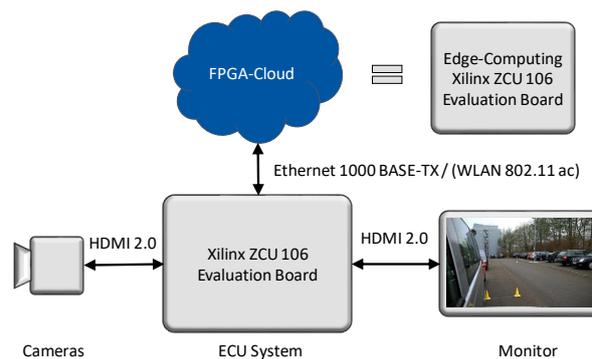


Figure 1. The designed concept. This concept consist of mainly four parts. The camera for image capturing. The monitor for the visualization. The local ECU System, which handles the local tasks, is implemented on a Xilinx ZCU 106 Evaluation board. The cloud or edge System for the advanced tasks is also implemented on second Xilinx ZCU 106 evaluation board.

edge because of their non-critical tasks. For the mirror replacement with a CMS, the basic function is the visualization of the image. This function must run locally to guarantee availability and the mandatory safety level of the system. The regulatory requirements do not demand features like an object detection or scene interpretation. Therefore, the outsourcing of these tasks into a cloud or edge is investigated.

Figure 1 shows the overall design concept. The design consists of four parts. The two main parts are the MPSoC-based ECU (electronic control unit) system and the FPGA-based cloud or edge system. The first functional task of this project is a real-time object detection in a cloud or edge device. This paper concentrates on the validation of the real-time transmission of the coded video signals into the cloud and the power consumption of both systems. Therefore, the image quality of the transmitted data, the time for the transmission and the power consumption in the local and the cloud system are discussed.

The next chapter introduces the development process of the system. The following chapters define the first use cases for the system, and discuss the power consumption, the latency and the system utilization. At the end a conclusion is drawn.

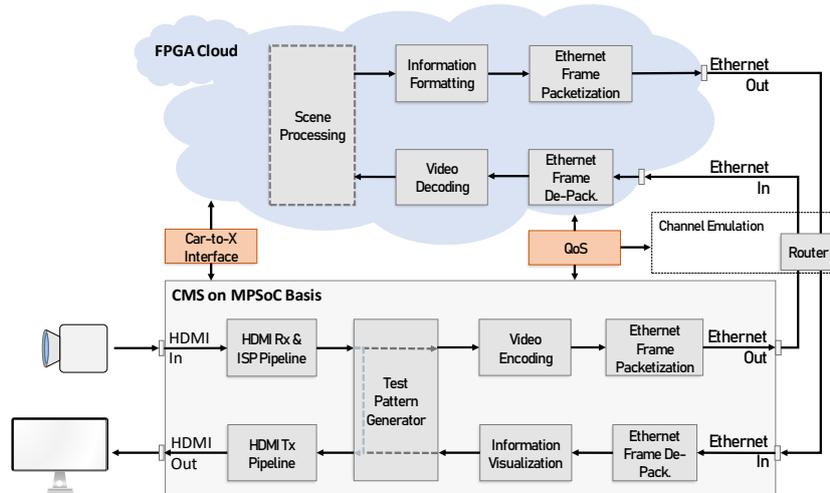


Figure 2. Functional concept of a hybrid image signal processing platform. External interfaces are marked orange.

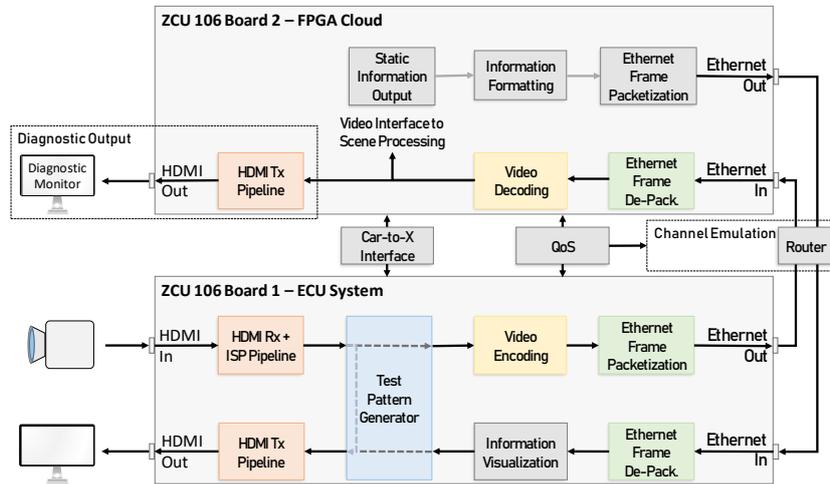


Figure 3. Functional prototype design concept of a hybrid processing CMS using two Zynq UltraScale+ ZCU106. HDMI interfaces in orange, cloud information interface in blue, video coding interface in yellow, packetization process in green.

## II. DEVELOPMENT PROCESS

The introduced platform is realized with two Xilinx Zynq UltraScale+ MPSoC ZCU106 [6] boards. These boards are used because of their chipset in combination with the provided connectivity. The main advantage is an integrated Video Codec Unit (VCU) [7] for real-time image coding. In addition, these boards enable prototype designs for various embedded vision applications. The development platform, in addition, is supposed to enable the channel emulation between the local ECU system and the FPGA-based cloud (see Figure 1). Channel Emulation shall include for example connection interrupts, packet loss and jitter.

### A. Prototype Design Concept

Figure 2 shows a functional design of the hybrid image signal processing (ISP) platform. The prototype design concept in Figure 3 utilizes two Zynq UltraScale+ ZCU106 boards as development platforms.

Colored blocks indicate elements of primary focus in reference to the development process. Grey blocks indicate elements with lower priority for the first design step.

The design for the ECU system is based on the Xilinx VCU HDMI capture design [7]. The design for the emulated FPGA cloud is based on the Xilinx VCU HDMI transmitter design [8].

The first ZCU106 board emulates the vehicle's ECU system (see Figure 1). A video stream is received from a camera and processed in order to be displayed immediately, without additional processing, on a monitor for local CMS functionality. The video stream is received and displayed via the board's HDMI interfaces. Simultaneously, the video stream is forwarded to the cloud with low latency. Encoding the forwarded video stream is used to reduce the transmitted data rate. Both boards shall transmit data via UDP over their Ethernet interfaces in the prototype stage. For the final stage,

wireless interfaces like 5G networks [9] are going to be validated.

The second ZCU106 board represents a cloud or edge emulation unit. The video stream, which is sent by the ECU system, is supposed to be accessible for future scene processing algorithms. Information about identified objects of interest shall be returned to the ECU system and overlaid on top of its video output. An additional video output on the second board provides access to analyze the quality as well as the latency of the received video stream. To access the ECU system's video stream, incoming data have to be depacketized and decoded. The received video stream will be forwarded to the additional video output and can be accessed by future object processing algorithms [10]. A video interface for scene processing is planned to be prepared. Moreover, a static information output will be included in order to access the object information output. The information output shall then be formatted and packetized to be forwarded to the ECU system. Detected objects will be visualized with overlays within the images. A synthesized picture or augmented reality picture that contains all detected objects shall be implemented in future work.

### B. Video Codec Unit

The video codec unit (VCU) IP core [7] enables video encoding and decoding. It supports both the H.264 and H.265 standard [11]. A maximum video input of 3840 x 2160 px at 60 fps is supported and limited to the following characteristics:

- color space: YCbCr 4:2:2, 4:2:0, 4:0:0
- color depth: 8 bit or 10 bit

The VCU is a hard IP block integrated inside the Programmable Logic (PL) with no direct connections to the Processor System (PS). "VCU operation requires the APU to service interrupts to coordinate data transfer. VCU applications running on the APU use Xilinx VCU Control Software library API to interact with encoder and decoder microcontroller" [7].

This project uses the provided prebuilt device topologies by Xilinx and configures their interfaces to fit the video input and output devices. After the configuration of the video interface, the framework GStreamer [12] is used to transport the video stream from the ECU system to the emulated FPGA cloud or edge system. All the following descriptions can be found in GStreamer documentation [12]. The following code snippets show parts of the basic parameter settings of an advanced video streaming pipeline.

```
video/x-raw, format=NV12, width=1920, height=1080, framerate=60/1
```

The video format is set in this line. Since the video is compressed through the video input pipeline, a suitable format, NV12, is selected. The NV12 format fits to

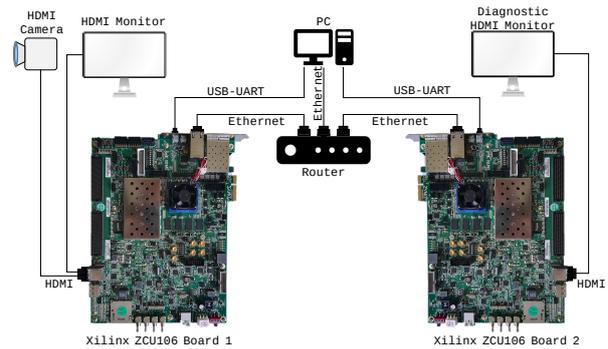


Figure 4. Test setup with two Zynq UltraScale+ ZCU106, which are connected to HDMI peripherals, PC and router.

a 4:2:0 YCbCr 8 bit video stream. Moreover, the matching resolution and framerate for the input device interface is chosen. The most basic factors that can be set in the encoding process using GStreamer are the following:

```
bitrate = 10000000, control-rate=constant
```

A variable bitrate should be avoided for live streaming to prevent inconsistent buffering. Instead, a constant bitrate is chosen by setting the property "control-rate" to "constant". With future application limitations in mind, a bitrate of 10 Mbps is selected.

```
gop-mode = low-delay-p
```

A sequence of frames is called group of pictures (GOP). The "gop-mode" is set to "low-delay-p" to adjust for low latency streaming and high compression rate. Therefore, the frame sequence contains mainly predictive frames which reduce data rate.

A test setup is realized based on two Zynq UltraScale+ ZCU106 boards (see Figure 4) to serve as a development platform. As described before, the first board emulates the ECU system and the second board emulates the FPGA cloud. Both boards are connected to the required interfaces that were introduced in the overall concept (see Figure 2). Additionally, both boards are connected to a USB-UART interface, which allows command window access from a PC to both boards. Moreover, a router operates as a network instance and connects both boards and a PC via Ethernet.

### III. USE CASES

This section examines the behavior of the system using the developed prototype design in different use cases. An optimized hardware profile is created for each video profile in order to investigate its power consumption. The system utilization on chip and its routing is presented for the most demanding video profile. Also, the system latency is determined for each video profile.

The system behavior of streaming raw and encoded video signals shall be compared. The raw video signal

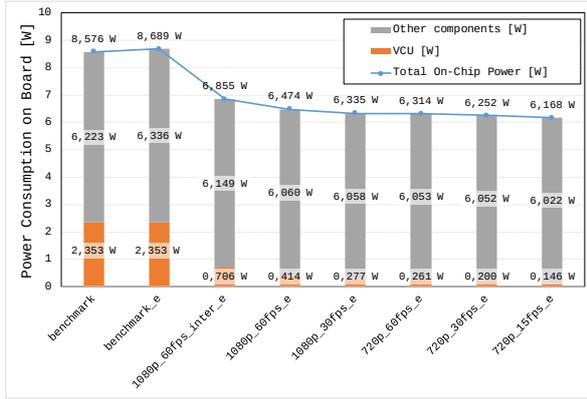


Figure 5. Power consumption of the ECU system for each hardware profile.

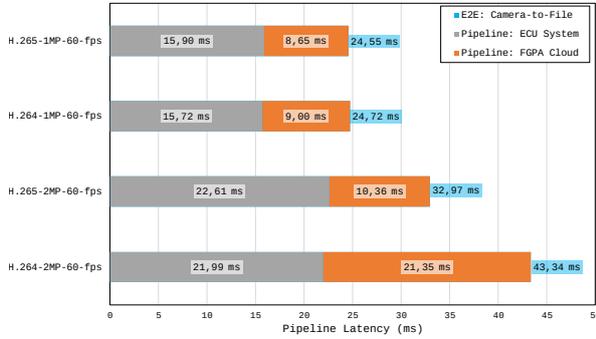


Figure 6. Concatenated GStreamer pipelines of the ECU system and emulated FPGA cloud that show the pipeline latency from camera to file-sink.

is defined as RGB, 4:4:4, 8 bit format. The compressed video signal shall be defined as YCbCr 4:2:0, 8 bit format. During the course of measurement it became apparent that the raw video format is not supported by the current prototype-system. The available v4l2 camera driver only supports the compressed video output format for the used HDMI-camera.

#### IV. POWER CONSUMPTION

The system's power consumption varies with different hardware profiles as shown in Table I. Inter-coding can lead to a lower data rate with increased latency. A series of hardware profiles is used to match the specifications of the image processing profiles, as seen in Table II.

No hardware profile can be derived from the display video profile "Profil1", since it is not supported by the available Xilinx IP cores used in this project. IP cores of each hardware designs are configured in such a way that they can serve the respective video profile to the maximum. Comparable values for the hardware design gradations of each hardware profile are shown in Figure 5.

The hardware profile "1080p\_60fps\_e" offers a good trade-off between power consumption and video signal

Table I  
 MODIFIED HARDWARE PROFILES FITTED TO THE IMAGE  
 PROCESSING PROFILES OF TABLE II.

Hardware Profile	Resolution (px)	Refresh Rate (fps)	Inter or Intra Coding
2MP-60fps-inter	1920 × 1080	60	inter
2MP-60fps-intra	1920 × 1080	60	intra
2MP-30fps	1920 × 1080	60	intra
1MP-60fps	1280 × 720	60	intra
1MP-30fps	1280 × 720	30	intra
1MP-15fps	1280 × 720	15	intra

Table II  
 IMAGE PROCESSING PROFILES FOR STATE-OF-THE-ART  
 AUTOMOTIVE APPLICATIONS.

Profile	Resolution (px)	Chroma Sub-sampling	Refresh Rate (fps)
2MP-RAW	1920 x 1080	4:4:4	60, 30
2MP-Compressed	1920 x 1080	4:2:0	60, 30
1MP-RAW	1280 x 720	4:4:4	60, 30, 15
1MP-Compressed	1280 x 720	4:2:0	60, 30, 15
Profil1	1280 x 275	4:4:4	60
Profil1-Compressed	1280 x 275	4:2:0	60

capacity for future work. The on board power consumption of 6.474 W is reduced by 21.1 % compared to the benchmark but only 11.1 % higher than the hardware profile with the lowest power consumption. The power consumption of the cloud system shows the same behavior as the ECU system, averaging 6.134 W.

#### V. INTERFACE TO INTERFACE LATENCY

This section examines the latency introduced by elements of the GStreamer pipeline [12]. Interface to interface latency is defined as the latency from source to sink. The interface to interface latency is determined with the GStreamer debugging tool. The results for end-to-end latency, from camera to file-sink, is shown in Figure 6. Its values are based on the previously presented tables in this subsection. It can be observed, that the pipeline latency of the 1 MP video streams is less than for the 2 MP video streams. The H.265 and H.264 coding profile for the 1 MP video stream result in nearly the same latency, averaging 24.64 ms, whereas the end-to-end latencies of the 2 MP video streams scatter slightly. This is due the fact that the pipeline latency of the emulated FPGA cloud roughly doubles from 10.36 ms to 21.35 ms.

#### VI. SYSTEM UTILIZATION

The system utilization of the hardware design has been analyzed with the Vivado utilization tool. Table III shows the system utilization of both designs.



Table III  
SYSTEM UTILIZATION OF THE ECU AND CLOUD SYSTEM DESIGN  
FROM THE VIVADO UTILIZATION REPORT OF THE ROUTED  
DESIGN.

Resource	Available	Utilization ECU	Utilization Cloud
LUT	230400	55244	58883
LUTRAM	101760	5588	4774
FF	460800	81214	81864
BRAM	312	70.5	69
URAM	96	44	44
DSP	1728	57	97
IO	360	21	22
GT	20	3	3
BUFG	544	26	26
MMCM	8	3	3

The average resource utilization for the ECU system design is 18.5%. The FPGA-based cloud design averages a resource utilization of 18.2%. Further expansions of the hardware designs are therefore possible.

## VII. CONCLUSION

A hybrid cloud-based processing platform is exemplarily implemented to validate the possibilities of real-time cloud-based processing architectures. The introduced platform is realized with two Xilinx Zynq UltraScale+ MPSoC ZCU106 boards. The goal is the recording of image data in a local system and sending in real-time to a cloud or edge system. Encoding the forwarded video stream is used to reduce the transmitted data and to minimize the overall latency by introducing an appropriate data compression. Both boards shall transmit data via UDP/IP over Ethernet in the prototype stage. For the final stage, wireless interfaces like 5G networks are going to be validated. For the coding, several video codecs like a H.264 or H.265 codec are implemented and validated. After the development of the prototype design platform, the system's behavior is investigated in different use cases.

This paper focuses on the timing for the sending, the energy costs of the local and the cloud or edge system and the required resources for the implementations of the different use cases. For example when having a 2 MP 60 fps video signal encoded with H.265 standard, the overall delay from the HDMI input in the local system to the cloud is 32.97 ms. The energy costs in the local ECU system are 6.47 W and in the cloud system 6.22 W for this use case. The logic utilization on both boards at this point is less than 20%.

## REFERENCES

- [1] Kachris, C. and Soudris, D. "A survey on reconfigurable accelerators for cloud computing", *26th International Conference on Field-Programmable Logic and Applications*. SwissTech Convention Centre, Lausanne, Switzerland, 29th August – 2nd September 2016, Piscataway, NJ, IEEE, pp. 1–10.
- [2] Iordache, A., Pierre, G., Sanders, P., J. G. D. F. Coutinho and Stillwell, M. "High Performance in the cloud with FPGA Groups", *2016 IEEE/ACM 9th International Conference on Utility and cloud Computing (UCC)*, pp. 1–10, 2016.
- [3] Terzis, A. (ed.). *Handbook of Camera Monitor Systems - The Automotive Mirror-Replacement Technology based on ISO 16505*, Springer International Publishing, 2016.
- [4] *Road vehicles — Ergonomic and performance aspects of Camera Monitor Systems — Requirements and test procedures*, ISO, 2015.
- [5] *Concerning the Adoption of Uniform Technical Prescriptions for Wheeled Vehicles, Equipment and Parts which can be Fitted and/or be Used on Wheeled Vehicles and the Conditions for Reciprocal Recognition of Approvals Granted on the Basis of these Prescriptions*, 2013.
- [6] Xilinx ZCU 106 Evaluation Board. *User Guide*, [https://www.xilinx.com/support/documentation/boards\\_and\\_kits/zcu106/ug1244-zcu106-eval-bd.pdf](https://www.xilinx.com/support/documentation/boards_and_kits/zcu106/ug1244-zcu106-eval-bd.pdf), 2018.
- [7] Xilinx Wiki, *Zynq UltraScale MPSoC VCU TRD 2018.3 - HDMI Video Capture*, <https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842136/XilinxDRMKMSHDMI-TxDriver>, accessed February 13th, 2019.
- [8] Xilinx Wiki, *Zynq Ultrascale MPSoC VCU TRD 2018.3 - HDMI Video Display*, <https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/55672840/Zynq+UltraScale+MPSoC+VCU+TRD+2018.3+-+HDMI+Video+Display>, accessed May 16th, 2019.
- [9] Skyworks, *5G in Perspective. A pragmatic Guide to what's next*, Skyworks Solutions Inc., 2017.
- [10] Yu, J., Guo, K., Hu, Y., Ning, X., Qiu, J., Mao, H., Yao, S., Tang, T., Li, B., Wang, Y. and Yang, H. "Real-time object detection towards high power efficiency", *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 704–708, 2018.
- [11] Xilinx. 2018. *H.264/H.265 Video Codec Unit v1.2*. [https://www.xilinx.com/support/documentation/ip\\_documentation/vcu/v1\\_2/pg252-vcu.pdf](https://www.xilinx.com/support/documentation/ip_documentation/vcu/v1_2/pg252-vcu.pdf), 2018, accessed May 15th, 2019.
- [12] GStreamer, *Open Source Multimedia framework*. [gstreamer.freedesktop.org](https://gstreamer.freedesktop.org/), 2019, accessed August 20th, 2019.



**Jannik Maier** studied electrical engineering and Information technology at the Ulm University of Applied Science. During his Bachelor study, he specialized on automotive electronics. Later he finished his Master's degree at the Technical University of Munich where he also studied in the field of electrical engineering with specialization on automation technologies. Prior to this he worked two years in an engineering office in the field of automated testing of

vehicles. Since 2018, Jannik Maier works as a researcher at the Ulm Technical University of applied science in the field of embedded systems. His research focus on cloud-based hybrid image processing architectures on SoC platform.



**Johannes Schäfer** studied electrical engineering at the Ulm University of Applied Science. He finished his Bachelor's degree in 2019. His Bachelor thesis focused on "Hybrid Image Processing Architecture on MPSoC-Basis for Cloud-based Camera-Monitor-Systems".



**Anestis Terzis** is professor for digital systems design and the head of the Institute of Communication Technology at Ulm University of Applied Sciences in Germany. Prior to this, he was with Daimler AG for ten years and worked in the Group Research & Mercedes-Benz Cars Development, with a main focus on future advanced driver assistance systems. Anestis Terzis received his diploma in Communications Engineering from Ulm University of Applied Sciences and his doctoral degree in Electrical Engineering from the University of Erlangen-Nuremberg. He is an expert member of the standardization and regulation committees (ISO, SAE, IEEE) in the field of camera monitor systems.



**Oliver Bringmann** received the Diploma (M.S.) degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, and the Doctoral (Ph.D.) degree in computer science from the University of Tübingen, Germany, in 2001. He was with the FZI Research Center for Information Technology, Karlsruhe, in various positions as a Department and Division Manager and a Member of the management board, until 2012. He has been a Professor and the

Director of the Chair for Embedded Systems with the University of Tübingen since 2012, where he is also serving as the Vice Head of the Department of Computer Science since 2014. He has authored and coauthored of more than 220 publications in the area of electronic design automation, embedded system design, and SoC architectures for automotive electronics and edge devices. His current research interests include electronic design automation, embedded system design, timing and power analysis of embedded software, embedded AI architectures, hardware-enhanced security, and robust perception. Oliver Bringmann is an IEEE member.

# Entwicklung eines energieautarken Türschildes mit E-Paper-Display und NFC-Konfigurationsschnittstelle

Andreas Angermayr, Elke Mackensen

**Zusammenfassung**—Elektronische Türschilder zur Darstellung von Informationen sind insbesondere in öffentlichen Gebäuden zwischenzeitlich weit verbreitet. Die Varianz dieser elektronischen Türschilder reicht vom Tablet-basierten Türschild bis hin zum PC-basierten Türschild mit externem Bildschirm. Zumeist werden die Systeme mit 230 V betrieben. Bei einer großen Summe von Türschildern in öffentlichen Gebäuden kann dies zu einem signifikanten Umsatz an Energie führen. Im Rahmen dieses Papers wird die Entwicklung eines energieautarken arbeitenden Türschildes vorgestellt, bei dem ein E-Paper-Display zum Einsatz kommt. Das Türschild lässt sich per Smartphone-App und NFC-Schnittstelle konfigurieren. Es wird insbesondere auf das Low-Power-Hardware-Design der Elektronik und energetische Aspekte eingegangen.

**Schlüsselwörter**—E-Paper, E-Ink, Low-Power, Digitale Türschilder, NFC.

## I. EINLEITUNG

In öffentlichen Gebäuden wie an Hochschulen, Ämtern oder beispielsweise Museen, aber auch in Firmen- und Bürogebäuden, setzen sich zunehmend elektronische Türschilder zur Informationsdarstellung durch. Weiterhin setzen sich auch in Einkaufsläden elektronische Displays zur Anzeige von Preisinformationen durch. Die Vorteile elektronischer Türschilder bzw. Displays liegen auf der Hand:

- Man kann elektronische Türschilder dynamisch und flexibel beschriften, d.h. gegenüber papierbasierten Türschildern können Daten der Türschilder schnell ausgetauscht werden.
- Informationen auf den Türschildern können stets auf dem neuesten Stand gehalten werden.
- Es ist möglich, elektronische Türschilder mit anderen elektronischen Systemen, wie z.B. Kalendersysteme, zu verknüpfen, um aktuelle Informationen automatisiert darzustellen.
- Der Zeitaufwand für die Aktualisierung eines elektronischen Türschildes ist wesentlich geringer, als die Wartung von Papiertürschildern. Insbesondere in großen Gebäuden kann der Zeitaufwand für die Aktualisierung von Papiertürschildern sehr hoch sein.

Andreas Angermayr, andreas.angermayr@hs-offenburg.de, Elke Mackensen, elke.mackensen@hs-offenburg.de, Hochschule für Technik, Wirtschaft und Medien Offenburg, Badstraße 24, 77652 Offenburg

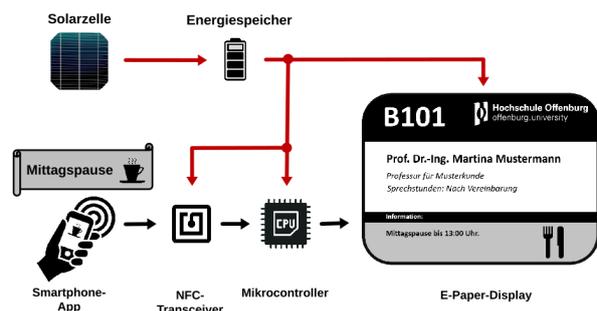


Abbildung 1. Grundsätzliche Idee für die Entwicklung eines energieautarken Türschildes.

Die Varianz elektronischer Türschilder reicht vom Tablet-basierten Türschild bis hin zum PC-basierten Türschild mit externem Bildschirm. Eine Konfiguration der Türschilder ist oftmals über WLAN oder Ethernet möglich. Zumeist werden derartige Systeme mit 230 V oder Batterien betrieben. Bei einer großen Summe von Türschildern in öffentlichen Gebäuden kann dies zu einem signifikanten Umsatz an Energie führen. Erfahrungen aus dem Betrieb elektronischer Türschilder an der Hochschule Offenburg zeigen einen weiteren Nachteil PC- oder Tablet-basierter elektronischer Türschilder auf: Derartige Türschilder werden oftmals mit Windows-Betriebssystem betrieben, sodass leider häufig Ausfälle zu verzeichnen sind.

Im Rahmen dieses Papers wird auf Grund der aufgezeigten Nachteile auf die Entwicklung eines energieautark arbeitenden digitalen Türschildes und die Implementierung eines Funktionsmusters eingegangen. Die erste Idee für Entwicklung des energieautarken Türschildes ist in Abbildung 1 dargestellt, was zu den folgenden Anforderungen an die Entwicklung führt:

- Die notwendige Energie für den laufenden Betrieb des Türschildes soll per Energy-Harvesting, in diesem Fall Solarzellen, aus der Umgebung gewonnen werden. Trotz Energy-Harvesting sollen möglichst viele Informationsupdates pro Tag (im Folgenden *Updates*) auf dem Display möglich sein, was zu der Forderung nach einem möglichst energiesparendem Elektronikdesign führt.
- Als vielversprechende Display-Technologie soll ein E-Paper-Display eingesetzt werden [1] - [5].

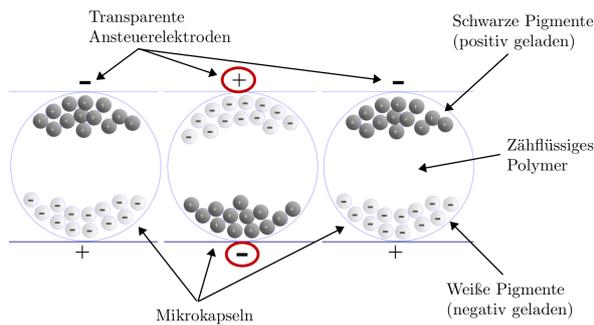


Abbildung 2. Querschnitt durch ein EPD-Panel (angelehnt an [6]).

Diese benötigen im Gegensatz zu LCD- oder OLED-Displays im statischen Betrieb keinerlei Energie.

- Die Entwicklung des Gesamtsystems soll auf einem Mikrocontroller ohne Betriebssystem basieren.
- Die Konfiguration des Türschildes (Angezeigter Text, Grafiken usw.) soll per Smartphone-App möglich sein. Als physische Schicht der Benutzerschnittstelle soll NFC (Near Field Communication) verwendet werden.

Im Folgenden wird zunächst in Abschnitt II auf den Stand der Technik der E-Paper-Displaytechnologie und kommerziell erhältlicher elektronischer Türschilder eingegangen. In Abschnitt III wird das detaillierte Hardware-Systemkonzept des energieautarken Türschildes vorgestellt und die Auswahl der Komponenten erläutert. Abschnitt IV stellt die Implementierung eines Funktionsmusters vor. In Abschnitt V werden detaillierte Messergebnisse des Gesamtsystems diskutiert und Optimierungspotenziale aufgezeigt. Abschließend wird in Abschnitt VI aufgrund der bisherigen Ergebnisse ein Fazit für die weitere Entwicklung des energieautarken Türschildes gezogen.

## II. E-PAPER-DISPLAYTECHNOLOGIE UND MARKTÜBERSICHT VERGLEICHBARER TÜRSCHILDER

Wie in der Einleitung erläutert, sind E-Paper-Displays ein vielversprechender Ansatz für die Realisierung eines energieautarken digitalen Türschildes, weshalb im Folgenden zunächst die E-Paper-Technologie etwas näher betrachtet wird und dann eine kurze Marktübersicht zu kommerziell erhältlichen E-Paper-Türschildern gegeben wird.

### A. E-Paper-Displaytechnologie

E-Paper-Displays (EPDs) sind Displays, deren Betrachtungseindruck jenem von bedrucktem Papier nachempfunden ist und die im statischen Betrieb keine Leistung benötigen. Die Anwendungsbereiche sind vielseitig und reichen mittlerweile von E-Book-Readern bis zu Benzinpreisanzeigen an Tankstellen.

EPDs basieren auf dem Effekt der Elektrophorese [5, 6]. Abbildung 2 zeigt den Querschnitt durch ein EPD-Panel. Dieses besteht aus transparenten Mikro kapseln, die ein zähflüssiges Polymer, sowie schwarze und weiße Pigmente enthalten, die entgegengesetzt geladen sind. Auf und unter dem Panel befindet sich eine Matrix aus transparenten Ansteuerelektroden, deren Anordnung letztlich die Bildpunkte definiert. Die Diffusion der Pigmente ist durch das Polymer stark eingeschränkt, sodass ein EPD ein Bild über Wochen und Monate hält.

Durch Anlegen eines E-Feldes über den entsprechenden Bildpunkten lässt sich das Bild verändern, indem die Pigmente je nach Richtung des Feldes nach oben oder nach unten wandern. Dieser Vorgang wird als Refresh bezeichnet.

Das E-Feld muss dabei im Zeitverlauf besonderen Anforderungen genügen, die den aktuellen Zustand des Pixels, den Zustand benachbarter Pixel, die Temperatur und weitere Parameter berücksichtigen, sodass aufwändige Wellenformen zur Ansteuerung der Pixel berechnet werden müssen, wofür ein EPD-(Timing)-Controller benötigt wird [6]. Moderne EPD-Controller sind so in der Lage, nur einen bestimmten Anteil der Pigmente zu bewegen, wodurch die Darstellung von Graustufen ermöglicht wird.

Das komplette Weiß- bzw. Schwarz-Überschreiben ist hingegen unkritisch, hierzu muss lediglich ein genügend starkes Feld ausreichend lange angelegt werden. Die Leistungsaufnahme des Panels beim Refresh ist dadurch begründet, dass die Partikel gegen den Strömungswiderstand des Polymers durch die Kapseln bewegt werden müssen, d.h. mit steigender Panelgröße steigt auch die Leistungsaufnahme beim Refresh.

### B. Marktübersicht vergleichbarer Türschilder

In Tabelle I sind einige derzeit erhältliche E-Paper-Türschilder einander vergleichend gegenübergestellt. Es wird klar, dass zwar kommerzielle Lösungen mit E-Paper-Display existieren, jedoch keine die energieautark arbeiten, d.h. mittels Energy-Harvesting betrieben werden. Stattdessen werden die Türschilder aus einer Batterie versorgt, was zu Ausfällen führen kann und den Wartungsaufwand erhöht.

Weiterhin gibt es derzeit noch keine Produkte, die über eine NFC-Schnittstelle verfügen. Die Konfiguration über WLAN erfordert ein Anmelden im WLAN-Netzwerk, die Konfiguration über ein USB-Kabel ist denkbar unkomfortabel. Eine NFC-Schnittstelle in Verbindung mit einer Smartphone-App verspricht hingegen eine schnelle und unkomplizierte Konfigurationsmöglichkeit.

## III. HARDWARE-SYSTEMKONZEPT UND AUSWAHL DER KOMPONENTEN

Im Folgenden wird das detaillierte Hardware-Systemkonzept des energieautarken Türschildes vorgestellt und die Auswahl der Komponenten erläutert.

Tabelle I  
 ÜBERSICHT EINIGER KOMMERZIELL ERHÄLTLICHER E-PAPER-TÜRSCHILDER.

Hersteller/Produkt	Energiequelle	Energieautark	Benutzer-Schnittstelle	Displaytechnologie	Preis in €
go2 / Türschild e-Ink [7]	Batterie	nein	USB	EPD	295
m.i.b. / door_ink [8]	Batterie	nein	"Funk"(W-Lan?)	EPD	?
e-shelf-labels / ROOMZ [9]	Batterie	nein	W-Lan	EPD	?

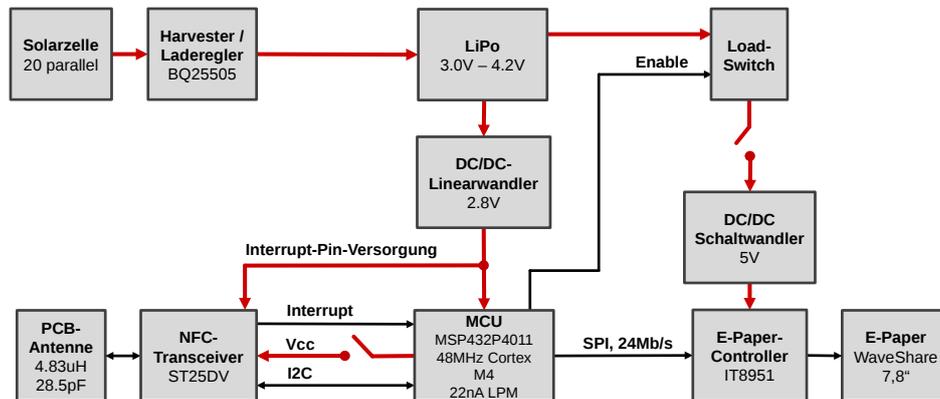


Abbildung 3. Hardware-Systemkonzept des energieautarken Türschildes.

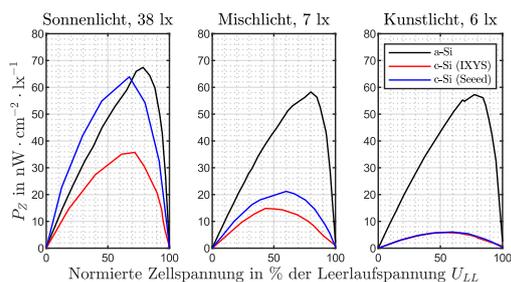


Abbildung 4. Vergleichsmessung von amorphen und kristallinen Siliziumzellen unter schwacher Sonnen-, Misch- und Kunstlicht-Beleuchtung.

Hierzu sind in Abbildung 3 die Systemkomponenten und deren Verschaltung dargestellt.

### A. Solarzellen

Für das Energy-Harvesting kommen amorphe Silizium-Solarzellen vom Typ Amorton AM-1454 von Panasonic [10] zum Einsatz. Die Verwendung amorpher Zellen wurde durch eine durchgeführte Messreihe begründet, in der die  $P(U)$ -Kurve verschiedener Zellen unter sehr schwacher Kunstlicht-, Mischlicht- und Sonnen-Beleuchtung zwischen 6 lx und 38 lx aufgenommen wurde. Das Ergebnis ist in Abbildung 4 dargestellt und zeigt, dass die amorphe Panasonic-Zelle (a-Si) den kristallinen Zellen (c-Si) unter den zu erwartenden Beleuchtungsbedingungen (dunkle Gänge und Korridore, wenig Sonnenlicht) klar überlegen ist.

### B. Spannungsversorgung

Die von den Solarzellen bereitgestellte Energie wird von einem BQ25505 Energy-Harvesting-IC von Te-

xas Instruments [11] in einen LiPo-Akku zwischengespeichert. Der IC benötigt zum Starten lediglich  $15 \mu\text{W}$  Leistung an einer Spannung von mindestens 600 mV und hat einen Ruhestrom von nur 400 nA. Da die maximale Eingangsspannung bei 5,1 V liegt, sind Solarzellen mit ausreichend kleiner Leerlaufspannung notwendig.

Ein LiPo-Akku ist anderen Energiespeichern wie beispielsweise Supercaps vorzuziehen, da die Selbstentladung geringer ist, weiter sind Akkus kleiner Kapazität vorzuziehen, da der Selbstentladestrom bei gleicher Selbstentladerate kleiner ist.

Zur Versorgung des Mikrocontrollers ist ein 2,8 V-Linearwandler verbaut, der E-Paper-Controller wird durch einen abschaltbaren Buck-Boost-Schaltwandler versorgt, der NFC-Transceiver wird direkt aus einem Mikrocontroller-Pin versorgt.

### C. NFC-Transceiver

Als NFC-Transceiver-IC kommt ein ST25DV von ST Microelectronics [12] zum Einsatz. Dieser verfügt über eine I<sup>2</sup>C-Schnittstelle und eine Mailbox zum einfachen Nachrichtenumsatz von der NFC- auf die I<sup>2</sup>C-Schnittstelle und umgekehrt. Weiterhin kann bei Events in der Mailbox oder der NFC-Schnittstelle ein Interrupt ausgegeben werden. Da sich der NFC-Transceiver-Teil des ICs standardkonform aus dem NFC-Feld versorgt, wird der Interrupt auch bei Trennung des ICs von der Spannungsversorgung ausgegeben. Weiterhin sorgen umfangreiche Dokumentation, Software-Development-Kits (SDKs) für Android und C-Anwendungen und Tools für das Antennendesign für kurze Entwicklungszeiten.



Abbildung 5. Display vor und nach Update ohne Weiß-Überschreibung nach Neustart des Controllers.

#### D. Display

Aktuell sind große EPDs in kleiner Stückzahl schwer beschaffbar. Ein Anbieter ist WaveShare, der zu den Displays passende EPD-Controller sowie Beispielcode mitliefert, was sich positiv auf die Entwicklungszeit auswirkt, sodass hier das Display 7,8inch e-Paper HAT [13] verwendet wird. Dieses hat eine Größe von etwa 174 mm × 128 mm bei einer sehr guten Auflösung von 1872 px × 1404 px und bis zu 16 Graustufen.

Nachteilig ist hierbei, dass der angebotene Controller eine Standby-Leistungsaufnahme von 300 mA hat, sodass ein Abschalten über einen Load-Switch zwingend erforderlich ist. Da der Controller zur Berechnung der Ansteuer-Wellenformen das aktuelle Bild im flüchtigen Speicher halten muss, der beim Abschalten korrumpiert wird, ist nach jedem Einschalten des Controllers ein Weiß- oder Schwarz-Überschreiben des kompletten Displays nötig, andernfalls kommt es beim Refresh zu fehlerhaften Bildern, wie in Abbildung 5 gezeigt.

#### E. Mikrocontroller

Der verwendete Mikrocontroller sollte über einen möglichst sparsamen Tiefschlafmodus verfügen, weiterhin muss ausreichend Programmspeicher vorhanden sein, da die auf dem Display darzustellenden Schriften und Bilder als Bitmap-Arrays gespeichert werden müssen. Es wird daher ein MSP432P4011-Mikrocontroller von Texas Instruments [14] verwendet. Dieser verfügt über einen 22 nA Tiefschlafmodus, 2 MB Flash Speicher und eine maximale Taktrate von 48 MHz.

#### F. Energiemanagement und Benutzerinteraktion

Im Ruhezustand sind die Spannungsversorgungen für den EPD-Controller und den NFC-Transceiver abgeschaltet, lediglich dessen Interrupt-Pin und der Mikrocontroller werden versorgt.

Möchte der Benutzer ein Update vornehmen, wählt er z.B. eine passende Abwesenheitsnachricht in der App aus und hält das Smartphone zur Übertragung an das Türschild. Der NFC-Transceiver löst einen Interrupt aus, der den Mikrocontroller aufweckt. Dieser schaltet anschließend die Versorgung des NFC-Transceivers ein, um die eintreffenden Nachrichten



Abbildung 6. Implementiertes Funktionsmuster des energieautarken Türschildes.

über die  $I^2C$ -Schnittstelle aus dessen Mailbox zu lesen.

Anschließend wird die Versorgung des Transceivers wieder abgeschaltet und die Versorgung des EPD-Controllers eingeschaltet, um das Display in einem ersten Refresh zunächst weiß zu überschreiben. Hierzu muss ein vollständiges Weiß-Bild über die SPI-Schnittstelle übertragen werden. Anschließend wird das neue Bild zum EPD-Controller übertragen und nach einem erneuten Refresh dargestellt. Danach wird die Versorgung des EPD-Controllers abgeschaltet und der Mikrocontroller legt sich wieder schlafen.

## IV. IMPLEMENTIERUNG DES GESAMTSYSTEMS

### A. Hardware-Implementierung

Abbildung 6 zeigt die Implementierung des im vorherigen Abschnitt erläuterten Hardware-Systemkonzeptes als Funktionsmuster. Dieses soll im Folgenden vorgestellt werden.

Oben in Abbildung 6 sind die Solarzellen zu sehen. Um die Funktion des Türschildes auch bei sehr schlechter Beleuchtung von wenigen Lux sicherstellen zu können, wurden 20 Zellen mit einer Gesamtfläche von  $218,5 \text{ cm}^2$  verbaut. Die Zellen wurden ohne Blocking-Dioden parallel verschaltet, da davon auszugehen ist, dass alle Zellen ähnlich bestrahlt werden. Eine Reihenschaltung ist aufgrund des Eingangsspannungsbereiches des Harvester-ICs nicht möglich.

Auf der mittig im Bild zu sehenden Leiterkarte befindet sich oben links der Energy-Harvesting-Teil mit dem  $70 \text{ mA-LiPo}$  Akku, der genug Energie für etwa 200 Updates vorhalten kann. Bei 1% Selbstentladerate beträgt der Selbstentladestrom etwa  $1,91 \mu\text{A}$ , d.h. auch unter schlechtester Beleuchtung muss dieser Strom mindestens durch den Harvester-IC nachgeliefert werden.

Im rechten oberen Teil der Leiterplatte sind das mittels Buchsenleiste aufgesteckte EPD-Controller-Board, sowie der 5V-Buck-Boost-Wandler zur Versorgung des EPD-Controllers zu sehen. Der Buck-Boost-Wandler benötigt eingangsseitig mindestens  $100 \mu\text{F}$  Kapazität, andernfalls kann es aufgrund der relativ hohen Ausgangsimpedanz des Harvester-ICs von etwa  $1,5 \Omega$  während eines Refreshs zu Spannungseinbrüchen kommen, was zu einem Reset des EPD-Controllers führt.

Mittig und unten auf der Leiterplatte befindet sich der NFC-Transceiver-Teil in dreifacher Ausführung. Zur Auslegung der Antennen wurde das *eDesign-Antenna-Tool* von ST [15] verwendet. Ziel war es, dass die Induktivität mit der internen Abstimm-Kapazität des NFC-Transceiver-ICs einen Schwingkreis mit Resonanzfrequenz  $f_r = 13,56 \text{ MHz}$  bildet, sodass kein Abgleich nötig und die Induktivität möglichst groß ist. Dabei wurden drei Antennen entworfen, deren Induktivität um  $-10\%$ ,  $0\%$  und  $+10\%$  um den errechneten optimalen Wert schwankt. Über Jumper ist auswählbar, welcher NFC-Transceiver mit dem Mikrocontroller verbunden wird.

Hierbei zeigte sich, dass alle drei Antennen in Bezug auf die Verbindungsqualität augenscheinlich gleich performant sind. Weiterhin passen die realisierten Induktivitätswerte gut mit den vom *eDesign-Antenna-Tool* errechneten Werten zusammen.

Im unteren Bildbereich ist schließlich das  $7,8''$ -EPD mit einer beispielhaften Nachricht im Informations-Bereich ganz unten auf dem Display zu sehen.

### B. Software-Implementierung

Die Firmware des Systems arbeitet interruptgetrieben ohne Betriebssystem. Der MSP432P4011-Mikrocontroller lässt sich aus dem energiesparendsten Tiefschlafmodus nicht direkt aufwecken, sondern muss nach einem Interrupt einen Reset durchführen und sich danach neu initialisieren.

Zur Abwicklung der NFC-Kommunikation wurde ein einfaches Nachrichtenprotokoll entwickelt. Dieses

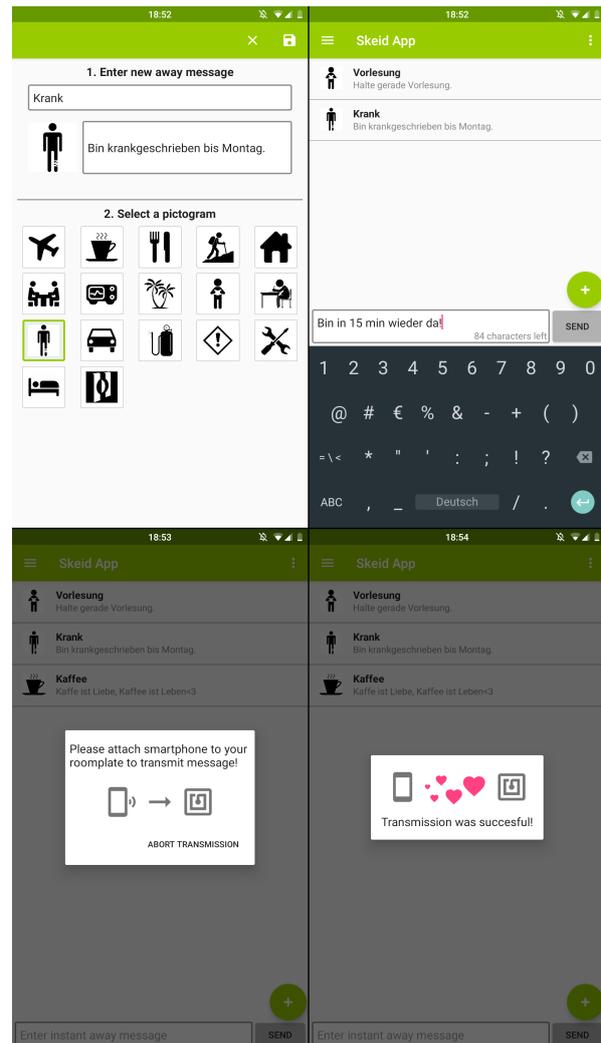


Abbildung 7. Fenster der App zum Anlegen (oben links) und Auswählen (oben rechts) von Abwesenheitsnachrichten, sowie Sendedialekt (unten).

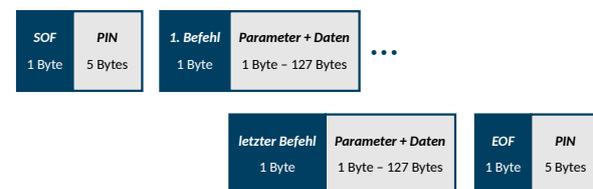


Abbildung 8. Rahmenformate und zeitlicher Ablauf des Kommunikationsprotokolls.

definiert die in Abbildung 8 gezeigten Frames mit einem Befehls- bzw. Steuer-Byte und bis zu 127 Byte Payload. Zusätzlich gibt es SOF (*Start-Of-Frame*) und EOF-Rahmen (*End-Of-Frame*), die als Payload eine Pin zur Nutzerauthentifizierung enthalten und zur Flusskontrolle dienen.

Abbildung 7 zeigt die implementierte Android-App, in der sich Nachrichten anlegen, speichern und versenden lassen, sowie die zugehörigen Benutzer-Dialoge zum Verbindungsauf- bzw. -abbau mit dem Türschild.



Abbildung 9. Zeitliches Systemverhalten während eines Türschild-Updates.

Firmware und App verwenden jeweils die SDKs, die von ST Microelectronics für den ST25DV-NFC-Transceiver-IC bereitgestellt werden, sodass sich die Entwicklung um die NFC-Schnittstelle als überraschend einfach erwies.

## V. MESSERGEBNISSE

Zur Analyse und Bewertung des Systemkonzeptes bzw. der Implementierung wurden verschiedene Messungen am Funktionsmuster durchgeführt, die hier erläutert werden.

### A. Zeitliches Verhalten während eines Displayupdates

Zunächst wurde das zeitliche Systemverhalten analysiert. Hierzu wurden an zwei freien Pins Signale ausgegeben und oszillographiert, welche anzeigen, ob der Mikrocontroller wach ist (grünes AWAKE-Signal) oder sich in der Initialisierungsphase befindet (blaues INIT-Signal). Zusätzlich wurden der NFC-Interrupt-Pin (gelbes INT-Signal) und der Enable-Pin des Display-Load-Switches (rotes EN\_EPD-Signal) aufgezeichnet. Abbildung 9 zeigt die Ergebnisse.

Das Update dauert relativ lange, obwohl die Benutzerinteraktion nach weniger als 2s abgeschlossen ist, wie am gelben INT-Signal zu sehen ist. Die Initialisierungszeit des Mikrocontrollers ist zu vernachlässigen, aber wie am roten EN\_EPD-Signal zu sehen ist, dauern die Übertragung des Weiß-Bildes und die anschließende Übertragung des darzustellenden Bildes mit etwa 10s sehr lange, was an der unzureichenden Auslastung des 24 Mbps-SPI-Busses durch den mit nur 48 MHz getakteten Mikrocontroller liegt. Hier wäre ein schnellerer Controller wünschenswert.

### B. Verfügbare Leistung durch das Energy-Harvesting

Mit dem in Abbildung 10 gezeigten Aufbau wurde die vom Harvester bereitgestellte Leistung in Abhängigkeit von der Beleuchtungsstärke gemessen. Hierzu wurde das Funktionsmuster an verschiedenen Orten aufgestellt, und die Beleuchtung der Solarzellen, sowie die Eingangs- und Ausgangsleistung des Harvester-ICs

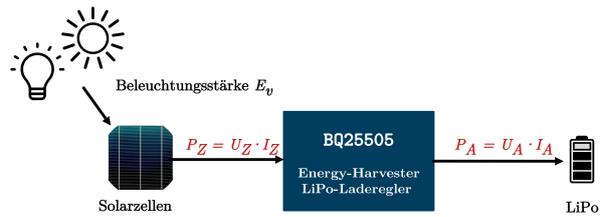


Abbildung 10. Schematischer Messaufbau zur Bestimmung der verfügbaren Leistung.

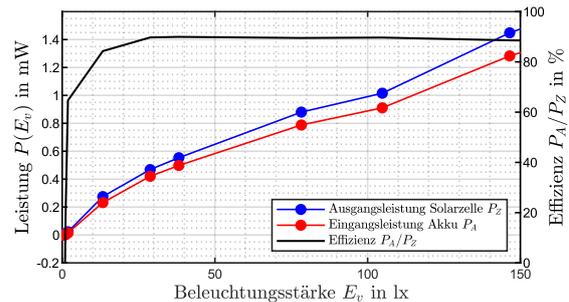


Abbildung 11. Verfügbare Leistung in Abhängigkeit von der Beleuchtungsstärke.

gemessen und daraus die Wandlungseffizienz berechnet. Abbildung 11 zeigt die Ergebnisse. Es zeigt sich, dass auch bei sehr schlechter Beleuchtung unter 10 lx nennenswert Leistung bereitgestellt wird, und dass die Effizienz bereits relativ hoch ist.

### C. Leistungsaufnahme und Energieumsatz des Türschildes

Zur Bestimmung des Energieumsatzes des Systems wurde eine *Source-Measure-Unit* (SMU) verwendet, aus der das System bzw. bestimmte Komponenten versorgt wurden und welche die Momentanleistung periodisch aufzeichnete.

1) *Energieumsatz während Update:* Zunächst wurde der Energieumsatz des Funktionsmusters während eines Updates gemessen. Hierzu wurde die SMU anstelle des LiPo-Akkus geklemmt und die Solarzellen abgeklemmt. Abbildung 12 zeigt die Ergebnisse. Bei  $t = 0,5s$  wacht der Mikrocontroller auf, die NFC-Kommunikation wird abgewickelt. Es folgt bei  $t = 1s$  das Einschalten des EPD-Controllers und die Übertragung des weißen Bildes in den EPD-Controller. Bei  $t = 5s$  wird das Display auf weiß refreshet, anschließend wird das darzustellende Bild in den EPD-Controller geschrieben, der abschließende Refresh folgt bei  $t = 11s$ . Die Refreshes sind an den hohen Leistungsspitzen zu erkennen.

Während der SPI-Kommunikation benötigt der EPD-Controller fast durchgehend über 300 mW Leistung, entsprechend steigt der Energieumsatz über der Zeit fast linear bis 4,5Ws an. Das System ist wach bis  $t = 15s$  und es zeigt sich, dass die Leistungsaufnahme

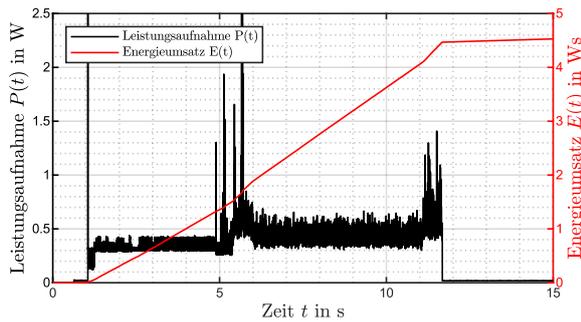


Abbildung 12. Energieumsatz des Türschildes während Displayupdate.

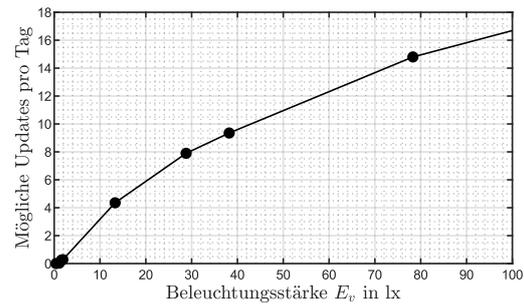


Abbildung 13. Anzahl möglicher Updates pro Tag in Abhängigkeit von der Beleuchtungsstärke.

me des Displaycontrollers um Größenordnungen über jener der übrigen Systemkomponenten liegt.

Der verwendete EPD-Controller bietet damit erhebliches Einsparpotenzial, sodass es wünschenswert wäre, einen für Low-Power-Designs passenderen Controller-IC, wie beispielsweise den S1D13521B01 von Epson [16] zu verwenden. Da diese ICs aber häufig extern mit einem EPD-Power-Management-IC und einem Flash-Speicher beschaltet werden müssen, die beim verwendeten Controller-Board bereits vorhanden sind, könnte sich dies nachteilig auf die Entwicklungszeiten auswirken.

2) *Leistungsaufnahme im Schlafzustand:* Mit dem vorherigen Messaufbau wurde die Leistungsaufnahme des Türschildes im Schlafzustand bestimmt. Damit ergab sich eine Leistungsaufnahme von etwa  $1,5 \mu\text{W}$ , d.h. bei  $2,8 \text{ V}$  eine Stromaufnahme von lediglich etwa  $535 \text{ nA}$ .

#### D. Anzahl der möglichen Updates pro Tag

Aus dem Energieumsatz pro Update und der verfügbaren Leistung in Abhängigkeit von der Beleuchtungsstärke wurde die Anzahl der pro Tag möglichen Updates berechnet. Das Ergebnis ist in Abbildung 13 zu sehen. Es zeigt sich, dass bei einer schlechten Beleuchtung von nur  $10 \text{ lx}$  im Mittel bereits drei Updates pro Tag ermöglicht werden.

Damit zeigt sich auch, dass die relativ große Solarzellenfläche durchaus notwendig ist, denn Messungen zeigten, dass beispielsweise die Beleuchtungsstärken vor den Räumen in der Hochschule Offenburg selten über  $20 \text{ lx}$  liegen.

#### E. Ressourcenverbrauch der Firmware

Wie oben angedeutet, sind Schriftarten und Bilder softwareseitig im Programmspeicher als Bitmap-Arrays hinterlegt. In Abbildung 14 ist die Belegung des Programmspeichers dargestellt. Die momentan 19 verfügbaren Piktogramme ( $320 \text{ px} \times 320 \text{ px}$  Auflösung, 4b Graustufen) belegen mit  $973 \text{ kB}$  knapp die Hälfte des Speichers, weitere  $577 \text{ kB}$  kommen durch 3 Schriftarten ( $133 \text{ px}$ ,  $64 \text{ px}$  und  $36 \text{ px}$  hoch) hinzu.

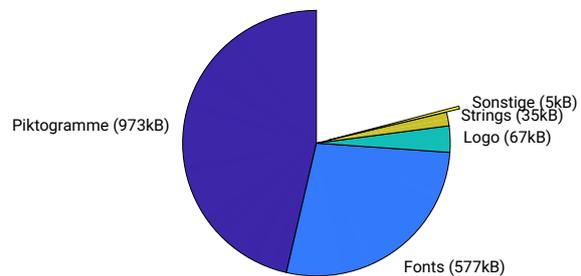


Abbildung 14. Aufteilung der Flashspeicherbelegung durch die Firmware.

Aufgrund der hohen Displayauflösung war es möglich, jede Schriftart ohne merkbare Unschärfe zur Anzeige auf dem Display um den Faktor 2 zu strecken, womit auf größere Schriftarten verzichtet werden konnte. Eine Komprimierung der Bitmaps z.B. durch eine Entropiekodierung würde den Speicherbedarf vermutlich wesentlich reduzieren, bei gleichem Speicher könnten mehr Bilder oder Schriftarten bereitgestellt werden.

## VI. ZUSAMMENFASSUNG UND FAZIT

In dieser Arbeit wurden die Entwicklung und Implementierung eines energieautarken elektronischen Türschildes mit E-Paper-Display und NFC-Konfigurationsschnittstelle vorgestellt, auf dem per Smartphone-App komfortabel Abwesenheitsnachrichten hinterlassen werden können.

Die NFC-Schnittstelle als wesentliche Neuerung zu bestehenden Lösungen erwies sich in der Handhabung mit dem Türschild als intuitiv und komfortabel.

Das System wird aus amorphen Silizium-Solarzellen versorgt und verfügt über einen LiPo-Akku als Energiespeicher. Durch sorgfältiges hard- und softwareseitiges Low-Power-Design beträgt die Leistungsaufnahme im Ruhemodus lediglich  $1,5 \mu\text{W}$ . Bedingt durch den anwenderfreundlichen, jedoch für Low-Power-Designs ungeeigneten EPD-Controller, verursacht ein Update

einen Energieumsatz von 4,5 Ws. Trotzdem zeigt sich, dass das System bei einer Zellfläche von knapp 220 cm<sup>2</sup> bei Beleuchtungsstärken von 10 lx bereits drei Updates pro Tag bereitstellen kann.

Es konnte damit gezeigt werden, dass die Verwendung eines E-Paper-Displays und konsequentes Low-Power-Design, insbesondere im Vergleich zu PC- bzw. LC-Display-basierten Lösungen, enormes Energiesparpotenzial bietet. Die in dieser Arbeit gewonnenen Erkenntnisse lassen sich ohne weiteres auf nicht-energieautarke Systeme erweitern, z.B. bei einer Versorgung aus Batterien oder Power-over-Ethernet (PoE), wo ein möglichst geringer Energieverbrauch ebenso wünschenswert ist.



**Andreas Angermayr** studierte Elektro- und Informationstechnik mit Schwerpunkt Kommunikationstechnik an der Hochschule Offenburg. Durch zahlreiche private und hochschulinterne Projekte konnte er sich umfangreiches Know-How im Bereich der Hardware-, Software- und System-Entwicklung, Software-Defined-Radio, sowie in der Elektronikfertigung aneignen. Besonders spannend findet er es, die Brücke zwischen Ultra-Low-Power und kurz-

zeitiger Performanz zu finden. Andreas Angermayr begann 2020 sein Elektrotechnik-Masterstudium und arbeitete seither auch als Akademischer Mitarbeiter im Schaltungstechniklabor. Leider konnte er seine Tätigkeiten an der Hochschule Offenburg nicht weiterführen, da er am 25.9.2021 bei einem tragischen Unfall in den Bergen ums Leben gekommen ist. Die Hochschule Offenburg hat mit ihm einen sehr engagierten, höflichen, verlässlichen, kompetenten, wissbegierigen und äußerst liebenswerten Mitarbeiter verloren.

#### LITERATURVERZEICHNIS

- [1] J. Heikenfeld: *The electronic Display of the Future*. In IEEE SPECTRUM, 26.02.2010. URL: <https://spectrum.ieee.org/computing/hardware/the-electronic-display-of-the-future> - Eingesehen: 22.04.2020.
- [2] A. Hellwig: *Warum sich der Einsatz eines E-Ink-Displays lohnen kann*. In ELEKTRONIK PRAXIS Nr. 8, 21.04.2016, S. 88–90.
- [3] G. Schryen, J. Karla: *Elektronisches Papier Displaytechnologie mit weitem Anwendungsspektrum*. In WIRTSCHAFTSINFORMATIK 44/6, 2002, S. 567–574.
- [4] M. R. Fernández, E. Z. Casanova, I. G. Alonso: *Review of Display Technologies Focusing on Power Consumption*. In SUSTAINABILITY, 7, 2015, S. 10854–10875, DOI: 10.3390/su70810854.
- [5] J. Heikenfeld, P. Drzaic, J.-S. Yeo, T. Koch: *Review Paper: A critical review of the present and future prospects for electronic paper*. In JOURNAL OF THE SID 19/2, 2011, S. 129–156, DOI: 10.1889/JSID19.2.129.
- [6] P. F. Bai, R.A. Hayes, M. Jin, L. Shui, Z. C. Yi, L. Wang, X. Zhang, G. Zhou: *Review of Paper-Like Display Technologies (Invited Review)*. In PROGRESS IN ELECTROMAGNETIC RESEARCH 147, 2014, S. 95–116. DOI: 10.2528/PIER13120405.
- [7] Bohmeyer & Schuster - Schilder-Verand.com: *go2 Türschild e-Ink*. URL: <https://www.schilder-versand.com/p/go2-tuerschild-e-ink-22011> - Eingesehen: 22.04.2020.
- [8] m.i.b. Terminal: *door\_ink* URL: <https://www.mib-terminal.de/produkte/tuerschildtechnik/funk-tuerschilder-door-ink> - Eingesehen: 22.04.2020.
- [9] e-shelf-labels: *ROOMZ* URL: <https://www.e-shelf-labels.com/products/digital-room-signage/roomz.html> - Eingesehen: 22.04.2020.
- [10] Panasonic Industry: *Amorphous Silicon Solar Cells, Amorphous Photosensors*, April 2020.
- [11] Texas Instruments: *BQ25505 - Ultra low-power boost charger with battery management and autonomous power multiplexer for primary battery in energy harvester applications*, Rev. E, März 2019.
- [12] ST Microelectronics: *Dynamic NFC/RFID tag IC with 4-Kbit, 16-Kbit or 64-Kbit EEPROM, and fast transfer mode capability*, Rev. 7, 2018.
- [13] Waveshare Wiki: *7.8inch e-Paper HAT*. URL: [https://www.waveshare.com/wiki/7.8inch\\_e-Paper\\_HAT](https://www.waveshare.com/wiki/7.8inch_e-Paper_HAT) - Eingesehen: 22.04.2020.
- [14] Texas Instruments: *MSP432P411x, MSP432P401x SimpleLink Mixed-Signal Microcontrollers*, Juni 2019.
- [15] ST Microelectronics: *eDesign Antenna*. URL: <https://eds.st.com/antenna/#/> - Eingesehen 22.04.2020.
- [16] Seiko Epson Corporation: *S1D13521B01 Epson / E Ink Broadsheet Hardware Functional Specification*, Rev. 1.2, 2008.



**Elke Mackensen** studierte Nachrichtentechnik an der Fachhochschule Offenburg. Nach dem Diplomabschluss in 1993 und einer dreijähriger Industrietätigkeit begann sie 1996 ihre wissenschaftliche Laufbahn an der Universität in Freiburg. Zunächst am Institut für Informatik wo sie eine ASIC-Design-Labor aufbaute, danach am Institut für Mikrosystemtechnik (IMTEK), wo sie 2006 über autarke drahtlose Mikrosysteme promovierte. Von 2006 bis 2010 leitete sie bei der Firma NewTec die Entwicklungsgruppe Smart Embedded Systems. Seit 2010 ist sie Professorin an der Hochschule in Offenburg. Ihre Professur beschäftigt sich mit der digitalen Schaltungstechnik und dem mikroelektronischen Systementwurf. Forschungsschwerpunkte sind drahtlose und auf Energy-Harvesting-basierte hochminiaturisierte Systeme.

# Problemfälle der Stabilität

Albrecht Zwick, Bernd Vettermann

**Zusammenfassung**—Bei vielen elektronischen Schaltungen können Stabilitätsprobleme auftreten. Zur Analyse kann man sehr gut das Millertheorem verwenden. Um die Stabilität zu beurteilen, muss man alle Bauelemente an eine gemeinsame Stelle der Operationsverstärkerschaltung transformieren. Hier werden die wichtigsten Problemfälle bei Schaltungen mit Operationsverstärkern dargestellt.

**Schlüsselwörter**—Stabilitätsprobleme, Millertheorem, Schaltungen mit Operationsverstärkern.

## I. GRUNDLAGEN

Abbildung 1 zeigt eine Parallelschaltung aus  $C$ ,  $R$  und  $L$ . Um gerade noch kein Überschwingen zu erhalten, muss die  $+20\text{dB/Dek}$ -Linie  $j\omega L$  immer größer sein als die gesamte Parallelschaltung (siehe Abbildung 2).

$$\left| j\omega L \right| > \left| \frac{1}{1/R + j(\omega C - 1/\omega L)} \right|$$

$$\left( \frac{1}{\omega L} \right)^2 < \left( \frac{1}{R} \right)^2 + (\omega C)^2 - 2 \frac{C}{L} + \left( \frac{1}{\omega L} \right)^2$$

Für  $\omega = 0$  gilt:

$$\left( \frac{1}{R} \right)^2 = 2 \frac{C}{L} \rightarrow R = \frac{1}{\sqrt{2}} \sqrt{\frac{L}{C}}$$

siehe Abbildung 2

Daraus erhält man den Wert  $\boxed{a = 2!}$  (1)

Beispiel: Abbildung 3 zeigt einen komplexen Spannungsteiler mit  $C$ ,  $R$  und  $L$ . Nur durch das Zusammenwirken der beiden Energiespeicher  $C$  und  $L$  kann es zum Schwingen kommen. Mit dem Widerstand  $R$  kann man die Schwingung beeinflussen.

Mit dem Parallelzeichen als neu eingeführtem Operator kann man die Gleichung mit  $j\omega$  vereinfachen. Für die Darstellung im Bodediagramm dieser Quotienten genügt die Errechnung eines Wertes bei einer einzigen Frequenz.

$$\frac{u_o}{u_i} = \frac{j\omega L}{j\omega L + R + \frac{1}{j\omega C}} = 1 \parallel j\omega \frac{L}{R} \parallel -\omega^2 LC$$

$$\text{für } f = 100 \text{ kHz:} \quad = 1 \parallel 0,7 \parallel 1$$

Albrecht Zwick, Bernd Vettermann, Hochschule Mannheim, Paul-Wittsack-Straße 10, D-68163 Mannheim.

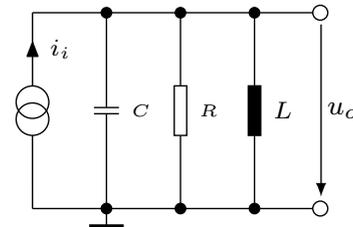


Abbildung 1. Parallelschaltung

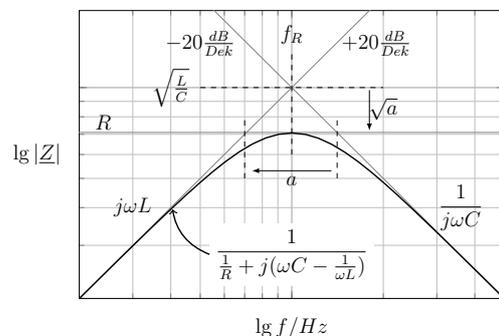


Abbildung 2. Bodediagramm der Parallelschaltung

In Abbildung 4 sind die drei verschiedenen Quotienten dargestellt. Die Parallelschaltung ergibt die Größe  $u_o/u_i$ . Die Linie des Quotienten  $\omega L/R$  begrenzt die Schwingung. Bei den in Abbildung 3 angenommenen Zahlenwerten erhält man mit  $R = 14\Omega$  den Wert  $a = f_o/f_b = 2$  und damit gerade noch kein Überschwingen auf der Frequenzachse. Man beachte: Das Ergebnis bei  $a = 2$  in Abbildung 4 ist die um  $-20\text{dB/Dek}$  gedrehte Kurve in Abbildung 2.

Bei kleiner werdendem  $a$  ergibt sich ein Überschwingen (Gain-Peaking).  $R = 0$  bedeutet  $a = 0$ . Man erhält die totale Resonanz. [1]

Abbildung 5 zeigt das Ergebnis über der Zeitachse bei einem Spannungssprung am Eingang.

Im Gegensatz zur passiven Schaltung ist sehr häufig bei einer elektronischen Schaltung mit aktiven und passiven Bauteilen nur ein Energiespeicher, z.B. eine Kapazität, sichtbar. Wenn es bei dieser Schaltung dann zu einem Schwingen kommt, muss eine Induktivität elektronisch erzeugt werden. Die Erzeugung des zweiten Energiespeichers kann man am besten durch das Millertheorem darstellen.

## II. MILLERTHEOREM

Ein Widerstand  $Z$  liegt zwischen einer Spannungsquelle  $u$  und einer davon abhängigen Spannungsquelle

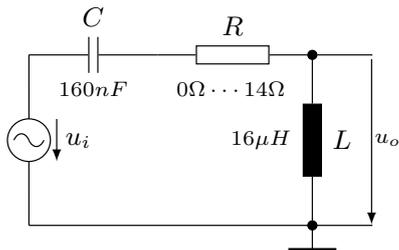


Abbildung 3. Spannungsteiler mit R, L und C

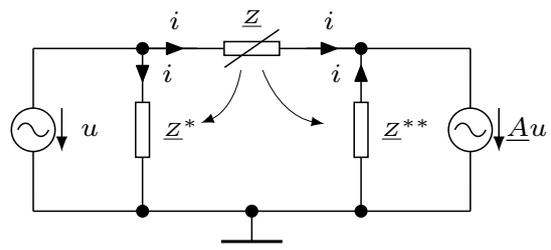


Abbildung 6. Grundschtung zum Millertheorem

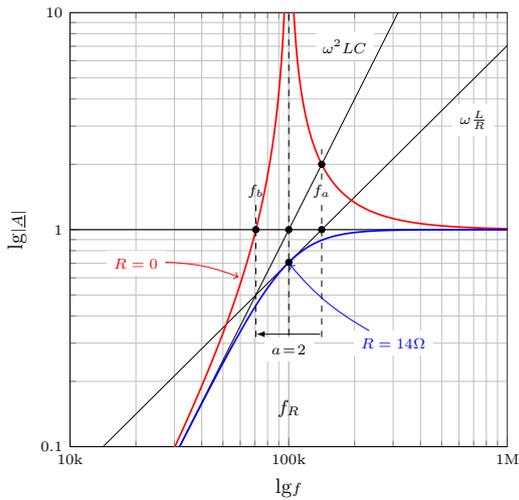


Abbildung 4. Bodediagramm mit  $R = 0$  und  $R = 14\Omega$

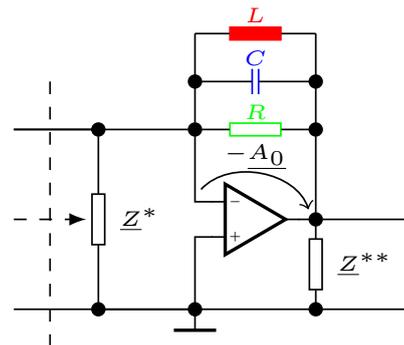


Abbildung 7. Millereffekt beim Operationsverstärker

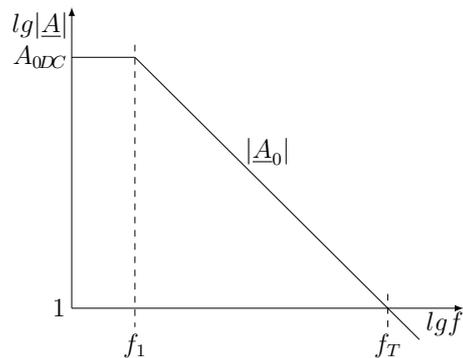


Abbildung 8. Bodediagramm des Operationsverstärkers

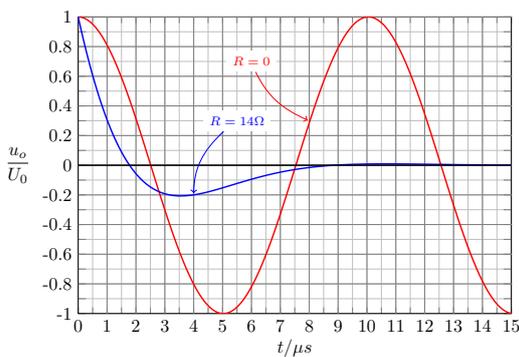


Abbildung 5. Eingangsspannungssprung mit  $R = 0$  und  $R = 14\Omega$

$\underline{A} \cdot u$ . Abbildung 6 zeigt die Grundschtung. Man ersetzt den Widerstand  $\underline{Z}$  durch die Widerstände  $\underline{Z}^*$  und  $\underline{Z}^{**}$ , durch die dann der gleiche Strom  $i$  fließt.  $\underline{Z}^*$  stellt auch den Eingangswiderstand dar.  $\underline{Z}^{**}$  ist jedoch nicht der Ausgangswiderstand der Schaltung. Dieser Widerstand  $\underline{Z}^{**}$  spielt meistens keine Rolle, da er am Ausgang des Operationsverstärkers parallel zur internen Spannungsquelle des Operationsverstärkers liegt.

Die Schaltung ist nur identisch, wenn man sie von  $u$  aus betrachtet, nicht jedoch von  $\underline{A} \cdot u$  (siehe Abbildung 6).

$$i = \frac{u - \underline{A} \cdot u}{\underline{Z}} = u \cdot \frac{1 - \underline{A}}{\underline{Z}} \rightarrow$$

$$i = \frac{u}{\underline{Z}^*} \quad \text{mit} \quad \underline{Z}^* = \frac{\underline{Z}}{1 - \underline{A}} \quad (2)$$

$$i = \frac{u - \underline{A} \cdot u}{\underline{Z}} = \frac{-\underline{A} \cdot u}{\underline{Z}^{**}} \rightarrow$$

$$\underline{Z}^{**} = \frac{\underline{Z}}{1 - \frac{1}{\underline{A}}}$$

### III. MILLEREFFEKT

Wenn  $\underline{A}$  groß ist und negativ ( $-\underline{A}_0$ ), dann nennt man diesen Effekt Millereffekt. Abbildung 7 zeigt als Beispiel die Transformationen beim Operationsverstärker.

Herleitung für die Transformationen des Widerstandes  $\underline{Z}$  in  $\underline{Z}^*$  beim Operationsverstärker. Siehe auch Abb. 8

$$\begin{aligned} \text{Für } f < f_T \text{ gilt: } Z^* &= \frac{Z}{1 - (-A_0)} = \frac{Z}{1 + A_0} \cdot Z^* \\ &= \frac{Z}{1 - A_0} = \frac{Z}{1 + \frac{A_{0DC}}{1 + j\frac{f}{f_1}}} \quad A_0 = \frac{-A_{0DC}}{1 + j\frac{f}{f_1}} \\ Z^* &= \frac{Z(1 + j\frac{f}{f_1})}{1 + j\frac{f}{f_1} + A_{0DC}} \quad f_1 \cdot A_{0DC} = f_T \cdot 1 = f_T \\ &= \frac{Z}{A_{0DC}} \cdot \frac{1 + j\frac{f}{f_1}}{1 + \frac{1}{A_{0DC}} + j\frac{f}{f_1 A_{0DC}}} \\ \boxed{Z^*} &= \frac{Z}{A_{0DC}} \cdot \frac{1 + j\frac{f}{f_1}}{1 + j\frac{f}{f_T}} \end{aligned} \quad (3)$$

Alle Bauteile können ebenso berechnet werden. Stromquellen werden 1 : 1 transformiert, da die Transformationen auf gleichem Strom basieren. Spannungsquellen werden wie Widerstände transformiert ( $u = i \cdot R$ ). Abbildung 9 zeigt die Transformation der Bauteile im Bodediagramm beim Operationsverstärker. Bei der Betrachtung des Millereffektes beginnt man am besten mit einer Frequenz  $f > f_T$  (siehe Abbildung 8). Bei dieser Frequenz ist der Operationsverstärker nicht mehr funktionsfähig. Der Ausgang kann als Null betrachtet werden. Der Widerstand  $\underline{Z}$  liegt somit auf Null und wird nicht transformiert. Betrachtet man jetzt die Frequenz immer kleiner werdend, so beginnt ab  $f_T$  der Operationsverstärker mit immer größer werdender Verstärkung zu transformieren. Die Frequenz  $f_T$  kann auch durch einen Spannungsteiler direkt am Ausgang des Operationsverstärkers erniedrigt werden. Ab der Frequenz  $f_1$  bleibt die Transformation dann wieder konstant (siehe Abbildung 9).

Man erkennt, dass aus einem ohmschen Widerstand  $R$  im Bereich  $f_1 < f < f_T$  eine Induktivität wird, aus einer Kapazität  $C$  ein Widerstand und aus einer Induktivität eine Superinduktivität. Eine Superinduktivität ergibt einen induktiven Widerstand  $\omega L$  deren Induktivität selbst noch frequenzabhängig ist.

$$j\omega L_S = \frac{j2\pi f L}{1 - A_0} \quad A_0 = j\frac{f_T}{f} \quad \frac{j2\pi f L}{1 - j\frac{f_T}{f}} = -\frac{2\pi f^2 L}{f_T}$$

$$\left[ -\frac{2\pi f^2}{f_T} L \right] \rightarrow -\frac{1}{f_T} \cancel{L} \cancel{1} R = -R$$

Eine Superinduktivität stellt somit einen negativen Widerstand dar, der quadratisch mit der Frequenz

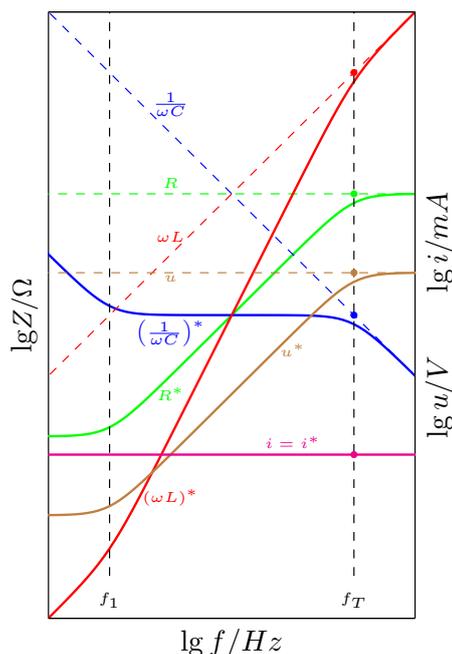


Abbildung 9. Millereffekt im Bodediagramm (Umgekehrter Bootstrapeffekt)

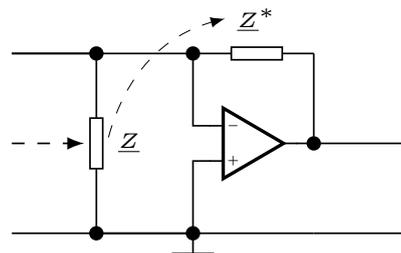


Abbildung 10. Umgekehrter Millereffekt beim Operationsverstärker

zunimmt. Spannungsquellen werden wie Widerstände transformiert, Stromquellen bleiben konstant. Von  $f_T$  zu kleineren Frequenzen her gesehen bedeutet der Millereffekt eine Linksdrehung um  $20dB/Dek$ .

#### IV. UMGEKEHRTER MILLEREFFEKT

Beim umgekehrten Millereffekt wird der Widerstand  $\underline{Z}$  am Eingang in die Rückführung des Operationsverstärkers in  $\underline{Z}^*$  transformiert (siehe Abbildung 10).

$$\begin{aligned} Z^* &= Z \cdot \left( 1 + \frac{A_{0DC}}{1 + j\frac{f}{f_1}} \right) \\ &= Z \cdot A_{0DC} \left( \frac{\frac{1}{A_{0DC}} + 1 + j\frac{f}{f_1}}{1 + j\frac{f}{f_1}} \right) \end{aligned}$$

$$\boxed{Z^* = Z \cdot A_{0DC} \cdot \frac{1 + j\frac{f}{f_T}}{1 + j\frac{f}{f_1}}} \quad (4)$$

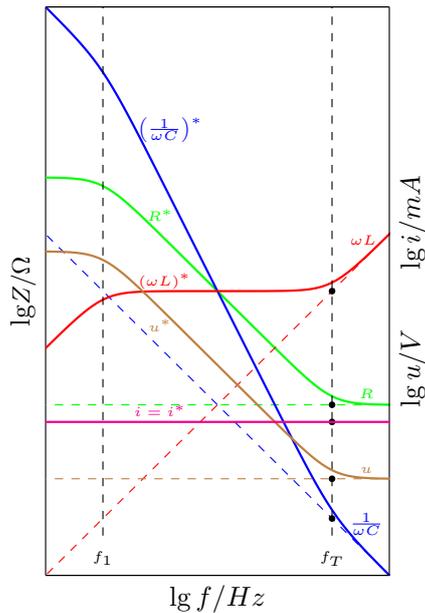


Abbildung 11. Umgekehrter Millereffekt (Bootstrapeffekt)

Abbildung 11 zeigt das Bodediagramm des umgekehrten Millereffektes. Der mathematische Zusammenhang wird durch die Gleichung 4 dargestellt. Von  $f_T$  zu kleineren Frequenzen hin gesehen bedeutet der umgekehrte Millereffekt eine Rechtsdrehung um  $20dB/Dek$ . Aus einem Widerstand wird jetzt im Frequenzbereich  $f_1 < f < f_T$  eine Kapazität, aus einer Induktivität ein Widerstand und aus einer Kapazität eine Superkapazität. Ebenso wie eine Superinduktivität stellt eine Superkapazität auch einen negativen Widerstand dar, der quadratisch mit der Frequenz abfällt. Spannungsquellen werden wie Widerstände transformiert, Stromquellen bleiben konstant.

$$\frac{1}{j\omega C_S} = \frac{1}{j2\pi f C} \left( \underbrace{1}_{\cancel{1}} - j \frac{f_T}{f} \right) = -\frac{f_T}{2\pi C f^2}$$

$$\left[ -\frac{f_T}{2\pi C f^2} \right] \rightarrow -\frac{1}{\cancel{1} \cancel{1} \cancel{1}} \frac{\Omega}{\cancel{1} \cancel{1} \cancel{1}} = -R$$

V. BOOTSTRAPEFFEKT

Wenn  $\underline{A}$  den Wert 1 hat oder etwas darunter, dann bezeichnet man diesen Effekt als Bootstrapeffekt. Bootstrapping bedeutet, sich an den eigenen Stiefelstrümpfen (Bootstrap) hochzuziehen. Der Widerstand  $\underline{Z}$  geht mit der Verstärkung  $\underline{A} = 1$  nach  $\infty$  (siehe Formel 2). Bei den Werten etwas kleiner als eins werden hohe Werte erreicht.

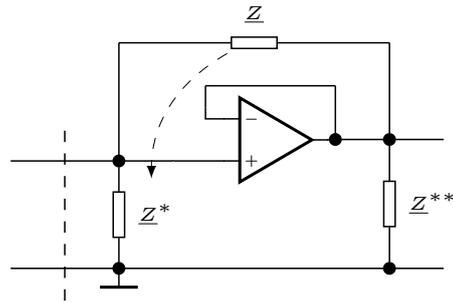


Abbildung 12. Bootstrapeffekt beim Operationsverstärker

$$\underline{Z}^* = \frac{\underline{Z}}{1 - \underline{A}} \quad \underline{A} = 1 \cdot \frac{1}{1 + \frac{1}{A_{0DC}}} \cdot \frac{1}{1 + j \frac{f}{f_T}}$$

$$\underline{A} = \frac{A_{0DC}}{1 + A_{0DC}} \cdot \frac{1}{1 + j \frac{f}{f_T}}$$

$$\underline{Z}^* = \underline{Z} \frac{(\underbrace{1}_{\cancel{1}} + A_{0DC})(1 + j \frac{f}{f_T})}{(1 + A_{0DC})(1 + j \frac{f}{f_T}) - A_{0DC}}$$

$$= \frac{\underline{Z} \cdot A_{0DC} (1 + j \frac{f}{f_T})}{1 + \underbrace{A_{0DC}}_{\cancel{1}} + j \frac{f}{\cancel{f_T}} + j \frac{f}{f_1} - \underbrace{A_{0DC}}_{\cancel{1}}}$$

$$\underline{Z}^* = \underline{Z} \cdot A_{0DC} \cdot \frac{1 + j \frac{f}{f_T}}{1 + j \frac{f}{f_1}} \quad (5)$$

Der Bootstrapeffekt entspricht im Bodediagramm dem umgekehrten Millereffekt (siehe Abb. 11). Die Gleichungen 5 und 4 sind identisch. Der Widerstand  $\underline{Z}^{**}$  spielt meistens keine Rolle, da er am Ausgang des Operationsverstärkers parallel zur internen Spannungsquelle des Operationsverstärkers liegt.

VI. UMGEKEHRTER BOOTSTRAPEFFEKT

Der umgekehrte Bootstrapeffekt (Gleichung 6) entspricht im Bodediagramm dem Millereffekt (siehe Abb. 9). Die Gleichungen 6 und 3 sind identisch.

$$\underline{Z}^* = \underline{Z} \cdot (1 - \underline{A}) = \underline{Z} \left( 1 - \frac{A_{0DC}}{1 + A_{0DC}} \frac{1}{1 + j \frac{f}{f_T}} \right)$$

$$= \frac{\underline{Z}}{1 + A_{0DC}} \frac{1 + \underbrace{A_{0DC}}_{\cancel{1}} + j \frac{f}{\cancel{f_T}} + j \frac{f}{f_1} - \underbrace{A_{0DC}}_{\cancel{1}}}{1 + j \frac{f}{f_T}}$$

$$\underline{Z}^* = \frac{\underline{Z}}{A_{0DC}} \cdot \frac{1 + j \frac{f}{f_1}}{1 + j \frac{f}{f_T}} \quad (6)$$

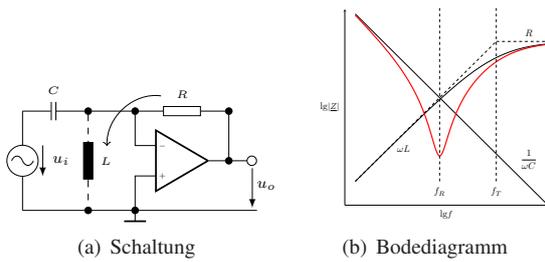


Abbildung 13. Problemfall 1, dargestellt mit dem Millereffekt

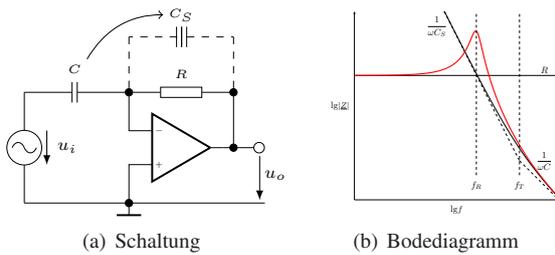


Abbildung 14. Problemfall 1, dargestellt mit dem umgekehrten Millereffekt

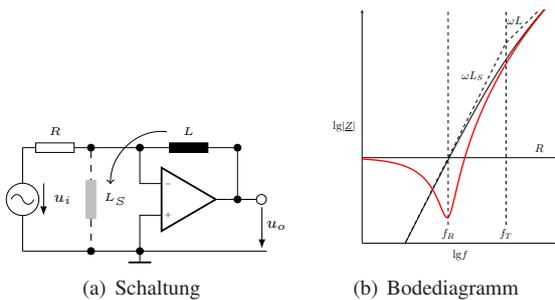


Abbildung 15. Problemfall 2, dargestellt mit dem Millereffekt

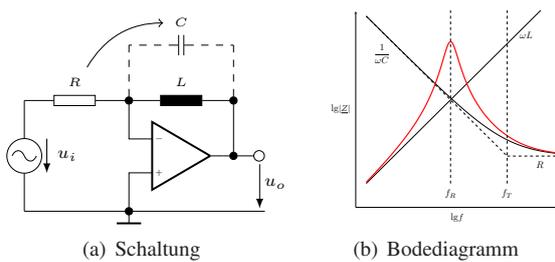


Abbildung 16. Problemfall 2, dargestellt mit dem umgekehrten Millereffekt

## VII. PROBLEMFÄLLE BEI OP-SCHALTUNGEN

### A. Invertierender Verstärker

Abbildungen 13, 14, 15 und 16 stellen Problemfälle dar, die bei einem invertierenden Verstärker auftreten können. Man kann das zurückgekoppelte Bauteil zum Eingang transformieren, oder das Bauteil am Eingang parallel zur Rückkopplung. In allen dargestellten Fällen erkennt man eine Resonanz. Kapazitäten am Eingang und Induktivitäten in der Rückführung ergeben meist Stabilitätsprobleme.

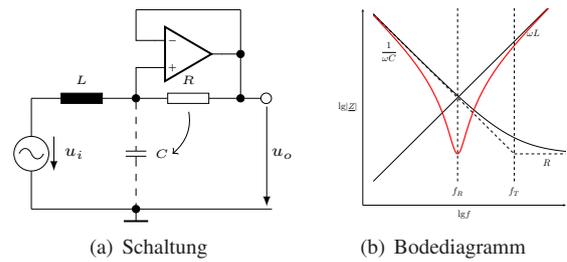


Abbildung 17. Problemfall 3, dargestellt mit dem Bootstrapeffekt

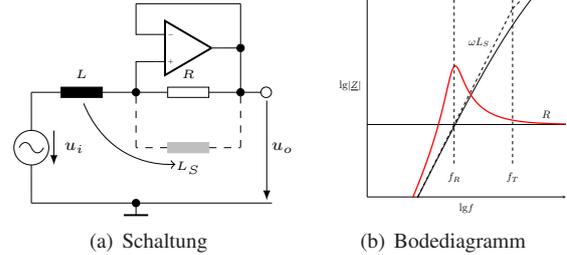


Abbildung 18. Problemfall 3, dargestellt mit dem umgekehrten Bootstrapeffekt

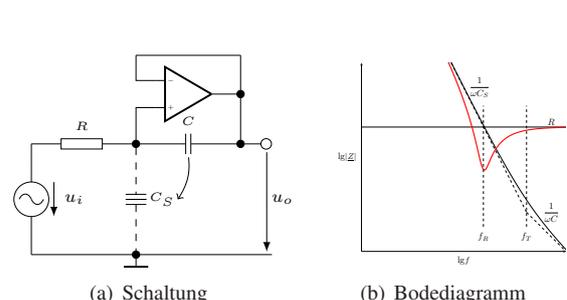


Abbildung 19. Problemfall 4, dargestellt mit dem Bootstrapeffekt

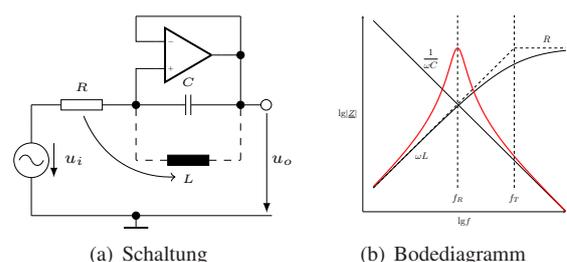


Abbildung 20. Problemfall 4, dargestellt mit dem umgekehrten Bootstrapeffekt

### B. Nichtinvertierender Verstärker

Abbildungen 17, 18, 19 und 20 stellen Problemfälle dar, die bei einem nicht invertierenden Verstärker auftreten können. Man kann das zurückgekoppelte Bauteil zum Eingang transformieren, oder das Bauteil am Eingang parallel zur Rückkopplung. In allen dargestellten Fällen erkennt man eine Resonanz. Kapazitäten in der Rückführung und Induktivitäten am Eingang ergeben meist Stabilitätsprobleme.

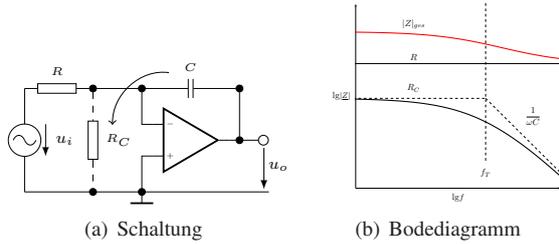


Abbildung 21. Kein Problemfall 1, dargestellt mit dem Millereffekt

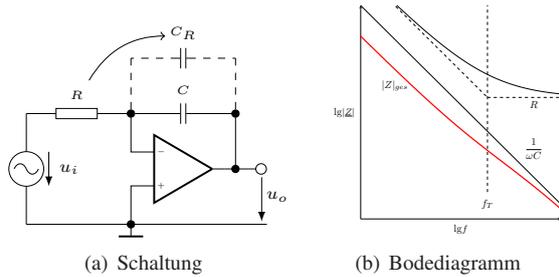


Abbildung 22. Kein Problemfall 1, dargestellt mit dem umgekehrten Millereffekt

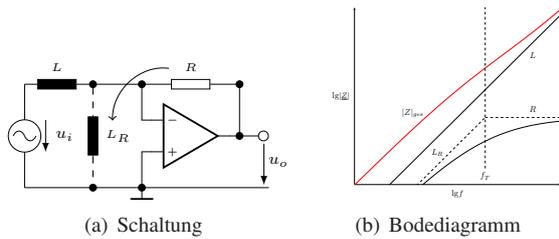


Abbildung 23. Kein Problemfall 2, dargestellt mit dem Millereffekt

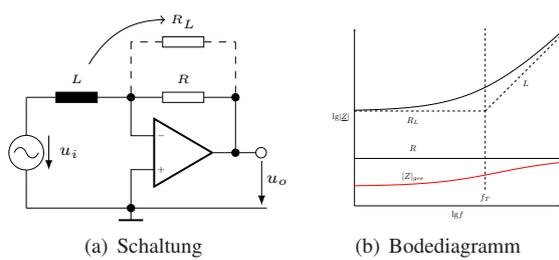


Abbildung 24. Kein Problemfall 2, dargestellt mit dem umgekehrten Millereffekt

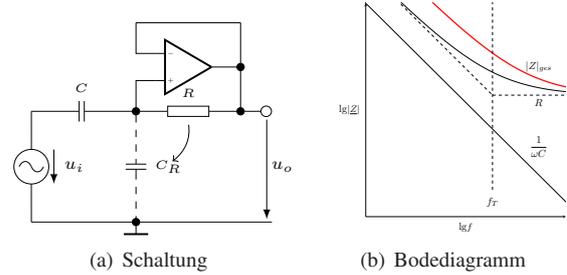


Abbildung 25. Kein Problemfall 3, dargestellt mit dem Bootstrapeffekt

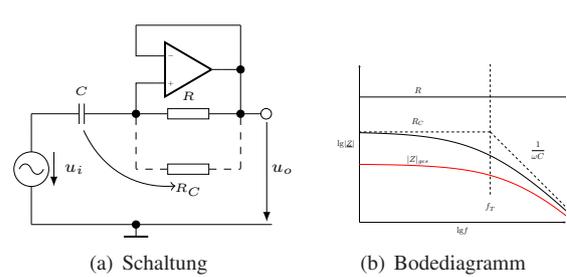


Abbildung 26. Kein Problemfall 3, dargestellt mit dem umgekehrten Bootstrapeffekt

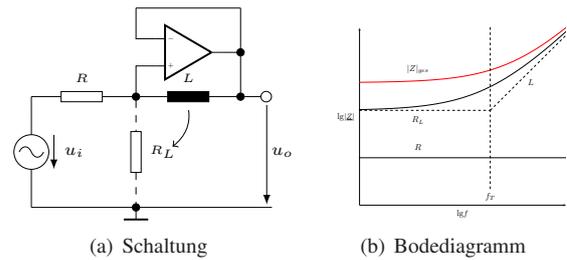


Abbildung 27. Kein Problemfall 4, dargestellt mit dem Bootstrapeffekt

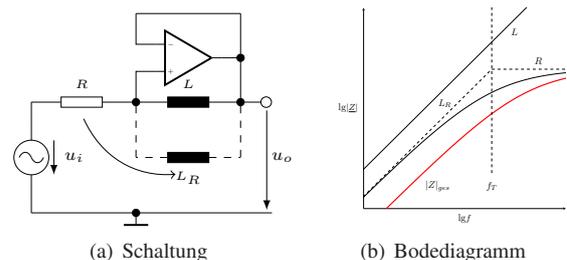


Abbildung 28. Kein Problemfall 4, dargestellt mit dem umgekehrten Bootstrapeffekt

VIII. KEINE PROBLEMFÄLLE BEI OP-SCHALTUNGEN

A. Invertierender Verstärker

Die Abbildungen 21, 22, 23 und 24 zeigen prinzipielle invertierende Schaltungen, die keine Probleme mit der Stabilität haben. Kapazitäten in der Rückführung und Induktivitäten am Eingang ergeben meist keine Stabilitätsprobleme.

B. Nichtinvertierender Verstärker

Die Abbildungen 25, 26, 27 und 28 zeigen prinzipielle nichtinvertierende Schaltungen, die keine Probleme mit der Stabilität haben. Kapazitäten am Eingang und Induktivitäten in der Rückführung ergeben meist keine Stabilitätsprobleme.

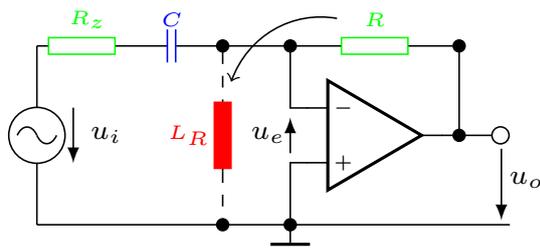


Abbildung 29. Millertransformation

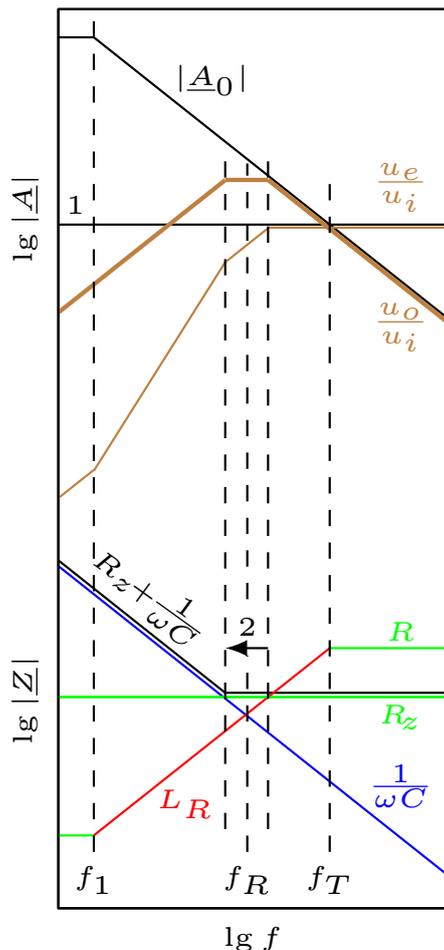


Abbildung 30. Berechnung und Konstruktion der Ausgangsspannung

IX. KONSTRUKTION DER AUSGANGSSPANNUNG

Beispiel: Differenzierer

Abbildung 29 zeigt die Schaltung eines kompensierten Differenzierers. Der Widerstand  $R$  in der Rückführung wird mit dem Millereffekt gegen Null transformiert. Am Eingang des Operationsverstärkers entsteht eine Serienschaltung aus  $RC$  und  $L_R$ . Hierbei wurden die beiden ebenfalls entstehenden Widerstände  $R$  bei  $f > f_T$  und  $\frac{R}{A_{0DC}}$  bei  $f < f_1$  weggelassen. Bei  $R_z = 0$  würden sie das Überschwingen bei einem hö-

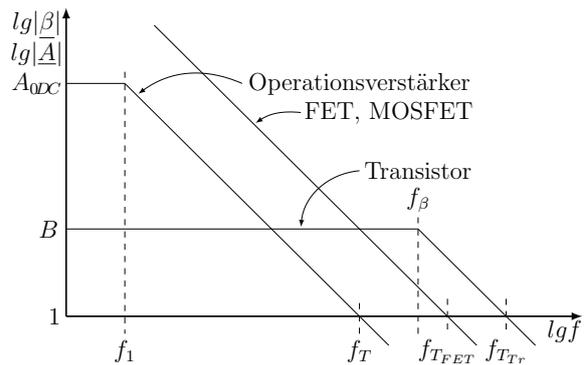


Abbildung 31. Vergleich Leerlaufverstärkung und Stromverstärkung

heren Wert begrenzen. Für den Spannungsteiler ergibt sich:

$$\frac{u_e}{u_i} = \frac{j\omega L_R}{j\omega L_R + R_z + \frac{1}{j\omega C}} = 1 \parallel \frac{j\omega L_R}{R_z + \frac{1}{j\omega C}}$$

Diese Gleichung kann man in zwei Quotienten mit einem Parallelzeichen verbunden umzeichnen. Bei einer Parallelschaltung bestimmt immer der kleinere Teil das Ergebnis. In Abbildung 30 wurde  $R_z$  so gewählt, dass sich der Faktor  $a = 2$  ergibt. Bis zur Frequenz  $f_R \cdot \sqrt{2}$  wird der Spannungsteiler  $\frac{u_e}{u_i}$  von  $\frac{j\omega L_R}{R_z + \frac{1}{j\omega C}}$  bestimmt,

danach von 1. Die Ausgangsspannung  $u_o$  ergibt sich aus der Multiplikation von  $\frac{u_e}{u_i}$  und  $|A_0|$ .

A. Schaltungen mit Transistor oder Feldeffekttransistor

Abbildung 31 zeigt den dualen Vergleich der Leerlaufverstärkung eines Operationsverstärkers mit der Stromverstärkung eines Transistors oder Feldeffekttransistors. Beim Transistor liegt der Bereich der Transformationen zwischen den Frequenzen  $f_\beta$  und  $f_{T_r}$ . Beim FET und MOSFET fehlt eine duale Frequenz  $f_1$  wie beim Operationsverstärker. Der Junction-FET und der MOSFET können bei der Kleinsignalbetrachtung gleich behandelt werden. Beim Operationsverstärker basiert das Millertheorem auf der Gleichheit der Ströme, beim Transistor, FET und MOSFET auf der Gleichheit der Spannungen vor und nach der Transformation. Beim Operationsverstärker werden die Bauteile zwischen der Rückkopplung und dem Eingang transformiert, beim Transistor können Bauteile zwischen der Basis und dem Emitter hin und her transformiert werden, beim FET und MOSFET zwischen Gate und Drain.

LITERATURVERZEICHNIS

[1] A. Zwick, J. Zwick, and X. P. Nguyen, *Signal- und Rauschanalyse mit Quellenverschiebung: Elektronische Schaltungen grafisch gelöst*, ser. SpringerLink Bücher. Berlin, Heidelberg: Springer Vieweg, 2015. [Online]. Available: <https://doi.org/10.1007/978-3-642-54037-0>



**Albrecht Zwick** lehrte seit 1974 als Professor an der Hochschule Mannheim und hält auch nach seiner Pensionierung 2011 noch Vorlesungen auf dem Gebiet der analogen Schaltungstechnik mit der Spezialisierung auf Stabilität elektronischer, rauscharmer Schaltungen.



**Bernd Vettermann** ist seit 1994 an der Hochschule Mannheim im Institut für integrierte Schaltkreise als Ingenieur tätig. Er erhielt 2006 von der Universität Mannheim den akademischen Grad eines Doktors der Naturwissenschaften.

# Universal Memory Automaton and Automated Verilog HDL Code Generation for a Cache Coherency Snooping Protocol

Matthias W. Fertig

**Abstract**—This paper introduces the concept of Universal Memory Automata (UMA) and automated compilation of Verilog Hardware Description Language (HDL) code at Register Transfer Level (RTL) from UMA graphs for digital designs. The idea is based on the observation that Push Down Automata (PDA) are able to process the Dyk-Language - commonly known as the balanced bracket problem - with a finite set of states while Finite State Machines (FSM) require an infinite set of states. Since infinite sets of states are not applicable to real designs, PDAs appear promising for types of problems similar to the Dyk-Language. PDAs suffer from the problem that complex memory operations need to be emulated by a specific stack management. The presented UMA therefore extends the PDA by other types of memory, e.g. Queue, RAM or CAM. Memories that are eligible for UMAs are supposed to have at least one read and one write port and a one-cycle read/write latency. With their modified state-transfer- and output-function, UMAs are able to operate user-defined numbers, configurations and types of memories. Proof of concept is given by an implementation of a cache coherency protocol, i.e. a practical problem in microprocessor design.

**Index Terms**—Finite state machines, sequential circuits, sequential synthesis, high-level and register-transfer level synthesis, methodologies for EDA, automata extensions, processors and memory architectures, push down automata, HDL compilation, digital design automation.

## I. INTRODUCTION

### A. Finite State Machines

In Digital Engineering, Finite State Machines (FSMs) [1]–[3, 7, 8, 12] are a standard design element to process regular languages. They are typically utilized to implement control logic. FSMs are given by a set

$$FSM = (S, S_0, F, \Sigma, \Gamma, \delta, \omega) \quad (1)$$

where  $S$  is a finite set of states,  $S_0$  is the initial state,  $F \subseteq S$  is a finite set of final states,  $\Sigma$  is the input alphabet,  $\Gamma$  is the output alphabet,  $\delta$  is the state transfer function and  $\omega$  is the output function. State transfers are defined by the *state transfer function*

$$\delta : \begin{cases} S \times \Sigma & \rightarrow S \\ s' & = \delta(s, \sigma) \end{cases} \quad (2)$$

Matthias W. Fertig, matthias.fertig@htwg-konstanz.de, University of Applied Sciences HTWG Konstanz, Alfred Wachtel Straße 8, 78462 Konstanz.

where a destination state  $s' \in S$  is reached from a source state  $s \in S$  by processing an input  $\sigma \in \Sigma$ . Outputs are defined by the *output function*

$$\omega : \begin{cases} S \times \Sigma & \rightarrow \Gamma \\ \gamma & = \omega(s, \sigma) \end{cases} \quad (3)$$

where  $\gamma \in \Gamma$  is an output symbol derived from a state  $s \in S$  and input  $\sigma \in \Sigma$ . This architecture is of type *Mealy* since the output function depends on inputs and states. *Moore* and *Simple Moore* architectures are deduced from simplified output functions. FSMs are called *finite* because they are built from a finite set of states, implemented by a state memory. *State memory* is of size  $\log_2(N_S)$  and required in all known FSM architectures, where  $N_S$  is the number of states in the finite set of states.

### B. Push Down Automata

Push Down Automata (PDA) [4]–[6, 9]–[11, 13], also called Stack Automata, are given by a set

$$PDA = (S, S_0, F, \Sigma, \Gamma, \delta, \omega, \mathbf{A}, \mathcal{X}) \quad (4)$$

where common elements equal those of the FSM.  $\mathcal{X}$  is the *stack alphabet* (I-B1) to operate the *stack memory* (II-A2)  $\mathbf{A}$ .

$$\mathcal{X} = \{PUSH(), POP(), TOP(), NOP()\} \quad (5)$$

The state transfer function of a PDA  $\delta$  is

$$\delta : \begin{cases} S \times \Sigma \times \mathcal{X} & \rightarrow S \times \mathcal{X} \\ (s', x) & = \delta(s, \sigma, x) \end{cases} \quad (6)$$

where  $x \in \mathcal{X}$  are stack operations and  $w \in \Sigma^*$  is a word on the input alphabet. The output function of the PDA is

$$\omega : \begin{cases} S \times \Sigma \times \mathcal{X} & \rightarrow \Gamma \times \mathcal{X} \\ (\gamma, x) & = \omega(s, \sigma, x) \end{cases} \quad (7)$$

1) *Stack alphabet*: A set of operations on the stack  $\mathbf{A}$ .  $\text{PUSH}(\mathbf{A}, w)$  puts an element  $w \in \Sigma^*$  on the stack,  $\text{POP}(\mathbf{A})$  returns the first element from the stack and deletes the first element,  $\text{TOP}(\mathbf{A})$  returns the first element and keeps the first element,  $\text{NOP}(\mathbf{A})$  is a no-operation on the stack, sometimes called the empty operation. Stack operations are utilized in state-transfer- and output-functions to conditionally read and write the stack.

## II. THE UNIVERSAL MEMORY AUTOMATON

Universal Memory Automata (UMA) are an innovative concept for operating multiple (parallel) memories in a finite state graph. While FSMs utilize state memory and PDAs utilize state and stack memory only, UMAs extend the state memory by multiple ( $k$ ) memories of selectable and configurable type. Eligible types of memories in this paper are last-in first-out (Stack)  $\mathbf{A}$  like in PDAs, first-in first-out memory (Queue)  $\mathbf{Q}$ , random-access (RAM)  $\mathbf{R}$  and content-addressable (CAM) memory  $\mathbf{C}$ . UMAs allow an arbitrary number and a user-defined configuration of those memories. PDAs are of course able to emulate all these types of memory by a specific stack management, but the additional effort makes PDAs more a theoretic model of computation than an applicable tool for real designs. If several and potentially different types of memories are desired, the effort to model those with the single stack of a PDA becomes even higher. To resolve this issue, this paper introduces the idea of operating multiple memories of variable type and configuration, which is particularly relevant for real applications. This is performed by the UMA.

Universal Memory Automata (UMA) are given by a set

$$UMA = (S, S_0, F, \Sigma, \Gamma, \delta, \omega, \mathbf{X}^k, \mathcal{X}^k) \quad (8)$$

where common elements equal those of the PDA and  $\mathbf{X}^k$  is an  $k$ -dimensional set of memories of selectable type, i.e.  $\mathbf{X} \in \{\mathbf{A}, \mathbf{Q}, \mathbf{R}, \mathbf{C}\}$ , operated by a  $k$ -dimensional memory alphabet  $\mathcal{X}^k$ . UMAs thereby control  $k$  memory instances  $\mathbf{X}^k$ ,  $k \in \mathcal{N}_0^+$ , each of different type if desired.

Memory operations  $\text{PUSH}()$ ,  $\text{POP}()$ ,  $\text{TOP}()$  and  $\text{NOP}()$  are contained in each of the  $k$ -dimensional memory alphabet,  $\mathcal{X}^k$ . Dimensions of memory and memory alphabet are connected to each other so that the  $i$ -th memory alphabet  $\mathcal{X}_i$  operates on the  $i$ -th memory instance  $\mathbf{X}_i$ , where  $1 \leq i \leq k$ . Operations are shared from PDA-theory and implemented by virtual functions in UMA-theory to perform according to the principle of operation of the adjacent memory. For memories of type  $\mathbf{R}$  and  $\mathbf{C}$  an address is required while for memories of type  $\mathbf{Q}$  and  $\mathbf{A}$  accesses is self-organized by the memory using an internal address pointer. The address width of the RAM or CAM is  $\log_2(N)$ , where  $N$  is the number of addressable entries

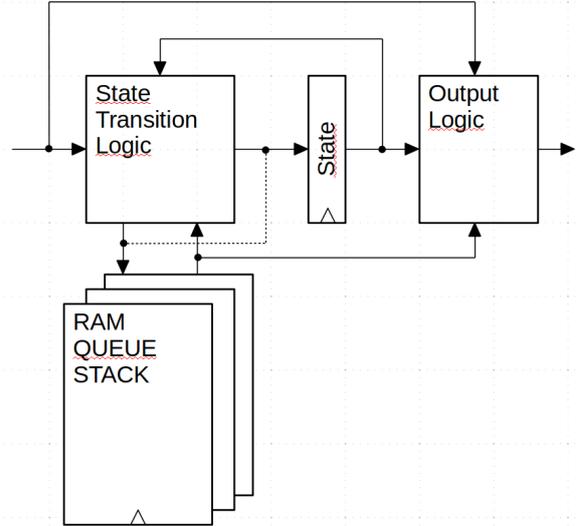


Figure 1. Universal Memory Automata (UMA) architecture block-diagram.

of the memory. Such type of UMA is of type *Mealy* architecture. *Moore* and *Simple-Moore* architectures are derived from a simplification of the output function similar to PDA and FSM. The UMA architecture blockdiagram is shown in Figure 1.

### A. Types of memory

1) *State memory*: The UMA state memory is equal to the state memory of FSM and PDA.

2) *Last-In First Out (Stack)*: A stack  $\mathbf{A}$  is a memory with  $N_A$  entries, each of width  $n_A$ . If  $n_A = 8$ , the operation  $\text{PUSH}(\mathbf{A}, 8'h00)$  stores eight bits, all of them zero, on top of the stack. A subsequent operation  $\text{PUSH}(\mathbf{A}, 8'hFF)$  stores eight bits, all of them one, on top of the stack.  $8'hFF$  is returned by  $\text{POP}(\mathbf{A})$  and  $8'h00$  by another  $\text{POP}(\mathbf{A})$ . Using two  $\text{TOP}(\mathbf{A})$  operations would return  $8'hFF$  twice.  $\text{POP}$  on an empty and  $\text{PUSH}$  on a full stack returns a zero entry plus an error indication.

3) *First-In First Out (Queue)*: A queue  $\mathbf{Q}$  is a memory with  $N_Q$  entries, each of width  $n_Q$ . For  $n_Q = 8$  the operation  $\text{PUSH}(\mathbf{Q}, 8'h00)$  stores eight bits, all of them zero, at the end of the queue. A subsequent operation  $\text{PUSH}(\mathbf{Q}, 8'hFF)$  stores eight bits, all of them one, at the end of the same queue.  $8'h00$  is returned by  $\text{POP}(\mathbf{Q})$  and  $8'hFF$  by another  $\text{POP}(\mathbf{Q})$ . The queue is then empty. Using two  $\text{TOP}(\mathbf{Q})$  operations would return  $8'h00$  twice.  $\text{POP}$  on an empty and  $\text{PUSH}$  on a full queue returns a zero entry plus an error indication.

4) *Random-Access Memory (RAM)*: A RAM  $\mathbf{R}$  is a memory with  $N_R$  entries, each of width  $n_R$  and an address of width  $\log_2(N_R)$ . For  $N_R = 8$ , the operation  $\text{PUSH}(\mathbf{R}, 3'b000, 8'hFF)$  stores eight bits, all of them one, at address 0.  $8'hFF$  is returned by  $\text{POP}(\mathbf{R}, 3'b000)$  and  $8'h00$  by another  $\text{POP}(\mathbf{R}, 3'b000)$ . Using two  $\text{TOP}(\mathbf{R}, 3'b000)$  operations would return  $8'hFF$  twice.

5) *Content-Addressable Memory (CAM)*: A CAM  $\mathbf{C}$  is a memory with  $N_C$  entries, each of width  $n_C$  and an address of width  $\log_2(N_C)$ . CAMs are high-speed search engines to return the address of content-specific memory entries in one cycle. For  $n_C = 8$  the operation  $\text{PUSH}(\mathbf{C}, 3'b000, 8'h00)$  stores eight bits, all of them zero, at address 0. A subsequent operation  $\text{PUSH}(\mathbf{C}, 3'b111, 8'hFF)$  stores eight bits, all of them one, at address 7.  $\text{TOP}(\mathbf{C}, 8'h00)$  returns the first address with content  $8'h00$ , i.e. 0.  $\text{POP}(\mathbf{C}, 8'hFF)$  returns the first address with content  $8'hFF$ , i.e. 7.  $\mathbf{C}$  is assumed to be initialized zero. While  $\text{POP}$  deletes the entry,  $\text{TOP}$  will keep the entry. It is left up to the designer of the CAM how to organize deleted entries, to indicate if entries are not found and to return the entry if required. UMA-theory is able to manage all types of indications within a single-cycle boundary.

### B. n-dimensional memory operations

If memory operations do not return status informations on write accesses for evaluation in state transfer logic, read memory operations  $\mathcal{X}_-$  are assigned to the input side of the state transfer function (eq. 6) and write operations  $\mathcal{X}_+$  are assigned to the output side. A set of *read operations* is defined

$$\mathcal{X}_- = \{\text{TOP}(), \text{POP}()\} \quad (9)$$

for the input side of the state transfer function and a set of *write operations*

$$\mathcal{X}_+ = \{\text{PUSH}(), \text{NOP}()\} \quad (10)$$

for the output side of the state transfer function, where  $\mathcal{X} = \mathcal{X}_- \cup \mathcal{X}_+$ . In state transfers, only one operation per memory instance is allowed for cycle alignment reasons. This concept is called  $l$ -dimensional reading and  $m$ -dimensional writing in this paper. For  $l$ - plus  $m$ -dimensional memory operations, at least  $\max(l, m)$  memories are required. The state transfer function  $\delta$  then becomes

$$\delta : \begin{cases} S \times \Sigma \times \mathcal{X}_-^l & \rightarrow & S \times \mathcal{X}_+^m \\ (s', x_{1+}, \dots, x_{m+}) & = & \delta(s, \sigma, x_{1-}, \dots, x_{l-}) \end{cases} \quad (11)$$

and the output function

$$\omega : \begin{cases} S \times \Sigma \times \mathcal{X}_-^l & \rightarrow & \Gamma \\ \gamma & = & \omega(s, \sigma, x_{1-}, \dots, x_{l-}) \end{cases} \quad (12)$$

where  $x_{i-} \in \mathcal{X}_{i-}$  and  $x_{i+} \in \mathcal{X}_{i+}$  are read and write operations on memory  $X_i$  respectively.

### C. Example of three-dimensional memory operations in state transfers

In case of a cache coherency snooping protocol with inputs  $rd$ ,  $tag$  and  $idx$ , states ID and RD and RAM

memories **TAG** and **MESI**, a state transition from idle state (ID) to read state (RD) is given by

$$\begin{aligned} & (RD, \text{PUSH}(\mathbf{TAG}, idx), \\ & \text{PUSH}(\mathbf{MESI}, 4'b0100)) \quad (13) \\ & = ID \wedge rd \wedge (tag \neq \text{TOP}(\mathbf{TAG}, idx)) \end{aligned}$$

where  $tag$  is the address tag and  $idx$  is the address index, i.e. the cache line index and the read/write addresses for **TAG** and **MESI**. In this transition  $l = 2$  and  $m = 1$ , i.e. a two-fold write operation and a one-fold read operation. A sequence of three memory accesses plus state transfer is coded into a single state transition. Eq. 13 reads as follows:

(Right-hand side:) **IF** the UMA is in idle state ID and a read operation occurs, i.e.  $rd$  is true, and the address tag does not equal the tag in the memory **TAG**,

(Left-hand side:) **THEN** transfer to read state RD, store the tag to RAM **TAG** at address  $idx$ , store the exclusive bit to RAM **MESI** at address  $idx$ .

## III. CACHE COHERENCY PROTOCOL

Caches are fast and comparably small memories nearby processor cores to provide low-latency data access and to avoid expensive accesses to main memory. As caches store copies of data, data consistency problems arise in case of multiple processors operating on the same (shared) data.

A cache coherency protocol performs book-keeping of memory entries loaded and modified by one or more processor cores in a multiprocessor system. The protocol aims to secure consistent data exchange between processor cores and the memory of the system, called coherency. *Dedicated caches* are caches dedicated to and therefore accessed by only a single processor core while *shared caches* are accessed by more than one processor core. In this paper, a cache coherency protocol for dedicated caches is implemented, where cache and cache coherency hardware are assigned to a single processor core. Such assignments of core, cache and cache coherency protocol are called *adjacent* in this paper. The coherency protocol engine is implemented by a UMA which observes (snoops) the address bus for memory access activities. A well known algorithm is the MESI-protocol, where a set of control bits indicates whether a cache line is modified (M), exclusive (E), shared (S) or invalid (I).

In this implementation, the UMA has five states. The **idle state** ID indicates no load/store operations on the address bus. The **read state** RD indicates that a read operation is performed by the adjacent processor. The **write state** WR indicates that the adjacent processor performs a write operation. The **remote read state** rRD indicates that a remote and not the adjacent processor performs a read operation. The **remote write state** rWR indicates that a write operation is performed

by a remote and not the adjacent processor. Slightly deviating from the original protocol, cache coherency status bits M, E, S and I are defined as follows for the UMA implementation.

A cache line is set to status **modified (M)**, if a write operation is performed by the adjacent processor on data associated with the cache line. A cache line is set to status **exclusive (E)**, if a read operation is performed by the adjacent processor on data associated with the cache line. A cache line is set to status **shared (S)**, if a read operation is performed by a remote processor on data associated with the cache line. A cache line is set to status **invalid (I)**, if a write operation is performed by a remote processor on data associated with the cache line.

MESI status indications are mutually exclusive and based on the following **two assumptions** for dedicated caches.

**First**, it is not relevant for data coherency if a remote processor core keeps a copy of data in its cache while an adjacent processor reads data from memory. Processor cores reading data will mark the respective cache line exclusive (E) while all others, keeping the cache line as well, will mark it shared (S) at this moment. Consistency problems occur in case of remote processors writing on this address and are resolved by setting all duplicate cache entries in the system to status invalid (I).

**Second**, it is not relevant for data coherency if an adjacent processor core keeps a copy of data in its cache while a remote processor writes data to memory. Processor writing data will mark the respective cache line modified (M) while all others, keeping the cache line as well, will mark it invalid (I) at this moment.

The two observations make a book-keeping of remote core memory activities obsolete in dedicated caches and thereby reduce the overhead for the protocol engine.

For cache entries marked invalid by the coherency protocol, an invalid is indicated by the output function, as shown by  $I=1$  in transitions (3.2), (C.2), (14.2) and (17.2). This supports counting of cache misses as part of hardware performance analysis. Transitions (1), (F) and (16) are split into (\*.1) and (\*.2) to avoid unnecessary memory action and thereby save power. If power consumption is not critical, SET\_TAG is performed regardless of TAG\_MATCH, i.e. TAG\_MATCH in (\*.1) can be removed and (\*.2) dropped completely.

#### A. Address organization and cache coherency

In computer systems, memory addresses are organized by page-tag and byte-index, where byte-index is a defined set of low order address bits to index bytes in random access memory, and page-tag are the remaining high-order bits. As every cache line might store a power of two number of bytes, the number of bytes in a cache line is given by  $N = 2^b$  where

$0 \leq b < n$ . Hence,  $b = \log_2 N$  low order index bits of the memory address might be unused in the cache address. In case of associative cache organization, a certain number of low order bits of the tag  $a$  are used to be associated with sets of associative cache memory.



In case of an  $m$ -way associative cache,  $a = \log_2(m)$  low order bits of the tag become part of the cache address to associate cache blocks. The tag is reduced by  $a$  bits.

$$tag = ADDR[n-1 : a + INDEX + b] \quad (14)$$

$$idx = ADDR[a + INDEX + b - 1 : b] \quad (15)$$

One-way associative caches are obtained for  $a = 0$ . For simplicity let  $a = 0$  in this paper so that the cache is one-way associative.

#### B. UMA graph for a cache coherency protocol

A refined version of the protocol in [14] has been implemented in this work. The cache coherency protocol is implemented by a set of states  $S = \{ID, RD, WR, rRD, rWR\}$  and an initial state  $S_0 = ID$ , named "idle".  $RD$  and  $rRD$  are states to account for "read" and "remote read" situations while  $WR$  and  $rWR$  are states to indicate "write" and "remote write" situations. The inputs to decide on situations are  $rd, wr, cp, res_n, addr(31:0)$  where  $rd$  indicates read accesses,  $wr$  indicates write accesses on address  $addr$ .  $cp$  indicates whether the access is performed by the adjacent core ( $cp = 1$ ) that the coherency protocol is accounting for or performed by a remote core ( $cp = 0$ ). For this implementation 32-bit addresses, six index bits  $idx = addr(7:2)$  and 24 tag bits  $tag = addr(31:8)$  are used. The protocol operates on two memories of type RAM, called MESI and TAG. MESI and TAG are addressed by  $idx$  to store the cache coherency status and address tag respectively. Figure 2 shows the UMA graph. State transfer and output expressions are shown in Table IV. To simplify expressions, the implementation allows the use of constants (Table III).

### IV. AUTOMATED HDL GENERATION OF UMA GRAPHS

The tools chain is built on an XML-like format to define the UMA graph for automated Verilog HDL compilation. A later version is supposed to support a visual graph representation with automatic import and export function of the shown configuration files. Simple configuration files consist at least of the series of tags, name, inputs, outputs, states, stateTransfers (Figure 3). The memory tag is required for PDAs and UMAs. The `expr` tag is required to define constant expressions.

Table I  
UMA STATE TRANSFER EXPRESSION FOR A TRANSITION FROM IDLE STATE TO READ STATE.

Comment	Expression (to be entered in a single line)
Source state	ID,
Condition	rd && cp && addr[31 : 8] != TOP(TAG, addr[7 : 2]),
Destination state	RD,
Output assignment	I = 1'b0,
Memory activity	PUSH(TAG, addr[7 : 2], addr[31 : 8]) PUSH(MESI, addr[7 : 2], EXCLUSIVE)

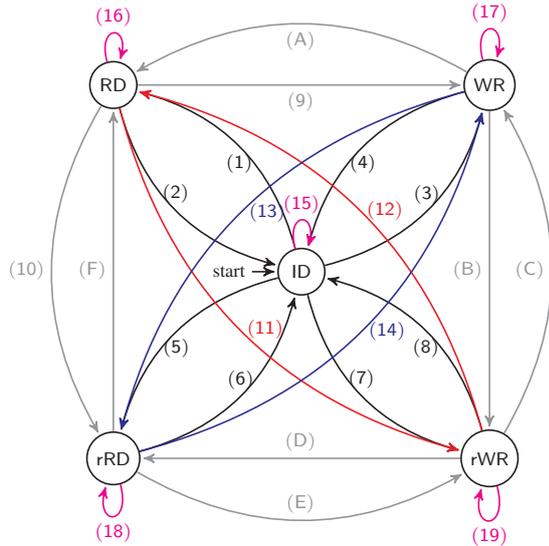


Figure 2. UMA graph for cache coherency protocol. State transition conditions are given in Table III and Table IV.

### A. Tags definitions

1) *name*: `< name > MESI < /name >` defines an UMA named MESI, which becomes the HDL module name.

2) *inputs*: `< inputs > clk, res_n, rd, wr, cp, addr[31 : 0] < /inputs >` defines a series of inputs to be used in expressions, state transfer and output functions.

3) *outputs*: `< outputs > I < /outputs >` defines an output I to indicate an invalid situation, where a cache entry is marked invalid in case of a remote write access on a memory location accounted by the protocol.

4) *states*: `< states > ID(100), RD(000), WR(001), rRD(010), rWR(011) < /states >` defines a set of states with manual encoding.

5) *stateTransfers*: `< stateTransfers > state transitions < /stateTransfers >` defines the state transfer function by a series of state transitions, line by line.

6) *state transitions*: Each state transition is defined by a single line with expression of the form

*source state, condition, destination state, output assignment, memory activity*

```

<name> MESI </name>
<type> UMA, Mealy </type>
<inputs> clk,res_n,rd,wr,cp,addr[31:0] </inputs>
<outputs> I </outputs>
<triggeredge> posedge clk, negedge res_n </triggeredge>
<stateCoding> man </stateCoding>
<states> ID(100), RD(000), WR(001), rRD(010), rWR(011) </states>
<memory>
TAG,ram,24,64; // 64 entries, 24 bit each
MESI,ram,4,64; // 64 entries, 4 bit each
</memory>

```

Figure 3. UMA configuration file (header section) for a cache coherency protocol.

Table II  
UMA MEMORY CONFIGURATIONS FOR THE CACHE COHERENCY PROTOCOL.

memory name	memory type	memory width	memory depth
TAG	ram	24	64
MESI	ram	4	64

where *src* is a source state, *condition* is a boolean expression containing  $l$ -fold memory read operations, i.e.  $\mathcal{X}_-^l$ . *dst* is the destination state and *output assignment* is an assignment to one or more outputs. *Memory activity* indicates  $m$ -fold memory write operations, i.e.  $\mathcal{X}_+^m$ . An example state transition corresponding to the expression in eq. 13 is shown in Table I. There, a memory named TAG of type ram is defined. TAG has a width of 12 bits and sixteen entries, i.e. the address width is four bits.

7) *memory*: `< memory > memory definitions < /memory >` defines a series of memories.

8) *memory definitions*: A memory is defined by a line with an expression of the form

*memory name, memory type, width, depth*

where *memory name* is the name of the memory, here TAG and MESI. *Memory type* is cam, ram, queue or stack. *Depth* is the number of bits per memory entry and *number* is the number of entries. The address width is determined automatically from the  $\log_2$  of the number of entries. An example memory configuration is shown in Table II.

9) *expr tag*: `< expr > constant expression definitions < /expr >` defines a constant expression to be used in state transfer and output expressions.

10) *constant expression definitions*: Constant expression can be used to easily avoid large expressions

```
RAM # (4, 64)      MESI_RAM(.clk(MESI_RAM_clk), .res_n(MESI_RAM_res),
                .wr(MESI_RAM_wr), .wr_addr(MESI_RAM_wr_addr[5:0]), .wr_data(MESI_RAM_wr_data[3:0]),
                .rd(MESI_RAM_rd), .rd_addr(MESI_RAM_rd_addr[5:0]), .rd_data(MESI_RAM_rd_data[3:0]));

wire READ;
wire TAG_MATCH;
always @( * )
begin
  case ( STATE )
  ID : begin
    READ & ~ TAG_MATCH ) begin
      NEXT_STATE <= RD;
      TAG_RAM_status <= PUSH_TAG_RAM(addr[7:2],addr[31:8]);
      MESI_RAM_status <= PUSH_MESI_RAM(addr[7:2],EXCLUSIVE);
    end ...
```

Figure 4. Auto-generated Verilog HDL for memory instantiation and the state transition shown in Table I

Table III  
EXPRESSION CONSTANTS FOR THE CACHE COHERENCY  
PROTOCOL SHOWN IN TABLE IV ON PAGE 41.

Constant	Expression
READ	= res_n & cpu & rd
WRITE	= res_n & cpu & wr
R_READ	= res_n & !cpu & rd
R_WRITE	= res_n & !cpu & wr
NOP	= !res_n    (!rd & !wr)
TAG_MATCH	= TOP(R, addr(7:2)) == addr(31:8)
IS_MODIFIED	= TOP(MESI, addr(7:2)) == 4'b1000
IS_EXCLUSIVE	= TOP(MESI, addr(7:2)) == 4'b0100
IS_SHARED	= TOP(MESI, addr(7:2)) == 4'b0010
IS_INVALID	= TOP(MESI, addr(7:2)) == 4'b0001
SET_TAG	= PUSH(R, addr(7:2), addr(31:8))
SET_MODIFIED	= PUSH(MESI, addr(7:2), 4'b1000)
SET_EXCLUSIVE	= PUSH(MESI, addr(7:2), 4'b0100)
SET_SHARED	= PUSH(MESI, addr(7:2), 4'b0010)
SET_INVALID	= PUSH(MESI, addr(7:2), 4'b0001)

in state transfers, in particular when complex memory operations are involved. A constant expression is defined by a line of the form *constant = expression*. The constant expressions used in the cache coherency protocol are shown in Table III.

11) *Others*: Other tags are for example type to choose between *uma*, *pda*, or *fsm* and architectures *mealy*, *moore* and *smoore*. *TriggerEdge* is used to configure a positive (*pos*) or negative (*neg*) clock edge, *stateCoding* to select manual, one-hot or gray encoding. These types of tags are defaulted automatically if unused or otherwise contained in the header section of a UMA definition file as shown in Figure 3.

### B. HDL generation

Boolean logic for state transfer, output functions and memory instances are compiled at Register Transfer Level (RTL) with Verilog HDL. The entire protocol including testbench is derived from a 140 lines configuration file.

The Cache Coherency Protocol for the finite state graph shown in Figure 2 instantiates two memories of type *ram*, named *TAG* and *MESI*, each with 64 entries, addressed by 6 address bits, i.e. *index = addr(7:2)*. Every cache line is 32 bits wide and thereby stores four bytes so that the *b*-field of the address is 2 bits wide, i.e. *index(1:0)*. Memory *TAG* stores the 24 bits wide address tag *addr(31:8)* and memory *MESI* stores

one-hot encoded coherency settings for cache lines addressed by *index(7:2)*, i.e. Modified, Exclusive, Shared and Invalid.

### C. Module interface

The module interface is derived from the header section shown in Figure 3. Name, inputs and output definitions are used straight forward in the module interface.

### D. Memory instantiation

Memories are instantiated as defined in the configuration file. The code fragments in Figure 4 shows the definition and instantiation of the RAM *MESI* and the state transition discussed in Table I.

### E. State transfer with memory read and output function with write access

State transfers are derived from the `<stateTransfers>` tag as shown Table I. Verilog code is compiled as shown in Figure 4, where the state transfer function is built on a case statement encapsulated in an *always* block. There, the current state is evaluated and the state transitions shown in the UMA graph (Figure 2) provide the expressions for the state transition logic (Table IV).

1) *Example*: In Table I and arc 1.1 in Table IV: **IF** the protocol engine is in state *idle* (*ID*), a read operation occurs and the corresponding read access to the TAG RAM causes a tag match, i.e. (*READ & TAG\_MATCH*), **THEN** the UMA transfers to read state (*RD*) and a write access is performed to change the coherency status of the cache entry to Exclusive when the read state is reached. This complex operation is defined by a single state transfer. The corresponding waveform is shown in Figure 5.

### F. Output function

Outputs are set according to the state and state transfer conditions in the case statement in case of a *Mealy* architecture and in a separate case statement and according to the state only in case of a *Moore* or *Simple Moore* architecture.

Table IV  
STATE TRANSITION DEFINITIONS FOR THE GRAPH REPRESENTATION OF THE CACHE COHERENCY PROTOCOL IN FIGURE 2.  
EXPRESSION CONSTANTS ARE DEFINED IN TABLE III ON PAGE 40.

Arc	SRC State	Condition	TGT State	Output	Memory operation
(start,2,4,6,8,15)	*	NOP	ID		
(1.1)	ID	READ & TAG_MATCH,	RD,	,	SET_EXCLUSIVE
(1.2)	ID	READ & !TAG_MATCH,	RD,	,	SET_TAG SET_EXCLUSIVE
(3.1)	ID	WRITE & TAG_MATCH & !IS_INVALID,	WR,	,	SET_MODIFIED
(3.2)	ID	WRITE & TAG_MATCH & IS_INVALID,	WR,	I=1,	
(5)	ID	R_READ & TAG_MATCH,	rRD,	,	SET_SHARED
(7)	ID	R_WRITE & TAG_MATCH,	rWR,	,	SET_INVALID
(9)	RD	WRITE & TAG_MATCH,	WR,	,	SET_MODIFIED
(A)	WR	READ & TAG_MATCH,	RD,	,	SET_EXCLUSIVE
(B)	WR	READ & TAG_MATCH,	rWR,	,	SET_INVALID
(C.1)	rWR	WRITE & TAG_MATCH & !IS_INVALID,	WR,	,	SET_MODIFIED
(C.2)	rWR	WRITE & TAG_MATCH & IS_INVALID,	WR,	I=1,	
(D)	rWR	R_READ & TAG_MATCH & IS_MODIFIED,	rRD,	,	SET_SHARED
(E)	rRD	R_WRITE & TAG_MATCH,	rWR,	,	SET_INVALID
(F.1)	rRD	READ & TAG_MATCH,	RD,	,	SET_EXCLUSIVE
(F.2)	rRD	READ & !TAG_MATCH,	RD,	,	SET_TAG SET_EXCLUSIVE
(10)	RD	R_READ & TAG_MATCH,	rRD,	,	SET_SHARED
(11)	RD	R_WRITE & TAG_MATCH,	rWR,	,	SET_INVALID
(12.1)	rWR	READ & TAG_MATCH,	RD,	,	SET_EXCLUSIVE
(12.2)	rWR	READ & !TAG_MATCH,	RD,	,	SET_TAG SET_EXCLUSIVE
(13)	WR	R_READ & TAG_MATCH,	rRD,	,	SET_SHARED
(14.1)	rRD	WRITE & TAG_MATCH & !IS_INVALID,	WR,	,	SET_MODIFIED
(14.2)	rRD	WRITE & TAG_MATCH & IS_INVALID,	WR,	I=1,	
(16.1)	RD	READ & TAG_MATCH,	RD,	,	SET_EXCLUSIVE
(16.2)	RD	READ & !TAG_MATCH,	RD,	,	SET_TAG SET_EXCLUSIVE
(17.1)	WR	WRITE & TAG_MATCH & !IS_INVALID,	WR,	,	SET_MODIFIED
(17.2)	WR	WRITE & TAG_MATCH & IS_INVALID,	WR,	I=1,	
(18)	rRD	R_READ & TAG_MATCH,	rRD,	,	SET_SHARED
(19)	rWR	R_WRITE & TAG_MATCH,	rWR,	,	SET_INVALID

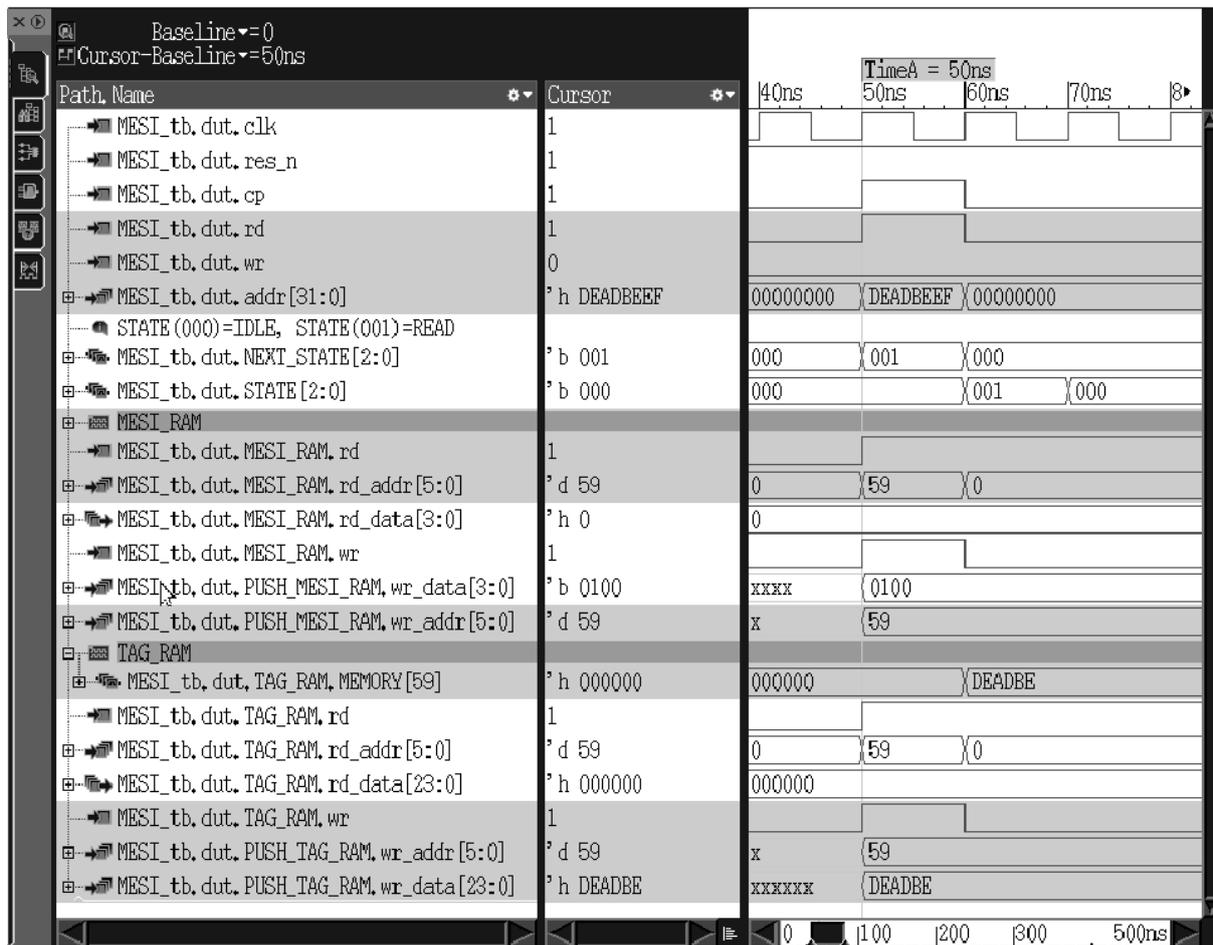


Figure 5. Read access of own CPU ( $cp = rd = 1$ ) causing a TAG to be stored in TAG\_RAM and coherency status 'exclusive', i.e. 0100, into MESI\_RAM.

## V. CONCLUSIONS

Universal Memory Automata (UMA) provide a formalism for using complex memory configurations and operations in finite state graphs. Compared to Push Down Automata (PDA), where only a stack and state memory are utilized or Finite State Machine (FSM), where only a state memory is utilized, Universal Memory Automata (UMA) extend these concepts by additional memory organizations, e.g. Queue, RAM and CAM. UMAs allow an arbitrary set and configuration of memories to be operated at the same time, where complex and multiple read/write memory operations are included in the state transfer and output functions. This feature does not enable UMAs to process another class of languages in the Chomsky Hierarchy but it makes them more flexible and intuitively applicable than PDAs and FSMs when implementing complex applications. With only 140 lines of parametrized configuration, a complex and adaptable Cache Coherency Protocol including test-bench is automatically implemented in the Hardware Description Language (HDL) Verilog at Register Transfer Level (RTL). The proposed theory is in general applicable to all types of memories within the state transfer cycle alignments, i.e. single-cycle read/write access. This shows the potential of Universal Memory Automata (UMAs) and provides the possibility for further extensions.

## REFERENCES

- [1] T.L. Booth. 1962. *Sequential Machines and Automata Theory* (1st ed.). Number 67-25924. John Wiley and Sons, Inc., New York. Library of Congress Card Catalog.
- [2] J. Carroll and D. Long. 1989. *Theory of Finite Automata with an introduction to Formal Languages*. Prentice Hall.
- [3] A. Gill. 1962. *Introduction to the Theory of Finite-state Machines*. McGraw-Hill.
- [4] J.E. Hopcroft and J.D. Ullman. 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, MA. ISBN 0-201-02988-X.
- [5] L. Boasson, J.-M. Autebert, J. Berstel. 1997. *Context-Free Languages and Push-Down-Automata*. Vol. 1. Springer-Verlag, 111-174.
- [6] J.D. Ullman, J.E. Hopcroft. 1967. "Nonerasing Stack Automata". *Journal of Computer System Sciences 1 (1967)*, 166–186. [https://doi.org/10.1016/s0022-0000\(67\)80013-8](https://doi.org/10.1016/s0022-0000(67)80013-8).
- [7] E.J. McCluskey. 1965. *Introduction to the Theory of Switching Circuits* (1st ed.). McCraw-Hill, New York. Library of Congress Card Catalog.
- [8] M. Minski. 1967. *Computation: Finite and infinite Machines* (1st ed.). Prentice-Hall, New Jersey.
- [9] L.J. Stockmeyer, R.E. Ladner, R.J. Lipton . 1984. "Stack Automata and Compiling". *SIAM J. Comput.* 13, (1984), 135–155. ISSN 0097-5397. <https://doi.org/10.1137/0213010>.
- [10] M.A. Harrison, S. Ginsburg, S.A. Greibach. 1967. "One-way Stack Automata". *J. ACM 14*, 1 (1967), 389–418. <https://doi.org/10.1145/321386.321403>
- [11] M.A. Harrison, S. Ginsburg, S.A. Greibach. 1967. "Stack Automata and Compiling". *J. ACM 14*, 1 (1967), 172–201. <https://doi.org/10.1145/321371.321385>.
- [12] S. Seshu. 1963. "Introduction to the theory of finite-state machines". *Proc. IEEE 51*, 9, Sep. 1963, 1275–1275. ISSN 0018-9219. <https://doi.org/10.1109/PROC.1963.2548>.
- [13] M. Snipser. 1997. *Introduction to the Theory of Computation*. PWS Publishing. ISBN 0-534-94728-X. Section 2.2: Pushdown Automata, pp. 101-114.
- [14] P. Zyska. 2018. *Implementation of a cache coherency protocol with extended Push Down Automata*. Bachelor Thesis, HTWG Konstanz, University of Applied Sciences.



**Matthias W. Fertig** received his academic degree Dipl. Inf. (MSCE) from the University of Mannheim in 2001. From 2003 to 2011 he worked at the IBM Research and Development GmbH in Böblingen, Germany, in the field of microprocessor design for IBM P- and Z-Servers. He holds patents in the field of computer architecture and silicon photonics and is a recipient of several IBM innovation and plateau awards. In 2011 he received a Dr. rer. nat. (PhD)

for his research on electromagnetic simulation at the department of Optoelectronics of Heidelberg University. From 2011 to 2015 he worked as a project manager for Dialog Semiconductor and as a CPM for Volvo CE. Since 2015 he has been a professor of computer engineering and holds the professorship for digital systems at the Konstanz University of Applied Sciences.

# Aufbau von Simulations- und Prognosemodellen energietechnischer Anlagen unter Einsatz von Deep Learning Verfahren

Carsten Stephan, Johannes Mast, Stefan Rädle, Joachim Gerlach

**Zusammenfassung**—Die vorliegende Arbeit untersucht am konkreten Beispiel einer Photovoltaikanlage die Generierung von Simulations- bzw. Prognosemodellen unter Anwendung algorithmischer Lösungsansätze der Künstlichen Intelligenz und des Maschinellen Lernens, insbesondere des Deep Learning. Im Rahmen der Arbeit wird aufgezeigt und evaluiert, wie auf Grundlage realer Randbedingungen, etwa Wetterdaten des Deutschen Wetterdienstes oder der geographische Standort einer Anlage, unter Einsatz von Deep Learning Ansätzen konkrete „personalisierte“ Modelle zur Berechnung der stündlichen Stromerzeugung der Anlage generiert werden können. Modelle dieser Art bilden insbesondere im Kontext erneuerbarer Energien eine wichtige Grundlage für die Realisierung von Steuerungskonzepten und Betreiberstrategien in virtuellen Kraftwerten und leisten somit einen wichtigen Beitrag zum Gelingen der Energiewende.

**Schlüsselwörter**—Erneuerbare Energien, Energiewende, Photovoltaik, Simulations- und Prognosemodelle, Künstliche Intelligenz, Maschinelles Lernen, Künstliche Neuronale Netze, Deep Learning

## I. EINLEITUNG

Das von der Bundesregierung beschlossene Klimaschutzprogramm [1] fordert ein massives Umdenken in Bezug auf erneuerbare Energien. Hierbei sind nicht nur Großunternehmen zur Handlung gefordert, sondern auch eine Mithilfe von Kommunen, Verbänden, bis hin zu Privatpersonen unabdingbar. Die Umstrukturierung der heutigen Stromerzeugung dient dabei nicht nur der Vermeidung von Treibhausgasen, welche u.a. durch Kohlekraftwerke erzeugt werden, sondern soll insbesondere auch die Stabilität des Stromnetzes sowie die Versorgungssicherheit garantieren. Erneuerbare Energiequellen stehen dabei oftmals vor der Problemstellung, dass diese aufgrund von variierenden Wetterverhältnissen schwer einschätzbar und zu kontrollieren sind. Diese Charakteristik spiegelt sich auch in der folgend betrachteten Photovoltaik wider.

In diesem Zusammenhang stellt der Betrieb von virtuellen Kraftwerken, dem logischen Zusammenschluss von stromerzeugenden Anlagen unterschiedlicher Art,

Carsten Stephan, stephaca@hs-albsig.de, Johannes Mast, mast@hs-albsig.de, Stefan Rädle, raedlest@hs-albsig.de, und Joachim Gerlach, gerlach@hs-albsig.de sind Mitglieder der Hochschule Albstadt-Sigmaringen, Jakobstraße 6, 72458 Albstadt-Ebingen

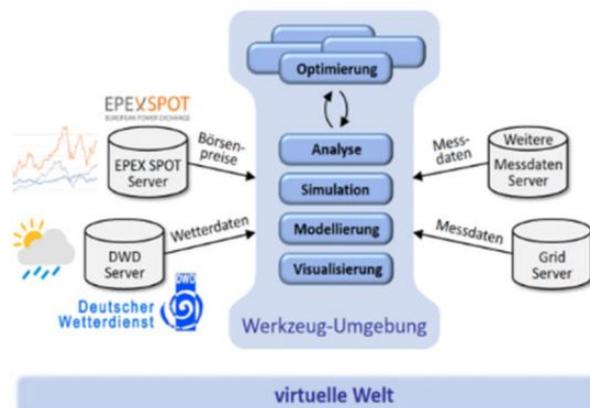


Abbildung 1. Werkzeugumgebung des Arbeitsbereichs.

einen immer größer werdenden Entscheidungsfaktor bei der Konzeption und dem späteren Aufbau von Anlagenverbänden dar. Ziel ist es hierbei, durch die Analyse von gegebenen und prognostizierbaren Umweltfaktoren (Sonne, Wind, Temperatur, Standort), den zur Verfügung stehenden und nutzbaren Anlagentypen (Photovoltaik, Solar, Windkraft, Geothermie, Wasserkraft, etc.) und deren Stromerzeugungsmöglichkeiten auf Grundlage der Umweltfaktoren eine auf künstlicher Intelligenz (KI) basierte virtuelle Umgebung zu schaffen.

## II. KONZEPTIONELLES VORGEHEN

Im Rahmen vorangegangener Arbeiten des Arbeitsbereichs „Cyber-Physical Energy Systems“ der Hochschule Albstadt-Sigmaringen ist eine Werkzeugumgebung (WZU) entstanden, welche die Modellierung, Simulation und Visualisierung hierarchischer, komplexer Systeme aus dem Bereich der Energietechnik ermöglicht.

Die in der Abbildung 1 dargestellte WZU dient zur virtuellen Analyse und Optimierung energietechnischer Szenarien. Auf diese Weise kann die Planung, der Aufbau und der Betrieb stromerzeugender Anlagen im virtuellen Kraftwerk unterstützt und auf virtueller Ebene exploriert werden. Hierdurch können Fragestellungen in Bezug auf die Befriedigung eines gegebenen Lastprofils, einer Minimierung der CO<sub>2</sub>-Emission oder einer Optimierung des Ertrags beim Handel an



Abbildung 2. KI-Schichtenmodell [7].

der Strombörse und weiterer Faktoren systematisch untersucht werden. Da die WZU eine Vielzahl von Schnittstellen zu externen Dienst-Anbietern (Deutscher Wetterdienst (DWD) [2], European Power Exchange (EPEX) [12]) sowie Messdatenservern anderer Partner besitzt, kann ein hochgradig akkurates Abbild konkreter Szenarien und Umweltfaktoren untersucht werden.

Im Gegensatz zu konventionellen Vorgehensweisen, die typischerweise auf der Anwendung physikalischer Modelle energietechnischer Komponenten und Anlagen basieren (entsprechend einer „Button-Up“-/„Whitebox“-Sichtweise) setzt die vorliegende Arbeit auf der Anwendung algorithmischer Lösungsansätze der Künstlichen Intelligenz (KI) und des Maschinellen Lernens (ML) auf (entsprechend einer „Top-Down“-/„Blackbox“-Sichtweise). Im Rahmen der Arbeit wird hierzu das Verhalten einer Photovoltaikanlage (PVA) untersucht und unter Anwendung von Deep Learning Verfahren ein Modell der PVA erzeugt, im weiteren Verlauf als Deep Learning Modell (DLM) bezeichnet.

KI spiegelt dabei ein Teilgebiet der Informatik wider, welcher sich mit der algorithmischen Abbildung von Verhalten befasst, welches von Menschen als „intelligent“ interpretiert werden kann (Abbildung 2). ML verarbeitet in diesem Zusammenhang eine Vielzahl von Daten, welche z.B. realen Messdaten entstammen oder durch externe Dienst-Anbieter bereitgestellt werden. Diese Erkenntnisse können anschließend einem künstlichen neuronalen Netz (KNN) zum Training bereitgestellt werden. Das KNN erzeugt auf Grundlagen von Eingabeparametern (in Form von Input-Neuronen) und derer Gewichtung (Auftrittswahrscheinlichkeit) zusammen mit einer beliebigen Anzahl an verknüpften Hidden-Neuronen, welche das Training eines Best-Case-Szenarios (dem Erlernen des neuronalen Netzes) implementieren, einen zu erwartenden Ausgabewert (Abbildung 3). Eine spezielle Form dieser KNN stellt dabei das Deep Learning dar, welches durch eine Rückkopplung von Neuronen-Schichten stabile Verknüpfung zwischen einzelnen Neuronen erzeugt.

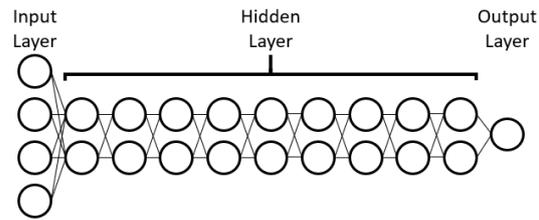


Abbildung 3. Künstliches Neuronales Netzwerk.

### III. AUFGABENSTELLUNG

Die Funktionsweise einer PVA basiert grundlegend darauf, dass einfallendes Sonnenlicht in der Solarzelle eine elektrische Spannung erzeugt. Mehrere Solarzellen werden hierbei elektrisch miteinander in Reihe geschaltet, wobei jede Zelle ungefähr 0,6 Volt bei 5,5 Ampere erzeugt. Diese Leistung wird dabei klassisch in Kilowatt (kW) oder in Kilowattpeak (kWp) angegeben. Da eine Zelle bei voller Bestrahlung eine Leistung von 3,4 W erzeugen kann, benötigt ein 1 kWp-Module ungefähr 300 Zellen, die untereinander verschaltet sind. Hinzu kommen einflussnehmende Faktoren, welche die Leistung der Photovoltaikanlage reduzieren können: Tag-Nacht-Rhythmus, Zenit- und Azimut-Winkel der Sonne, Bewölkung, Luftverschmutzung, usw. Basierend auf diesen Grundlagen kann jedoch nur sehr aufwendig und mit einer hohen Unsicherheit behaftet prognostiziert werden, welche Leistung eine solche PVA über einen gewissen Zeitraum liefert.

DLM können hierfür die Lösung für eine schnelle und kosteneffiziente Prognose liefern. Basierend auf den genannten Einflussfaktoren können PVA oder (wie im Rahmen dieser Arbeit zugrunde gelegt) hinreichend genaue PVA-Simulationen die Leistungsdaten für einen vorher festgelegten Zeitraum erzeugen. Ändert sich jedoch der vorgegebene Zeitraum, die Größe des Photovoltaikmodules oder deren Einflussfaktoren, so müssen diese Anlagen bzw. Simulationen modifiziert und in variiert (bei realen PVA nur bedingt rekonstruierbarer) Konstellation neu berechnet werden. DLM können hingegen jeden Datensatz als Trainingsdatensatz nutzen und das dadurch erzeugte Modell mit jedem weiteren modifizierten Trainingsdatensatz erweitern und optimieren.

Für die Simulation einer PVA wird hierbei auf ein MATLAB/Simulink-Modell entstammend der CARNOT-Modell-Bibliothek [9] zurückgegriffen. Das CARNOT-Modell bietet die Möglichkeit aus prognostizierten Einflussfaktoren die zu erwartende Ausgangsleistung zu berechnen. Sie bietet jedoch keine Möglichkeit zur Anpassung der Größe des eigentlichen Modules und benötigt zudem für die Vorbereitung und Berechnung einer Simulation einen hohen Zeitaufwand. Auf diese Weise können mittels des Simulationsmodells Leistungsdaten im benötigten Umfang generiert werden, welche zum Training des DLM

zum Einsatz kommen. Das daraus entstehende Modell benötigt anschließend keine weitere Modellierung, sondern prognostiziert auf ausschließlicher Grundlage der Einflussfaktoren automatisch die zu erwartende Stromerzeugung für das vorgegebene Zeitintervall.

#### A. Bereitstellung der Ausgangsdaten

Zur Vereinheitlichung der Ausgangsdatensätze wurden die folgenden Kriterien zugrunde gelegt:

- |      |  |           |
|------|--|-----------|
| i)   | Aufstellwinkel der PVA:                      | 30 Grad   |
| ii)  | Ausrichtung der PVA:                         | Süd       |
| iii) | Strahlungsleistung WSW:                      | 500 Watt  |
| iv)  | Strahlungsleistung SSW:                      | 1000 Watt |
| v)   | Strahlungsleistung<br>Zenit-Winkel < 0 Grad: | 0 Watt    |

Auf Grundlage der oben genannten Ausgangswerte kann die stündliche Leistung der Sonneneinstrahlung für jeden Zeitraum berechnet werden. Diese Strahlungsleistung verringert sich ab dem Zeitpunkt der Sommersonnenwende bis zur Wintersonnenwende durch den jeweiligen Anteil der Strahlungsdifferenz. Entgegengesetztes gilt für den Zeitraum von Wintersonnenwende bis zur Sommersonnenwende. Zudem wurde der prozentuale Unterschied zwischen einem Schaltjahr und Nichtschaltjahren beachtet.

Weitere Einflussfaktoren auf die Strahlungsintensität stellen zudem die Werte des Bewölkungsgrades, der Umgebungstemperatur sowie des Luftdrucks dar, welche bei der Berechnung der Gesamtdatensätze berücksichtigt wurden.

#### B. Simulation in MATLAB

Das PVA-Modell von CARNOT wurde für die Erzeugung von Trainingsdaten und zum Einlernen eines PVA-DLM genutzt. Die Eingabedaten zu diesem Modell stellen hierbei aufbereitete Wetter- und Sonnenstanddaten dar, welche als MAT-Datei in die Simulation eingebunden werden. Die bei der Simulation erzeugten Ausgabedaten werden als zu erwartende Ausgaben zum Training des neuronalen Netzwerkes verwendet. Die hierfür benötigten Ausgangsdatensätze, sowie die erzeugten Trainingsdatensätze, müssen nach Abschluss der Simulation ins Dateiformat CSV exportiert werden. Diese dienen dem DLM anschließend als Trainings- und Testdatensätze.

#### C. Erzeugung eines Deep Learning Models

Auf Basis der erzeugten Trainingsdaten wird mittels der Programme Tensorflow [10] und Keras [11] ein DLM unter Einsatz der Programmiersprache Python entwickelt. Nach Abschluss des Trainings wird das erzeugte Modell exportiert und auf einen Testzyklus eines anderen Standortortes angewendet. Ein Trainingszyklus wird dabei durch den Gesamtdatensatz eines

Jahres für einen vorher definierten Standort durchgeführt. Die hierbei erzeugten stündlichen Werte werden mit den simulierten Werten aus CARNOT verglichen und das DLM auf Grundlage der prozentualen Abweichungen angepasst und Neuberechnet. Ziel ist es hierbei, die geeignete Dimensionierung und Konfiguration eines neuronalen Netzes zu generieren.

#### IV. CHARAKTERISTIK VON PHOTOVOLTAIKANLAGEN

PVA generieren ihre Energie aus der durch die Sonne erzeugten Strahlungsenergie. Die ankommende Strahlungsenergie hängt hierbei von mehreren Faktoren, welche den Wirkungsgrad einer PVA wesentlich beeinflussen, ab. Diese Faktoren lassen sich wie folgt zusammenfassen:

- Aufstellwinkel der Strahlung auf die PVA
- Jahreszeit
  - o Sommersonnenwende
  - o Wintersonnenwende
- Umweltfaktoren
  - o Luftverschmutzung
  - o Luftfeuchtigkeit
  - o Wolkendichte
  - o Luftdruck

Ausgehend von diesen Faktoren lässt sich der Auftrittswinkel als einzige Variable manuell regulieren. Alle weiteren Faktoren sind vom Zeitpunkt der Stromerzeugung oder den klimatischen Bedingungen abhängig. Die effizienteste vertikale Ausrichtung einer PVA in den Breitengraden von Deutschland liegt dabei zwischen 30 und 35 Grad, sowie einer exakten Ausrichtung nach Süden. Einen weiteren Faktor in Bezug auf die Strahlungsenergie der Sonne stellt die Jahreszeit dar. So verfügt Deutschland aufgrund des Breitengrades über einen Winter-/Sommersonnenwenden-Rhythmus. Dieser führt dazu, dass zum Zeitpunkt der Sommersonnenwende mit einer maximalen Sonnenleistung von 1000 Watt und zum Zeitpunkt der Wintersonnenwende mit einer maximalen Sonnenleistung von 500 Watt gerechnet werden kann. Bezugnehmend auf die Wettereinflüsse werden für die Berechnungen Datensätze des DWD ausgewertet. Diese beinhalten alle benötigten Messwerte, welche die Grundlage zur Berechnung der auftretenden Restleistung der Sonne auf die PVA bilden.

#### V. LÖSUNGSANSATZ

Zur Aufbereitung der Wetter- und Sonnenstanddaten wurden geeignete Python-Skripte entwickelt und implementiert. Diese passen die unterschiedlichen Ausgangsdatenformate der Wetter- und Sonnenstanddaten an, berechnen fehlende Datensätze und führen anschließend eine Vollständigkeits- und Konsistenzanalyse der aufbereiteten Datensätze durch. Diese Datensätze werden in das Programm MATLAB importiert,

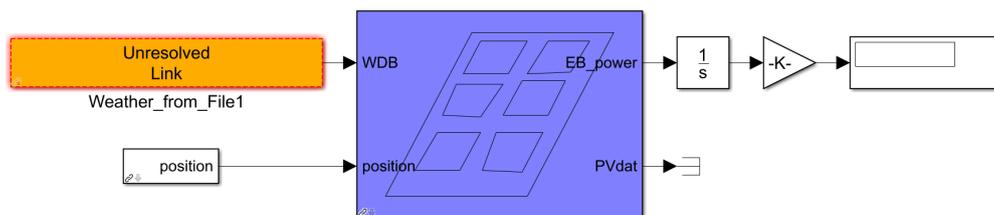


Abbildung 4. Original MATLAB/Simulink-Modell.

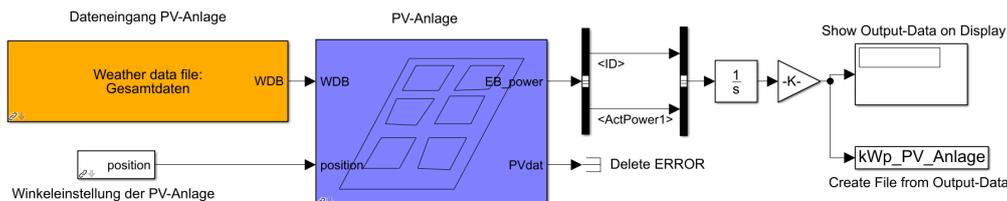


Abbildung 5. Überarbeitetes MATLAB/Simulink-Modell.

woraufhin die PVA-Simulation die Stromerzeugungswerte der Anlage bezugnehmend auf den Standort und das Jahr der Berechnung durchführt.

Die Ausgangsdatensätze, welche aus Wetter- und Sonnenstanddaten bestehen, sowie die durch die Simulation erzeugten Stromerzeugungsdaten, bilden die Grundlage zum Training des neuronalen Netzes. Dieses wird unter der Zuhilfenahme der Python-Bibliotheken Tensorflow und Keras mit Hilfe von Python-Skripten erzeugt. Eine Validierung des DLM findet auf Grundlage der Simulationsdaten der CARNOT-PVA statt.

## VI. VI. ENTWICKLUNG DES DLM

### A. Photovoltaikanlage in MATLAB

Das originale MATLAB/Simulink-Modell (Abbildung 4) muss für die Verwendung als Werkzeug für die Erzeugung von Trainingsdaten zunächst an drei Stellen abgeändert werden (Abbildung 5). Als Dateneingang wird hierbei das Modul „Weather from File“ durch das Modul „Weather Data File“ ausgetauscht. An diesen Dateneingang kann der Gesamtdatensatz in Form einer MAT-Datei importiert werden. Die bei der Simulation erzeugten Ausgabedaten müssen zudem auf die reine Stromerzeugung reduziert werden. Durch das Hinzufügen eines „BUS-Selectors“, sowie eines „BUS-Creators“ am Ausgangspunkt der PVA werden überschüssige Datensätze entfernt und als Matrix wieder zusammengeführt. Übrig bleiben hierbei die Werte „ActPower1“ (Leistung in kWp) und der Zeitstempel in 3600-Sekunden-Schritten (stündlich). Die hierbei entstehenden Datensätze werden im weiteren Verlauf als Trainingsdatensätze verwendet, welche durch das Modul „Save to File“ nach Abschluss der Simulation im MAT-Format gespeichert werden.

### B. Wetterdaten des Deutschen Wetterdienstes

MATLAB benötigt für die Analyse von Wetter- und Sonnenstanddaten bestimmte Anzahl an Parametern.

Diese Parameter, im weiteren Verlauf als Gesamtdatensatz bezeichnet, werden hierbei über das Modul „Import Weather Data from File“ eingebunden. Da dieses Modul jedoch auch für andere Anlagen genutzt werden kann und soll, verfügt es über eine größere Anzahl an Messdatensätze als für eine PVA benötigt werden.

- 00-Zeitachse: Stündlich
- 01-Zeitstempel: Format DDMMYYYY
- 02-Zenit-Winkel: 0 bis 90 Grad
- 03-Azimut-Winkel: 0 bis 360 Grad
- 04-Sonnenstrahlung normal:  $< 1000 \text{ W/m}^2$
- 05-Sonnenstrahlung horizontal:  $< 1000 \text{ W/m}^2$
- 06-Umgebungstemperatur:  $-50$  bis  $+50 \text{ }^\circ\text{C}$
- 07-Temperatur Sonnenstrahlung:  $-50$  bis  $+50 \text{ }^\circ\text{C}$
- 08-Relative Feuchtigkeit: 0% bis 100%
- 09-Niederschlagsmenge:  $\geq 0 \text{ mm}$
- 10-Wolkenmenge: 0 bis 8
- 11-Luftdruck:  $\geq 0 \text{ Pascal}$
- 12-Windgeschwindigkeit:  $\geq 0 \text{ m/s}$
- 13-Windrichtung:  $0^\circ$  bis  $360^\circ$
- 14-Einfallswinkel Sonne zur Ausrichtung PVA
- 15-Einfallswinkel Sonne in Längenrichtung
- 16-Einfallswinkel Sonne in Breitenrichtung
- 17-Direkte Sonneneinstrahlung:  $\leq 1000 \text{ W/m}^2$
- 18-Indirekte Sonneneinstrahlung:  $\leq 1000 \text{ W/m}^2$

Abbildung 6 zeigt die Struktur der vom DWD zur Verfügung gestellten Datensätze.

1) *Generierung des Zeitstempels und Datums:* Den Anfang bei der Befüllung des Gesamtdatensatzes bildet ein generierter stündlicher Zeitstempel. Hierbei wurde beachtet, dass sich die Anzahl der Stunden in einem Schaltjahr von 8760 auf 8784 Einträge erhöht. Eine eindeutige Zuordnung weiterer Ausgangsdaten kann anhand des Datums gewährleistet werden und ermög-

Ausgewählte Produkte, Stationen/Gebiete herunterladen							
Kurzname	Titel	Stationen/Gebiete	Zeitraum von	Zeitraum bis	Größe		
TT_TU_MN009	Stündliche Stationsmessungen der Lufttemperatur auf 2 m Höhe in °C	1 von 616	01.01.1893	29.03.2020	< 5 MB		
RF_TU_MN009	Stündliche Stationsmessungen der relativen Feuchte in %	1 von 616	01.01.1893	29.03.2020	< 5 MB		
R1_MN008	Stündliche Stationsmessungen der Niederschlagshöhe in mm	1 von 1029	01.09.1995	30.03.2020	< 5 MB		
FF_MN008	Stündliche Stationsmessungen der Windgeschwindigkeit in 10 m Höhe in m/s	1 von 666	01.01.1949	29.03.2020	< 5 MB		
DD_MN008	Stündliche Stationsmessungen der Windrichtung in 10 m Höhe in Grad	1 von 666	01.01.1949	29.03.2020	< 5 MB		
PO_MN008	Stündliche Stationsmessungen des Luftdruckes auf Stationshöhe in hPa	1 von 279	01.04.1970	29.03.2020	< 5 MB		
N_MN008	Stündliche Stationsmessungen des Bedeckungsgrades in Achtel	1 von 588	01.01.1949	29.03.2020	< 5 MB		

Abbildung 6. Datensätze des DWD.

licht zugleich eine Fehlererkennung im Bezug auf die Wetter- und Sonnenstanddaten.

2) *Import der Sonnenstanddaten:* Die Datensätze der Sonnenstanddaten, welche aus der Datenbank des Sun-Earth-Tools (SET) [3] importiert werden, weisen, im Vergleich zu den Wetterdaten des DWD, eine komplett andere Struktur auf. Diese ist nicht zeilenweise nach Stunden geordnet, sondern unterteilt sich zeilenweise in Tage und spaltenweise in Stunden, von 00:00 bis 24:00 Uhr. Um dieses Format in ein einheitliches Format umzuwandeln, werden im ersten Schritt die Datensätze der Sonnenstanddaten ausgelesen und anhand des Zeitstempels der passenden Zeile zugeordnet. Fehlende Datensätze, welche mit einem Bindestrich notiert wurden, stellen einen negativ Zenit-Winkel dar, welche beim Importieren durch den Wert 0 ersetzt werden. Dadurch kann der PVA eindeutig signalisiert werden, dass keine Stromproduktion stattfinden kann.

3) *Berechnung der Sonnenleistung:* Auf Grundlage der genannten Ausgangswerte kann die stündliche Leistung der Sonnenstrahlung für jeden Zeitpunkt berechnet werden. Diese Strahlungsleistung verringert sich ab dem Zeitpunkt der Sommersonnenwende bis zur Wintersonnenwende durch den jeweiligen Anteil der Strahlungsdifferenz. Entgegengesetztes gilt für den Zeitraum von Wintersonnenwende bis zur Sommersonnenwende. Zudem wurde der prozentuale Unterschied zwischen einem Schaltjahr und Nichtschaltjahren beachtet.

4) *Berechnung der Sonneneinstrahlung:* Die berechnete Strahlungsleistung kann direkt auf die Werte des Datensatzes „Sonneneinstrahlung horizontal“ übernommen werden, da es sich hierbei um Datensätze ohne Einflussfaktoren, wie z.B. Zenit-, Azimut-Winkel, Wolkenmenge oder Anstellwinkel der Photovoltaikanlage handelt. Die Datensätze: „Sonneneinstrahlung normal“, „direkte Sonneneinstrahlung auf der Oberfläche“ und „indirekte Sonneneinstrahlung auf der Oberfläche“ beinhalten jedoch Einflussfaktoren, welche in die Strahlungsleistung mit eingerechnet werden müssen.

5) *Berechnung der Wolkenmenge:* Die Wolkenmenge wird beim DWD als Bedeckungsgrad von 0 bis

8 angegeben. Da die PVA in MATLAB zur Berechnung jedoch einen Wert zwischen 0 und 1 benötigt, muss dieser durch eine Skalierung angepasst werden. Dabei gilt, dass ein vollständig bedeckter Himmel einen Bedeckungsgrad von 8 und ein ein vollständig klarer Himmel einen Bedeckungsgrad 0 aufweist. Daraus ergibt sich, dass die Strahlungsleistung bei einem Bedeckungsgrad von 8 um 90 Prozent verringert wird, entsprechend einer Strahlungsleistung von 0,1. Gleichsam verringert sich die Strahlungsleistung bei einem Bedeckungsgrad von 0 um 10 Prozent auf eine Strahlungsleistung von 0,9. Grundlage für die fehlenden unteren 10 Prozent der Umrechnung stellt die Restsonnenenergie dar, welche auch bei einem komplett bedeckten Himmel durch die Wolkendecke gelangt. Zudem werden die fehlenden oberen 10 Prozent durch eine natürliche und unnatürliche Luftverschmutzung in den oberen Luftschichten verhindert. Diese werden u.a. durch einwehenden Staubverwirbelungen oder die Abgasaussonderungen überschneidender Luftfahrzeug-routen begründet.

6) *Einfallswinkel der Sonne in Längenrichtung:* Der Datensatz „Einfallswinkel Sonne“ im Gesamtdatensatz ergibt sich aus dem Aufstellwinkel der Photovoltaikanlage, welcher im MATLAB/Simulink-Modell standardmäßig mit 30 Grad angegeben ist und dem aktuellem Zenit-Winkel der Sonne. Bei der Berechnung des Einfallswinkels wurde zudem beachtet, dass bei einem Einfallswinkel von mehr als 90 Grad sich dieser rückläufig verhält, sowie, dass bei einem Zenit-Winkel von unter 0 Grad keine Sonnenstrahlung auf die Photovoltaikanlage projizieren werden kann.

7) *Einfallswinkel der Sonne in Breitenrichtung:* Die Datensätze des Einfallswinkels der Sonne in Breitenrichtung können fast 1:1 von den Azimut-Werten des SET übernommen werden. Jedoch ist auch hier zu beachten, dass wie im vorhergehenden Abschnitt begründet keine Sonneneinstrahlung bei einem Zenit-Winkel unter 0 Grad zu erwarten ist.

8) *Berechnung des Luftdruckes:* Der Luftdruck wird durch den DWD in Hektopascal (hPa) bereitgestellt. MATLAB benötigt für der Berechnung der Simulation

diese Angabe jedoch in Pascal (Pa). Eine Multiplikation mit dem Faktor 100 generiert hierbei den passenden Datensatz.

9) *Übernahme weiterer Datensätze:* Die Datensätze „Umgebungstemperatur“, „Temperatur der Sonnenstrahlung“, „Relative Feuchtigkeit“, „Niederschlagsmenge“, „Windgeschwindigkeit“ und „Windrichtung“ können direkt aus den Quelldatensätzen übernommen werden. Diese Datensätze benötigen keine weitere Aufbereitung.

10) *Korrektur fehlender Datensätze:* Die Wetterdaten des DWD weisen teilweise die Problematik auf, dass Datensätze nicht vollständig generiert wurden. Grund hierfür könnte ein temporärer Ausfall der Messübertragung oder Wartungsarbeiten an einer der Messeinheiten sein. Dieser Umstand führt unweigerlich dazu, dass die zur Verfügung stehenden Daten nicht fortlaufend und damit nicht fehlerfrei in den Gesamtdatensatz integriert werden können. Eine Korrektur der Datensätze kann somit nur anhand eines direkten Abgleichs und einer ungefähren Schätzung der fehlenden Datensätze geschehen. Hierfür wird der Durchschnitt der vorhergehende letzten fünf Werte und der nachfolgenden fünf Werte (falls diese nicht den bei der Fehlererkennung eingetragenen Fehlerwert 403 ausweisen) erstellt und an Stelle des fehlenden Wertes eingetragen. Dies ermöglicht eine genauere approximierte Schätzung der zu ermittelnden Gesamtleistung der PVA als der Eintrag eines Standardmittelwertes.

11) *Speicherung der Gesamtdatensätze:* Der erzeugte Gesamtdatensatz wird im Format einer CSV-Datei abgespeichert. Die dadurch erzeugte Datei kann in MATLAB eingebunden und in das vorgegebene MAT-Format umgewandelt werden.

### C. Erzeugung der Stromerzeugungsdaten in MATLAB

Nachdem alle Einflussfaktoren, welche für die Simulation einer PVA in MATLAB/Simulink benötigt werden, generiert wurden, können diese in MATLAB importiert und zur Erzeugung der Leistungsdaten verwendet werden.

1) *Import der Gesamtdatensätze:* Die Ausgangsdatsätze werden im Dateiformat CSV bereitgestellt. Zur Implementierung in das PVA-Modell von MATLAB/Simulink müssen diese zunächst ins MAT-Format umgewandelt werden. Anschließend ist eine Importierung in das PVA-Modell möglich.

2) *Export der Leistungsdaten:* Nach Abschluss der Simulation weisen die Leistungsdaten einen Datenüberschuss im Bereich der Zeilen 2 bis 12 auf (Vermutung des Autors: Das MATLAB/Simulink-Modell speichert beim Hochfahren der Simulation zusätzliche Temporärwerte zwischen), zusätzlich beinhaltet der Datensatz noch die nicht mehr benötigte Spalte

„Zeitstempel“. Die überschüssigen Datensätze können innerhalb der MAT-Datei entfernt werden. Abschließend stellen die exportierten Leistungsdaten, welche vom MAT-Format ins CSV-Format umgewandelt wurden, den Output-Layer Anteil der Trainingsdatensätze für das DLM bereit.

### D. Erstellung eines Deep Learning Modells

Ziel bei der Erstellung des DLM war es, ein Modell zu generieren, welches die Funktionalität einer PVA hinreichend genau approximiert. Hierzu wurde dem DLM antrainiert, bei welchen Einflussfaktoren welche Ausgangsleistung zu erzeugen ist. Das Python-Skript trainiert hierbei auf Basis der Gesamtdatensätze das zugrunde liegende KNN, um anschließend anhand von Variationen und Prognosen anderer Einflussfaktoren deren zu erwartender stündlicher Stromerzeugung zu schätzen.

1) *Datei- und Ordnerstruktur:* Bevor das Python-Skript mit dem Importieren der Datensätze beginnen kann, muss zunächst die Ordner- und Dateistruktur der Wetter- und Sonnenstanddaten überprüft werden. Sollten beim Download der Datensätze Fehler aufgetreten sein, erkennt eine Funktion fehlende Dateien und bricht die Verarbeitung ab. Der Benutzer des Skripts bekommt daraufhin eine Fehlermeldung, welche ihn dazu auffordert, die fehlende/n Datei/en der Berechnung zur Verfügung zu stellen. Zugleich werden hierbei die Ausgabedateien und Ordnerstrukturen erstellt, welche zum Abspeichern der Gesamtdatensätze benötigt werden.

2) *Skalieren der Datensätze:* Um eine bessere Berechnungsgrundlage, für die Erzeugung des DLM zu schaffen, wurden die Trainingsdatensätze vor der Verarbeitung auf einen einheitlichen Wertebereich skaliert. Diese Angaben beruhen hierbei auf der maximalen Anzahl an Parametern, ohne Dopplungen, wobei der größte Wert auf 1,0 und der niedrigste Wert auf 0,0 abgebildet wurde.

3) *Trainieren des Neuronalen Netzes:* Herzstück des DLM ist das Trainingskript, welches unter Einbindung von Tensorflow und Keras entwickelt wurde. In diesem Skript wird das KNN des DLM eingelernt. Einstellbar sind hierbei die Faktoren „Optimize-Level“, „Active-First“, „Active-Second“, „Loss-Level“, „Hidden-Layer“, „Dense-Layer“ (Anzahl der Neuronen je Hidden-Layer), „Epochen“ und „Batch-Size“. Abbildung 7 zeigt hierbei, welche verschiedenen Einstellungen, bezogen auf die Anzahl der Hidden-Layer und derer Anzahl an Neuronen, beim Testen unterschiedlicher Netzstrukturen welche Treffergenauigkeiten (verglichen zum Referenzmodell) lieferten. So kann bereits bei einer geringen Anzahl an Layern (2 bis 14 Hidden- sowie Dense-Layern) eine sehr hohe Treffergenauigkeit (größer 92 Prozent) erreicht werden. Ein KNN, welches über eine hohe

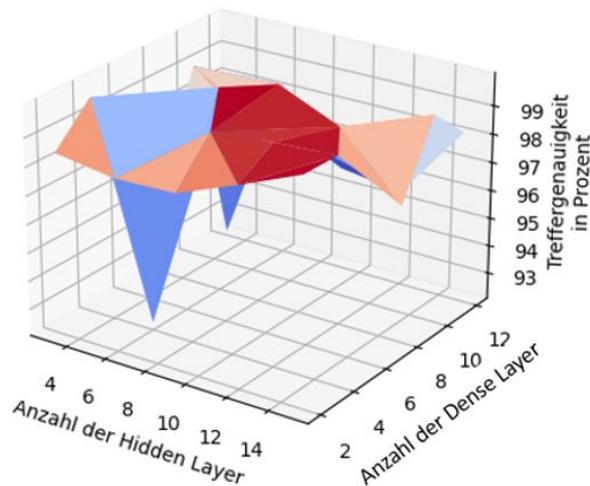


Abbildung 7. Treffergenauigkeit in Abhängigkeit der Layer.

bis sehr hohe Anzahl von Layern (32 bis 128 und mehr) verfügt, wird beim Training eines DLM für PVA somit nicht benötigt. Dies hat des Weiteren den Vorteil, dass beim Training des DLM Rechenleistung und -zeit eingespart werden können.

4) *Export des Deep Learning Models:* Für den Export der erzeugten DLM wurden die drei gängigsten Speicherformate genutzt. Diese beziehen sich auf das Keras-Format (Dateiformat: .pb, inkl. der erzeugten Zufallsgewichtungen zur Initialisierung des KNN), dem „Weights and Architecture“-Format (Dateiformat: .h5) und dem JSON-Format (Dateiformat: .json). Auf Grundlage dieses Exports wurde das Training und die Validierung des DLM durchgeführt. In diesem Zuge wurde das KNN mit der höchsten Treffergenauigkeit der WZU als Standardmodell bereitgestellt.

## VII. TEST UND VALIDIERUNG DES NEURONALEN NETZES

Das gemäß Kapitel VI erstellte DLM wurde im Rahmen experimenteller Untersuchungen für die Prognose der Stromerzeugung eingesetzt. Hierzu wurden die Gesamtdatensätze vom Modell analysiert und stundenweise berechnet. Die daraus entstandenen Datensätze wurden anschließend kumuliert und dem Benutzer als zu erwartende Nennleistung ausgegeben. Abbildung 8 stellt einen Teilausschnitt der trainierten DLM in Bezug auf deren Treffergenauigkeiten dar. Die Tabelle zeigt auf, dass bei einer fast identischen Treffergenauigkeit (der Unterschied liegt hier knapp über 0,2 Prozent) das KNN auf eine geringe Anzahl an Layer, Epochen und Batches reduziert werden kann. Dies verringert nicht nur die Trainingsdauer sondern erhöht zugleich auch die Effizienz von Berechnungen des DLM.

Wie in Abbildung 8 zu erkennen ist, führt die Exploration verschiedener Konfigurationen des KNN bei experimenteller Betrachtung zum Ergebnis, dass ein

	Dense-Layer	Hidden-Layer	Epochs	Batch-Sizes	Accuracy
1	9	2	10	8	99.998227
2	15	8	5	2	99.993839
3	9	8	10	6	99.993776
4	15	2	10	2	99.993654
5	15	4	2	4	99.984667
6	15	12	15	10	99.984471
7	6	4	2	2	99.977920
8	9	2	5	10	99.975583
9	15	2	5	6	99.971816

Abbildung 8. Treffergenauigkeit in Abhängigkeit der Layer.

Aufbau entsprechend Zeile 1, bestehen aus 9 Dense-Layer mit 2 Hidden-Layer, sowie einer Epochenanzahl von 10 und einer Batch-Size von 8, zu einer maximalen Treffergenauigkeit von 99,998227 Prozent führt. Zur Geschwindigkeitsoptimierung des DLM wurde jedoch für die finale Bereitstellung auf die Konfiguration aus Zeile 7 zurückgegriffen. Dieses weist eine nur minimal geringe Treffergenauigkeit in Vergleich zum KNN aus Zeile 1 auf, besitzt jedoch durch seine reduzierte Anzahl an Dense-Layern, Epochen und Batches einen deutlich geringeren Berechnungsaufwand. Dieser resultiert aus der Tatsache, dass sich die Berechnungszeit eines Durchlaufs zur Fehlererkennung aus der Anzahl der zu durchlaufenden Epochen multipliziert mit der Anzahl an Batches in Bezug auf die Größe des Netzes ergibt. Zeile 1 stünde hierbei zur Zeile 7 in einem Verhältnis von 20:1. Die daraus entstehende Berechnungszeit wäre 15-mal schneller als beim KNN aus Zeile 1.

## VIII. GENERIEREN VON STROMERZEUGUNGSDIAGRAMMEN

Nach Abschluss der Berechnungen steht dem Benutzer die Möglichkeit zur Verfügung, sich die Stromerzeugungswerte grafisch darstellen zu lassen. Hierbei werden vom Python-Skript vier Optionen bereitgestellt:

- Darstellung des zeitlichen Verlaufs der Stromerzeugung eines Jahres in Tagen
- Darstellung des zeitlichen Verlaufs der Stromerzeugung eines Jahres in Monaten
- Darstellung des zeitlichen Verlaufs der Stromerzeugung eines Monats in Tagen
- Integrierte Darstellung aller drei vorigen Diagramme

Nach entsprechender Auswahl werden die erzeugten Diagramme angezeigt und im vorgegebenen Verzeichnis abgespeichert. Abbildungen 9 veranschaulicht, wie die Leistung der PVA zur Mitte des Jahres ansteigt und anschließend wieder absinkt.

Wie Abbildungen 9 bis 11 verdeutlichen und aufzeigen, bildet der vom DLM prognostizierte Leistungsver-

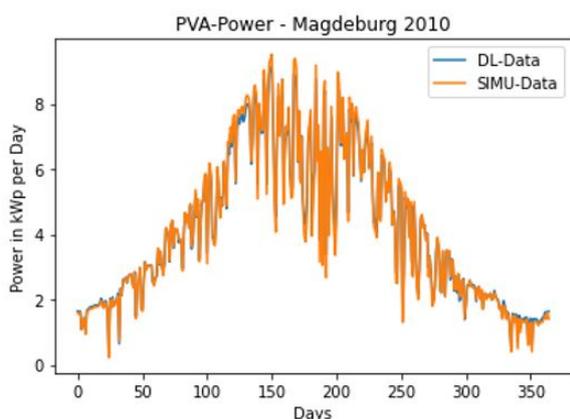


Abbildung 9. Diagramm Stromerzeugung Jahr in Tagen.

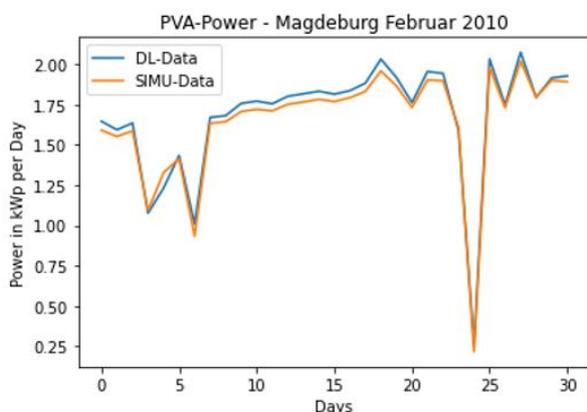


Abbildung 10. Diagramm Stromerzeugung Monat in Tagen.

lauf den zugrundeliegenden Simulationsverlauf hoch akkurat ab.

## IX. ZUSAMMENFASSUNG UND AUSBLICK

Die vorliegende Arbeit demonstriert am Beispiel einer Photovoltaikanlage die gewinnbringende Nutzbarmachung eines durch Deep Learning generierten Modells zur Prognose der stündlichen Stromerzeugung der Anlage. Zur systematischen Exploration und Bewertung erfolgte eine Erzeugung von Trainingsdatensätzen auf Basis eines, die physikalischen Eigenschaften der Anlage abbildenden, MATLAB/Simulink-Modells. In experimentellen Untersuchungen konnte gezeigt werden, dass unter Einsatz von Deep Learning Verfahren KI-basierte Modelle mit extrem hoher Prognosegenauigkeiten (bis zu 99,998227 Prozent) bei deutlicher Reduzierung der Modellkomplexität und -effizienz generiert werden konnten.

Durch Bereitstellung des entstandenen Deep Learning Modells einer Photovoltaikanlage in die Werkzeugumgebung des Arbeitsbereichs konnte auf diese Weise ein wichtiger Beitrag zur simulativen Analyse und Optimierung energietechnischer Szenarien im Kontext eines virtuellen Kraftwerks geleistet werden.

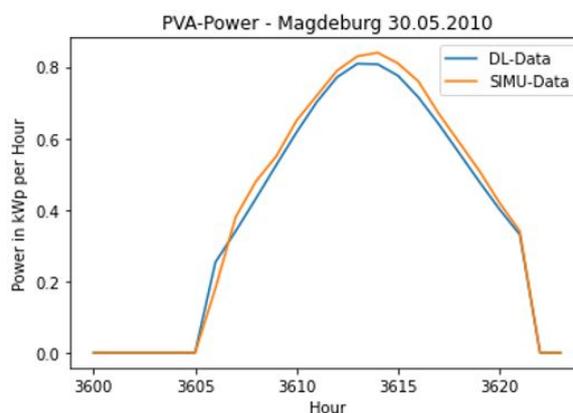


Abbildung 11. Diagramm Stromerzeugung Tag in Stunden.

## DANKSAGUNG

Diese Arbeit wurde gefördert durch das Bundesministerium für Bildung und Forschung (BMBF) im Rahmen des Projekts „Optimierung der Nachhaltigkeit energietechnischer Systeme unter Einsatz intelligenter maschineller Lernverfahren“. (Förderkennzeichen: 13FH757IX6).

## ABKÜRZUNGSVERZEICHNIS

DLM	Deep Learning Modell
PVA	Photovoltaikanlage
DWD	Deutscher Wetterdienst
SET	Sun-Earth-Tool
KI	Künstliche Intelligenz
KNN	Künstliche Neuronale Netze
BMWi	Bundesministerium für Wirtschaft und Energie
WZU	Werkzeugumgebung
WSW	Wintersonnenwende
SSW	Sommersonnenwende

## LITERATURVERZEICHNIS

- [1] Klimaschutzprogramm 2030 der Bundesregierung zur Umsetzung des Klimaschutzplans 2050, Bundesministerium für Wirtschaft und Energie (BMWi), 08.10.2019.
- [2] CDC-Portal des DWD, <https://cdc.dwd.de/portal>.
- [3] Sun-Earth-Tool, <https://www.sunearthtools.com>.
- [4] Viktor Wesselak und Sebastian Voswinckel, *Photovoltaik - Wie Sonne zu Strom wird*, Springer Verlag, 1992.
- [5] Bundesministerium für Wirtschaft und Energie (BMWi), *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung*, 28.09.2010.
- [6] Joachim Steinwedner und Roland Schwaiger, *Neuronale Netze programmieren mit Python*, Rheinwerk Computing, 2 aktualisierte Auflage, 2020.
- [7] Matthieu Deru und Alassane Ndiaye, *Deep Learning mit Tensorflow, Keras und Tensorflow.js*, Rheinwerk Computing, 2 aktualisierte Auflage, 2020.
- [8] Harry Wirth, *Aktuelle Fakten zur Photovoltaik in Deutschland*, Fraunhofer ISE, Download von <https://www.pv-fakten.de>, Fassung vom 14.05.2021.
- [9] CARNOT - Fachhochschule Aachen – Dokumentation unter: <https://fh-aachen.sciebo.de/index.php/s/0hxub0iJrui3ED>.
- [10] Tensorflow - Tensorflow Framework - Webseite, <https://www.tensorflow.org/>, 02.06.2021.
- [11] Keras – Keras Simple. Flexible. Powerful - Webseite, <https://keras.io/>, 02.06.2021.
- [12] EPEX - European Power Exchange, Webseite, <https://www.epexspot.com/en>, 02.06.2021.



**Carsten Stephan** studiert seit 2019 an der Hochschule Albstadt-Sigmaringen an der Fakultät Informatik im Fach IT-Security. Studienbegleitend bringt er sich als studentische Hilfskraft in der Arbeitsgruppe *Cyber Physical Energy Systems* von Prof. Dr. Gerlach ein.



**Johannes Mast** erhielt den akademischen Grad des M.Eng in Systems Engineering im Jahr 2017 an der Hochschule Albstadt-Sigmaringen, nachdem er dort zuvor den Bachelor in Technischer Informatik absolvierte. Aktuell arbeitet er als wissenschaftlicher Mitarbeiter im vom BMBF geförderten Forschungsprojekt OptiNETS - Optimierung der Nachhaltigkeit energietechnischer Systeme unter Einsatz intelligenter maschineller Lernverfahren.



**Stefan Rädle** erhielt den akademischen Grad des M.Eng in Systems Engineering im Jahr 2018 an der Hochschule Albstadt-Sigmaringen, nachdem er dort zuvor den Bachelor in Technischer Informatik absolvierte. Aktuell arbeitet er als wissenschaftlicher Mitarbeiter im vom BMBF geförderten Forschungsprojekt OptiNETS - Optimierung der Nachhaltigkeit energietechnischer Systeme unter Einsatz intelligenter maschineller Lernverfahren.



**Joachim Gerlach** arbeitete nach seinem Studium der Informatik an der TU Karlsruhe und seiner Promotion am Lehrstuhl für Technische Informatik an der Universität Tübingen in den Jahren 2002 bis 2009 bei der Robert Bosch GmbH, Geschäftsbereich Automobilelektronik im Bereich der Halbleiterentwicklung. Seit 2009 ist er Professor an der Hochschule Albstadt-Sigmaringen in den Studiengängen *Technische Informatik*, *IT-Security* und *Systems Engineering*.



## **MULTI PROJEKT CHIP GRUPPE**

### **Hochschule Aalen**

Prof. Dr. Bürkle, (07361) 576-2103  
heinz-peter.buerkle@htw-aalen.de

### **Hochschule Albstadt-Sigmaringen**

Prof. Dr. Gerlach, (07571) 732-9155  
gerlach@hs-albsig.de

### **Hochschule Esslingen**

Prof. Dr. Lindermeir, (0711) 397-4221  
walter.lindermeir@hs-esslingen.de

### **Hochschule Furtwangen**

Prof. Dr. Rülling, (07723) 920-2503  
ruelling@hs-furtwangen.de

### **Hochschule Heilbronn**

Prof. Dr. Gessler, (07940) 1306-184  
gessler@hs-heilbronn.de

### **Hochschule Karlsruhe**

Prof. Dr. Ng, (0721) 925-1520  
Herman-Jalli.Ng@hs-karlsruhe.de

### **Hochschule Konstanz**

Prof. Dr. Schick, (07531) 206-657  
cschick@htwg-konstanz.de

### **Hochschule Mannheim**

Prof. Dr. Giehl, (0621) 292-6860  
j.giehl@hs-mannheim.de

### **Hochschule Offenburg**

Prof. Dr. Mackensen, (0781) 205-4770  
elke.mackensen@hs-offenburg.de

### **Hochschule Pforzheim**

Prof. Dr. Kesel, (07231) 28-6567  
frank.kesel@hs-pforzheim.de

### **Hochschule Ravensburg-Weingarten**

Prof. Dr. Siggelkow, (0751) 501-9633  
siggelkow@hs-weingarten.de

### **Hochschule Reutlingen**

Prof. Dr. Wicht, (7121) 271-7090  
bernhard.wicht@reutlingen-university.de

### **Technische Hochschule Ulm**

Prof. Dr. Terzis, (0731) 502-8341  
anestis.terzis@thu.de

[www.mpc-gruppe.de](http://www.mpc-gruppe.de)