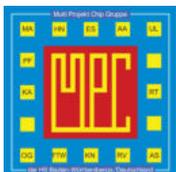


# MPC

MULTI PROJECT CHIP GRUPPE  
BADEN-WÜRTTEMBERG

**Herausgeber:** Technische Hochschule Ulm    **Ausgabe:** 60    **ISSN:** 1868-9221    **Workshop:** Konstanz Juli 2018

- 1 EURORACTICE, the IC Design Infrastructure for European Academia**  
Romano Hoofman, John McLean, Thomas Drischel, IMEC
  
- 7 RF CMOS Technologies and Designs for Mobile Applications**  
Peter Baumgartner, Intel Deutschland GmbH
  
- 13 Radarsignalverarbeitung mit einem Zynq-basierten Embedded System**  
Torsten Bandel, Frank Kesel, Wolf-Henning Rech, Hochschule Pforzheim
  
- 19 Anwendung eines heuristischen Approximationsverfahrens zur Multi-Kriterien Optimierung von Energieclustern**  
Stefan Rädle, Johannes Mast, Joachim Gerlach, Hochschule Albstadt-Sigmaringen
  
- 29 Aufbau und Evaluation eines FPGA-basierten Miniatur-DSP-Systems für professionelle Audioanwendungen**  
Christian Epe, Bernhard M. Rieß, Hochschule Düsseldorf
  
- 47 Implementierung des Secure Hash Algorithmus SHA-256**  
Malek Safieh, Johann-Philipp Thiers, Jürgen Freudenberger, Hochschule Konstanz
  
- 51 CAN FD Signalmanipulationsschaltung auf FPGA-Basis**  
Marcel Maier, Stefan Buntz, Lothar Schmidt, Anestis Terzis, Hochschule Ulm



Cooperating Organisation  
Solid-State Circuit Society Chapter  
IEEE German Section





# Inhaltsverzeichnis

<b>EUROPRACTICE, the IC Design Infrastructure for European Academia</b> .....	<b>1</b>
Romano Hoofman, John McLean, Thomas Drischel, IMEC	
<b>RF CMOS Technologies and Designs for Mobile Applications</b> .....	<b>7</b>
Peter Baumgartner, Intel Deutschland GmbH	
<b>Radarsignalverarbeitung mit einem Zynq-basierten Embedded System</b> .....	<b>13</b>
Torsten Bandel, Frank Kesel, Wolf-Henning Rech, Hochschule Pforzheim	
<b>Anwendung eines heuristischen Approximationsverfahrens zur Multi-Kriterien Optimierung von Energieclustern</b> .....	<b>19</b>
Stefan Rädle, Johannes Mast, Joachim Gerlach, Hochschule Albstadt-Sigmaringen	
<b>Aufbau und Evaluation eines FPGA-basierten Miniatur-DSP-Systems für professionelle Audioanwendungen</b> .....	<b>29</b>
Christian Epe, Bernhard M. Rieß, Hochschule Düsseldorf	
<b>Implementierung des Secure Hash Algorithmus SHA-256</b> .....	<b>47</b>
Malek Safieh, Johann-Philipp Thiers, Jürgen Freudenberger, Hochschule Konstanz	
<b>CAN FD Signalmanipulationsschaltung auf FPGA-Basis</b> .....	<b>51</b>
Marcel Maier, Stefan Buntz, Lothar Schmidt, Anestis Terzis, Hochschule Ulm	

**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):

Joachim Gerlach, Hochschule Albstadt-Sigmaringen, Jakobstraße 6, D-72458 Albstadt-Ebingen

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

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

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

Anestis Terzis, Technische Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

Alle Rechte vorbehalten

Diesen Workshopband und alle bisherigen Bände finden Sie im Internet unter:  
<http://www.mpc-gruppe.de/de/workshopbaende.html>

# EUROPRACTICE, the IC Design Infrastructure for European Academia

Romano Hoofman, John McLean, Thomas Drischel

**Abstract**—The EUROPRACTICE service has enabled more than 600 European universities and research institutes easy and affordable access to advanced microelectronics-based design tools, prototyping and dedicated training for over 25 years. This service has been used by virtually all universities in Europe engaged in microelectronics to facilitate and strengthen their teaching and research activities.

**Index Terms**—ASIC design, IC fabrication, prototyping, design tools, dedicated training, customer support.

## I. INTRODUCTION

The electronic system and semiconductor industry has been a key contributor to European economic growth and prosperity, and an important enabler of European success in the communications, consumer electronics, industrial machinery, and automotive industries. This sector will also provide solutions for the important issues in our society and serve as the foundation for progress in energy conservation, renewable energy, transportation, telecommunication, biotechnology, medical, and many other fields. Semiconductors are critical to European industry and its welfare must be prioritized to keep the European industry competitive.

In order to ensure that Europe remains competitive in new product design (IP, IC, MEMS, heterogeneous systems) and prevent all aspects of design being outsourced in the future to other regions like Asia, it is absolutely necessary that European industry has enough high quality well-trained SoC design engineers.

Consequently, universities must continue to have access to state-of-the-art EDA tools and prototype fabrication for IC, MEMS and heterogeneous technologies.

The European Commission has supported broker services that offer European universities, researchers and small and medium-sized enterprises (SMEs) appropriate access to design tools, advanced technologies, design kits, IP blocks and training to support their education, prototyping and small volume production. These services have been offered by EUROCHIP (1989-1995) and EUROPRACTICE (1995-2018). The

Romano Hoofman, romano.hoofman@imec.be, IMEC, Kapeldreef 75, B3001 Leuven, Belgium,

John McLean, john.mclean@stfc.ac.uk, Microelectronics Support Centre, STFC, Harwell Campus, Oxfordshire OX11 0QX, UK,

Thomas Drischel, thomas.drischel@iis.fraunhofer.de, Fraunhofer IIS, Am Wolfsmantel 33, 91058 Erlangen, Germany



Figure 1. Geographical spread of Europractice members.

European Commission funding contributes to the sustainability and affordability of the service. It should however be noted that although the service receives European funding, more than 75 % of its turnover is from its customers.

Today, after more than 25 years of successful continuous operation, more than 600 European universities and research institutes in 42 countries close to Europe are using the service (as illustrated in Figure 1). In addition over 500 companies and non-European academic institutes have had their IC designs prototyped and produced in small volume.

In summary the main objectives of the EUROPRACTICE service are:

- providing European academic institutions with affordable access to industry-standard and state-of-the-art design tools for their education and research,
- offering affordable access to standard and state-of-the-art IC prototyping, together with providing technology information, cell libraries and design kits for the offered technologies,
- providing training courses and additional assistance in the adoption of advanced IC, MEMS or photonics design flows,

- stimulating academic institutions to work on increasingly advanced designs, or on designs using emerging technologies.

## II. DESIGN TOOL ACCESS AND SUPPORT

The EUROPRACTICE Design Tool portfolio (managed by STFC) includes virtually all the tools available from leading design tool vendors, such as Cadence, Lueda, Phoenix, Mentor and Synopsys. The latest design tool partners are shown in Figure 2. This design tool portfolio is exceptional in terms of its breadth and depth compared to other schemes worldwide.

These advanced design, simulation, verification and implementation tools can be used individually or as part of multi-vendor design flows for a wide range of applications such as programmable devices, ASIC, MEMS, photonic IC (PICs), System on Chip (SoC), System in Package (SiP), Board and Systems level (including embedded software).

Design tools are made available in vendor specific bundles that are cost effective, easy to install and are enhanced annually under maintenance contracts to add new functionality. For example in the case of Cadence, there is an IC bundle, a Systems bundle, plus optional bundles for advanced flows such as 3D-IC, System in Package, advanced mixed-signal library characterisation and sub 20 nm design. As these optional tools and methodologies become more mainstream, they will be moved to the main bundles and replaced by new advanced options. This keeps the cost down for the majority of customers while not limiting others in their chosen speciality.

The overall service is lean and efficient by phasing out design tools which are no longer relevant or in popular demand within the community. Design tools are either licensed on a local server at the customer site or centrally on fault-tolerant servers hosted at STFC. Moreover, in order to ease installation, STFC has developed easy to deploy, preconfigured, known good installations, including environment setup scripts requiring no further local customization.

## III. ACCESS TO AND SUPPORT FOR PROTOTYPING

Over the last decades, multi project wafer (MPW) services have been used for prototyping designs in various microelectronic technologies. Leading foundries (both for customized ASICs, MEMS and photonics) have recognized EUROPRACTICE as the ideal partner to offer these MPW services to smaller industrial users and academia as EUROPRACTICE can offer both access and technical support. Currently, more than 600 designs per year are fabricated via EUROPRACTICE.

The current technology portfolio includes a wide range of technologies from 0.7  $\mu\text{m}$  to 22 nm, from digital logic, RF, mixed-signal to High Voltage and from SiGe, MEMS to Si-photonics. The service is kept to a manageable size by endeavouring to obtain the

maximum functionality from the minimum number of foundries or technologies.

The ASIC technologies are offered from 7 foundries (see Table 1), including the three largest foundries worldwide, i.e. TSMC, GLOBALFOUNDRIES and UMC. Five of these foundries have production sites within Europe (namely OnSemiconductor, ams, GLOBALFOUNDRIES, IHP and X-FAB).

In the More-than-Moore domain, EUROPRACTICE has offered MPW runs in MEMSCAP technologies, which include Poly-Multi-User-MEMS-Processes (MUMPs), SOI-MUMPs and piezo-MUMPs. In addition, Teledyne Dalsa MEMS MIDIS technology has also been offered through CMC in Canada as part of the cooperation between the MPW centers worldwide. In 2018, also X-FAB MEMS technologies have been added to the portfolio.

Besides the traditional MEMS technologies, EUROPRACTICE also offer optical photonics technologies (in particular Si-photonics). The MPW service in these technologies at CEA-Leti and imec was set-up in ePIXfab, but was transferred to EUROPRACTICE in 2015.

In addition, IHP offers an integrated SiGe-Photonics process based on their SG25H4 high frequency technology. EUROPRACTICE also offers photonics packaging together with Tyndall National Institute in Ireland. The photonics ecosystem continues to gather momentum attracting new users (both from academia and from industry) and increasing technical scope of the photonics offering via EUROPRACTICE. Research and development continues to be active amongst telecom, datacom and bio-sensing sectors.

Both imec and Fraunhofer IIS are responsible for the fabrication/prototyping service and have installed a comprehensive support infrastructure with the following tasks:

- i. negotiating with foundries and cell library vendors to introduce new technologies and associated new or updated technical information (documentation, design kits, cell libraries),
- ii. checking new design kits and adapting existing design kits received from the foundry to design tool versions,
- iii. completion of NDAs with academics in order to distribute the design kits and libraries,
- iv. distribution of design kits and libraries under NDA control,
- v. providing technical support to European Academia in the different design tools, cell libraries, technologies specifics, and also checking of their designs before fabrication.

### *Statistics of prototyped designs*

In 2017, a total of 614 designs have been prototyped, a significant increase compared to 2016, when already a record-high number of 575 designs were



Figure 2. The design tool partners of EUROPRACTICE.

Table I  
THE EUROPRACTICE FOUNDRY PARTNERS FOR ASIC PROTOTYPING AND THE OFFERED TECHNOLOGY OPTIONS / NODES.

Foundry	Classification	Technology Nodes
ON Semiconductor	CMOS, HV, EEPROM	0.7 $\mu\text{m}$ – 0.18 $\mu\text{m}$
ams	CMOS, HV, SiGe, OPTO	0.35 $\mu\text{m}$ – 0.18 $\mu\text{m}$
IHP	SiGe, Bipolar, RF, CMOS, Photonics	0.25 $\mu\text{m}$ – 0.13 $\mu\text{m}$
X-Fab	CMOS, HV, EFLASH, OPTO	0.18 $\mu\text{m}$
TSMC	CMOS, HV, RF, LP	0.18 $\mu\text{m}$ – 28 nm
UMC	CMOS, EFLASH, EEPROM, Image Sensor, RF, LP	0.18 $\mu\text{m}$ – 28 nm
GLOBAL FOUNDRIES	CMOS, FDSOI, RF, LP	55 nm – 22 nm

fabricated. 70% of the designs were submitted by European universities and research institutes while the remaining 30% of the designs were from non-European universities (20%) and commercial companies worldwide (10%). Traditionally, there are strong activities in all European countries, especially Germany and Switzerland where 105 and 67 designs respectively were submitted through the EUROPRACTICE service.

EUROPRACTICE offers a good technology and geometry mix. Over the years, a shift towards more advanced technologies has been observed. However, as can be seen in Figure 3, 0.18  $\mu\text{m}$  / 0.15  $\mu\text{m}$  and 0.13  $\mu\text{m}$  / 0.11  $\mu\text{m}$  technologies still represent almost half of the total designs and even the older nodes, such as 0.35  $\mu\text{m}$ , still remain very popular with more than 100 designs. In addition, the number of designs in Silicon Photonics and MEMS has increased considerably compared to 2016, which is thanks to the high number of designs in the imec Si-photonics technologies (namely 59 designs in total). During the coming years, a special effort will be made to promote and boost MEMS fabrication services.

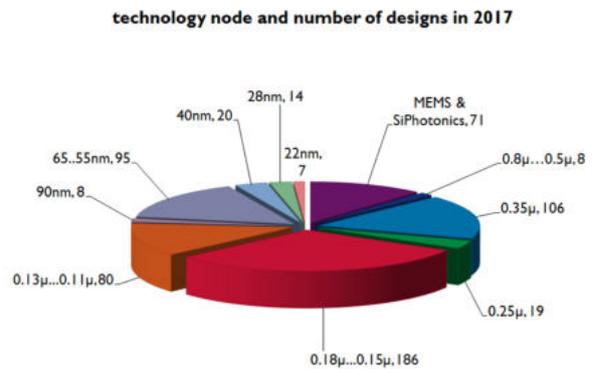


Figure 3. The number of fabricated designs (in 2017) for the different technologies.

#### IV. TRAINING COURSES

The EUROPRACTICE training courses for European universities and Research Institutes are primarily aimed at academic staff and PhD students. Unlike training courses which address single topics or individual design tools, the EUROPRACTICE training courses address a design flow which makes these training courses an efficient way to acquire new knowledge



Figure 4. Picture from a typical EUROPRACTICE training course at STFC, Didcot (UK).

and ideally suited to new PhD students and junior engineers with a need to quickly become productive with a design flow. Since the courses are based on the EUROPRACTICE design tools, PDKs and Technologies, participants will be able to directly apply the techniques learnt on the training course when they return back to their own organisation and make full use of the EUROPRACTICE services/infrastructure in their innovation, research and training.

Courses include a strong element of practical sessions where participants are able to extensively practice the concepts described in lectures and have access to experts who are able to answer questions. Where it is known that a design flow is well supported by multiple vendors and/or processes, then multiple course variants are offered that reflect the design tool / process installed base.

Training courses follow a “train-the-trainer” philosophy, so that participants can convey the knowledge acquired to colleagues within their own organisation. Training course participants are provided with course notes (manuals) which they can then keep and refer to at a late date when applying the techniques to their own work.

The training courses have been primarily held in the dedicated training facilities available at the sites of the EUROPRACTICE partners (i.e. imec, STFC and Fraunhofer), where the various design flows and technologies are well established and supported by appropriate training infrastructures.

## V. STIMULATION ACTIVITIES

### A. Lowering entry-barrier for prototyping by the *mini@sic* Program

Although MPW runs are a good solution to provide lower prototyping costs (typically 10% of a full mask prototype run), there is a minimum fabrication cost set by the foundry corresponding to a minimum design area to be fabricated (standard foundry business model).

Since many designs can be much smaller than the minimum charged design size on regular MPW runs, the concept of *mini@sic* runs was introduced in 2003 – which was a solution to offer a further reduced fabrication cost and consequently lowered the entry-barrier for prototyping in advanced technologies. The *mini@sic* principle is based on the methodology that 2 to 3 times per year on selected MPW runs a minimum area MPW block size will be bought by EUROPRACTICE, which is resold in smaller cheaper sub-blocks. This *mini@sic* MPW principle was extended over the years towards 90 nm, 65 nm, 40 nm, 28 nm CMOS, 22 nm FDSOI and even Si-photonics to accommodate affordable prototype fabrication of small designs for the European universities and research institutes. Recently, EUROPRACTICE has further lowered the entry barrier for using advanced technologies by introducing the concept of a microblock for the TSMC 28 nm, where a minimum area of 1 mm<sup>2</sup> is made available for only 10 k€.

### B. First User Stimulation Action Program

As part of the current EC-funded project (EUROPRACTICE2016), a First User Stimulation Program was launched to encourage EUROPRACTICE university institutions to fabricate their first IC or first IC in an advanced or in a More-than-Moore technology. Successful applications were awarded a free design fabrication or fabrication at a highly-reduced-cost.

Six special stimulation contests were organized for ASICs, MEMS and Si-photonics. In total, 96 applications were submitted to these contests by 73 universities from 23 countries. The applications were judged by 5 independent expert panels and 50 designs were selected for fabrication. As can be seen from Figure 5, most submitted and awarded applications originated from Italy, UK and Germany. In Germany, applications from the following universities were awarded:

- Eberhard Karls University of Tübingen
- University of Kassel
- South Westphalia University of Applied Sciences
- Dortmund University of Applied Sciences & Arts
- Ruhr-University Bochum
- Braunschweig University of Technology
- Paderborn University

## VI. EVOLUTION OF THE SERVICE PORTFOLIO

To this day, the integrated flow from design tools through design kits to IC fabrication is the core of the EUROPRACTICE one-stop service. However, it is important to follow industry trends and listen to customer needs. Therefore, the EUROPRACTICE design tool and fabrication portfolio has evolved and been significantly enhanced over time to remain relevant.

In 2018, EUROPRACTICE has initiated and/or added the following to its portfolio:

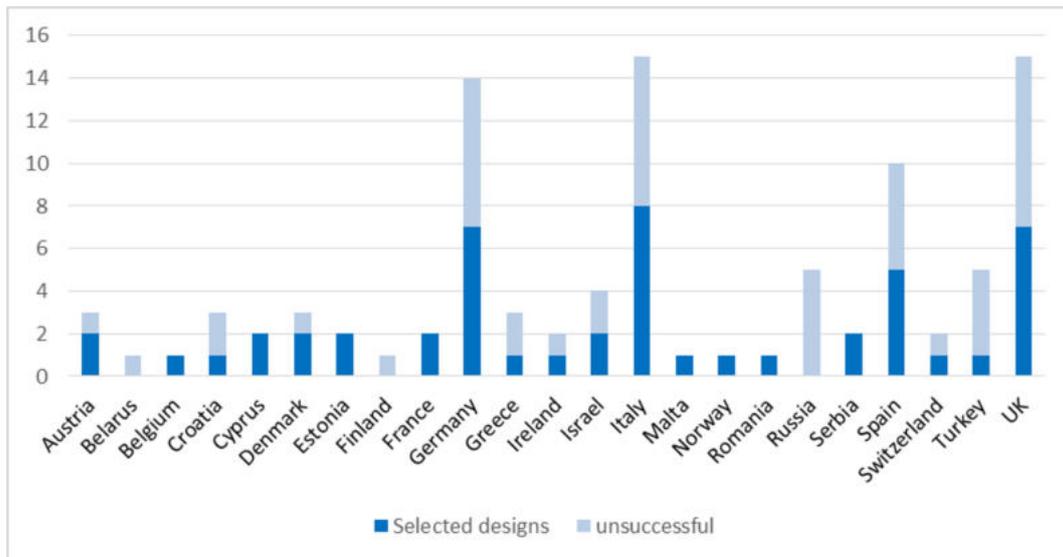


Figure 5. The total number of received applications in the First User Stimulation Program from various European countries, where the dark blue bars represent the awarded designs and the light blue unsuccessful applications.

- MEMS technologies from X-FAB
- eMemory design competition, where 10 selected designs will receive free memory IP in different foundry technologies,
- Ultra-small design blocks in 28 nm TSMC technology (microblock concept).

In order to keep the entire service affordable and accessible a new project proposal has been submitted to the European Commission, which will guarantee funding until the end of 2021. This new project proposal builds upon the well-established, widely-used and successful Europractice service and extends towards SMEs and System Integration. This will be achieved by enhancing the services provided to European academia, extending the services to SMEs, and encouraging new users from non-traditional sectors. The new project will also add new technologies and design tools to serve new markets, diversify the service towards smart system integration, and further support the training of future generations of engineers that will be required for the growing digital economy in Europe.

## VII. CONCLUSIONS

European academic institutes enjoy affordable, one-stop access to the best design tools, design flows, and processes normally only available to the largest multinational corporations. European fabless SMEs have easy access to prototype and small volume fabrication, packaging and test. This common European microelectronics-based design infrastructure facilitates cooperation, underpins research, teaching and helps innovation. Graduates entering industry from European academic institutes are more appropriately trained and more able to meet future industry challenges.

As the requirements of European academic institutes have evolved, so have the offerings of the service.

The EUROPRACTICE user community of today is far broader both geographically and thematically than could have been envisaged at the beginning of EUROPRACTICE.

## ACKNOWLEDGEMENTS

The European Commission is acknowledged for its continuous support under several Framework Programs FP2, FP3, FP4, FP5, FP6, FP7 and H2020. The current project (EUROPRACTICE2016) has received funding under grant agreement #688226, which is exclusively used to support European universities and research laboratories.



**Romano Hoofman** received the M.Sc. degree from Wageningen University (NL) in 1995 and the Ph.D. degree from Delft University of Technology (NL) in 2000. He started his career in industry at Philips Research and NXP Semiconductors, where he covered a variety of R&D topics, ranging from CMOS integration to sensors. In 2016, he took on the position as Program Director at imec, responsible for project and innovation management of EURO-PRACTICE and related services.



**John McLean** has a B.Sc. Honours Degree in Electrical and Electronic Engineering and a PhD for Research in Computer Aided Design of Digital Systems. He joined the STFC Rutherford Appleton Laboratory as part of the Computer Aided Design Group and subsequently was Group Leader of both the Microelectronics Design and Systems Design Groups. In 1989, he became a member of the Executive Board of the EUROCHIP Service Organization and has managed STFC's EUROPRACTICE design tool activity since 1995.



**Thomas Drischel** received the degree of Diplom Ingenieur from the Fachhochschule Georg Simon Ohm in Nürnberg in 1998. He has been working for Fraunhofer IIS in the EUROPRACTICE project as a support engineer for ams and IHP technologies. In this function he was responsible for the coordination of the multi-project and small volume wafer fabrication. Since 2013 he is leader of the Virtual ASIC Foundry group and responsible for the EUROPRACTICE IC fabrication service.

# RF CMOS Technologies and Designs for Mobile Applications

Peter Baumgartner

**Abstract**—Digital technology scaling results in improving high frequency capabilities for radio frequency (RF) and millimeter wave (mmW) applications. This paper discusses performance trends till 28nm technology and investigates important parameters for future RFCMOS technologies.

**Index Terms**—CMOS, scaling, high frequency, RFCMOS.

## I. INTRODUCTION

Compared to digital circuits, analog, RF and mmWave circuits require additional passive devices, like inductors, varactors and transmission lines. As shown in Figure 1, there are also a lot of additional parameters, related to noise, quality factors, parasitics etc. to be considered.

From a pure technology view, all necessary devices are already available in plain digital CMOS technologies. For example inductors and transformers can be created with available metal lines, transistors can be used as varactors. For RF and mmWave designs it is important, that these devices are enabled in a design environment (PDK, process development kit) with accurate modeling.

The main difference between a digital and an RF CMOS technology is therefore the enhanced and accurate device modeling and PDK enablement. This additional effort and the need for stable process parameters results in a later start (typically a year) of RF offerings and RF circuit development.

## II. CMOS SCALING FOR RF CIRCUITS

The cross section of a CMOS technology with the scaling from one technology to the next generation is shown schematically in Figure 2. The Figure also shows, where the different devices are realized in the CMOS technology.

The technology can be divided into front end of line (FEOL), which contains all the silicon related processing, including the contacts. In this part the transistors, varactors, diodes and resistors are created. During development of the next technology node, there is strong focus to increase the transistor density and to

	Digital	Analog	RF	mmWave
		Noise Flicker, RF Gate Resistance, Sub. Netw.		
		Flicker Noise		
		Q-Factors, Resonance, dielec. Absorption		
			Q, Tuning Range	
			Q, Resonance, Coupling	
			Distributed, TML Devices	

Figure 1. Devices used for digital, analog, RF and mmWave circuits and important parameters.

reduce the dimensions. The process part, which creates the active transistors is also called front end of line (FEOL).

The back end of line (BEOL) of a CMOS technology contains all the wiring structures with metal levels and vias. This back end of line can be further divided into a lower part, which scales together with the front end of line, since all the front end devices need to be connected with these layers. This part is typically used to create linear capacitors, since the shrinking dimensions result in increasing capacitor densities.

The upper part of the BEOL is typically not scaling between different technology nodes and is providing thicker metal layers with high conductivity for low ohmic power routing. For RF circuits, it is very common to have a very thick upper copper metallization (in the range of  $3\ \mu\text{m}$ ) which enhances the  $Q$  factor of inductors, transformers and transmission lines.

From this picture it is already understandable, that active devices and capacitors benefit from technology scaling. Inductive devices have much less scaling potential and circuit designers try to minimize the usage of these devices.

Peter Baumgartner, peter.baumgartner@intel.com, Intel Deutschland GmbH, München.

Intel is a trademark of Intel Corporation or its subsidiaries in the U.S. and/or other countries

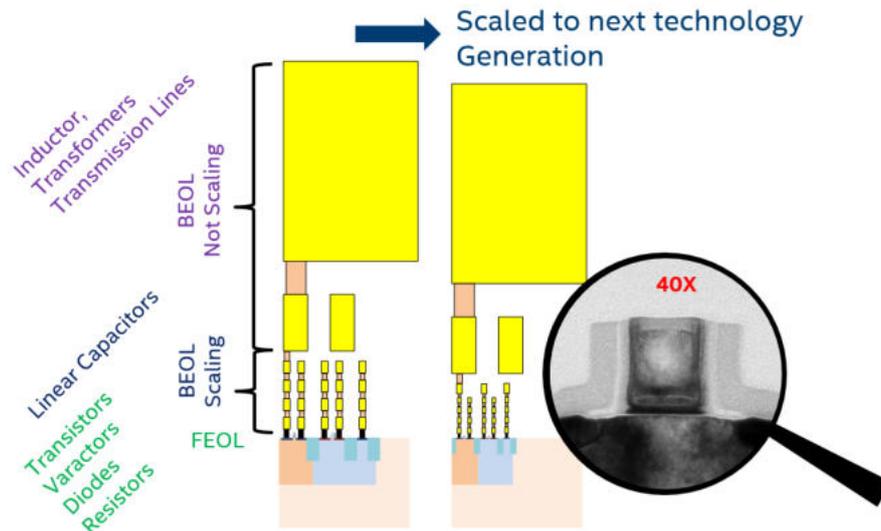


Figure 2. Schematic cross section of a CMOS technology, placement of the different devices and scaling to next technology node.

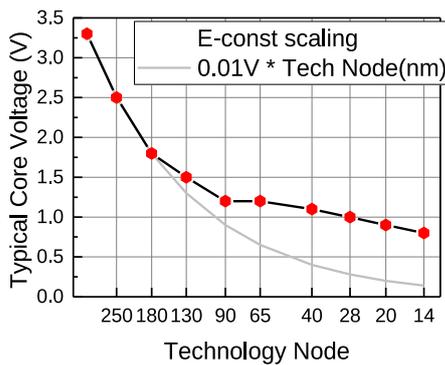


Figure 3. Real voltage scaling over different technology nodes compared with an ideal voltage scaling (constant field scaling).

Figure 3 compares the typical core supply voltage of technology nodes ranging from 250 nm down to 14 nm with an ideal theoretical voltage scaling, derived from a constant electric field boundary conditions. The ideal voltage scaling was valid down to 180 nm technology node (with 1.8 V supply voltage). This ideal geometry and voltage scaling resulted in huge performance and power improvements.

After that (130 nm and below) the voltages reduction is limited by threshold voltages which are related to leakage requirements. To continue the device and performance scaling, innovation beyond geometry scaling was needed. Some examples for this are strain engineering to boost mobility in the transistor channel, high- $k$  dielectrics with metal gate to reduce gate leakage (larger physical thickness at same gate capacitance) and FinFET or Silicon on Insulator (SOI) technologies to improve the electrostatic control of the gate electrode and allow additional device scaling. Bulk technologies are available down to 20 nm node,

beyond that FinFET or SOI technologies are needed [1]–[4].

The technology scaling results in a strong improvement of RF performance, as shown in Figure 4. The left plot shows measured transit frequencies ( $f_T$ ) and maximum oscillation frequencies ( $f_{MAX}$ ) from 330 nm technology down to 28 nm node. The results are collected from test chips and S-parameter characterization of the Intel RF modeling team.

The transit frequency  $f_T$ , which is the frequency where the current gain of the transistor is reduced to one, scales very systematically with technology nodes. This figure of merit (FOM) benefits strongly from  $g_m$  improvements and capacity reduction. The  $f_{MAX}$  figure of merit, which is related to a unity of Masons gain [1], depends on additional parasitics like gate resistance and Miller capacitance and has therefore a stronger layout dependency. The plot highlights values from device layouts which were optimized for  $f_{MAX}$  (not available for every technology). The  $f_{MAX}$  also improves strongly with technology, but there is a saturation from the highly optimized 65 nm to 28 nm visible.

The right plot in Figure 4 shows the correlation of  $f_T$  with the supply voltage of the corresponding devices. As expected, the increased performance of the advanced technologies comes with a reduced voltage capability. But the voltage dependency is much weaker, than for example Johnsons figure of merit (which is related to break down fields) for bipolar transistors. The supply voltage in CMOS is more limited by reliability mechanisms like hot carrier degradation, time dependent dielectric breakdown and not by break down fields. For low voltages these degradation mechanisms are less critical.

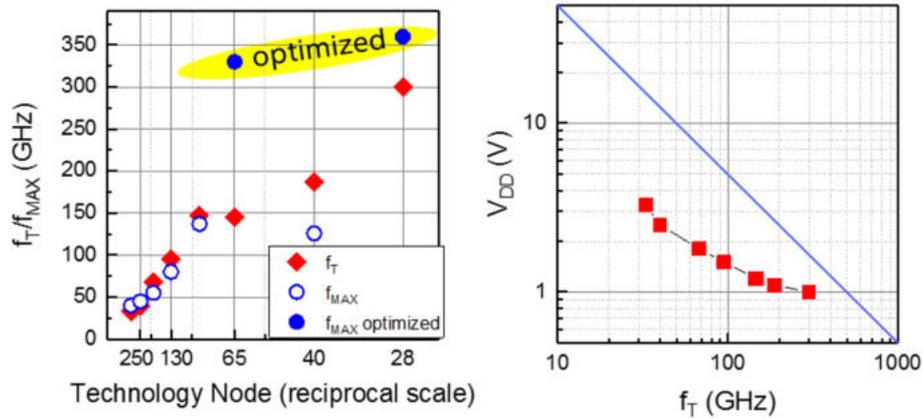


Figure 4. Trend for transit frequency  $f_T$  and maximum oscillation frequency  $f_{MAX}$  over technology nodes (left). Voltage vs.  $f_T$  scaling (right).

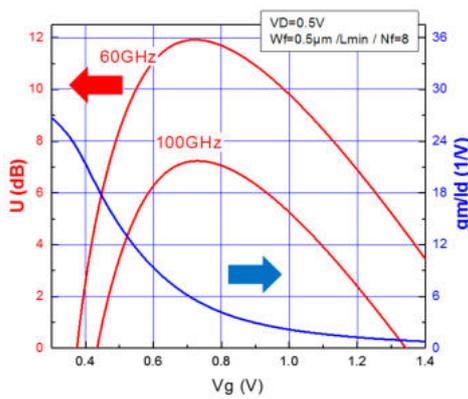


Figure 5. Masons gain  $U$  for two frequencies (60 GHz and 100 GHz) and transconductance divided by bias current ( $g_m/I_d$ ) as a function of gate bias.

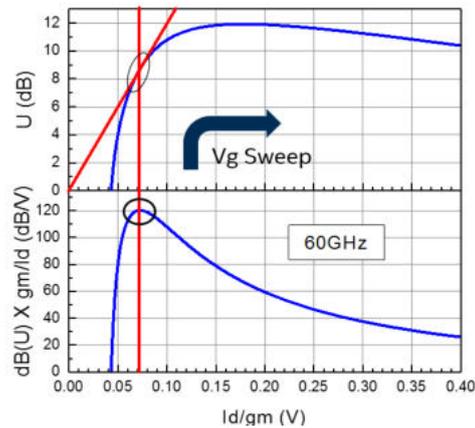


Figure 6. Method to determine power efficient bias condition by plotting gain ( $U$ ) vs. bias current efficiency ( $I_d/g_m$ ). This leads to maximum of  $dB(U) \cdot g_m/I_d$  as FOM for power efficient gain.

### III. CORRELATION OF DIGITAL PERFORMANCE AND RF PERFORMANCE

To get more insight how digital driven scaling interacts with RF figures of merit, a model based correlation study was performed. Device parameters like gate length, width, gate oxide thickness, gate resistance where changed to study sensitivity of digital performance vs. RF performance.

For the digital performance the frequency of a schematic ring oscillator with fan out 2 was used. To be able to study single devices (e.g. NMOS without the complementary PMOS), the complementary device was constructed by mirroring all currents with current controlled current sources.

The ring oscillator frequency should be correlated to a RF/mmWave figure of merit which reflects the performance and also the energy efficiency which is very important for mobile applications.

At mmWave frequencies the main problem is to create enough gain. The gain decreases with frequency, as clearly visible in Figure 5 showing masons gain  $U$

as a function of gate voltage for different frequencies [5].

An important transistor parameter for energy efficiency is the ratio of transconductance and bias current  $g_m/I_d$ . The  $g_m/I_d$  curve in Figure 5 shows the trade of between gain and RF performance and energy efficiency. Higher gate voltages, which increase the speed and the gain of the transistor result in reduced  $g_m/I_d$  values.

For a circuit relevant figure of merit it is therefore necessary to define a relevant bias condition which is relevant for a high efficiency design. The maximum gain condition (also corresponding to  $f_{MAX}$ ) corresponds to a very low efficiency (low  $g_m/I_d$ ) bias condition.

It is necessary to find a good trade of between gain (to minimize the number of needed stages) and power per stage. Figure 6 shows a method to define the best bias condition for such a trade of. The upper curve plots  $U$  vs.  $I_d/g_m$ . The optimum bias condition is defined by the tangent of the line through origin

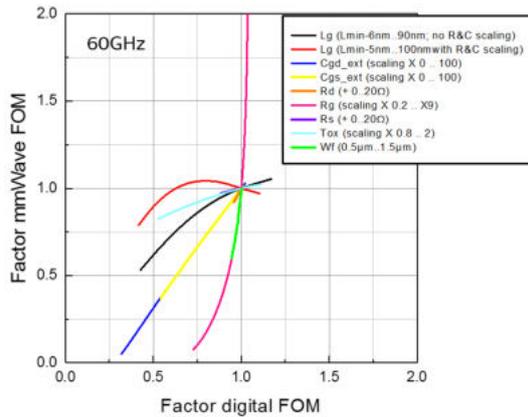


Figure 7. Spice simulated correlation of digital figure of merit (ring oscillator frequency) vs. mmWave FOM ( $\text{dB}(U) \cdot g_m/I_d$ ) for different process variations.

(red) and the  $\text{dB}(U)$  vs.  $I_d/g_m$  curve (blue). Increasing the gate voltage beyond this point, results in a small increase of  $U$ , but a large penalty in  $g_m/I_d$ . Decreasing the voltage results in a strong reduction of  $U$  with only slight  $g_m/I_d$  improvement.

This optimum biasing condition is frequency dependent, at higher frequency it shifts to larger gate voltages. For the optimum biasing condition the value of  $\text{dB}(U) \cdot g_m/I_d$  is maximized, as the lower Plot in Figure 6 demonstrates. This maximum value of  $\text{dB}(U) \cdot g_m/I_d$  is a good figure of merit for gain and energy efficiency at a realistic and optimized device bias point.

To study the sensitivity of the described digital and mmWave Figure of Merit on process parameters, simulations with modified 28 nm RF spice models were performed. Starting point are scalable PSP RF compact models. The investigated parameters include gate length, gate width, gate resistance, gate oxide thickness and additional parasitic capacitances and resistances.

Figure 7 shows the correlation of the digital FOM (ring oscillator frequency) with the mmWave power efficiency FOM ( $\text{dB}(U) \cdot g_m/I_d$ ). This correlation is important for future RF technologies. Parameters which show a good correlation between analog and RF/mmWave benefit automatically from digital scaling and can be expected to further improve in future.

Parameters which show high sensitivity for RF but low sensitivity for digital, need special attention from the RF community and can lead to RF specific technology optimization.

Parasitic capacitances, series resistances and gate oxide thickness show a good correlation between digital and RF and need therefore no additional RF attention beyond digital.

More complicated are gate length, gate width and gate resistance. An additional gate resistance (red line) has huge impact on the mmWave FOM, since the

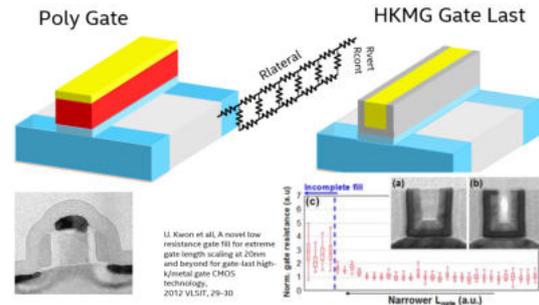


Figure 8. CMOS gate construction for Poly Gate and Metal Gate and visualization of gate resistance.

gate signal is already attenuated before reaching the transistor. The impact on the digital FOM is much lower. Since capacitance have higher impact on digital performance than gate resistance, a digital optimized technology will favor process tradeoffs which reduce capacitance but increase gate resistance (e.g. using thinner gate conductor).

The correlation for the transistor width is identical to the gate resistance variation, since increasing gate finger width leads to increasing gate resistance.

For the gate length scaling two different approaches were chosen. The scalable RF model scales parasitic capacitances and resistances with the device geometry. The black Lg line shows a gate length scaling, where all external parasitic resistances and capacitances are kept constant. Only the intrinsic channel length is varied. A good correlation of digital and mmWave is observed here. However the benefit for mmWave is reduced compared to digital, since shorter gate length also decrease analog parameters like  $g_m/I_d$  or  $g_m/g_{ds}$ .

With scaled parasitics (red Lg curve) we observe an optimum value for mmWave, which is in this case achieved for longer gate length. The gate resistance has a very strong length dependency and this leads here to improved mmWave performance for slightly increased gate length. Further increase of gate length leads to further reduced speed,  $f_T$  and  $f_{MAX}$  values and therefore a reduction of the FOM.

Figure 8 shows schematically the gate construction for poly gate (e.g. used down to 65 nm and 45 nm technologies) and metal gate technologies (used currently for advanced technologies). For poly gate technologies the gate resistance is reduced by a silicide. The contact resistance of the silicide to the doped poly is very important. The metal gates consist of a low ohmic core material (e.g. aluminium) and liner and workfunction materials which are important to adjust the threshold voltage of the device. This are typically materials with higher resistance. With shrinking dimensions it is a challenge to reduce the gate resistance.



#### IV. SUMMARY

Gate resistance is a major challenge for next generation RF and mmWave CMOS technologies and improvements vs. the baseline digital technology will be needed. Without enhancements the best performance moves from shortest gate length to slightly increased gate length and therefore the benefit of scaling can't be utilized anymore. It is beneficial for future process development, that most technology parameters show a strong correlation between digital circuits and RF and mmWave circuits.

#### ACKNOWLEDGEMENT

The author thanks the colleagues from the Intel RF modeling department.

#### REFERENCES

- [1] I. Post et al, "A 65 nm CMOS SOC Technology Featuring Strained Silicon Transistor for RF Applications", in *emph2006 International Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2006.
- [2] L. Poulain, "Small signal and HF noise performance of 45 nm CMOS technology in mmWave range", in *2011 IEEE Radio Frequency Integrated Circuits Symposium*, Baltimore, MD, USA, June 2011.
- [3] P. Vandervoorn et al, "A 32nm low power RF CMOS SOC technology featuring high-k/metal gate", in *2010 Symposium on VLSI Technology*, Honolulu, Hawaii, June 2010.
- [4] C.-H. Jan et al, "A 14 nm SoC platform technology featuring 2nd generation Tri-Gate transistors, 70 nm gate pitch, 52 nm metal pitch, and  $0.0499 \mu\text{m}^2$  SRAM cells, optimized for low power, high performance and high density SoC products", in *2015 Symposium on VLSI Technology (VLSI Technology)*, Kyoto, Japan, June 2015.
- [5] S. J. Mason, "Power Gain in Feedback Amplifiers", *IRE Trans. on Circuit Theory*, vol. CT-1, pp. 20-25, June 1954.



**Peter Baumgartner** holds a M.Sc (Diplom) and Ph.D. (Dr. rer nat) in Physics from TU Munich (1993) and Walther Schottky Institut (1997). He started 1997 at Siemens in technology development for embedded flash technologies, focusing on transistor development and reliability engineering. He is an Intel Principal Engineer for RF and Analog devices, working on device and technology topics for RF, mmWave and power management applications.



# Radarsignalverarbeitung mit einem Zynq-basierten Embedded System

Torsten Bandel, Frank Kesel, Wolf-Henning Rech

**Zusammenfassung**—Im ZAFH „MikroSens“ forschen die Hochschulen Ulm, Heilbronn und Pforzheim sowie die Universität Ulm an innovativer Radarsensorik für industrielle Anwendungen. Die Aufgabe der Hochschule Pforzheim in diesem Verbundprojekt ist es, ein so genanntes „Sensor-Backend“ zu entwickeln, welches zur Signalverarbeitung der von einem 160-GHz Radarsensor gelieferten Signale verwendet wird. Für die Signalverarbeitung wird ein Zynq Z-7030 FPGA von Xilinx eingesetzt.

In diesem Beitrag wird insbesondere die Signalverarbeitung im Zynq beschrieben. Das Kernstück der Signalverarbeitung ist eine zweidimensionale FFT. Hierfür wurden mit Hilfe der „High-Level-Synthese“ aus entsprechenden C-Codes zwei IP-Cores synthetisiert. Diese sind über ein AXI-Bussystem an den DRAM-Hauptspeicher des Zynq-Systems angebunden. Das Ergebnis der 2D-FFT, die sogenannte Range-Doppler-Map, kann dann über einen Web-Server visualisiert und gleichzeitig via UDP auf einen Host-Rechner in Matlab exportiert werden.

## I. EINLEITUNG

Das ZAFH MikroSens beschäftigt sich damit, innovative Millimeterwellen-Sensorik für unterschiedliche industrielle Anwendungsfelder zugänglich zu machen. Dies schafft ein breites Aufgabenspektrum mit Themen aus Bereichen wie zum Beispiel Mediensensorik, Straßenzustandsüberwachung oder Gewässersensorik, wo Grundlagenforschung betrieben und Machbarkeitsstudien erarbeitet werden. Ein weiteres Ziel von MikroSens ist es, einen Radarsensor zu entwickeln, der durch einfache Konfiguration für diese Vielzahl von Anwendungen als Demonstrator eingesetzt werden kann. Mit dem Radar-Sensor sollen prinzipiell Objekte erkannt und deren Geschwindigkeit ermittelt werden können. Auf Grund der Vielzahl an unterschiedlichen Anwendungen, die abgedeckt werden sollen, ergeben sich folgende Anforderungen an das Radar-Sensorsystem:

- 100 MS/s Abtastrate
- Nyquist-Reichweite bis 30 m, Objektauflösung bis 20 mm
- Nyquist-Geschwindigkeit bis  $\pm 8$  m/s, Geschwindigkeitsauflösung bis 3 cm/s
- 10 Hz Messrate

Torsten Bandel, torsten.bandel@hs-pforzheim.de,  
 Frank Kesel, frank.kesel@hs-pforzheim.de,  
 Wolf-Henning Rech, wolf-henning.rech@hs-pforzheim.de,  
 Hochschule Pforzheim, D-75175 Pforzheim, Germany

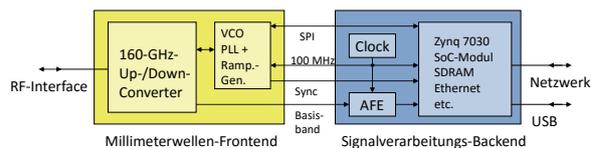


Abbildung 1. Blockschaltbild der Sensor-Hardware.

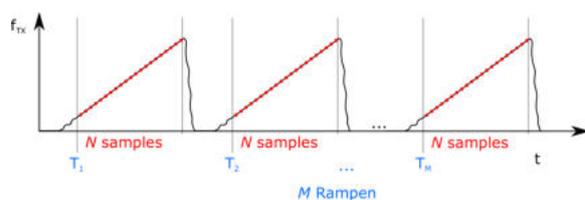


Abbildung 2. Frequenz über Zeit-Darstellung eines chirp-sequence-Radars.

Um diese Anforderungen zu erfüllen, wird sowohl ein HF-Frontend mit ausreichend Bandbreite und Dynamik, als auch ein performantes Signalverarbeitungs-Backend mit genügend Rechenleistung benötigt. Das HF-Frontend kommt von der Universität Ulm und beinhaltet einen selbst entwickelten 160-GHz-Up-Down-Converter, sowie die erforderliche HF-Signalgenerierung durch einen Breitband-VCO und PLL. Das Signalverarbeitungs-Back-End wurde an der Hochschule Pforzheim entwickelt und beinhaltet die Taktverteilung, analoge NF-Signalverarbeitung sowie die digitale Signalverarbeitung und die Steuerung des Sensorsystems, wie in Abbildung 1 gezeigt.

Der Sensor wurde als FMCW-Radar (*Frequency Modulated Continuous Wave Radar*) nach dem „chirp-sequence“-Verfahren aufgebaut. Hierbei werden nacheinander mehrere linear frequenzmodulierte „Rampen“ ausgesendet, siehe Abbildung 2. Jede Rampe wird mit  $N$  Samples abgetastet und eine Messung beinhaltet  $M$  Rampen. Objekte können durch eine Laufzeitmessung innerhalb einer Rampe erkannt werden. Hierbei wird die empfangene Frequenz einer von einem Objekt reflektierten Welle mit der momentanen Frequenz der Senderampe verglichen. Die Rampen müssen steil genug sein, sodass die Dopplerverschiebung innerhalb einer Rampe vernachlässigt werden kann. Die Geschwindigkeit wird nun über die Dopplerverschiebung erkannter Objekte über mehrere Rampen hinweg ermittelt. Diese Berechnungen erfolgen mittels einer 2D-FFT über die Datenmatrix (Gleichung (1)) einer

gesamten Rampensequenz der Größe  $M \times N$ . Eine detaillierte Beschreibung der Signalverarbeitung von FMCW-Radar-Systemen ist in [5] zu finden.

$$A_{i,j} = \begin{pmatrix} S_{1,1} & S_{1,2} & \dots & S_{1,N} \\ S_{2,1} & S_{2,2} & \dots & S_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ S_{M,1} & S_{M,2} & \dots & S_{M,N} \end{pmatrix} \quad (1)$$

Um die Signalverarbeitung für das Radar-Sensorsystem zu implementieren, wurde ein Zynq Z-7030 FPGA von Xilinx ausgewählt. Der Zynq besteht aus zwei Teilen, dem „Processing System“ (PS) und der „Programmierbaren Logik“ (PL). Das PS bietet ein vollwertiges Cortex-A9 Dual-Core-Mikroprozessorsystem der Firma ARM mit gängigen Peripherieeinheiten, wie zum Beispiel SPI, USB, Ethernet. Die PL ist ein vollwertiges Kintex-7-FPGA. Das AMBA-AXI-Bussystem verbindet hierbei interne Komponenten wie CPU, FPGA und DRAM-Controller. Somit können zum Beispiel über die AXI-GP-Ports IP-Cores in der PL zur Laufzeit per Software konfiguriert werden oder die IP-Cores können mit Hilfe der AXI-HP-Ports (High Performance) direkt auf Daten im Hauptspeicher zugreifen. Eine vollständige Darstellung des Zynq-Systems kann z.B. [1] entnommen werden.

## II. SYSTEMKONZEPT

Das Sensorsystem soll ohne zusätzlich notwendige Computer über einfache Schnittstellen bedienbar sein. Daher wird ein Linux-Betriebssystem auf dem Zynq eingesetzt um den Zugriff auf standardisierte Schnittstellen, wie UDP oder Web-Sockets, zu vereinfachen. Für die Konfiguration des Sensors und die Visualisierung der Ergebnisdaten wird unter Linux daher ein Webserver ausgeführt, so dass per Internet-Browser auf das Sensorsystem zugegriffen werden kann. Abbildung 3 zeigt das implementierte Signalverarbeitungssystem auf dem Zynq. Hierbei sind PS und PL getrennt dargestellt und über das SDRAM als Daten-Schnittstelle miteinander verbunden. Verwendet wird hier ein Linux-System mit einem auf das Zynq-Prozessorsystem abgestimmten Kernel. Zusätzliche Kernaltreiber ermöglichen den Zugriff auf die IP-Cores in der PL aus dem Anwendungsprogramm heraus. Die IP-Cores mit DMA (Direct Access Memory) bekommen einen festen Bereich im DRAM-Hauptspeicher zugeteilt, welcher für Linux nicht verwendet wird, um Konflikte bei Speicherzugriffen zwischen IP-Cores und Linux-Betriebssystem zu vermeiden.

Ein Anwendungsprogramm initialisiert die IP-Cores und steuert den Messablauf, wobei die Signalverarbeitung im Wesentlichen in den IP-Cores der PL ausgeführt wird. Die ADC-Daten werden in der PL über das serielle Interface JESD204b eingelesen. Xilinx bietet

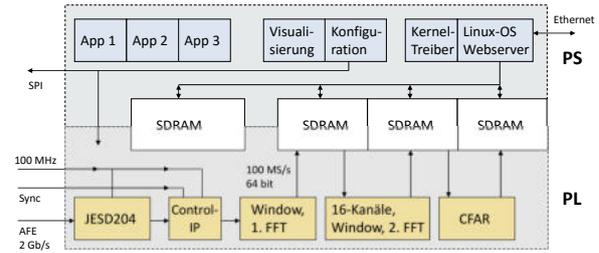


Abbildung 3. Systemimplementierung mit dem Zynq-FPGA. Die Blöcke im oberen PS-Teil sind entsprechende Anwendungen, die in Software implementiert wurden. Der untere PL-Teil zeigt die Blöcke, welche als IP-Cores in Hardware implementiert wurden. Der Block SDRAM bezeichnet den DDR3-Hauptspeicher des Zynq-Systems über welchen der Datenaustausch zwischen Software und IP-Cores stattfindet.

hierfür einen IP-Core [4], der die ankommenden Daten decodiert und parallelisiert. Der JESD-IP-Core wird zusammen mit dem ADC initialisiert und streamt Daten bis er per Software in den „Sleep-Modus“ gesetzt wird. Der darauffolgende „Control-IP-Core“ steuert die in Abschnitt I erwähnte Rampensequenz und erkennt, wann gültige Daten anliegen. Diese werden dann passend formatiert, so dass der IP-Core diese Daten für die erste FFT einlesen kann.

Das JESD-Interface, sowie der „Control-IP-Core“ werden von einem externen Quarz-Oszillator getaktet. Dies ist notwendig, da diese IP-Cores mit der PLL zur Erzeugung der Rampen und dem ADC synchron betrieben werden müssen. Der Control-IP-Core wird von der Software zyklisch über den Zustand der Rampensequenz abgefragt und teilt der Software mit, dass das Einlesen der Daten abgeschlossen ist. Zwischen Control-IP-Core und der ersten FFT-Stufe befindet sich ein Synchronisations-FIFO, so dass ab der ersten FFT-Stufe ein interner PL-Takt verwendet werden kann. Dies erlaubt eine schnellere Taktung des Signalverarbeitungsteils und dient zusätzlich als Puffer um Datenverluste zu vermeiden.

Die 2D-FFT wird nun mit Hilfe von zwei FFT-IP-Cores berechnet. Der erste FFT-IP-Core ist eine Pipeline-Streaming-FFT mit AXI-Stream-Interface am Eingang und AXI-Master-Interface am Ausgang. Somit können die ankommenden Daten einfach eingelesen und die Ergebnisdaten direkt in den Speicher geschrieben werden. Die FFT erhält vom Control-IP-Core 16-Bit Festkomma Eingangsdaten im Zweierkomplement-Format, wobei die Bitbreite des ADC 14 Bit beträgt. Der Imaginärteil der komplexen Eingangsdaten wird innerhalb des IP-Cores konstant auf Null gesetzt. Um bei der Berechnung der Festkomma-FFT keine Genauigkeit zu verlieren wird die erste FFT als unskalierte Radix-2-FFT implementiert. Der Anstieg der Bitbreite am Ausgang ( $w_{Bit,out}$ ) ist in Gleichung (2) beschrieben und deckt den Worst-Case eines Überlaufs ab [3].

$$w_{Bit,out} = w_{Bit,in} + \log_2(FFT \ Length) + 1 \quad (2)$$

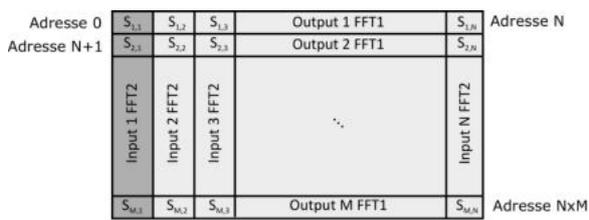


Abbildung 4. Ergebnis-Speichermatrix nach der ersten FFT mit relativen Wortadressen.

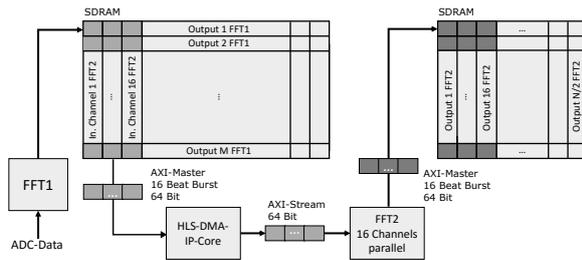


Abbildung 5. Darstellung der Speicherzugriffe bei der zweistufigen FFT.

Im Beispiel von 4096 Punkten pro Rampe beträgt die Ausgangsdatenbreite 27 Bit, was dann auf 32 Bit erweitert wird. Somit ist ein komplexes Sample nach der ersten FFT insgesamt 64 Bit breit. Bei einer Messkonfiguration von 64 Rampen mit je 4096 Punkten ergibt dies eine Speichergröße der Ergebnismatrix von 2 MB. Da das Zynq-FPGA in der PL nur über rund 1,2 MB an integriertem „Block-RAM“ verfügt, ist es nicht möglich, die Ergebnisdaten in der PL zwischen zu speichern. Daher werden die Ergebnisdaten über das AXI-Master-Interface des FFT-IP-Cores linear in den DRAM Hauptspeicher des Zynq geschrieben. Abbildung 4 verdeutlicht die Speicheranordnung der Ergebnisdaten nach der ersten FFT mit relativen Wortadressen, wobei ein Wort aus acht Byte besteht.

Diese Anordnung ist ungünstig für die zweite FFT, da diese, wie in Abbildung 4 dargestellt, über die Spalten der Ergebnismatrix berechnet werden muss. Somit entstehen beim Einlesen eines Eingangsvektors 32 kB große Sprünge (4096 Samples je 8 Byte) zwischen zwei Samples. Dadurch werden die Speicherzugriffe auf das SDRAM sehr langsam, da jedes Wort mit einem Einzeltransfer geholt werden muss und keine „Burst-Transfers“ für den Eingangsvektor möglich sind. Um dieses Problem zu umgehen, wird die zweite FFT mit 16 Kanälen implementiert, um einerseits auch den Durchsatz der FFT zu erhöhen und andererseits über einen zusätzlichen DMA-IP-Core mit Hilfe von Burst-Transfers (mit je 16 „Beats“ pro Burst) einen performanten Speicherzugriff zu ermöglichen.

Abbildung 5 stellt dar, wie der DMA IP-Core die Daten in 16-Beat-Bursttransfers mit je 64 Bit aus dem Speicher liest. Ein Bursttransfer holt hierbei die Daten für alle 16 Kanäle. Anschließend springt der DMA-Core zur nächsten Zeile der Matrix und liest die

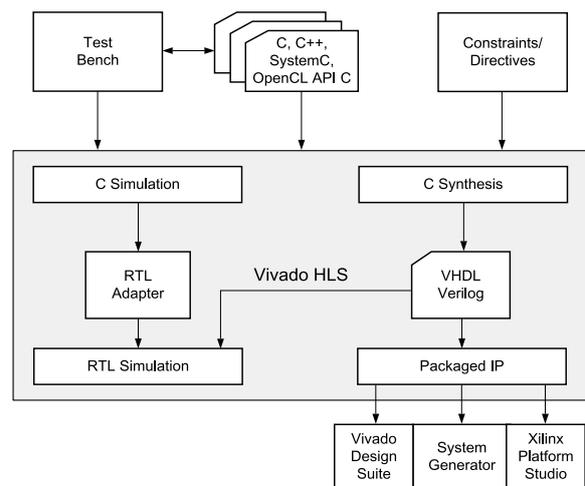


Abbildung 6. High-Level Synthese. Bildquelle [2].

nächsten Daten für die 16 Kanäle. Am Ausgang des DMA-Cores werden die Daten über ein 64 Bit breites AXI-Stream-Interface an den FFT-IP-Core weiterleitet, der die Daten dann auf die richtigen Kanäle aufteilt.

Auf Grund der Symmetrie der Fourier Transformation muss die zweite FFT auch nur über die Hälfte der Ergebnismatrix von FFT1 berechnet werden. Somit ist die Ergebnismatrix von FFT2 nur  $M \times N/2$  groß. Der FFT2-IP-Core besitzt am Ausgang ein AXI-Master-Interface und schreibt die Daten in der gleichen Anordnung in den Speicher, wie der DMA-Core die Daten gelesen hat. Somit stehen die Daten wieder in derselben Reihenfolge im Speicher. Die Ergebnis-Matrix der zweiten FFT beinhaltet nun in Zeilenrichtung die Entfernung von Objekten und in Spaltenrichtung die Geschwindigkeit. Diese Ergebnis-Matrix dient nun für die weitere Signalverarbeitung, wie beispielsweise der Zielerkennung durch den CFAR-Algorithmus (Constant False Alarm Rate).

### III. ENTWURFSPROZESS MIT HIGH-LEVEL SYNTHESE

Die benötigten IP Cores, welche im vorangegangenen Abschnitt im Rahmen des Systemkonzepts besprochen wurden, sind mit Hilfe der High-Level-Synthese (HLS) entwickelt worden. Hierfür wurden für jeden IP Core entsprechende Funktionen in C/C++ geschrieben. Obgleich es ebenso möglich wäre, das Systemkonzept mit Hilfe der Xilinx IP Cores (FFT, DMA) im Vivado IP Integrator umzusetzen, wurde aus folgenden Gründen die HLS verwendet:

- Die Anzahl und Konfiguration der FFT-Cores lässt sich im C-Code sehr schnell ändern.
- Die DMA-Funktion (AXI-Master-Interface) für den Zugriff auf den SDRAM-Hauptspeicher sowie die AXI-Stream- und AXI-Lite-Interfaces können mit Hilfe von wenigen Pragmas im C-Code beschrieben werden und die benötigte Hardware wird von Vivado HLS generiert.



Abbildung 7. IP Integrator Symbol für FFT1.

- Die Aufteilung der Eingangswerte auf die verschiedenen FFT-Kanäle von FFT2 kann mit C-Funktionen relativ einfach beschrieben werden.
- Eine Verifikation der Funktionalität und der per HLS generierten VHDL-Codes kann sehr einfach mit einer C-Testbench vorgenommen werden. Hierzu wurden mit Matlab entsprechende Eingangs-Stimuli und die zugehörigen Ausgangs-Referenzdaten erzeugt, welche in der C-Testbench eingelesen wurden.

Abbildung 6 zeigt den Ablauf der HLS (siehe auch [2]). Die Funktion des IP Cores wird mit C/C++ codiert. Hierzu wird eine „Toplevel“-Funktion benötigt, welche dann im generierten HDL-Code ebenfalls die oberste Schaltungsebene darstellt. Der C-Code kann hierarchisch in Unterfunktionen gegliedert werden und diese Hierarchie wird dann ebenfalls in eine HDL-Hierarchie umgesetzt. Für die Simulation wird eine „Testbench“ benötigt, die ebenfalls in C/C++ als „main“-Funktion geschrieben wird. Der erste Schritt besteht darin, die Funktionalität des C-Codes mit Hilfe dieser Testbench zu verifizieren („C Simulation“). Für die eigentliche High-Level Synthese („C Synthesis“) werden vom Benutzer Vorgaben bezüglich der Interfaces der Schaltung und der Optimierung der Schaltungsstruktur gemacht („Constraints/Directives“). Das Ergebnis der Synthese ist dann eine Register-Transfer-Architektur der Hardware, welche als VHDL- und Verilog-Code generiert wird. Ein optionaler aber empfehlenswerter Schritt ist dann die Verifikation des erzeugten HDL-Codes durch eine „Co-Simulation“ der C-Testbench mit dem HDL-Code („RTL Simulation“). Im Fall der im vorigen Abschnitt besprochenen beiden FFT-IP-Cores wurden für die Testbench mit Matlab Eingangs- und Referenzwerte für beide FFTs erzeugt, die dann auch für die „RTL Simulation“ verwendet wurden.

Im Folgenden soll nun anhand der ersten FFT (FFT1) die Vorgehensweise bei der HLS etwas genauer erläutert werden. Dies entspricht auch weitgehend der Vorgehensweise bei der zweiten FFT. Listing 1 zeigt einen Ausschnitt aus dem C-Code für FFT1. Die Datentypen der Argumente der Funktion `fft_1c_sin` bestimmen zusammen mit den Direktiven in Form von Pragmas (Zeile 6 bis 14) das Interface der durch HLS generierten Hardware. Das Argument `in0` stellt

Listing 1  
QUELLCODE-AUSSCHNITT FÜR FFT1.

```

1 void fft_1c_sin(data_in_t *in0,
2   if_out_t *out0,
3   const data_in_t window[FFT_LENGTH/2],
4   unsigned ramps) {
5
6   #pragma HLS INTERFACE s_axilite port=return
7   #pragma HLS INTERFACE s_axilite port=ramps
8   #pragma HLS INTERFACE s_axilite port=window
9
10  #pragma HLS INTERFACE m_axi port=out0 \
11  offset=slave max_write_burst_length=16
12
13  #pragma HLS INTERFACE axis register both \
14  port=in0
15
16  ...
17
18  copy_input(in0, xn, window);
19  hls::fft<params_t>(xn, yn, &fft_status, \
20    &fft_config);
21  copy_output(yn, out0, ramps);
22
23  ...

```

die Eingangswerte für die FFT dar und diese werden durch ein AXI-Stream-Interface implementiert (Pragma siehe Zeile 13-14). Das Argument `out0` stellt die Ausgangswerte der FFT dar und für diese wurde ein AXI-Master-Interface implementiert (Pragma siehe Zeile 10-11), so dass der IP-Core als DMA-Master selbständig Daten in den DRAM-Hauptspeicher des Systems transferieren kann.

Die beiden restlichen Argumente `ramps` (Anzahl der zu verarbeitenden Rampen/Chirps) und `window` (Daten für die FFT-Fensterung) wurden als User-Register im FFT-Core implementiert, welche über ein AXI-Lite-Interface schreibbar sind (Zeilen 6-8). Die Daten für die Fensterfunktion der FFT sind damit vor Beginn der FFT frei programmierbar. Ferner sorgt das Pragma in Zeile 6 auch dafür, dass der IP-Core ein Steuer- und Statusregister erhält, welches ebenfalls über das AXI-Lite-Interface schreib- und lesbar ist. Abbildung 7 zeigt das Ergebnis der Interface-Synthese in Form des Symbols des IP Cores, wie es dann im Xilinx Vivado IP Integrator erscheint. Damit kann der Core in das Gesamtsystem eingebaut werden. Neben den erwähnten Interfaces sind noch ein Takt- und Reseteingang sowie ein Interrupt-Ausgang vorhanden, welche standardmässig erzeugt werden. Über entsprechende User-Register können Interrupts freigeschaltet werden, so dass der Core beispielsweise einen Interrupt erzeugt, wenn die Abarbeitung beendet ist.

Abbildung 8 zeigt die innere Struktur des FFT1-IP-Cores, welche sich aus den Codezeilen 18 bis 21 in Listing 1 ergibt. Die Funktion `copy_input` sorgt im Wesentlichen dafür, dass die Eingangsdaten (reelle Werte) mit den Daten der Fensterfunktion multipliziert werden und dass die imaginären Werte zu Null gesetzt werden. Die Funktion `fft` in Zeile 19 ruft eine Bibliotheksfunktion für die FFT auf. Hiermit wird während der Synthese der entsprechende Xilinx FFT-

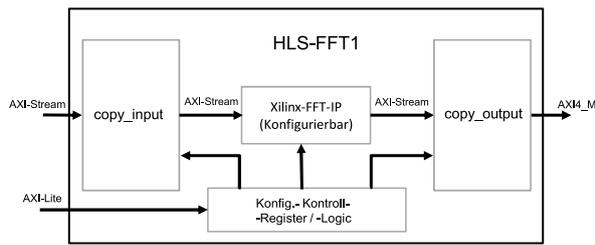


Abbildung 8. Hardwarestruktur der FFT1.

IP-Core eingebaut [3]. Dieser kann über den Template-Parameter `params_t` statisch und über das Funktionsargument `fft_config` zur Laufzeit konfiguriert werden [2]. Die Funktion `copy_output` kopiert die Ausgangsdaten der FFT in den Hauptspeicher des Systems. Hierbei wird eine entsprechende Adressarithmetik implementiert, welche dafür sorgt, dass jedes Sample, bestehend aus Real- und Imaginärteil, aufeinanderfolgend in den Speicher kopiert wird. Ferner wird noch dafür gesorgt, dass die aufeinanderfolgenden Rampen/Chirps ebenfalls sequentiell im Speicher aufeinander folgen (siehe Abschnitt II). Sind alle Rampen verarbeitet worden, dann bricht der Adresszeiger um, so dass für die nächste Messung die Daten im Speicher wieder überschrieben werden.

#### IV. PERFORMANCE UND RESSOURCENVERBRAUCH

Für die Performance-Untersuchungen wurde ein spezielles System auf dem Zynq-FPGA konfiguriert, bei welchem die Daten für die erste FFT nicht über die JESD-Schnittstelle zugeführt wurden. Mit Hilfe eines zusätzlichen DMA-Cores wurden Testdaten aus dem DRAM-Hauptspeicher gelesen und der ersten FFT zugeführt. Die Performance wurde unter Linux mittels eines Software-Timers für jeden IP-Core separat ermittelt. Hierdurch werden das Starten des IP-Cores per Software mit erfasst und auch die Speicherzugriffe. Mit dieser Konfiguration kann auch die Totzeit, die der IP-Core für die erste FFT zwischen zwei aufeinanderfolgenden FFTs benötigt, mitgemessen werden. Die Hardware wurde für die Performance-Untersuchung auf eine Chirp-Sequenz von 64 Rampen mit je 4096 Punkten konfiguriert und wird mit 100 MHz getaktet.

Die gemessene Ausführungszeit beträgt 3,4 ms für die erste FFT und 5,0 ms für die zweite FFT Stufe. Zum Vergleich benötigt die 2D-FFT für eine gleich große Datenmatrix auf einem PC mit 3,6 GHz I7-Quad-Core-Prozessor, unter Verwendung der 2D-FFT in Matlab, rund 5 ms für die Berechnung. Für die mögliche Maximalkonfiguration von 256 Rampen und 4096 Punkten lässt sich dank der  $N \cdot \log(N)$  Skalierung der FFT eine Zeit von 14 ms für die erste FFT-Stufe und ca. 27 ms für die zweite FFT-Stufe abschätzen, so dass hiermit die erwünschte Messfrequenz von 10 Hz auch für die Maximalkonfiguration sichergestellt werden kann.

Die erste FFT-Stufe benötigt nach jeder FFT mindestens eine Totzeit von 40 Takten (entspricht 400 ns), bevor die nächsten Daten eingelesen werden können. Da die Pausenzeit zwischen zwei Chirp-Sequenz-Rampen meist im Bereich von  $50 \mu\text{s}$  bis  $300 \mu\text{s}$  liegt, je nach Geschwindigkeitsauflösung, kann also auch sichergestellt werden, dass die Daten der Chirp-Sequenz problemlos der ersten FFT zugeführt werden können.

Tabelle I zeigt den Ressourcenaufwand an LUTs, Flip-Flops, Block-RAM und DSP-Einheiten der einzelnen HLS-IP-Cores im Verhältnis zu den verfügbaren Einheiten des Zynq 7030. Aufgrund der hohen Parallelisierung innerhalb der zweiten FFT-Stufe wird hier ein Großteil der Ressourcen in Anspruch genommen, wobei die Gesamtauslastung bezogen auf den Zynq 7030 mit maximal 31 % noch relativ gering ist.

Innerhalb der jeweiligen IP-Cores wird LUT-RAM überwiegend für die FFT- und Window-Koeffizienten verwendet, wohingegen das BRAM für die Eingangs- und Ausgangspuffer, sowie für das Speichern von Zwischenergebnissen verwendet wird. Auffällig ist hier, dass die erste FFT aufgrund der Streaming-Eigenschaften mit 15 BRAM-Blöcken einen relativ hohen Speicherbedarf hat. Die letzten beiden Zeilen in Tabelle I zeigen den Gesamtverbrauch des Systems aus Abbildung 3 ohne CFAR. Die Gesamtauslastung liegt auch hier mit Ausnahme der DSP-Einheiten unter 30 %, womit ausreichend Platz für eine Erweiterungen der zweiten FFT-Stufe auf 256 Rampen pro Messung, sowie die Implementierung des CFAR-IP-Cores verfügbar ist.

#### V. ZUSAMMENFASSUNG UND AUSBLICK

Der Zynq 7030 bietet ausreichend Performance und Ressourcen für die Realisierung des Radar-Demonstrators und erfüllt die Anforderungen aus Abschnitt I. Die Architektur des Dual-Core-ARM-9-Prozessors mit integriertem FPGA erlaubt es, ein flexibel konfigurierbares System aufzubauen, das komfortabel bedienbar ist. Die Signalverarbeitungsperformance des Systems ist vergleichbar mit der 2D-FFT aus Matlab, berechnet auf einem I7-Quad-Core PC mit 3,6 GHz. Hierbei ist zu beachten, dass die IP-Cores im PL-Teil des Zynq nur mit 100 MHz getaktet werden. Dies zeigt deutlich, dass durch Auslagerung von rechenintensiven Anwendungen in FPGA-Hardware auch mit kleinen eingebetteten Systemen hohe Rechenleistungen erzielt werden können, für die anderenfalls sehr leistungsfähige Rechnersysteme mit entsprechendem Platz- und Energiebedarf notwendig wären.

Die Probleme der Verzögerungen durch die Speicherzugriffe der zweiten FFT-Stufe aufgrund der Anordnung der Daten konnten durch die Implementierung einer entsprechenden Adressarithmetik im DMA-IP-Core und die Verwendung mehrerer FFT-Kanäle ausgeglichen werden, so dass die Daten wieder mittels

Tabelle I  
RESSOURCENAUFWAND DER HLS-IP-CORES.

	LUT-Logic	LUT-RAM	FF	BRAM	DSP
Zynq 7030	78600	26600	157200	265	400
FFT1	3187	1198	6813	15	20
DMA FFT2	427	18	928		
FFT2	13009	3262	33310	35	104
Summe	16623	4478	41051	50	124
% Usage	21,2	16,8	26,1	18,9	31,0
Gesamtsystem	19931	4746	45450	50	124
% Usage	25,4	17,8	28,9	18,9	31,0

Bursttransfers gelesen und geschrieben werden können. Durch die Verwendung mehrerer FFT-Kanäle wird die Abarbeitung der zweiten FFT auch parallelisiert, was ebenfalls zu einer Leistungssteigerung führt und auch nochmals die Vorteile der Hardwareimplementierung zeigt. Weitere Leistungssteigerungen sind dabei möglich, indem weitere FFT-Kanäle implementiert und damit eine noch stärkere Parallelisierung der zweiten FFT erreicht werden kann.

Der Einsatz der High-Level Synthese hat sich für die Implementierung der IP-Cores bewährt. Damit können relativ schnell unterschiedliche Architekturen implementiert und verglichen werden, zudem vereinfacht sie auch die Implementierung und Anbindung komplexer Schnittstellen, wie zum Beispiel eines AXI4-Master-Interfaces.

Der Radar-Sensor wurde mit dem 160-GHz-Front-End erfolgreich in Betrieb genommen und wird als Demonstrator eingesetzt. Der Radar-Sensor soll weiterhin um diverse Funktionen erweitert werden. So wird zum Beispiel die Signalverarbeitung noch für eine Chirpsequenz der Länge  $4096 \times 256$  angepasst und der CFAR-Algorithmus wird als Hardware IP-Core umgesetzt. Das Web-Interface, das momentan nur zur Visualisierung der Radardaten dient, wird um Konfigurationsmöglichkeiten erweitert, so dass zum einen Chirp-Sequenz-Parameter, wie Rampensteilheit und Pausenzeit zwischen den Rampen, einstellbar sind und zum anderen auch die Hardware umkonfiguriert werden kann, damit unterschiedliche Kombinationsmöglichkeiten von Anzahl an Rampen und Punkten pro Rampe möglich sind.

#### DANKSAGUNG

Die hier beschriebenen Arbeiten sind ein Teilprojekt des ZAFH „MikroSens“, welches vom Land Baden-Württemberg und der Europäischen Union im Rahmen des Europäischen Fonds für regionale Entwicklung (EFRE) gefördert wird. Die Autoren bedanken sich beim Land Baden-Württemberg und der EU für die Möglichkeit dieses Projekt durchzuführen.

#### LITERATURVERZEICHNIS

- [1] Zynq-7000 SoC Technical Reference Manual, UG585 (v1.12.2) July 1, 2018.
- [2] Vivado Design Suite User Guide High-Level Synthesis, UG902 (v2017.4) February 2, 2018.
- [3] Fast Fourier Transform v9.0 LogiCORE IP Product Guide Vivado Design Suite, PG109 November 18, 2015.
- [4] JESD204 v7.1 LogiCORE IP Product Guide Vivado Design Suite, PG066 April 05, 2017.
- [5] FM/CW Radar Signals and Digital Processing, Donald E. Barriky, NOAA Technical Report ERL 283-WPL 26 by NTIS National Technical Information Service, July 1973.



**Torsten Bandel** erhielt den akademischen Grad des Bachelor of Engineering in Elektrotechnik von der Hochschule Pforzheim 2013 und den akademischen Grad des Master of Science in Embedded Systems ebenfalls von der Hochschule Pforzheim 2017. Seit 2017 ist er wissenschaftlicher Mitarbeiter im Forschungsprojekt MikroSens und entwickelt hauptsächlich analoge und digitale Hardware für Radar-Sensoren.



**Frank Kesel** erhielt den akademischen Grad Dipl.-Ing. in Elektrotechnik von der Universität Karlsruhe 1988 und den Grad Dr.-Ing. in Elektrotechnik von der Universität Hannover 1994. Von 1990 bis 1998 arbeitete er für die Robert Bosch GmbH und anschließend für Philips Semiconductors im Entwurf von digitalen ASICs. Er ist seit 1999 Professor an der Hochschule Pforzheim und beschäftigt sich in Lehre und Forschung mit dem Thema Entwurf von digitalen Schaltungen und Systemen mit FPGAs und SOPCs.



**Wolf-Henning Rech** erhielt den akademischen Grad Dipl.-Ing. in Elektrotechnik von der Universität Karlsruhe 1985 und den Grad Dr.-Ing. in Elektrotechnik von der Universität Karlsruhe 1991. Von 1991 bis 1997 arbeitete er für die Robert Bosch GmbH im Bereich der optischen Kommunikation. Er ist seit 1997 Professor an der Hochschule Pforzheim und beschäftigt sich in Lehre und Forschung mit dem Thema Hochfrequenz- und Mikrowellen-Elektronik.

# Anwendung eines heuristischen Approximationsverfahrens zur Multi-Kriterien Optimierung von Energieclustern

Stefan Rädle, Johannes Mast, Joachim Gerlach

**Zusammenfassung**—Die vorliegende Arbeit untersucht die Übertragung und Anwendung von algorithmischen Lösungsansätzen aus dem Bereich der künstlichen Intelligenz (KI) in den Bereich des Systementwurfs anhand eines Anwendungsbeispiels aus dem Bereich der Energietechnik. Aufbauend auf einer Werkzeugumgebung zur skalierbaren Modellierung und Simulation energietechnischer Anlagen und Systeme [1] wird ein KI-Ansatz für die optimierte Steuerung eines virtuellen Kraftwerks im Hinblick auf multiple und variierende Zielkriterien angewendet. Im Rahmen von experimentellen Untersuchungen wird aufgezeigt, in welchem Umfang KI-basierte Lösungsverfahren in der Lage sind, die Nutzung eines virtuellen Kraftwerks zum Ausgleich des Regelenergiebedarfs eines Stromnetzbereichs durch eine intelligente Steuerung der Anlagen zu optimieren, im Vergleich zu konventionellen Steuerungsansätzen. Hierbei fließen komplexe Randbedingungen, wie z.B. Wetterverläufe und anlagenspezifische Steuerungsanforderungen, sowie verschiedenartige Optimierungskriterien, wie z.B. Optimierung des Ressourceneinsatzes oder Minimierung der CO<sub>2</sub>-Emissionen, in die Betrachtung ein.

**Schlüsselwörter**—Energietechnische Systeme, Erneuerbare Energien, Virtuelles Kraftwerk, Künstliche Intelligenz, Intelligente Anlagen-Steuerung, Simulated Annealing

## I. EINLEITUNG

Die Umsetzung der von der Bundesregierung Deutschland beschlossenen Energiewende erfordert den massiven Ausbau von sogenannten erneuerbaren Energien mit Photovoltaik und Windkraft als den bekanntesten Vertretern. Die Umstrukturierung der Stromerzeugung soll wesentlich dazu beitragen, den Ausstoß von Treibhausgasen und den damit einhergehenden Klimawandel auf ein akzeptables Niveau zu senken [2]. Die zunehmende Integration solcher erneuerbaren Energien in die vorhandene Strom-Infrastruktur bringt allerdings auch zahlreiche technische Probleme mit sich. Eines der größten Probleme stellt dabei die Netzstabilität dar. Die durch ihre Wetterabhängigkeit stark schwankende Stromproduktion von Photovoltaik und Windkraft führt auch zu zunehmenden Schwankungen im Stromnetz, die wiederum aufwendig durch konventionelle Schattenkraftwerke ausgeglichen werden müssen [3].

Stefan Rädle, raedle@hs-albsig.de, Johannes Mast, mast@hs-albsig.de, Joachim Gerlach, gerlach@hs-albsig.de, Hochschule Albstadt-Sigmaringen, Jakobstraße 6, 72458 Albstadt-Ebingen

In diesem Zusammenhang spielt die Bildung von virtuellen Kraftwerken – dem logischen Zusammenschluss von mehreren, stromerzeugenden Anlagen unterschiedlichen Typs – eine immer größere Rolle. Ziel dabei ist es, die Anlagenverbünde „von außen“ wie ein einzelnes Kraftwerk betrachten und regeln zu können [4]. Die Regelung solcher virtuellen Kraftwerke erweist sich aufgrund komplexer Randbedingungen wie z.B. Wetterverläufen, anlagenspezifischen Steuerungsanforderungen oder weiterer zu beachtenden Lastkurven allerdings als eine schwierige Aufgabe. Hierbei lohnt es sich zu untersuchen, inwieweit KI-basierte Lösungsverfahren, die sich in Bereichen mit ähnlichen komplexen Randbedingungen wie der Sprach- und Gesichtserkennung bereits durchgesetzt haben, auch zur Optimierung der Anlagen-Steuerungen hinsichtlich des Regelenergiebedarfs gewinnbringend eingesetzt werden können.

## II. GRUNDLAGEN

Im folgenden Kapitel werden die Grundlagen behandelt, auf deren Basis die restliche Arbeit aufgebaut ist. Diese umfassen die Werkzeugumgebung, die zur Modellierung und Simulation des energetischen Gesamtsystems eingesetzt wurde, die spezifischen Randbedingungen der Steuerungen der einzelnen Komponenten sowie die KI-Methodik „Simulated Annealing“ (SA).

### A. Werkzeugumgebung

Durch die Arbeiten des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Forschungsprojekt SADE („Simulative Analyse dezentraler Energieversorgungskonzepte“) der Hochschule Albstadt-Sigmaringen ist eine Werkzeugumgebung entstanden, die die Modellierung, Simulation und Visualisierung hierarchischer, komplexer Systeme aus dem Bereich der Energietechnik ermöglicht.

Abbildung 1 stellt besagte Werkzeugumgebung dar. Für eine realitätsnahe Simulation der Systeme sind mehrere Anbindungen an externe Quellen gegeben, um auf deren Daten zugreifen zu können. Dazu zählen die Strom-Börsenpreise der EPEX SPOT, Wetterdaten des deutschen Wetterdienst DWD sowie Messdaten von realen, im Feld befindlichen Anlagen durch die Projektpartner. Die in dieser Umgebung simulierten

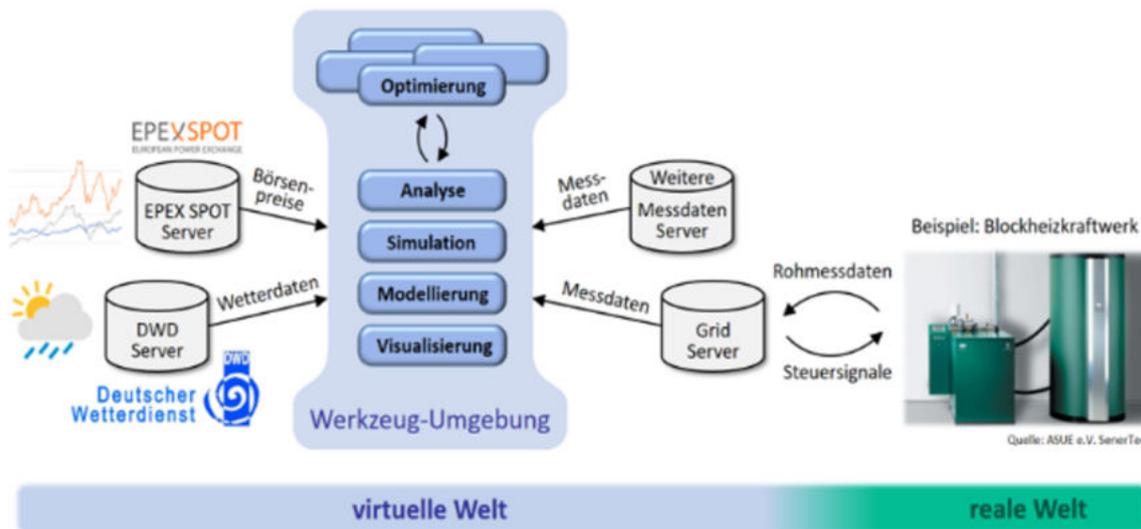


Abbildung 1. Werkzeugumgebung des SADE-Projekts.

Leistungskurven der Systemkomponenten dienen als Basis für die Entwicklung der optimierten Steuerung.

### B. Randbedingungen der Steuerungen

Zur Entwicklung einer optimierten Steuerung für ein virtuelles Kraftwerk müssen die spezifischen Randbedingungen aller Teil-Anlagen, aus denen es besteht, beachtet werden. Hierfür wird im Folgenden beispielhaft von einem aus drei Anlagen bestehenden virtuellen Kraftwerk ausgegangen, deren Randbedingungen in den nachfolgenden Abschnitten erläutert werden:

- Ein Blockheizkraftwerk (BHKW)
- Eine Photovoltaik-Anlage (PV)
- Eine Windkraft-Anlage (WK)

**1) BHKW-Randbedingungen:** Die wichtigste Randbedingung bei der Steuerung eines BHKWs stellt die thermische Lastkurve dar, die abgedeckt werden muss. Eine optimierte Steuerung muss gewährleisten, dass zu jedem Zeitpunkt ausreichend Wärmeenergie in einem Wärmespeicher vorhanden ist, um die Last kompensieren zu können, darf das BHKW aber auch nicht zu lange aktiv halten, da sonst ein Speicherüberlauf auftreten kann, was eine Notabschaltung zur Folge hätte. Weiterhin sind bei einem BHKW für gewöhnlich Mindestlauf- und -kühlzeiten zu beachten, die von einer Steuerung nicht unterschritten werden dürfen und dadurch die Anzahl der Ein- bzw.- Ausschaltvorgänge während eines bestimmten Zeitraums limitieren.

**2) PV-Randbedingungen:** Die Randbedingungen einer PV-Anlage sind in Form ihrer Steuerungsmethodik gegeben. Die gängigste Variante stellt dabei die „Softe Abregelung“ dar. Diese ermöglicht es, die Stromproduktion der Anlage auf prozentuale Anteile der Maximalleistung zu reduzieren, welche wiederum durch die Strahlungsintensität der Sonne zu einem Zeitpunkt

geprägt ist. Ein gängiger Wert hierfür ist z.B. eine 25%-Schrittweite. [5]

**3) WK-Randbedingungen:** Bei WK-Anlagen erfolgt die Steuerung normalerweise über eine „Pitch“-Regelung, die durch Rotation um ihre eigene Achse die Rotorblätter aus dem Wind dreht und dadurch die Auftriebskraft nahezu stufenlos reduzieren kann. Hier ist weiterhin zu beachten, dass das Ausmaß der Steuerungsmöglichkeiten sich je nach Windgeschwindigkeit ändert. Hierfür sind folgende Geschwindigkeitsbereiche definiert: [6]

- Bei Geschwindigkeiten kleiner als vier m/s können die Rotorblätter nicht angetrieben werden, eine externe Steuerung ist nicht möglich.
- Bei Geschwindigkeiten zwischen vier m/s und zwölf m/s befindet die Anlage sich im Normalbetrieb ohne Einschränkungen für die Steuerung.
- Bei Geschwindigkeiten zwischen zwölf m/s und 25 m/s muss die Leistungsabgabe automatisch durch eine von der Steuerung vorgegebene Mindest-Abregelung gesichert werden.
- Ab einer Geschwindigkeit von 25 m/s muss die Anlage abgeschaltet werden, eine Steuerung ist nicht möglich.

### C. KI-Methodik

Zur Lösung der Problemstellung wurde die KI-Methodik „Simulated Annealing“ verwendet. Hierbei handelt es sich um eine heuristische stochastische Suchmethodik mit dem Ziel, mit einem akzeptablen Zeitaufwand eine für ein Optimierungsproblem möglichst gute Lösung innerhalb eines komplexen Zustandsraums zu finden. Die Funktionsweise dieser Methodik ist dabei an den Abkühlungsprozess heißer Metalle angelehnt: Zu Beginn, wenn noch eine hohe Temperatur vorliegt, nimmt die Gitterstruktur des

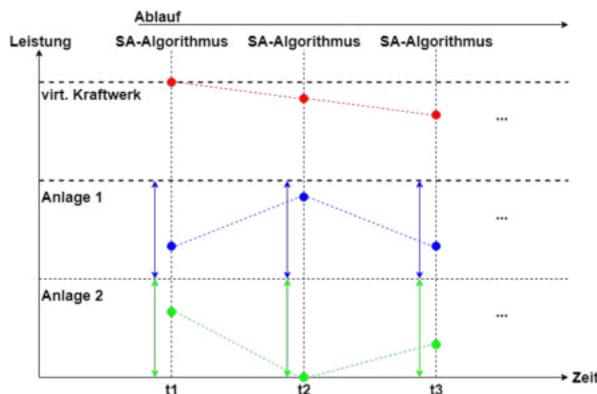


Abbildung 2. Definition der Steuerungsmethodik.

Metalls zufällig stabile (gute Lösungen) oder instabile (schlechte Lösungen) Zustände an. Mit zunehmender Abkühlung nimmt die Wahrscheinlichkeit für das Entstehen instabiler Zustände ab und es stellt sich gegen Ende eine stabile Struktur ein (finale Lösung). Die Übernahme schlechterer Zustände zu Beginn erlaubt es der Methodik, lokal gute Lösungen im Zustandsraum zu überwinden und so den gesamten Raum durchsuchen zu können. Um die Methodik an eine gegebene Problemstellung zu adaptieren, müssen zwei Funktionen bereitgestellt werden: Die *Nachbarschaftsfunktion* bestimmt anhand eines momentan ausgewählten Zustands zufällig einen alternativen Zustand, sodass beide Zustände im Zustandsraum nahe beieinanderliegen. Die *Fehlerfunktion* bewertet einen gegebenen Zustand als mögliche Lösung hinsichtlich des Optimierungsproblems, wobei ein Zustand eine umso bessere Lösung darstellt, je niedriger der Fehlerwert ist. Die Methodik ruft iterativ beide Funktionen auf, um neue Zustände zu generieren und zu bewerten. Zustände mit einem besseren Fehlerwert werden immer übernommen, Zustände mit einem schlechteren Fehlerwert werden nur mit einer Wahrscheinlichkeit übernommen, die mit zunehmender Iterationszahl sinkt (Abkühlvorgang). Ist der Abkühlvorgang abgeschlossen, wird der bis dahin beste gefundene Zustand als Lösung empfohlen. [7]

### III. ANALYSE UND IMPLEMENTIERUNG

Das folgende Kapitel befasst sich mit der Umsetzung der optimierten Steuerungsmethodik. Hierfür wird zuerst auf Basis der zuvor beschriebenen Anlagen-Randbedingungen die allgemeine Problematik der Steuerungslogik erläutert. Anschließend wird anhand eines konkreten Beispiels die Auswahl der KI-Methodik Simulated Annealing begründet und deren Adaption an die Problemstellung über die Definition der Nachbarschafts- und Fehlerfunktion erläutert.

#### A. Analyse der Problemstellung

Ziel der Methodik soll es sein, durch Ausnutzung aller Steuerungsmöglichkeiten der einzelnen Anlagen

innerhalb des virtuellen Kraftwerks die in Summe entstehende Leistungskurve so anzupassen, dass sie den Regelenergiebedarf des Stromnetzes möglichst gering hält und so vor allem die Integration von erneuerbaren Energien in die Stromproduktion erleichtert. Dazu wird von einer vorgegebenen Verbraucherkurve ausgegangen, die vom virtuellen Kraftwerk möglichst gut nachgebildet werden soll, dabei jedoch keine der spezifischen Randbedingungen der Anlagen verletzen darf. Neben dem Ausgleich des Regelenergiebedarfs als Hauptziel soll es weiterhin möglich sein, auf zusätzliche Nebenkriterien optimieren zu können wie beispielsweise die Minimierung der CO<sub>2</sub>-Emissionen.

In Abbildung 2 ist die grundlegende Funktionsweise dieser Steuerungsmethodik dargestellt. Die Leistungskurven der Anlagen werden zeitlich diskretisiert und anschließend von links nach rechts – zeitlich fortlaufend – betrachtet. Für jeden Zeitpunkt  $t$  soll ein SA-Algorithmus dessen Optimierung übernehmen. Dazu regelt er zufällig einzelne Anlagen nach oben bzw. unten und manipuliert so den in Summe entstehenden Leistungspunkt des virtuellen Kraftwerks, der daraufhin auf das Optimierungsziel hin überprüft wird. Sind alle Zeitpunkte abgearbeitet, ist sowohl für das virtuelle Kraftwerk eine Leistungskurve entstanden, die das Gesamtergebnis der Optimierung bewertet, als auch Leistungskurven für alle Teil-Anlagen, die zum Generieren konkreter Steuerungssignale und damit zur Umsetzung der gefundenen Lösung dienen.

#### B. Mathematische Formulierung

Zur Entwicklung der Steuerungsmethodik soll im Vorfeld eine mathematische Formulierung stattfinden, um die Problematik in eine durch Informationstechnik lösbare Form zu bringen. Hierfür wird von einem allgemeinen virtuellen Kraftwerk ausgegangen, welches aus  $M$  Teil-Anlagen beliebiger Typen besteht. Weiterhin seien zu jeder Teil-Anlage  $i$  folgende Funktionen in Abhängigkeit der Zeit  $t$  gegeben:

- Die Funktionen  $P_{i,\min}(t)$  und  $P_{i,\max}(t)$  stellen die theoretisch minimal und maximal möglichen Leistungsgrenzen einer Anlage  $i$  zum Zeitpunkt  $t$  dar. Die theoretische, minimale Leistung der Anlagen beträgt dabei für gewöhnlich Null – da sie sich alle abschalten lassen – wird hier aber trotzdem als eigenständige Funktion vorgesehen, um die Kompatibilität für etwaige, spätere Anpassungen zu erhalten. Beim BHKW der Beispiel-Anlagen entspricht die maximale Leistungsgrenze zu jedem Zeitpunkt der Nennleistung des Motors, bei der PV- und WK-Anlage wird diese durch die von der Werkzeugumgebung anhand von Wettervorhersagen simulierten Leistungskurven dargestellt. Die simulierte WK-Kurve berücksichtigt dabei auch die im Abschnitt „WK-Randbedingungen“ angegebenen Windgeschwindigkeiten, sodass die

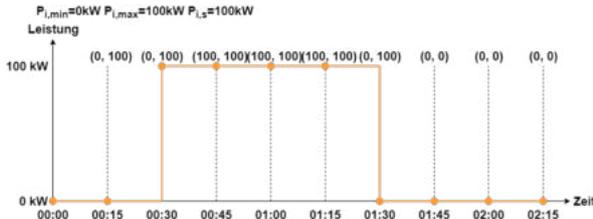


Abbildung 3. Beispiel für die Anlagen-Funktionen.

Steuerungsmethodik hierauf nicht gesondert achten muss.

- Die Funktionen  $\tilde{P}_{i,\min}(t)$  und  $\tilde{P}_{i,\max}(t)$  stellen entsprechend die tatsächliche minimale und maximale Leistungsgrenze dar. Der Unterschied zu den theoretischen Werten besteht darin, dass hier Einschränkungen durch die Steuerungsanforderungen einer Anlage mitberücksichtigt werden. Bei den Beispiel-Anlagen wird dies durch die Mindestlauf- und kühlzeiten sowie durch die thermische Lastkurve des BHKWs repräsentiert.
- Zuletzt seien noch die Funktionen  $P_{i,s}(t)$  und  $P_i(t)$  gegeben. Die Funktion  $P_{i,s}(t)$  bestimmt die Schrittweite einer Anlage  $i$  zum Zeitpunkt  $t$ . Diese entspricht dem kleinsten Leistungsbetrag, um den der Leistungswert einer Anlage innerhalb ihrer Grenzen hoch- bzw. runtergeregelt werden kann. Beim BHKW entspricht dieser Wert der Nennleistung des Motors und bei der PV-Anlage kann hierdurch die prozentuale Regelung realisiert werden. Da sich die WK-Anlage theoretisch stufenlos regeln lässt, muss hier durch den Anlagenbetreiber ein Mindestwert spezifiziert werden. Die Funktion  $P_i(t)$  schließlich stellt den tatsächlichen, durch die Methodik festgelegten Leistungspunkt einer Anlage  $i$  zum Zeitpunkt  $t$  dar (alle erwähnten Leistungswerte sind in Watt angegeben).

Zur Verdeutlichung der einzelnen Funktionen ist in Abbildung 3 ein beispielhafter Kurvenverlauf für ein BHKW und die zugehörigen Funktionswerte dargestellt. Da die theoretischen Leistungsgrenzen sowie die Schrittweite konstant über die gesamte Zeitreihe hinweg sind, sind sie nur einmalig angegeben. Die tatsächlichen Leistungsgrenzen sind zu jedem Zeitpunkt als Tupel aus minimaler und maximaler Grenze angegeben. Zwischen zwei Zeitpunkten liegt hier beispielhaft ein zeitlicher Abstand von 15 Minuten. Zu Beginn der Zeitreihe wird davon ausgegangen, dass das BHKW noch keinen Einschränkungen unterliegt. Folglich entsprechen die tatsächlichen Leistungsgrenzen den theoretischen Werten - null und 100 Kilowatt (kW). Zum Zeitpunkt 00:30 Uhr entscheidet sich die Methodik dazu, dass BHKW einzuschalten, indem es die Schrittweite einmalig auf den Leistungswert aufaddiert. Ab hier greift nun die Mindestlaufzeit des BHKWs von beispielhaft einer Stunde; dementsprechend darf es

während der folgenden drei Zeitpunkte nicht wieder ausgeschaltet werden. Das BHKW teilt dies der übergeordneten Methodik mit, indem es die minimale Leistungsgrenze ebenfalls auf 100 kW setzt und so die Mobilität zwischen minimaler und maximaler Grenze auf null reduziert. Die Methodik kann somit während dieser Zeitpunkte die Leistungsabgabe des BHKWs nicht verändern. Ab 01:30 Uhr ist die Mindestlaufzeit abgelaufen und das BHKW setzt die minimale Grenze wieder auf null: Die Mobilität steigt wieder auf 100 kW und die Methodik kann das BHKW wieder ausschalten (falls notwendig). Nach dem Ausschaltvorgang greift entsprechend die Mindestkühlzeit von beispielhaft einer Stunde. Hier setzt das BHKW die maximale Grenze auf null, um so wiederum die Mobilität auf null zu reduzieren und einen Einschaltvorgang zu verhindern. Entsprechend könnte das BHKW verfahren, wenn es zu einem Zeitpunkt aufgrund der thermischen Last zwangsweise ein- bzw. ausgeschaltet werden muss. Die WK- bzw. die PV-Anlage können die obere Grenze nutzen, um der Methodik zu jedem Zeitpunkt die über die simulierten Kurven definierten, maximal möglichen Leistungsabgaben mitzuteilen.

Durch Anwendung dieser Funktionen ist es somit möglich, dass jeder Anlagentyp seine spezifischen Randbedingungen berücksichtigen und diese Informationen neutral an die übergeordnete Methodik weiterreichen kann. Hierauf aufbauend kann die tatsächliche Leistungsabgabe einer Anlage  $i$  zum Zeitpunkt  $t$  als Summe aus der minimalen Leistungsgrenze und einem natürlichen Vielfachen  $k_i(t)$  der Schrittweite beschrieben werden (1):

$$P_i(t) = \tilde{P}_{i,\min}(t) + k_i(t) \cdot P_{i,s}(t) \quad (1)$$

Um die obere Leistungsgrenze der Anlage einzuhalten, wird für den Koeffizienten  $k_i(t)$  ebenfalls eine obere Grenze  $k_{i,\max}(t)$  definiert, die sich durch das Ersetzen von  $P_i(t)$  durch  $\tilde{P}_{i,\max}(t)$  bestimmen lässt. Der Koeffizient  $k_i(t)$  lässt sich demnach wie folgt definieren (2):

$$k_i(t) \in \mathbb{N}_0 : k_i(t) \leq \left\lceil \frac{\tilde{P}_{i,\max}(t) - \tilde{P}_{i,\min}(t)}{P_{i,s}(t)} \right\rceil \quad (2)$$

Betrachtet man nun die gesuchte Leistung des virtuellen Kraftwerks  $P(t)$ , lässt sich diese als Summe der Leistungswerte aller  $M$  Teil-Anlagen definieren (3):

$$P(t) = \sum_{i=1}^M P_i(t) \quad (3)$$

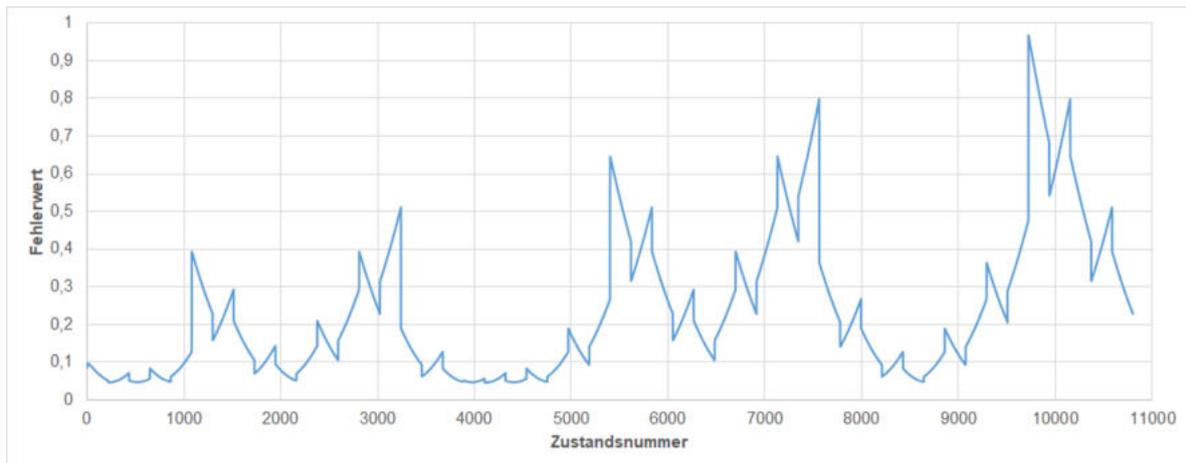


Abbildung 4. Beispielhafter Zustandsraum.

In der resultierenden Summe sind jeweils  $M$  Werte für die minimale Leistungsgrenze sowie für die Schrittweite und den Koeffizienten vorhanden. Die Produktterme aus Koeffizienten und Schrittweiten lassen sich alternativ außerhalb des Summensymbols als  $M$ -elementige Vektoren zusammenfassen. Diese Darstellung ist für die spätere Optimierungslogik von Vorteil (4):

$$\begin{aligned}
 P(t) &= \vec{k}(t) \cdot \vec{P}_s(t) + \sum_{i=1}^M \tilde{P}_{i,\min}(t) \\
 \vec{k}(t) &= (k_1(t), k_2(t), \dots, k_M(t)) \\
 \vec{P}_s(t) &= (P_{1,s}(t), P_{2,s}(t), \dots, P_{M,s}(t))
 \end{aligned} \tag{4}$$

Die Gesamtleistung des virtuellen Kraftwerks zum Zeitpunkt  $t$  lässt sich nun abschließend formulieren als Summe aller minimalen Leistungsgrenzen sowie dem Skalarprodukt eines Koeffizienten-Vektors und eines Schrittweiten-Vektors. Der Wert des Koeffizienten-Vektors ist dabei für jeden Zeitpunkt  $t$  separat zu bestimmen.

### C. Auswahl der Methodik

Für die Auswahl der Methodik ist in Abbildung 4 der Verlauf des Fehlerwerts für einen beispielhaften, kleineren Zustandsraum mit ca. 11000 Zuständen dargestellt, für den eine erschöpfende Analyse noch mit vertretbarem Aufwand durchgeführt werden konnte. Zur Visualisierung wurden alle Werte des entsprechenden Koeffizienten-Vektors systematisch durch Inkrementieren eines Elements durchlaufen, der Fehlerwert jeder Kombination errechnet und die Ergebnisse als Liste dokumentiert. Wie in der Abbildung zu erkennen, ist der Zustandsraum durch zahlreiche lokale Minima geprägt. Zur Lösung der Problematik ist demnach eine Methodik vonnöten, die in der Lage ist, solche lokalen

Minima zu überwinden um möglichst das globale Minimum zu finden. Aus diesem Grund wurde in dieser Arbeit die Methodik Simulated Annealing gewählt, da diese durch randomisierte Berücksichtigung auch schlechterer Zustände in der Lage ist, lokale Minima zu überwinden und so im mittleren Fall eine akzeptable Lösung zu generieren.

### D. Nachbarschaftsfunktion

Anhand der vorangegangenen mathematischen Formulierung ist es die Aufgabe des SA-Algorithmus, die gegebene Funktion für die Leistung des virtuellen Kraftwerks so anzupassen, dass ein variierendes Optimierungsziel erreicht werden kann. Zur Manipulation der Leistungsabgabe dient dabei der Koeffizientenvektor  $\vec{k}$ . Entsprechend wird ein Zustand für den SA-Algorithmus hier als ein möglicher Koeffizientenvektor definiert. Der zu durchsuchende Zustandsraum beschreibt die Menge aller für einen Zeitpunkt gültigen Koeffizienten-Vektoren. Als Nachbarschaftsfunktion wird demnach eine Funktion  $N(x)$  gesucht, welche einen  $M$ -elementigen Vektor auf einen alternativen  $M$ -elementigen Vektor abbildet (5):

$$N : (\mathbb{N}_0)^M \rightarrow (\mathbb{N}_0)^M \tag{5}$$

Für die Manipulation eines gegebenen Vektors stehen der Nachbarschaftsfunktion dabei folgende Möglichkeiten zur Verfügung:

- Anzahl der zu ändernden Elemente des Vektors.
- Art der Änderung (Erhöhung/Verringerung).
- Größe der Elementänderung.

Da die Nachbarschaftsfunktion nur kleinere Änderungen vornehmen soll, wurde sich hier dafür entschieden, pro Aufruf lediglich ein Element des Vektors zu manipulieren. Die Art der Elementänderung sowie

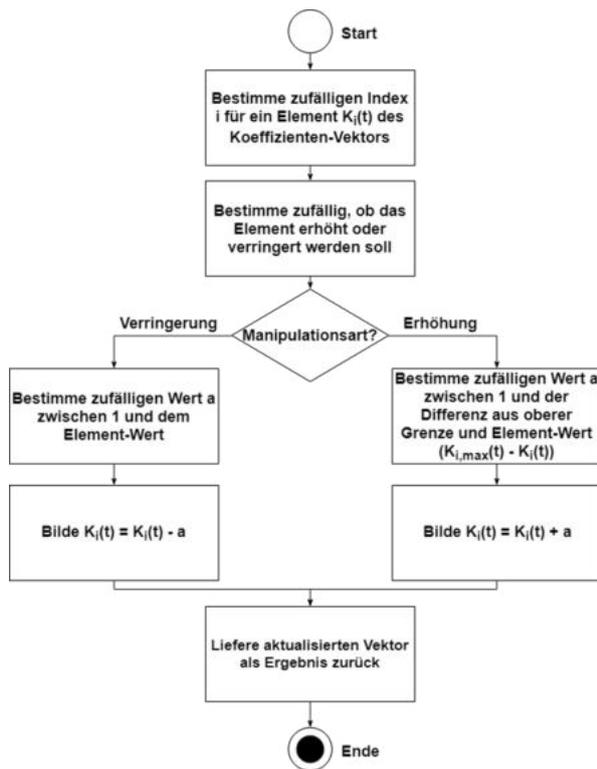


Abbildung 5. Ablauf der Nachbarschaftsfunktion.

die Größe werden entsprechend den Anforderungen an eine Nachbarschaftsfunktion zufällig bestimmt, jedoch unter Beachtung der Koeffizienten-Grenzen.

In Abbildung 5 ist der genaue Ablauf der Nachbarschaftsfunktion dargestellt. Zu Beginn wird ein zufälliger Wert zwischen eins und der Anzahl der Koeffizienten-Vektorelemente  $M$  gebildet und so ein Element des Vektors ausgewählt, welches abgeändert werden soll. Anschließend wird – ebenfalls zufällig – bestimmt, auf welche Art das Element verändert werden soll: Erhöhung oder Verringerung. Wird Verringerung gewählt, ist ein weiterer, zufälliger, ganzzahliger Wert zu bestimmen, der zwischen eins (minimale Änderung) und dem momentanen Elementwert liegt (maximale Änderung). Bei einer Erhöhung wird analog ein Wert zwischen eins und der Differenz aus oberer Schranke des Koeffizienten und Elementwert gebildet. Zuletzt wird das Vektorelement aktualisiert, indem der zuvor gewählte Wert subtrahiert bzw. aufaddiert wird.

### E. Fehlerfunktion

Zur Bewertung eines durch die Nachbarschaftsfunktion gebildeten Leistungswertes soll im folgenden Abschnitt eine Fehlerfunktion für den SA-Algorithmus entwickelt werden. Diese enthält beispielhaft folgende drei Optimierungskriterien:

- Abweichung vom Regelenergiebedarf.
- Ausmaß des Ressourceneinsatzes.
- Menge der CO<sub>2</sub>-Emissionen.

Das primäre Optimierungsziel stellt dabei der Ausgleich des Regelenergiebedarfs dar. Die Optimierungsziele „Ressourceneinsatz“ und „Menge der CO<sub>2</sub>-Emissionen“ sind sekundär und sollen der Methodik bei der Lösungsfindung dienen, wenn hinsichtlich des Regelenergiebedarfs mehrere gute Lösungen möglich sind. Anhand der Zustandsdefinition aus vorigem Abschnitt wird demnach eine Funktion  $E(x)$  benötigt, die einen  $M$ -elementigen Vektor auf einen reellen Wert abbildet (6):

$$E : (\mathbb{N}_0)^M \rightarrow \mathbb{R} \quad (6)$$

Dabei sollen alle Teil-Fehlerfunktionen der einzelnen Optimierungsziele im Intervall  $[0;1]$  liegen, um eine gewichtete Kombination zu ermöglichen.

1) *Optimierungsziel: Regelenergiebedarf:* Wie bereits im Abschnitt „Analyse der Problemstellung“ erläutert, muss die Leistungskurve des virtuellen Kraftwerks zur Optimierung des Regelenergiebedarfs eine vorgegebene Verbrauchercurve möglichst gut treffen. Hierfür wird die Differenz aus dem tatsächlichen Leistungspunkt und dem Verbraucherpunkt gebildet. Durch die Division mit dem Maximum aus der oberen Leistungsgrenze und dem Verbraucherpunkt (und durch die Potenzierung) wird der Wert in das im vorigen Abschnitt definierte Intervall überführt. Das Quadrieren weiterhin hat den Sinn, größere Abweichungen besonders hart zu bestrafen während kleinere Abweichungen eher akzeptiert werden (7):

$$A(t) = \left( \frac{P(t) - V(t)}{\max(\tilde{P}_{\max}(t), V(t))} \right)^2 \quad (7)$$

Dabei stellt  $P(t)$  die Leistung des virtuellen Kraftwerks zum Zeitpunkt  $t$  dar,  $V(t)$  die benötigte Verbraucherleistung,  $\tilde{P}_{\max}(t)$  die maximal mögliche Leistung und  $A(t)$  den Fehlerwert.

2) *Optimierungsziel: Ressourceneinsatz:* Das Optimierungsziel „Ressourceneinsatz“ hat den Sinn, die Anlagensteuerung zu einem Zeitpunkt  $t$  so zu bestimmen, dass sie während den darauffolgenden Zeitpunkten möglichst wenig eingeschränkt ist, um so das Optimierungspotential so hoch wie möglich zu halten. Ein Beispiel hierfür sind die Mindestlauf- und -kühlzeiten des BHKWs. Entscheidet die Methodik sich dafür, den Zustand des BHKWs (eingeschaltet bzw. ausgeschaltet) zu einem Zeitpunkt zu ändern, ist sein Zustand gegebenenfalls für mehrere nachfolgende Zeitpunkte unveränderlich. Die entsprechende Dimension im Zustandsvektor ist demnach fix in ihrem Wert und die Größe des Zustandsraum halbiert sich, was auch den Verlust von – hinsichtlich der Fehlerfunktion

Tabelle I  
 CO<sub>2</sub>-WERTE DER BEISPIEL-ANLAGEN.

Anlagentyp	CO <sub>2</sub> -Wert $C_i$
BHKW	425 g/kWh
PV-Anlage	75 g/kWh
WK-Anlage	25 g/kWh

– guten Zuständen zur Folge haben und das Ergebnis insgesamt verschlechtern kann. Zur Bestimmung dieser Auswirkung kann das Verhältnis aus theoretischen und tatsächlichen Leistungsgrenzen einer Anlage herangezogen werden, da sich die tatsächlichen Grenzen – wie im Abschnitt „Mathematische Formulierung“ beschrieben – von den theoretischen Grenzen nur durch besagte Steuerungsauswirkungen unterscheiden. Der Fehlerwert für den Einsatz einer Ressource  $R_i(t)$  kann demnach wie folgt bestimmt werden (8):

$$R_i(t_0) = \frac{1}{T - t_0} \cdot \sum_{T=t_0+1}^T \frac{M_i(t) - \tilde{M}_i(t)}{M_i(t)} \quad (8)$$

$$M_i(t) = P_{i,\max}(t) - P_{i,\min}(t)$$

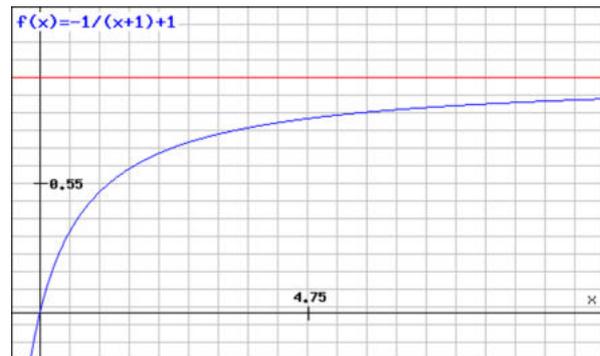
$$\tilde{M}_i(t) = \tilde{P}_{i,\max}(t) - \tilde{P}_{i,\min}(t)$$

Ausgehend davon, dass zu einem Zeitpunkt  $t_0$  eine Steuerungsentscheidung für eine Anlage  $i$  getroffen wurde, wird deren mittlere Auswirkung auf die Mobilität aller nachfolgenden Zeitpunkte (bei insgesamt  $T$  Zeitpunkten) ermittelt. Der Quotient ist dabei so definiert, dass ein Wert von null resultiert, wenn theoretische und tatsächliche Mobilität übereinstimmen, also keine Auswirkungen bestehen. Im anderen Extremfall, wenn die tatsächliche Mobilität null beträgt, also keine Steuerung mehr möglich ist, wird er Wert eins gebildet. Der Fehlerwert für das gesamte virtuelle Kraftwerk  $R(t)$  ergibt sich wiederum als Mittelwert aus allen  $M$  Teil-Anlagen (9):

$$R(t) = \frac{1}{M} \cdot \sum_{i=1}^M R_i(t) \quad (9)$$

3) *Optimierungsziel: CO<sub>2</sub>-Minimierung*: Ein weiteres sekundäres Optimierungsziel stellt die Reduktion der CO<sub>2</sub>-Emissionen des virtuellen Kraftwerks dar. Dieser Fehlerwert soll dazu führen, dass die Methodik – wenn möglich – eher Anlagen mit geringen CO<sub>2</sub>-Emissionen hochregelt und Anlagen mit hohen CO<sub>2</sub>-Emissionen runterregelt. Die Bestimmung des Fehlerwerts  $C_i(t)$  für eine Anlage  $i$  erfolgt durch die Berechnung der absolut produzierten Menge an CO<sub>2</sub> (in Gramm) (10):

$$C_i(t) = P_i(t) \cdot \Delta t \cdot C_i \quad (10)$$


 Abbildung 6. Normierungsfunktion: CO<sub>2</sub>-Emissionen.

Die Integration der Anlagenleistung (zur Bestimmung der erzeugten Energiemenge) erfolgt dabei näherungsweise durch Multiplikation mit dem Zeitintervall  $\Delta t$ . Ein anlagenspezifischer Wert  $C_i$  definiert die Umsetzung der Energiemenge in die CO<sub>2</sub>-Emissionen (in g/kWh).

Tabelle 1 zeigt die für die hier betrachteten Beispiel-Anlagen gewählten CO<sub>2</sub>-Werte [8]. Die Definition eines solchen Wertes für die PV- und WK-Anlage ist prinzipiell nicht nötig, da ihre CO<sub>2</sub>-Emissionen nur durch den Bau entstehen und nicht während ihrer Betriebsdauer (eine Steuerung hat also keinen Einfluss auf die Emissionen) wird hier aber dennoch vorgenommen, um die Wirkung dieses Optimierungsziels demonstrieren zu können. Die gesamten CO<sub>2</sub>-Emissionen des virtuellen Kraftwerks ergeben sich wiederum als Summe aller Teil-Anlagen (11):

$$\bar{C}(t) = \sum_{i=1}^M C_i(t) \quad (11)$$

Zur Normierung des Fehlerwerts wurde in dieser Arbeit eine Hyperbelfunktion auf die CO<sub>2</sub>-Emissionen angewendet.

Abbildung 6 zeigt den Verlauf einer möglichen Hyperbelfunktion (blau) sowie ihre Asymptote (rot), die für den ersten und vierten Quadranten gültig ist. Der abschließende Fehlerwert für die CO<sub>2</sub>-Emissionen  $C(t)$  ergibt sich wie folgt (12):

$$\bar{C}(t) = \begin{cases} f(\bar{C}(t)), & \bar{C}(t) \geq 0 \\ 0, & \bar{C}(t) < 0 \end{cases} \quad (12)$$

4) *Bestimmung des Gesamt-Fehlerwerts*: Die vom SA-Algorithmus tatsächlich verwendete Fehlerfunktion  $E(t)$  bildet sich als gewichtete Summe der zuvor definierten drei Fehlerwerte für die verschiedenen Optimierungsziele (13):

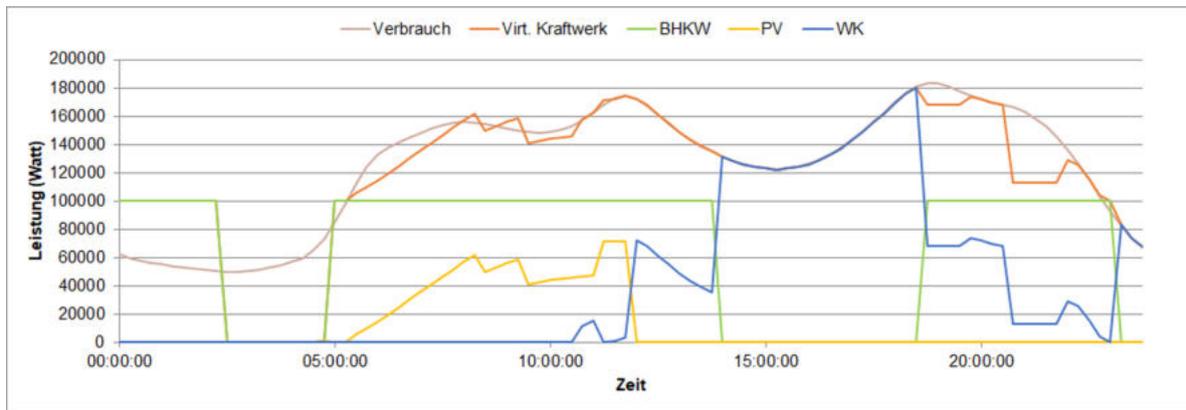


Abbildung 7. Validierung der Steuerungsmethodik.

$$E(t) = a \cdot A(t) + b \cdot R(t) + c \cdot C(t) \quad (13)$$

Über die Gewichtungsfaktoren  $a$ ,  $b$  und  $c$  können dabei - je nach Bedarf - die Prioritäten der einzelnen Optimierungsziele angepasst werden.

#### IV. VALIDIERUNG

Zur Validierung sind in Abbildung 7 der exemplarische Verlauf einer zu treffenden Verbraucherkurve, die angepasste Leistungskurve des virtuellen Kraftwerks sowie die Leistungskurven aller drei Teil-Anlagen gegeben. Sowohl die Verbraucherkurve als auch die Leistungskurven der PV- und WK-Anlage beziehen sich dabei auf eine Simulation der Anlagen für einen Tag mit gegebenen, realen Wetterverläufen an einem bestimmten Standort. Bei der Leistungskurve des BHKWs wurde dabei auf die zusätzliche Berücksichtigung einer thermischen Last verzichtet, da diese die Steuerungsmöglichkeiten des BHKWs und das daraus resultierende Ergebnis so stark beeinflussen würde, dass es nur durch die Erläuterung mehrerer, weiterer Lastkurven erklärbar wäre und dadurch die eigentliche Optimierungsabsicht in den Hintergrund gerät. Die Mindestlauf- und -kühlzeit wurde jeweils auf eine Stunde festgelegt. Für die WK-Anlage wurde beispielhaft eine Schrittweite von 1000 Watt angenommen. Die Gewichtung der einzelnen Optimierungskriterien wurde auf 70 % für den Regelenergiebedarf, 10 % für den Ressourceneinsatz und 20 % für die Minimierung der CO<sub>2</sub>-Emissionen festgelegt. Der Fokus für dieses Beispiel liegt also auf der Optimierung des Regelenergiebedarfs.

Wie der Abbildung zu entnehmen ist, schafft es die Leistungskurve des virtuellen Kraftwerks in Zeiten, zu denen überhaupt genügend Anlagenleistung vorhanden ist, die Verbraucherkurve gut zu treffen. Zu den Zeitpunkten, an denen im virtuellen Kraftwerk zu wenig Leistung vorhanden ist, regelt die Methodik alle Anlagen auf ihren maximal möglichen Leistungsertrag,

um der Verbraucherkurve so nah wie möglich zu kommen. Auffällig hierbei ist, dass ab ca. 12:00 Uhr, wenn aufgrund der Wetterlage zunehmend Leistung an der WK-Anlage abrufbar ist, die Methodik diese immer stärker bevorzugt bis die benötigte Leistung zeitweise sogar ausschließlich durch die WK-Anlage gedeckt werden kann. Dies lässt sich durch den vergleichsweise geringen CO<sub>2</sub>-Wert der WK-Anlage sowie der Optimierung auf geringe CO<sub>2</sub>-Emissionen erklären. Weiterhin fällt hier auf, dass die Methodik – um die WK-Anlage hochregeln zu können – zuerst schrittweise die PV-Anlage runterregelt und erst danach das BHKW ausschaltet. Gründe hierfür sind zum einen die nach einem Schaltvorgang eintretenden Einschränkungen der BHKW-Steuerung, die die Methodik durch die Optimierung des Ressourceneinsatzes so lange wie möglich zu verhindern versucht, und zum anderen die feineren Steuerungsmöglichkeiten der PV-Anlage im Vergleich zum BHKW.

#### V. FAZIT

Die vorliegende Arbeit demonstriert am Beispiel des Simulated Annealing Algorithmus den Nutzen von Verfahren der Künstlichen Intelligenz für die Steuerung von virtuellen Kraftwerken. Zur Verdeutlichung wurde eine Optimierung zur Begrenzung des Regelenergiebedarfs gewählt, um die Stromnetzbelastung verringern und so die Integration von Erneuerbaren Energien erleichtern zu können. Durch weitere, sekundäre Kriterien können dabei sowohl die Steuerbarkeit des virtuellen Kraftwerks als auch dessen CO<sub>2</sub>-Emissionen zusätzlich optimiert werden.

Durch einen beispielhaften Kurvenverlauf konnte gezeigt werden, dass die Steuerungsmethodik zu jedem Zeitpunkt in der Lage ist, die Verbraucherlast möglichst gut zu treffen und dabei gleichzeitig die CO<sub>2</sub>-Emissionen minimieren sowie die Anlagenflexibilität erhalten kann.

## LITERATURVERZEICHNIS

- [1] A. Sauter, J. Gerlach und O. Bringmann, *Simulative Analyse dezentraler Energieversorgungskonzepte*, Albstadt-Ebingen. Letzter Zugriff 01.07.2018.
- [2] *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung*, 28.09.2010. [Online], Available: , [Zugriff am 19 06 2018].
- [3] J. Unger und A. Hurtado, *Energie, Ökologie und Umvernuft*, Heidelberg: Springer-Verlag GmbH, 2013.
- [4] Bundesministerium für Wirtschaft und Energie, *Was ist eigentlich ein „Virutelles Kraftwerk“?*, Bundesministerium für Wirtschaft und Energie, 14.07.2015. [Online]. Available: , [Zugriff am 18 06 2018]. Letzter Zugriff 01.07.2018.
- [5] C. Märkel, *Photovoltaik Anlagen wirtschaftlich runterregeln*, DAA Deutsche Auftragsagentur GmbH, o. J., [Online]. Available: , [Zugriff am 11 06 2018].
- [6] Bundesverband WindEnergie, *Leistungsbegrenzung und -regelung*, Bundesverband WindEnergie, o. J.. [Online]. Available: , [Zugriff am 18 06 2018].
- [7] D. L. Poole, A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, 2. Hrsg., Cambridge: Cambridge University Press, 2017.
- [8] M. Günther, *Energieeffizienz durch Erneuerbare Energien*, Heidelberg: Springer-Verlag GmbH, 2015.



**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 BMWi geförderten Forschungsprojekt SADE - Simulative Analyse dezentraler Energieversorgungskonzepte.



**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 BMWi geförderten Forschungsprojekt SADE - Simulative Analyse dezentraler Energieversorgungskonzepte.



**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“.



# Aufbau und Evaluation eines FPGA-basierten Miniatur-DSP-Systems für professionelle Audioanwendungen

Christian Epe, Bernhard M. Rieß

**Zusammenfassung**—Dieses Paper betrachtet aktuelle Systeme zur digitalen Audiosignalverarbeitung hinsichtlich Ihrer Baugröße, Leistung und Kosten. Daraufhin wird ein FPGA-basiertes System entwickelt, welches trotz geringer Baugröße eine Audio-Performance erreicht, die bisher nur großen und vor allem hochpreisigen Systemen vorbehalten war. Hierzu erfolgt die Erstellung eines Konzeptes woraufhin ein Schaltplan entwickelt und als Leiterplatte mit SMD-Bauteilen umgesetzt wird. Um die Tauglichkeit zur Audiosignalverarbeitung zu demonstrieren werden IIR- und FIR-Filter sowie ein Delay-Effekt in VHDL realisiert und auf dem FPGA implementiert. Die Audio-Performance und die Signalverarbeitung werden abschließend elektrisch vermessen und die Ergebnisse dem Stand der Technik gegenübergestellt.

**Schlüsselwörter**—FPGA, DSP, ADC, DAC, High-Quality-Audio, Audio-Wandler, IIR, FIR

## I. EINLEITUNG

Die digitale Audiosignalverarbeitung hat die analoge Audiosignalverarbeitung sowohl in der Consumer- wie auch der professionellen Audio-Branche bereits weitgehend ersetzt. Klassisch werden in der digitalen Audiosignalverarbeitung digitale Signalprozessoren (DSP) eingesetzt, die dank ihrer Architektur bereits nützliche Funktionen zur Signalverarbeitung integriert haben. Eine interessante Alternative dazu stellen frei konfigurierbare Field-Programmable-Gate-Arrays (FPGA) dar, die sich durch ihre hohe Flexibilität und Skalierbarkeit auszeichnen. Ob der DSP oder das FPGA der bessere „Signalverarbeiter“ ist, kann nicht pauschal gesagt werden. Kapitel II soll hierüber mehr Aufschluss geben. Beide Ansätze, mit DSP wie auch mit FPGA, werden oft als System-on-a-Chip (SoC) Lösung mit integrierten Analog/Digital-(AD)- und Digital/Analog-(DA)-Wandlern angeboten. Damit ist der komplette Signalweg auf einem Baustein integriert. Nachteil dieser Lösungen ist die oft begrenzte Qualität der integrierten AD- und DA-Wandler wodurch das Gesamtsystem oftmals für professionelle Audioanwendungen ungeeignet ist.

Christian Epe, christian.epe@hs-duesseldorf.de, Fachbereich Medien, Hochschule Düsseldorf, Münsterstraße 156, 40476 Düsseldorf,  
Bernhard M. Rieß, bernhard.riess@hs-duesseldorf.de, Fachbereich Elektro- und Informationstechnik, Hochschule Düsseldorf, Münsterstraße 156, 40476 Düsseldorf.

Hauptziel dieser Arbeit war die Entwicklung eines neuen FPGA-basierten Systems zur Digitalen Signalverarbeitung (DSV-System) für Audiosignale inklusive AD- und DA-Wandlung, welches bei geringer Baugröße professionellen Audioanwendungen gerecht wird. Hierzu wird in Kapitel III zunächst der aktuelle Stand der Technik, insbesondere die in der professionellen Audiotechnik eingesetzten Wandler-Systeme betrachtet. Hieraus hervorgehend wird in Kapitel IV ein Hardware- und VHDL-Konzept vorgestellt, dessen Umsetzung in Kapitel V beschrieben wird. Für den Bau des neuen Systems werden hoch qualitative AD- und DA-Wandler sowie ein FPGA ausgesucht und auf einer Leiterplatte in SMD-Technik integriert. Um die Tauglichkeit zur Audiosignalverarbeitung zu demonstrieren, wurde ein „Vocal-Channel-Strip“ eines Mischpultes bestehend aus Trittschall-Filter, Equalizer und Delay-Effekt auf dem FPGA implementiert.

Kapitel VI verifiziert mit Hilfe elektrischer Messungen die Audio-Performance und die Signalverarbeitung des neuen Systems. In einem abschließenden Fazit in Kapitel VII wird es dem Stand der Technik gegenübergestellt.

## II. DSP vs. FPGA

Lange Zeit war der DSP die erste Wahl, wenn es darum ging Algorithmen der digitalen Signalverarbeitung in Systeme zu integrieren. Dank seines Aufbaus ist er in der Lage entsprechende Operationen in Echtzeit durchzuführen. Vergleichsgrößen der Prozessoren sind üblicherweise „Million-Instructions-Per-Second“ (MIPS), also wie viele Millionen Befehle in einer Sekunde ausgeführt werden können. Diese Angabe ist beim Vergleich verschiedener IC-Architekturen aber eher ungeeignet, da sich die Befehlssätze und deren Umsetzung auf einem IC sehr stark unterscheiden können. Im Fall von DSPs wird daher häufig die Anzahl der durchführbaren „Multiply-Accumulate“-Operationen pro Sekunde (MAC/s) betrachtet, also das gleichzeitige Multiplizieren und Addieren in einem Taktzyklus, da sie in vielen Algorithmen der digitalen Signalverarbeitung verwendet werden.

P. Ekas [1] veröffentlichte 2007 einen anschaulichen Vergleich des Performance-Anstiegs von Texas Instruments DSPs zu Altera FPGAs über einen Zeitraum von



mehr als 15 Jahren. Er beobachtete, dass die FPGA-Performance, hinsichtlich der MAC-Operationen, über die Zeit wesentlich stärker gestiegen war, als die DSP-Performance. M. Shirvaikr und T. Bushnaq [2] zeigten 2015 einen Performance-Vergleich von DSP und FPGA an einem konkreten Anwendungsbeispiel. Sie ließen einen DSP und ein FPGA unter Versuchsbedingungen eine Wavelet-Transformation durchführen und stellten die Ergebnisse sowie die benötigten System-Ressourcen gegenüber. Die Ergebnisse zeigen, dass das FPGA in den einschlägigen Bewertungskriterien überlegen ist. Er schaffte die ihm gestellte Aufgabe in 5-mal kürzerer Zeit bei einem Sechstel des Energiebedarfs. Jedoch ist die Programmierung beim FPGA wesentlich aufwändiger. Es sind fast viermal mehr Code-Zeilen nötig, um die Wavelet-Transformation auf einem FPGA umzusetzen. Grund für die unterschiedliche Komplexität bei der Programmierung sowie der relativ großen Spanne beim Energiebedarf sind die stark unterschiedlichen Hardwarearchitekturen von DSP und FPGA.

FPGAs bestehen aus einer großen Ansammlung von einfachen Logikbausteinen und werden erst durch ihre Konfiguration, bzw. durch entsprechende Verschaltung dieser Bausteine zu einem Bauteil mit einer bestimmten Funktion. Hierdurch ergibt sich eine sehr hohe Flexibilität, da das FPGA somit zum maßgeschneiderten IC für die ihm gestellten Operationen konfiguriert werden kann. Die Programmierung erfolgt mittels einer Hardware-Beschreibungssprache HDL (Hardware-Description-Language). Je nachdem wie ein Algorithmus auf Logik-Bausteine aufgeteilt werden muss, können sich kritische Pfade und damit zu hohe Signallaufzeiten ergeben. Sie stellen die eigentliche Grenze bei der Umsetzung von Algorithmen dar.

Der DSP hingegen ist bereits eine konfigurierte Hardware mit festgelegter Funktion. Gewisse Algorithmen der digitalen Signalverarbeitung sind bereits integriert, z.B. Filter-Funktionen. Speicher und andere vorkonfigurierte Teilkomponenten werden über eine Art Betriebssystem geregelt. Somit ist er zwar schneller einsatzbereit, aber dafür weniger flexibel. Zudem beansprucht er auch für ungenutzte Teilkomponenten Energie. Die Programmierung erfolgt mit der „höheren“ Programmiersprache C/C# und ist im Gegensatz zu HDL wesentlich anschaulicher und einfacher umzusetzen.

Ein weiteres und nicht unerhebliches Vergleichskriterium zwischen DSP und FPGA wurde 2011 von M. Yasir [4] untersucht. Er veröffentlichte einen Kosten-/Performance-Vergleich, aus dem hervorgeht, dass ab einer geforderten Leistung von 300 MMAC/s das FPGA günstiger als der DSP ist.

Die bereits erwähnte, aufwändigere Programmierung eines FPGA verlängert die Entwicklungszeit. In einem von der Firma Altera [4] durchgeführten Vergleich dauerte die Entwicklung eines FPGA-basierten

DSP-Systems zwar länger und führte damit zu höheren Entwicklungskosten, jedoch sollen die Kosten für Wartungsarbeiten bzw. nötige Updates zur Ausbesserung von Software-Fehlern für ein FPGA-basiertes System wesentlich geringer sein als bei einem DSP-basierten System. Grund hierfür ist wieder die grundlegend unterschiedliche Arbeitsweise beider IC-Architekturen. Der DSP arbeitet seine Aufgaben in „Threads“ ab, welche durch eine Art Betriebssystem auf seine Systemressourcen verteilt werden. Hierbei kann es zu vielen unterschiedlichen Kombinationen und Zusammenspielen der Komponenten kommen, die nicht alle während der Entwicklungsphase berücksichtigt oder getestet werden können. Diese Tatsache führt zu unvorhersehbaren Fehlern und damit zu höheren Kosten für vermehrte Wartungsarbeiten und Updates. Beim FPGA werden die einzelnen Aufgaben mit dezidiert Hardware bearbeitet. Dabei steht jeder Aufgabe ihr eigener Speicher, DSP-Leistung und Input-/Output-Schnittstelle zur Verfügung. Zudem stehen bei der FPGA-Programmierung umfangreiche Testmöglichkeiten zur Verfügung, mit denen alle Teilbereiche individuell getestet werden können.

Zusammenfassend kann gesagt werden, dass der Einsatz von FPGAs für leistungsfähige DSP-Systeme unter Berücksichtigung der Bewertungskriterien, wie Performance, Kosten und Flexibilität zurzeit die bessere Wahl ist. Jedoch muss man sich bewusst sein, dass die Programmierung und Entwicklung komplexer ist als bei einem DSP. FPGA-Hersteller haben diese Tatsache erkannt und bieten vermehrt Tools zur einfacheren Umsetzung von DSP-Algorithmen auf dem FPGA an.

### III. STAND DER TECHNIK

In diesem Kapitel sollen aus dem aktuellen Stand der Technik Anforderungen bzw. eine Leistungsbeschreibung für das zu entwickelnde DSV-System gewonnen werden. Hierzu werden zunächst Kriterien von AD- und DA-Wandlern hinsichtlich der Audiotechnik betrachtet und anschließend auf dem Markt erhältliche Chips und Systeme vorgestellt (Abschnitt III.A).

Die in Kapitel II geführte Diskussion führte zur Entscheidung, einen FPGA als eigentlichen Signalverarbeiter in das System zu integrieren. Daher folgt in Abschnitt III.B die Betrachtung verschiedener einsetzbarer FPGAs.

#### A. AD- und DA-Wandler in der professionellen Audio-Technik

In der professionellen Audiotechnik stellt die Tonproduktion mitunter die höchsten Ansprüche an die AD- und DA-Wandler hinsichtlich Signal-Rausch-Abstand (SNR), Bittiefe, Impulstreue und vor allem Klangneutralität. Gängige Abtastraten in der Produktion betragen 44,1 kHz (CD-Qualität) oder 48 kHz (Film und Fernsehen). Auch 96 kHz oder 192 kHz werden von einem Großteil erhältlicher Produkte, wie Wandler

oder Audio-Interfaces unterstützt. In der Realität werden sie jedoch recht selten genutzt, da die erzeugten Datenmengen für die meisten Nutzer den klanglichen Mehrwert nicht rechtfertigen. Im oberen Preissegment platzieren wenige Hersteller bereits 384 kHz-fähige Wandler. Für die Tonproduktion gibt es jedoch momentan keine digitale Audioworkstation (DAW), die diese Daten entsprechend weiterverarbeiten kann. Für ein neues Produkt sind Abtastraten von bis zu 192 kHz also sinnvoll. Die Unterstützung höherer Abtastraten wäre zum jetzigen Zeitpunkt nicht die Priorität.

Ein hoher SNR-Wert ist vor allem beim AD-Wandler wünschenswert, da bei der Tonproduktion üblicherweise mehrere aufgenommene und vor allem dynamische Signale miteinander gemischt werden. Dabei summiert sich das bei der Wandlung entstandene Quantisierungsrauschen der Einzelsignale. Die Ausspielung bzw. Wiedergabe der Tonproduktion geschieht in den meisten Fällen mit wesentlich weniger Kanälen und verringerter Dynamik (Dynamikkompression), so dass der SNR beim DA-Wandler geringer sein kann.

Die in der Tonproduktion verwendeten Bittiefen für Audiorohdaten betragen heute fast durchgehend 24 bit. Höhere Bitraten (bis zu 32 bit) werden von DAWs nur zur Verrechnung von Effekten genutzt, um die Genauigkeit von Rechenoperationen zu erhöhen und damit zusätzlich entstehendes Rauschen durch Rundungsfehler gering zu halten. Für einen Großteil der Wiedergabemedien, wie CD oder Film, wird die Bittiefe nach der Produktion auf 16 bit verringert. Bei Datenreduktionsverfahren, wie beispielsweise MP3 oder AAC kann die Bittiefe zeitweise noch geringer sein. Die 24-Bit-Obergrenze macht aus Sicht der Wandlung insofern Sinn, da aktuelle Wandler-Chips noch 15-20 dB unterhalb des rechnerisch möglichen SNRs von 146 dB liegen (Gleichung 1).

$$SNR = 6,02 \cdot bit + 1,76 \text{ [dB]} \quad (1)$$

Höhere Bittiefen bis zu 32 bit (I<sup>2</sup>S-Datenstrom: 32 bit pro Kanal, bei 2 Kanälen) wären zum jetzigen Zeitpunkt somit nicht unbedingt die Priorität.

Die Impulstreue wird vor allem vor der AD- und nach der DA-Wandlung erzeugt bzw. beeinflusst. Analoge Tiefpassfilter zur Einhaltung der Nyquist-Frequenz sind hierbei der bestimmende Faktor. Zu steile Filter erzeugen ungewollte, frequenzabhängige Phasenverzerrungen, welche das Impulsverhalten verschlechtern und im schlimmsten Fall eine wahrnehmbare, zeitliche Signaldispersion verursachen können. Oversampling verschiebt die Nyquist-Frequenz in Bereiche außerhalb des Hörbereichs. Je höher die Nyquist-Frequenz, desto flacher kann die Filterflanke eines analogen Anti-Aliasing-Filter sein. Somit kann im Hörbereich bis 20 kHz die Phasenverzerrung gering gehalten werden. Daher ist ein möglichst hoher Oversampling-Faktor wünschenswert.

Selbst Consumer-Wandler sind heute sehr klangneutral. Nicht-Linearitäten bei der Wandlung lassen sich messtechnisch als Klirrvverzerrungen (Total Harmonic Distortion, THD) oder Klirrfaktor messen. Je nach Frequenz werden Klirrvverzerrungen für den Menschen hörbar. Im Bassbereich gilt ein Klirrfaktor von 0,05 (5 %) als hörbar, bei 1 kHz bis 4 kHz allerdings bereits ab einem Wert von 0,005 (0,5 %). Hifi-Verstärker werden heute beispielsweise bis maximal 0,1 % ausgelegt und liegen damit weit unter der menschlichen Wahrnehmungsschwelle. Neben der Angabe in Prozent ist die Angabe des Dämpfungsmaßes  $L_k$  in Dezibel gebräuchlich.

Zur Auswahl eines AD- und eines DA-Wandlers wurden ICs einschlägiger Hersteller wie Cyrrus Logic, Analog Devices, Texas Instruments und Asahi Kasei Microdevices (AKM) recherchiert und hinsichtlich der oben genannten Kriterien verglichen. Es wurden ICs ausgesucht die mindestens 24 bit / 192 kHz fähig sind. Hinsichtlich der SNR- und THD-Werte muss davon ausgegangen werden, dass die Angaben der Hersteller in der Realität nicht ganz erreicht werden können. Nötige analoge Ein- bzw. Ausgangsstufen verursachen ihrerseits Rauschen und tragen zur Verschlechterung der Werte bei. Zudem werden die Werte unter teils fragwürdigen Idealbedingungen ermittelt.

Auch sogenannte Codec-Wandler, kurz Codecs, wurden in Betracht gezogen. Bei ihnen handelt es sich um ICs, die AD- und DA-Wandler in einem Chip vereinen. Der Vorteil liegt im geringeren Platzbedarf auf einer Platine. Die von den Herstellern angegebenen SNR-Werte der recherchierten Codecs von maximal 114 dB nutzen die für das DSV-System geplante Bittiefe von 24 bit (entspricht  $\sim 146$  dB) zu wenig aus. Daher wurde der Einsatz von Codec-Wandlern nicht weiterverfolgt.

Die AD-Wandler von AKM waren denen der anderen Hersteller in allen Parametern überlegen. Sie bieten eine Bittiefe von bis 32 bit und eine maximale Abtastrate von 768 kHz. Zudem wiesen die Wandler die besten THD-Werte auf. Für das Projekt wurde der AD-Wandler AK5397EQ gewählt der mit einem SNR von 127 dB im Stereobetrieb das geringste Rauschen verspricht.

Auch bei den DA-Wandlern kommt der Beste hinsichtlich des SNR-Wertes von AKM. Der AK4497SEQ bietet ebenfalls eine Bittiefe von 32 bit und eine maximale Abtastrate von 768 kHz und ist somit maximal kompatibel zum zuvor gewählten AD-Wandler. Die Wahl fiel aus Kostengründen jedoch auf den AK4495SEQ, welcher bei einem Zehntel des Preises lediglich 3 dB mehr Rauschen bei der Wandlung (SNR = 123 dB) verursacht. Wie in diesem Abschnitt bereits erwähnt wurde, ist ein höheres Rauschen auf der DA-Seite des geplanten Systems vertretbar.



Tabelle I

GRÖSSE UND KOSTENVERGLEICH: KATEGORIE 1 - KOMPAKTE  
DSV-SYSTEME; KATEGORIE 2 – EVALUATIONSBOARDS;  
KATEGORIE 3 – PROF. AUDIO-WANDLER.

	Bit-Tiefe	Abtast-rate (kHz)	Fläche (cm <sup>2</sup> )	Preis (€)
<b>Kategorie 1 - Kompakte DSV-Systeme</b>				
freeDSP	24	48	80	70
Monacor MDSP-24KIT	24	48	84,48	189
MiniDSP HD	24	96	110,67	207
<b>Kategorie 2 - Evaluationboards</b>				
EVAL-21489-EZLITE	24	192	193,75	600
Atlys Spartan-6 FPGA Trainer Board	18	48	n.b.	462
Nexys Video Artix-7 FPGA Trainer Board	24	96	n.b.	462
<b>Kategorie 3 - Professionelle Audio-Wandler</b>				
DAD AX32	24	384	19", 2 HE	7.119
RME Fireface 802	24	192	19", 1 HE	1.261
Antelope Eclipse 384	24	384	19", 2 HE	7.595

Tabelle II

PERFORMANCE-VERGLEICH: KATEGORIE 1 - KOMPACTE  
DSV-SYSTEME; KATEGORIE 2 – EVALUATIONSBOARDS;  
KATEGORIE 3 – PROF. AUDIO-WANDLER.

	SNR ADC (dB, A-w.)	THDN ADC (dB)	SNR DAC (dB, A-w.)	THDN DAC (dB)
<b>Kategorie 1 - Kompakte DSV-Systeme</b>				
freeDSP	100	-83	104	-90
Monacor MDSP-24KIT	n.b.	n.b.	n.b.	n.b.
MiniDSP HD	102	-94	102	-100
<b>Kategorie 2 - Evaluationboards</b>				
EVAL-21489-EZLITE	112	-94	107	-94
Atlys Spartan-6 FPGA Trainer Board	90	-74	89	-74
Nexys Video Artix-7 FPGA Trainer Board	98	-90	101	-90
<b>Kategorie 3 - Professionelle Audio-Wandler</b>				
DAD AX32	123	-120	128	-115
RME Fireface 802	116	-104	118	-104
Antelope Eclipse 384	123	-108	129	-108

### B. Vergleichbare Systeme zur Digitalen Audiosignalverarbeitung

Ein System zur Digitalen Audiosignalverarbeitung (DSV), ob mit Digitalem Signalprozessor oder FPGA realisiert, lässt sich kaum eins zu eins vergleichen. Häufig unterscheiden sie sich in der Leistung, Anzahl der verfügbaren Kanäle und Systemparameter wie Bittiefe und Abtast-rate. Kompakte Systeme die einen digitalen Signalprozessor verwenden sind wesentlich häufiger zu finden, als FPGA-basierte Systeme. Bei der Recherche nach FPGA-basierten DSV-Projekten ist lediglich ein Projekt gefunden worden, welches ein Xilinx Spartan 3 FPGA mit einem Codec von Texas Instrument (TLV320AIC23B) [5] verwendet. Der Codec erfüllt mit gerade 98 dB SNR und einer maximalen Abtast-rate von 48 kHz aber nicht die gestellten Anforderungen. Das geplante FPGA-basierte DSV-System soll den hohen Anforderung einer Studioumgebung (siehe Abschnitt III.A) gerecht werden und zudem möglichst kompakt sein. Um den Stand der Technik zu definieren, soll der Markt in drei Kategorien 1 bis 3 unterteilt und verglichen werden (vgl. Tabelle I und Tabelle II).

1) *Kompakte DSV-Systeme:* Kategorie 1 betrachtet drei kompakte DSV-Systeme deren Platinen in der Größe von 80 cm<sup>2</sup>-111 cm<sup>2</sup> liegen. Zwei sind mit einem SoC-DSP und einer mit einem vollwertigen DSP mit Codec-Wandler aufgebaut. Die SoC-

Varianten werden mit einer grafischen Oberfläche programmiert, deren Handhabung sehr einfach aber in der Leistungsfähigkeit hinsichtlich implementierbarer Algorithmen sehr eingeschränkt ist. Zudem sind die nur durchschnittlichen SNR-Werte von 100 dB (AD-Wandler) und die maximale Abtast-rate von nur 48 kHz für hochqualitative Studioanwendungen unzureichend. Zwar bietet das dritte System mit einem vollwertigen DSP mehr Rechenleistung und eine doppelt so hohe Samplerate von 96 kHz, jedoch werden die SNR-Werte ebenfalls mit nur 102 dB angegeben. Ein Vorteil all dieser Systeme sind die geringen Preise von 70 bis 207 €.

2) *Evaluationboards:* Kategorie 2 betrachtet drei Evaluationboards von denen eines auf einem digitalen Signalprozessor und zwei auf einem FPGA basieren. Mit diesem Vergleich soll der Überlegung nachgegangen werden, ob ein Evaluationboard mit einem IC für digitale Signalverarbeitung für professionelle Audioanwendungen verwendbar ist. Grundsätzlich dienen Evaluationboards Entwicklern dazu den Chip zu testen oder Programmcode parallel zur Hardware-Entwicklung zu schreiben. Dazu verfügen sie meistens über Standardperipherien wie Taster, LEDs oder, wie in diesem Fall, einem Audio-Codec, um Audiofunktionen und Algorithmen zu testen. Lediglich eines der betrachteten Systeme bietet eine Bittiefe und Abtast-rate im Bereich professioneller Audiotechnik. Der SNR liegt mit 107 dB etwas über dem Mittelfeld, jedoch

noch nicht in der professionellen Liga. Ein weiterer Nachteil der Evaluationboards sind die größeren Ausmaße gegenüber den kompakten DSV-Systemen aus Kategorie 1. Zudem werden sie nur in sehr geringen Stückzahlen produziert und sind mit bis zu 600 € sehr viel teurer.

3) *Professionelle Audio-Wandler in der Studiotchnik*: In der Kategorie 3 werden drei der zurzeit besten auf dem Markt erhältlichen Produkte im Bereich Audiowandler vorgestellt. Diese Betrachtung wird lediglich mit Hinblick auf die Audio-Parameter durchgeführt. Da es sich um reine Wandler im 19"-Format handelt, ist ein Größenvergleich nicht sinnvoll. Auch die Preise liegen weit über denen der Produkte der anderen Vergleichskategorien. Professionelle Wandler in der Audioproduktion weisen demnach eine Mindestabtastrate von 192 kHz und einem SNR von 116 dB und mehr auf.

Das Ziel dieser Arbeit soll sein, die Audioperformance der Kategorie 3 mit den Abmessungen und dem Preisbereich der Kategorie 1 zu vereinen. Demnach soll ein FPGA-basiertes DSV-System folgende Mindestanforderungen erfüllen:

- Bittiefe mindestens 24 bit
- Abtastrate mindestens 192 kHz
- SNR größer 116 dB
- Abmessungen kleiner 110 cm<sup>2</sup>
- Materialkosten geringer als 200 €

### C. Auswahl eines FPGA

Ein FPGA besteht hauptsächlich aus einer Gruppe von Basiszellen, den sogenannten Configurable Logic Blocks (CLB), programmierbaren Verdrahtungselementen und Input-/Output-Anschlüssen. Hinzukommen je nach Typ noch Zusatzelemente wie beispielsweise Speicherblöcke, Taktgeneratoren und eine entsprechende Programmierschnittstelle. Bei der Entwicklung von FPGA-basierten Systemen wird im Normalfall für eine geplante Hardware- bzw. Funktionsbeschreibung ein passender Chip mit entsprechender Anzahl an Logik-Gattern ausgewählt. Der Anspruch an professionelle Audiosignalverarbeitung gilt vor allem an die Wandler. Das FPGA muss nur entsprechende Ressourcen, wie Speicher, Logik-Gatter und Clock-Signale bereitstellen. Das geplante System stellt daher zunächst folgende Anforderungen an ein FPGA:

- Es müssen genügend I/O-Ports vorhanden sein. Im Fall der in Abschnitt III.A gewählten Wandler sind das mindestens 25 Ports. Hinzu kommen Ports für Taster und LED-Anzeigen.
- Das FPGA muss die nötigen Clock-Signale für AD- und DA-Wandler erzeugen können. Im Fall der gewählten Wandler bedeutet das bei einer maximalen Abtastrate von 192 kHz demnach ein Masterclock-Signal von mindestens 24,5 MHz.

- Das DSV-System soll mit verschiedenen Algorithmen der digitalen Audiosignalverarbeitung programmiert werden können. Die Anzahl der Gatter kann je nach Aufgabenstellung und Prioritätensetzung, wie das Verhältnis von Geschwindigkeit und benötigter Chip-Fläche, unterschiedlich sein. Das sollte in einem Hardwarekonzept derart berücksichtigt werden, dass man den FPGA bei Bedarf gegen ein anderes Modell austauschen kann.

Die Firma Xilinx hat mit Abstand den größten Anteil am FPGA-Markt (49 % [6] Stand 2015). Spartan-FPGAs werden von Xilinx als kostenoptimierte Chips beworben und passen daher gut zu der in Abschnitt III.B aufgestellten Anforderung an einen Systempreis von weniger als 200 €. Zudem sind auf dem Markt einige Spartan-6-Entwicklungsplattformen zu einem moderaten Preis von ca. 100 € erhältlich, die für ein Prototyping genutzt werden können. Andere Xilinx-FPGA-Serien wie Virtex und Kintex bieten FPGAs mit kleinerer Fertigungstechnik (Geschwindigkeitsgewinn, geringere Leistungsaufnahme), werden aber aufgrund des Preises und vor allem wegen der Preise für ein Entwicklungsboard von mehreren tausend Euro verworfen. Daher fällt die Wahl auf einen Spartan-6. Der Spartan-6 ist in 45 nm-Prozesstechnologie gefertigt und verfügt über eine hohe Anzahl an I/O-Ports. Die Programmierung erfolgt über die kostenlose Software *ISE Design Suite* ©Xilinx. Das Spartan-6-FPGA ist mit unterschiedlicher Anzahl an Logik-Blöcken erhältlich. Eine Besonderheit stellen die im Spartan-6 verfügbaren DSP-Slices dar. Sie dienen dazu Algorithmen der digitalen Audiosignalverarbeitung, speziell MAC-Operationen durchzuführen.

## IV. KONZEPT

### A. Hardwarekonzept

Zur Fertigung des Systems sollen die Möglichkeiten der Hochschule genutzt werden. Hier ist das Verlöten von SMD-Bauteilen möglich, soweit Anschlüsse außen am Chip zur Verfügung stehen. Unter Berücksichtigung dieser Rahmenbedingungen wurde zunächst ein FPGA-Typ ausgesucht, welcher, ähnlich einem Entwicklerboard, bereits auf einer Platine verlötet erhältlich ist. Zudem sollen die Ausmaße kompakt und vor allem bezahlbar sein. Aufbauend darauf soll die FPGA-Platine um eine Tochterplatine ergänzt werden, auf dem sich AD- und DA-Wandler befinden. Das bietet den Vorteil, dass das FPGA auch wie in Abschnitt III.C gefordert, ausgetauscht werden kann und die Rechenkapazität des Systems damit erweiterbar bleibt. Nach ausgiebiger Recherche über Entwicklerboards mit einem Xilinx Spartan-FPGA wurden kompakte Systeme gefunden und verglichen. Die Wahl fiel dabei auf das Produkt XuLA2-LX9 bzw. -LX25 (Abbildung 1) der Firma XESS [8].



Abbildung 1. XESS XuLA2-LX-25 [8].

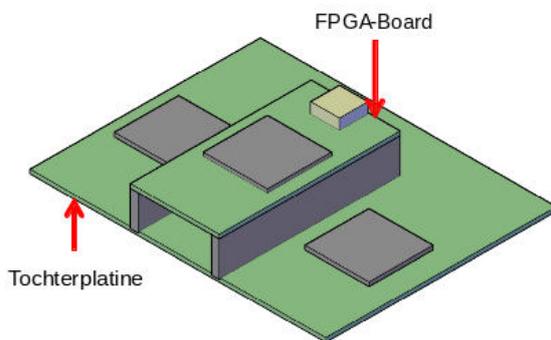


Abbildung 2. Konzept des mechanischen Aufbaus: XuLA2 und Tochter-Platine.

Die Boards der Firma XESS sind von ihren Ausmaßen sehr gut geeignet und verfügen über eine Mini-USB-Schnittstelle. Zudem sind sehr viele I/O-Pins direkt am Rand der Platine erreichbar. Der Unterschied zwischen dem LX9 und dem LX25 liegt hauptsächlich in der Anzahl der verfügbaren FPGA-Logik-Blöcke. Für eine Skalierbarkeit des Systems wäre der Austausch aufgrund der gleichen Pin-Belegung von LX9 und LX25 später ohne weiteres möglich.

Für das neue System soll der XuLA2 um hochwertige analoge und audiotaugliche Ein- und Ausgänge erweitert werden. Dabei soll eine Tochterplatine unter das XuLA2-Board passen (Abbildung 2). Hierzu sollen als AD-Wandler der AK5397EQ und als DA-Wandler der Typ AK4495SEQ eingesetzt werden.

Die Wandler-Chips müssen mit einer grundlegenden Beschaltung von Spannungsversorgung und analoger Ein-/Ausgangsschaltung versehen werden, bevor sie schließlich per I<sup>2</sup>S- und entsprechender Konfigurationsschnittstelle an das FPGA-Board angebunden werden können. Es sollen Taster und LEDs vorgesehen werden, um später die Umschaltung von Audioparametern kontrollieren zu können. Um die Belegung von FPGA-Ports gering zu halten, werden die LEDs über einen LED-Treiber implementiert, der über eine serielle Schnittstelle gesteuert werden kann. Die Audioanschlüsse sollen symmetrisch ausgelegt sein, so dass je nach Einsatzgebiet später XLR- oder Klinken-

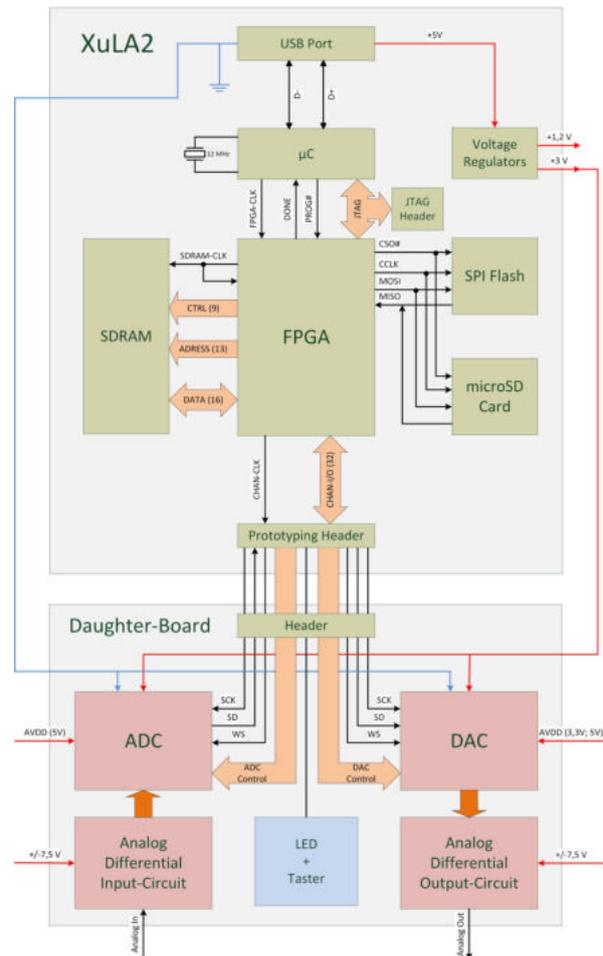


Abbildung 3. Blockschaltbild des Prototypen.

anschlüsse angebracht werden können. Hierfür sollen auf der Tochterplatine zunächst Pins bereitgestellt werden, an die später Buchsen angesteckt werden können.

Im unteren Teil des Blockschaltbildes (Abbildung 3) ist der schematische Aufbau der geplanten Tochterplatine zu erkennen. Auf ihr befinden sich die Wandler mit den zugehörigen analogen Ein- und Ausgangsstufen, ein LED-Treiber mit 8 LEDs und 5 Taster. Elemente zur Erzeugung zusätzlicher Versorgungsspannungen sind der Übersicht halber nicht mit eingezeichnet. Die am Prototyping-Header anliegende 3,3 V-Spannung versorgt, neben einem LED-Treiber und den Tastern (Pullup-Taster), digitale Abschnitte von AD- und DA-Wandler.

Beide Wandler benötigen zudem 3,3 V und 5 V für analoge IC-Abschnitte. Für eine gute Audio-Performance müssen diese Spannungen separat erzeugt werden und besonders sauber, also rauscharm sein. Die Ein- und Ausgangsverstärker verlangen nach einer symmetrischen Spannungsversorgung. Um Rundfunktaugliche Pegel (+6 dBu, plus Headroom) verarbeiten zu können, wird diese Spannung auf mindestens  $\pm 7,5$  V festgelegt. Als Grundlage der Stromversorgung können zwei 9 V-Batterien oder bei Bedarf ein Netzteil

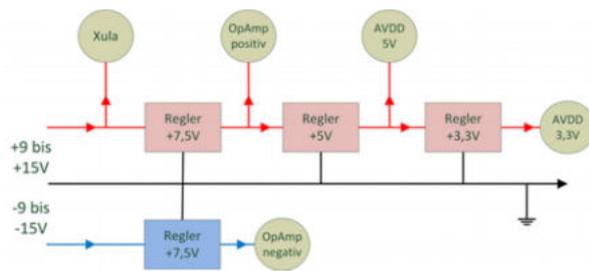


Abbildung 4. Konzept der Spannungsversorgung.

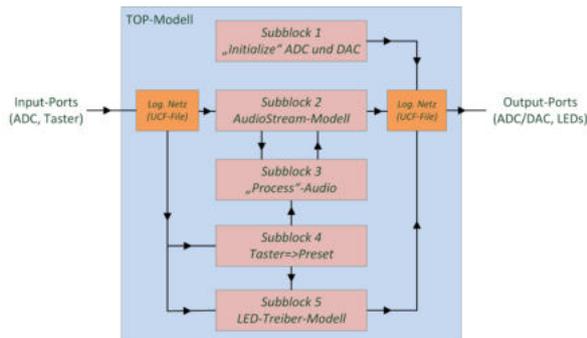


Abbildung 5. Konzept der Hardware-Beschreibung.

mit einer Spannung von bis zu  $\pm 15\text{ V}$  dienen. Spannungsregler sollen daraus die notwendigen Spannungen erzeugen. Die Verschaltung der Regler wird in Abbildung 4 gezeigt.

### B. Konzept der Hardware-Beschreibung (VHDL)

Als Basis für die Hardware-Beschreibung (VHDL-Modell) dient ein hierarchisch aufgebautes Top-Modell, welches nötige Funktionen des DSV-Systems in Subblocks bereitstellt (Abbildung 5). Das Top-Modell stellt zudem mit Hilfe eines User-Constraint-Files (UCF-File) die Verknüpfung zwischen FPGA-Ports und den Ports des VHDL-Modells her. Gemäß Abbildung 5 nimmt Subblock 1 nach dem Start des Systems zunächst die Konfiguration von AD- und DA-Wandler vor.

Nach der Initialisierung startet Subblock 2 den Audio-Datenstrom. Der Subblock erzeugt die für die I<sup>2</sup>S-Schnittstellen nötigen Clock-Signale und empfängt bzw. sendet den seriellen Datenstrom vom AD-Wandler bzw. zum DA-Wandler. Aus dem seriellen Datenstrom werden die Audiowerte (Samples, 24- bis 32-Bit-Wortbreite) für den linken und rechten Kanal mittels Schieberegister gewonnen und dem Subblock 3 zur Weiterverarbeitung zur Verfügung gestellt.

Im Subblock 3 „Process“-Audio erfolgt die eigentliche Signalverarbeitung anhand der nun vorliegenden Audiowerte. Das bearbeitete Sample wird anschließend dem Subblock 2 wieder zurückgegeben.

Der Subblock 4 verarbeitet die Signale der Taster. Hierbei wird bei der Taster-Betätigung jeweils ein Bit in einem 11-Bit Daten-Preset gesetzt. Das Preset wird

anschließend an den Subblock 5 und an den Subblock 3 gesendet. Letzteres nimmt anhand des Presets eine entsprechende definierte Signalverarbeitung vor.

Subblock 5 erzeugt eine dem Preset entsprechende LED-Anzeige. Hierzu wird ein serieller Datenstrom erzeugt und an einen LED-Treiberbaustein gesendet. Der AD-Wandler verfügt über die Möglichkeit eine Übersteuerung über einen Input-Port an das FPGA zu melden. Diese Information wird vom Input-Port direkt an den Subblock 5 übergeben, um eine Übersteuerung direkt auf den LEDs anzeigen zu können.

### C. Konzept der Audiosignalverarbeitung

Innerhalb des Subblocks 3 „Process“-Audio aus Abschnitt IV.B soll die eigentliche Audiosignalverarbeitung der vom AD-Wandler eingehenden Audio-Samples geschehen, bevor sie zurück an den DA-Wandler bzw. den Ausgang des DSV-Systems geleitet werden. Hierzu bieten sich eine Vielzahl von Bearbeitungsmöglichkeiten an, wie stationäre oder adaptive Filterung, Kompression oder Erzeugung von Echo oder Hall. Da das Hardware-Konzept 2-kanalige AD- und DA-Wandler vorsieht, wäre auch das Mischen oder Subtrahieren von Signalen möglich. Es ließe sich beispielsweise eine Matrizierung von Mid-/Side-Signalen auf ein Stereosignal vornehmen. Ebenso wäre das Erzeugen von Stereosignalen aus einem 1-kanaligen Signal denkbar oder die Umsetzung einer Frequenzweiche für Lautsprechersysteme. In Frage kommt auch das Erzeugen von 2-kanaligem Hall oder Echo, um den Raumeindruck einer Stimme oder eines Instruments zu verbreitern. In der Live-Beschallung sind nach wie vor viele analoge Mischpulte anzutreffen, die mit Hilfe des geplanten Prototyps mit Funktionen aufgewertet werden können, die ansonsten nur teuren Digitalpulten vorbehalten sind. Dem Konzept der Audiosignalverarbeitung in dieser Arbeit soll daher ein Mischpultkanalzug als Vorlage dienen. Durch die Umsetzung eines solchen Signalweges zur Bearbeitung von Sprache oder Gesang, können in umfassender Weise die Möglichkeiten des DSV-Systems aufgezeigt werden.

Analoge Mischpulte mittlerer Größe (16-24-Kanäle) verfügen zur Signalbearbeitung pro Kanal lediglich über diverse Equalizer-Funktionen. Hinzu kommen noch Funktionen des Signalaroutings zu einem Effektgerät, zu Gruppen-Fadern oder zu einem Summenausgang, ein Panorama-Regler, ein Kanal-Fader und ein Mute-Schalter, um das Signal stumm zu schalten. Wünschenswert wären eine detaillierte Klangbearbeitung und ein Echo- oder Hall-Effekt.

Abbildung 6 zeigt ein Kanalzug-Schema, welches in diesem DSV-System implementiert werden soll. Das Eingangssignal durchläuft zunächst einen zuschaltbaren 100 Hz-Hochpassfilter zur Entfernung von Trittschallgeräuschen, welche häufig bei in der Hand gehaltenen Mikrofonen auftreten.

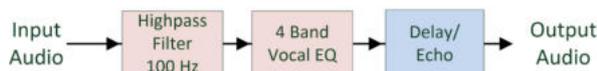


Abbildung 6. Konzept der Audiosignalverarbeitung.

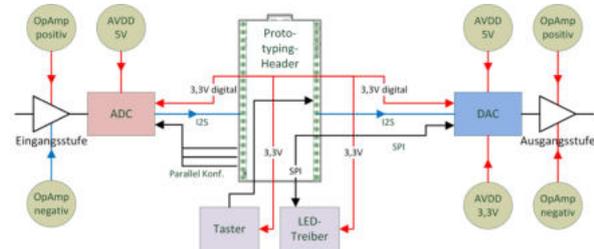


Abbildung 7. Blockschaltbild des Tochterplatine.

Im Anschluss folgt die klangliche Bearbeitung mittels eines 4-Band-Vocal-Equalizers. Im Gegensatz zu einem Equalizer eines analogen Mischpultes kann im DSV-System wesentlich detaillierter gefiltert werden. Hier sollen typische Presets abgerufen werden, die die Sprachverständlichkeit erhöhen und bei Bedarf den Pegel des Grundtonbereiches der menschlichen Stimme anheben oder absenken können. Im Bereich um 200 Hz liegt der Grundtonbereich der menschlichen Stimme. Mit einer Pegelanhebung in diesem Bereich lässt sich einer Stimme mehr Fülle und Wärme hinzufügen. Im Bereich 800 Hz bis 1000 Hz befinden sich oft nasal klingende Anteile, die herausgefiltert werden sollen. Zur Erhöhung der Präsenz und Sprachverständlichkeit sind zusätzlich zwei Anhebungen im Bereich von 1,5 kHz bis 2 kHz und 3,5 kHz bis 5 kHz üblich.

Nach der klanglichen Bearbeitung soll ein zuschaltbarer Echo-Effekt (Delay) folgen. Um diesem Effekt mehr Tiefe zu verleihen, wird das zu verzögernde Signal mit Hilfe eines Tiefpasses vorgefiltert.

## V. UMSETZUNG EINES PROTOTYPEN

Dieses Kapitel beschreibt den Hardwareaufbau sowie die Hardware-Beschreibung in VHDL.

### A. Hardwareumsetzung

In diesem Abschnitt wird die grundlegende Hardwareumsetzung vorgestellt. Zudem werden einige Besonderheiten beschrieben. Weitere Details zur Umsetzung sind in Quelle [7] zu finden.

1) *Blockschaltbild der Tochterplatine:* Abbildung 7 zeigt das Blockschaltbild der Tochterplatine. Die von dem Spannungsreglern (siehe Abbildung 4) erzeugten Spannungen sind mit den entsprechenden Bauteilen verbunden. Die Abbildung zeigt zudem Details des Signalfusses. Demnach werden der AD-Wandler mit Hilfe einer parallelen Programmierschnittstelle und der DA-Wandler und der LED-Treiber mit einer seriellen Verbindung (SPI) konfiguriert.

2) *Analoge Eingangsstufe:* Die äußeren Rahmenbedingungen legen in erster Linie den erlaubten Pegel am Eingang des AD-Wandlers fest. Die Verstärkung der Eingangsstufe (Abbildung 8) muss anschließend dem gewünschten maximalen Eingangspegel angepasst werden, um den Wandler entsprechend korrekt auszusteuern. Der AK5397EQ erlaubt an einem seiner Eingangs-Ports (positiver oder negativer Eingang) maximal  $2,8 V_{pp}$  (2,13 dBu). Die Spannung muss zudem mit einem Bias von 1,9 V angelegt werden, so dass sich die resultierende Spannung an einem Port zwischen 0,5 V und 3,3 V bewegt. Um nun eine Verstärkung für die OP-Stufe festlegen zu können, müssen mögliche Eingangspegel definiert werden. Die zu definierenden Pegel beziehen sich jeweils auf die, auf beide Eingangs-Ports zusammen gegebene Spannung. Diese teilen sich dann jeweils auf den positiven und negative Eingangs-Port auf. Als mögliche Eingangspegel sollen drei Fälle betrachtet und die entsprechende Verstärkung definiert werden:

- „Rundfunkpegel“: Der Rundfunknormpegel ist ein Standard für professionelle Audiogeräte und liegt bei +6 dBu ( $1,44 V_{RMS}$ ).
- „Consumer-Pegel“: Der Consumer-Pegel wird häufig mit -10 dBV ( $0,245 V_{RMS}$ ) definiert.
- „Mikrofon“: Die Empfindlichkeit eines dynamischen Standardmikrofons (z.B. Shure SM58) liegt bei 2 mV/Pa. Definiert man 115 dB SPL bis 120 dB SPL als maximal geplanten Schalldruck vor dem Mikrofon (Schlagzeug in einem Meter Entfernung) führt das zu einem Ausgangssignal von 32 mV (-27 dBu). Eine Verstärkung von 35 dB wäre nötig, um den Wandler richtig anzusteuern.

Um die Verstärkung für alle drei Fälle anpassen zu können, wird R1 aus Abbildung 8 mit Hilfe eines Jumpers variabel gestaltet. Die Werte für R2, R3 und R4 werden aus dem Datenblatt des AK5397EQ übernommen. Für jeden Fall wird zudem eine Aussteuerungsreserve (7 dB bzw. 10 dB) eingeplant. Mit Hilfe folgender Formel wurden für R1 folgende Werte (Standard E-Reihe) berechnet:

$$L_{dB} = 20 \cdot \log \left( \frac{R1}{R2} \cdot \frac{R1}{R3 + R4} \right) \quad (2)$$

- „Rundfunkpegel“:  $R1 = 430 \Omega$
- „Consumer-Pegel“:  $R1 = 2,7 k\Omega$
- „Mikrofon“:  $R1 = 39 k\Omega$

Mit den gewählten Widerstandswerten für R1 führen folgende Pegel zur Vollaussteuerung des Wandlers:

- Max. Eingangspegel „Rundfunkpegel“: +14 dBu
- Max. Eingangspegel „Consumer-Pegel“: -4 dBV
- Max. Schalldruckpegel „Mikrofon“: 122 dB SPL

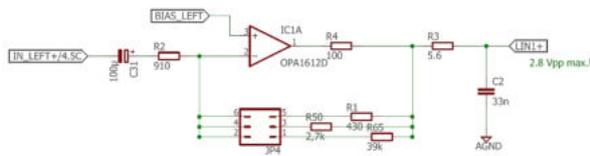


Abbildung 8. Analoge Eingangsstufe.

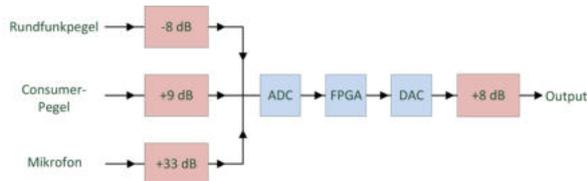


Abbildung 9. Gainstruktur des Systems.

Die resultierende Gain-Struktur des Audiopfades je nach Jumper-Konfiguration in der Eingangsstufe ist noch einmal in Abbildung 9 dargestellt.

3) *Platine und mechanischer Aufbau:* Aus dem fertigen Schaltplan wurde ein Layout für eine 4-lagige Platine erstellt. Um die Bauform möglichst kompakt zu halten, wurden größtenteils SMD-Bauteile verwendet. Die Oberseite der Platine bildet Layer 1. Auf ihm wurden alle SMD-Bauteile untergebracht. Außerdem wird hier das primäre Routing der Stromversorgung, Audiosignal- und teilweise digitalen Signalleitungen vorgenommen. Um das Audio-Board kompakt zu halten, wurde das aufzusetzende FPGA-Board unter das Audio-Board verlegt. Hierdurch lassen sich der LED-Treiber und die Taster mittig auf der Oberseite der Platine platzieren.

Layer 2 stellt den Masse-Layer (GND) für die gesamte Schaltung dar. Um Störungen der analogen Schaltungsteile durch die digitalen Schaltungsteile zu vermeiden, werden die zugehörigen Massebereiche voneinander getrennt und im Bereich der zentralen Versorgungsspannung zusammengeführt. Mit Hilfe von Vias wird die Masse direkt in der Nähe der jeweils angeschlossenen Bauteile an die Oberfläche geführt.

Auf Layer 3 werden die Versorgungsspannungen zu den jeweiligen Bauteilen verteilt. Auch hier musste darauf geachtet werden analoge und digitale Schaltungsteile zu trennen.

Auf Layer 4 verlaufen unter anderem die I<sup>2</sup>S-Signale mit seriell angelegten 0-Ohm-Widerständen, welche im Falle von zu starken „Overshoots“ der Clock-Signale bestückt werden können.

### B. Umsetzung des VHDL-Modells

In diesem Abschnitt wird die Modellierung des in Abschnitt IV.B beschriebenen Konzept in VHDL erläutert. Die finale Modell-Struktur ist in Abbildung 11 dargestellt. Während der Modellierung wurden Audio-Abtastraten bis zu 192 kHz erfolgreich getestet. Die finale Implementierung erfolgte in 48 kHz, da bei

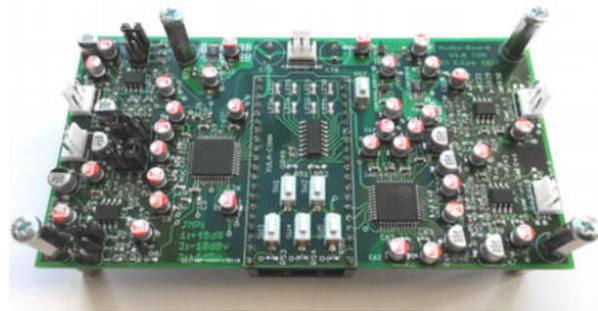


Abbildung 10. Mit Bauteilen bestückte Platine.

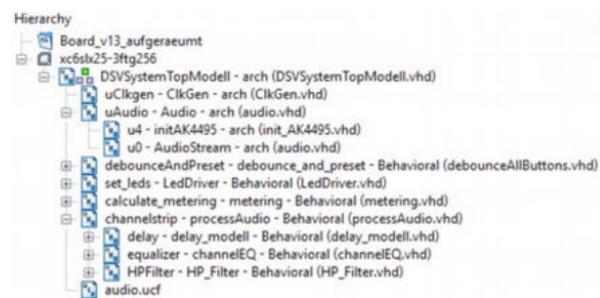


Abbildung 11. VHDL-Hierarchie der Hardware-Beschreibung.

dieser Abtastrate laut der Datenblätter von AD- und DA-Wandler die beste Audio-Performance (SNR) zu erreichen ist.

Die geplanten Subblöcke *Initialize*- und *AudioStream* wurden in einem übergeordneten Modell *Audio* zusammengefasst. Zudem wurden zwei zusätzliche Modelle *ClkGen* und *Metering* erstellt. *Metering* liegt vom Signalpfad her vor dem *LedDriver-Modell* und regelt den Wechsel zwischen LED-Statusmeldungen und Übersteuerungsanzeige (OVF) des AD-Wandlers. Insgesamt verfügt das *DSVSystemTop-Modell* damit über sechs Subblocks und einem User-Constraint-File *Audio.ucf*.

#### 1) *DSVSystemTOP-Modell:*

Das *DSVSystemTOP-Modell* (Abbildung 5) nimmt neben dem Signalarouting zwischen den in Abschnitt V.B genannten Subblocks auch die Verbindung nach außen mit Hilfe des User-Constraint-Files vor. *Audio.ucf* enthält die entsprechenden Verknüpfungen von FPGA-Ports zu den Ports der VHDL-Modelle.

Da die Konfiguration des AD-Wandlers AK5397EQ parallel über 13 Ports erfolgt, wird diese direkt im Top-Modell ausgeführt. Die Konfiguration in einen Subblock zu verlegen, würde lediglich bedeuten 13 einfache Logik-Signale weiterzuleiten. An dieser Stelle wurde dieser Schritt der Übersicht halber eingespart. Der DA-Wandler wird mit Hilfe des *InitAK4495-Modells* über die serielle Schnittstelle SPI (3-Wire) konfiguriert.

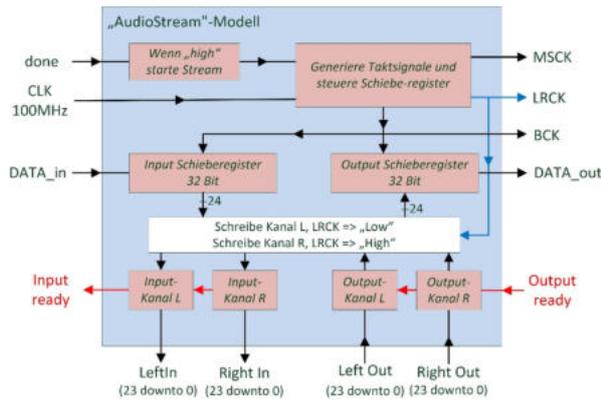


Abbildung 12. „AudioStream-Modell“.

### 2) AudioStream-Modell:

Das *AudioStream-Modell* (Abbildung 12) stellt dem *DSVSystemTop-Modell* die Audiodaten vom AD-Wandler als Vektor, also parallel zur weiteren Verarbeitung zur Verfügung. Umgekehrt nimmt es verarbeitete Signale in Vektorform an, um sie anschließend seriell zum DA-Wandler zu senden.

Nach der Initialisierung von AD- und DA-Wandler wechselt der Eingang *done* (siehe Abbildung 12) auf den Zustand „high“ und der Audio-Stream wird gestartet. Dabei wird ein vom AD-Wandler eingehender, serieller Datenstrom *DATA\_in* im Takt der Bit-Clock *BCK* in ein Schieberegister übernommen und in Abhängigkeit der Left-/Right-Clock *LRCK* in die Vektoren *Input-Kanal L* und *Input-Kanal R* geschrieben. Wenn beide Kanäle vollständig empfangen und geschrieben sind, wird dies über den Ausgang *Input ready* signalisiert. Die Vektoren können im *DSVSystemTop-Modell* nun zur Signalverarbeitung genutzt werden. Gleichzeitig übernimmt das Modell verarbeitete Signale in zwei Vektoren *Output-Kanal L* und *Output-Kanal R* und schreibt sie in Abhängigkeit von *LRCK* nacheinander in ein Output-Schieberegister, welches die Daten über einen seriellen Datenstrom *DATA\_out* im Takt der Bit-Clock *BCK* zum DA-Wandler ausgibt.

Neben den Clock-Signalen *LRCK* und *BCK* erzeugt das Modell noch eine Masterclock *MSCK* für den AD- und DA-Wandler. Das Verhältnis von *LRCK* und *BCK* ergibt sich direkt aus der Spezifikation der I<sup>2</sup>S-Schnittstelle. Die Masterclock beträgt immer ein ganzzahliges Vielfaches der Audioabtastrate und hängt von den Möglichkeiten und der Konfiguration des jeweiligen Wandler-ICs ab.

### 3) DebounceAndPreset-Modell:

Das *DebounceAndPreset-Modell* (Abbildung 13) verarbeitet die Signale der fünf Taster, erstellt je nach Eingabe ein Preset mit Informationen über die durchzuführende Signalverarbeitung und gibt entsprechende Informationen darüber über die 8 LEDs auf dem Audioboard aus.

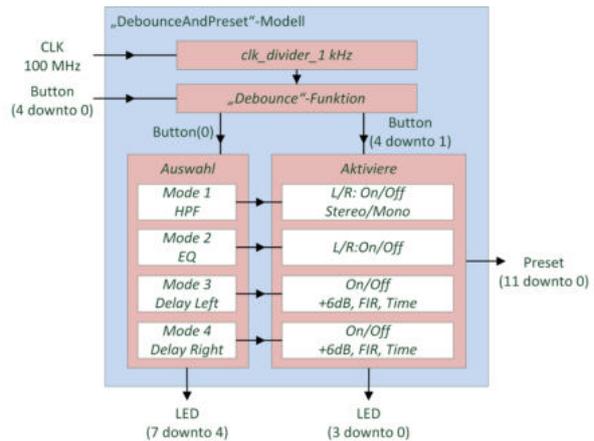


Abbildung 13. DebounceAndPreset-Modell.

Zunächst nimmt ein Subblock *clk\_divider\_1kHz* eine Skalierung der eingehenden Taktfrequenz von 100 MHz auf 1 kHz vor. Die Abfrage der 5 Taster *Button(4 downto 0)* erfolgt daher 1000 Mal in der Sekunde. Bevor die Taster Bits in diversen Vektoren setzen, werden sie mit Hilfe einer „Debounce“-Funktion entprellt. Anschließend wird Taster 1 (*Button(0)*) dazu genutzt, um in einem Benutzermenü einen von vier Modi auszuwählen. Der momentan gewählte Modus wird auf den ersten 4 LEDs *LED(7 downto 4)* ausgegeben (Abbildung 13). Somit weiß der Nutzer in welchem Modus er sich gerade befindet. In jedem Modus kann der Nutzer mit den Tastern *Button(4 downto 1)* Funktionen aktivieren, welche auf den zweiten 4 LEDs *LED(3 downto 0)* dargestellt werden. Zustandsinformationen über die aktivierten Modi werden anschließend in einem gemeinsamen Vektor *preset(11 downto 0)* abgelegt und dem *DSVSystemTop-Modell* zur Verfügung gestellt. Anhand des Vektors *preset* werden Funktionen (siehe Tabelle III) innerhalb der Signalverarbeitung aktiviert oder geroutet. Eine genauere Beschreibung der einzelnen Funktionen folgt in Abschnitt V.B.2) bis V.B.4)c). Nicht alle Menüfunktionen werden in den *preset*-Vektor übernommen. Über den Modus 1 lässt sich ein Stereo-Modus aktivieren (Abbildung 13). Das bedeutet, dass aktivierte Funktionen der Signalverarbeitung (siehe Tabelle 3) immer für beide Audiokanäle gelten. Die Information über einen aktivierten Stereo-Modus wird anderen Subblöcken nicht direkt mitgeteilt, sondern beim Schreiben des *preset*-Vektors berücksichtigt. Beispielsweise werden beim Aktivieren des Hochpassfilters im Stereo-Modus *preset(11)* und *preset(10)* auf „high“ gesetzt.

### 4) ProcessAudio-Modell:

Das Konzept der Audiosignalverarbeitung aus Abschnitt IV.C wird im *processAudio-Modell* (Abbildung 14) umgesetzt. Hier erfolgt die Verarbeitung der vom *AudioStream-Modell* bereitgestellten Daten der 24 Bit breiten Audiokanäle *LeftIn* und *RightIn*, welche in

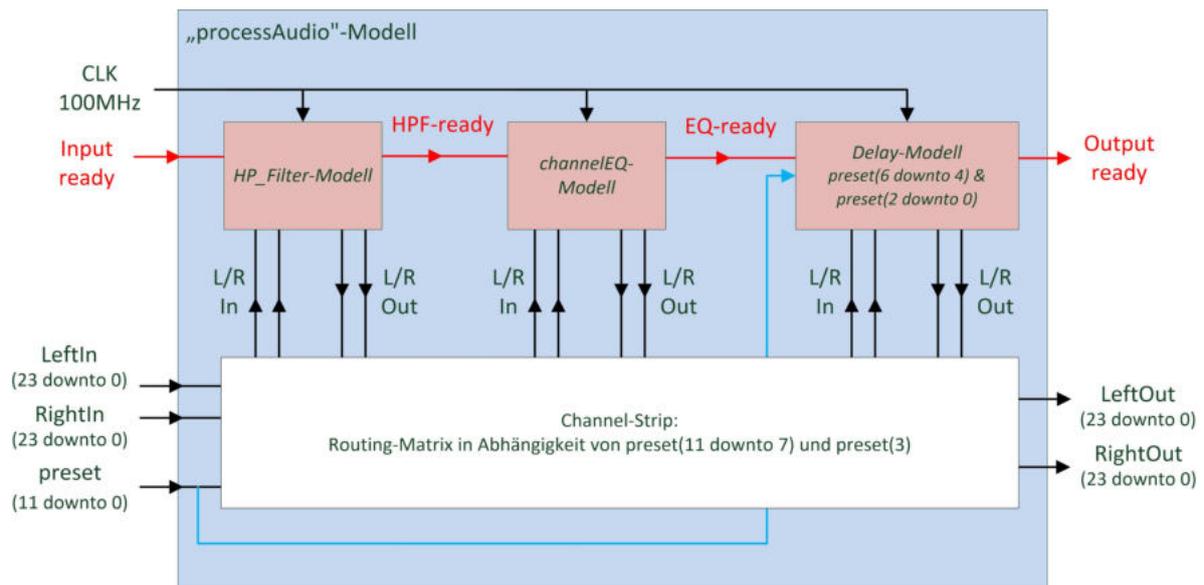


Abbildung 14. processAudio-Modell.

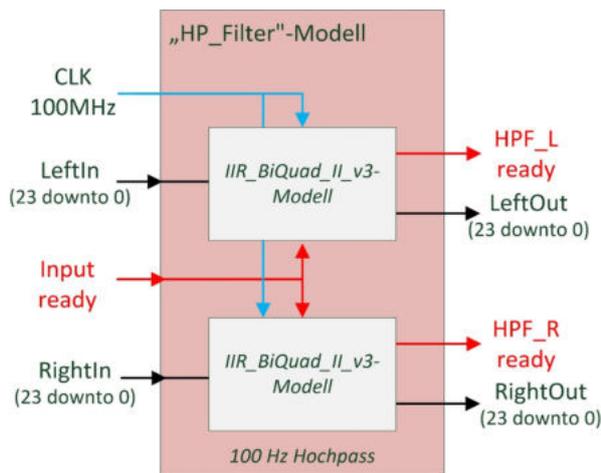


Abbildung 15. HP\_Filter-Modell, 100 Hz Hochpassfilter.

Tabelle III  
FUNKTIONEN IM PRESET.

Preset (bit)	Funktionsbeschreibung
11	Aktiviere HPF 100Hz, Kanal L
10	Aktiviere HPF 100Hz, Kanal R
9	Aktiviere Equalizer, Kanal L
8	Aktiviere Equalizer, Kanal R
7	Aktiviere Delay-Effekt, Kanal L
6	Verstärke Delay um 6dB, Kanal L
5	Schalte FIR-Vorfilter im Delay-Effekt aus, Kanal L
4	Variiere Delay-Zeit, Kanal L
3	Aktiviere Delay-Effekt, Kanal R
2	Verstärke Delay um 6dB, Kanal R
1	Schalte FIR-Vorfilter im Delay-Effekt aus, Kanal R
0	Variiere Delay-Zeit, Kanal R

a) HP\_Filter-Modell

Abbildung 14 links dargestellt sind. Das Modell enthält drei Subblöcke *HP\_Filter-Modell*, *channelEQ-Modell* und *Delay-Modell* sowie eine Routing-Matrix, welche, in Abhängigkeit der im Vektor *preset(11 downto 0)* befindlichen Einstellungen, ein Routing der Audiodaten *LeftIn* und *RightIn* vornimmt. Somit kann mit Ausnahme des *Delay-Modells*, jeder dargestellte Subblock in den Signalweg einbezogen werden. Das *Delay-Modell* liegt parallel im Audiosignalweg. Die einzelnen Subblöcke melden über eine *ready-Leitung*, wenn sie die jeweilige Signalverarbeitung abgeschlossen haben. Das *Delay-Modell* erhält zudem die Teilvektoren *preset(6 downto 4)* und *preset(2 downto 0)*, um Funktionen innerhalb des Modells zu aktivieren.

Das *HP\_Filter-Modell* aus Abbildung 14 wird in Abbildung 15 detailliert dargestellt. Gemäß dem Konzept in Abschnitt IV.C erfolgt in diesem Block die Umsetzung eines Trittschallfilters, welcher Audiosignale mit einem 100 Hz Hochpassfilter, 2. Ordnung, Filtergüte 0,7071 bearbeitet. Die Umsetzung erfolgt mit Hilfe eines BiQuad-Filters, dessen Koeffizienten mit einem BiQuad-Kalkulator berechnet wurden. Der Kalkulator lieferte als Ergebnis Filterkoeffizienten in Dezimalform mit einem Wertebereich von -2 bis 2. Zur Nutzung in einer VHDL-Umgebung müssen die Koeffizienten in ein Fixed-Point-Binär-Format umgerechnet werden. Eine Rechenvorschrift dazu findet sich in der Literatur von Tony Storey [9]. Die eigentliche Filterung

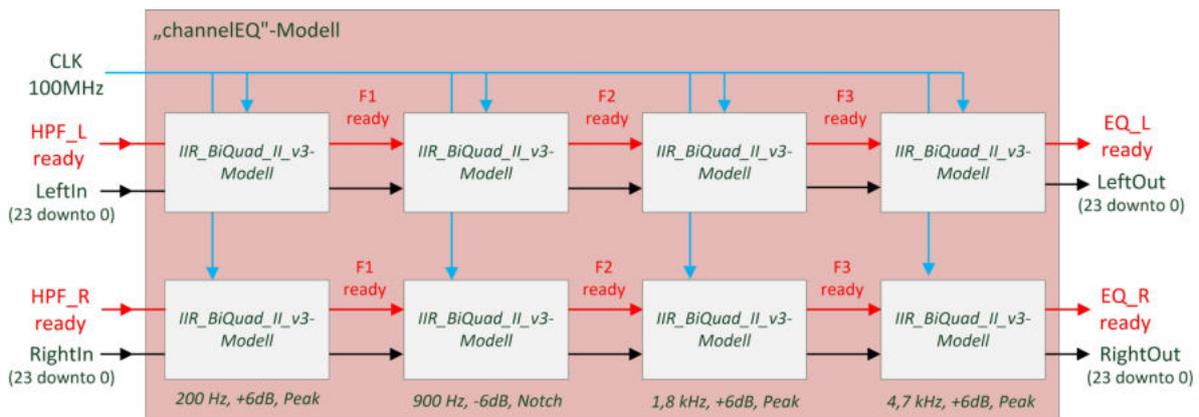


Abbildung 16. Filterparameter des channelEQ-Modells.

Tabelle IV  
FILTERPARAMETER DES CHANNELLEQ-MODELLS.

	Art	Frequenz	Gain	Güte
<b>Filter 1</b>	Peak	200 Hz	+6 dB	1,5
<b>Filter 2</b>	Notch	900 Hz	-6 dB	1,5
<b>Filter 3</b>	Peak	1,8 kHz	+6 dB	2,0
<b>Filter 4</b>	Peak	4,7 kHz	+6 dB	0,707

übernimmt ein Subblock *BiQuad\_II\_v3* (Version 3), welches von Tony Storey programmiert und zur freien Verfügung gestellt wurde. Ausführliche Informationen zur Version 1 des Modells liefert die Quelle [9]. Das Modell wurde von Storey zur Version 3 weiterentwickelt und verwendet statt fünf DSP-Multiplizierern nun einen einfachen „Shift-Add“-Multiplizierer zur Filterberechnung. Bei der Einbindung des Modells stellte sich heraus, dass das Filter einen ungewollten Pegelabfall zur Folge hatte. Daher wurde das Modell von Storey leicht modifiziert, um den Pegelabfall zu kompensieren. Das Modell nimmt einen 24-Bit-Vektor an und gibt nach der Filterung einen 24-Bit-Vektor zurück. Daher werden im *HP\_Filter-Modell* zwei *BiQuad\_II\_v3-Modelle* instanziiert, um beide Audiokanäle entsprechend filtern zu können.

#### b) channelEQ-Modell

Das channelEQ-Modell (Abbildung 16) nimmt die in Abschnitt IV.C beschriebene Filterung der anliegenden Eingangssignale vor. Hierfür werden für zwei Audiokanäle insgesamt 8 Biquad-Filter verwendet. Es wird das gleiche *Biquad-Modell* wie in Abschnitt V.B.4)a) verwendet und mehrfach für jeden Kanal, jedoch mit verschiedenen Filterkoeffizienten instanziiert. Die Filterparameter werden in Tabelle IV aufgelistet.

Jede Filterstufe signalisiert der nachfolgenden Stufe über ein *ready*-Signal, wenn die Berechnung abgeschlossen wurde. Nach Abschluss der Filterung stehen beide Kanäle jeweils mit eigenem *EQ-ready*-Signal für die nachfolgende Verarbeitung zur Verfügung.

#### c) Delay-Modell mit FIR-Filter

Wie in Abschnitt V.B.4) bereits erwähnt liegt das *Delay-Modell* parallel im Signalweg. Das Modell erhält beide Audiosignale, welche nach der Verarbeitung dem ursprünglichen Signal hinzuaddiert werden. Abbildung 17 zeigt der Übersicht halber eine einkanalige Darstellung des *Delay-Modells*. Im *processAudio-Modell* ist das Delay zweikanalig bzw. für den linken und rechten Kanal ausgeführt. Die Umsetzung wird mit Hilfe eines Ringspeichers, in der Zeichnung als *Ringbuffer* bezeichnet, inklusive einer Rückführung aufgebaut. Der Abstand zwischen *write*- und *read*-Zeiger bestimmt die Verzögerungszeit des Delay-Effektes. Das Eingangssignal wird im Pegel um 6 dB reduziert, bevor es dem *Ringbuffer* zugeführt wird. Das Signal wird hierdurch ebenfalls um 6 dB abgeschwächt. Auf diese Weise ergibt sich ein abklingendes Echo des Input-Signals, welches im *processAudio-Modell* dem Originalsignal hinzuaddiert wird. Bevor das Delay-Signal das Modell verlässt, wird es nochmals um 6 dB abgesenkt. Die Ringbuffer für beide Audiokanäle wurde mit Hilfe des *IPCores-Buffer-Plugins* in *ISE Design Suite* (Xilinx) realisiert.

Um den Delay-Effekt klanglich natürlicher zu gestalten, wird das Eingangssignal mit einem FIR-Tiefpassfilter mit einer Grenzfrequenz von 4 kHz und einer Ordnung von 10 (11 Filter-Koeffizienten) vorverarbeitet. Die Filterkoeffizienten für den Filter wurden zuvor mit Matlabs *FilterDesigner* (Mathworx) erstellt und dann mit Hilfe des *IPCores-FIR-Plugins* (Xilinx) realisiert. Die Vorfilterung nimmt eine Dämpfung der hohen Frequenzen vor. Hierdurch kann ein natürlicheres Echo, welches mit der Entfernung normalerweise an hohen Frequenzen verliert, nachgebildet werden. Der Delay-Effekt kann je nach Verzögerungszeit auch einen Raumeindruck vermitteln und damit einen Hall-Effekt simulieren.

Über den eingehenden Teilvektor *preset(6 downto 4)* lassen sich der FIR-Filter an- und abschalten, die Verzögerungszeit über den *Ringbuffer* und die 6dB

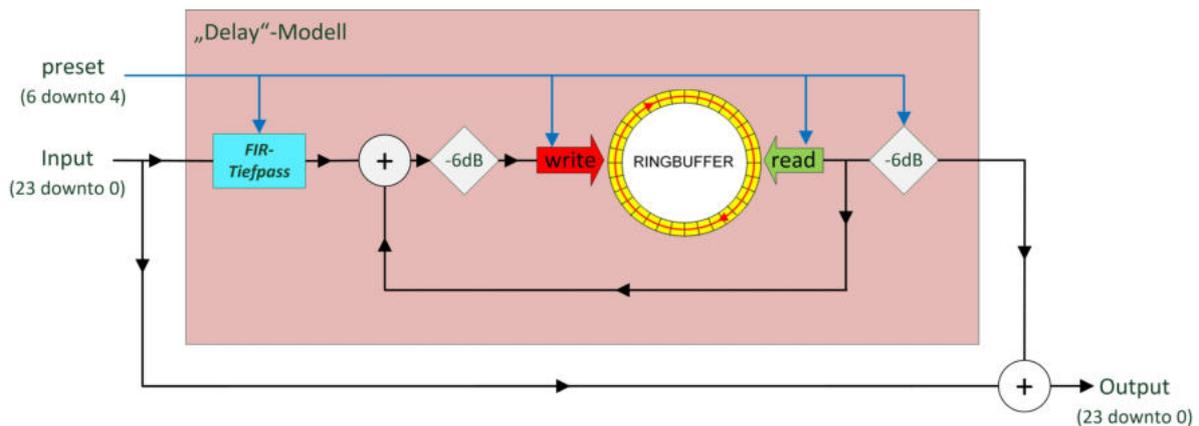


Abbildung 17. Einkanalige Darstellung des Delay-Modells.

Tabelle V  
AUDIOTECHNISCHE MESSWERTE DES DSP-SYSTEM.

Bei 1 kHz, 20 Hz–20 kHz, SR 48 kHz	Max.
Vollaussteuerung bei „Rundfunkpegel“	+14,7 dBu
Vollaussteuerung bei „Consumer-Pegel“	-1,0 dBu
Vollaussteuerung bei „Mikrofon“	-24,3 dBu
Signal-to-Noise-Ratio (A-weighted)	117,77 dB
Crosstalk	-131,59 dB
THD+N Ratio (at +5 dBu)	-100,8 dB (0,00091 %)
THD Ratio (at +5 dBu)	-105,0 dB (0,00056 %)
THD+N Level (at -7 dBu)	-98,85 dBu
THD Level (at -7 dBu)	-112,84 dBu
System-Laufzeit	1,472 ms

Ausgangsdämpfung variieren. Hierdurch lassen sich unterschiedlich klingende Effekte generieren. Vergleiche hierzu noch einmal mit Tabelle III.

## VI. ERGEBNISSE

Im Folgenden werden nun Messungen vorgestellt, die die Audio-Performance (Abschnitt VI.A) und die Eigenschaften der Signalverarbeitung (Abschnitt VI.B) beschreiben. In Abschnitt VI.C erfolgt eine Gegenüberstellung mit dem in Kapitel III aufgeführten Stand der Technik.

### A. Messungen der audio-charakteristischen Eigenschaften

Bei der Messung der audio-charakteristischen Eigenschaften ist die Signalverarbeitung im DSV-System abgeschaltet und es findet somit lediglich eine AD- und DA-Wandlung bzw. Weiterleitung durch den FPGA statt. D.h. die Signale werden 1:1 durch das FPGA durchgeleitet. Charakteristische Messwerte sind der Tabelle V zu entnehmen.

Demnach erreicht das DSV-System in der Einstellung „Rundfunkpegel“ seine Vollaussteuerung bei

+14,7 dBu. Die Signal-to-Noise-Ratio liegt bei 118 dBu (gerundet auf ganze dB) und ist mit dem SNR des RME Fireface 802 aus der Kategorie 3 (Abschnitt III.B, Tabelle I) vergleichbar. Die Datenblätter von AD- und DA-Wandler lassen vermuten, dass der SNR maßgeblich durch den DA-Wandler bestimmt wird, da er laut Datenblatt einen Wert von bis zu 120 dB aufweisen soll, der AD-Wandler jedoch bis zu 127 dB. Um diese Vermutung bei der Messung des Gesamtsystems bestätigen zu können, wurde für eine der SNR-Messungen der Pegel in der digitalen Ebene durch das FPGA um 60 dB angehoben. Nach der Verstärkung überwiegt das Rauschen des AD-Wandlers inklusive der Eingangsstufe gegenüber dem DA-seitigen Rauschen um ein Vielfaches (mind. 50 dB  $\Rightarrow$  Faktor 316), so dass der DA-seitige Rauschanteil bei der folgenden Überlegung vernachlässigt werden kann.

Von dem gemessenen Rauschpegel wird nun die Verstärkung von 60 dB subtrahiert und man erhält das sogenannte „Equivalent-Input-Noise“ (E.I.N.). Der Bereich zwischen E.I.N. und Vollaussteuerung stellt dann den Dynamikbereich dar, welcher, wie in der Vorüberlegung erwähnt, nun vom AD-Wandler und Eingangsstufe bestimmt ist. Der mit dieser Methode ermittelte SNR auf der AD-Seite beträgt näherungsweise 123,9 dB.

Abbildung 18 zeigt den Frequenzgang und die Verstärkung des DSV-Systems. Im Bereich zwischen 20 Hz und 20 kHz ist das Übertragungsverhalten sehr linear. Zu 20 kHz hin gibt es einen geringen Abfall von 0,5 dB. Je nach Konfiguration bzw. Wahl des digitalen Anti-Aliasing-Filters des AD-Wandlers kann sich das Verhalten im Frequenzgang an dieser Stelle ändern. Zudem ist erkennbar, dass das System eine Grundverstärkung von 1,25 dB (Angabe bei 1 kHz) aufweist.

Abbildung 19 zeigt den Frequenzgang von 10 kHz bis 40 kHz. Es lässt sich eine gute Dämpfung im Sperrbereich von mehr als 100 dB feststellen. In das Basisband zurückgespiegelte Aliasingprodukte werden also um 100 dB gedämpft wiedergegeben.

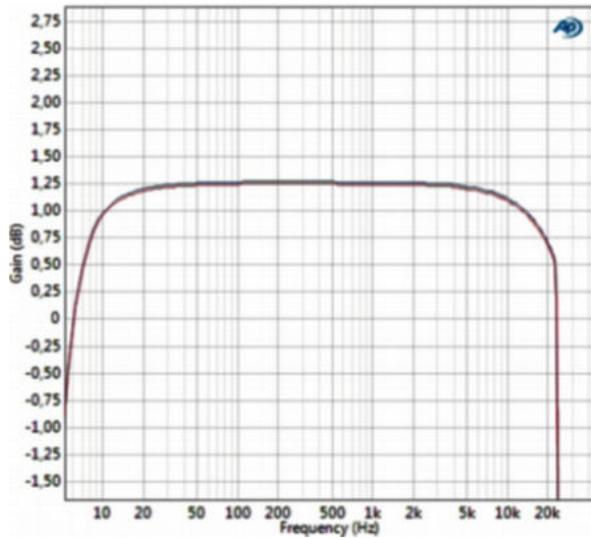


Abbildung 18. Frequenzgang und Gain bei 14 dBu Input, Kanal Links (blau), Kanal Rechts (rot).

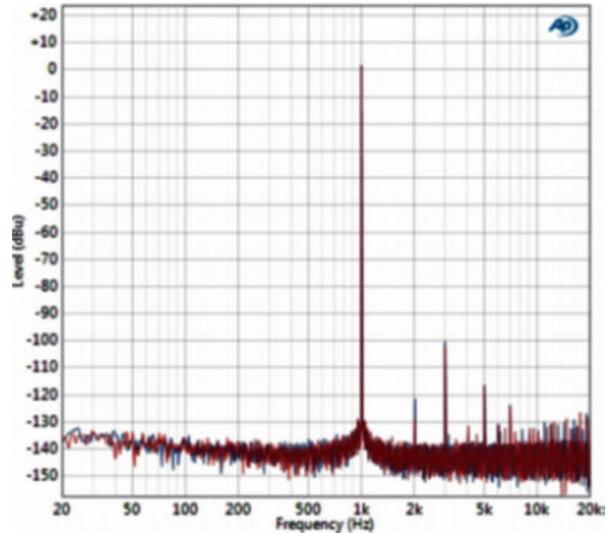


Abbildung 20. 1 kHz Frequenzspektrum bei 0 dBu Input, Kanal Links (blau), Kanal Rechts (rot).

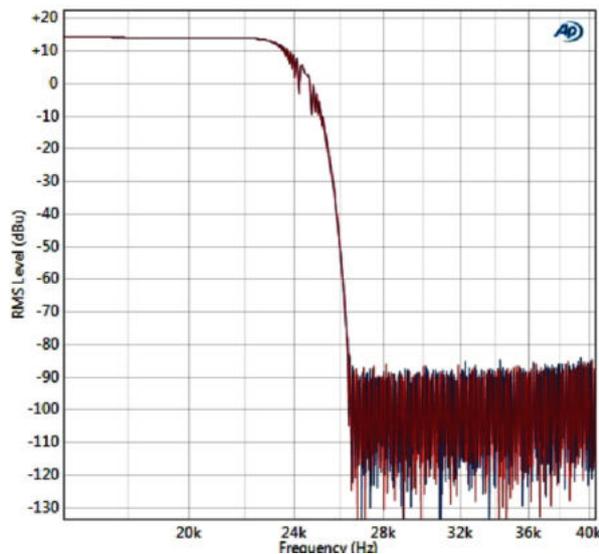


Abbildung 19. Dämpfung im Sperrbereich bei 14 dBu Input, Kanal Links (blau), Kanal Rechts (rot).

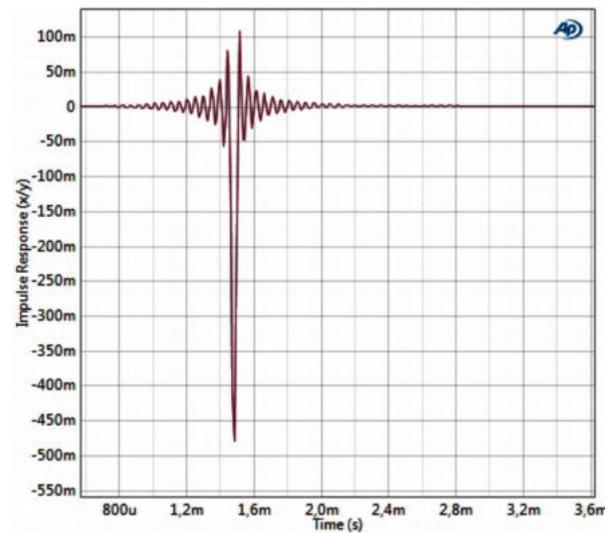


Abbildung 21. Impulsantwort des DSP-Systems, Kanal Links (blau), Kanal Rechts (rot).

Abbildung 20 zeigt das Frequenzspektrum des Systems bei Anregung mit einem 1 kHz-Sinussignal mit einem Pegel von 0 dBu. Das Spektrum wird häufig als „Fingerabdruck“ eines Systems bezeichnet. Zu erkennen ist, dass das System Obertöne (THD) erzeugt. Die Obertöne werden im besten Fall um 105 dB gedämpft (siehe auch Tabelle V). Hierbei handelt es sich um einen guten bis sehr guten Wert. Vergleiche hier mit Tabelle I aus Abschnitt III.B.

Abbildung 21 zeigt die Impulsantwort des Systems. Das System verursacht eine Phasenumkehr (Hauptimpuls geht nach unten). Dies ist auf den invertierenden Verstärker der analogen Eingangsstufe zurückzuführen. Wenn es die Anwendung erfordert, kann die Phasenumkehr in der digitalen Ebene einfach mit einer

Multiplikation mit -1 kompensiert werden. Die Form des Impulses entspricht einer SI-Funktion, welche der Impulsantwort eines Tiefpasses bei der halben Abtast-rate (Anti-Aliasing-Filter) entspricht.

Der Pegelverlauf des Systems ist im Bereich von -80 dBu bis 14 dBu linear. Hierzu wurde eine Messung der Linearitätsabweichung (Abbildung 22) durchgeführt. Der Anstieg der Messkurve unterhalb von 80 dBu begründet sich durch das bei kleinen Pegeln stärker ins Gewicht fallende Rauschen.

### B. Funktion der Algorithmen der digitalen Signalverarbeitung

Um die implementierte digitale Signalverarbeitung zu verifizieren, wurden Frequenz- und Phasengänge der Filter und eine Messung der Delay-Impulsantwort

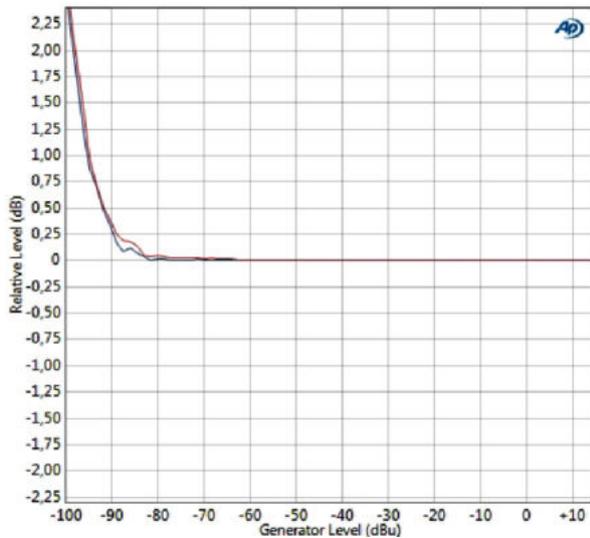


Abbildung 22. Linearitätsabweichung, Kanal Links (blau), Kanal Rechts (rot).

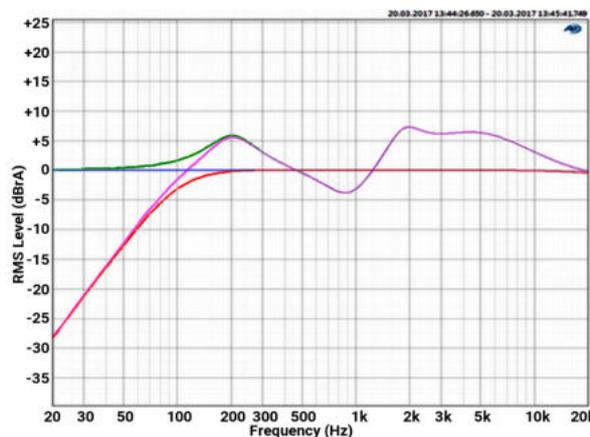


Abbildung 23. Frequenzgang der implementierten Filter: „Flat“ (blau), „HPF“ (rot), „ChannelEQ“ (grün) und „HPF+ChannelEQ“ (lila).

vorgenommen. Abbildung 23 zeigt 4 Filterkurven „Flat“ (blau), „HPF“ (rot), „ChannelEQ“ (grün) und „HPF+ChannelEQ“ (lila). Der Hochpassfilter (HPF) hat, wie vorgesehen, seine Grenzfrequenz bei 100 Hz. Auch der Phasengang (Abbildung 24) weist, wie bei einem Filter zweiter Ordnung üblich, eine Phasenverschiebung von  $+90^\circ$  am  $-3$  dB-Punkt auf. Auch der ChannelEQ (grüne Kurve in Abbildung 23 und Abbildung 24) arbeitet wie erwartet. Darüber hinaus zeigen die Abbildungen eine Kombination aus Hochpassfilter und ChannelEQ. Damit entspricht die Filterkurve dem Filterkonzept aus IV.C.

Die Funktion des Delay-Effektes beinhaltet logarithmische Pegelabsenkungen. Um diese auf einer vertikalen Diagrammchse sinnvoll darstellen zu können, wurde zunächst eine Impulsantwort des Delay-Effektes gemessen und anschließend als „Energie-Time-Curve“ (Abbildung 25) dargestellt. Dargestellt werden zwei

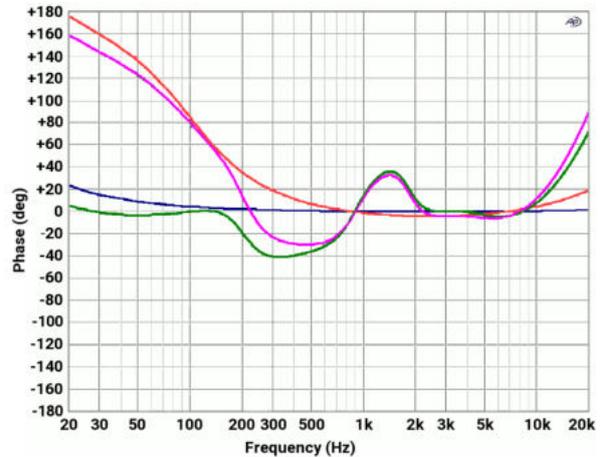


Abbildung 24. Phasengang der implementierten Filter: „Flat“ (blau), „HPF“ (rot), „ChannelEQ“ (grün) und „HPF+ChannelEQ“ (lila).

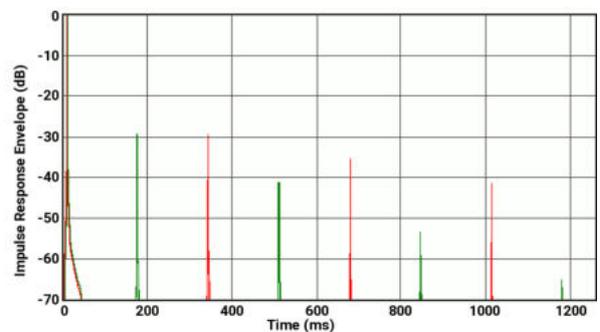


Abbildung 25. Delay-Impuls-Muster des Delay-Effektes, "Time1" in Grün, "Time2" in Rot.

Delay-Impuls-Muster. Das grüne Muster entspricht der Delay-Grundeinstellung. Es werden Impulse im Abstand von 167 ms ( $8.000 \text{ Sample} \cdot 1/(48000 \text{ Hz})$ ) erzeugt. Der starke Abfall der ersten Wiederholung ist beabsichtigt und begründet sich durch das implementierte FIR-Filter und die Absenkung um 6 dB am Ausgang des Delays. Im Anschluss daran fällt jede weitere Reflexion um weitere 6 dB durch die Dämpfung in der Rückkopplung ab. Variiert man die Delay-Zeit entsprechend Tabelle III, erzeugt das System ein Delay-Impuls-Muster (in Abbildung 25 in Rot) im Abstand von 334 ms ( $16.000 \text{ Sample} \cdot 1/(48000 \text{ Hz})$ ).

### C. Vergleich der Spezifikationen mit dem aktuellen Stand

Tabelle VI und Tabelle VII zeigen eine Zusammenfassung der Ergebnisse für das neue FPGA-basierte DSV-System. Beim Vergleich der SNR-Werte ist der niedrigere Wert eines jeden Systems relevant, da er den Dynamikbereich des Gesamtsystems bestimmt. Die Werte neuen Systems (118 dB) sind mit der Kategorie 3 (116 dB bis 123 dB, Tabelle II) vergleichbar. AD-seitig kann es (bei Rundung auf ganze dB) die anderen Systeme sogar um 1 dB übertreffen. Mit THD-/THDN-

Tabelle VI  
ZUSAMMENFASSUNG DER MESSERGEBNISSE FÜR DAS NEUE  
FPGA-BASIERTE DSV-SYSTEM.

SNR/ ADC (dB, A-w.)	THDN/ ADC (dB)	SNR/ DAC (dB, A-w.)	THDN/ DAC (dB)
124	-101	118	-101

Tabelle VII  
KENNDATEN DES NEUEN FPGA-BASIERTE DSV-SYSTEMS.

Bit-Tiefe	Abtastrate (kHz)	Fläche (cm <sup>2</sup> )	Preis (€)
32	bis zu 384	76,23	191 bis 240

Werten von -101 dB gruppiert sich das neue System an der Untergrenze der Kategorie 3 ein.

Die Platinenfläche beträgt ca. 76 cm<sup>2</sup>, womit das DSV-System flächenmäßig das kleinste aller Systeme ist (vgl. Tabelle I). Auch der Preis des Systems gliedert sich wie geplant und je nach Wahl des FPGAs in die Kategorie 1 ein oder liegt etwas darüber (mit XuLA2-LX25).

## VII. FAZIT

Ziel dieser Arbeit war die Entwicklung eines Systems zur digitalen Audiosignalverarbeitung inklusive Analog-Digital-Analog-Wandlung, welches bei geringer Baugröße professionellen Audioanwendungen gerecht werden kann. Zur Umsetzung des Systems wurden aktuell auf dem Markt erhältliche AD- und DA-Wandler verglichen und entsprechende Komponenten für ein Prototypensystem ausgewählt. Zudem wurde ein vorkonfektioniertes FPGA-Board ausgesucht, um es als Basis für einen Prototyp nutzen zu können. Anschließend wurden ein Schaltplan und ein Platinen-Layout für eine steckbare Tochterplatine erstellt und mit entsprechenden Bauteilen bestückt. Im Anschluss erfolgte die Implementierung einer beispielhaften Signalverarbeitung in VHDL zur Nachbildung eines Mischpultkanalzugest. Die Audio-Performance und die Signalverarbeitung wurden daraufhin elektrisch vermessen und die Messergebnisse dem recherchierten Stand der Technik gegenübergestellt.

Das entwickelte System ist äußerst kompakt und kleiner als alle in der Recherche gefundenen Systeme. Seine AD- und DA-Wandler sind zweikanalig ausgelegt. Die elektrischen Messungen aus Abschnitt VI.A bestätigen eine Audio-Performance, die bisher nur hochpreisigen Geräten vorbehalten war. Besonders herausstechend ist der hohe Signalrauschabstand (SNR), welcher das System auch für anspruchsvolle Aufgaben der digitalen Audiosignalverarbeitung tauglich macht, wie zum Beispiel der Einsatz im Tonstudio als Kanal-erweiterung für Mischpulte oder als rauscharme aktive FIR-Filter-Frequenzweiche für Studiolautsprecher.

Da das System aus einer separaten Tochterplatine mit aufsteckbaren FPGA besteht, ist es äußerst va-

riabel, da beide Komponenten getrennt voneinander weiterentwickelt werden können bzw. bei Bedarf austauschbar sind.

Die in der Audiosignalverarbeitung implementierten Filter wurden für eine Abtastrate von 48 kHz und einer Bittiefe von 24 bit berechnet. Das System kann darüber hinaus mit einer maximalen Abtastrate von 384 kHz und eine Bittiefe von bis zu 32 bit genutzt werden. Messungen der Frequenzgänge und Impulsantworten bestätigen die Wirksamkeit der Audiosignalverarbeitung. So wurden IIR- und FIR-Filter sowie ein Delay-/Hall-Effekt mit verschiedenen Einstellmöglichkeiten implementiert. Die Audiosignalverarbeitung ist sowohl im Stereo- als auch im Mono-Betrieb unabhängig voneinander nutzbar. Hier bietet sich der Einsatz als Miniaturmischpult an.

Die Materialkosten des Systems betragen je nach Wahl des FPGAs 190 € bis 240 €. Damit liegt es in einem Preisbereich in dem bisher nur Geräte mit wesentlich schlechterer Audio-Performance zu finden waren. Man berücksichtige zusätzlich noch die Tatsache, dass es sich beim entwickelten System um einen Prototyp in individueller Fertigung und keine Massenproduktion handelt. In einer Massenproduktion würde der Gesamtpreis noch niedriger ausfallen. Allerdings müssten die Kosten für das Engineering mit einkalkuliert werden.

Damit ist es gelungen die Vorzüge hochwertiger Audiogeräte in einem kompakten FPGA-basierten System zur digitalen Audiosignalverarbeitung umzusetzen. Ein Nachteil des Systems ist die Tatsache, dass sich die Konfiguration des FPGAs mittels VHDL wesentlich aufwändiger gestaltet als bei einem DSP und damit ein wesentlich kleinerer Kreis von Programmierern in der Lage ist das System entsprechend zu konfigurieren.

Das entwickelte System beherbergt Potential für unzählige Weiterentwicklungen. Beispielsweise lassen sich der AD- oder der DA-Wandler-Abschnitt durch digitale Audioschnittstellen austauschen und eröffnen damit neue Einsatzgebiete. Falls die Zielanwendung bekannt ist, wäre es auch denkbar Funktionen einzusparen und damit belegte FPGA-Ports für digitale Schnittstellen freizugeben. Hierdurch könnte das System noch kompakter gestaltet werden. Auch die beidseitige Bestückung der Platine würde für weitere Platzersparnis sorgen.

Das System lässt sich momentan mit zwei 9V Batterien oder mit einem entsprechenden Netzteil betreiben. Betrachtet man den Anwendungsfall „Mischpult“ liegt der Wunsch nach einer Stromversorgung mittels einer Phantomspeisung (9 V bis 48 V) nahe, die bei Bedarf an ein Kondensatormikrofon weitergereicht werden kann.

Die Austauschbarkeit des FPGAs bietet die Möglichkeit für leistungshungrigere Anwendung ein stärkeres FPGA zu verwenden. Hierfür müsste lediglich eine entsprechende Platine zum Aufstecken auf das

Audioboard konstruiert werden. Es wäre auch denkbar das FPGA durch einen digitalen Signalprozessor zu ersetzen, dessen Programmierung für den Laien anwenderfreundlicher wäre als die Programmierung in VHDL. Hierfür müsste allerdings noch ein separater Mikrokontroller vorgesehen werden, der die Konfiguration von AD- und DA-Wandler vornimmt.

#### LITERATURVERZEICHNIS

- [1] P. Ekas, *FPGAs rapidly replacing high-performance DSP capability*, <http://signal-processing.mil-embedded.com/article-id/?2051>, USA California, Feb. 2007.
- [2] M. Shirvaikar, T. Bushnaq, *A comparison between DSP and FPGA platforms*, <https://www.researchgate.net/publication/228453392>, März 2015.
- [3] M. Yasir, *FPGA or DSP Processor - Parameters to Make the Right Choice*, <https://www.fpgarelated.com/showarticle/21.php>, Dez. 2011.
- [4] Altera, *White Paper - FPGA vs. DSP Design Reliability and Maintenance*, San Jose, Kanada, Mai 2007.
- [5] B. TheMason, *Audio-DSP mit Spartan 3-FPGA*, [http://www.mikrocontroller.net/articles/Audio-DSP\\_mit\\_Spartan\\_3-FPGA#Hardware](http://www.mikrocontroller.net/articles/Audio-DSP_mit_Spartan_3-FPGA#Hardware), Feb. 2017.
- [6] G. Leopold, *Intel Eyes Datacenter, IoT with Altera Deal*, <https://www.enterprisetech.com/2015/06/01/intel-eyes-datacenter-iot-with-altera-deal/>, Juni 2015.
- [7] C. Epe, *realHDAudio-Board with FPGA (XuLA 2) and 124/118 dB SNR*, <http://realhdaudio.de/2018/04/18/xula2/>, April 2018.
- [8] XESS, *XESS XuLA2*, <http://www.xess.com/static/media/manuals/XuLA2-manual.pdf>, Dez. 2012.
- [9] T. Storey, *IIR Filter Design in VHDL*, <https://eewiki.net/display/LOGIC/IIR+Filter+Design+in+VHDL+Targeted+for+18-Bit,+48+KHz+Audio+Signal+Use>, März 2017.



**Christian Epe** erhielt sein Diplom als Ton- und Bildingenieur im Jahr 2010 und arbeitete danach zunächst in der Akustik-Branche. Seit 2012 ist er wissenschaftlicher Mitarbeiter mit Schwerpunkt Audio-Technik an der Hochschule Düsseldorf (HSD). Er absolvierte 2017 zu dem den Masterstudiengang Elektro- und Informationstechnik an der HSD.



**Bernhard Rieß** schloss sein Studium der Elektrotechnik und Informationstechnik an der Technischen Universität München im Jahr 1992 ab. Anschließend war er wissenschaftlicher Mitarbeiter am Lehrstuhl für Rechnergestütztes Entwerfen an der TU München und wurde dort 1996 zum Dr.-Ing. promoviert. Seit Anfang 1997 war er als Entwicklungsingenieur bei der Infineon Technologies AG in München/Neubiberg tätig. 2012 wurde er von der Hochschule Düsseldorf als Professor für das Lehrgebiet Mikroelektronik berufen.



# Implementierung des Secure Hash Algorithmus SHA-256

Malek Safieh, Johann-Philipp Thiers, Jürgen Freudenberger

**Zusammenfassung**—Digitale Signaturen zum Überprüfen der Integrität von Daten, beispielsweise von Software-Updates, gewinnen zunehmend an Bedeutung. Im Bereich der eingebetteten Systeme kommen derzeit wegen der geringen Komplexität noch überwiegend symmetrische Verschlüsselungsverfahren zur Berechnung eines Authentifizierungscodes zum Einsatz. Asymmetrische Kryptosysteme sind rechenaufwendiger, bieten aber mehr Sicherheit, weil der Schlüssel zur Authentifizierung nicht geheim gehalten werden muss. Asymmetrische Signaturverfahren werden typischerweise zweistufig berechnet. Der Schlüssel wird nicht direkt auf die Daten angewendet, sondern auf deren Hash-Wert, der mit Hilfe einer Hashfunktion zuvor berechnet wurde. Zum Einsatz dieser Verfahren in eingebetteten Systemen ist es erforderlich, dass die Hashfunktion einen hinreichend großen Datendurchsatz ermöglicht. In diesem Beitrag wird eine effiziente Hardware-Implementierung der SHA-256 Hashfunktion vorgestellt.

**Schlüsselwörter**—Asymmetrische Kryptosysteme, Digitale Signaturen, Hashing

## I. EINLEITUNG

Die Bedeutung der Informationssicherheit für die Kommunikation nimmt stetig zu. Digitale Signaturen werden zur Verifikation der Integrität sowie der Urheberschaft einer Nachricht benötigt. Für diese Signaturen wird ein Hashwert aus der Nachricht erzeugt mit dem die Integrität überprüft werden kann. Dieser Hashwert wird vom Absender mit einem privaten Schlüssel verschlüsselt, wodurch garantiert werden kann, dass diese Signatur von ihm berechnet wurde.

Der Empfänger kann nun mit dem öffentlichen Schlüssel des Absenders den Hashwert entschlüsseln und aus der Nachricht selber einen Hashwert berechnen. Stimmen die beiden Hashwerte überein, so ist die Nachricht vollständig verifiziert.

In diesem Beitrag wird eine Hardware-Implementierung der Hashfunktion SHA-256 zur Überprüfung der Integrität von digitalen Nachrichten vorgestellt. Dabei werden die Hashwerte in Echtzeit berechnet. Eine Hardware-Implementierung kann durch Parallelisierung einen deutlich höheren Datendurchsatz erreichen als eine Implementierung

J. Freudenberger, jfreuden@htwg-konstanz.de, Institut für Systemdynamik an der Hochschule Konstanz, Alfred-Wachtel-Str. 8, 78462 Konstanz

M. Safieh, msafieh@htwg-konstanz.de, und J. Thiers, jothiers@htwg-konstanz.de sind Mitarbeiter des Instituts für Systemdynamik an der Hochschule Konstanz, Alfred-Wachtel-Str. 8, 78462 Konstanz

Tabelle I  
BEISPIELE FÜR SHA-256 HASHWERTE.

Eingang ASCII	Hashwert in Hexadezimal
wort	22eb019fccc5ca26262ae180643fa1a8f41c05ac5c9a2c69b4d053a04cb0d10d
Wort	0d072b1900825050326794460472052379b52919b5ca1257673953dc93a28981
word	98c1eb4ee93476743763878fcb96a25fbc9a175074d64004779ecb5242f645e6

in Software [1]. Das Ziel dieser Implementierung ist eine möglichst platzsparende Umsetzung des SHA-256 Verfahrens in Hardware, die trotzdem einen ausreichenden Datendurchsatz für die Verwendung in einem Speichercontroller erzielt.

Hashfunktionen erfüllen strengste Vorgaben bezüglich der Sicherheit und dienen dazu, große Datenmenge auf kleinere Hashwerte mit fester Länge abzubilden [1, 2]. Die hierbei verwendeten Rechenvorschriften dürfen nicht umkehrbar sein (Einwegfunktionen). Sie werden so bestimmt, dass eine kleine Änderung der Eingangsdaten zu komplett unterschiedlichen Hashwerten führt. Als Beispiel hierfür sind die Hashwerte für die Eingangsdaten „wort“, „Wort“ und „word“ in Tabelle I aufgeführt.

Im Jahr 2002 wurde das SHA-2 Verfahren (Secure Hash Algorithm 2) vom National Institute of Standards and Technology (NIST) publiziert [3]. Dieser Standard gilt als hinreichend sicheres Verfahren zur Erzeugung eines Hashwertes. Als ein guter Kompromiss zwischen Platzeffizienz und Sicherheit kommt in diesem Beitrag das Verfahren SHA-256 zum Einsatz. Im zweiten Abschnitt dieser Publikation wird auf das Prinzip des SHA-256 Verfahrens eingegangen. Die Hardwareimplementierung dieses Verfahrens sowie die Implementierungsergebnisse werden im Abschnitt III verdeutlicht. Zuletzt werden die Ergebnisse dieser Implementierung im Abschnitt IV diskutiert.

## II. BESCHREIBUNG DES VERFAHRENS

Das Grundprinzip des Hashverfahrens ist als Blockdiagramm in Abbildung 1 gezeichnet. Dabei wird eine Nachricht mit beliebiger Länge in insgesamt drei Blöcken bearbeitet. Zunächst wird die Funktionsweise der einzelnen Blöcke beginnend von links verdeutlicht.

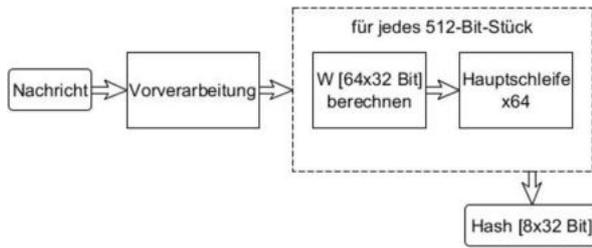


Abbildung 1. Schematischer Ablauf des SHA-256.

Tabelle II  
AUFFÜLLEN EINER 1024 BIT NACHRICHT.

1536 Bit = 3·512 Bit		
1024 Bit	448 Bit	64 Bit
Nachricht	1   0000 ...	1024

### A. Vorverarbeitung der Nachricht

Im ersten Block wird die empfangene Nachricht erweitert bzw. aufgefüllt, sodass die Länge der aufgefüllten Nachricht ein Vielfaches von 512 Bit ist [4] [5]. Danach wird sie in 512 Bit Abschnitten sequenziell zum nächsten Bearbeitungsblock weitergeleitet.

Die erhaltenen Daten können nach dem Auffüllen in drei Felder aufgeteilt werden. Im ersten Feld steht die unveränderte Nachricht. Das zweite Feld wird mit Nullen gefüllt, wobei die Anzahl der gefüllten Bits so bestimmt wird, dass die Gesamtanzahl der Bits aller drei Felder ein Vielfaches von 512 Bit ist. Außerdem wird das MSB des zweiten Feldes beim Auffüllen auf „1“ gesetzt. Im dritten Feld wird die originale Länge der Nachricht vor dem Auffüllen eingetragen, wofür 64 Bit reserviert sind, hierdurch wird auch die Länge der Nachricht auf  $2^{64} - 1$  Bits beschränkt.

Tabelle II ist ein Beispiel für das Auffüllen einer 1024 Bit Nachricht, die direkt in das erste Feld kopiert wird. Im dritten Feld wird die Länge bzw. Anzahl der Bits der Nachricht eingetragen. Die Anzahl der Bits des zweiten Feldes beträgt in diesem Beispiel  $3 \cdot 512 \text{ Bit} - (1024 + 64) \text{ Bit} = 448 \text{ Bit}$ . Somit entstehen drei Abschnitte der Länge 512 Bit, die zum nächsten Bearbeitungsblock weitergeleitet werden.

### B. Berechnung der Hashfunktion

Insgesamt werden 64  $W_i$ -Werte der Länge 32 Bit benötigt. Diese Werte sind für die Berechnung des Hashwertes in der Hauptschleife erforderlich, daher müssen diese vor der Aktivierung der Hauptschleife berechnet werden. Die ersten 16 Werte werden mit den 512 Bit des aktuell bearbeiteten Abschnittes gefüllt. Für die Bestimmung der weiteren Werte wird Gleichung (1) verwendet.

$$W_i = W[i - 16] + s_0 + W[i - 7] + s_1 \quad (1)$$

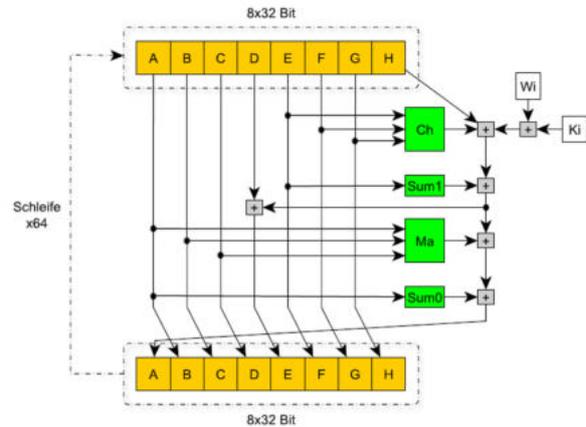


Abbildung 2. Aufbau der Hauptschleife des SHA-256.

Die Variablen  $s_0$  und  $s_1$  sind wie folgt definiert

$$s_0 = (W[i - 15] \ggg 7) \wedge (W[i - 15] \ggg 18) \wedge (W[i - 15] \gg 3)$$

$$s_1 = (W[i - 2] \ggg 17) \wedge (W[i - 2] \ggg 19) \wedge (W[i - 2] \gg 10)$$

wobei  $\gg x$  eine Bitverschiebung und  $\ggg x$  eine Rotation um  $x$  Bits nach rechts bezeichnen.

### C. Hauptschleife

Die Hauptschleife wird für jeden 512 Bit Abschnitt 64-mal mit dem entsprechenden  $W_i$ -Wert durchgeführt, wie es in der Abbildung 2 zu sehen ist. Zusätzlich werden die  $K_i$ -Parameter benötigt, die im Standard definiert sind [3]. Außerdem sind die Variablen A-H für die Berechnung erforderlich, die jeweils eine Länge von 32 Bit haben. Diese werden mit im Standard festgelegten Werten initialisiert und in jedem Durchlauf umgerechnet. In Abbildung 2 sind diese Variablen als gelbe Blöcke dargestellt.

Nach der Bearbeitung eines 512 Bit Abschnittes werden die resultierenden Variablen mit dem temporären Hashwert Exklusiv-Oder verknüpft und für die Berechnung des nächsten Abschnittes mit dem neuen temporären Hashwert beschrieben.

Die grünen Blöcke in Abbildung 2 bezeichnen die Funktionen, mit denen die Variablen A-H in jedem Schleifendurchlauf berechnet werden. Diese Funktionen beschreiben eine logische Verknüpfung der Eingangsvariablen und sind in den Gleichungen (2) bis (5) definiert. Nach 64 Schleifendurchläufen gelten die umgerechneten Variablen A-H als Ausgangsdaten der Hauptschleife.

$$\text{Ch}(E,F,G) = (E \wedge F) \wedge (\sim E \wedge G) \quad (2)$$

$$\text{Ma}(A,B,C) = (A \wedge B) \wedge (A \wedge C) \wedge (B \wedge C) \quad (3)$$

$$\text{sum}_0(A) = (A \ggg 2) \wedge (A \ggg 3) \wedge (A \ggg 22) \quad (4)$$

$$\text{sum}_1(E) = (E \ggg 6) \wedge (E \ggg 11) \wedge (E \ggg 25) \quad (5)$$

### III. IMPLEMENTIERUNG DER SHA-256

Der SHA-256 Algorithmus arbeitet intern mit 32-Bit-Wörtern. Daher wurden Ein- und -Ausgang der Hashfunktion mit dieser Breite gewählt. Somit wird ein 512-Bit-Block der Eingangsdaten in 16 Takten übergeben. Ist ein 512-Bit-Block fertig verarbeitet, wird der *Finished*-Ausgang gesetzt, um zu signalisieren, dass neue Werte entgegengenommen werden können.

Wenn keine Eingangsdaten mehr vorhanden sind, wird der *lastChunk*-Eingang gesetzt, der dafür sorgt, dass das Modul die Eingangsdaten auffüllt und nach Verarbeitung des letzten Blockes den Hashwert in acht Takten zurückgibt. Dieser Aufbau ist in Abbildung 3 schematisch dargestellt. Der *ChipSelect*-Eingang dient dem Aktivieren beziehungsweise Deaktivieren des Moduls.

Das Top-Modul besteht aus zwei Teilen, die aber zusammen in einem Modul implementiert wurden, um Register und Logik zu sparen. Dies ist zum einen die Berechnung der  $W_i$ -Werte und das Auffüllen der Nachricht (siehe Abschnitt II.A und II.B) und zum anderen die Hauptschleife (Abschnitt II.C). Diese beiden Teile sind lediglich durch einen der  $W_i$ -Werte sowie den temporären Hashwert gekoppelt.

Das Top-Modul besitzt sechs Untermodule (in Abbildung 3 grün dargestellt), sowie zwei Registerbänke. Die Funktionen *S0* und *S1* werden für die Berechnung der  $W_i$ -Werte benötigt und werden in Abschnitt II.B näher erläutert. Die Untermodule *Ch*, *Sum0*, *Sum1* und *Ma* sind die Funktionen der Hauptschleife, die in Abbildung 2 zu sehen sind. Diese sind durch die Gleichungen (2) bis (5) definiert.

Zur Speicherung des temporären Hashwertes werden 256 Register benötigt, die in acht Blöcke mit je 32 Registern aufgeteilt werden. Nachdem die aufgefüllte Nachricht bearbeitet wurde, steht in diesen Registern der endgültige Hashwert.

Die 64  $W_i$ -Werte werden, wie in Gleichung (1) gezeigt, aus ihren Vorgängerwerten berechnet und anschließend nacheinander in die Hauptschleife übergeben. Durch eine parallelisierte Berechnung der  $W_i$ -Werte und der Hauptschleife kann sowohl Platz als auch Zeit gespart werden, da für die Berechnung der  $W_i$ -Werte lediglich die letzten 17 Werte benötigt werden und somit auch nur diese gespeichert werden müssen. Deshalb werden, wie in Abbildung 3 gezeigt, anstatt  $64 \times 32$  Bit (siehe Abschnitt II.B) nur  $17 \times 32$  Bit Register benötigt. Hierbei wird immer nur in den höchsten  $W_i$ -Wert geschrieben und auch immer dieser an die Hauptschleife weitergegeben, wodurch auch Logik für die Ansteuerung eingespart werden kann. Die restlichen  $W_i$ -Werte werden lediglich jeden Takt um eine Position weitergeschoben.

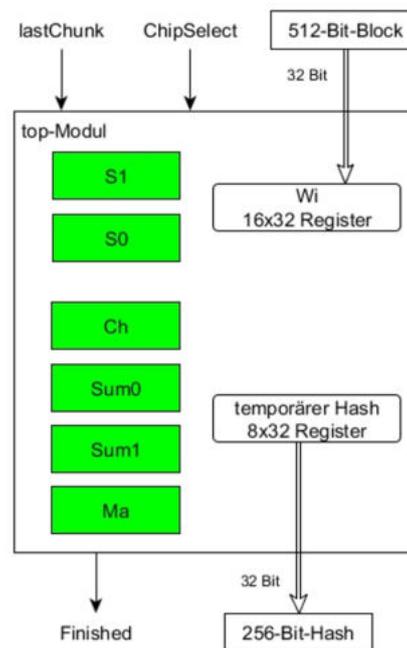


Abbildung 3. Skizze der Implementierung des SHA-256.

Tabelle III  
RESSOURCEN DER FPGA-IMPLEMENTIERUNG.

LUT	Register	Frequenz
1127	762	160 MHz

### IV. ERGEBNISSE

Tabelle III gibt die benötigten Hardware-Ressourcen für die Implementierung des SHA-256 Verfahrens an, d.h. die Anzahl der Look-Up-Tabellen (LUT) und Register sowie die Betriebsfrequenz auf dem Xilinx Virtex 7 FPGA.

Der erzielte Datendurchsatz ist von der Länge der Eingangsdaten abhängig, da das Auffüllen zu einem Overhead in der Bearbeitungszeit führt. Die Verarbeitung eines 512-Bit-Blockes benötigt 74 Takte, wobei während der ersten 16 Takte die Daten gleichzeitig eingelesen wird. Der erzielte Datendurchsatz ist für beispielhafte Längen der Eingangsdaten in Tabelle IV aufgeführt. Bei Anwendungen in Speichersystem werden aber keine Blocklängen kleiner 8192 Bit unterstützt, daher kann einen Durchsatz von ca. 130 Mbyte/s erreicht werden.

Die Anzahl der benötigten Register konnte gegenüber der Implementierung nach [6] mehr als halbiert werden, während die Datenrate selbst bei 1024 Bit Eingangsdatenlänge mehr als verdoppelt wurde. Gegenüber der Implementierung aus [1] konnte bei 1024 Bit der Durchsatz um knapp 50% gesteigert werden.

Tabelle IV  
DATENDURCHSATZ FÜR VERSCHIEDENE DATENMENGEN.

Länge der Eingangsdaten [Bit]	Datendurchsatz [Mbit/s]
1024	738,4
2048	885,6
4096	984
8192	1041,6



**Jürgen Freudenberger** ist seit 2006 Professor für Kommunikationssysteme an der Hochschule Konstanz. Dort leitet er das Informations- und Medienzentrum. Seine Forschungsarbeit beschäftigt sich vorrangig mit der Entwicklung von Algorithmen im Bereich der Signalverarbeitung und der Codierung für zuverlässige Datenübertragung sowie mit der effizienten Umsetzung der Verfahren in Hard- und Software.

## V. ZUSAMMENFASSUNG

In diesem Beitrag wurde eine effiziente Hardware-Implementierung der SHA-256 Hashfunktion vorgestellt. Zum Einsatz digitaler Signaturen in eingebetteten Systemen ist es erforderlich, dass die Hashfunktion einen hinreichend großen Datendurchsatz ermöglicht. Die Implementierung auf einem Xilinx Virtex 7 FPGA erzielt einen Datendurchsatz von rund 130 Mbyte/s und ist mit einer Größe von rund 1100 LUT und 800 Registern sehr kompakt.



**Malek Safieh** erhielt den akademischen Grad des B.Eng. in Elektro- und Informationstechnik im Jahr 2015 sowie den Grad des M.Eng. in Elektrische Systeme im Jahr 2017 von der HTWG-Konstanz. Er ist wissenschaftlicher Mitarbeiter am Institut für Systemdynamik der Hochschule Konstanz.

## LITERATURVERZEICHNIS

- [1] K. K. Ting, S. C. L. Yuen, K. H. Lee und P. H. W. Leong, *An FPGA based SHA-256 Processor*, Glesner M., Zipf P., Renovell M. (eds) Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream. FPL 2002., 16 August 2002.
- [2] J. Wolf., *C von A bis Z*, Galileo Press GmbH. ISBN: 3836214113, 2009.
- [3] N. I. o. S. a. Technology, *Secure Hash Standard, Federal Information Processing Standard (FIPS), FIPS 180-2*, Techn. Ber. DEPARTMENT OF COMMERCE, Aug. 2002.
- [4] R.-N. J. P. S. Bakhtiari, *Cryptographic Hash Functions: A Survey*, Technical Report 95-09, Department of Computer Science, University of Wollongong, July 1995.
- [5] N. I. o. S. a. Technology, *Secure Hash Standard, Federal Information Processing Standard (FIPS), FIPS 180-4*, Techn. Ber. DEPARTMENT OF COMMERCE, Aug. 2015.
- [6] O. K. N. Sklavos, „On the hardware implementation of the SHA-2 (256, 384, 512) Hash functions“, *Proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'03)*, Vol. V, Mai 2003.



**Johann-Philipp Thiers** erhielt den akademischen Grad des B.Eng. in Elektro- und Informationstechnik im Jahr 2017 von der HTWG-Konstanz. Er ist wissenschaftlicher Mitarbeiter am Institut für Systemdynamik der Hochschule Konstanz.



# CAN FD Signalmanipulationsschaltung auf FPGA-Basis

Marcel Maier, Stefan Buntz, Lothar Schmidt, Anestis Terzis

**Zusammenfassung**—Die Automobilhersteller sind dabei, die Datenrate von CAN zu erhöhen. Dieser Artikel gibt einen Überblick über ein FPGA-basiertes Entwicklungsboard für ein CAN-Kommunikationssystem im Automotive-Bereich, um die Signalintegrität bei einer Erhöhung der Datenrate (CAN FD) sicherzustellen.

Für dieses Entwicklungsboard wurde eine Signalmanipulationsschaltung in der Hardwarebeschreibungssprache VHDL entworfen, mit welcher CAN-Kommunikationsnachrichten vor der Übertragung beliebig manipuliert und verändert werden können. Damit die Signalmanipulationsschaltung über einen Mikrocontroller oder Computer ansteuerbar ist, wurde ein SPI-Interface implementiert und es steht eine Windows Benutzeroberfläche zur Verfügung. Diese Schaltung erlaubt eine Untersuchung des Kommunikationssystems bezüglich der Robustheit gegenüber fehlerhaften Bits.

## I. EINLEITUNG UND AUFGABENSTELLUNG

Zur Bereitstellung neuer Funktionalitäten im Automobil werden verschiedene Vernetzungstechnologien im Fahrzeug verwendet, so dass die zur Verfügung stehenden Bandbreiten immer mehr zunehmen. Ein Beispiel hierfür ist automotive Ethernet mit einer Datenrate von 100 Mbit/s (IEEE802.3 Std. 100BASE-T1). Eine dagegen seit Jahrzehnten im Kfz verwendete, weit verbreitete und sehr gut etablierte Vernetzungstechnologie ist der CAN-Bus. Aufgrund seiner sehr guten Eignung für automobiler Anwendungen ist der CAN-Bus auch heutzutage aus dem Technologieportfolio eines modernen Fahrzeugs nicht wegzudenken. Jedoch besteht zwischen der Datenrate des klassischen CAN (bis zu 500 kbit/s) und automotive Ethernet (ab 100 Mbit/s), bzw. FlexRay (10 Mbit/s) eine Lücke, welche durch die Erweiterung des CAN-Protokolls zu CAN FD (CAN flexible datarate) teilweise gefüllt werden kann. Hierbei wird durch eine Erhöhung der Baudrate innerhalb des Datenfeldes des CAN-Rahmenformats und eine Vergrößerung des zulässigen Nutzdatenfeldes eine größere mittlere Datenrate erreicht [1]. Seit 2011 arbeitet die Automobilindustrie an der Entwicklung von CAN FD um eine für die zukünftigen Anforderungen passende Vernetzungstechnologie zu schaffen. Da CAN FD im Datenfeld mit einer Datenrate von bis zu 2 Mbit/s, bzw. teilweise sogar mit bis zu 5 Mbit/s betrieben

Marcel Maier, mamaier@hs-ulm.de, Lothar Schmidt, schmidt@hs-ulm.de, Anestis Terzis, terzis@hs-ulm.de, Hochschule Ulm, Pritwitzstraße 10, 89075 Ulm.

Stefan Buntz, stefan.buntz@daimler.com, Daimler AG, Wilhelm-Runge-Str. 11, 89081 Ulm.

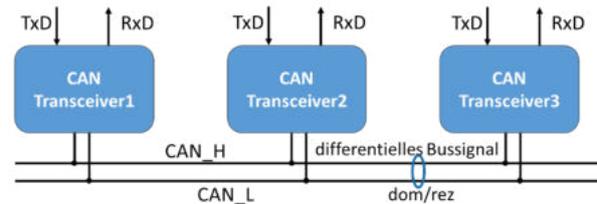


Abbildung 1. Anschluss Steuergeräte-Transceiver an CAN-Bus.

werden kann [2], bietet sich diese Lösung für Anwendungen, deren Anforderungen bezüglich Datenrate oberhalb von 500 kbit/s liegen, aber kein FlexRay oder gar Ethernet benötigen, an. Solche Anwendungen sind auch heute noch weit verbreitet im Fahrzeug. Ein Beispiel ist die Übertragung von Objektdaten eines Radarsensors für autonome Fahrfunktionen. Solange keine Rohdaten benötigt werden, genügen oft die durch CAN FD bereitgestellten Bandbreiten.

Durch die Erhöhung der Baudrate im Datenfeld des CAN FD Rahmens kommt es jedoch im Bereich des Abtastzeitpunktes des CAN Signals zu sogenannten Glitches, die eine Fehlinterpretation eines Bits verursachen können. Um die Fehlertoleranz von CAN FD Netzwerken gegenüber solchen Glitches zu untersuchen wurde in [3] eine Schaltung entwickelt mit der solche Glitches gezielt in die CAN Kommunikation eingefügt werden können.

## II. CAN FD

Die Grundlagen des CAN-Buses sind beispielsweise in [4] nachzulesen. Die Kommunikation zwischen den Steuergeräten findet über spezielle CAN-Transceiver und einen differenziell ausgeführten CAN-Bus statt (vgl. Abbildung 1). Sobald Daten durch ein Steuergerät gesendet werden, das heißt ein Transceiver einen Transmitter-exchange-Data-Befehl (TxD-Befehl) erhält, wird das rezessive CAN-Bussignal durch ein dominantes Start-of-Frame-Bit (SOF-Bit) auf dominanten Pegel gesetzt. Ab diesem Zeitpunkt empfangen alle an den CAN-Bus angeschlossenen Steuergeräte über ihre Transceiver bzw. das Receiver-exchange-Data-Signal (RxD-Signal) die gesendeten Daten.

In den folgenden Abschnitten werden die für diese Arbeit relevanten Details zusammengefasst.

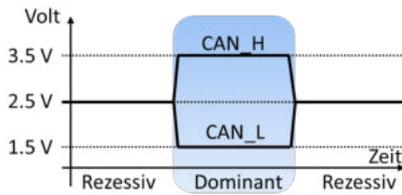


Abbildung 2. Dominanter und rezessiver Pegel des CAN Busses.

### A. CAN FD – Bit-Zustände und Rahmenformat

Die für diese Arbeit relevanten Punkte des CAN Systems sind die differentielle Übertragungstechnik über 2 verdrehte Adern und die Unterscheidung von „dominanten“ und „rezessiven“ Bits. Die beiden Signale auf den Adern werden typischerweise mit „CAN\_high“ (CAN\_H) und „CAN\_low“ (CAN\_L) bezeichnet. Die Unterscheidung von dominantem und rezessivem Bit, die für die Arbitrierung (Priorisierung) der CAN-Botschaften verwendet wird lässt sich wie folgt erläutern (siehe auch Abbildung 2):

- Rezessives Bit: Der rezessive Zustand ist ein passiv (über Widerstand) eingepprägter Spannungspegel von 2,5 V auf beiden Signalleitungen (CAN\_H und CAN\_L).
- Dominantes Bit: Der dominante Zustand ist ein aktiv durch den CAN Transceiver auf den Bus eingepprägter Spannungspegel von 3,5 V (CAN\_H), bzw. 1,5 V (CAN\_L).

Durch diese unterschiedliche „Wertigkeit“ der beiden Zustände lässt sich eine Priorisierung der Botschaften erreichen, da jeder Sender den rezessiven Zustand des CAN-Busses durch ein dominantes Bit überschreiben kann.

Da hierbei jedoch die Ausbreitungsgeschwindigkeit auf der Leitung beachtet werden muss, ist die Datenrate im Arbitrierungsfeld des CAN-Rahmenformates (siehe Abbildung 3 [3]) durch die maximale Leitungslänge des CAN-Busses limitiert. Typischerweise liegt hier die maximale Datenrate für Verkablungen im Kfz bei maximal 500 kbit/s.

Wie bereits erwähnt bietet CAN FD im Vergleich zum klassischen CAN jedoch zusätzlich die Möglichkeit im Datenfeld des CAN-Rahmenformates sowohl die Anzahl an enthaltenen Bits zu erhöhen (von 64 bit auf bis zu 512 bit), als auch die Möglichkeit die Datenrate zu erhöhen (bis zu 2 Mbit/s, bzw. 5 Mbit/s).

### B. Ringing Effekt und Auswirkung bei CAN FD

Durch die Verwendung eines rezessiven Zustandes lässt sich geschickt eine Priorisierung und verlustfreie Arbitrierung von Botschaften auf dem CAN-Bus erreichen, jedoch entsteht hierdurch auch ein Nachteil, bedingt durch die physikalische Ausdehnung des Busses. Durch die schwache (resistive) Erzeugung des rezessiven Signalpegels, wird dieser durch Einflüsse der Bus-topologie (Reflexion an Impedanz Stoßstellen wie z.B.

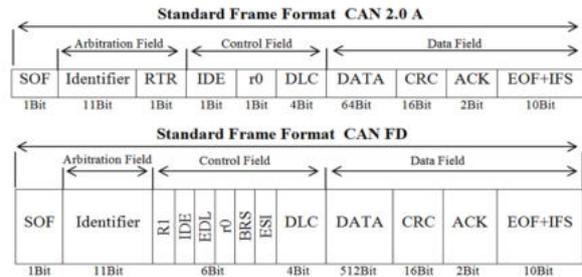


Abbildung 3. Vergleich zwischen klassischem CAN und CAN FD.

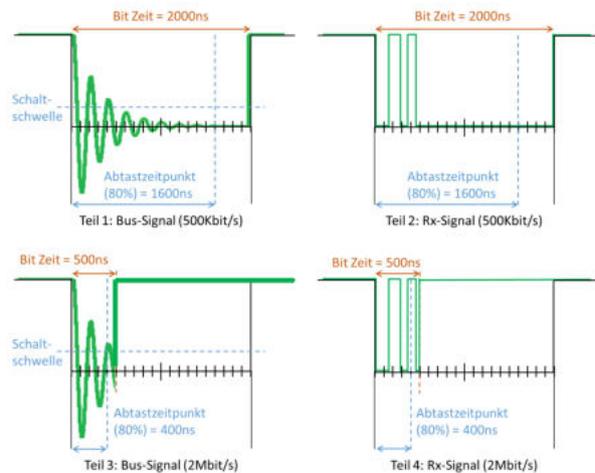


Abbildung 4. Einfluss des Ringings auf Bits und das Rx-Signal bei 500 kbit/s und bei 2 Mbit/s.

Stubs oder Splices) stark beeinflusst und schwankt bei vielen in Fahrzeugen vorkommenden realen Topologien stark um den eigentlichen Soll-Spannungspegel von 2,5 V. Diese Schwankungen der Busspannung führen im Transceiver zu Fehlinterpretationen des Bussignals und damit zu fehlerhaften Zuständen/Pegeln des Rx-Signals des Transceivers.

Dieses sogenannte „Ringing“ (dt. „klingeln“) ist unabhängig von der Datenrate. Bei einem CAN Signal mit einer Datenrate von 500 kbit/s (Bitdauer 2000 ns) spielt sich der größte Teil des Ringings typischerweise innerhalb der ersten 25 % der Bitzeit (bis ca. 500 ns) ab. Deshalb wird typischerweise der Abtastzeitpunkt (Digitalisierung des Rx-Datensignals) innerhalb des Bits in das letzte Viertel des Bits gelegt (in Abbildung 4 - Teil 2 bei 80 % der Bitzeit) und ist damit sehr robust gegenüber dem Ringing und den durch das Ringing entstehenden Glitches (fehlerhaftes Rx-Signal). Wird aber bei CAN FD innerhalb des Datenfeldes die Baudrate auf z.B. 2 Mbit/s erhöht, so verkürzt sich die Bitzeit auf 500 ns (Abbildung 4 – Teil 3) und das Ringing ist über die ganze Bitdauer stark vorhanden, dadurch entstehen im Rx-Signal Fehler/Glitches, die auch beim Abtasten (bei 400 ns) vorhanden sein können.

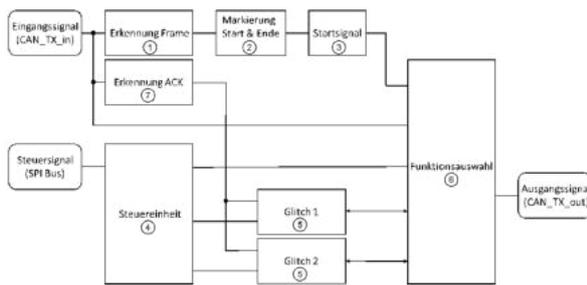


Abbildung 5. Blockschaltplan Anwendungsschaltung [3].

### III. SIGNALMANIPULATIONSSCHALTUNG

Damit CAN FD Topologien auf die Robustheit gegenüber Bitfehler überprüft werden können, wurde in [3] eine aufsteckbare FPGA-basierte Entwicklungsplatine für die CAN FD Steuergeräte entwickelt.

#### A. Board Realisierung

Die Entwicklungsplatine ist mit einem Xilinx FPGA der Familie Spartan 6 (XC6SLX9) ausgestattet. Durch eine FPGA-basierte Ausführung sind eine maximale Flexibilität und eine Änderbarkeit der gewünschten Funktion möglich. Zudem entsteht eine einfache Entwicklungs- und Testumgebung für digitale Schaltungen.

Der FPGA ist an alle Signal Ein- und Ausgänge der CAN FD Steuergeräte angeschlossen, sobald die Boards miteinander verbunden werden. Dadurch besteht eine Verbindung zur SPI-Schnittstelle des CAN FD Mikrocontrollers, wodurch eine externe Ansteuerung der Signalmanipulationsschaltung möglich ist.

Die Konfiguration des FPGAs ist über die USB-Schnittstelle oder den JTAG-Anschluss möglich [3].

#### B. Realisierung FPGA-Schaltung

Die Realisierung der FPGA-Schaltung ist in zwei Teile unterteilt. Es wurde eine Anwendungsschaltung für den Hardwarebasierten FPGA-Entwurf zur Signalmanipulation in VHDL entworfen (vgl. Abbildung 5). Außerdem entstand zur Ansteuerung der Anwendungsschaltung über den Mikrocontroller (SPI-Schnittstelle) eine grafische Bedienoberfläche, welche mit CAPL über CANoe programmiert wurde.

Die in der Hardwarebeschreibungssprache VHDL entworfene Anwendungsschaltung, welche die Signalmanipulation in einer CAN-Nachricht durchführt, ist in mehrere Teilprojekte unterteilt. Die Xilinx Entwicklungsumgebung ISE (Integrated Synthesis Environment) ermöglicht es, aus VHDL-Code Komponenten mit In- und Outputs zu generieren, die über Symbole in einem Schaltplan (Schematic) verwendet werden können. Damit lassen sich verschiedene Projekte durch ein sogenanntes Schematic-Design zusammenfassen.

Die Kommunikation über den CAN-Bus besteht aus vielen nacheinander gesendeten seriellen Nachrichten. Das bedeutet, dass der serielle Datenstrom

zwei Zustände hat, zwischen welchen ständig hin und her geschaltet wird (vergleichbar mit vielen aneinander gereihten Einsen und Nullen). Damit die Signalmanipulation durchgeführt werden kann besteht die Anwendungsschaltung aus folgenden Teilkomponenten (Abbildung 5).

1. **Erkennung Frame:**  
Zur Erkennung der CAN-Nachrichten im seriellen Datenstrom werden im ersten Teil die Flankenwechsel mit einer Flankenerkennung detektiert.
2. **Markierung Start & Ende:**  
Ein retriggerbarer Zähler der  $11 \mu\text{s}$  lang zählt, markiert den Start und das Ende einer Nachricht, indem er die detektierten Flankenwechsel erhält (vgl. Abb. 8). Innerhalb einer CAN-Nachricht wird nach fünf gleichen Bits ein sogenanntes Stuffbit eingefügt. Aus diesem Grund findet auch im langsamen Arbitrierungsfeld ( $500 \text{ kbit/s}$ ) mindestens alle  $10 \mu\text{s}$  ein Flankenwechsel statt ( $11 \mu\text{s} - 10 \mu\text{s} = 1 \mu\text{s}$  Reserve).
3. **Startsignal:**  
Am Anfang jeder CAN-Nachricht wird ein kurzer Startimpuls erzeugt, welcher die programmierbaren Zähler in den Blöcken Glitch 1 und Glitch 2 startet.
4. **Steuereinheit:**  
Die Steuereinheit erhält über den Mikrocontroller Daten, welche die Funktion und die Zählzeit der programmierbaren Zähler bestimmt.
5. **Glitch 1 und Glitch 2:**  
Diese Blöcke generieren an beliebiger Stelle mit beliebiger Länge in der CAN-Nachricht einen Glitch. Einstellbar über die programmierbaren Zähler.
6. **Funktionsauswahl:**  
Die Funktionsauswahl entscheidet anhand der Steuerdaten, welche Glitchart (rezessiv, dominant, konstant) verwendet wird und die Anzahl der verwendeten Glitche (ein oder zwei).
7. **Erkennung ACK:**  
Wenn ein Bestätigungsbit (Acknowledge-Bit) erkannt wird, werden die programmierbaren Zähler zurückgesetzt und der Glitch unterdrückt [3].

#### C. Emulation von Bitfehlern

Die Emulation beliebiger Bitfehler ist über eine Bedienoberfläche, welche mit der CAPL (Communication Access Programming Language) der Entwicklungsumgebung CANoe entworfen und programmiert wurde, möglich. Abbildung 6 zeigt beispielhaft eine manipulierte CAN-Nachricht.

Dazu wird die originale CAN-Nachricht (Tx-Signal, Abbildung 6, blau Kurve bzw. `can_tx_in`), bevor sie über den CAN-Bus gesendet wird, der Signalmanipulationsschaltung übergeben. Wie unter B. beschrieben wird, findet zuerst eine Analyse der CAN-Nachricht

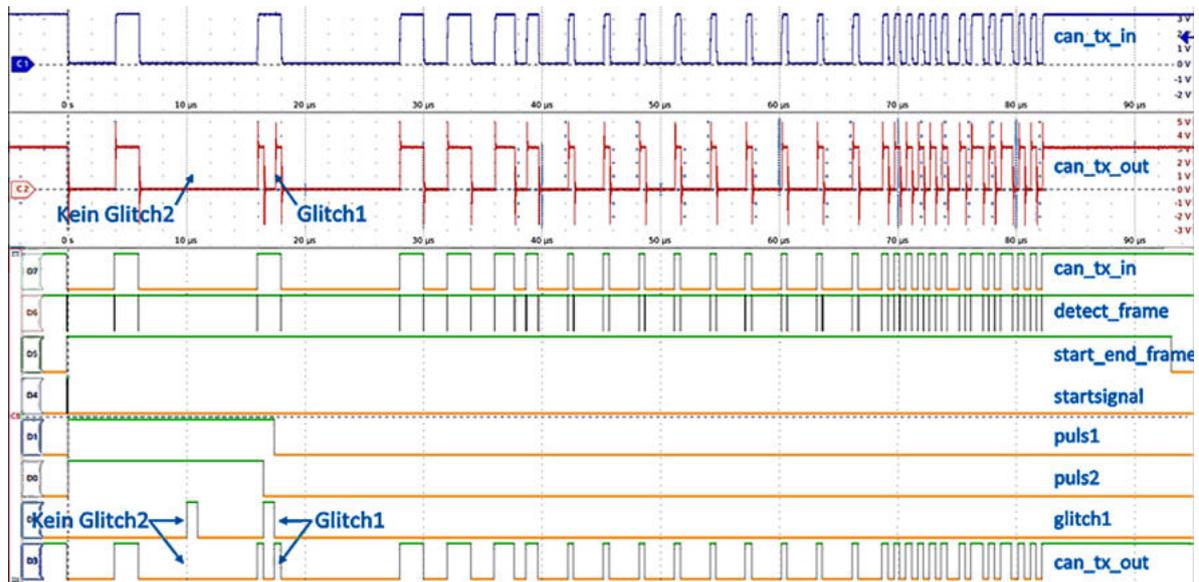


Abbildung 6. Messung einer manipulierten CAN-Nachricht – Glitchart: „dominanter Glitch“, Anzahl Glitche: „Zwei“ [3].

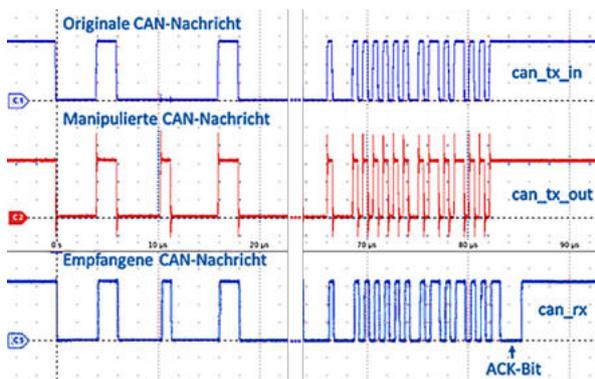


Abbildung 7. Messung „Übertragung Glitch über CAN-Bus“ [3].

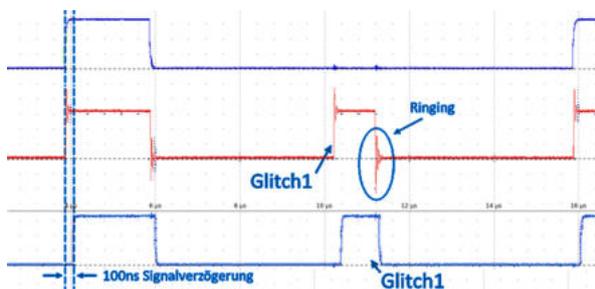


Abbildung 8. Messung „Übertragung Glitch“ vergrößert [3].

statt. Nachdem die Flanken detektiert (detect\_frame) und die Nachricht markiert ist (start\_end\_frame), startet der Startimpuls zwei Zähler. Diese Zähler sind mit unterschiedlichen Zählzeiten geladen und erzeugen dadurch unterschiedlich lange High-Pegel am Ausgang (Puls1 und Puls2). Durch die unterschiedlich lange Laufzeit ist es möglich einen Glitch an beliebiger Stelle mit beliebiger Länge, nach Start der Nachricht, zu platzieren. Aus Puls1 und Puls2 wird Glitch1 erzeugt.

Puls3 und Puls4 sind aus Platzgründen nicht dargestellt, erzeugen allerdings Glitch2.

Für die gezeigte Messung ist die Funktion „dominanter Glitch“ ausgewählt. Diese Art von Glitch wirkt sich nur auf einen rezessiven Pegel aus, weshalb nur der Glitch1 eine Auswirkung auf die originale CAN-Nachricht hat (vgl. Abbildung 6, rote Kurve bzw. can\_tx\_out) [3].

#### D. Anwendung

Um die Übertragung einer manipulierten CAN-Nachricht über den CAN-Bus darzustellen, zeigen die Messungen in Abbildung 7 und Abbildung 8 die originale CAN-Nachricht (oberer blauer Signalverlauf) und die manipulierte CAN-Nachricht (mittlerer roter Signalverlauf) eines Teilnehmers, bevor sie durch den Transceiver gesendet wird. Der untere blaue Signalverlauf zeigt die Empfangene CAN-Nachricht, nachdem diese von einem Transceiver eines zweiten Teilnehmers empfangen wurde. Anhand des ACK-Bit am Ende des CAN-Frames (Abbildung 7) und der Signalverzögerung (Abbildung 8) ist erkennbar, dass der untere blaue Signalverlauf eine empfangene Nachricht darstellt.

Der obere blaue und mittlere rote Signalverlauf wurde bereits in der vorherigen Messungen (Abbildung 6) dargestellt. Wenn diese beiden kontinuierlichen Analogsignale miteinander verglichen werden, ist zusätzlich zum Glitch der Signalmanipulationsschaltung ein sogenanntes Ringing an den Signalfanken erkennbar (vgl. Abbildung 8, mittlerer roter Signalverlauf).

Ringing entsteht durch Reflexionen des Signals auf der Kommunikationsleitung und gilt in CAN FD-Netzwerken als Schlüsselherausforderung. Reflexionen treten aufgrund von Impedanz-Fehlanpassungen in einem Netzwerk auf. Das Ringing ist aufgrund spezi-

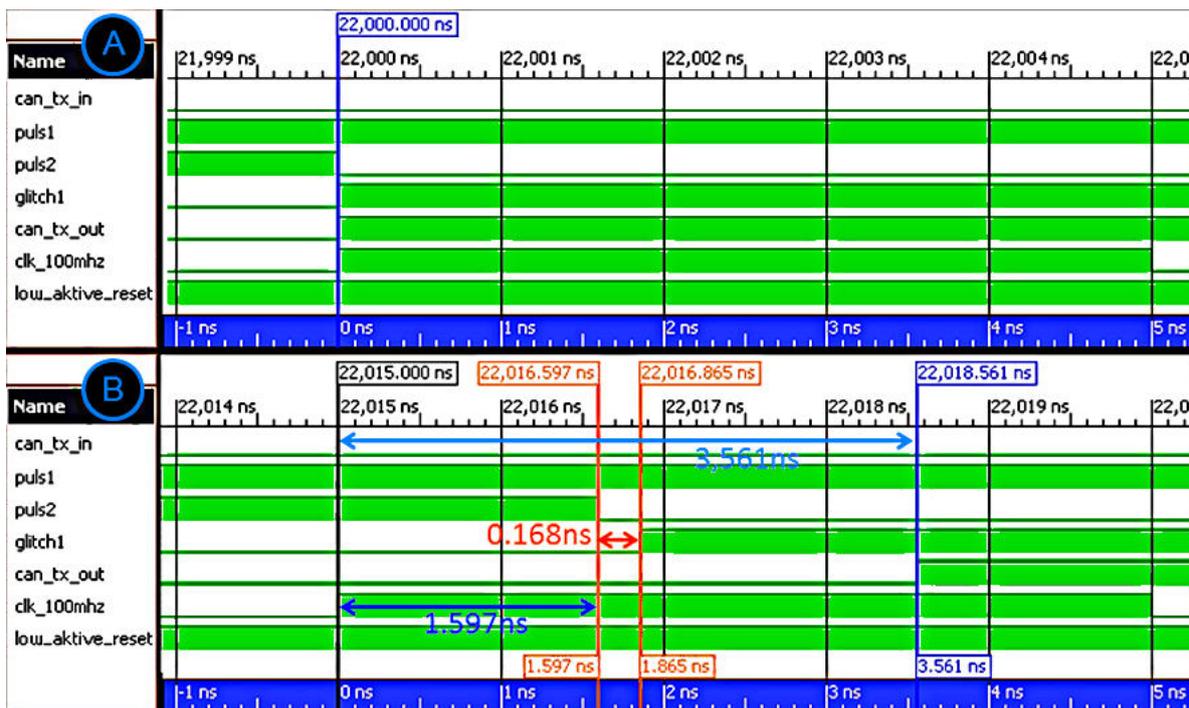


Abbildung 9. Verhaltenssimulation (A), Timing-Simulation (B).

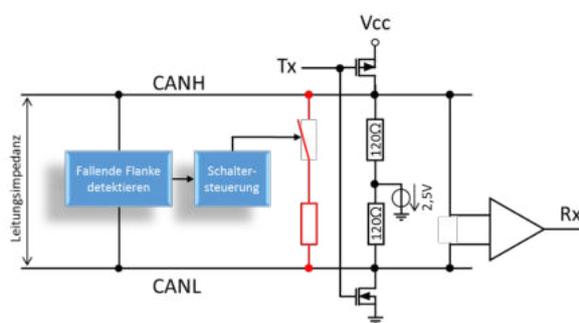


Abbildung 10. RSC-Technologie.

FPGA_Glitch_Schematic			
Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Register	368	11440	3%
Number of Slice LUTs	520	5720	9%
Number of occupied Slices	165	1430	11%
Number of MUXCYs used	144	2860	5%
Number of bonded IOBs	15	102	14%
Number of BUFG/BUFGMUXs	2	16	12%

Abbildung 11. Verwendete und verfügbare Logik-Ressourcen des Spartan 6 FPGAs.

eller RSC-Transceiver (Ringing-Suppression-Circuitry-Transceiver) auf der Empfangsseite nicht zu erkennen (vgl. Abbildung 8, unterer blauer Signalverlauf). Das grundlegende Prinzip der RSC-Schaltung stellt Abbildung 10 dar. Die spezifizierte Schaltung ändert zur Unterdrückung des Ringings für eine kurze Zeit die

Gesamtimpedanz. Dies erfolgt bei der Zustandsänderung eines dominanten auf einen rezessiven Pegel.

Die Dauer der Änderung ist von der gewählten Bitrate abhängig. Der standardisierte Abtastzeitpunkt für CAN FD-Nachrichten ist wie bereits erwähnt zu 80% der Bitzeit definiert (vgl. Abbildung 4). Das Ringing begrenzt die maximale Bitrate aufgrund der schlechteren Signalintegrität.

Durch das Ringing in CAN FD-Netzwerken entsteht eine Anwendung für die Signalmanipulationsschaltung. Bei Messungen und Simulation verschiedener CAN FD-Topologien können diese Störimpulse (vgl. Abbildung 4 – Teil 2 und 4) „manuell“ erzeugt werden, indem über die Glitch-Schaltung in den CAN FD-Sendepfad (TX) zwei Glitche eingefügt werden.

Das FPGA-Entwicklungsboard mit konfigurierter Glitch-Schaltung, kann somit z.B. speziell zum Test der RSC-Transceiver eingesetzt werden [3].

#### IV. ERGEBNISSE

In Abbildung 9 sind ein vergrößerter Ausschnitt der Verhaltenssimulation (A) und der Timing-Simulation (B) dargestellt (Xilinx Simulator). Die Timing-Simulation visualisiert die mögliche Verhaltensweise für eine hohe Umgebungstemperatur und die geringste interne Betriebsspannung (worst case). Beim Vergleich der Simulationen könnten durch die Timing-Simulation die Verzögerungen von einzelnen internen Signalpfaden bestimmt werden. Die Gesamtverzögerung im „worst case“ beträgt 18,561 ns und ist in Bezug auf die Datenrate von CAN FD vernachlässigbar



klein. Nach erfolgreicher Implementierung des VHDL-Designs konnten die benötigten Logik-Ressourcen, welche in Abbildung 11 zusammengefasst sind, mit der Xilinx Software ermittelt werden.

## V. FAZIT

Im Rahmen dieser Arbeit entstand eine FPGA-Entwicklungsumgebung für verschiedene Test- und Manipulationsschaltungen, welche zur Absicherung der Signalintegrität in CAN FD-Netzwerken eingesetzt werden. Hierfür existiert nun eine spezielle aufsteckbare FPGA-Leiterplatte für die CAN FD-Steuergeräte der Firma Daimler AG. Die im FPGA implementierte ansteuerbare Signalmanipulationsschaltung ermöglicht Bitmanipulationen an beliebigen Stellen in einer CAN FD-Nachricht. Die Schaltung ist benutzerfreundlich über eine grafische Benutzeroberfläche ansteuerbar [3].



**Marcel Maier** erhielt den akademischen Grad Bachelor of Engineering (B.Eng) in Elektro- und Informationstechnik im Jahr 2018 von der Hochschule Ulm.



**Stefan Buntz** arbeitet seit seiner Promotion im Jahr 2008 bei der Daimler AG in Ulm. Dort arbeitet er an der Entwicklung neuer Bussysteme und Kommunikationsnetze. Ein besonderer Schwerpunkt seiner Arbeit lag und liegt auf den Themen Physical Layer und EMV von Automotive-Kommunikationssystemen. Insbesondere das Thema Automotive Ethernet ist seit Anfang 2008 eines seiner Kerngebiete. In den Jahren 2004 bis 2008 promovierte Hr. Buntz an der Universität Ulm mit dem Thema „Bauteil-Störfestigkeitsprüfverfahren für Kfz-Bussysteme“, wo Hr. Buntz auch seit 1998 studierte.

## DANKSAGUNGEN

Die Bachelorarbeit zu dieser Veröffentlichung entstand im Daimler Forschungszentrum Ulm in der damaligen Abteilung „Absicherung Hard- und Software Qualität“ während des Sommersemesters 2018.

Meinen besonderen Dank möchte ich dabei Herrn Dr.-Ing. Stefan Buntz für die Übernahme der Betreuung aussprechen. In gleicher Weise bedanke ich mich bei Herrn Prof. Dr.-Ing. Anestis Terzis für die Übernahme der Betreuung seitens der Hochschule Ulm. Weiterhin möchte ich besonders Herrn Prof. Dr. Lothar Schmidt für die Übernahme der Zweitkorrektur danken.

Für das Korrekturlesen möchte ich mich bei meiner Freundin Corinna Keller und meinem Kollegen Emanuel Panholzer recht herzlich bedanken.

## LITERATURVERZEICHNIS

- [1] Holger Zeltwanger, „CAN FD - schnell und zuverlässig“ Artikel in *Elektroniknet*: URL: <https://www.elektroniknet.de/elektronik-automotive/bordnetz-vernetzung/can-fd-schnell-und-zuverlaessig-105590.html>, 2014.
- [2] CAN FD ISO 11898-1:2015-12, *Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling..*
- [3] M. Maier, *Entwicklung einer FPGA-basierten Signalmanipulationsschaltung für serielle Datenübertragungssysteme*. Unveröffentlichte Bachelorarbeit, Hochschule Ulm, 2018.
- [4] W. Zimmermann, R.Schmidgall, „Bussysteme in der Fahrzeugtechnik,“ *Springer Fachmedien*, Wiesbaden 2014.



**Lothar Schmidt** ist seit 2016 als Professor für Elektronik an der Hochschule Ulm tätig. Sein Lehrgebiet umfasst elektronische Bauelemente, integrierte analoge und mixed-signal Schaltungen sowie Bauelemente und Schaltungen der Leistungselektronik. Zuvor besetzte er unterschiedliche leitende Funktionen überwiegend im Bereich Chipentwicklung unter anderem bei Maximum Integrated Products, Texas Instruments und zuletzt bei Micronas. Die Entwicklungen umfassten unter anderem Multi-Gbps-Chipsets für optische Datenübertragung, Direct-Conversion Transceiver für drahtlose Kommunikationstechnik sowie Sensoren und Embedded Controller für Automotive-Applikationen. Lothar Schmidt ist Autor und Co-Autor zahlreicher internationaler Veröffentlichungen und Patente. Er beendete sein Studium der Elektrotechnik an der Ruhr-Universität in Bochum 1987 als Dipl.-Ing. und promovierte ebenda 1993 im Bereich monolithisch integrierte Gbps-Schaltungen. Lothar Schmidt ist seit 1999 Mitglied im IEEE.



**Anestis Terzis** ist seit 2012 Professor für den Entwurf digitaler Systeme an der Hochschule Ulm und leitet den Studiengang Fahrzeugelektronik. Zuvor arbeitete er für ca. zehn Jahre bei der Daimler AG im Bereich Forschung und Vorentwicklung auf dem Gebiet der Fahrerassistenzsysteme. Anestis Terzis studierte in Ulm Nachrichtentechnik und erhielt den akademischen Grad Dipl.-Ing. (FH). Seine Promotion zum Dr.-Ing. erfolgte am Lehrstuhl für Technische Elektronik an der Universität Erlangen-Nürnberg. Aus seiner bisherigen Arbeit gingen zahlreichen Veröffentlichungen sowie Fachbücher hervor. Er ist aktives Mitglied in internationalen Normungsgremien im Rahmen der ISO sowie des IEEE und in Programmkomitees von Konferenzen.



## **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. Ihle, 0170 837 4728  
marc.ihle@ieee.org

### **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)