MPC

MULTI PROJECT CHIP GRUPPE
BADEN-WÜRTTEMBERG

Herausgeber: Hochschule Ulm
Ausgabe: 58/59
ISSN: 1868-9221
Workshop: Reutlingen Juli 2017
Workshop: Offenburg Februar 2018

Band 58 Workshop: Reutlingen Juli 2017

Band 59 Workshop: Offenburg Februar 2018

Cooperating Organisation
Solid-State Circuit Society Chapter
IEEE German Section

Inhaltsverzeichnis

Inhaltsverzeichnis

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, Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

Mitherausgeber (Peer Reviewer):
Heinz-Peter Bürkle, Hochschule Aalen, Anton-Huber-Straße 25, D-73430 Aalen
Joachim Gerlach, Hochschule Albstadt-Sigmaringen, Jakobstraße 6, D-72458 Albstadt-Ebingen
Jürgen Giehl, Hochschule Mannheim, Paul-Wittsack-Straße 10, D-68163 Mannheim
Elke Mackensen, Hochschule Offenburg, Badstraße 24, D-77652 Offenburg
Anestis Terzis, Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

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

Wege aus der Energiekrise der Digitalisierung

Bernd Höfflinger

Zusammenfassung—**Die Nanometer-Roadmap hat eine gigantische Datenexplosion ermöglicht, die besonders bei Video- und Mobil-Daten im Internet dazu geführt hat, dass das Internet bald ein Drittel der globalen Stromversorgung benötigen wird. Dies wird zu einer Wachstumsbremse führen, und das 2016 offiziell verkündete Ende der Roadmap bei etwa 10 Nanometer hat bereits die Strategie ausgebremst, dass digitale Daten nichts kosten.**

Die Chancen und der Bedarf sind nun sehr groß mehr Informationen effizienter und mit weniger Bits zu gewinnen, zu verarbeiten und zu kommunizieren. In einer ganzheitlichen Betrachtung wird die physikalische Welt natur-effizient z.B. logarithmisch erfasst, inspiriert vom natürlichen Seh-System, mit einem neuronalen Netzwerk intelligent verarbeitet und, auf das Wesentliche komprimiert, dann kommuniziert. Der Energiebedarf für die Übertragung eines digitalen Bildes kann mit dieser Strategie in 10 Jahren auf ein Tausendstel reduziert werden.

Auf dem Weg dahin gibt es viele originelle Design-Projekte wie z.B. quasi-logarithmische A/D-Umsetzer, intelligente, hoch-dynamische Pixel, robuste Niedervolt-Digitaltechnik wie die CMOS Differentielle Transmission-Gate-Logik, das zweilagige Perceptron als Standard-Element neuronaler Netze, Leading-One-First Quasi-logarithmische Multiplizierer und darauf aufgebaute Fourier-Transformatoren, die z.B. wahrnehmungsrelevante Kompression und 3D-Bilder bedienen. Die regionale Kompetenz in Energie-Spar-Schaltungstechnik ist der Grundstein für energie-autonome Chip-Systeme, die den zentralen Fortschritt zu intelligenten Mensch-Maschine-Systemen ausmachen, vorbei an der z.Zt. absehbaren Energiekrise der Digitalisierung.

Schlüsselwörter—**Internet Blackout, Energieeffizienz, Video, Information statt Daten, Neuro-Computing, 3D Integration, HDR Vision.**

I. Die Datenexplosion

Nach der Erfindung der planaren, integrierten Siliziumschaltung 1959 durch Hoerni und Noyce gelang es so auffallend, die Zahl der Transistoren pro Chip alle 18 Monate zu verdoppeln, dass deren Freund und Partner Gordon Moore dies schon 1965 als eine Fortschritts-Regel veröffentlichte, die bald Gesetzes-Charakter bekam, weil sie über viele Jahrzehnte realisiert werden konnte und wie keine andere Technologie das Wachstum der Weltwirtschaft bestimmen konnte. Der Weltverband SEMATECH organisierte eine Gruppe aus Hunderten von Experten, die die International Technology Roadmap for Semiconductors (ITRS) entwickelte und fortschrieb [1]. Die treibende Kraft für die Strategie der ITRS war, dass die Transistorlängen und Leitungsbreiten alle 18 Monate auf das 0,7-fache

Bernd Höfflinger, bhoefflinger@t-online.de
Leonberger Str. 5, 71063 Sindelfingen.

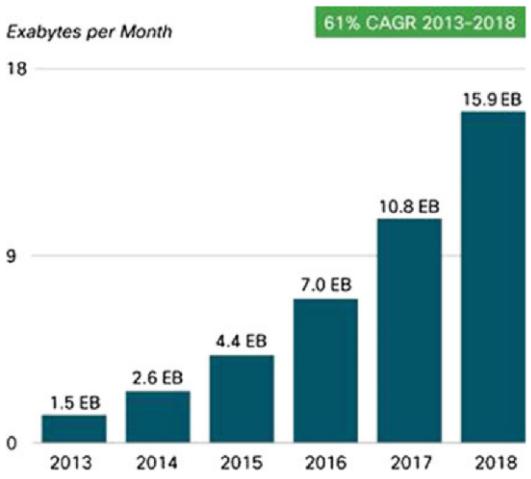


Abbildung 1. Die Datenrate im mobilen Internet nach der CISCO-Studie 2015 [1].

Maß reduziert werden konnten von einer Industrie, die mit einer Forschungs- und Investitionsquote von 30% vom Umsatz ohne Parallele ist. Die Informations- und Kommunikationstechnik (IUK) profitierte am meisten, und in ihr wächst das gewichtigste Gebiet, die mobile Kommunikation im Internet, in ihrer Datenrate um 60% pro Jahr ([2] und Abbildung 1).

In der Fortschreibung bis 2021 wird weiter mindestens dieses Wachstum angenommen, auch weil das IOT (Internet of Things) und Autonome Mobilität hinzukommt. Dieser Annahme eines so hohen exponentiellen Wachstums steht nun die Dramatik gegenüber, dass der simple exponentielle Fortschritt auf der Technologie-Seite vor kurzem verabschiedet wurde.

II. Das Ende der International Technology Roadmap of Semiconductors (ITRS)

Dass die ständige Strukturverkleinerung entlang der Nanometer-Roadmap zu einem praktischen und fundamentalen Halt kommen würde, war lange voraussehbar. Trotzdem ist es ein historisches Ereignis, dass die bedeutende ITRS-Arbeitsgruppe mit dem Bericht 2015 aufgelöst wurde [3]. Eine fundamentale Aussage zum Abschluss ist, dass die minimale, praktische Strukturbreite auf den Chips bei 10nm liegen wird.

Inzwischen werden sog. 7nm-Technologien vorgestellt und 5nm angekündigt. Diese Maße für minimale Kanallängen von Transistoren geben aber nicht die

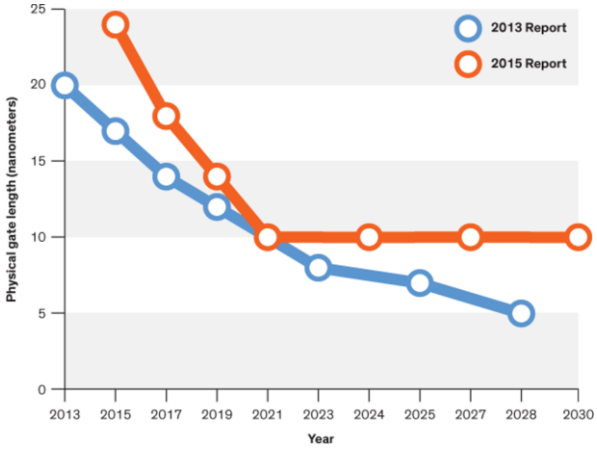


Abbildung 2. Die ITRS Roadmap, Report 2015 [3].

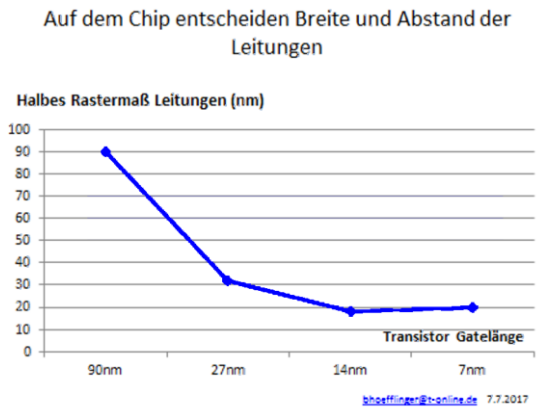


Abbildung 3. Halbes Rastermaß Poly-Silizium und erste Metallisierung.

erreichbaren planaren Bauelemente-Dichten auf den Chips wieder, wie Abbildung 2 zeigt.

Dieser harte Anschlag in der zweidimensionalen Dichte auf den Chips bedeutet, dass jede Erhöhung in Daten-Kapazität und Daten-Rate proportional mit steigenden Kosten und steigender Energie verbunden wäre.

Dieser historische Roadblock hat zu beachtlichen „Last-Minute“ - Neu-Formationen von Innovations-Programmen geführt:

- Ein gewisser Nachfolger der ITRS wurde die „International Roadmap for Devices and Systems“ (IRDS) [4].
- Die weltgrößte Ingenieurvereinigung IEEE hat die interdisziplinäre Initiative „Rebooting Computing“ formiert [5].
- Die amerikanische National Science Foundation (NSF) hat Programme für „Energy-efficient computing“ [6]

Es kommt jetzt also neben Nanometern, die nur ein Mittel zum Zweck sind, endlich auf das eigentliche Ziel an, nämlich die Energie pro Funktion zu reduzieren und damit natürlich auch Kosten und Performance zu steigern, nicht nur um Prozente, sondern um Größenordnungen, eine Strategie, die wir schon 2012 nannten:

- **Von Nanometer zu Femtojoule [7]–[9].**

III. Die Energiekrise des Internets

Der Energiebedarf des Internet wird bisher von keiner Kommission oder Arbeitsgruppe verfolgt und fortgeschrieben. Noch immer ist die Berkeley-Studie von 2011 [10] eine maßgebliche Referenz. Publiziert wurden Schätzungen von 150 GW 2013 [8] und Anteile von 15% am Weltstrombedarf, der 2010 etwa 2TW betrug. Neben Berkeley hatten wir auf Grund von Konferenz-Informationen für 2010 für Server 36GW geschätzt [7], gleich mit Berkeley, und eine Gesamtleistung aus der Steckdose 2010 von 80GW für das

Tabelle I
Elektrischer Leistungsbedarf des Internet
[6], Chapter 12.1.

Year	**2010**	**2015**	**2020**
Mobile traffic (TB/s)	0.11	1.50	13.20
Server performance (rel.)	1	20	200
Energy efficiency (rel.)	1	7	25
Electrical power (GW)	40.0	120	240
PC‘s power (GW)	30.0	30	30
Mobile devices (GW)	0.4	28	56
Infrastructure (GW)	10	50	200
Total operative el. power (GW)	**80.4**	**228**	**526**
Embodied el. power (GW)	80.0	228	526
Total power (GW)	**160.4**	**456**	**1052**
Global el. power generation (GW)	**2300**	**2800**	**3300**

Internet, der Leistung von 80 Kernkraftwerken. Eine Verdreifachung bis 2015 auf etwa 250GW hat sicher stattgefunden, weil allein der Mobilverkehr sich mehr als verzehnfacht hatte. Nach CISCO wird dieser zwischen 2015 und 2020 sich weiter verneunfachen. Am Ende der Nanometer-Roadmap ist es dann relativ optimistisch, dass der Energiebedarf in diesen 5 Jahren nur um das 2,5-fache zunimmt auf 550GW. Die Berkeley-Annahme, dass die eingebaute Energie (embodied energy = emrgy), d.i. Rohstoffe, Herstellung, Anlagenbau, Kühlung, Reparatur, Entsorgung, etwa noch einmal so viel Energie benötigt, bleibt sicher in etwa gültig, sodass insgesamt elektrische Leistung von 1,1TW 2020 benötigt würde, ein Drittel der Weltproduktion (Tabelle I). Dieses Internet-Wachstum würde 80% des jährlichen Zuwachses der Welt-Strom-Produktion benötigen.

Dieses Szenario ist unrealistisch und trotzdem optimistisch, da die Energieeffizienz je Bit sich alle 2,5 Jahre verdoppeln müsste. Wenn das Internet nur für die Steckdose mehr Leistung benötigt als alle Kernkraftwerke der Welt (etwa 400) produzieren, haben wir eine

Tabelle II
ENERGIEBEDARF PRO BIT IN ELEKTRONEN-VOLT:
1EV = 1,6 X 10^{-19} JOULE (WS).

Distanz	Auftrag	Elektronenvolt
100 nm	1Bit Speicher	1.000
1 cm	Chip-Länge	100.000
	Gehirn-Synapse	60.000
10 cm	Leiterplatte	6 Mio.
1 m	Computer-Regal	50 Mio.
1 km	Zelle	1 Mrd.
1000 km	Zelle-Server	30 Mrd.

ernste Krise vor uns, und wir schauen genauer auf die Beiträge dazu.

IV. DAS TEURE INTERNET-BIT UND DIE VIDEO-INFLATION

Wie die CISCO-Studien belegen, benötigt Video 70% der mobilen Internet-Bits. Zu den Größenordnungen (Energieeffizienz 2015):

- 1min Standard Video: 35 MB 00,26 kWh,
- 1 Standard-Film 120min = 4 GB = 32 kWh,
- Facebook-Nutzer bewegen 4 Mrd. Videos/Tag = 1 GW (1 Kernkraftwerk oder 300 Winräder auf 30 km^2).

5G Mobil und 4K High-Definition-Video lt. Ankündigungen werden den Datenverkehr verzehnfachen! Radikale Innovationen werden nötig sein.

Tabelle II zeigt, dass die Mikroelektronik auf dem Chip, was die Energie für das Lesen von einem Speicher-Bit betrifft, inzwischen mit der Gehirn-Synapse vergleichbar ist. Teuer ist dann schon der Weg zur Zelle, und von dieser zum Server wird das Bit 300.000 mal teurer als auf dem Chip. Dieses teure Internet-Bit bedeutet, dass wir unglaublich sparsam mit ihm umgehen. Die neue Epoche der Informationstechnik verlangt Fortschritt durch Datenreduktion, d.h.,

- **Mehr Information (letztlich Intelligenz) mit weniger Bits!**

Diese Zielsetzung wird mehr Aufmerksamkeit bekommen und in eine neue Ära mit gewichtigen Innovationen führen.

V. REAL DIGITAL

Die Informationen, die uns bewegen und die wir brauchen, betreffen im wesentlichen unsere physikalische Welt. Für deren Erfassung gilt das Webersche Gesetz: wir begreifen und brauchen alle Informationen N mit etwa konstanter relativer Genauigkeit $\mathrm{d}N$, d.h., $\mathrm{d}N/N = \mathrm{const}$. Dies gilt übrigens auch für VR und AR. Wieviele Datenwerte brauchen wir dann, wenn wir eine Distanz von 1m bis 100m mit einer Genauigkeit von 1% messen wollen? Da wir bei 1m eine Auflösung von 1cm brauchen, ergeben sich 100 Werte pro Meter und 10.000 Werte für die 100m. Eine gigantische

Tabelle III
LEADING-ONES-FIRST (LOF) MULTIPLIZIERER
MIT 6B GENAUIGKEIT, [8] CHAPTER 12.3.

Integer Index	j + k	-1	-2	-3	-4	-5
$a_j = 1, b_k = 1$	1	b_{k-1}	b_{k-2}	b_{k-3}	b_{k-4}	b_{k-5}
If $a_{j-1} = 1$[a]		(1)	(b_{k-1})	(b_{k-2})	(b_{k-3})	(b_{k-4})
If $a_{j-2} = 1$[a]			(1)	(b_{k-1})	(b_{k-2})	(b_{k-3})
If $a_{j-3} = 1$[a]				(1)	(b_{k-1})	(b_{k-2})
If $a_{j-4} = 1$[a]					(1)	(b_{k-1})
If $a_{j-5} = 1$[a]						(1)

The complexity is of the order O(6^2/2) independent of the word length n

[a] Otherwise the inputs from this line are 0

Tabelle IV
ZAHL DER TRANSISTOREN FÜR STANDARD-MULTIPLIZIERER
UND GENAUIGKEITSBESTIMMTE LOF- MULTIPLIZIERER,
[8] CHAPTER 12.3.

Word length n	8b	16b	24b
Standard Booth-Wallace	1,600	6,400	14,400
LOF 6b precision (1.6%)	540	1,030	1,620
LOF 10b precision (0.1%)		2,100	2,700

Verschwendung von Daten. Denn für 1% Genauigkeit brauchen wir 100 Werte für die Dekade von 1m bis 10m und weitere 100 Werte für die Dekade von 10m bis 100m, also insgesamt nur 200 Werte, ein Datenkompressionsfaktor von 50x. Trivial, aber der Rechenschieber wurde in vielen Ländern der Welt (inkl. USA) in den Schulen nie eingeführt. Für die digitale Multiplikation von Daten der realen Welt bedeutet dies, dass wir Multiplikation nicht vom least-significant Bit (LSB) her aufrollen, sondern vom Leading-One-First (LOF). Mit z.B. 6 msb Genauigkeit (1,6%) verbleibt dann die in Tabelle III dargestellte Additionsaufgabe. Sie hat die Komplexität O(6^2/2) unabhängig von der Wortlänge n, während der Addierer im LSB-First Multiplizierer im Booth-Wallace Code die Komplexität O(n^2/2) hat. Im Vergleich ergeben sich insgesamt die Transistorzahlen der Tabelle IV für Standard-CMOS. Für 16b Wortlänge ergeben sich für 6b Genauigkeit Einsparungen von 83% bei der Transistorzahl und natürlich Verbesserungen bei Geschwindigkeit und Energie je Multiplikation.

Diese drastische Umdenke beim Multiplizierer führt uns zur großen Abakus-Rechenkultur und auf die Neu-Bewertung von Wertebereichen, Genauigkeiten und ganzzahligen bzw. Gleitkomma- Zahlensystemen für reale Anwendungen. z.B. gerade auch beim Neuro-Computing.

VI. ROBUSTE NIEDERVOLT-SCHALTUNGSTECHNIK

Die digitale statische Standard-CMOS-Schaltungstechnik hat mit ihrer Störsicherheit und unvergleichlichen Sicherheit und Einfachheit in der Schaltungsautomatisierung weltweit seit 1980 zu unermesslichen Schaltungsbibliotheken geführt, einem „Know-How“, gegen das Innovationen es

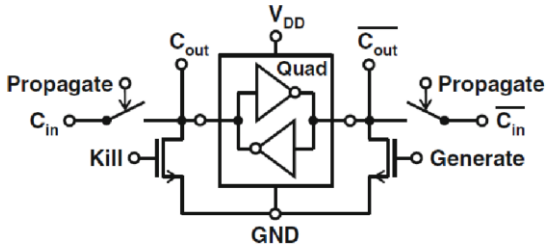


Abbildung 4. CMOS Low-Voltage Differntial Transmission-Gate (LVDTG) Logik.
Manchester-Carry Zelle eines Addierers [7, 11]. Die Schalter sind NMOS-Transistoren mit Null Volt Schwellenspannung. Die übrigen Transistoren werden Sub-Threshold betrieben. Das kreuzgekoppelte Inverter-Paar arbeitet als Differenzverstärker und Ausgangsbeschleuniger mit vollem Spannungshub.

schwer haben. Hinzu kam, dass diese Schaltungen so robust waren, dass man die Versorgungsspannung gegenüber dem Nominalwert senken konnte und noch immer Funktion bekam, wenn auch mit geringerer Geschwindigkeit. Auf dem Weg von 5V zu 1,5V hat dies funktioniert. Unter 1,5V Betriebsspannung wurde der Geschwindigkeitsverlust drastisch und die Störsicherheit sowie Ausbeute schlechter. Die Minimal-Maße in den Zellen konnten zur Erhaltung der Sicherheit kaum mehr verkleinert werden, was mit zu dem Ende der Nanometer-Roadmap geführt hat.

Für Energie-Effizienz, Robustheit und Geschwindigkeit muss eine neue Schaltungstechnik kommen.

Die bis heute führende Kandidatin ist die CMOS LVDTG Logik [11]–[13]:

- Nachgewiesene Robustheit bis 210mV
- Beste Energieeffizienz (197fJ)
- Höchste Geschwindigkeit (240 MOP/s)
- Bestehende Schaltungsbibliothek [14].

Das zitierte Chip [13] ist ein JPEG Coder in einem 40nm-Prozess, entwickelt und statistisch getestet an der KU Leuven, Belgien, und dort wurde auch die LVDTG-Bibliothek [14] entwickelt. Im ESPRIT Grundlagenprojekt HIPERLOGIC war die LVDTG_Logik 1998 in Stuttgart entwickelt worden. Die Ergebnisse für Multiplizierer in einem 0,8 μm Prozess mit selbstjustierenden 100nm DMOS-Transistoren [11] gewannen 2000 den Best Paper Award der IEE-Konferenz für Efficient Computing. In Korea wurden dann 2008 mit einem 180nm-Prozess in LVDTG-Technik mit Bootstrapping beachtliche Ergebnisse erzielt. Nicht nur schneidet Standard-CMOS in der Effizienz deutlich schlechter ab, auch die Geschwindigkeit ist 10-mal schlechter als die der KU Leuven. Details finden sich in [7, 8] inklusive der Projektion, dass 2020 in einem 20 nm-Prozess mit LVDTG-Logic, LOF-Multiplikation und 3D-Integration eine 16x16b Multiplikation mit 1 fJ Energie erreicht werden kann.

Dies vermittelt beachtliche Aussichten besonders für die multipikations-intensive Video-Signalverarbeitung.

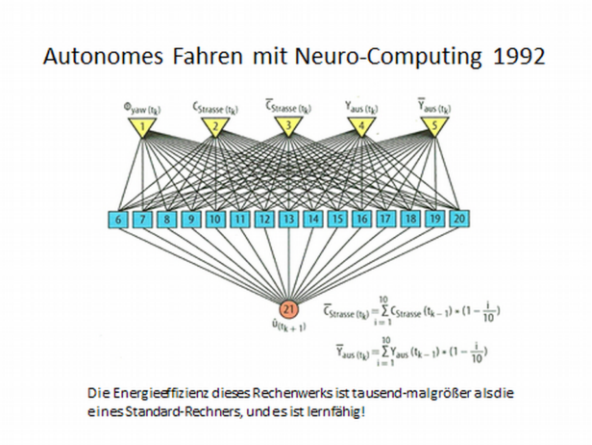


Abbildung 5. Lernfähiger Lenkassistent 1992: Eingangssignale sind die Straßenkrümmung, der Gierwinkel und die Verschiebung gegen die Fahrbahnmitte. Die Gewichte der Synapsen wurden aus den Daten eines Profi-LKW-Fahrers auf einer 1,8 km langen Autobahnstrecke erlernt [15].

Diese wird aber auch fundamental durch das Neuro-Computing erneuert.

VII. Neurocomputing

Erfassung und Nutzung realer Informationen mit Sensorik und lernfähigen Rechnern sind seit den fünfziger Jahren ein attraktives Forschungsgebiet. Eine signifikante zweite Welle startete in den 80-er Jahren mit beachtlichen Ergebnissen Anfang der neunziger Jahre [15] (Abbildung 5). Sie konnte sich aber gegen die raschen Fortschritte konventioneller von-Neumann-Maschinen und deren Standard-Status nicht durchsetzen. Die neue Welle begann 2010 mit der höchst effizienten Gesichtserkennung auf Smartphones, vorangetrieben in Taiwan und Südkorea. Ihre Energieeffizienz ist 1.000-mal höher als konventionelles Rechnen, das wegen des hohen Datenbedarfs und Datenverkehrs zwischen Speichern und Rechenwerken besonders vom Ende der Nanometer-Roadmap betroffen ist. Nach der gewichtigen Behandlung des Brain-Inspired Neurocomputing 2012 [7] und 2016 [8] zeigen die neuesten industriellen Ankündigungen und die institutionellen Neuorientierungen [4]–[6], dass eine Zeitenwende eingetreten ist. Der Einsatz lernfähiger Neurocomputer in industriellen, medizinischen und Verkehrs-Anwendungen fordert fundamental neue Entwicklungs- und Zertifizierungs-Regeln sowie -Standards, wie seinerzeit die Einführung integrierter Schaltungen in der industriellen Regelungstechnik in den 80-er Jahren.

VIII. Hochdynamische Bildsensorik

Natürlich lebt alle Informationsverarbeitung von der Qualität der Informationsgewinnung. Hier ist Bildinformation die komplexeste und wegen ihrer vorherrschenden Rolle in unserer realen Welt die wichtigste. Deshalb beherrscht auch Video das Internet.

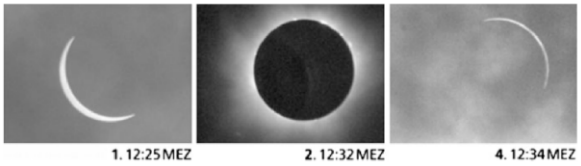


Abbildung 6. Drei Bilder aus dem HDRC-Video mit konstanter Blende von der Sonnenfinsternis in Deutschland am 11. August 1998.

Das menschliche Sehsystem (HVS = Human Visual System) ist das beeindruckendste System für Informationsgewinnung und -verarbeitung, beginnend beim Sensor mit logarithmischer Kennlinie, d.h., konstanter Kontrastauflösung von etwa 1% (kleinster wahrnehmbarer Unterschiede von Helligkeit), über einen Dynamik-Bereich von 1 Mio. zu 1. Der Mensch konnte mit Film diese logarithmische Kennlinie über etwa 4 Dekaden nachbilden, nutzen und kultivieren zu hochqualitativer Foto- und Filmtechnik.

Leider hat das Charge-Coupled Device (CCD) eine fundamental lineare Kennlinie für Spannung als Funktion von Ladung, und mit der Erfindung des CCD-Bildsensors 1970 (Photonen Ladung) verewigte sich diese lineare Kennlinie mit einer Dynamik von nur 1000:1. Wegen dieser beschränkten Dynamik müssen in der Praxis Mehrfachbelichtungen bei verschiedenen Blenden oder mehrere Pixel parallel für verschiedene Helligkeitsbereiche betrieben und die jeweils verwertbaren Ergebnisse zu den gewünschten Bildern zusammengerechnet werden. Aus dieser Art von HDR-Bildtechnik ist ein gigantisches Fach- und Geschäftsgebiet geworden mit vielen Giga-Operationen pro Bild und entsprechendem Zeit- und Energiebedarf.

Als die Automobilindustrie 1986 eine Kamera mit einer Dynamik von 1 Mio.:1 und 1.000 Bildern pro Sekunde als Entwicklungsziel vorgab, musste eine neue Lösung in Silizium gefunden werden. Die Erfindung 1992, einen MOS-Phototransistor unterhalb seiner Schwellenspannung wie in Abbildung 7 zu nutzen, wo die Drain-Source-Spannung der perfekte Logarithmus des Foto-Stroms über viele Dekaden ist, führte zum HDRC-Sensor, der bis heute unübertroffen ist [16]. Schon 1998 konnten mit einem HDRC-Bildsensor bei konstanter Blende mit 10bit Output/Pixel ein Video der Sonnenfinsternis mit absoluter Helligkeit aufgenommen werden. Der Stand 2015 ist ein Mega-Pixel-Sensor mit einem Dynamikbereich von 1mlux bis 10klux, 10Mio. zu 1. Der 10-bit Ausgang eines HDRC-Pixels bietet eine Kontrastauflösung besser als 1% über >3 Dekaden Luminanz, ähnlich dem menschlichen Auge, womit der HDRC-Sensor nicht nur bereits bei der Acquisition eine etwa 4-fache Kompression der Pixeldaten gegenüber den üblichen CMOS-Sensoren, sondern auch in der Bildverarbeitung, -übertragung und -wiedergabe weitere Größenordnungen an Datenkompression dank der Verwandtschaft zum menschlichen

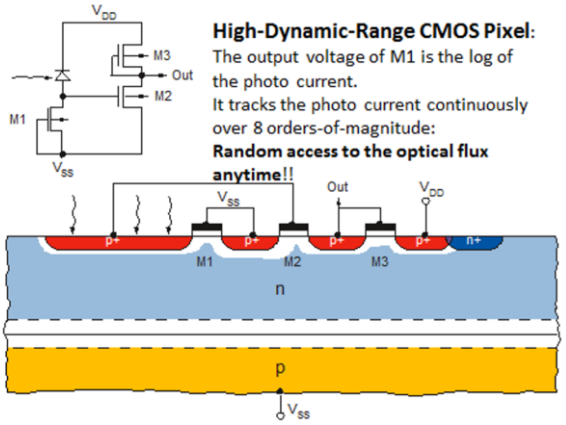


Abbildung 7. Das Pixelelement des hochdynamischen CMOS-Bildsensors. PMOS-Ausführung in der n-Wanne. Schaltbild und Querschnitt.

Sehsystem (HVS) im Vergleich zum unnatürlichen linearen Bildsensor aufweist.

Natürlich ist die HVS-nahe Bildsensorik besonders kompatibel mit Neurocomputing, sodass eine ganzheitliche Sicht der Bildgewinnung und -verarbeitung hier große Potentiale für Fortschritt bietet.

IX. Fazit: Energie Pro Bild

Selbst wenn wir den Schritt von Nanometer pro Transistor zu Femtojoule / Operation als neue Strategie vollzogen haben, um eine Energiekrise zu vermeiden, sehen wir nicht umfassend genug voraus. Das hohe erreichte Integrationsniveau ermöglicht und fordert, dass wir ganzheitlich vom Input bis zum Ergebnis entwickeln. Da Video der bit- und energiehungrigste Teil der Digitalisierung und des Internet ist, muss unser Fokus sein, die Energie pro Bild konsequent und um Größenordnungen zu verbessern. Hier ist eine Liste wesentlicher Innovationsgebiete:

- HDRC log. Bildsensorik
- Neuro-Computing
- Niedervolt-DTG Schaltungstechnik
- Digitaler Rechenschieber
- 3D Integration (Ch.3 in [8])
- Energie-autonome Chips (Ch.19 in [8]).

Mit der synergetischen Entwicklung dieser Technologien kann die Energie pro Bild in 10 Jahren um drei Größenordnungen gegenüber dem heutigen Stand verbessert werden. Damit wird nicht nur eine Energiekrise mit Stagnation vermieden, sondern nachhaltiges Wachstum der Informations- und Kommunikationstechnik ermöglicht.

LITERATURVERZEICHNIS

[1] www.itrs2.net
[2] Cisco Visual Networking Index. www.cisco.com
[3] Rachel Courtland: *Transistors could stop shrinking in 2021*, IEEE Spectrum, Sept. 2016, pp. 9-11.
[4] irds.ieee.org
[5] http://www.rebootingcomputing.ieee.org/
[6] Energy-efficient computing, 2015 Report to the National Science Foundation (NSF).pdf.
[7] Hoefflinger, B. (ed.): *CHIPS 2020, A Guide to the Future of Nanoelectronics*, Springer 2012.
[8] Hoefflinger, B. (ed.): *CHIPS 2020 Vol.2, New Vistas in Nanoelectronics*, Springer 2026.
[9] Hoefflinger, B.: *Nanoelektronik: Auswege aus ihrer Energiekrise*, ELEKTRONIK 2/2016, pp. 28-33, 3/2016, pp.37-39.
[10] Raghavan, B.: *The energy and emergy of Internet*, ICSI and UC Berkeley 2011.
[11] Grube, R. et al.: *0.5V CMOS Logic delivering 25 million 16x16c multiplications at 400fJ*, IEEE Computer Elements Workshop, Mesa, AZ, 2000. Also: [7], chapter 3.6.
[12] Chapter 12.3 in [8].
[13] Reynders, N.: *A 210mV 5MHz variation-resilient, near-threshold JPEG encoder in 40nm CMOS*, 2014 IEEE Intl. Solid-State Circuits Conf., Digest Tech. Papers 27.3, Feb. 2014, and private communication.
[14] Reynders, N.: *Variation-resilient building blocks for ultra-low-energy sub-threshold design*, IEEE Trans. Circuits & Systems II vol.59(2), pp.898-902, 2012.
[15] Neusser, S. et al.: *Neural Control for lateral vehicle guidance*, IEEE Micro 13(1), p.57f, 1993,
[16] Chapter 13.4 in [8].

Bernd Höfflinger studierte Physik in Göttingen und München und promovierte 1967. Nach einer Professur an der Cornell University wurde er1970 MOS Produktleiter bei Siemens. An der Uni Dortmund gründete er die Elektrotechnik und baute eine Pilotlinie. Nach der Leitung zweier Fakultäten in USA baute und leitete er 20 Jahre lang das Institut für Mikroelektronik Stuttgart (IMS CHIPS).

Hochschule Kempten
University of Applied Sciences

Implementation and Testing of a FPGA Based Sigma Delta Analog to Digital Converter

Martin Knauer, Jörg Vollrath

***Abstract*—In this work a 10-bit sigma-delta analog to digital converter was implemented using only a NEXYS 3 FPGA board in combination with some passive components. In addition, a simple R2R digital to analog converter was implemented, which allowed testing the ADC with synchronous digital signals produced by the FPGA. INL and DNL was measured for both devices. The overall setup revealed a SNR of 33.8 dB with an ENOB of 5.3 bit. But with some minor adaptations (e.g. compensation of an offset error) a resolution of up to 8 bit can be achieved. It was shown that with such a simple approach a fully functioning sigma-delta analog to digital converter can be built.**

***Index Terms*—Sigma-Delta Converter, R2R DAC, FPGA.**

I. Introduction

Sigma-delta (Σ-Δ) analog to digital converters are very popular in industry. They provide high resolution for low-price production costs [1]. Furthermore, the basic structure is relatively simple and can easily be implemented in nano-CMOS processes [2]. This makes sigma-delta ADCs ideal for applications such as industrial sensing, audio, and narrow bandwidth radio frequency devices. In this work, a real sigma-delta ADC was realized with a NEXYS 3 FPGA board (Digilent) and some passive components. Furthermore, the FPGA board in combination with a self-made 10 bit R2R DAC was used to generate clock-synchronous analog signals to test and observe the limits of the DAC. The goal was to test and verify the implemented ADC and show possibilities and limits of such an approach by only using relatively simple hardware.

II. The Sigma-delta Topologie

The main principle of an ordinary 1st-order Σ-Δ converter is based on oversampling, where the internal sampling rate is much higher than the maximum frequency of the analog input signal. The oversampling rate (OSR) is thereby defined as the quotient of the internal clock rate and the input bandwidth. The input voltage is summed with the output of a feedback loop first and is then connected to an integrator. A comparator outputs a logic 1 if the integrator output is greater than, or equal to zero volts and otherwise it outputs a logic 0. By using a 1-bit DAC, the digital output is then fed back to the summing node and subtracted from the input signal. This is repeated according to the OSR, which increases the accuracy of the converted signal [2]. The resulting bit-stream is then representing the input voltage and is subsequently digitally filtered to produce a slower stream of multi-bit samples [3].

Martin Knauer, martin.knauer@hs-kempten.de,
Jörg Vollrath, joerg.vollrath@hs-kempten.de

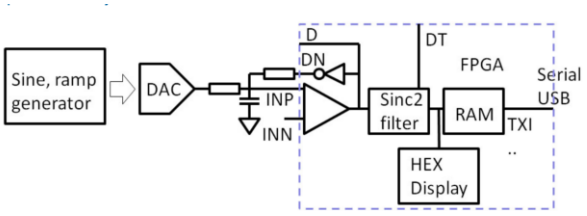


Figure 1. Structure of the sigma-delta analog to digital converter.

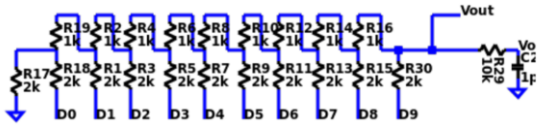


Figure 2. Schematic of the R2R digital to analog converter.

In Figure 1 the structure of the ADC can be seen. The two resistors must match so the gain of the modulator is unity. The capacitor acts as integrator [4].

III. Implementation

To get an analog signal (without use of an expensive signal generator) which is both precise and in sync with the sampling of the ADC, a digital signal is produced by the FPGA and converted by a DAC.

A. 10 bit R2R DAC

For the DAC, a simple R2R converter was built. It is basically a voltage divider consisting of resistors of 1 kΩ and 2 kΩ. The schematic and the finished structure can be seen in Figure 2 and 3. Here it is important to note that the resistors are precise, especially for the higher order bit, to get a sufficient resolution.

B. Sigma-Delta ADC

The schematic of the passive components of the Sigma Delta Converter can be seen in Figure 4. Here VAWG2 is the reference voltage for the comparator

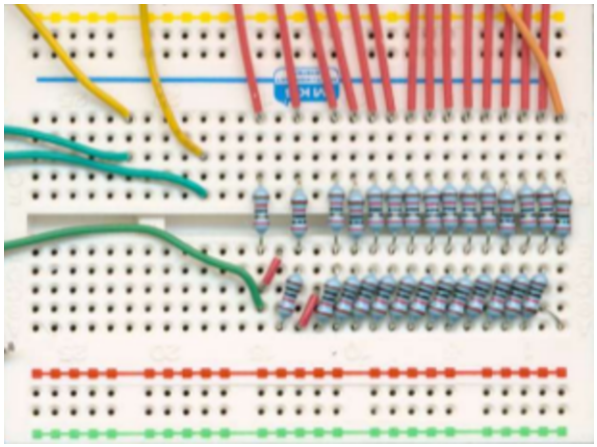


Figure 3. Structure of the R2R digital to analog converter.

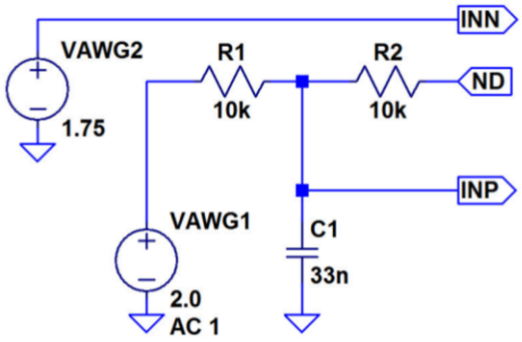


Figure 4. Schematic of the passive components of the Sigma Delta Converter.

on the FPGA, contact INN. VAWG1 is realized by a digital signal generated on the FPGA which is then converted by the R2R DAC.

The Value of C1 and R2 can be calculated as:

$$C1 > 12K_bT\frac{2^{2B}}{V_{FS}^2}$$

$$R2 \ll \frac{0.72}{Bf_SC}$$

An oversampling rate of 2048 and an input bandwidth of 6.25 kHz was chosen. This leads to an effective number of bit (ENOB) of around

$$ENOB = \frac{30\log(OSR) - 5.17 + 1.76}{6.02} \approx 15.94$$

R1/R2 and C1 were then set to a resistance of 100 kΩ and a capacitance of 22 pF. The 1-bit DAC is realized on the FPGA which gives an analog voltage on Pin ND connected to the summation node. Pin INP is the second input for the comparator.

The digital bit-stream is then fed to a SINC2 filter with 26 bit. It is important that the word length is long enough and that the filter is of second order (for a 1st order modulator). Since the serial interface has only 38400 baud-rate and the conversion can be faster, a dual port RAM buffer (Inst_RAM) was implemented with a size of 2WBUFFER = 16 k with 16-bit output values taken from the upper bit of the SINC2 filter output.

A UART is used to stream a serial output of data to the USB port. The serial interface streams a new line for each output data word in hexadecimal.

IV. Analysis

A. Testing the R2R DAC

The R2R DAC was tested with an Electronic Explorer Board (Digilent). A digital signal was generated and connected to the inputs of the converter and the analog output was then measured by the internal oscilloscope of the board. Since the oscilloscope has only a limited resolution of 10 bit, multiple measurements were conducted using a preamplifier. In this way, eleven measurements with a voltage range of 300 mV each were done to cover the whole 3.3 V output range. The testing was done with a relatively slow ramp signal and the resulting measured values were analyzed afterwards. Thereby, the first 10 values of each code were omitted to avoid influences of the settling time. The remaining values of each code were averaged and the DNL was calculated as:

$$DNL(i) = \frac{V_{out}(i) - V_{out}(i-1) - LSB}{LSB}$$

INL is computed by summing all DNL values up to the current code. In Figure 5 the values of DNL and INL can be seen.

In the resulting data, there is a unique pattern which can be seen best in the INL graph: the data points are quite continuous at the y axis but after a constant number of codes the values are jumping. By counting these data clusters, we get 11 sets (first “jump” is at code 93, second one at around 186 and so on). This cannot be explained by inaccuracies of the resistors because then the jumps have to be located at a code position of 2 to the power of N (where N is between 1 and 9) or a combination of these positions if several resistors are imprecise.

A reasonable explanation for this pattern is a static gain error of the preamplifier of the oscilloscope. Having this in mind, the data values should look even

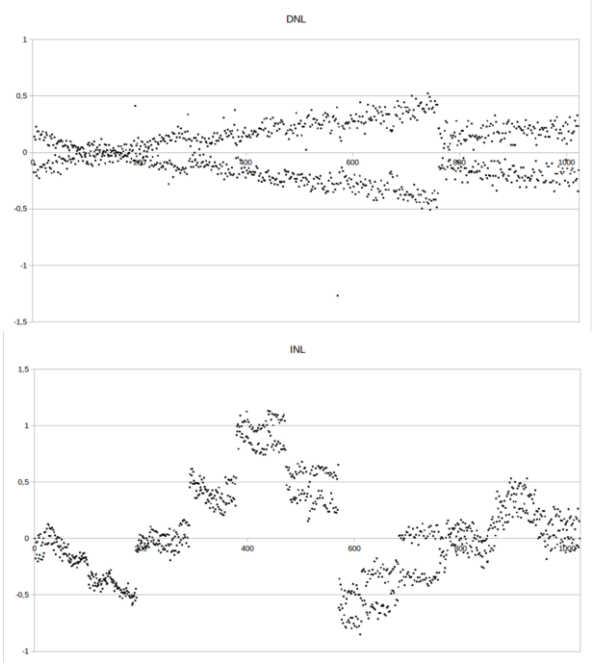


Figure 5. DNL and INL of the 10-bit R2R converter.

better in reality (without the measuring error) and if we assume that the values of INL (which are now above 1) are actually within ± 1, we get a DAC with a resolution of 9 bit.

B. Testing the Sigma-Delta ADC

For the analysis, Python in combination with the modules NumPy/SciPy and MatPlotLib were used. The measurement was done with a digital ramp signal with a frequency of around 2 Hz. This was chosen to have at least one complete ramp fitting in the 16 k data words stored in the internal RAM buffer. The digital signal was then converted by the DAC and afterwards converted back to a digital signal by the ADC. The output values were analyzed with a histogram test.

An array with 1024 entries was computed, representing the 1024 codes of the 10-bit converter. Afterwards, the number of occurrence was determined. With these numbers of occurrences, the values for DNL can be calculated as:

$$DNL_i = \frac{N_i - N_{AVG}}{N_{AVG}}$$

INL is then calculated as:

$$INL_i = INL_{i-1} + DNL_i$$

DNL and INL can be seen in Figure 6.

The values of DNL are all within ± 2, which is really good considering the R2R DAC only has a resolution

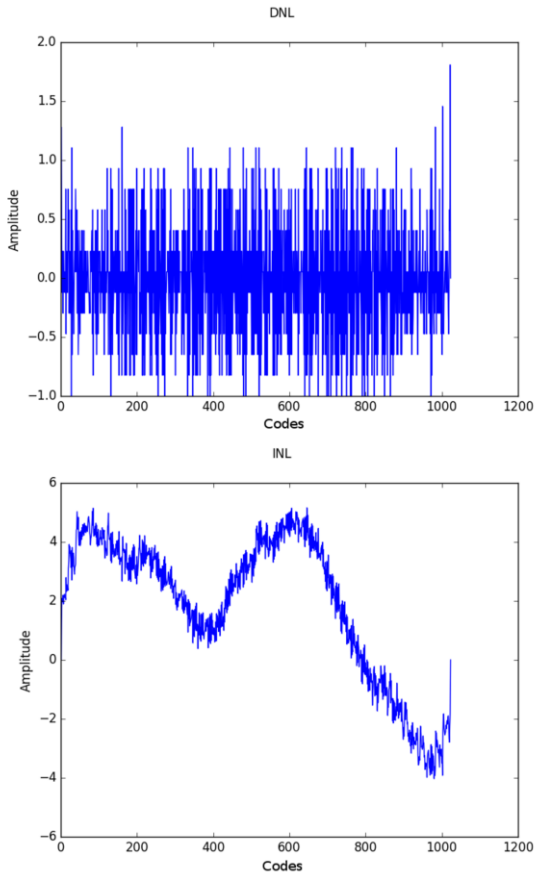


Figure 6. DNL and INL of the sigma delta converter.

of around 9 bit. INL is, however, with values between ± 5 more diverse. This would mean a loss of 3-4 bit. But with some minor adjustments this could surely be improved.

To determine the signal to noise ratio and ENOB, a FFT of a sine signal measurement was performed. To avoid bleeding effects, it is beneficial to use a whole number of sine periods for the FFT. In this way, it is not necessary to apply windowing.

Furthermore, the number of periods should be a prime number. Otherwise, the signal is sampled at exactly the same points for multiple sine periods. In our case 43 sine periods were taken.

In Figure 7, the sine signal is shown in time- and frequency domain. The signal frequency is represented by the first peak of the FFT (around 43 Hz with a magnitude of 53.95 dB). Particularly noticeable are the high magnitudes of the harmonics. They are at around 86 Hz (16.04 dB), 129 Hz (11.47 dB) and so on, which leads to a total noise magnitude of 20.17 dB. By the following two equations the Signal to Noise ratio and ENOB can be calculated:

$$SNR = 20 \log \left(\frac{V_{eff,signal}}{V_{eff,noise}} \right) \text{dB}$$

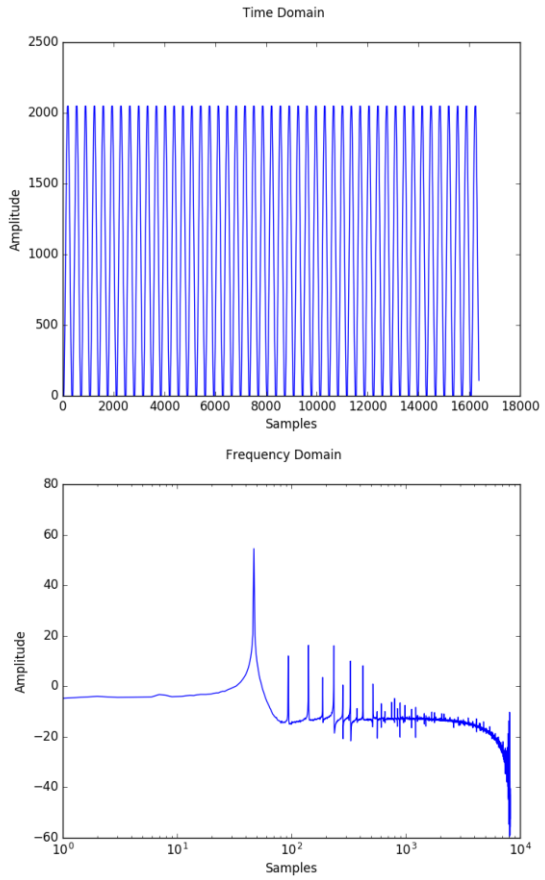


Figure 7. Converted sine signal in time- and frequency domain.

$$ENOB = \frac{SNR - 1.76\text{dB}}{6.02\text{dB}}$$

The SNR can be determined as 33.78 dB which leads to an ENOB of only 5.3 bit. The harmonics are most likely caused by an offset error. If we assume that those harmonics are caused by such an offset, which could be compensated without much effort, the SNR can be increased drastically. For example, if the first ten harmonics are ignored for the noise floor calculation, the SNR is at around 50 dB. This leads to an increased ENOB of 8 bit.

V. Conclusion

The sigma delta ADC built with the NEXYS3 FPGA board was a success. Although the resolution is not perfect yet, it could be improved to 8 bit with some minor adaptations. Even with limited time and equipment, it was possible to build and test a fully functioning sigma delta analog to digital converter.

References

[1] Guessab, S., P. Benabes, and R. Kielbasa, *Passive Delta-Sigma Modulator for Low-Power Applications*, MWSCAS'04., Feb. 2004: pp. 295-298.

[2] A. Roy, M. Meza, J. Yurgelon and R. J. Baker, *An FPGA based passive k-delta-1-sigma modulator*, 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS), Fort Collins, CO, 2015, pp. 1-4.

[3] Baker, R. Jacob, *CMOS Circuit Design, Layout, and Simulation*, Third Edition, Wiley-IEEE Press, 2010.

[4] A. Roy and R. J. Baker, *A passive 2nd-order sigma-delta modulator for low-power analog-to-digital conversion*, 2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS), College Station, TX, 2014, pp. 595-598.

Martin Knauer received the academic degree B. Eng. in electrical engineering and information technology from the University of Applied Sciences Kempten in 2015 and will be finished with his masters in electrical engineering in 2018. Currently he is working at the University part time as a research associate.

Dr. Ing. Joerg E. Vollrath received 1989 his Dipl. Ing. and 1994 his Ph. D. in electrical engineering, semiconductor technology at the University of Darmstadt, Germany. Since then he worked for the memory division of Siemens and Infineon Technologies and Qimonda in various locations in the USA and Germany. He is now a Professor for Electronics at the University of Applied Science, Kempten, Germany. His expertise and interest lies in the field of design of analog and digital circuits, programmable logic, test, characterization, yield, manufacturing and reliability. He has published 22 papers and has currently 23 patents.

FPGA Implementierung eines skalierbaren, seriellen Radizier Algorithmus

David Schall, Norbert Reifschneider

***Zusammenfassung*—Dieser Beitrag beschäftigt sich mit der Hardware-Implementierung eines Algorithmus zur Berechnung der Quadratwurzel aus sehr großen Binärzahlen. Beim vorliegenden PLD-Entwurf wird auf eine skalierbare, serielle VHDL-Modellierung gesetzt. Dadurch ist es möglich, mit verhältnismäßig wenig Hardwareaufwand selbst aus Zahlen mit mehreren tausend Bitstellen die Wurzel zu berechnen, ohne dabei an Taktgrenzen zu stoßen.**

***Schlüsselwörter*—FPGA, Radizieren, Hardware, Algorithmus.**

I. Einleitung

Bei kryptographischen Verfahren wie der RSA-Verschlüsselung werden sehr große Primzahlen verwendet, um einen ausreichenden Schutz zu gewähren. Beim RSA-Verfahren ist das Brechen des Schlüssels äquivalent mit der Faktorisierung eines Produktes aus zwei sehr großen Primzahlen. Um bei steigender Rechenleistung und möglicherweise neu entwickelten Algorithmen zur Faktorisierung auch in Zukunft diese Sicherheit zu gewährleisten, werden immer größere Primzahlen benötigt. Es gibt verschiedene mathematische Verfahren, um eine große Primzahl zu finden, z.B. den Rabin-Miller oder den AKS Algorithmus. Beide Verfahren haben Vor- und Nachteile, auf die an dieser Stelle nicht weiter eingegangen werden soll.

Die aktuelle Forschungsarbeit von Prof. Dr. Norbert Reifschneider beschäftigt sich mit einem neuen, kombinierten Verfahren zur Findung bzw. Verifikation gefundener Primzahlen. Insbesondere sollen die bereits in PC-Assembler entwickelten, software-basierten Verfahren durch wesentlich schnellere Hardware-Implementierungen ersetzt werden.

Der folgende Beitrag befasst sich mit der Umsetzung einer einzelnen mathematischen Operation, die im Rahmen des neuen Verfahrens benötigt wird, nämlich dem Ziehen der Quadratwurzel. Dazu wird in Abschnitt II.A als erstes auf die Herleitung des Algorithmus eingegangen, der als Grundlage für die Hardware-Implementierung genutzt wird, bevor in Abschnitt II.B dieser durch ein Rechenschema abgebildet wird. In Kapitel III wird erläutert, wie die einzelnen Schritte des Rechenschemas durch Hardware umgesetzt werden und welche Besonderheiten sich durch die serielle Arbeitsweise ergeben. Abschließend werden die Ergebnisse beurteilt, die die Implementierung liefert.

David Schall (david-schall@web.de) war Student der Hochschule Heilbronn.

Norbert Reifschneider (norbert.reifschneider@hs-heilbronn.de) ist Professor an der Hochschule Heilbronn, Fakultät für Mechanik und Elektronik.

II. Radizier Algorithmus

Für das Berechnen der Quadratwurzel gibt es verschiedene mathematische Ansätze. Die geläufigsten, wie z.B. das Heron-Verfahren [1], stellen Näherungsverfahren dar. Der Vorteil dabei ist, dass diese sehr schnell ausreichend genaue Ergebnisse liefern können. Allerdings ist durch die den Verfahren zugrunde liegende Rekursion nicht genau determiniert, wann die gewünschte Genauigkeit erreicht wird. Diese Eigenschaft sowie die Notwendigkeit des Festlegens eines geeigneten Startwertes machen diesen Ansatz ungünstig für die Ausführung in Hardware. Außerdem muss beachtet werden, dass es sich in diesem speziellen Fall bei den Operanden um sehr große Zahlen handelt, bei denen jede weitere arithmetische Operation eine deutliche Auswirkung auf die Berechnungsdauer bedeutet.

Ein weiterer Ansatz leitet sich aus den binomischen Formeln ab und wird auch beim schriftlichen Wurzelziehen verwendet. Da bei diesem Ansatz keine der Nachteile eines Näherungsverfahrens auftreten, eignet es sich besser für den vorliegenden Anwendungsfall. Weiterhin konnte der Algorithmus im Rahmen der vorliegenden Arbeit so weit optimiert werden, dass er mit einer einzigen Subtraktion pro Zyklus auskommt.

A. Ansatz des Radizier Algorithmus

Der Ansatz für den Algorithmus liegt darin, jede Dezimalstelle einzeln und unabhängig voneinander zu betrachten. Das bedeutet, eine Zahl mit zwei Ziffern AB kann auch als $10 \cdot a + b$ dargestellt werden. Wenn nun von dieser Zahl das Quadrat berechnet werden soll, ergibt sich dabei die erste binomische Formel [2].

$$(10a + b)^2 = 100a^2 + 20ab + b^2 \tag{1}$$

$$\Rightarrow$$

$$10a + b = \sqrt{100a^2 + 20ab + b^2} \tag{2}$$

Da nicht eine Vorschrift für das Quadrieren, sondern eine für die Wurzelberechnung gesucht ist, wird die Formel umgestellt wie in (2). Zu erkennen ist, dass

sich der Radikand somit aus drei Komponenten zusammensetzt, nämlich zwei quadratischen und einem linearen.

Die Idee des Verfahrens ist nun, die einzelnen Teile nacheinander zu berechnen und dann stellenrichtig vom Radikanden abzuziehen. Begonnen wird dabei mit dem größten Teil $100 \cdot a^2$. Dafür wird zunächst der Radikand durch 100 geteilt und dann innerhalb des entstehenden Quotienten die größtmögliche Quadratzahl ermittelt. Im Beispiel (3.1) ist $5^2 = 25$ die größte quadratische Zahl, die noch kleiner als 31 ist. Somit wurde a mit 5 und der erste Teil von (2) mit 2500 bestimmt. Dieser Teil wird vom Radikanden subtrahiert, damit b berechnet werden kann. Der entstehende Restradikand soll nun den anderen beiden Teilen der binomischen Formel $20ab+b^2$ entsprechen. Dafür wird im folgenden Schritt bestimmt, wie oft $20a$ in den Restwert unter der Bedingung passt, dass $20ab + b^2$ nicht größer als der Restwert wird. Im Beispielfall (3.2) wird 6 für b ermittelt und ist damit die Einerstelle der Wurzel $(50 + 6)^2 = 3136$.

$$\begin{array}{rll} \sqrt{3136} = & 56 & \\ \underline{-25} & -100a^2 = -2500 & (3.1) \\ 636 & \frac{636}{20a} = \frac{636}{100} = 6 \rightarrow b & (3.2) \\ \underline{-636} & 636 \geq 20ab + b^2 = 636 \ \checkmark & (3.3) \\ 0 & & \end{array}$$

Der vorangegangene Ansatz bezog sich auf Zahlen mit lediglich zwei Stellen. Jedoch kann dieser für Zahlen mit mehr Stellen erweitert werden. Siehe dazu Formel (4).

$$(100a + 10b + c)^2 \tag{4}$$

$$= \quad 100^2a^2 + 2000ab + 10^2b^2 + 200ac + 20bc + c^2 \tag{5}$$

$$= \quad (100a)^2 + 2 \cdot (100a) \cdot 10b + (10b)^2 + 2 \cdot (100a + 10b) \cdot c + c^2 \tag{6}$$

Die Umstellung, die in Formel (6) vorgenommen wurde, lässt ein Schema erkennen. In jeder Zeile wird eine neue Stelle der Wurzel ermittelt. Dafür ist es notwendig, den bisher errechneten Wurzelwert w_{akt} und den Stellenwert zu berücksichtigen. Allgemein formuliert lässt sich dies mit (7) darstellen:

$$2w_{akt} \cdot \text{Stellenwert}_n \cdot n + n^2 \tag{7}$$

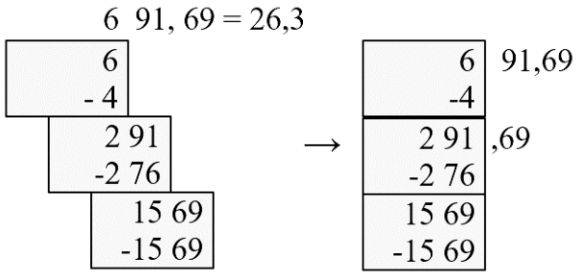


Abbildung 1. Beispieldarstellung des gleitenden Rechenfensters.

n entspricht dabei der nächsten Wurzelziffer. Diese Vorschrift gilt auch für den ersten Schritt, da in diesem Fall w_{akt} gleich Null wäre.

B. Rechenschema

Aus dem Ansatz wurde ein Rechenschema entwickelt, das wie folgt aufgebaut ist:

1 Radikand, vom Komma aus, in Blöcke mit je zwei Dezimalstellen aufteilen.
2 Im Block mit dem höchsten Stellewert beginnen und dort die größte Quadratzahl suchen, die in den Wert passt. Die Wurzel dieser Zahl bildet die höchste Stelle des Ergebnisses.
3 Die Quadratzahl vom Zweierblock subtrahieren.
4 Dem Rest den nächsten Zweierblock des Radikanden anhängen. Im Weiteren wird dieser Wert r genannt.
5 Diesen Wert durch 20 und den bisher errechneten Wurzelwert w_{akt} teilen.
6 Wenn das abgerundete Ergebnis n die Bedingung $20 \cdot w_{akt} \cdot n + n^2 \leq r$ erfüllt, ist n die nächste Ziffer des Wurzelergebnisses. Falls nicht, muss Schritt 4 und 5 mit einem niedrigeren Wert für n versucht werden.
7 Sobald die Bedingung erfüllt ist, wird $20 \cdot w_{akt} \cdot n + n^2$ von r subtrahiert.
8 Die Schritte 4 bis 7 werden so oft wiederholt, bis die Wurzel gezogen wurde.

Wie aus der Herleitung zu erkennen ist, sind während jedes Zyklus immer nur zwei weitere Stellen des Radikanden zu beachten. Diese Erkenntnis wird im Rechenschema in der Form genutzt, dass mit einem gleitenden Rechenfenster gearbeitet wird. Anstatt die Berechnungen mit dem originalen Stellenwert durchzuführen, wird dieses Fenster in jedem Zyklus um zwei Stellen verschoben, wie in Abbildung 1 dargestellt. Damit vereinfacht sich der Stellenwert_n aus (7) zu einem konstanten Multiplikationsfaktor von 10:

$$2w_{akt} \cdot 10 \cdot n + n^2 \tag{8}$$

C. Radizieren im Binärsystem

Da die Computerwelt mit dem Binärsystem arbeitet, muss das in B. beschriebene Rechenschema an dieses System angepasst werden. Der Unterschied zwischen dem dezimalen und dem binären System besteht bekanntlich aus dem Wertebereich von lediglich 0 bis 1 beim binären System und den Stellenwerten jeder Ziffer mit Potenzen von zwei.

Beim Wechsel in das Binärsystem ändert sich der binomische Ansatz (1) zur Formel (9) und die Rechenvorschrift (8) im Rechenschema zu (10).

$$(2a+b)^2 = 4a^2 + 2 \cdot 2a \cdot b + b^2 \tag{9}$$

$$2w_{akt} \cdot 2 \cdot n + n^2 \tag{10}$$

Das restliche Schema verändert sich zwar nicht, allerdings ergeben sich einige Vereinfachungen. Zum einen kann die Multiplikation mit vier durch eine Verschiebung um zwei Bitstellen nach links ersetzt werden und n^2 durch n.

Weiterhin kann das Suchen eines geeigneten n in den Schritten 6 und 7 durch einen einfachen Versuch mit $n = 1$ ersetzt werden. Falls die Prüfung nicht erfolgreich war, muss n Null sein:

$$(w_{akt} \ll 2) \cdot n + n \quad | \quad n \in [0;1] \tag{11}$$

III. Implementierung des Algorithmus

A. Software Algorithmus

Um das in II beschriebe Rechenschema mithilfe eines Rechners auszuführen, muss dieses in einen Software Algorithmus umgesetzt werden. Abbildung 2 stellt den Programmablauf grafisch dar. Darin ist zu erkennen, dass insgesamt drei Register benötigt werden. Eines zum Speichern des Eingangsradikanden, ein zweites, den Akkumulator, mit dem die eigentliche Verarbeitung durchgeführt wird und zuletzt noch das Wurzelregister selbst. Dieses Register wird während der Berechnung als zweites Operanden Register genutzt, das den Vergleichswert enthält. Wie im Vorangegangenen erläutert, wird der Vergleichswert nach Formel (11) mit $n = 1$ berechnet. Dies ergibt die Binärstruktur: $WW \ldots W01$. W ist der aktuelle Wurzelwert, die letzten beiden Bits „01“ sind immer gleich und ergeben sich direkt aus (11). Ist dieser Wert nun kleiner oder gleich dem im Akkumulator, wird er vom Radikanden subtrahiert, ansonsten kann die Subtraktion übersprungen werden, da (11) mit Null für n Null ergibt. Folglich ändern sich in jedem Zyklus nur die Bits des aktuellen Wurzelwerts im Register. Diese werden um eine Stelle nach links verschoben und an der dritten Stelle wird n eingefügt: $WW \ldots W01 \Rightarrow WW \ldots Wn01$.

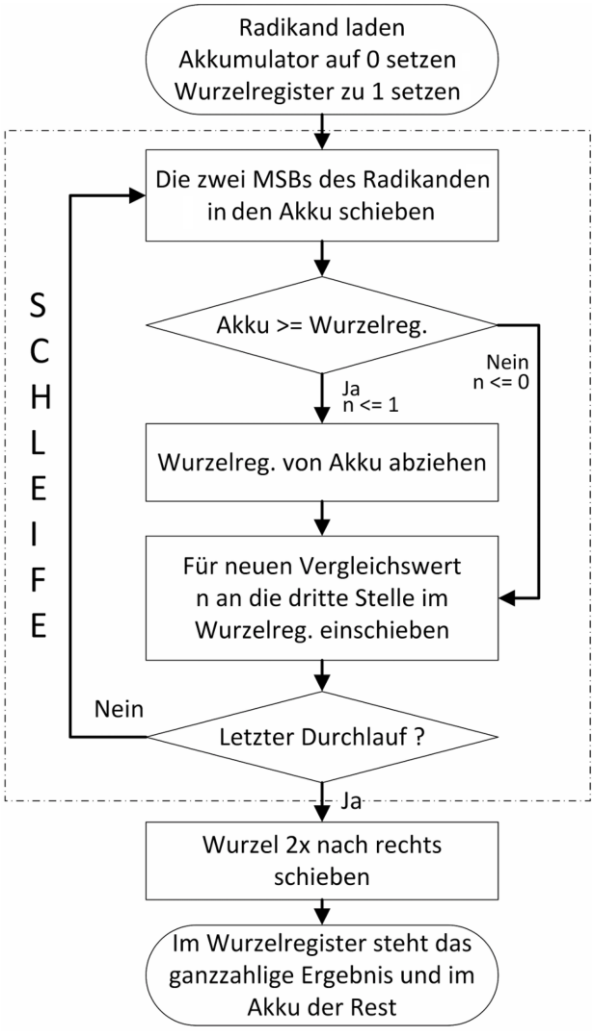


Abbildung 2. Software Algorithmus.

Wenn nun allerdings der Algorithmus auf irgendeiner Plattform umgesetzt werden soll, müssen deren Eigenschaften beachtet werden. Bei einer fertigen CPU sind dies vor allem die verfügbaren Befehle sowie die Bitbreite der Register. Da es sich bei den benötigten Befehlen um einfache Vergleichs-, Subtraktions-, und Schiebebefehle handelt, stellt dies kein Problem dar.

Dafür ist im vorliegenden Anwendungsfall die Bitbreite der Register von entscheidender Bedeutung. Kein General Purpose Prozessor verfügt über Register mit Bitbreiten von mehreren tausend Stellen, weshalb die einzige Möglichkeit darin besteht, jede Operation in mehreren Schritten durchzuführen (Softwarelösung). Eine weitere Möglichkeit besteht darin, eine spezielle Hardware Architektur zu entwickeln, bei der diese Bitbreiten möglich sind (Hardwarelösung).

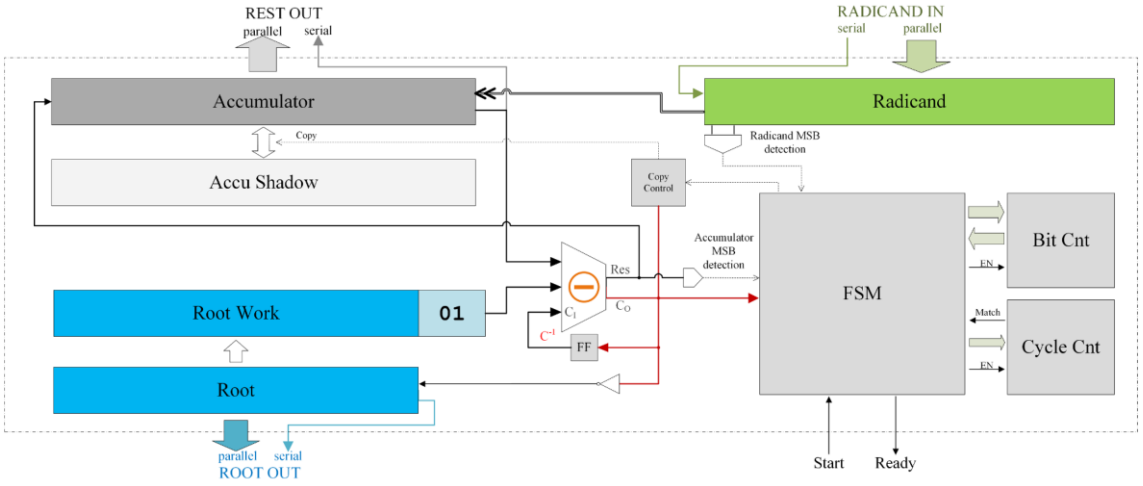


Abbildung 3. Gesamtübersicht der Hardware Implementierung.

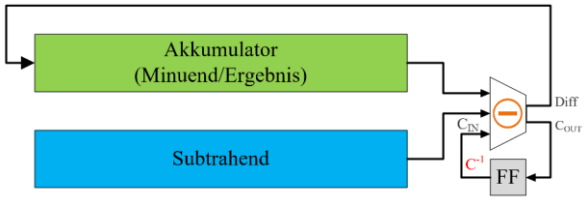


Abbildung 4. Serieller Subtrahierer des Radizier Algorithmus.

B. Hardware Umsetzung

Für die Umsetzung des Algorithmus in Hardware kann die Aufgabe grob in zwei Funktionsblöcke unterteilt werden. Erstens einer Steuereinheit, die über die Steuerung der einzelnen Operationsschritte den eigentlichen Algorithmus abbildet. Zum anderen die Verarbeitungskette, die die notwendigen Operationen wie Subtraktion, Vergleich und Verschiebung ausführt.

1) Subtraktion: Da Bitverschiebungen in Hardware sehr einfach durch serielle Schieberegister realisiert werden können, besteht die Hauptaufgabe der Verarbeitungskette im Vergleichen und Subtrahieren des Wurzelregisters und des Akkumulators. Ein Vergleich wird mit einem Prozessor ebenfalls mit Hilfe einer Subtraktion realisiert, bei der das Ergebnis verworfen und nur die Status Flags im Condition Code Register entsprechend gesetzt werden. Damit bleibt als einzige arithmetische Operation die Subtraktion, die es zu implementieren gilt.

Das Hauptproblem bei der Subtraktion beziehungsweise Addition ist, dass für die Berechnung einer Bitstelle stets auf das Carry-Bit der vorrangegangenen Stelle gewartet werden muss. Diese Zeit lässt sich aus der Gatterlaufzeit der ersten Addierstufe δ_S berechnen sowie der Durchlaufzeit des Carrys durch jede weitere Stufe δ_C. Für die Berechnungsdauer des Bits der Stufe n würde sich damit $n \cdot \delta_C + \delta_S$ ergeben, die gewartet werden muss, bis das Ergebnis sicher verfügbar ist.

Bei einer für mehrere tausend Bitstellen ausgelegten Implementierung ist die lineare Abhängigkeit der Laufzeit von der Zahl der Bitstellen nicht akzeptabel. In heutigen Prozessoren besitzen die Addierer spezielle Schaltungen, die durch zusätzliche Funktionen die Carry Bits parallel vorherberechnen. Beispiele hierfür sind Carry-Select- oder Carry-Look-Ahead Addierer [3]. Das Problem dieser Schaltungen ist, dass der Geschwindigkeitsgewinn mit deutlichem Mehraufwand an Hardware erkauft wird, der bei großer Bitstellenzahl inakzeptabel hoch wird. Zusätzlich tritt eine lineare Steigerung der parasitären Kapazitäten in Abhängigkeit von der Zahl der Addierstufen auf.

Eine mögliche Kaskadierung zur Vermeidung der großen Kapazitäten würde andererseits wieder zu einer Beeinträchtigung des kritischen Pfades beitragen.

Die einzige Möglichkeit, die für diesen speziellen Anwendungsfall noch bleibt, ist die Implementierung einer seriellen Architektur unter Einsatz eines seriellen Subtrahierers. Dieser besteht normalerweise aus drei Registern für Minuend, Subtrahend und Ergebnis, einer Subtraktionszelle und einer Speicherzelle für den Carry. Wie Abbildung 4 darstellt, wurde in diesem Fall das Ergebnisregister mit dem Minuendenregister kombiniert, indem das Ergebnisbit der Subtraktionszelle nicht in ein weiteres Register gelangt, sondern zurückgeführt wird. Nach dem Laden der Werte werden pro Takt Minuend und Subtrahend um ein Bit nach rechts in die Subtraktionszelle geschoben. Das Ergebnis der Ein-Bit-Operation wird zurück in den Akkumulator geschoben. Der Übertrag wird in einem Flip-Flop für die nächste Bitstelle gespeichert.

Durch die Serialisierung ergibt sich für den kritischen Pfad des Carrys lediglich die Verzögerungszeit der Subtraktionszelle. Da diese Zeit unabhängig von der Größe der Operanden ist, können auch sehr große Bitbreiten mit derselben Taktfrequenz berechnet werden.

Ein Nachteil dieser Architektur ist, dass für die Berechnung von N Operanden-Bits auch N Takte nötig sind. Diese Tatsache hat zur Folge, dass die Subtraktion bzw. der Vergleich, der ebenfalls eine Subtraktion impliziert, die mit Abstand zeitaufwändigsten Schritte sind.

2) Vergleich: Im vorliegenden Fall allerdings wird eigentlich zwei Mal dieselbe Subtraktion ausgeführt. Zuerst beim Vergleich des Akkumulators mit dem Wurzelregister, im nächsten Schritt die eigentliche Subtraktion, falls deren Bedingung erfüllt wurde. Aus diesem Grund wurde das Design so verändert, dass nur eine Operation ausgeführt werden muss.

Dazu wird der aktuelle Inhalt des Akkumulators vor Beginn der Subtraktion in das „Accu Shadow"-Register (Abbildung 3) kopiert.

Falls der Wert des Wurzelregisters größer war als der des Akkumulators, was am Status des Carry-Flags nach der Subtraktion erkennbar ist, wird der Inhalt des Shadow-Registers parallel zurückgeladen in den Akkumulator. Andernfalls wird das Ergebnis direkt als neuer Restradikand im Akkumulator behalten. Insgesamt ergibt sich damit eine sehr effiziente Berechnung.

Das Carry-Flag ist in negierter Form auch gleichzeitig die nächste Binärstelle der Wurzel.

Nach einer erfolgreichen Subtraktion (und damit einer weiteren 1 im Wurzelregister) können gleich mehrere führende Stellen im Akkumulator zu 0 geworden sein. Daher ermittelt die Logik während des Subtraktionsvorgangs die Position der nunmehr führenden 1 im Akkumulator. War die Subtraktion erfolgreich, werden im Wurzelregister so viele Nullen angefügt, dass ihre führende 1 in der ersten Zweiergruppe des Akkumulators zu liegen kommt, die nicht Null ist. Auf diese Weise werden weitere Subtraktionen vermieden, die nicht erfolgreich sein können, was eine erneute Steigerung der Effizienz des Verfahrens bedeutet.

3) Steuereinheit: Um den korrekten Ablauf des Algorithmus zu steuern, wurde eine separate Komponente entwickelt, die die einzelnen Hardwaremodule in den jeweiligen Arbeitsschritten korrekt steuert. Da sowohl die Serialisierung als auch die im letzten Abschnitt dargestellten Optimierungen die Komplexität deutlich erhöhen, wurde diese Steuereinheit in Form eines Zustandsautomaten realisiert. Abbildung 5 zeigt die einzelnen Zustände, die während der Berechnung durchlaufen werden.

Im Ausgangszustand *Wait for Start* kann von außen auf das Radikanden-, Wurzel- und Rest-Register zugegriffen werden, um die Werte ein- bzw. auszulesen. Die Implementierung bietet für jeden Wert sowohl eine serielle als auch eine parallele Schnittstelle, um entweder eine Konfiguration für schnelles Laden oder eine für weniger Pins zu ermöglichen. Durch Setzen des Start Signals kann das Radizieren des zuvor geladenen Werts gestartet werden.

Der Zustandsautomat wechselt daraufhin in den Zustand *Find MSB*, der aus Optimierungsgründen eingebaut wurde. In ihm wird der Radikand so lange verschoben, bis die erste signifikante Stelle an der linken Seite des Eingangs-Register ausgerichtet ist. Dadurch werden bei Werten, die nicht die volle Bitbreite besitzen, unnötige Subtraktionszyklen mit den führenden Nullen eingespart.

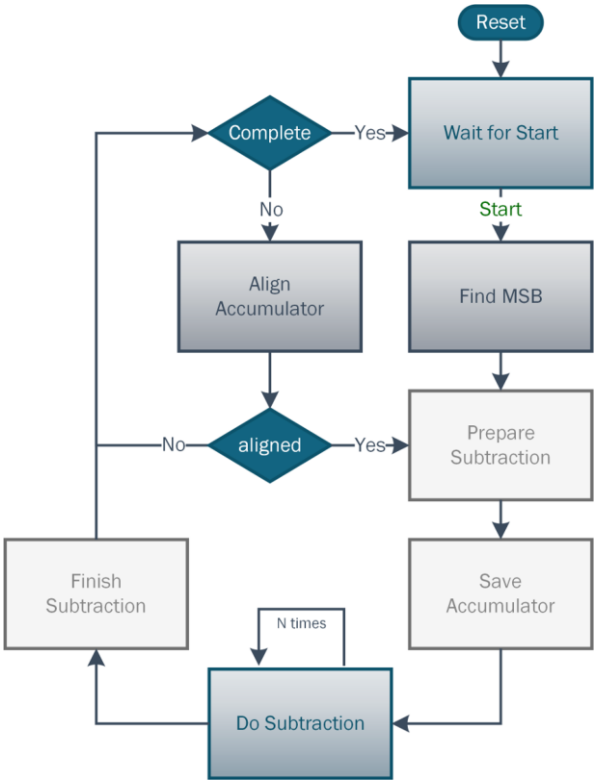


Abbildung 5. Vereinfachtes Zustandsübergangsdiagramm der Steuereinheit des Radizier Algorithmus.

Im nächsten Zustand *Prepare Subtraction* werden die beiden Most Signifikant Bits (MSBs) aus dem Radikanden-Register in das Arbeitsregister verschoben. Weiterhin wird dem aktuellen Wurzelwert, der zu Beginn Null ist, die Bitfolge 01 als LSBs angehängt und in das Wurzelarbeitsregister geladen. Diese Operation entspricht der Umsetzung der Formel (11).

Um die Subtraktion je nach Vergleichsresultat wieder rückgängig machen zu können, wird im Zwischenschritt *Save Workregister* das Arbeitsregister gespeichert.

Die Subtraktion wird dann im Zustand *Do Subtraction* ausgeführt, indem Akkumulator und Wurzelarbeitsregister bitweise in die Subtraktionszelle geschoben werden. Das Ergebnis gelangt direkt zurück in den Akkumulator. Dieser Zustand bleibt N Takte bestehen, entsprechend der Anzahl der Bitstellen.

Im letzten Schritt des Zyklus *Finish Subtraction* wird der Status des Carry-Flags ausgewertet. Ist dieses gesetzt, wird eine Null in das Wurzelregister geschoben und der zwischengespeicherte Wert des ursprünglichen Radikanden zurück in den Akkumulator geladen. Andernfalls wird das Ergebnis im Akkumulator belassen

und eine Eins als neue Wurzelstelle gesetzt. Zusätzlich wird in diesem Schritt überprüft, ob der Radikand vollständig verarbeitet wurde. Entsprechend wird ein neuer Zyklus gestartet oder mit dem *Ready*-Signal die abgeschlossene Radizierung signalisiert.

Da Primzahlen nur ganzzahlige Werte haben können, werden vom vorgestellten Design auch nur ganzzahlige Werte berechnet. Damit ist der Radikand vollständig verarbeitet, sobald der Subtraktionszklus mit den letzten beiden Stellen beendet wurde. Der möglicherweise auftretende Rest verbleibt im Akkumulator und kann von dort ausgelesen werden.

Im Falle, dass die Berechnung noch nicht abgeschlossen ist, wird nicht direkt mit einer neuen Subtraktion gestartet, sondern zuvor in einen weiteren Optimierungszustand *Align Accumulator* gewechselt. In diesem wird die zuvor beschrieben Ausrichtung des Akkumulators an das Wurzelregister durchgeführt, indem so lange zwei Bit des Radikanden in den Akkumulator und Nullen in das Wurzelregister geschoben werden, bis in beiden Registern die führenden Einsen im selben Zweierblock stehen. Falls während des Vorgangs die letzten Bits des Radikanden eingeschoben werden, bevor die Register ausgerichtet sind, muss auch keine Subtraktion mehr ausgeführt werden und die Berechnung ist beendet.

IV. Gesamtsystem

Abbildung 3 zeigt die finale Gesamtstruktur der Schaltung. Der Radikand wird in ein eigenes Register geladen, aus dem pro Zyklus zwei weitere Bits im Akkumulator verarbeitet werden. Da für diese Anwendung nur die Berechnung ganzzahliger Werte gedacht ist, kann auch ein Rest entstehen. Dieser befindet sich nach dem Radizieren im Akkumulator. Die Wurzel wird im *Root* Register erzeugt, indem nach der Subtraktion das Carry-Flag negiert als neues LSB eingeschoben wird. Zu Beginn eines neuen Zyklus wird der neue Wurzelwert ins *Root Work* Register kopiert.

A. Bitbreite

Wie aus der Beschreibung des Algorithmus in Abschnitt II hervorgeht, werden pro Zyklus immer zwei Bit des Radikanden verarbeitet, um ein Bit der Wurzel zu generieren. Damit benötigen die Wurzelregister und Schnittstellen nur die halbe Bitbreite des Radikanden.

$$N_{Wurzel} = N_{Rad}/2 \tag{12}$$

Für die Bitbreite des Restes ist eine Worst-Case Analyse sinnvoll. Wie aus den Funktionsverläufen aus Abbildung 6 ersichtlich wird, ist der Restwert immer bei einem Maximum, wenn der zu radizierende Wert um eins kleiner ist als eine Quadratzahl:

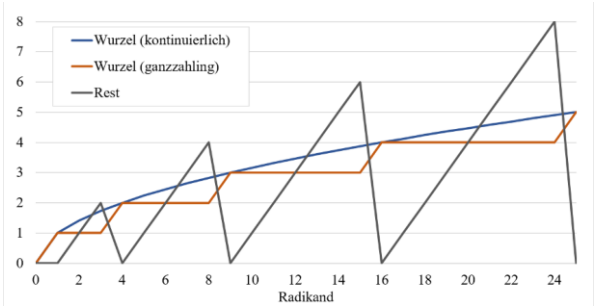


Abbildung 6. Kontinuierliche Wurzelfunktion sowie ganzzahlige Funktionen der Wurzel und des Rests.

$$a^2 - 1 = (a-1)^2 + r \tag{13}$$

$$r = 2 \cdot (a-1) \tag{14}$$

a entspricht der Wurzel einer Quadratzahl, r dem gesuchten Rest. Durch die Gleichungen (13, 14) ergibt sich für das Maximum, dass dieses immer genau doppelt so groß ist wie der Wurzelwert, womit der Restwert maximal eine Binärstelle mehr benötigt als die Wurzel:

$$N_{Rest} = N_{Rad}/2 + 1 \tag{15}$$

B. Berechnungsdauer

Die Bitbreiten des Radikanden haben auch direkte Auswirkung auf die Berechnungsdauer, denn wie bereits erläutert, ist die Subtraktion die mit Abstand zeitaufwändigste Operation. Um das Wurzel-Arbeitsregister vollständig vom Akkumulator zu subtrahieren, werden $N_{Wurzel} + 2$ Takte n_{clk} benötigt.

Hinzu kommen noch drei Takte für die Zustände zum Laden, Sichern und Auswerten der Operanden. Da immer zwei Bit innerhalb eines Zyklus verarbeitet werden, sind insgesamt $N_{Rad}/2$ Subtraktionen nötig, womit sich für die Gesamtberechnungsdauer Formel (16) ergibt:

$$n_{clk} = \left(\frac{N_{Rad}}{2} + 5\right) \cdot \frac{N_{Rad}}{2} \tag{16}$$

Allerdings besitzen die Radikanden nicht immer die volle Bitbreite der Implementierung. Ohne den Zustand Find MSB wäre die Berechnungsdauer unabhängig von der Position des MSB im Radikanden und rein von der maximalen, implementierten Bitbreite abhängig. Durch diesen Optimierungsschritt ist es möglich, die unnötigen Subtraktionszyklen mit

Tabelle I
VERGLEICHSTABELLE VERSCHIEDENER IMPLEMENTIERUNGEN. DIE ANZAHL AN FLIPFLOPS UND LOOKUP TABLES BERUHEN AUF WERTEN MIT XILINX 7-SERIES FPGAS. FÜR ALLE BERECHNUNGSZEITEN T WURDE MIT EINER TAKTFREQUENZ VON 100MHZ GEARBEITET. DER FALL RADICAND $= 1$ IST DER BESTMÖGLICHE FALL. ES WIRD NUR EINE SUBTRAKTION AUSGEFÜHRT, WESHALB AUCH DER FAKTOR $1/2$ EINGERECHNET WIRD.

$N_{Rad,Imp}$	512	1024	4096	6144
# FF	1,6k	3,1k	12,3k	18k
# LUT	1,1k	2,4k	8,3k	12k
$n_{clk} \mid \{Radicand = 1\}$	516	1028	4,1k	6,1k
$t \mid \{^{Radicand=1}_{f=100MHz}\}$	5.16μs	10.3μs	41μs	61,5μs
$n_{clk} \mid \left\{N_{Rad} = \frac{N_{Rad,Imp}}{2}\right\}$	16,8k	66,4k	1,1M	2,36M
$t \mid \{^{N_{Rad}=\frac{N_{Rad,Imp}}{2}}_{f=100MHz}\}$	168μs	664μs	10,5ms	23,7ms
$n_{clk} \mid \{N_{Rad} = N_{Rad,Imp}\}$	33k	132k	2,1M	4,7M
$t \mid \{^{N_{Rad}=N_{Rad,Imp}}_{f=100MHz}\}$	33,4μs	1,32ms	21ms	47ms

nicht signifikanten Stellen zu überspringen. Aufgrund der Zweibit-Verschiebung werden zum Ausrichten des Radikanden $[(N_{Rad,Imp} - N_{Rad})/2]$ Takte benötigt. Für die Berechnungsdauer mit Optimierung lässt sich damit Gleichung (17) herleiten:

$$n_{clk} = \left(\frac{N_{Rad}}{2} + 5\right) \cdot \frac{N_{Rad}}{2} + [(N_{Rad,Imp} - N_{Rad})/2] \tag{17}$$

Durch den zweiten Optimierungsschritt *Align Accumulator* ist eine zusätzliche Reduktion der Berechnungsdauer möglich. Allerdings ist diese Effizienzsteigerung rein vom Radikandenwert abhängig und kann nicht durch eine Formel dargestellt werden. Durch Langzeitversuche mit Zufallszahlen ließ sich ermitteln, dass im statistischen Mittel noch einmal die Hälfte an Subtraktionen durch diese Optimierung eingespart werden können:

$$n_{clk} = \frac{1}{2}\left(\frac{N_{Rad,Imp}}{2} + 5\right) \cdot \frac{N_{Rad}}{2} + [(N_{Rad,Imp} - N_{Rad})/2] \tag{18}$$

$$n_{clk} \approx \frac{1}{2} \cdot \frac{1}{4} \cdot N_{Rad,Imp} \cdot N_{Rad} \tag{19}$$

Bei sehr großen Bitbreiten kann Formel (19) für eine grobe Approximation dienen. Dabei wird deutlich, dass der erste Optimierungsschritt zu einer linearen Abhängigkeit der Berechnungsdauer von der Bitbreite des zu radizierenden Wertes führt. Die Erhöhung der Implementierungsbitbreite wirkt sich dagegen quadratisch aus. Die zweite Optimierung wird mit dem Faktor $1/2$ berücksichtig.

V. FAZIT

Ausgehend von der Problemstellung der Entwicklung einer Hardwarelösung zum Radizieren sehr großer Zahlen ist eine sehr anwendungsspezifische FPGA-Implementierung entstanden. Die zu Beginn beschriebenen allgemeinen Ansätze wurden so umgesetzt, dass die Implementierung durch den Einsatz der GENERATE-Anweisung in VHDL skalierbar ist. Beim Einsatz entsprechend komplexer FPGAs können mehrere parallele Strukturen in einem Chip implementiert werden, von denen jede Bitbreiten von bis zu mehreren tausend Stellen verarbeiten kann.

Tabelle I zeigt den Vergleich verschiedener Implementierungen mit einem XILINX Zynq 7020 FPGA. Zu sehen ist, dass selbst 6k Bit breite Radikanden mit 100 MHz berechnet werden können. Bei so großen Bitbreiten stellen die Hardware-Ressourcen des FPGAs die Grenzen dar und nicht die Schaltung selbst. Insbesondere ist die Zahl der benötigten Flipflops die bestimmende Limitierung. Bei 8k Bit großen Operanden werden bereits mehr als 24.576 Flipflops benötigt. In einem XILINX Zynq 7020 FPGA (106.400 Flipflops) könnten daher maximal 4 solcher Komponenten implementiert werden.

Ferner ist für die Zukunft geplant, die Schieberegister in Form von RAM Bausteinen zu implementieren, die über z.B. 16 Bit breite seriell/parallel-Register beschrieben bzw. gelesen werden. Das geht immer dann, wenn nicht parallel auf die Registerinhalte zugegriffen werden muss. Im vorliegenden Fall müsste dann auf die Möglichkeit verzichtet werden, die Daten parallel mit dem System auszutauschen. Wenn das RAM jedoch mit einer Breite des Datenbusses ausgestattet ist, die dem eines zusätzlich implementierten embedded Prozessors entspricht, dann könnten die Daten auf diesem Wege immer noch sehr schnell transferiert werden.

Inzwischen sind allerdings FPGAs mit weit mehr als 500.000 Flipflops verfügbar. Nächstes Zwischenziel des Gesamtprojektes ist der Entwurf eines „Giant Number Numerical Processors“. Dieser soll neben der hier beschriebenen Wurzelfunktion sämtliche mathematische Grundoperationen auf sehr große Zahlen beherrschen und auf Maschinenebene programmierbar sein. Damit wird es dann möglich, Programme zu schreiben, die z.B. den AKS Algorithmus direkt abbilden.

LITERATURVERZEICHNIS

[1] D. Fowler und E. Robson, *Square Root Approximations in Old Babylonian Mathematics*, in Historica Mathematica 25, Elsevier B.V., 1998, p. 366–378.
[2] G. Merziger, G. Mühlbach, D. Wille und T. Wirth, *Formeln + Hilfen Höhere Mathematik*, Binomi, Barsinghausen, 2014.
[3] P. D. N. Wehn, *Microelectronic Circuit and Systems, Arithmetic Modules*, Mircoelectronic Systems Design Research Group, Kaiserslautern, 2017.
[4] J.L. Hennessy, D.A. Patterson, *Computer Architecture – a Quantitative Approach*, 2. Auflage, Morgan Kaufmann, 1996.

David Schall studierte Elektronik und Informationstechnik an der Hochschule Heilbronn und erhielt im Jahre 2017 den akademischen Grad B. Eng. Er setzt derzeit sein Studium zum Erlangen des Grad des Masters an der Technischen Universität in Kaiserslautern fort.

Norbert Reifschneider wurde 1997 von der Universität Kassel zum Dr.-Ing. promoviert. Von 2005 bis 2009 lehrte er als ordentlicher Professor an der Hochschule Augsburg, seitdem an der Hochschule Heilbronn in den Gebieten Programmierung, Digitaltechnik, Programmierbare Logikschaltungen, Embedded Systems, Digitale Signalverarbeitung und Mikrocontroller / Mikroprozessoren.

Hochschule Offenburg
offenburg.university

Implementierung von Softcore-Prozessoren und/oder weiteren IPs (Intellectual Property) in FPGAs

Artur Werner, Patrick Moser, Elke Mackensen

***Zusammenfassung*—Die zunehmende Integration von kompletten Systemen auf einem Chip (System-on-Chip, SoC) erfordert auch immer die Integration einer Recheneinheit bzw. eines Prozessorkerns. Möchte man insbesondere Low-Power-SoC-Systeme entwickeln, z.B. drahtlose Sensor-SoC-Systeme für Anwendungen im Rahmen von Industrie 4.0, ist die Implementierung eines solchen Prozessorkerns mit hohen Herausforderungen verbunden. Prinzipiell können hierfür verschiedene Ansätze verfolgt werden, nämlich die Implementierung einer Hardcore-Prozessor-IP (IP = Intellectual Property) oder einer Softcore-Prozessor-IP. Im vorliegenden Beitrag wird zunächst auf den derzeitigen Stand der Technik verfügbarer Hardcore- oder Softcore-Prozessoren unter den Randbedingungen der Low-Power-Anforderungen und der weiten Verbreitung des Cores in industriellen Anwendungen eingegangen. Schließlich werden die Ergebnisse der Implementierung und Evaluierung eines derzeit frei verfügbaren 16-bit MSP430-kompatiblen Softcore-Prozessors auf einem Altera-Cyclon-FPGA vorgestellt. Aus den Ergebnissen wird ein entsprechendes Fazit für die Implementierung von Low-Power-SoC-Systeme gegeben.**

***Schlüsselwörter*—FPGA, System-on-Chip, Soft- und Hardcore-Prozessoren, Intellectual Properties, Low-Power-SoC-Systeme, CRC, Wishbone.**

I. Einleitung

Hochintegrierte System-on-Chip-Systeme (SoC-Systeme) finden eine immer größere Verbreitung in der Industrie. Insbesondere drahtlos kommunizierende Sensor-SoC-Systeme bilden z.B. die Grundlage für die vielfältigen Anwendungen im Rahmen von Industrie 4.0 [1, 2] oder medizinischen Anwendungen [3]. Der Entwurf von SoC-Systemen erfordert aber auch immer die Integration eines Prozessorkerns. Gerade an den Entwurf von drahtlos kommunizierenden Sensor-SoC-Systemen, die eventuell auch noch per Energy-Harvesting betrieben werden sollen, sind hohe Anforderungen gestellt, die natürlich auch bei der Integration eines Prozessorkerns berücksichtigt werden müssen [4]. Derartige SoC-Syteme müssen eher minimalistisch in Bezug auf Energiebedarf, Größe und Kosten ausgelegt werden. Hinzu kommen spezielle Anforderungen an die Integration eines Prozessorkerns, wie beispielsweise die Anforderung, dass dies ein in der Industrie weitverbreiteter Core bzw. ein weit verbreitetes Instruction-Set sein sollte, um z.B. Standardtools für die Firmware-Entwicklung und das Debugging einsetzen zu können.

Prinzipiell können drei Ansätze bei der Integration eines Prozessorcores verfolgt werden:

- Die Implementierung einer Hardcore-Prozessor-IP. Hierunter versteht man das Vorliegen der Prozessor- Hardwareblöcke in Form eines vorliegenden Layouts in einer vorgegebenen Zieltechnologie, z.B. in einer CMOS-180-nm-Technologie. Bei FPGAs besteht derzeit der Trend, auf dem FPGA bereist einen leistungsfähigen Hardcore-Prozessor zu integrieren. Ein Beispiel ist die Zynq-Familie von Xilinx (https://www.xilinx.com/products/silicon-devices/soc.html) mit einem ARM-Cortex-Kernel.
- Die Implementierung einer Softcore-Prozessor-IP. Hierunter versteht man das Vorliegen der Prozessor-IP als synthetisierbarer „Software-Code“ in Form einer nachträglich noch änderbaren Hardwarebeschreibungssprache wie VHDL, Verilog oder einem optimierten, vorkompilierten RTL-Code (RTL = Register Transfer Level). Softcore-Prozessor-IPs können auf beliebige Zieltechnologien synthetisiert werden.
- Die Entwicklung eines eigenen Prozessorkerns, wie beispielsweise in [5] beschrieben. Dies ist sicherlich eine Variante, um einen optimalen Prozessorkern zu entwerfen, hat jedoch verschiedenen Nachteile, die dazu führen, dass dieser Ansatz nicht weiter verfolgt wird: Die Entwicklung eines eigenen Prozessorcores bedarf auch immer der Entwicklung und Pflege von den entsprechenden Entwicklungswerkzeugen (Programmierumgebung, Compiler…), was im Zeitalter sich permanent ändernder Betriebssysteme nur schwer zu pflegen ist. Insbesondere wird aber der oben aufgeführten Anforderung nicht nachgekommen, einen der Industrie weitverbreiteter Core zu implementieren.

Im Abschnitt II des Beitrags wird zunächst auf die Anforderungen an die Prozessor-Integration mit dem Fokus auf die Entwicklung energieautarker drahtlo-

Artur Werner, awerner@stud.hs-offenburg.de,
Patrick Moser, patrick.moser@hs-offenburg.de,
Prof. Dr. Elke Mackensen, elke.mackensen@hs-offenburg.de

ser Sensor-SoC-Systeme eingegangen. Es wird aufgezeigt, warum die Variante der Implementierung einer Softcore-Prozessor-IP gegenüber der Variante der Implementierung einer Hardcore-Prozessor-IP wesentliche Vorteile hat.

In Abschnitt III wird auf den Stand der Technik verfügbarer Softcore-Prozessor-IPs eingegangen, welche die aufgestellten Anforderungen erfüllen. Auf Grund der verfügbaren Softcore-Prozessoren wurde ein 16-bit MSP430-kompatibler Softcore-Prozessor auf einem Altera-Cyclon-FPGA implementiert und evaluiert. Dabei wurden die folgenden Aspekte untersucht:

- Wie schnell ist es möglich einen solchen Softcore-Prozessor auf einem FPGA zu implementieren?
- Welche Bestandteile der üblichen MSP430-Mikrocontroller-Serien sind durch den Softcore bereits abgedeckt?
- Welche Möglichkeiten bestehen, den Softcore um eigene oder auch frei verfügbare IP-Module zu erweitern? Die Entwicklung einer beispielhaften Applikation mit der Implementierung eines eigenen CRC-Moduls und eines frei verfügbaren IP-Moduls in VHDL zeigen die einfachen Erweiterungsmöglichkeiten auf.
- Welche Möglichkeiten der Programmierung und des Debuggens sind für den Softcore-Prozessor vorhanden?
- Welche Ressourcen werden von der Implementierung im FPGA verbraucht?

In Abschnitt IV wird auf diese Implementierung und die daraus resultierenden Ergebnisse detailliert eingegangen.

Zusammenfassend wird auf Grund der Resultate in Abschnitt V ein entsprechendes Fazit für die Entwicklung von insbesondere drahtlosen energieautarken Sensor-SoC-Systemen gegeben.

II. Anforderungen an die Prozessor-Integration mit dem Fokus auf energieautarke drahtlose Sensor-SoC-Systeme

Die wichtigste Anforderung bei Entwicklung drahtloser energieautarker Sensor-SoC-Systeme ist es, dass alle Komponenten eine geringe Strom- bzw. Leistungsaufnahme besitzen. Diese Anforderung gilt natürlich auch für den zu integrierenden Prozessorkern.

Um eine möglichst geringe Leistungsaufnahme zu erzielen, sollten die folgenden Anforderungen erfüllt sein:

- Für drahtlose energieautarke Sensor-SoC-Systeme ist eine 8-bit- bzw. 16-bit-Prozessor-Architektur zumeist für die Datenverarbeitung ausreichend.
- Die Prozessorarchitektur soll möglichst schlank sein, um nur die wirklich notwendigen Komponenten auf einem Chip integrieren zu müssen. Es soll ein minimaler Logikressourcenverbrauch in der entsprechenden Zieltechnologie erreicht werden.

Des Weiteren sind folgende Anforderungen an die Prozessor-Integration gerichtet:

- Die Prozessorarchitektur soll eine standardisierte Schnittstelle aufweisen, über welche weitere Komponenten hinzugefügt und angebunden werden können.
- Die Prozessorarchitektur soll mit verschiedenen Zieltechnologien umgesetzt werden können. Für einen ersten Test beispielsweise auf einem FPGA und für endgültige Applikationen beispielsweise in einer CMOS-65-nm-Technologie.
- Aus Sicht der Anwendung der drahtlosen Sensor-SoC-Systeme soll die Prozessorintegration möglichst kostengünstig sein.
- Der Prozessorkern soll ein in der Industrie weitverbreiteter Core sein bzw. ein weit verbreitetes Instruction-Set aufweisen, um z.B. Standardtools für die Firmware-Entwicklung und das Debugging einsetzen zu können.
- Es soll ein On-Chip-Debugging möglich sein.

Die oben genannten Anforderungen führen zum Ausschluss der Implementierung eines Hardcore-Prozessors. Zum einen weisen diese im Allgemeinen nicht die notwendige Flexibilität auf, um die Anforderung zu erfüllen, dass der Prozessor in einer minimalen Konfiguration verwendet werden kann, da diese als tatsächliches Layout in einer bestimmten Technologie vorliegen. Zum anderen sind Hardcore-Prozessoren meist mit entsprechenden Lizenzkosten verbunden. Ist der Hardcore-Prozessor bereits auf dem FPGA integriert, so fallen in den meisten Fällen zwar keine Lizenzkosten an, aber dafür werden entsprechende Stückpreise für die FPGAs verlangt.

Prinzipiell kommen deshalb für die anvisierte Realisierung drahtloser energieautarker Sensor-SoC-Systeme nur Softcore-Prozessoren in Frage, auf die nun im folgenden Abschnitt näher eingegangen wird.

III. Verfügbare Softcore-Prozessoren mit dem Fokus auf energieautarke drahtlose Sensor-SoC-Systeme

Zu den im Abschnitt II genannten Anforderungen an die Prozessoren, sind weitere Vergleichskriterien notwendig, um Softcore-Prozessoren miteinander vergleichen zu können. Die Auswahl der Vergleichskriterien wurde durch ein Paper unterstützt, welches auf der 24. Internationalen „Field Programmable Logic and Applications (FPL)“ Konferenz veröffentlicht wurde [6]. Im genannten Paper hat eine Gruppe von Ingenieuren, über 180 Open-Source-Prozessoren untersucht und Prinzipien vorgestellt, um einen geeigneten Softcore-Prozessor auszuwählen. Die dort aufgeführten wichtigsten Kriterien für einen Vergleich sind [6]:

- Art der Lizenzierung,
- Befehlssatzarchitektur (instruction set architecture, ISA),
- Compiler und Assembler,
- sowie Dokumentation.

Für diese Arbeit wurden die oben genannten Vergleichskriterien durch folgende Kriterien erweitert:

- Hardwarebeschreibungssprache des Softcores,
- Wortbreite des Prozessors,
- Schnittstelle für Erweiterbarkeit
- und Verfügbarkeit als Open-Source.

Eine wichtige Quelle für Open-Source-Prozessoren ist die Internet-Plattform OpenCores (siehe https://opencores.org/). Dort werden Open-Source-Prozessoren in verschiedenen Entwicklungsstadien aufgeführt: *Planung*, *Pre-Alpha*, *Alpha*, *Beta*, *Stabil* und *Ausgereift*. Alle Prozessoren die sich vor der *stabilen* Entwicklungsstufe befinden, sind für einen weiteren Vergleich nicht relevant. Prozessoren die sich bereits im *stabilen* Entwicklungsstadium befinden, beinhalten alle vorgesehen Funktionen, enthalten eventuell kleinere Bugs, sind jedoch zuverlässig einsetzbar. Bei weiteren Quellen für Softcore-Prozessoren ist nicht notwendigerweise der Entwicklungsstatus angegeben oder keine Dokumentation vorhanden, weshalb diese nicht näher betrachtet wurden und schwerer zu bewerten sind.

Anhand der in Abschnitt II genannten Anforderungen und den in diesem Abschnitt zusätzlich aufgeführten Vergleichskriterien, wurden über 40 Softcore-Prozessoren in dieser Arbeit verglichen. Das Ergebnis des Vergleichs ist in Tabelle I festgehalten. Auf einige der Kriterien soll im Folgenden näher eingegangen werden.

Wie man der Tabelle I entnehmen kann, gibt es verschiedene Lizenzmodelle. MicroBlaze, PicoBlaze, Nios II und der Cortex-M1 stehen unter einer zahlungspflichtigen Lizenz, wobei die Lizenz über den Kauf eines FPGAs erworben wird. Mit *Proprietary* (deutsch: firmeneigen, geschützt) sind diese Prozessoren Eigentum der jeweiligen Firma.

Für Open-Source Projekte stehen verschiedene Open-Source Lizenzen zur Verfügung. Die bekanntesten und auch in dieser Tabelle vorzufinden sind:

- General Public License (GPL)
- Lesser General Public License (LGPL)
- Massachusetts Institute of Technology (MIT)
- Berkeley Software Distribution (BSD).

Die GPL-Lizenz gibt vor, wenn ein unter der GPL lizenziertes Werk vertrieben wird, dieses nur unter der gleichen Lizenz weiter vertrieben werden darf. Die LGPL-Lizenz löst diese Einschränkung. Die MIT- und BSD-Lizenz gehören zu den freizügigen Open-Source-Lizenzen, mit denen eine Wiederverwertung von lizenzierten Inhalten erlaubt wird. All die genannten Lizenzen erlauben es, Software auszuführen, zu studieren, zu ändern und zu verbreiten.

Auch bei dem Vergleichskriterium Compiler- und Assembler ist zu erkennen, dass es verschiedene Varianten gibt. Mit der GNU-Compiler-Collection (GCC) ist eine Kompiliersammlung gegeben, die in der Lage ist mehrere Programmiersprachen zu übersetzen und wird daher von den meisten Prozessoren benutzt.

Wie bereits in Abschnitt II erläutert, ist eine wichtige Anforderung die Erweiterbarkeit des Prozessors, um zusätzliche Komponenten über eine möglichst standardisierte Schnittstelle anzubinden. Als standardisierte Schnittstellen sind zum einen die Advanced Microcontroller Bus Architecture (AMBA) und zum anderen Wishbone vorzufinden. Beide definieren detailliert, wie eine Kommunikation zwischen Modulen zu erfolgen hat. AMBA ist eine Spezifikation von ARM und wird unter anderem mit ARM-Prozessoren genutzt (siehe https://www.arm.com/products/processores). Wishbone hingegen ist eine Open-Source Spezifikation und wurde durch opencores.org erstellt [7].

Die Entscheidung, welcher Prozessor ausgewählt werden soll, ist ganz davon abhängig, welche Anwendung beziehungsweise welches System entwickelt werden soll. Für drahtlose energieautarke Sensor-SoC-Systeme ist wie zuvor aufgezeigt eine 8-bit- bzw. 16-bit-Prozessor-Architektur zumeist ausreichend. Wie man an der Tabelle I erkennen kann, sind die Prozessoren, die mit einander verglichen wurden farblich markiert. Rot markierte haben zwei oder mehr Unstimmigkeiten zu den aufgestellten Anforderungen in Abschnitt II, gelb markierte eine und grüne keine Unstimmigkeit mit den Kriterien. Die grün markierten, sind Prozessoren mit denen ohne weitere Bedenken ein System aufgebaut werden kann.

Aus der Tabelle I wird ersichtlich, dass einige Prozessoren rot markiert sind. Wie zum Beispiel der miniMIPS Softcore-Prozessor, welcher mit der Nummer 16 aufgelistet ist. An diesem, sowie an weiteren rot markierten Prozessoren, fehlen wichtige Angaben bezüglich Architektur, Schnittstelle oder Lizenz. Ohne einer vorhandenen beziehungsweise ausführliche Dokumentation können notwendige Informationen nicht sichergestellt und verifiziert werden.

Die Entscheidung für einen zu implementierenden Prozessor fiel auf den NEO430 (https://github.com/stnolting/neo430), welcher mit der Nummer 17 in Tabelle I gelistet ist. Die Tatsache, dass dieser Prozessor auf der MSP430-Architektur basiert, die insbesondere durch energiesparende Mechanismen bekannt ist, macht ihn für die Nutzung und Untersuchung sehr interessant.

Tabelle I
ERGEBNISTABELLE BEIM VERGLEICH DERZEIT VERFÜGBARER SOFTCORE-PROZESSOREN.

Nr.	Name	Architektur	Bits	Lizenz	Open-Source	Download-bar	HDL	Dokumentation	Compiler & Assembler	Schnittstelle
1	aeMB	MicroBlaze	32	LGPL	ja	ja	Verilog		GCC	Wishbone
2	ARM-compatible core	ARMv2	32	LGPL	ja	ja	Verilog		GNU toolset	Wishbone
3	Atlas	inspired ARM ar	16	GPL	ja	ja	VHDL			Wishbone
4	Cortex-M1	ARM V6	32	Proprietary	nein	nein	VHDL/Verilog			CorLink (AMBA)
5	CPU86	8088 processor		GPL	ja	ja	VHDL			
6	F32c	RISC-V	32	BSD	ja	ja	VHDL		GCC	
7	FPz8		8	LGPL	ja	ja	VHDL		ZDS-II DIE	
8	Lattice Micro32	LatticeMicro32	32	Open IP Core	ja	ja	VHDL/Verilog	ja		Wishbone
9	Lattice Micro8	LatticeMicro8	8	Open IP Core	ja	ja	VHDL/Verilog	ja		Wishbone
10	LEON2	SPARC V8	32	LGPL	ja	ja	VHDL	ja	RCC (RTOS)	AMBA
11	LEON3	SPARC V8	32	GPL	ja	ja	VHDL	ja	BCC	AMBA
12	LEON4	SPARC V8	32		ja/nein		VHDL		BCC	AMBA-2.0
13	MAIS		32	CC BY-NC 3.0	ja	ja	VHDL	ja		
14	MC8051		8	LGPL	ja	ja	VHDL	ja	C51 Development Tools	
15	MicroBlaze	MicroBlaze	32	Proprietary	nein	nein	VHDL/Verilog	ja		PLB, OPB, FSL, LMB, AXI4
16	miniMIPS		32	LGPL	ja	ja	VHDL			
17	NEO430	MSP430	16	LGPL	ja	ja	VHDL	ja	msp430-gcc	Wishbone
18	Nios II	Nios II	32	Proprietary	nein	nein	VHDL/Verilog	ja		Avalon switch fabric
19	OpenFire	MicroBlaze	32	MIT	ja	nein	Verilog			
20	openMSP430	MSP430	16	BSD	ja	ja	Verilog	ja	GCC	
21	OpenRISC	OpenRisc 1000	32	LGPL	ja	ja	Verilog	ja	GCC	Wishbone
22	OpenSPARC T1	SPARC V9	64	GPL	ja	ja	Verilog	ja	Solaris Studio	
23	PacoBlaze	PicoBlaze	8	modified BSD	ja	ja	Verilog	ja	KCAsm	
24	Panda	RISC	8	Boost Lizenz	ja	ja	VHDL	ja	HASM (Assembler)	
25	PicoBlaze	PicoBlaze	8	Proprietary	ja	ja	VHDL/Verilog	ja		
26	Plasma	MIPS I	32		ja	ja	VHDL		gcc-mips-11	
27	Propeller 2		32							
28	Proteus	RISC	16	Boost Lizenz	ja	ja	VHDL	ja	HASM (Assembler)	
29	PULPino	RISC-V	32	Solderpad Har	ja	ja		ja		
30	Simply RISC S2 Core	SPARC V9	64	GPL	ja	ja	Verilog	ja	GCC	Wishbone/AMBA
31	SpartanMC		18		ja	ja				
32	TG68	68000 CPU	16	LGPL	ja	ja	VHDL			
33	Ucore		16		ja	ja				
34	Zet	IA-32		GPL	ja	ja				
35	ZPUino		32							
36	Zylin CPU (ZPU)		32	BSD	ja	ja	VHDL	ja	GCC	

IV. IMPLEMENTIERUNG EINES 16-BIT MSP430-SOFTCORE-PROZESSORS AUF EINEM ALTERA-CYCLONE-FPGA

Das Blockschaltbild des für eine FPGA-Implementierung ausgewählten Softcore-Prozessors NEO430, ist in Abbildung 1 dargestellt. Der NEO430 verfügt über Standard-Komponenten wie Timer, einen Watchdog Timer (WDT), einer Wishbone-Schnittstelle (WB32), einer USART-Schnittstelle, GPIOs und einen internen Bootloader, sowie internen Speicher für Programm-Code und Daten.

Optionale, bereits vorhandene Komponenten sind in Abbildung 1 mit gestrichelter Umrandung dargestellt. Dies bedeutet, sie können weggelassen werden, um die Größe des Systems zu minimieren und um somit Ressourcen zu sparen. Dies erfolgt über die Generic-Konfiguration im Top-Level-Design der VHDL-Code-Implementierung. Die Wishbone-Schnittstelle ermöglicht die Anbindung von zusätzlicher Hardware, wodurch ein System perfekt an die Anforderung angepasst werden kann.

Die bereitgestellten Open-Source-Daten für den NEO430-Prozessor (VHDL-Code, Dokumentatom, Programmierdateien usw.) sind so aufgebaut, dass dieser sofort mit einem Test-Setup auf einen FPGA geladen und die verschiedenen Fähigkeiten ausgetestet werden können. Die Generierung des C-Programmes erfolgt über ein einziges „make“-Kommando und ermöglicht schnelle Änderungen des Programmes. Mit dem NEO430 wurde ein Beispielsystem aufgebaut, welches ein eigen entwickeltes CRC-Modul über die Wishbone-Schnittstelle anbindet.

A. *NEO430 im Vergleich zum MSP430*

Der NEO430 ist **kein** MSP430-Klon, sondern eine komplett neue Implementierung, die jedoch mit der Instruction-Set-Architektur des MSP430 voll kompatibel ist. Dadurch ist es möglich für die Softwareentwicklung des NEO40 den msp430-gcc Compiler von Texas Instruments (TI) zu verwenden. Da es sich bei dem NEO430 um eine komplett neue Implementierung handelt, sind diverse Unterschiede mit der MSP430-Produktlinie von TI vorhanden.

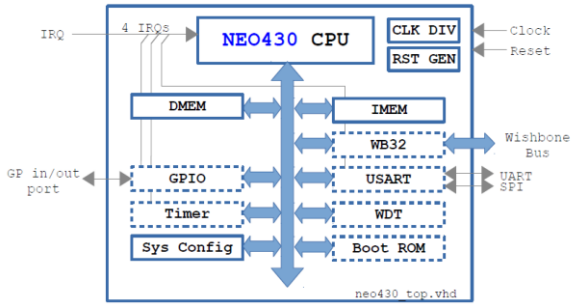


Abbildung 1. NEO430 Blockschaltbild [8].

Die wesentlichen Unterschiede sind wie folgt [8]:

- Kein Hardware-Multiplizierer.
- Speziell angepasste Speicheradressierung, benötigt das NEO430-Linker- und -Compiler-Skript.
- Nur 4 anstatt 16 Interrupt-Kanäle.
- Speicherkapazität maximal 32 kB für das Programm und 28 kB für Daten.
- Eine Taktdomäne für den kompletten Prozessor.
- Nur ein Energiesparmodus.
- Wishbone-kompatible Schnittstelle zur Anbindung von IP-Modulen.
- Interner Bootloader mit Text-Interface.

Zwar existieren gegenüber dem MSP430 deutliche Unterschiede, jedoch besteht jederzeit die Möglichkeit den NEO430 durch eigene Module/Komponenten über die Wishbone-Schnittstelle zu erweitern. So könnte beispielsweise ein für die Anwendung optimal angepasster Multiplizierer auf dem FPGA erzeugt werden und über Wishbone angebunden werden.

Neben den bereits genannten Unterschieden, gibt es noch weitere Unterschiede in Bezug auf Komponenten, die normalerweise in MSP430-Mikrocontroller von TI enthalten sind und im NEO430 nicht vorhanden sind. Diese sind insbesondere Mixed-Signal-Komponenten wie Komparatoren, Verstärker, sowie ADC und DAC.

B. Programmierung und Debbuging

Die Programmierung des NEO430 erfolgt über eine von zwei Möglichkeiten:

1) Über die UART-Schnittstelle: Der NEO430 ist mit einer UART-Schnittstelle ausgestattet, über die das kompilierte Programm in den Programspeicher des NEO430 geladen werden kann. Das Kompilieren des Programmes erfolgt mit einem „make.bat“- und „compile.bat“-Skript. Durch die Ausführung der Skripte entsteht eine binäre Datei „main.bin“. Diese Datei kann über die UART-Schnittstelle in den Programmspeicher hochgeladen werden. Dazu muss eine Verbindung vom Computer mit dem FPGA hergestellt werden. Zur Übertragung des Programms über die UART-Schnittstelle kann beispielsweise ein einfaches Terminal-Programm genutzt werden. Der NEO430 muss über einen Soft-Reset, in den Bootloader-Zustand versetzt werden. Mit diversen Einstellungen, wie zum Beispiel der Baudrate und Parity-Bit-Einstellung, erfolgt die Übertragung der binären Datei.

2) Über die SPI-Schnittstelle: Bei dieser Alternative wird ein nicht-flüchtiger Speicherbaustein, z.B. ein EEPROM genutzt. Dieses ist über die SPI-Schnittstelle des NEO430 mit dem Prozessor verbunden. Beschrieben wird das EEPROM einmalig mit Hilfe des Bootloaders und der UART-Schnittstelle. Der Bootloader ist zusätzlich in der Lage das Programm automatisch aus dem EEPROM zu beziehen und auszuführen. Dies erfolgt, wenn der Prozessor neu konfiguriert wird und von der Adresse des Bootloaders startet. Es besteht die Möglichkeit die Konfigurationsdatei für den FPGA im gleichen oder einen anderen EEPROM abzuspeichern. Dadurch kann das System, nach einem Verlust der Spannungsversorgung, den FPGA selbst konfigurieren und den NEO430 mit einem Programm programmieren.

Aus Abschnitt II geht hervor, das Standardtools für die Programmierung und des Debugging gefordert sind. In dieser Arbeit wurde die Entwicklungsumgebung Eclipse genutzt. Hierzu wurden die beiden Skripte des NEO430 in die Eclipse IDE eingebunden, um anschließend das Programm zu kompilieren. Das Hochladen des kompilierten Programms erfolgt weiterhin über einen der beiden genannten Methoden. Das On-Chip-Debuggen des NEO430 ist nicht möglich, da eine Debugging-Einheit fehlt. Damit dies in Zukunft möglich wird, müsste eine Debugger-Einheit entwickelt und an den internen Bus angeknüpft werden, um so den Zugriff auf die CPU zu erhalten.

C. Erweiterbarkeit durch zusätzliche Hardware-IPs

Die Wishbone-Kommunikationsschnittstelle ist eine SoC-Interconnection-Architektur für IP-Module [7]. Der Nutzen dieser Architektur findet sich in der Design Wiederverwendung von IP-Modulen, sowie einer standardisierten Integration und Kommunikation zwischen mehreren IP-Modulen auf einem SoC wieder.

Die Wishbone-Spezifikation gibt nicht vor, wie die Implementierung aussehen soll, sondern wie die Kommunikation zwischen zwei oder mehr Teilnehmern erfolgt. Daher ist der Entwickler für die Implementierung von Schreib- und Lesezugriffe auf das jeweilige Modul zuständig.

Beim NEO430 gibt es bereits Funktionen, um Daten an einen Wishbone-Slave zu senden oder anzufordern. Die Slave-seitige Anbindung an den Wishbone-Bus, erfolgt über eine Adressdeklarierung und wird mit einem Prozess zur Signalerkennung des Bus-Systems gelöst. Dieser Prozess überprüft die Signalzustände von Strobe, Cycle, Adresse, Write Enable und Data Select. Die Bestätigung des Slaves erfolgt mit einem

```
-- wb read/write process
  read_write_proc: process(clk_i)
  begin
   if(rising_edge(clk_i)) then
    if(rst_i = '0') then
     -- global reset
    elsif((wb_stb_i = '1') AND (wb_cyc_i = '1')) then
     if(wb_we_i = '1') then -- write access
      case wb_addr_i is
       when slave_addr => -- slave address
        case wb_sel_i is
         when "0011" => -- die unteren 2 Bytes von wb_dat_i(31
            downto 0)
          data_in <= wb_dat_i(15 downto 0);
          -- start_crc <= '1';
         when others =>
          data_in <= (others => '0');
        end case;
        wb_ack <= '1';
       when others =>
      end case;
     elsif(wb_we_i = '0') then -- read access
      case wb_addr_i is
       when slave_addr => -- slave address
        -- do smthfor read access
        wb_ack <= '1';
       when others =>
      end case;
     end if;
    else -- (wb_stb_i OR wb_cyc_i) = '0'
     wb_ack <= '0';
    end if;
   end if;
  end process read_write_proc;
```

Abbildung 2. Eine in VHDL implementierte Slave-seitige Wishbone-Schnittstelle.

Acknowledgment, womit dem Master mitgeteilt wird, dass die Übertragung erfolgreich war.

Eine solche Wishbone-Implementierung in VHDL ist in Abbildung 2 zu sehen. Der Prozess ist für einen Schreib- und Lesezugriff zuständig. Dazu werden bei jeder steigenden Taktflanke, die zwei Signale wb_stb_i und wb_cyc_i überprüft, ob diese vom Master gesetzt worden sind. Wenn das der Fall ist, wird zuerst geprüft ob es sich um ein Schreib- oder Lesezugriff handelt, anschließend mit einer Case-Anweisung die Adresse vom Signal wb_addr_i und mit der nächsten Case-Anweisung der Data Select geprüft. Nach der letzten Überprüfung erfolgt die Zuweisung der Wishbone Daten an das interne Daten Signal und das Setzen des Acknowledgments Signals wb_ack. Bei der Implementierung wurde auf eine allgemeingültige Lösung gesetzt, damit diese einfach übernommen und auf die jeweiligen Anforderungen eines IP-Moduls angepasst werden kann. Dazu muss lediglich die Case-Anweisung für den Daten Select und die Zuweisung an interne Signale angepasst werden.

Um die einfache Anbindung eigener Komponenten über eine Wishbone-Schnittstelle zu testen, wurde ein eigenes CRC-Modul (CRC = Cyclic redundancy check) in VHDL entwickelt. Das CRC-Modul soll 16-bit-Daten vom NEO430 über das Wishbone-Interface empfangen, anschließend eine CRC-Berechnung durchführen und mit dem empfangenen CRC vergleichen. Bei einer erfolgreichen Übereinstimmung der beiden CRC, werden von den empfangenen 16-bit-Daten, 8-bit-Informationen an LEDs zur visuellen Überprüfung ausgegeben.

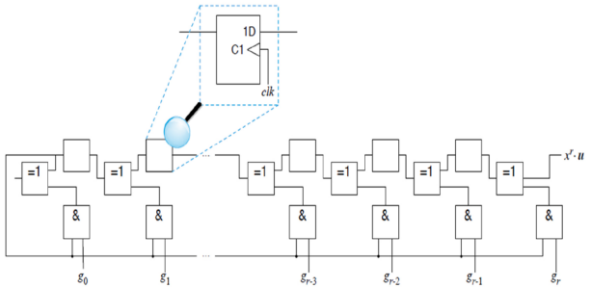


Abbildung 3. Aufbau einer CRC-Encoder Schaltung [11].

Die Implementierung arbeitet nach der Funktionsweise, wie sie in Abbildung 3 zu sehen ist. Das CRC-Modul besitzt ein Schieberegister. Die einzelnen Elemente des Schieberegisters, werden mit XOR-Gattern verbunden. An den zweiten Eingang des XOR-Gatters sitzt ein UND-Gatter, über welches die Anbindung des Generatorpolynoms für die CRC-Berechnung erfolgt. Befindet sich im MSB des Schieberegisters eine logische/binäre 1, so wird das Generatorpolynom über die gr-Eingänge der UND-Gatter an die XOR-Gatter weitergereicht. Anschließend wird die Berechnung durchgeführt und das Schieberegister einmal nach links durchgeschoben. Sollte eine logische 0 im MSB-Element stehen, erfolgt nur eine linke Schiebeoperation der rechts einführenden Daten.

Diese serielle Arbeitsweise wurde in VHDL mit einer IF-Anweisung implementiert. Diese überprüft das MSB Element des Schieberegisters auf eine führende logische/binäre 1. In diesem Fall werden die XOR-Verknüpfung mit dem Generatorpolynom, die Schiebeoperation und das Einführen des nächsten Datenbits durchgeführt. Trifft der Fall nicht zu, erfolgt eine Schiebeoperation um ein Bit. Bei 16-bit-Daten und einer seriellen Implementierung, dauert die Berechnung des CRC 16 Takte. Mit einer Frequenz von 50 MHz benötigt das CRC-Modul eine Berechnungszeit von 320 ns bis ein Ergebnis an den LEDs sichtbar wird und eine weitere Berechnung erfolgen kann.

D. Ressourcenverbrauch, Kenndaten und Performanceabschätzungen

Die Implementierung des NEO430 auf einen FPGA verbraucht dessen Logikelemente und embedded Speicher. In dieser Arbeit wurde für die Implementierung das Development Board DE2-115 von Terasic genutzt [9]. Auf diesem befindet sich ein Cyclone IV FPGA von Intel (Altera) und bietet etwa 115 k Logikelemente (LE) an, sowie integrierte Speicherblöcke die 9 kbit umfassen. Ein LE besteht aus einem programmierbaren Register und einer Look-Up Table (LUT) mit vier Eingängen, welche jede mögliche Kombination aus vier Variablen umsetzen kann [10].

Tabelle II
ERMITTLUNG DES LOGIK- UND SPEICHER- RESSOURCEN-VERBRAUCHS AUF EINEM CYCLONE IV-FPGA UND VERGLEICH MIT EINER REFERENZIMPLEMENTIERUNG.

Getestete Konfiguration	**Verbrauch an Logik-elementen**	**Verbrauch an Speicher**
Minimale	676	9,5 kB
Default	1094	9,2 kB
Default, max. Speicher	1114	68,6 kB
Default, ohne Wishbone	1051	9,2 kB
Default, mit CRC-Modul	1253	9,2 kB
Referenzwert aus [8]	1188	9,8 kB

Als Referenzwert für den Ressourcenverbrauch dienen die 1188 LE aus der Dokumentation des NEO430 [8]. Die Implementierung erfolgte ebenfalls auf einem Cyclone IV, jedoch mit nur 22 k Logikelementen. Bei diesem Verbrauch wurde der NEO430 in seinen default-Einstellungen integriert, d.h. die Referenzimplementierung umfasst jedes Modul aus Abbildung 1, sowie 4 kB Programm-, 2 kB Datenspeicher und 2 kB Speicher des Bootloaders. Mit den gleichen Einstellungen wurde der NEO430 auf dem Cyclone IV des DE2-115-Development-Boards implementiert. Zur Synthese wurde das in Quartus Prime Version 17.0 integrierte Synthese-Werkzeug verwendet. Die Ergebnisse der eigenen Implementierung sind in Tabelle II, in der Default-Konfiguration dargestellt.

Es wurden im Gegensatz zur Referenzimplementierung 1094 LE und 9,2 kB Speicher verbraucht. In Tabelle II ist zu dem der Ressourcenverbrauch weiterer getesteter Konfigurationen auf dem Cyclone IV des DE2-115-Development-Boards dargestellt. Es wurden folgende weitere Konfiguration des NEO430 getestet:

- Minimale Konfiguration; implementiert nur die CPU, sowie 4 kB Programm- und 2 kB Datenspeicher. Das CPU-Modul des NEO430 verbraucht 676 LE
- Default Konfiguration mit maximalem Speicher; hierfür wurde die maximal adressierbare Speicherkapazität von 64 kB des NEO430 getestet. Verbraucht werden 68,6 kB integrierter Speicher des FPGAs, sowie 1114 LE.
- Default Konfiguration ohne Wishbone-Interface; bei einem Verbrauch von 1051 LE und einem Differenzvergleich mit der default Konfiguration, verbraucht das Wishbone Interface in etwa 40 LE.
- Default Konfiguration mit CRC-Modul; die Beispielapplikation mit dem CRC-Modul wird über Wishbone mit dem NEO430 in seinen Standard-Einstellungen verbunden und verbraucht hierfür 1253 LE, von denen 160 LE das CRC-Modul mit seinem Wishbone Slave-Interface beansprucht. Der Verbrauch von 160 LE im Vergleich mit der Minimalen Konfiguration von 676 LE ist relativ gesehen sehr viel für ein CRC-Modul.

Der implementierte NEO430 arbeitet mit einer Frequenz von 50 MHz die das DE2-115 Development-Board zur Verfügung stellt. Aus der Dokumentation des NEO430 geht hervor, dass dieser auf der Altera-Cyclone IV-Plattform mit etwa 116 MHz, betrieben werden kann [8]. Bei in dieser Arbeit verwendetem FPGA ergab der Synthetisierungbericht in der Default-Konfiguration mit CRC-Modul eine maximale Frequenz von etwa 104 MHz.

V. FAZIT

Im Rahmen dieses Beitrags wurden die Resultate der Untersuchung von möglichen Softcore-Prozessoren für die Integration energieautarker drahtloser Sensor-SoC-Systeme vorgestellt. Zunächst wurde der Stand der Technik verfügbarer Softcore-Prozessoren vorgestellt. Danach wurden die Resultate aus der Implementierung eines MSP340-kompatiblen Softcore-Prozessors, dem NEO430, auf einem Cyclon-IV-FPGA aufgezeigt.

Durch die starke Verbreitung der MSP430-Serie in der Industrie, ist der NEO430 ein interessanter Kandidat für Low-Power-Sensor-SoC-Systeme. Es hat sich gezeigt, dass die wesentlichen Anforderungen an einen Softcore-Prozessor für die Integration in Low-Power-Sensor-SoC-Systeme mit dem NEO430 erfüllt werden:

- Der NEO430 verfügt über eine schlanke 16-bit-Architektur mit optionalen Komponenten, der einen minimalen Logikverbrauch zur Folge hat.
- Über eine standardisierte Wishbone-Schnittstelle können weitere Komponenten einfach hinzugefügt werden.
- Der NEO430 ist plattformunabhängig und in VHDL verfügbar, d.h. einer Implementierung auf eine Zieltechnologie, z.B. CMOS-65-nm, ist möglich.
- Durch das Lizenzmodell des NEO430 kann dieser frei in eigenen Projekten eingesetzt werden.
- Es handelt sich beim NEO430 um einen gut dokumentieren Softcore, mit dem es möglich ist, innerhalb weniger Tage einen lauffähigen Prozessor zu implementieren. Der größere Zeitaufwand entsteht bei der Implementierung zusätzlicher eigener, über Wishbone anzuschließender Module.

Einen gewissen Nachteil bringt die Programmierung mit sich. Der NEO430 kann per Eclipse und unter Verwendung des MSP430-Intsructions über UART programmiert werden. Nicht möglich sind derzeit die Verwendung von Standard-Texas-Instruments-Programmier-Tools und das On-Chip-Debuggen der C-Programme. Nichts desto trotz ist der NEO430-Softcore-Prozessor sehr interessant für eine Implementierung in Low-Power-Sensor-SoC-Systeme. Die Erweiterung des vorhandenen Systems durch eine entsprechende Debugg-Einheit wird in Erwägung gezogen.

LITERATURVERZEICHNIS

[1] Hrsg. Ministerium für Finanzen und Wirtschaft Baden-Württemberg, Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA: *Strukturstudie „Industrie 4.0 für Baden-Württemberg“*, 2014.

[2] Positionspapier der AG Silicon Saxony: *„Bedeutung der Mikroelektronik für Industrie 4.0.“* URL: https://www.siliconsaxony.de/fileadmin/user_upload/Dokumente__pdf__xls__etc._/Positionspapier-Industrie4.0.pdf – Letzer Zugriff 22.07.17

[3] Bhattacharyya, M.; Dusch, B.; Jansen, D.; Mackensen, E.: *Design and Verification of a Mixed-Signal SoC for Biomedical Applications.* In: Proceeding of the 54. MPC-Workshop. Ulm, Juli 2015. Pages 43-38. ISSN 1868-9221.

[4] Mackensen, E.; Kuntz, W.; Müller, Cl.: *Enhancing the Lifetime of Autonomous Microsystems in Wireless Sensor Actuator Networks (WSANs).* In: Proceedings of the EUROSENSORS XIX Conference. Barcelona, 2005.

[5] Jansen D., Fawaz N., Bau D., Durrenberger M.: *A Small High Performance Microprocessor Core SIRIUS for Embedded Low Power Designs; Demonstrated in a Medical Mass Application of an Electronic Pill.* In: Embedded System Design: Topics, Techniques and Trends, Pages. 363-372, Springer, ISBN 987-0-387-72257-3.

[6] Jia, R.; Yu Lin, C.; Guo, Z.; Chen, R.; Wang, F.; Gao, T.; Yang, H.: *A Survey of Open Source Processors for FPGAs. Published in: Field Programmable Logic and Applications (FPL)*, 24th International Conference, 2014, INSPEC Accession Number: 14692827

[7] OpenCores (Hrsg): *Wishbone B4 WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores.* URL: https://opencores.org/cdn/downloads/wbspec_b4.pdf - Letzter Zugriff 08.08.2017.

[8] Nolting, S.: *The NEO430 Processor.* URL: https://github.com/stnolting/neo430/blob/master/doc/NEO430.pdf - Letzter Zugriff 06.08.2017

[9] Terasic (Hrsg.): *DE2-115 User Manual*. URL: https://www.terasic.com.tw/cgi-bin/page/achive_download.pl?Language=English&No=502&FID=cd9c7c1feaa2467c58c9aa4cc02131af - Letzter Zugriff 08.08.2017

[10] Altera (Hrsg.): *Cyclone IV Device Handbook, Volume 1.* URL: https://www.altera.com/content/dam/alterawww/global/en_US/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf - Letzter Zugriff 08.08.2017

[11] Hoffmann, D. W.: *Eine Einführung in die Informations- und Codierungstheorie.* Springer Vieweg, 2014. ISBN 978-3- 642-54002-8.

Artur Werner studiert Elektrotechnik / Informationstechnik mit dem Schwerpunkt Kommunikationstechnik im 7. Semester an der Hochschule Offenburg und arbeitet derzeit an seiner Abschlussarbeit.

Patrick Moser studierte Elektrotechnik / Informationstechnik mit dem Schwerpunkt Kommunikationstechnik an der Hochschule Offenburg und schloss sein Studium 2017 mit dem akademischen Grad eines Bachelor of Engineering ab. Seit dem Sommersemester 2017 absolviert er ein Master-Studium und ist akademischer Mitarbeiter im Mikroelektronik Labor an der Hochschule Offenburg.

Elke Mackensen studierte Nachrichtentechnik an der Fachhochschule Offenburg. Nach ihrem Diplomabschluss 1993 befasste sie sich mit dem Aufbau einer CAE-Abteilung bei der Firma Hekatron. Nach dreijähriger Industrietätigkeit begann sie ihre wissenschaftliche Laufbahn an der Universität in Freiburg. Zunächst am Institut für Informatik wo sie eine ASIC-Design-Labor aufbaute, danach dann am Institut für Mikrosystemtechnik (IMTEK), wo sie 2006 über autarke drahtlose Mikrosysteme promovierte. Von 2006 bis 2010 leitete sie bei der Firma New-Tec die Entwicklungsgruppe Smart Embedded Systems. Seit 2010 ist sie Professorin an der Hochschule in Offenburg. Ihre Professur beschäftigt sich mit der digitalen Schaltungstechnik und dem mikroelektronischem Systementwurf. Weitere Forschungsschwerpunkt sind drahtlose und auf Energy-harvesting-basierte hochminiaturisierte Sensorsyteme.

Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Implementierung einer speichereffizienten Huffman-Decodierung

Malek Safieh, Daniel Rohweder, Jürgen Freudenberger

***Zusammenfassung*—In diesem Beitrag wird die Hardware-Implementierung eines Datenkompressionsverfahrens auf einem FPGA vorgestellt. Das Verfahren wurde speziell für Kompression kurzer Datenblöcke in Flash-Speichern entwickelt. Dabei werden Quelldaten mithilfe eines Encoders komprimiert und mit einem Decoder verlustlos dekomprimiert. Durch die Reduktion der Datenrate kann in Flash-Speichern die Übertragungsdauer zum Lesen und Schreiben reduziert werden. Ebenso ist eine Kompression von Nutzdaten sinnvoll, um zusätzliche Redundanzen für einen Fehlerschutz einfügen zu können, ohne den Gesamtspeicherplatzbedarf zu erhöhen.**

***Schlüsselwörter*—Flash-Speicher, Datenkompression, Fehlerkorrektur, Huffman Codierung, Recency-Rank-Calculator.**

I. Einleitung

Die Bedeutung von Flash-Speichern als nichtflüchtiger Massenspeicher nimmt stetig zu. In einem Flash-Speicher werden Informationen in Floating-Gates gespeichert, die geladen und gelöscht werden können. Diese Floating-Gates halten ihre elektrische Ladung ohne Energieversorgung. Allerdings können Informationen beim Auslesen fehlerhaft gelesen werden. Die Fehlerwahrscheinlichkeit der Speicherzellen hängt von der Speicherdichte und der verwendeten Flash-Technologie (Single-Level Cell (SLC), Multi-Level Cell oder Triple-Level Cell (TLC)) ab. Die Fehlerwahrscheinlichkeit steigt im Laufe der Nutzung, daher wird die Lebensdauer eines Flash-Speichers durch die Anzahl der erzielbaren Schreib- und Löschzyklen angegeben [1]. Um eine zuverlässige Informationsspeicherung zu gewährleisten, sind Verfahren zur Fehlerkorrektur (Error Correction Coding, ECC) erforderlich. In heutigen Systemen werden häufig Bose-Chaudhuri-Hocquenghem (BCH) Codes (Vgl. [2]) für die Fehlerkorrektur verwendet [3], [4]. Außerdem wurden verkettete Codierungsschemata vorgeschlagen [5]–[10].

Datenkompression findet bislang noch wenig Anwendung für Flash-Speicher. Die Datenkompression kann aber ein wichtiger Bestandteil eines Speichersystems sein, der die Systemzuverlässigkeit verbessert. Datenkompression reduziert die Datenmenge, die vom Host-System übertragen wird, und kann zum Beispiel die Write-Amplification reduzieren [16]. Write-Amplification ist ein unerwünschtes Phänomen in Flash-Speichern und bezieht sich auf die Tatsache, dass die Menge der im Flash-Speicher geschriebenen Daten in der Regel ein Vielfaches der Datenmenge des Host-Systems ist. Ein Flash-Speicher muss vor dem Schreibvorgang gelöscht werden, wobei die Granularität der Löschoperation typischerweise viel kleiner als die der Schreiboperation ist. Daher führt der Löschvorgang häufig zu einem Kopieren von Benutzerdaten in andere Speicherzellen. Diese Write-Amplification verkürzt die Lebensdauer von Flash-Speichern.

Alternativ kann die Datenkomprimierung verwendet werden, um die Fehlerkorrekturfähigkeit der ECC-Einheit zu steigern. Die Reduktion der Datenmenge ermöglicht es, mehr Paritätsbits zur Fehlerkorrektur zu speichern [11], [12]. In dieser Arbeit beschreiben wir ein Datenkompressionsverfahren, das für Flash-Speicher angewendet werden kann. Das Ziel der Datenkompression ist es, die Benutzerdaten so zu reduzieren, dass die Redundanz für die Fehlerkorrekturcodierung erhöht werden kann. Diese zusätzliche Redundanz verbessert die Zuverlässigkeit des Datenspeichersystems.

Die Steuereinheit (Flash-Controller) für einen Flash-Speicher arbeitet auf einer Blockebene mit typischen Blockgrößen von 512 Byte bis 4 Kilobyte, wobei die einzelnen Blöcke unabhängig voneinander gelesen werden können. Daher muss die Datenkompression kleine Datenblöcke komprimieren, dabei muss sie verlustfrei arbeiten, d.h. die Daten müssen fehlerfrei decodierbar sein. Weiterhin muss die Datenkompression einen hohen Datendurchsatz erzielen, wobei der Einsatz in Flash-Controllern nur eine moderate Komplexität erlaubt. Um diesen Anforderungen an den Kompressionsalgorithmus Rechnung zu tragen, schlagen wir ein sehr einfaches Kompressionsverfahren vor [12].

Der Codec kombiniert einen Move-to-Front-Algorithmus [13, 17] mit dem Huffman-Entropiecodierungsverfahren, dem vermutlich bekanntesten Algorithmus für die verlustlose Datenkompression. Die Entropiecodierung mit dem Huffman-Verfahren wird erreicht, indem verschiedene Codewörter mit variabler Länge den empfangenen Symbolen zugeordnet werden. Die Komprimierung erfolgt dabei, indem die häufig empfangenen Symbole vom Encoder auf kurze Codewörter abgebildet

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 D. Rohweder, drohwede@htwg-konstanz.de sind Mitarbeiter des Instituts für Systemdynamik an der Hochschule Konstanz, Alfred-Wachtel-Str. 8, 78462 Konstanz.

werden. Der Move-to-Front-Algorithmus transformiert die unbekannte Wahrscheinlichkeitsverteilung der Quellensymbole so, dass die Verteilung am Ausgang gut durch eine fixe Verteilung angenähert werden kann. Das Huffman-Verfahren muss daher nicht adaptiv implementiert werden, sondern kommt mit einem einzigen Wörterbuch aus.

Eine Hardware-Architektur für den Encoder, insbesondere für den Move-to-Front-Algorithmus, wurde in [14] vorgestellt. In diesem Beitrag wird die Implementierung eines effizienten Verfahrens für die Huffman- Decodierung vorgestellt. Das Verfahren beruht auf dem Konzept eines partitionierten Codebaums [15]. Der Codebaum des Huffman-Codes wird dabei in Abschnitte aufgeteilt, die jeweils effizient durch Lookup-Tables repräsentiert werden können. Durch eine geeignete Partitionierung kann der Speicheraufwand für die Repräsentation der Teilwörterbücher klein gehalten werden. Allerdings sind dann mehrere Lesevorgänge zur Decodierung eines Codewortes erforderlich. In der vorgestellten Implementierung wurde eine Partitionierung gewählt, die eine schnelle Decodierung aller Codeworte bis zu einer Länge von acht Bit ermöglicht. Diese Länge deckt alle Codeworte mit einer Auftrittswahrscheinlichkeit größer 0,001 ab. Entsprechende Codeworte können innerhalb eines Takts decodiert werden. Für längere Codewörter sind weitere Lookups erforderlich. Aufgrund der geringen Auftrittswahrscheinlichkeit beeinflussen diese zusätzlichen Zyklen den erzielbaren Datendurchsatz kaum. Der vorgestellte Decoder ist sehr speichereffizient und erzielt trotzdem einen hohen Datendurchsatz.

II. Beschreibung des Kompressionsverfahrens

Eine Datenkompression mithilfe des Verfahrens nach Huffman [16] wird erreicht, indem eine feste Anzahl an Quellensymbolen auf Huffman-Codewörter variabler Länge abgebildet wird. Dabei ist die Auftrittswahrscheinlichkeit der Quellensymbole von entscheidender Bedeutung für die Datenkompressionsrate. Ein adaptives Datenkompressionsverfahren hat die Aufgabe, die Auftrittswahrscheinlichkeit der Quellensymbole zu ermitteln und eine passende Abbildung durchzuführen, sodass häufig auftretende Quellensymbole auf kurze Huffman-Codewörter und selten auftretende Quellensymbole auf längere Huffman-Codewörter abgebildet werden.

Um eine adaptive Zuordnung der Quellsymbole auf Huffman-Codewörter anhand ihrer Auftrittswahrscheinlichkeit durchzuführen, wird vor dem Huffman-Encoder der sog. Move-to-Front-Algorithmus (MTF) angewendet. Entsprechend wird nach der Huffman-Decodierung ebenfalls eine Rückabbildung durchgeführt.

Abbildung 1 zeigt eine Reihenschaltung aus MTF, Huffman- und ECC-Encoder, der die nach der Kompression zur Verfügung stehende Übertragungskapazität für den Fehlerschutz nutzt.

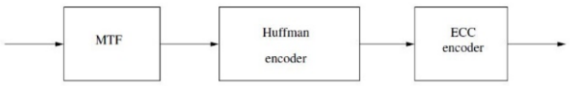


Abbildung 1. Reihenschaltung von MTF, Huffman-Encoder und einer Fehlerschutzeinheit.

A. Beschreibung des MTF-Algorithmus

Beim MTF-Algorithmus handelt es sich um einen selbst organisierenden Listenalgorithmus. Die Grundidee ist es, die Liste so zu verändern, dass Symbole mit hoher Auftrittswahrscheinlichkeit an den Anfang der Liste gestellt werden.

MTF-Algorithmus:

1. Ein Wörterbuch mit allen vorkommenden Zeichen wird generiert.
2. Vom Anfang des Blocks beginnend:
   a) Suche das aktuelle Symbol im Wörterbuch und gebe den aktuellen Index aus.
   b) Schiebe das Symbol im Wörterbuch an die erste Stelle.

Der Algorithmus bewirkt eine Transformation der Auftrittswahrscheinlichkeitsfunktion. Folgende Abbildung zeigt die Transformation der Auftrittswahrscheinlichkeitsverteilung einer Textdatei durch Anwendung des MTF-Algorithmus. Die Ausgangsverteilung kann durch eine angepasste Log-Normalverteilung (rote Kurve in Abbildung 2) approximiert werden. Die Textdatei besteht aus ASCII-codierten Zeichen mit acht Bit je Zeichen. Das Codealphabet besteht aus insgesamt 256 Zeichen. Auf Basis der approximierten Verteilung kann ein fixer Huffman-Code verwendet werden, ohne die Codierung für unterschiedliche Daten ändern zu müssen.

Der Move-to-Front-Algorithmus, wie er hier verwendet wird, ist in [14] ausführlich vorgestellt und untersucht worden. Ebenfalls werden verschiedene Versionen des MTF beschrieben und miteinander verglichen.

III. Huffman-Decoder

Im Folgenden wird nun der Huffman-Decoder beschrieben.

A. Aufbau des Huffman-Decoders

Bei der Decodierung werden pro Systemtakt acht Bit vom Speichermedium adressiert und gelesen, die aus hintereinander geschriebenen Huffman-Codewörtern bestehen. Wegen der variablen Länge muss Anfang und Ende eines jeden Codewortes erkannt und die

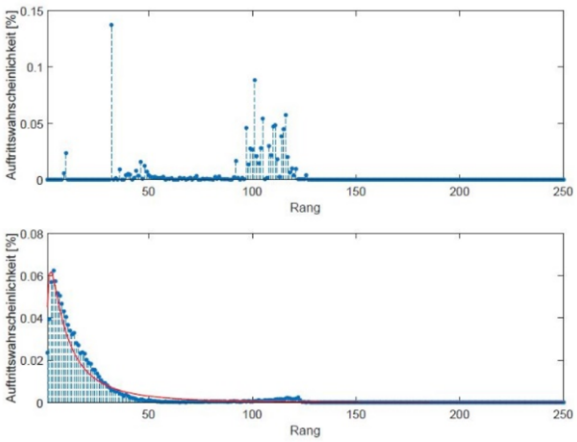


Abbildung 2. Auftrittswahrscheinlichkeit vor (obere Verteilung) und nach (untere Verteilung) Anwendung des MTF-Algorithmus.

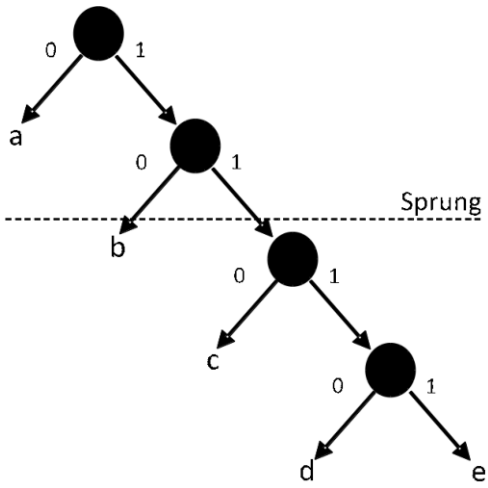


Abbildung 3. Huffman-Codebaum.

Codeworte so getrennt werden. Dann kann die Rückabbildung auf Symbole erfolgen. In der Regel erfolgt die Decodierung durch bitweises Überprüfen der gelesenen Folge anhand des Codebaums (siehe Abbildung 3), bis ein vollständiges Codewort im Codebaum gelesen bzw. decodiert wird. Eine solche Decodierung benötigt so viele Systemtakte wie die Bitanzahl eines Codewortes und ist daher für die Anwendung in Speichermedien ungeeignet. Der vorgestellte Huffman-Decoder bietet dagegen die Möglichkeit, nicht nur die Erkennung sondern auch eine verlustlose Decodierung eines Codewortes in den meisten Fällen innerhalb eines Systemtaktes zu erreichen. Dabei wird die präfixfreie Codierung sowie die Verwendung des effizienten Lookup-Tabelle-Verfahrens aus [15] ausgenutzt.

Außerdem beinhaltet der Huffman-Decoder auf Grund der variablen Codewortlänge einen Puffer, der die innerhalb eines Systemtaktes für die Decodierung nicht verwendeten Bits einer Bitfolge für die weitere Decodierung im nächsten Systemtakt zwischenspeichert.

Tabelle I
AUFBAU EINER LUT ZELLE.

Eine Zelle		
Feld 1 → 1 Bit	Feld 2→ 4 Bit	Feld 3 → 10 Bit

B. Lookup-Tabelle (LUT)

Die verlustlose Decodierung mit dem LUT-Verfahren ist schnell und speichereffizient, da ihre Realisierung für alle 256 Codeworte mit insgesamt 664 eindeutig adressierten Speicherzellen der Breite 15 Bit vollständig umgesetzt werden kann. Mit jedem Systemtakt wird genau eine Zelle der LUT durch die gelesenen acht Bit der Codewortfolge adressiert. Die 15 Bit eines Tabelleneintrags können in drei Felder unterteilt werden (siehe Tabelle I):

- Das erste Feld der Zelle besteht aus einem einzigen Bit. Falls dieses Bit den Wert „0“ annimmt, ist das gesuchte Codewort vollständig in den gelesenen acht Bit enthalten. Andernfalls wurde das entsprechende Codewort noch nicht vollständig gelesen.
- Das zweite Feld liefert die Länge des gefundenen Codewortes, die auf Grund der präfixfreien Generierung immer erkennbar ist.
- Im dritten Feld steht das gefundene Codewort, falls der Inhalt des ersten Feldes „0“ ist. Sonst ist kein Codewort sondern eine Offsetadresse zum Fortsetzen der Suche zu finden.

Wird eine Offsetadresse ausgewertet, müssen noch drei Bit der nächsten Bitfolge gelesen werden. Diese werden zur Offsetadresse addiert, um die Zellenadresse für ein Codewort zu finden, das länger als acht Bit ist. Somit werden maximal zwei Systemtakte für die Decodierung eines Codewortes benötigt. Codeworte mit einer Länge bis zu acht Bit können in einem Zyklus decodiert werden. Mit dem folgenden Beispiel wird die Funktionsweise des LUT erläutert. Es handelt sich dabei um eine Decodierung mithilfe des Huffman-Codebaums in Abbildung 3. Hierfür werden nur fünf Zeichen betrachtet. Somit bleibt die Länge für das erste Feld ein Bit, während das zweite Feld zwei Bit und das dritte Feld vier Bit enthält.

Beispiel:

Es wird die folgende Codewortfolge decodiert „0101111“, wobei immer zwei Bit je Zyklus gelesen werden.

Im ersten Schritt wird die Bitfolge „01“ gelesen. Die Bitfolge wird durch ein „0“ Bit zur Adresse „001“ ergänzt. Mit dieser Adresse wird die zweite Zelle der Lookup-Tabelle (siehe Tabelle II) adressiert.

Das erste Feld dieser Zelle ist „0“ . In diesem Fall wurde ein vollständiges Codewort gelesen. Die Länge

Tabelle II
BEISPIEL LOOKUP-TABELLE.

	Zellenfelder			
	1 Bit	2 Bit	4 Bit	
Adresse	**Sprung**	**Länge**	**Inhalt**	**Symbol**
000	0	01	0000	a
001	0	01	0000	a
010	0	10	0010	b
011	1	10	0100	Offset
Sprung				
100	0	01	0110	c
101	0	01	0110	c
110	0	10	1110	d
111	0	10	1111	e

dieses Codewortes beträgt „01_b“ $= 1_d$ und steht im zweiten Feld. Das erkannte Codewort lautet daher „0_b“ und wird mit dem Symbol „a“ decodiert. Entsprechend kann auch das zweite Codewort „10“ mit nur einem Leseschritt decodiert werden.

Im dritten Schritt wird die Bitfolge „11_b“ gelesen und zur Adressierung verwendet. Der zugehörige Eintrag steht in der vierten Zeile der Lookup-Tabelle. Das erste Feld dieses Eintrags ist „1“. Dies bedeutet, dass noch kein vollständiges Codewort gefunden wurde und deshalb weitergelesen werden muss. Der Inhalt des zweiten Feldes beträgt „10_b“ $= 2_d$ und bezeichnet die Länge bzw. die Anzahl der gefundenen Bits eines gültigen Codewortes. Im dritten Feld steht nun die Offsetadresse „100_b“. Im nächsten Systemtakt werden die nachfolgenden zwei Bit „$11b$“ von der Codewortfolge gelesen und zur Offsetadresse addiert. Das Ergebnis dieser Addition „111_b“ ist die neue Leseadresse. Mit diesem Eintrag wird das Codewort sicher gefunden.

Das erste Feld dieser Zelle ist „0“ . Dies bedeutet, dass ein Codewort gefunden wurde. Im zweiten Feld steht die Länge „10_b“ $= 2_d$. Diese Angabe definiert die zusätzlich gefundenen Bits des Codewortes. In diesem Fall beträgt die Gesamtlänge des erkannten Codewortes $2 + 2 = 4_d$. Der Inhalt des dritten Feldes ist „1111_b“ und wird in das Codewort „e“ decodiert.

Falls für die Decodierung eines Codewortes mit diesem Verfahren nicht genau die gelesenen acht Bit einer Folge benötigt werden, müssen die restlichen nicht verwendeten Bits für die nächste Decodierung zwischengespeichert werden. Hierfür entsteht der Bedarf an einem Puffer, der die Daten in der richtigen Reihenfolge sortiert und zwischenspeichert.

C. Puffer

Für eine effiziente Implementierung wurde die Länge des Puffers auf 64 Bit festgelegt, da die Adressierungslogik dafür verhältnismäßig klein ist. Mit jeder steigenden Flanke des Systemtaktes wird der Puffer mit den nachfolgenden acht Bit vom Speichermedium gefüllt. Zur Initialisierung werden die gepufferten Codeworte erst nach der Speicherung der ersten beiden Bitfolgen decodiert. Da die Anzahl der zur Decodierung verwendeten Bits im zweiten Feld der LUT klar definiert ist, können diese im gleichen Systemtakt aus dem Puffer entfernt werden. Hierfür wird ein Pointer verwendet, der die nächste freie Stelle im Puffer markiert und die richtige Verschiebung des Puffers garantiert.

Da es sich um eine Datenkompression handelt, ist die Anzahl der vom Speichermedium gelesenen Bits im Mittel niedriger als die Anzahl der ausgegebenen Bits. Das heißt, die Mehrheit der im Speichermedium gelagerten Codewörter ist mit weniger als acht Bit beschrieben. Daher füllt sich der Puffer bei fortlaufender Decodierung ständig auf. Reicht die Pufferlänge von 64 Bit nicht aus, um alle anstehenden Symbole zu speichern, dann wird das Lesen für einen Zyklus angehalten, um einen Pufferüberlauf zu verhindern.

Tabelle III
SYNTHESEERGEBNIS.

	LUTs	**Register**	**Slices**
MTF-Encoder	2162	2048	753
Huffman-Encoder	594	107	164
Huffman-Decoder	1002	92	276
MTF-Decoder	1740	2048	2048

IV. ERGEBNISSE

A. Syntheseergebnisse

Das beschriebene Verfahren wurde in Verilog implementiert und erfolgreich für einen Xilinx Virtex7 synthetisiert. Es werden für diese Implementierung 4380 Register, 5331 LUT und 3076 Slices benötigt. Die maximale Taktfrequenz beträgt 60MHz.

Tabelle III zeigt die Verteilung der Ressourcen auf die einzelnen Bereiche. An dieser Stelle ist zu erwähnen, dass eine zusammenhängende Implementierung besser optimiert werden kann und die Summe der einzelnen Funktionsblöcke meist höher ist.

B. Vergleich zum PDLZW-Verfahren

Bei dem PDLZW-Verfahren handelt es sich um einen weiteren bekannten Datenkompressionsalgorithmus [18]. Tabelle IV zeigt einen direkten Vergleich zu dem von uns vorgestellten Verfahren. Sie zeigt die durchschnittliche und die maximale Blocklänge in Bytes nach Anwendung des Algorithmus.

Man erkennt, dass das PDLZW-Verfahren im Mittel eine höhere Kompressionsrate erzielt, die Größe Max. ist der Mittelwert über alle Dateien im Korpus für den jeweils größten Datenblock. Dieser Wert kann als Gütemaß für schlecht komprimierbare Daten interpretiert werden. Hier erzielt das kombinierte Verfahren aus

Tabelle IV
BLOCKLÄNGE IN BYTES NACH DER KOMPRESSION FÜR DEN CALGARY CORPUS.

MTF+Huffman		**PDLZW**	
∅	Max.	∅	Max.
786,4	837,1	720,7	853,6

Tabelle V
SYNTHESEERGEBNIS FÜR DEN PDLZW.

	LUTs	**Register**	**Max. Taktfrequenz**
PDLZW-Encoder	26514	12561	87MHz

MTF und Huffman-Codierung einen deutlich besseren Wert.

Tabelle V zeigt die erforderliche Hardware für die PDLWZ-Implementierung.

Hier benötigt nur der Encoder schon rund das Fünffache an LUT und das Dreifache an Register im Vergleich zum vorgestellten Codec. Die maximale Taktfrequenz ist jedoch für das PDLZW-Verfahren höher.

V. FAZIT

In diesem Beitrag wurde die Hardware-Implementierung eines Datenkompressionsverfahrens vorgestellt. Das Verfahren kombiniert den MTF-Algorithmus mit einer Huffman-Codierung. Für die Kompression kurzer Datenblöcke erzielt dieses Verfahren vergleichbare Kompressionsgewinne wie der PDLZW-Algorithmus, hat jedoch eine deutlich geringere Komplexität.

LITERATURVERZEICHNIS

[1] R. Micheloni, A. Marelli, and R. Ravasio, Error Correction Codes for Non-Volatile Memories, Springer, 2008.

[2] A. Neubauer, J. Freudenberger, and V. Kühn, Coding Theory: Algorithms, Architectures and Applications. John Wiley & Sons, 2007.

[3] W. Liu, J. Rho, and W. Sung, „Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories“, in *IEEE Workshop on Signal Processing Systems Design and Implementation (SIPS)*, Oct. 2006, pp. 303-308.

[4] J. Freudenberger and J. Spinner, „A configurable Bose-Chaudhuri-Hocquenghem codec architecture for flash controller applications“, in *Journal of Circuits, Systems, and Computers*, vol. 23, no. 2, pp. 1-15, Feb. 2014.

[5] C. Yang, Y. Emre, and C. Chakrabarti, „Product code schemes for error correction in MLC NAND flash memories“, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 12, pp. 2302-2314, Dec. 2012.

[6] F. Sun, S. Devarajan, K. Rose, and T. Zhang, „Design of on-chip error correction systems for multilevel NOR and NAND flash memories“, in *IET Circuits, Devices Systems*, vol. 1, no. 3, pp. 241-249, June 2007.

[7] S. Li and T. Zhang, „Improving multi-level NAND flash memory storage reliability using concatenated BCH-TCM coding“, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 10, pp. 1412-1420, Oct 2010.

[8] J. Oh, J. Ha, J. Moon, and G. Ungerboeck, „Rs-enhanced TCM for multilevel flash memories“, in *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1674-1683, May 2013.

[9] J. Spinner, J. Freudenberger, and S. Shavgulidze, „A soft input decoding algorithm for generalized concatenated codes“, in *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3585-3595, Sept. 2016.

[10] J. Spinner, M. Rajab, and J. Freudenberger, „Construction of high-rate generalized concatenated codes for applications in non-volatile flash memories“, in *2016 IEEE 8th International Memory Workshop (IMW)*, May 2016, pp. 1-4.

[11] N. Xie, G. Dong, and T. Zhang, „Using lossless data compression in data storage systems: Not for saving space“, in *IEEE Transactions on Computers*, vol. 60, no. 3, pp. 335-345, March 2011.

[12] J. Freudenberger, A. Beck, and M. Rajab, „A datacompression scheme for reliable data storage in non-volatile memories“, in *IEEE 5th International Conference on Consumer Electronics (ICCE)*, Sept 2015, pp. 139-142.

[13] P. Elias, „Interval and recency rank source coding: Two on-line adaptive variable-length schemes“, in *IEEE Transactions of Information Theory*, vol. 33, no. 1, pp. 3-10, Jan. 1987.

[14] J. Freudenberger, M. Rajab, D. Rohweder und M. Safieh, „A codec architecture for the compression of short data blocks“, to be published in *Journal of Circuits, Systems, and Computers (JCSC)*, Vol. 27, Issue 2, Feb. 2018.

[15] R. Hashemian, „Design and hardware implementation of a memory efficient Huffman decoding“, in *IEEE Transactions on Consumer Electronics*, vol. 40, no. 3, pp. 345-352, Aug 1994.

[16] D. A. Huffman, „A method for the construction of minimum-redundancy codes“, in *Proceedings of the I.R.E.*, pp. 1098-1101, 1952.

[17] J. L. Bentley, D. Sleator, R. Tarjan, and V. Wei, „A locally adaptive data compression scheme“, in *Communications of the ACM*, 1986, 29(4), pp. 320-330.

[18] Ming-Bo Lin, „A Hardware Architecture for the LZW Compression and Decompression Algorithms Based on Parallel Dictionaries“, in *Journal of VLSI signal processing systems for signal, image and video technology*, pp. 369-381, 2000.

Prof. Dr. 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.

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.

Daniel Rohweder 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.

Robert Bosch Zentrum für Leistungselektronik
Hochschule Reutlingen . Universität Stuttgart . Robert Bosch GmbH

Entwurf kontextbasierter PCells für Hochfrequenzanwendungen in modernen CMOS-Technologien

Matthias Thoma, Daniel Marolt, Jürgen Scheible, Gregor Tretter, Göran Jerke

***Zusammenfassung*—Im Vergleich zum digitalen Layoutentwurf weist der analoge Layoutentwurf einen wesentlich geringeren Automatisierungsgrad auf. Dies gilt insbesondere für den Layoutentwurf von Hochfrequenzschaltungen, wo Einflüsse der lokalen Layoutumgebung besonders zu berücksichtigen sind. Bei dieser sog. Kontextabhängigkeit geraten sowohl Optimierungsalgorithmen als auch herkömmliche Generatoransätze schnell an Grenzen. In dieser Arbeit wird eine funktionale Erweiterung des bekannten Generatorprinzips eingesetzt, die es erlaubt, Informationen aus der Layoutumgebung der Instanz in die Layoutgenerierung einzubeziehen. Mit dieser sog. kontextbasierten PCell gelingt die Automatisierung konkreter, bisher nur manuell lösbarer Probleme des Layoutentwurfs von Hochfrequenzschaltungen. Die Arbeit zeigt das Potential kontextbasierter PCells für die weitere Steigerung des Automatisierungsgrades im analogen Layoutentwurf.**

***Schlüsselwörter*—PCell, kontextbasierte PCell, Layoutentwurf, Hochfrequenztechnik.**

I. Einleitung

A. Motivation

Radarsysteme im Frequenzbereich von 77 bis 81 GHz stellen einen essentiellen Bestandteil fortschrittlicher Fahrassistenzsysteme und des autonomen Fahrens dar [1]. Diese Radarsysteme enthalten analoge Schaltungsteile zur Erfassung und Übertragung der Informationen und digitale Schaltungsteile zur Verarbeitung der Informationen. Während in früheren Entwürfen digitale und analoge Schaltungen auf getrennten Technologien hergestellt und in einem Multi-Chip System (System in Package SiP) zusammengeführt wurden, ermöglichen die modernen CMOS-Prozesse die Integration aller Schaltungsteile auf demselben Chip (System on Chip SoC) [1]. Neben den Kostenvorteilen der CMOS-Technologie lassen sich damit deutlich mehr digitale Kontrollfunktionen integrieren.

Matthias Thoma, matthias.thoma@student.reutlingen-university.de, Daniel Marolt, daniel.marolt@reutlingen-university.de, Jürgen Scheible, juergen.scheible@reutlingen-university.de, Reutlingen University, Alteburgstraße 150, 72762 Reutlingen.

Gregor Tretter, gregor.tretter@de.bosch.com, Goeran Jerke, goeran.jerke@de.bosch.com, Robert Bosch GmbH, 72760 Reutlingen.

Diese wiss. Arbeit wurde teilweise mit Mitteln aus den ENIAC und BMBF Förderprojekten THINGS2DO (Förderkennz. 16ES0505) und autoSWIFT (Förderkennz. 16ES03) gefördert.

Der Layoutentwurf der analogen Schaltungen eines solchen SoC-Radarsystems ist dabei erheblich aufwendiger als derjenige der digitalen Schaltungen. Während in der Digitaltechnik durch die Verwendung von Standardzellen und Optimierungsalgorithmen ein hoher Automatisierungsgrad erzielt wird, zeichnet sich der Layoutentwurf analoger Hochfrequenz-Schaltungstechnik durch ein hohes Maß an schaltungsspezifischen Anforderungen aus, was sich in einem entsprechend aufwendigen, manuellen Layoutentwurf widerspiegelt.

Komplexität entsteht im digitalen Layoutentwurf in erster Linie durch die hohe Anzahl der zu integrierenden Bauelemente. Diese aus der Quantität der Bauteile erwachsende Komplexität ist auch als „*More Moore*“ Problematik bekannt [2]. Im Layoutentwurf der analogen Schaltungstechnik ist die Bauteilanzahl um ein Vielfaches geringer. Jedoch sind die funktionalen Anforderungen an die Schaltung wesentlich vielfältiger und schwieriger umzusetzen als im Bereich der Digitaltechnik. Da schon geringe Abweichungen der elektrischen Eigenschaften zu Fehlfunktionen führen können, erfordert der Layoutentwurf ein hohes Maß an Fachwissen und Sorgfalt. Diese aus der geforderten Qualität erwachsende Komplexität wird auch „*More than Moore*“-Problematik genannt [2].

In der Hochfrequenztechnik ist das Ausmaß der Komplexität gegenüber niederfrequenten Analogschaltungen nochmal gesteigert. Dies wirkt sich auch im Layoutentwurf aus. Hierbei müssen die parasitären Einflüsse, welche die im Layout platzierten Struktur und die in ihrer lokalen Umgebung (also in ihrem *Kontext*) vorhandenen Objekte aufeinander ausüben, berücksichtigt werden. Aufgrund dieser extremen „More than Moore“-Problematik ist der Layoutentwurf momentan nur mittels manueller Platzierung der Geometrien durchExperten der Hochfrequenztechnik möglich.

B. Problembeschreibung

Der Layoutentwurf ist die Transformation einer in symbolischer Form gegebenen Funktionsbeschreibung eines ICs in Form eines Schaltplans in eine zur Fertigung genutzte Beschreibung in Form der Maskendaten des ICs [3]. Diese Transformation erfordert die

Einhaltung von Randbedingungen. Daneben wird die Erreichung von Optimierungszielen angestrebt.

Man unterteilt die im Layoutentwurf einzuhaltenden Randbedingungen in drei Kategorien. Es gibt technologische, funktionale und entwurfsmethodische Randbedingungen [3].

- Technologische Randbedingungen haben die Aufgabe, die Herstellbarkeit des ICs in dem verwendeten Halbleiterprozess sicherzustellen. Sie hängen daher von der zur Herstellung genutzten Technologie ab. Aus ihnen werden die unter dem Begriff *Designregeln* geführten geometrischen Entwurfsregeln abgeleitet.
- Funktionale Randbedingungen lassen sich in schaltungsspezifische Anforderungen und prozessspezifische Anforderungen unterteilen. Die funktionalen Randbedingungen gewährleisten die Einhaltung des angestrebten elektrischen Verhaltens und der geforderten Zuverlässigkeit eines Schaltkreises. Aus diesem Grund nennt man sie auch elektrische Randbedingungen. Aus den funktionalen Randbedingungen werden die sogenannten *elektrischen Entwurfsregeln* und *Constraints* formuliert. Die Constraints werden im Entwurfsablauf definiert und müssen im Entwurfsablauf umgesetzt werden. Zu Constraints zählen beispielsweise Symmetrievorgaben oder Vorgaben für Leiterbahnführungen. Die prozessspezifischen funktionalen Randbedingungen sind hingegen für eine gewählte Technologie allgemein gültig und daher in elektrischen Entwurfsregeln darstellbar.
- Entwurfsmethodische Randbedingungen werden eingeführt, um den Schwierigkeitsgrad des Layoutentwurfs abzumildern. Sie dienen dazu, das Layout einer rechnergestützten Bearbeitung zugänglich zu machen. Dabei stellen sie eine künstliche Einschränkung der vorhandenen Freiheitsgrade dar. Beispielsweise werden damit zusätzliche Regeln definiert, die in der Digitaltechnik den Einsatz von Standardzellen erlauben.

Die Hochfrequenztechnik zeichnet sich dabei durch das besondere Zusammenspiel dieser Randbedingungen aus. Schon geringste parasitäre Kapazitäten oder Induktivität können zu Störungen der Schaltungsfunktionen führen. Aus diesem Grund müssen sensible Schaltungsbereiche mittels einer Simulation ihrer *elektromagnetischen Eigenschaften* (EM-Simulation) analysiert werden. Damit wird untersucht, inwieweit ein bestimmter Schaltungsteil durch Geometrien in dessen Kontext parasitär beeinflusst wird. Da solche Simulationen hohen Rechenaufwand benötigen, können sie nur in kleinen Bereichen und mit auf einem festen Raster liegenden Geometrien effizient durchgeführt werden.

Dies kann an der Platzierung von *Dummy-Metall-Geometrien* beispielhaft verdeutlicht werden:

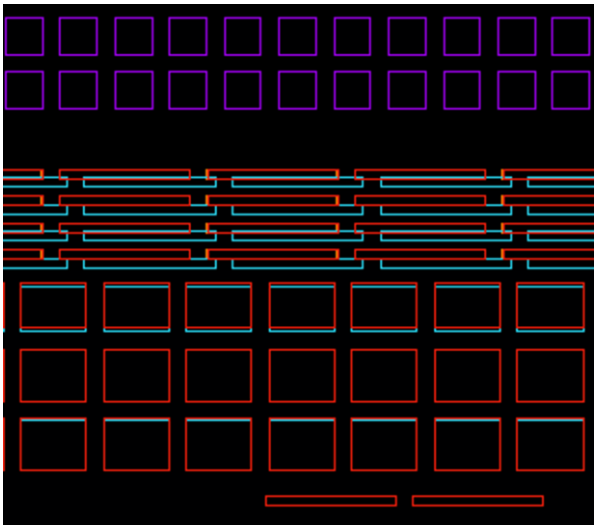


Abbildung 1. Beispiel von Dummy-Metall-Geometrien, welche mit einem Standard Füller Tool erzeugt wurden.

Dummy-Metall-Geometrien sind Strukturen, die keine elektrische Funktion in der Schaltung haben, sondern lediglich im physikalischen Layout platziert werden, um eine durch die Technologie vorgegebene Metalldichte zu erreichen. Diese Dichte wird in Metall pro Flächeneinheit gemessen und ist nötig, um bei Planarisierungsschritten in der Fertigung des ICs eine möglichst gleichmäßige Oberfläche zu erhalten.

Da Dummy-Metall-Geometrien keinerlei funktionale Aufgaben erfüllen, werden diese gewöhnlich im Anschluss an den Layoutentwurf mittels polygonbasierter Füller-Tools, die vom Technologiegeber bereitgestellt werden, automatisch generiert und platziert. Da hierbei nur technologische Randbedingungen berücksichtigt werden, hat der Anwender keine Möglichkeit, die Position und Form dieser Strukturen an entwurfsmethodische Randbedingungen, die sich aus der EM-Simulation ergeben, anzupassen.

Abbildung 1 zeigt beispielhaft die Geometrie, welche durch ein solches polygonbasiertes Standard Füller-Tool erstellt wurden. Wie zu erkennen ist, liegen unterschiedliche Größen und Formen vor. Auch ist die Position der Strukturen nicht einheitlich. Der in Abbildung 1 dargestellte Bereich lässt sich durch die unregelmäßige Position und Größe der Dummy-Geometrien nicht effizient mittels einer EM-Simulation untersuchen.

Da Füller-Tools keine für die EM-Simulation geeigneten Geometrien erstellen, werden nach dem Stand der Technik in den sensiblen Bereichen Dummy-Geometrien manuell durch Experten der Hochfrequenztechnik platziert. Dabei müssen die Experten nicht nur auf eine bezüglich EM-Simulation geeignete Rasterung achten (entwurfsmethodische Randbedingung), sondern auch Dichte- und Abstandsregeln hinsichtlich Herstellbarkeit einhalten (technologische Randbedingungen) und zusätzlich noch den parasitären

Einfluss der Dummy-Metall-Geometrien auf funktionsrelevante Schaltungsteile berücksichtigen (funktionale Randbedingungen).

Dieses Beispiel verdeutlicht die zwei zentralen Herausforderungen der vorliegenden Arbeit: (1) die übergreifende Erfüllung von Randbedingungen aller drei Kategorien, und (2) die Einbeziehung des Kontexts eines Schaltungsteils in den Layoutentwurf.

C. Beitrag dieser Arbeit

1. Im nachfolgenden Abschnitt II wird darauf eingegangen, warum die bestehenden Automatisierungsansätze im Layoutentwurf nicht für die Probleme des Layoutentwurfs hochfrequenter Schaltungen geeignet sind. Dazu werden die bestehenden Automatisierungsstrategien (Optimierungsalgorithmen und prozedurale Generatoren) bezüglich ihrer Eignung untersucht.
2. In Abschnitt III wird ein neuer Ansatz eines prozeduralen Generators, die sogenannte *kontextbasierte PCell*, vorgestellt. Dabei wird die bestehende Implementierungsmöglichkeit kontextbasierter PCells in der CAD-Software Cadence Virtuoso diskutiert.
3. Abschnitt IV stellt die Entwicklung einer kontextbasierten PCell vor, welche die automatische Erstellung von Dummy-Metall-Geometrien ermöglicht.
4. In Abschnitt V werden die Ergebnisse der Arbeit diskutiert.
5. Abschnitt VI schließt mit einer Zusammenfassung und einem Ausblick.

II. Stand der Technik

Top-Down vs. Bottom-up

Die bekannten Verfahren zur Automatisierung lassen sich grundsätzlich unterscheiden in Ansätze, die auf Optimierungsalgorithmen basieren und prozedurale Ansätze [2]. Optimierungsalgorithmen nutzen Suchverfahren zur Lösungsfindung in einer mathematisch abstrahierten Problembeschreibung. Prozedurale Verfahren, zu denen die Layoutgeneratoren zählen, reproduzieren die auf Erfahrung eines Entwurfsexperten basierende Vorgehensweise. Die beiden Ansätze entsprechen zwei grundsätzlich unterschiedlichen Automatisierungsparadigmen. In Anlehnung an die bei den beiden Ansätzen jeweils eingenommenen hierarchischen Entwurfsperspektiven lassen sich gem. [2] die optimierungsbasierten Verfahren einem „*Top-Down Paradigma*“ und die prozeduralen Verfahren einem „*Bottom-Up Paradigma*“ zuordnen. Die prinzipiellen Grundstrukturen der beiden Ansätze werden in Abbildung 2 gegenübergestellt.

Optimierungsalgorithmen sind charakterisiert durch ein meist iteratives Vorgehen, in dem fortlaufend

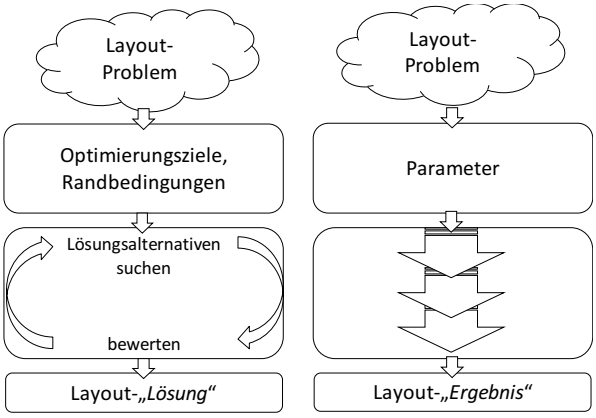


Abbildung 2. Automatisierungs-Paradigmen nach [2].

Lösungen generiert und bewertet werden. Das Problem des Layoutentwurfs wird dabei als komplexes Optimierungsproblem mit verschiedenen Randbedingungen und Optimierungszielen aufgefasst [3]. Alle relevanten Randbedingungen und Optimierungsziele müssen bei der Verwendung eines Algorithmus mathematisch formalisiert und explizit formuliert werden.

Aus diesem Grund eigenen sich Optimierungsalgorithmen in erster Linie für die quantitativ zwar anspruchsvollen, dafür aber gut formalisierbaren „More Moore“ Probleme, wie sie in der Digitaltechnik vorliegen. Da im analogen Layoutentwurf deutlich mehr verschiedenartige Randbedingungen vorliegen und diese zum Teil nur unzureichend mathematisch formalisiert werden können, lassen sich Optimierungsalgorithmen für derartige „More than Moore“ Probleme nur bedingt einsetzen [4]. Insbesondere für die vorliegende Problemstellung eignen sich Optimierungsalgorithmen aus diesem Grund nicht.

„More than Moore“ Probleme lassen sich daher mittels des in Abbildung 2 rechts dargestellten „*Bottom-up*“ Paradigmas besser mit Hilfe prozedural arbeitender Generatoren automatisieren. Die Implementierungsform prozeduraler Generatoren in Cadence Virtuoso wird *Parameterized Cell (PCell)* genannt. Der Begriff *PCell* wird daher in der Fachwelt und auch im weiteren Verlauf der Arbeit als Synonym für einen prozeduralen Generator verwendet.

Im Gegensatz zu Optimierungsalgorithmen, die iterativ eine Layout-Lösung anpassen, arbeiten PCells mit einer Prozedur, die ein durch den Experten vordefiniertes Layout-Ergebnis reproduziert. Das Layout-Ergebnis lässt sich dabei in Abhängigkeit von vordefinierten Parametern anpassen. Eine PCell zeichnet sich dadurch aus, dass sie deterministische und wiederholbar eindeutige Ergebnisse liefert. So gibt es für jede Kombination aus Parameterwerten genau ein wohldefiniertes Layout-Ergebnis. Der den PCell-Code erzeugende Layoutexperte implementiert dabei sein Expertenwissen direkt in den Automatismus der PCell. Dadurch ist eine PCell in der Lage, Randbe-

dingungen und Optimierungsziele implizit (d.h. ohne die Notwendigkeit einer Formalisierung derselben) zu berücksichtigen. Aus diesem Grund eignen sich PCells insbesondere für qualitativ komplexe Probleme, wie sie im analogen Layoutentwurf aufgrund der hohen Diversität der Randbedingungen vorliegen [5].

Eine PCell besteht zum einen aus einem Programm, welches das Layout-Ergebnis in Abhängigkeit der Parameter generiert. Zum anderen ist eine PCell ein im Layoutentwurf instanziierbares Objekt. Der Anwender hat lediglich über die Änderung der Parameterwerte der PCell die Möglichkeit, das Layout-Ergebnis der PCell zu verändern.

Die einzige Möglichkeit, Kontextinformationen in einer PCell-Prozedur zu berücksichtigen, besteht über die Parametrisierung der PCell. Hierzu kann zunächst die manuelle Anpassung der PCell-Parameterwerte durch den Nutzer eingesetzt werden. Eine weitere, in jüngster Zeit eingeführte Möglichkeit, Kontextinformationen in die PCell zu überführen, wird von der sogenannten „*Statically Context-Enhanced PCell*“ verwendet. Dabei werden Informationen automatisiert aus dem Kontext formatiert und mittels der verfügbaren PCell-Parameter in die PCell-Prozedur übertragen.

Während der Kontextinformationsbezug damit noch in gewisser Weise möglich ist, ist eine Interaktion der PCell-Prozedur mit dem Kontext unter der Verwendung einer herkömmlichen PCell nicht möglich. Eine herkömmliche PCell kann weder automatisch auf Änderungen im Kontext reagieren, noch selbst automatisch Änderungen an sich oder dem Kontext vornehmen. Der herkömmlichen PCell fehlt damit ein entscheidendes Merkmal, das im Layoutentwurf hochfrequenter Schaltungen benötigt wird: **Die Interaktion mit dem Kontext.** Um PCells also für Problemstellungen im Layoutentwurf hochfrequenter Schaltungen nutzbar zu machen, ist eine Erweiterung des Konzepts von Nöten. Diese Erweiterungen sind Thema des nächsten Kapitels.

III. Kontextbasierte PCells

Eine kontextbasierte PCell ist nicht mehr nur durch die Parameter bestimmt, sondern gleichzeitig durch ihren Kontext. Das bedeutet, dass innerhalb der PCell-Prozedur Kontextinformationen verarbeitet werden können, die eine Interaktion mit dem Kontext ermöglicht. Die PCell kann dadurch sowohl sich selbst in Abhängigkeit des Kontexts anpassen als auch Geometrien im Kontext erstellen, verändern oder entfernen.

Die im Cadence Virtuoso Layout-Tool verfügbare Implementierungsform einer kontextbasierten PCell ist die sogenannte *React After-PCell*.

React After-Implementierungsform

Die React After-PCell stellt eine kontextbasierte Erweiterung des herkömmlichen PCell-Ansatzes dar. Zur Implementierung wurde das Cadence PCell Designer

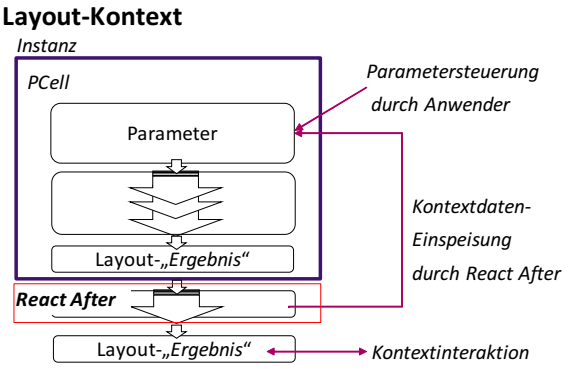


Abbildung 3. „React After“-Implementierung der kontextbasierten *PCell*.

Tool verwendet [6], welches das Funktionsprinzip der React After-PCell unterstützt.

Der Ablauf einer React After-PCell ist in Abbildung 3 dargestellt. Wie zu erkennen ist, bleiben die Funktionalitäten der herkömmlichen PCell vorhanden. Somit besteht noch immer eine Instanz, welche in Abhängigkeit von Parametern ein Layout-Ergebnis generiert.

Diese PCell-Struktur wird um eine sog. *React After-Prozedur* erweitert, die im Anschluss an die Evaluation der Kern-PCell abläuft. Die React After-Prozedur ergänzt die herkömmliche PCell um die fehlende Kontextinteraktion. Dabei bestehen die nachfolgend beschriebenen drei Schnittstellen.

1. Parametersteuerung: Dies ist die herkömmliche Eingabemaske, welche die manuelle Anpassung der PCell durch den Anwender ermöglicht.
2. Kontextinteraktion: Diese Schnittstelle unterstützt den Datenaustausch zwischen React After-Prozedur und Kontext.
3. Kontextdateneinspeisung: Hiermit werden Daten aus dem Kontext innerhalb der PCell-Instanz verarbeitet.

Die Parametersteuerung durch den Nutzer erfolgt weiterhin über die native Eingabemaske für Parameterwerte der PCell. Die React After-Prozedur ermöglicht die bereits erwähnte direkte Interaktion mit dem Kontext.

Um innerhalb der PCell-Instanz Kontextdaten zu verarbeiten oder die PCell in Abhängigkeit des Kontexts anzupassen werden dabei weiterhin die PCell-Parameter genutzt. Die Kontextdaten müssen also in ein entsprechendes Format gebracht werden, z.B. muss ein Polygon in Form einer Punktliste und einer Information über die dazugehörige Metalllage übertragen werden.

IV. Umsetzung für ein Konkretes Layoutproblem im Hochfrequenzentwurf

Mithilfe der React After-Implementierungsform wurde die nachfolgend beschriebene kontextbasierte PCell zur Generierung der eingangs beschriebenen Dummy-Metall-Geometrien entwickelt. Bei der Erstellung von Dummy-Metall sind alle Geometrien auf ihren Kontext abzustimmen. Dabei ist auf das Zusammenspiel der nachfolgenden Randbedingungen zu achten.

- Funktionale Randbedingungen: Analyse des funktionalen Einflusses der Dummy-Metall-Geometrien.
- Technologische Randbedingungen: Einhaltung von Dichte- und Abstandsregeln, um die Herstellbarkeit zu gewährleisten.
- Entwurfsmethodische Randbedingungen: Ermöglichung der EM-Simulation.

A. Anforderungen an die kontextbasierte PCell

Aus diesen Randbedingungen lassen sich die nachfolgend aufgelisteten Anforderungen an die entwickelte kontextbasierte PCell ableiten.

- Der Anwender soll die Möglichkeit haben, Form und Raster der Dummy-Metall-Geometrien für jede im Halbleiterprozess verfügbare Metalllage individuell anzupassen.
- Die PCell muss die konfigurierten Dummy-Metall-Geometrien in Abhängigkeit des Kontextes erstellen. Dabei muss die PCell für jede Metalllage den entsprechenden Mindestabstand zu den Metall-Geometrien im Kontext auslesen und einhalten. Dieser ergibt sich aus den jeweiligen Designregeln des genutzten Halbleiterprozesses.
- Um die Metalldichte anpassen zu können, muss der Nutzer die kontextabhängige Metalldichte über die Eingabemaske der PCell auslesen können.
- Um gezielte Bereiche mit Dummy-Metall-Geometrien auszufüllen, ist es nötig, dass der Nutzer den zu füllenden Bereich anpassen kann.

B. Beschreibung der entwickelten kontextbasierten PCell

Bei der React After-Implementierungsform bleibt die herkömmliche PCell-Instanz erhalten. Damit lassen sich die erstellten Dummy-Metall-Geometrien unter einer Instanz bündeln und somit gemeinsam erstellen und löschen. Der Nutzer kann diese über die Parameterwerte der Eingabemaske oder durch manuelle Anpassungen am Kontext verändern. Jedoch kann es nicht passieren, dass er versehentlich einzelne Dummy-Metall-Geometrien löscht oder verschiebt.

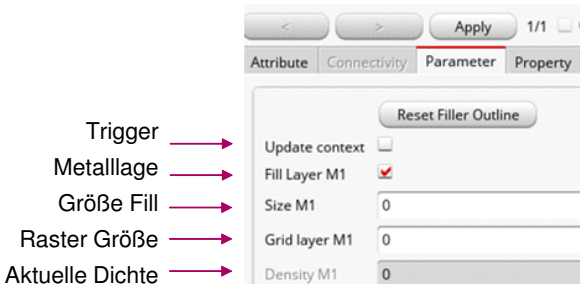


Abbildung 4. Ausschnitt der grafischen Eingabemaske für Parameter der kontextbasierten *PCell*.

Tabelle I
Generelle Parameter der kontextbasierten Füller-PCell.

Parameter-Definition	**Funktion**
Button-Parameter („*Reset Filler Outline*“)	Zurücksetzten der zu füllenden Region auf einen Standardwert.
Boolscher-Parameter („*Update Context*“)	Kontextdateneinspeisung

Die grafische Oberfläche der kontextbasierten PCell ist in Abbildung 4 dargestellt. Die zwei in Tabelle 1 dargestellten Parameter dienen der generellen Konfiguration der PCell. Für jede im Halbleiterprozess verfügbare Metalllage sind die in Tabelle 2 dargestellten Parameter verfügbar.

Die zu füllende Fläche wird über eine sogenannte *Fluid*-Geometrie bestimmt. Das Vorgehen zur Bestimmung dieser Fläche ist in Abbildung 5 angedeutet.

Ein *Fluid* hat eine feste Ausgangsgeometrie. Über die geometrischen Befehle lässt sich die Form des Fluids durch den Anwender anpassen. Damit kann dieser die zu füllende Region genau festlegen. Der Button-Parameter „*Reset Filler Outline*“ dient zum Zurücksetzen auf die Ausgangsgeometrie.

Über die in Abbildung 4 gezeigten Parameter lassen sich nach Anpassen der zu füllenden Region für jede Metalllage die individuellen Dummy-Metall-Geometrien generieren.

Über den *Float*-Parameter „*Density M1*“ kann dabei die aktuelle Dichte der gewählten Metalllage ausge-

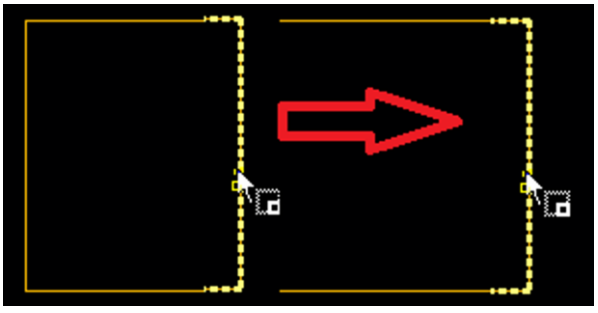


Abbildung 5. Layout-Ausschnitt der *Fluid*-Geometrie der kontextbasierten *PCell*.

Tabelle II
PCELL-PARAMETER JE METALLLAGE.

Parameter-Definition	Funktion
Button-Parameter („*Fill Layer M1*“)	Auswahl, ob die Metalllage gefüllt werden soll.
Float-Parameter („*Size M1*“)	Bestimmung der Kanntenlänge der Metall-Dummy-Geometrien.
Float-Parameter („*Grid layer M1*“)	Bestimmung des Rasters.
Float-Parameter („*Density M1*“)	Ausgabe der aktuellen Metalldichte.

Abbildung 6. Layout-Ausschnitt erstellter *Dummy-Geometrien* einer Metalllage.

lesen werden. Dabei wird ein Eingabefeld zu einem Ausgabefeld umfunktioniert. Wie in Abbildung 4 zu erkennen ist, ist das Feld ausgegraut und kann nicht vom Anwender editiert werden. Über die beschriebene Kontextdateneinspeisung der React After-Prozedur werden die Daten aus dem Kontext an dieses Feld übergeben.

Abbildung 6 zeigt die durch die kontextbasierte *PCell* generierten Dummy-Metall-Geometrien einer Metalllage. Wie zu erkennen ist, liegen die Dummies auf einem festen Raster. Weiterhin ist zu erkennen, dass die beiden im Kontext liegenden Geometrien, also die beiden größeren Polygone, nicht mit Dummies ausgefüllt werden. Die kontextbasierte PCell hält dabei automatisch den für jede Lage und Geometrie spezifischen Mindestabstand zu den Dummies ein. Die hierfür relevanten Informationen werden aus einer Technologiedatei ausgelesen und in der PCell-Prozedur verarbeitet.

V. DISKUSSION

A. Beurteilung der entwickelten kontextbasierten PCell

Mit der Kontextinteraktion und Kontextdateneinspeisung wurde das vorliegende Problem erfolgreich gelöst. Damit ermöglicht die kontextbasierte PCell die Automatisierung eines bis dato aufwendigen und nur manuell durchführbaren Schritts des Layoutentwurfs.

Der quantitative Nutzen ist schwer in Zahlen zu greifen. Jedoch lässt sich feststellen, dass die manuelle Erstellung der Metall-Dummy-Geometrien zum Teil einige Stunden in Anspruch nehmen kann. Zieht man in Betracht, dass dies in einem größeren Layoutentwurf durchaus über mehr als einhundertmal durchgeführt werden muss, so lässt sich daraus die eingesparte Entwurfszeit abschätzen.

Neben der Zeitersparnis trägt die entwickelte kontextbasierte PCell auch zur Steigerung der Qualität des Layoutentwurfs bei. Da Dummies bisher in aufwendiger Handarbeit erstellt werden mussten, war die Simulation verschiedener Dummy-Geometrien ein zeitaufwendiger Prozess. Entsprechend wurde lediglich die Auswirkung *einer* erstellten Dummy-Geometrie untersucht. Die automatisierte Erstellung der Geometrien ermöglicht es nun, verschiedene Dummy-Geometrien zu erstellen und deren Einfluss auf den Kontext zu simulieren. Damit ist es nun möglich, aus verschiedenen Dummy-Geometrien diejenige zu wählen, die den geringsten parasitären Einfluss auf den Kontext aufweist.

B. Beurteilung der PCell-Implementierungsform React After

Die entwickelte kontextbasierte PCell stellt eine signifikante Weiterentwicklung gegenüber einer herkömmlichen PCell dar. Allerdings weist die Implementierungsform einige Hürden auf, welche die Entwicklung einer kontextbasierten PCell erschweren.

Zu nennen ist hier in erster Linie die Kontextdateneinspeisung. Dafür müssen die Parameter der herkömmlichen PCell verwendet werden. Entsprechend ist eine Formatierung der Kontextinformation in einen passenden Parameterwert innerhalb der React After-Prozedur notwendig. Zum einen erhöht sich dadurch die Laufzeit der PCell, zum anderen muss in Abhängigkeit der Kontextdaten ein individueller SKILL-Code entwickelt werden, der die Kontextdaten entsprechend formatiert und in der herkömmlichen PCell-Prozedur weiterverarbeitet.

Weiterhin besteht bei der Implementierungsform die Gefahr, dass es zu einer Rekursion kommt. Dies ist der Fall, wenn die React After-Prozedur Parameterwerte der PCell ändert. Dabei wird die PCell erneut evaluiert, wodurch auch die React After-Prozedur erneut durchlaufen wird. Werden nun wieder Parameterwerte verändert, so besteht das Risiko einer Endlosschleife, die letztendlich zu einem Absturz der CAD Software führen kann. Aus diesem Grund ist es notwendig, die Kontextabhängigkeit über *Boolsche* Parameter auszulösen. Diese werden nach einer manuellen Interaktion zurückgesetzt, was eine erneute Evaluation der PCell verhindert.

VI. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden die bestehenden Ansätze zur Automatisierung des Layoutentwurfs beschrieben. Dabei wurde festgestellt, dass weder Generatoren noch Optimierungsalgorithmen zur Automatisierung spezieller Probleme des Layoutentwurfs von Hochfrequenz-Schaltungen hinreichend geeignet sind. Optimierungsalgorithmen scheiden wegen der Verschiedenartigkeit und Menge der vorliegenden Randbedingungen aus, da diese sich nicht hinreichend klar und in effizienter Weise formalisieren lassen. Herkömmliche Generatoren scheiden aufgrund der fehlenden Kontextabhängigkeit aus. Aus diesem Grund wurde eine funktionale Erweiterung des bekannten Generatorprinzips, die kontextbasierte PCell mit deren aktuell verfügbarer Implementierungsform der React-After-PCell vorgestellt.

Im Anschluss wurde die Umsetzung einer konkreten kontextbasierten PCell beschrieben, mit deren Hilfe ein bisher nur manuell lösbares Layoutproblem einer automatischen Lösung zugeführt werden konnte: die Erstellung von Dummy-Metall-Geometrien im Layoutentwurf von Hochfrequenz-Schaltungen. Die verwendete Implementierungsform der React After-PCell deckt dabei alle erforderlichen Funktionalitäten ab. Sie stellt damit eine innovative Problemlösung dar und beweist, dass die Erweiterung des PCell-Prinzips um die Kontextabhängigkeit signifikante Beiträge zur weiteren Automatisierung des analogen Layoutentwurfs beitragen kann.

Die vorgestellte kontextbasierte PCell ist lediglich ein Beispiel für die Automatisierung eines Problems des Layoutentwurfs. In der Praxis wurden durch die Verwendung kontextbasierter PCells bereits weitere Problemstellungen des Layoutentwurfs von Hochfrequenz-ICs, wie die Erzeugung und Verlegung von Transmission-Lines (geschirmte Leitungen) oder die Erzeugung spezieller HF-Routing-Strukturen für die DC-Spannungsversorgung, automatisiert [7].

Die Arbeit zeigt Ansatzpunkte für die weitere Verbesserung kontextbasierter PCells. Eine wichtige Funktionalität wäre dabei die Weiterentwicklung der Kontextdateneinspeisung, so dass die im Kontext arbeitende Prozedur über eine Variable direkt mit der PCell kommunizieren kann. Dadurch könnte die aufwendige Formatierung der Daten entfallen, was die Entwicklung deutlich erleichtern würde.

Grundsätzlich konnte mit der Arbeit die qualitäts- und produktivitätssteigernde Wirkung kontextbasierter PCells bei der Anwendung auf Probleme des Layoutentwurfs von Hochfrequenz-Schaltungen in modernen CMOS-Technologien nachgewiesen werden. Die dabei gelösten Probleme resultieren allerdings nicht nur aus den hohen Frequenzen, sondern auch aus den sehr kleinen Strukturgrößen. Diese führen zu neuartigen Randeffekten, welche auch im Layoutentwurf niederfrequenter Schaltungen berücksichtigt werden müssen. Die Nutzung kontextbasierter PCells kann daher zukünftig auch im Layoutentwurf von Niederfrequenz-Schaltungen wesentlich zur Erhöhung der Automatisierung beitragen.

Literaturverzeichnis

[1] N. Rohani, J. Zhang, J. Lee, J. Bai, „A 28-nm CMOS 76−81-GHz power amplifier for Automotive radar applications“, in *17th Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems*, 15.01.-18.01.2017, Hyatt Regency Phoenix, Phoenix, Arizona, USA, pp. 85-87, DOI=10.1109/SIRF.2017.7874378.

[2] J. Scheible, J. Lienig, „Automation of Analog IC Layout – Challenges and Solutions“, in *Proc. of the Symp. on Int. Symp. on Physical Design - ISPD '15*, 29.03.-1.4.2015, Monterey, CA, USA, pp. 33-40. DOI=10.1145/2717764.2717781.

[3] J. Lienig, *Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen für die Entwurfsautomatisierung*, Springer-Verlag, 2006. ISBN 978-3-540-29942-4, DOI=10.1007/3-540-29942-4.

[4] D. Marolt, J. Scheible, G. Jerke, V. Marolt, „CAPABLE: A Layout Automation Framework for Analog IC Design,“, *54. MPC-Workshop*, Ulm, Germany, 10.07.2015, ISSN 1868-9221, pp. 49-59.

[5] D. Marolt, J. Scheible, G. Jerke, V. Marolt, „Analog layout automation via self-organization: Enhancing the novel SWARM approach“, in *Proc. of the IEEE 7th Latin American Symposium on Circuits and Systems (LASCAS 2016)*, 28.02.2016 -02.03.2016, Florianopolis, Brasilien, pp. 55-58, DOI=10.1109/LASCAS.2016.7451008.

[6] G. Jerke, V. Marolt, C. Buerzele, P. Herth, T. Burdick, „Visual PCell Programming with Cadence PCell Designer“, in *Proc. Cadence CDNLive EMEA 2013*, Munich, Germany, 2013.

[7] M. Thoma, *Entwurf kontextbasierter PCells für Hochfrequenzschaltungen in modernen CMOS-Technologien*, Masterthesis am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen, 30.06.2017.

Matthias Thoma erhielt den akademischen Grad des B. Eng. im Jahr 2014 von der Dualen Hochschule Baden-Württemberg im Fach Wirtschaftsingenieurwesen. Seit Ende 2014 studiert er im Masterstudiengang Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik und arbeitete als Hilfswissenschaftler in der dortigen EDA-Forschungsgruppe. Im Ramen seiner Masterthesis arbeitet er in Kooperation mit der Robert Bosch GmbH am Thema der vorliegenden Arbeit.

Daniel Marolt studierte Mechatronik an der Hochschule Reutlingen und erhielt dort im Jahre 2008 den akademischen Grad B. Eng. und 2009 den Abschluss M. Sc.. Seit 2010 ist er als wissenschaftlicher Mitarbeiter an der Hochschule Reutlingen tätig, wo er 2011 am Robert Bosch Zentrum für Leistungselektronik Aufgaben in Forschung und Lehre übernahm und nun in Kooperation mit der Robert Bosch GmbH an EDA-Projekten arbeitet.

Jürgen Scheible erhielt 1987 den akademischen Grad des Dipl.-Ing. und 1991 den Grad des Dr.-Ing. jeweils in Elektrotechnik von der Universität Karlsruhe. Von 1992 bis 2010 arbeitete er im Geschäftsbereich Automotive Electronics der Robert Bosch GmbH, zuletzt als Leiter der Entwicklungsabteilung für den IC-Layoutentwurf. Seit 2010 ist er Professor für Electronic Design Automation am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen, wo er zur Automatisierung des Entwurfs integrierter analoger Schaltkreise forscht.

Gregor Tretter erlangte 2011 den akademischen Grad des Dipl.-Ing. und 2016 den Grad des Dr.-Ing. in Elektrotechnik von der Technischen Universität Dresden auf dem Gebiet der integrierten analogen Schaltungstechnik. Schwerpunkt seiner Arbeit waren Hochfrequenz-Breitbandschaltungen und Datenwandlerstrukturen. Seit 2016 arbeitet er bei der Robert Bosch GmbH an integrierten Millimeterwellenschaltungen.

Göran Jerke erlangte 2000 den akademischen Grad des Dipl.-Ing. von der Technischen Universität Dresden auf dem Gebiet der Feinmechanik und Elektroniktechnologie. Schwerpunkt seiner Arbeit sind die Konzipierung und Entwicklung von EDA-Lösungen zur Vermeidung von Elektromigration in IC-Metallisierungen und dem Constraint-geführten IC-Designentwurf. Seit 2000 arbeitet er bei der Robert Bosch GmbH im Bereich der EDA-Forschung und Vorausentwicklung.

Implementierung eines CDMA-Transceivers für ein Virtex 5 FPGA

Lukas Nothhelfer, Lothar Schmidt, Anestis Terzis

***Zusammenfassung*—Vorgestellt wird der Entwurf eines CDMA-Transceivers, der mithilfe der Hardwarebeschreibungssprache VHDL in einen FPGA vom Typ Virtex 5 (Fa. Xilinx) implementiert wird. Der FPGA ist Bestandteil des Development Boards Genesys™. Das Ziel des Entwurfes ist es, für eine leitungsgebundene Übertragung im Basisband, die maximal mögliche Taktfrequenz des Transceivers zu ermitteln. Im Sinne einer Entwurfsraumexploration werden daher in unterschiedlichen Architekturen verschiedene Ressourcen des FPGAs genutzt. Anhand der Ergebnisse des Post-Place-and-Route (Post-PAR) Timing Reports können Engpässe in den Schaltungen ermittelt werden. Durch Optimierung der VHDL-Beschreibungen, bzw. Verwendung anderer Hardwareressourcen wird versucht die optimale Architektur zu finden.**

Der Transceiver wurde am realen System einerseits mit interner Verbindung zwischen Sender und Empfänger untersucht und andererseits bei einer leitungsgebundenen Übertragung zwischen zwei Boards.

***Schlüsselwörter*—CDMA Transceiver, FPGA, VHDL, Post-PAR Timing Report, Demodulator, DSP48E-Slice**

I. Einleitung

Dieser Beitrag basiert auf den Ergebnissen von [1], in der erstmals an der Hochschule Ulm verschiedene Architekturkonzepte für einen CDMA-Transceiver in einem Virtex 5 FPGA untersucht wurden. Ziel dieser Untersuchung ist es eine Architektur zu finden, mit der, unter Vernachlässigung der Leistungsaufnahme, die maximal mögliche Taktfrequenz für eine leitungsgebundene Punkt-zu-Punkt-Übertragung möglich ist. Die Freiheitsgrade liegen hierbei in der Verwendung der Ressourcen, die der Virtex 5 anbietet. Im Sinne einer Entwurfsraumexploration sollen unterschiedliche Ressourcen gezielt eingesetzt und deren Auswirkungen auf die Systemfrequenz in Simulation und Realität gegenübergestellt werden. In weiterführenden Projekten soll der Transceiver für eine Funkübertragung in einem Mehrbenutzersystem eingesetzt werden. Als Entwicklungsumgebung steht die Software ISE Webpack 14.2 von Xilinx zur Verfügung.

II. Code Division Multiple Access (CDMA) mit Direct Sequence Spread-Spectrum (DSSS)

Die theoretische Grundlage für die nachfolgenden Untersuchungen bildet das „Code Divison Multiple Access“-Vielfachzugriffsverfahren, das vielen Sendern erlaubt, gleichzeitig ein Übertragungsmedium zu nutzen, ohne sich gegenseitig zu stören. Dabei wird jedem Sender eine einzigartige Codefolge zugeordnet, mit der sein Sendespektrum gespreizt wird. Über einen entsprechenden Empfänger (z.B. Korrelationsempfänger) kann nach der Übertragung der ursprünglich gesendete Datenstrom zurückgewonnen werden. Die Aufweitung des Sendespektrums hat zum Vorteil, dass einerseits Störungen unterdrückt werden, und andererseits die spektrale Leistungsdichte des gesendeten Signals gesenkt wird. Dies macht es auch leichter die gesetzlichen Vorschriften zur Einhaltung von Grenzwerten, in Bezug auf die verursachte Störstrahlung, einzuhalten [2].

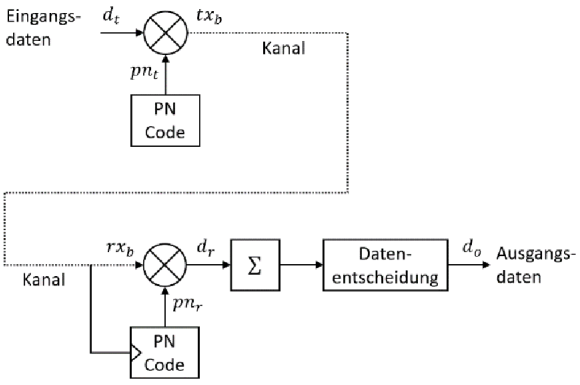


Abbildung 1. Aufbau einer DSSS Übertragungsstrecke.

Die Kanalseparierung mittels spezieller Codefolgen, wie es bei CDMA angewendet wird, kann durch verschiedene Verfahren realisiert werden. Für die folgende Untersuchung wird das „Direct Sequence Spread-Spectrum“-Verfahren (DSSS) verwendet, bei dem der zu sendende Datenstrom mit einer höherfrequenten Spreizfolge vor der Übertragung multipliziert wird.

Der Aufbau einer solchen Übertragungsstrecke ist in Abbildung 1 dargestellt. Der Datenstrom wird im Sender mit einer hochfrequenten Codefolge (m-Folge) multipliziert und das daraus resultierende Signal über den Kanal an den Empfänger übertragen. Als Codefolge wird ein sogenannter Pseudo-Noise-Code (PN-Code) verwendet. Dort wird aus dem empfangenen Signal das eigentliche Datensignal durch einen Korrelationsempfänger mit Schwellwertentscheider zurückgewonnen. Zur Korrelationsbildung erzeugt der Emp-

Lukas Nothhelfer, lukas.nothhelfer@stud-mail.uni-wuerzburg.de, Universität Würzburg, Am Hubland, 97074 Würzburg.
Lothar Schmidt, schmidt@hs-ulm.de, und Anestis Terzis, terzis@hs-ulm.de, Hochschule Ulm, Prittwitzstraße 10, 89075 Ulm.

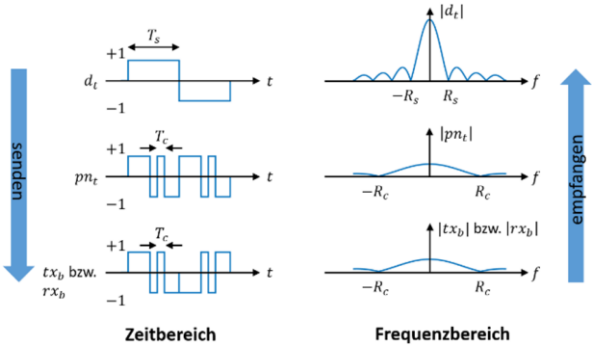


Abbildung 2. Zeit- und Frequenzdarstellung der Signale eines DSSS-Systems.

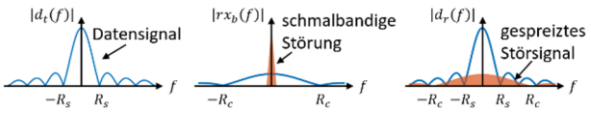


Abbildung 3. Signalspektren bei schmalbandiger Störung.

fänger für sich selbst ebenfalls eine hochfrequente Codefolge (m-Folge), die identisch ist mit der, die der Sender benutzt hat und mit der das ankommende Signal korreliert wird. Bei der Korrelation werden beide Signale miteinander multipliziert und das Ergebnis über eine Periode der Codefolge akkumuliert. Der Schwellwertentscheider trifft aufgrund des Akkumulationsergebnisses die Datenentscheidung. Die aus diesen Vorgängen resultierenden Signalverläufe sind beispielhaft in Abbildung 2 dargestellt.

Der Nutzen der Signalspreizung wird klar, wenn man betrachtet, welche Störungen in einem Kanal auftreten können. Bei einer schmalbandigen Störung wird das Störsignal im Empfänger durch die Multiplikation mit der hochfrequenten Spreizfolge gespreizt, wohingegen das Datensignal durch die korrelative Kompression aus dem Rauschen herausgehoben wird (vgl Abbildung 3).

Ist bei einer breitbandigen Störung die empfangene Rauschleistung nicht allzu hoch, kann auch hier das Datensignal aus dem Rauschen gehoben werden (siehe Abbildung 4).

A. M-Folgen

Einen wesentlichen Bestandteil eines DSSS-Übertragungssystems bilden die sogenannten Pseudo-Noise-Folgen (PN-Folgen). Dabei handelt es sich um pseudo-zufällige, periodische Codefolgen, die üblicherweise über linear zurückgekoppelte Schieberegister erzeugt werden. Die Rückkopplung erfolgt über ein XOR-Glied, was der modulo-2-Addition entspricht. Mit einem Schieberegister der Stufenanzahl n können Folgen der maximalen Länge $L = 2^n - 1$ erzeugt werden (m-Folgen). Üblicherweise werden die Positionen, an denen die Rückkopplungen erfolgen, in eckigen Klammern angegeben. In Tabelle

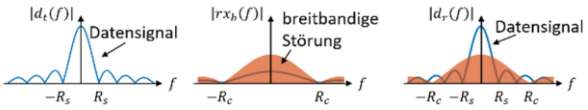


Abbildung 4. Signalspektren bei breitbandiger Störung.

Tabelle I
RÜCKKOPPLUNGEN ZUR ERZEUGUNG VON M-FOLGEN.

Stufen	Folgen-länge	Rückkopplungen	Unterschied-liche Codes
4	15	[4, 1]	2
5	31	[5, 3], [5, 4, 3, 2], [5, 4, 2, 1]	6
6	63	[6, 1], [6, 5, 2, 1], [6, 5, 3, 2]	6

1 ist eine Übersicht über mögliche Rückkopplungen für m-Folgen gegeben [3].

Ein wichtiges Beurteilungskriterium von PN-Folgen sind deren Korrelationseigenschaften. Dabei unterscheidet man zwischen der Autokorrelation (Korrelation der Folge mit sich selbst) und der Kreuzkorrelation (Korrelation der Folge mit einer anderen Folge). Die Korrelation zwischen zwei Folgen kann man sich gedanklich so vorstellen: Verbindet man jeweils den Anfang und das Ende einer Folge, so entsteht ein Ring. Legt man nun die beiden Ringe so übereinander, dass sich die Anfangspunkte genau gegenüberstehen, und zählt die Anzahl an Übereinstimmungen und subtrahiert die Anzahl an Nichtübereinstimmungen, so hat man den Korrelationswert für die Verschiebung $\tau = 0$ bestimmt. Führt man dies für jede mögliche Verschiebung durch, so führt dies zur Korrelationsfunktion [2].

In (1) ist die diskrete periodische Autokorrelationsfunktion (PAKF) gegeben und in (2) die diskrete periodische Kreuzkorrelationsfunktion (PKKF).

$$\varphi_{xx}(\tau) = \sum_{i=1}^{L} x(i) \cdot x(i+\tau) \quad 0 \leq \tau \leq L-1 \quad (1)$$

$$\varphi_{xy}(\tau) = \sum_{i=1}^{L} x(i) \cdot y(i+\tau) \quad 0 \leq \tau \leq L-1 \quad (2)$$

Eine wichtige Besonderheit von m-Folgen ist die scharfe Autokorrelationsfunktion. Mit Ausnahme der Verschiebung $\tau = 0$ liefern alle Verschiebungen den Wert -1. An der Stelle $\tau = 0$ hat die Autokorrelationsfunktion ihr Maximum erreicht, was genau der Folgenlänge entspricht (siehe Abbildung 5).

Die Werte der Kreuzkorrelationsfunktion sind wichtig, wenn man eine PN-Folge für die Tauglichkeit in einem Mehrbenutzersystem überprüfen möchte, da sie Aufschluss über die Interferenzeigenschaften mit anderen PN-Codes gibt. Da hier kein Mehrbenutzersystem entworfen wird, soll nicht näher darauf eingegangen werden.

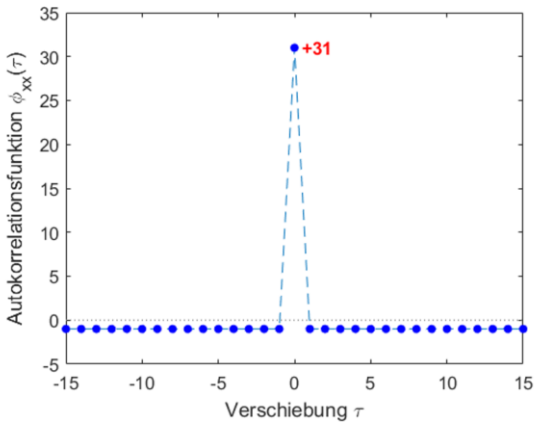


Abbildung 5. Autokorrelationsfunktion einer m-Folge der Länge 31.

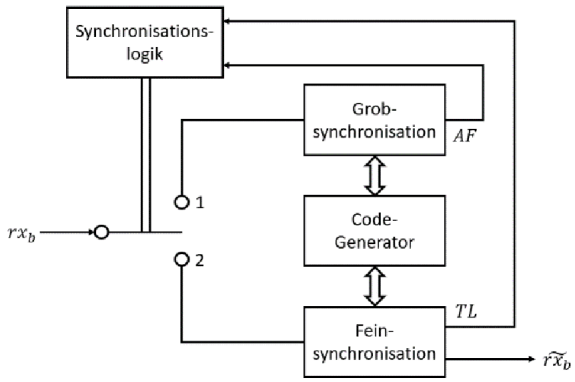


Abbildung 6. Blockschaltbild der Synchronisationseinheiten.

B. PN-Dekorrelator

Die scharfe Autokorrelationsfunktion von m-Folgen kann man sich zunutze machen, wenn man einen ankommenden Datenstrom nach einem bestimmten Muster durchsuchen möchte. Zur Rückgewinnung der ursprünglichen Information korreliert der Empfänger über eine Periode seine lokal erzeugte m-Folge mit dem empfangenen Datenstrom. Liegt am Ende der Periode der Maximalwert der Autokorrelationsfunktion positiv vor, so wurde eine „1“ übertragen. Liegt der Maximalwert hingegen negativ vor, so wurde eine „0“ gesendet.

C. Synchronisation

Die Synchronisation in einem DSSS-Übertragungssystem ist notwendig, um die Phasenlagen der PN-Folgen im Sender und Empfänger in Übereinstimmung zu bringen. Da in dem Empfangssignal keine Information über den Anfang und das Ende der zur Modulation verwendeten PN-Folge verfügbar ist, muss der Empfänger die Phasenlage seiner lokal erzeugten PN-Folge mit dem Empfangssignal in Übereinstimmung bringen. Erst bei korrekter Phasenlage ist eine Demodulation möglich. Bei der Synchronisation unterscheidet man zwischen zwei Phasen: Grobsynchronisation und Feinsynchronisation. Bei der Grobsynchronisation wird die Phasenlage des Empfangssignals mit der im Empfänger erzeugten PN-Folge bis auf eine Abweichung von $\pm T_c/2$ zur Übereinstimmung gebracht. T_c gibt dabei die Periodendauer eines Elements (Chips) der PN-Folge an. Die Feinsynchronisation dient dazu, die verbleibende Phasendifferenz auszulöschen. Dazu wird während dieser Phase der Wert der PAKF in kleinen Schritten maximiert. Der Wechsel von Grobsynchronisation auf Feinsynchronisation wird von einer Synchronisationslogik gesteuert [2] (siehe Abbildung 6).

Das empfangene Signal rx_b wird zunächst an die Grobsynchronisation weitergegeben (Schalter in Stellung 1). Ist die Grobsynchronisation abgeschlossen, so wird dies der Synchronisationslogik durch das Signal AF mitgeteilt. Die Synchronisationslogik bewegt nun den Schalter in Stellung 2, wodurch das empfangene Signal an die Feinsynchronisation umgeleitet wird. Treten im empfangenen Signal Störungen auf, kann dies dazu führen, dass die Regelschleife ausrastet und erneut eine Grobsynchronisation notwendig wird. In diesem Fall wird der Wechsel durch das Signal TL angezeigt.

Als Maß für die Übereinstimmung der Phasenlage des Empfangssignals mit der im Empfänger erzeugten PN-Folge dient die Korrelation beider Signale. Zu Beginn der Synchronisation definiert man das Unsicherheitsintervall, das jenen Bereich bezeichnet, in dem die Phase zu suchen ist. Im schlimmsten Fall ist das Unsicherheitsintervall genauso breit wie die PN-Folge lang ist. Bei der Grobsynchronisation wird das Unsicherheitsintervall in diskreten Schritten durchsucht. Hierbei kommen parallele (Maximum Likelihood) und serielle Suchverfahren zum Einsatz. Das parallele Suchverfahren ist dabei das einfachste, schnellste aber auch aufwändigste Verfahren, und in der Praxis sehr kostenintensiv. Bei diesem Verfahren werden alle möglichen Phasen parallel durchsucht und die Korrelation mit der höchsten Übereinstimmung gewählt.

Bei seriellen Suchverfahren wird nur ein Korrelator verwendet, mit dem das Unsicherheitsintervall seriell durchsucht wird. Nach jeder Korrelation wird der

Codetakt um $T_c/2$ angehalten, bis die korrekte Phase und somit das Maximum der Korrelationsfunktion gefunden ist. Wenn zu Beginn keine Information über die Phasenlage verfügbar ist, so ist jede Phase gleichwahrscheinlich. In diesem Fall wird das Suchen in gerader Linie verwendet, bei dem das Unsicherheitsintervall blind in einer Richtung durchsucht wird [2].

Darüber hinaus existieren noch weitere serielle Suchverfahren, die in Abbildung 7 zusammengefasst

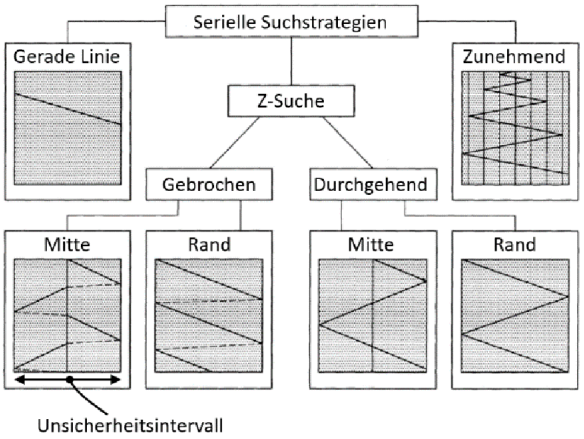


Abbildung 7. Übersicht über die seriellen Suchverfahren.

sind.

III. Field Programmable Gate Array (FPGA)

Als Field Programmable Gate Array (FPGA) werden programmierbare Logikbausteine bezeichnet, die individuell programmiert werden können. Die Bezeichnung „Field Programmable“ hat ihren Ursprung darin, dass die Bausteine vom Anwender „im Feld“ programmiert werden können und nicht nur vom Hersteller über ein aufwendiges Verfahren anhand von Metallmasken.

Ein moderner FPGA (z.B. Virtex 5) besteht aus vielen einzelnen Basiszellen (CLBs), die über programmierbare Verbindungen miteinander in horizontaler und vertikaler Richtung verbunden sind (siehe Abbildung 8) [4].

Jede Basiszelle beinhaltet einen Vorrat an digitalen Grundbauelementen, wie z.B. D-Flipflops, Schieberegistern, Multiplexern und ladbaren Tabellen (engl. Look-Up-Table, LUT), über die komplexe logische Funktionen umgesetzt werden können. Neben den Basiszellen besitzt ein moderner FPGA weitere Funktionsblöcke, wie z.B. Taktressourcen und -routen, Ein- und Ausgabeblöcke, Blockspeicher (Block RAM) und in Hardware implementierte Multiplizierer (z.B. DSP48E), mit denen Rechenoperationen, wie z.B. multiplizieren und akkumulieren besonders effizient ausgeführt werden können.

A. Basiszellen (CLBs)

Die Basiszellen (engl. Configurable Logic Blocks, CLBs) stellen die Hauptressourcen zur Implementierung sequentieller und kombinatorischer Schaltungen zur Verfügung. Abhängig von der Ausführungsart kann der Virtex 5 bis zu 25920 Basiszellen enthalten, die in einer 240 × 108 (Zeilen × Spalten) Matrix angeordnet sind. In der verwendeten Version (XC5VLX50T) liegen 3600 Basiszellen in einer 120 × 30 Anordnung vor [4].

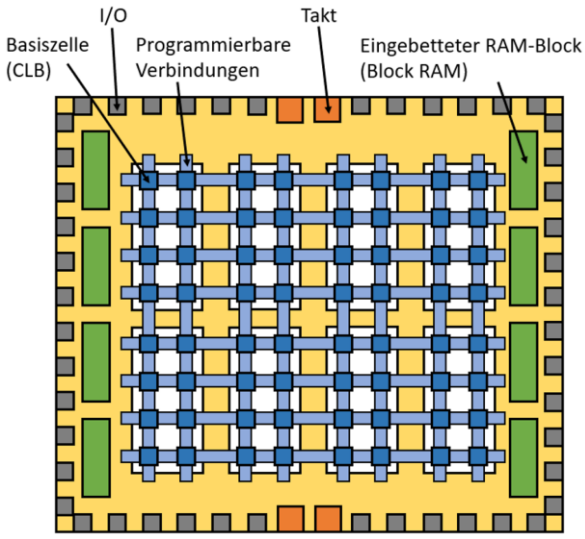


Abbildung 8. Moderne FPGA-Architektur.

Abbildung 9. Digilent Genesys FPGA Board.

B. Taktressourcen und -routen

Im Virtex 5 werden zwei Arten von Taktrouten unterschieden: Globale und regionale Routen. Anhand der globalen Taktrouten können alle Arten von sequentiellen Ressourcen im gesamten FPGA versorgt werden. Deren Anzahl ist jedoch auf 32 beschränkt. Anhand regionaler Routen können für kleinere Teilschaltungen zusätzliche Pfade verwendet werden.

Zur Erzeugung von Takten stehen im Virtex 5 sogenannte Clock Management Tiles (CMTs) zur Verfügung. Jeder CMT beinhaltet zwei Digital Clock Manager (DCM) und eine Phase-Locked-Loop (PLL). Beide ermöglichen es, aus einem Referenztakt einen Takt beliebiger Frequenz zu erzeugen [5].

C. I/O Elemente

I/O Elemente stellen die Schnittstelle zwischen den Pads und internen Verbindungen des FPGA dar. Über

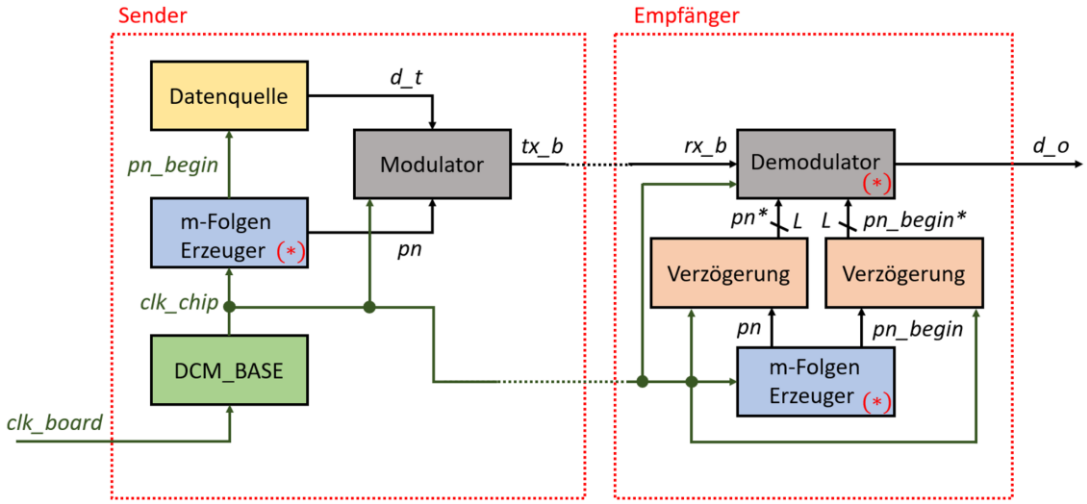


Abbildung 10. Aufbau des entworfenen Transceivers.

diese Schnittstellen wird eine Vielzahl an Standards, z.B. LVDS und LVCMOS unterstützt. Für die Übertragung auf langen Leitungen eignet sich insbesondere der erweiterte LVDS-Standard (LVDSEXT_25), der außerdem eine interne Terminierung mit einem 100 Ohm Widerstand ermöglicht [5].

D. DSP48E-Slice

Der im Virtex 5 enthaltene DSP48E-Slice eignet sich insbesondere für rechenaufwändige Signalverarbeitungsschaltungen. Der zahlreiche Funktionsumfang dieses Slices umfasst unter anderem: Multiplizieren und akkumulieren (MACC), addieren von drei Eingängen, bitweise Logikoperationen, detektieren von voreingestellten Pattern und komplexe Multiplikation bei Kaskadierung des Slices.

IV. Digilent Genesys Virtex 5 FPGA Board

Das Digilent Genesys FPGA Board (siehe Abbildung 9) bildet die Hardware-Grundlage für die vorliegende Arbeit. Es beinhaltet einen FPGA vom Typ Virtex 5 in der Ausführung LX50T. Neben zahlreichen Schnittstellen, über die eine Vielzahl an Protokollen unterstützt werden (z.B. Ethernet, HDMI, USB), verfügt dieses über einen On-Board-Programmer mit dem der FPGA über ein einfaches USB-Kabel programmiert werden kann [6].

V. Entwurf eines CDMA-Transceivers

Vor dem Entwurf des Transceivers müssen einige Schlüsselfragen geklärt werden. Dazu gehören z.B., welche PN-Folgenlängen verwendet werden und ob neben dem modulierten Signal weitere Informationen an den Empfänger übertragen werden sollen.

A. Rahmenbedingungen festlegen

Als PN-Folgen kommen hier ausschließlich m-Folgen zum Einsatz. Die Folgenlänge wurde auf $L = 7...255$ festgelegt. Da es hier hauptsächlich um die Untersuchung möglicher Schaltungsarchitekturen geht, wird der Chiptakt ebenfalls an den Empfänger übertragen.

B. Architekturentwurf

Bei der Architekturentwicklung sind die von A. Goiser ([2]) und J. Meel ([3]) vorgestellten Architekturen für einen CDMA-Transceiver berücksichtigt worden. Im Wesentlichen besteht der Sender aus drei Blöcken: Einer Datenquelle, einem Erzeuger für m-Folgen (Generator oder Speicher) und einem Modulator, der die Multiplikation des Datensignals von der Datenquelle mit der m-Folge aus dem Erzeuger ausführt. Der Empfänger besitzt ebenfalls einen Erzeuger, der für den Demodulator eine m-Folge zur Verfügung stellt. Diese wird dort mit dem Empfangssignal multipliziert und korreliert. Darüber hinaus beinhaltet der Transceiver ein Taktmanagement, das den Boardtakt des Digilent Genesys FPGA-Boards auf einen gewünschten Chiptakt hoch- bzw. heruntersetzt. In Abbildung 10 ist der Aufbau des entworfenen CDMA-Transceivers abgebildet.

Die Taktaufbereitung des Chiptaktes im Sender erfolgt über einen Digital Clock Manager (DCM), über den sich der gewünschte Chiptakt einstellen lässt. Der Datentakt für die Datenquelle wird über ein Flag erzeugt, das der m-Folgen Erzeuger zu Beginn einer neuen PN-Folge generiert und für die Dauer des ersten Chips der Folge anhält. Da der Chiptakt ebenfalls übertragen wird, benötigt der Empfänger keine takterzeugende bzw. regenerierende Einheit. Der Demodulator im Empfänger vereint die Komponenten Korrelator und Detektor in einem einzigen Modul.

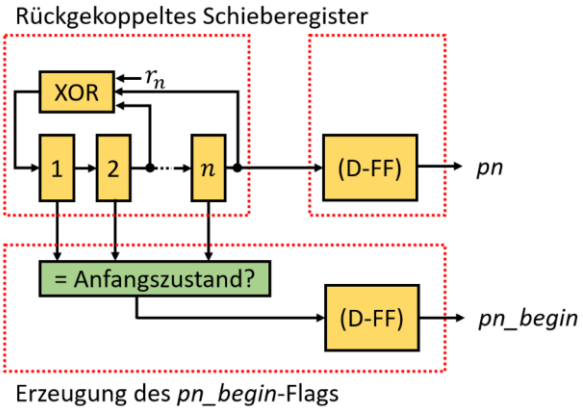


Abbildung 11. PN-Generator in der ersten Architekturvariante.

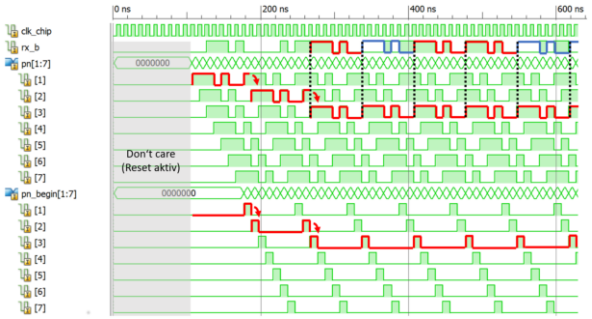


Abbildung 12. Synchronisation der lokal erzeugten m-Folge mit dem empfangenen Signal im Demodulator am Beispiel einer [3, 1]-Folge.

Zur Synchronisation der PN-Folgen werden über zwei Verzögerungsmodule sowohl die PN-Folge als auch das Flag, das den Beginn der Folge markiert, in allen möglichen Verzögerungen auf einen L-Bit breiten Bus gelegt, der am Demodulator anliegt. L gibt dabei die Länge der PN-Folge an. Auf diese Weise stehen dem Demodulator alle Verschiebungen der PN-Folge zur Verfügung, die der Reihe nach ausprobiert, und jeweils über eine Periode mit dem Empfangssignal korreliert werden. Der Korrelationsanfang und das Korrelationsende wird über das Flag gesteuert, das den Beginn der PN-Folge markiert. Hat die Korrelation ihr Maximum bzw. Minimum erreicht, so wechselt der Demodulator auf eine andere, zeitlich verschobene PN-Folge, bis die optimale Übereinstimmung mit der PN-Folge des Senders vorliegt.

Im Hinblick auf die maximal mögliche Taktfrequenz sind in der Entwicklung der einzelnen Module mehrere Architekturen umgesetzt und untersucht worden. Dies betrifft besonders die Module, die im Vergleich zu anderen eine komplexere Aufgabe haben und zudem mit der höchsten auftretenden Systemfrequenz (Chipfrequenz) getaktet werden. In Abbildung 10 sind die Module, für die mehrere Architekturen entwickelt wurden, mit einem $(*)$ markiert. Der Grundaufbau des Transceivers bleibt dabei entsprechend Abbildung 10 unverändert.

C. Datenquelle, Modulator und Verzögerungsmodul

Die Datenquelle ist in Form einer Endlosschleife beschrieben, die immer das gleiche Bitmuster „10101100111000“ hervorbringt, das in Form von Distributed RAM im FPGA gespeichert wird. Sie wird durch ein Flag getaktet, das der Erzeuger für die m-Folgen zum Beginn einer neuen m-Folge erzeugt. Somit wird zum Beginn einer neuen m-Folge ein neues Datenbit von der Datenquelle ausgegeben.

Der Modulator im Sender besteht im Wesentlichen aus einer getakteten XNOR-Verknüpfung, mit der die Multiplikation der m-Folge mit dem Datenstrom abgebildet wird.

Das Verzögerungsmodul beinhaltet ein L-Bit breites Schieberegister, das durch Abgriffe nach jeder einzelnen Stufe die gewünschten Verzögerungen des Eingangssignals erzeugt. Es hat somit einen 1-Bit breiten Eingangsport und einen L-Bit breiten Ausgangsport.

Die Architekturen für die Datenquelle, den Modulator und das Verzögerungsmodul bleiben bei allen Schaltungsvarianten gleich.

D. Architektur 1: PN-Generator und sequentieller Demodulator

In einem ersten Schaltungsentwurf ist der Transceiver vollständig in VHDL ohne Verwendung hardwarespezifischer Funktionselemente beschrieben worden. Davon ausgenommen sind die unabdingbaren Elemente zur Erzeugung des Chiptaktes (DCM), sowie die zur Ein- und Ausgabe nötigen Buffer-Elemente. Die m-Folgen werden in dieser ersten Variante anhand eines PN-Generators (d.h. über ein rückgekoppeltes Schieberegister) erzeugt (siehe Abbildung 11). Zur Erzeugung des *pn_begin*-Flags werden zyklisch die inneren Zustände des Schieberegisters überprüft. Ist das Schieberegister wieder in den Anfangszustand gelangt, so wird das Flag ausgegeben.

Der Demodulator ist bei dieser Architektur als ein einziger VHDL-Prozess beschrieben, der alle Rechenoperationen ausführt (Synchronisation, Korrelation und Datenentscheidung).

Die Synchronisation ist in Form eines seriellen Suchens in gerader Linie mit Steigung $m = 1$ umgesetzt. Der Korrelationsanfang und das Korrelationsende wird über das *pn_begin*-Flag gesteuert. In Abbildung 12 ist die Verhaltenssimulation des Demodulators dargestellt, in der der Folgenwechsel zu sehen ist, solange die Folgen nicht synchron laufen.

Die Simulationsergebnisse, die sich für diese erste Architektur aus dem Post-PAR Timing Report ergeben, sind in Abbildung 13 zusammengefasst.

Aus dem Post-PAR Timing Report geht zudem hervor, dass für $n = 7$ die größte Verletzung der Pfadvorgabe im Demodulator vorliegt. Außerdem ist

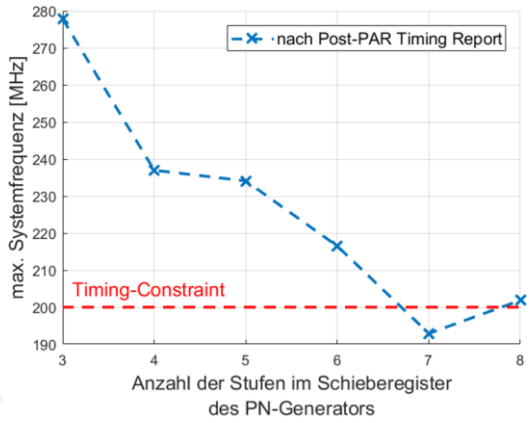


Abbildung 13. Mögliche Systemfrequenzen nach dem Post-PAR Timing Report für Architektur 1.

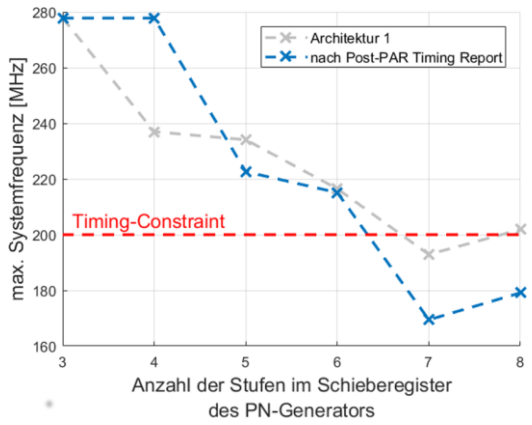


Abbildung 14. Mögliche Systemfrequenzen nach dem Post-PAR Timing Report für Architektur 2 in Gegenüberstellung zu Architektur 1.

der Prozess zur Erzeugung des *pn_begin*-Flags im PN-Generator eine der Schwachstellen für kleine Folgenlängen.

E. Architektur 2: Verbesserter PN-Generator und paralleler Demodulator

In einer zweiten Architektur wurde versucht die Engstellen, die sich nach Analyse des Post-PAR Timing Reports für Architektur 1 ergeben haben, durch andere VHDL-Beschreibungen zu umgehen. Als erste Maßnahme wurde dabei der Prozess im PN-Generator, zur Erzeugung des *pn_begin*-Flags, durch einen Zähler ersetzt. Zusätzlich wurde der einzige Prozess im Demodulator in drei Prozesse aufgeteilt, die parallel synchronisieren, akkumulieren und die Datenentscheidung treffen. Die Prozesse kommunizieren untereinander mithilfe von Steuersignalen. Die Simulationsergebnisse dieser Änderungen sind in Abbildung 14 dargestellt.

Der Bereich, in dem die Pfadvorgabe nicht erfüllt werden kann, hat sich gegenüber der ersten Architekturvariante vergrößert. Für beide Verletzungen des

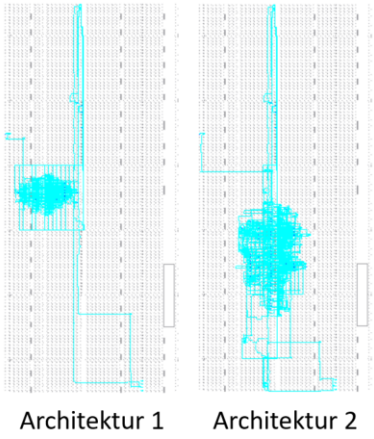


Abbildung 15. Gegenüberstellung des Routings für beide Architekturvarianten.

Timing-Constraints in Abbildung 14 ist nach dem Post-PAR Timing Report der Demodulator verantwortlich.

Die neue Demodulator-Architektur bringt folglich trotz Parallelisierung gegenüber der ersten Architektur eine Verschlechterung für große Folgenlängen mit sich. Mit der Änderung im PN-Generator hingegen lässt sich für $n = 4$ ($L = 15$) die Systemfrequenz deutlich steigern, was für $n = 5$ und 6 allerdings in etwas niedrigeren Frequenzen resultiert.

Aufgrund des neu hinzugekommenen Zählers steigt bei dieser Architektur der Verdrahtungsaufwand, was aus Abbildung 15 hervorgeht, in der das Routing für beide Architekturen für $L = 255$ gegenübergestellt ist.

F. Architektur 3: Einsatz des DSP48E-Slices im Demodulator

Mit dem DSP48E-Slice können Rechenoperationen, die für die digitale Signalverarbeitung besonders relevant sind, hardwareoptimiert durchgeführt werden. Für den Demodulator sind dabei folgende Funktionen des Slices interessant: Multiplizieren und Akkumulieren (MACC), sowie Vergleichen des Akkumulationsergebnisses auf einen fest eingestellten Wert (PATTERNDETECT) [6]. Der DSP48E-Slice soll dabei so eingesetzt werden, dass so wenig wie möglich kombinatorische und sequentielle Schaltungen außerhalb des Slices notwendig sind. Die verwendeten Komponenten des DSP48E-Slices sind in Abbildung 16 dargestellt.

Gegenüber den ersten beiden Architekturen kann mit dem neuen Demodulator für alle Folgenlängen das Timing-Constraint erfüllt werden (insbesondere für große n, siehe Abbildung 17). Anhand einer Post-PAR Simulation wurde für diese Architektur ermittelt, bei welcher Frequenz die Demodulation noch erfolgreich funktioniert. Dies ist in Abbildung 17 anhand des roten Kurvenverlaufes. Für $n = 8$ wird die, in der freien Version des ISE Webpack zulässige, Schaltungsgröße überschritten, was zu einer deutlichen Reduzierung der Simulationsgeschwindigkeit führt. Aus diesem Grund

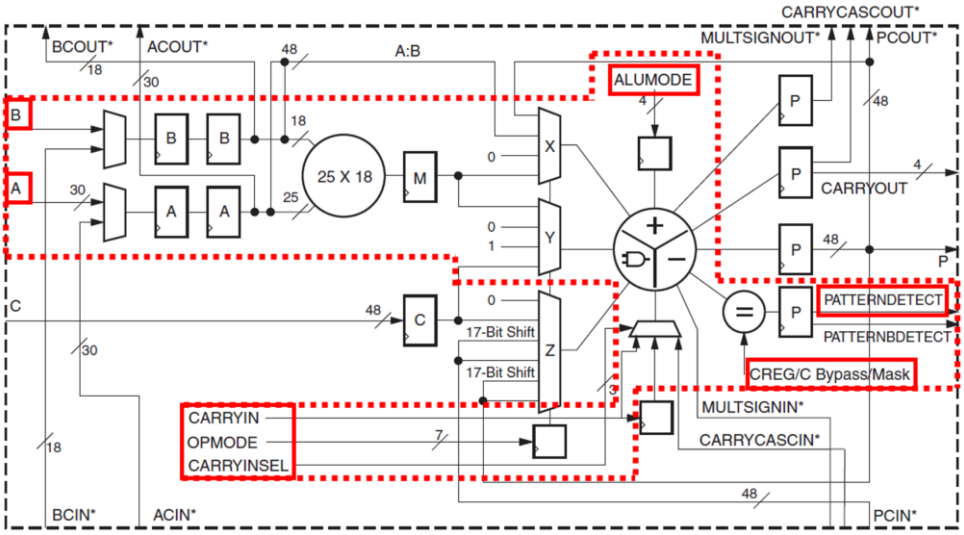


Abbildung 16. Verwendete Komponenten des DSP48E-Slices im Demodulator.

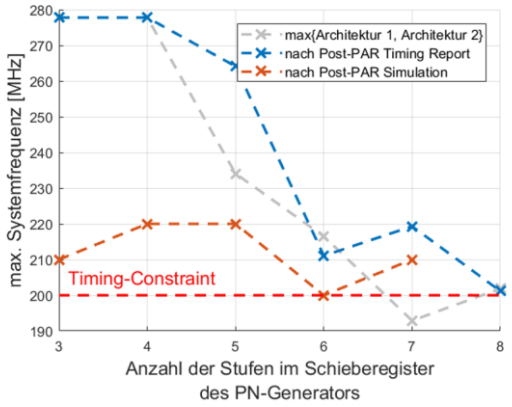


Abbildung 17. Mögliche Systemfrequenzen nach dem Post-PAR Timing Report für Architektur 3 in Gegenüberstellung zu den vorangegangenen Architekturen.

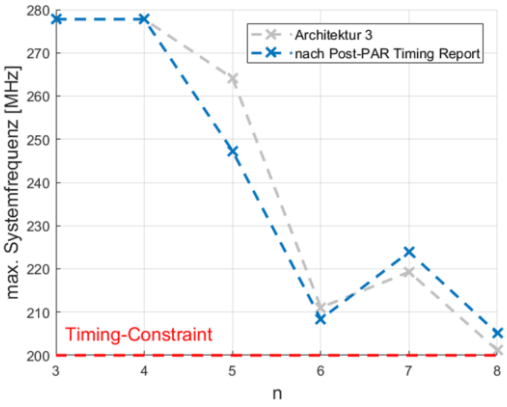


Abbildung 18. Mögliche Systemfrequenzen nach dem Post-PAR Timing Report für Architektur 4.

Tabelle II
THEORETISCH MÖGLICHE DATENRATEN MIT ARCHITEKTUR 3.

L	Max. Systemfrequenz (nach Post-PAR Simulation)	Mögliche Datenrate
7	210 MHz	30,0 MBit/s
15	220 MHz	14,7 MBit/s
31	220 MHz	7,1 MBit/s
63	200 MHz	3,2 MBit/s
127	210 MHz	1,7 MBit/s

wurde die Post-PAR Simulation für diese Folgenlänge nicht durchgeführt. Aus den Ergebnissen der Post-PAR Simulation lassen sich zusätzlich die theoretisch möglichen Datenraten ermitteln (siehe Tabelle II).

Wie bei allen Architekturen zu erkennen, ergibt sich nach dem Post-PAR Timing Report eine maximale Systemfrequenz für kleine und große Folgenlängen. Diese stellen den Flaschenhals der Schaltung dar. Aus dem Post-PAR Timing Report geht hervor, dass die Systemfrequenz für kleine und große n durch das Schieberegister im PN-Generator und in den Verzögerungsmodulen begrenzt ist.

G. Architektur 4: PN-Speicher

Um die Anzahl der Schieberegister im FPGA zu reduzieren, wurde in einer vierten Architektur die Verwendung eines PN-Speichers untersucht.

Die m-Folgen werden dabei als Distributed RAM im FPGA gespeichert und zyklisch ausgegeben. Gegenüber dem PN-Generator haben sich dabei geringfügige Verbesserungen für große Folgenlängen ergeben (siehe Abbildung 18).

VI. MESSUNGEN AM REALEN SYSTEM

Der entworfene Transceiver soll in späteren Projekten in einem Mehrbenutzersystem für eine Funkübertragung verwendet werden. Aus diesem Grund

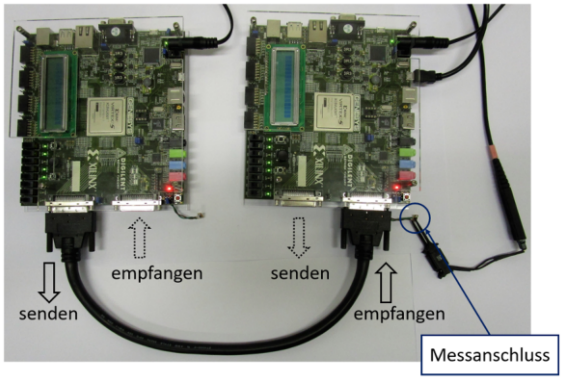


Abbildung 19. Laboraufbau zur Datenübertragung zwischen zwei Boards mit dem entworfenen Transceiver.

Tabelle III
ERGEBNISSE DER MESSUNG AM REALEN SYSTEM BEI VERWENDUNG DES TRANSCEIVERS ZUR ÜBERTRAGUNG ZWISCHEN ZWEI BOARDS.

L	m-Folgen (Erzeuger)	Demo-dulator	Chip-frequenz	Mögliche Datenrate
7	PN-Generator (mit Zähler)	DSP48E	200 MHz	28,5 Mb/s
15	PN-Generator (mit Zähler)	DSP48E	200 MHz	13,3 Mb/s
31	PN-Generator (mit Zähler)	DSP48E	200 MHz	6,4 Mb/s
63	PN-Generator (mit Zähler)	DSP48E	200 MHz	3,1 Mb/s
127	PN-Speicher	DSP48E	190 MHz	1,4 Mb/s
255	PN-Speicher	DSP48E	190 MHz	745 kb/s

wurde hier als erster Schritt dieses Projektes die leitungsgebundene Übertragung zwischen zwei Boards getestet. Bei der Übertragung wird das modulierte Signal des Senders und der Chiptakt an den Empfänger übermittelt. Die Übertragung erfolgt differentiell über ein VHDCI-Kabel, das an die VHDC-Steckverbinder des FPGA-Boards angeschlossen wird (siehe Abbildung 19).

Die entworfenen Architekturen wurden bei unterschiedlichen Chipfrequenzen untersucht. Da der Demodulator für eine fehlerfreie Übertragung konfiguriert ist, kann über das Ausgangssignal des Demodulators festgestellt werden, ob die Übertragungsstrecke für eine gegebene Frequenz funktioniert. Ist im Ausgangssignal das in der Datenquelle gespeicherte Bitmuster erkennbar, so hat die Übertragung und Demodulation erfolgreich funktioniert.

Bei den Messungen wurden experimentell die Architekturen ermittelt, für die die höchst möglichen Taktfrequenzen möglich sind. Tabelle 3 fasst die Ergebnisse dieser Untersuchung zusammen.

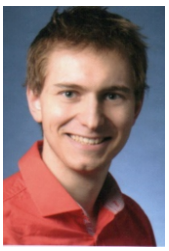
VII. ZUSAMMENFASSUNG UND AUSBLICK

Im Sinne einer Entwurfsraumexploration sind für einen CDMA-Transceiver unterschiedliche Hardwareressourcen eines Virtex 5 FPGAs zum Einsatz gekommen und untersucht worden. Dabei wurde festgestellt, dass sich für kurze Folgenlängen der PN-Generator (LUTs und D-FFs) und für lange Folgen der PN-Speicher (Distributed RAM) am besten eignet. Für die Rechenoperationen im Demodulator ließen sich mit dem DSP48E-Slice die besten Ergebnisse erzielen.

Mit dem Ziel den Transceiver für eine Funkübertragung einzusetzen sollte im Empfänger als nächstes eine Schaltung zur Taktrückgewinnung aus dem übertragenen Signal implementiert werden. Außerdem wäre es denkbar in weiteren Projekten für den Sender- und Empfänger-Port einsteckbare Funkmodule zu entwerfen, über die die modulierten Daten gesendet, bzw. empfangen werden können.

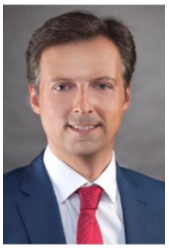
LITERATURVERZEICHNIS

[1] L. Nothhelfer, *Entwurf und FPGA-Implementierung eines CDMA-Transceivers*, Hochschule Ulm, Ulm 2017.
[2] A. M.J. Goiser, *Handbuch der Spread-Spectrum Technik*, Springer, Wien 1998.
[3] J. Meel, *Spread Spectrum (SS), Introduction*, De Nayer Instituut, Sint-Katelijne-Waver, Okt. 1999.
[4] F. Kesel, R. Bartholomä, *Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs*, Oldenbourg Wissenschaftsverlag GmbH, München 2013.
[5] Xilinx, *Virtex 5 FPGA User Guide UG190*, Xilinx Inc, März 2012.
[6] Digilent, *Digilent Genesys Reference Manual*, Digilent Inc., Washington 2016.
[7] Xilinx, *Virtex-5 FPGA XtremeDSP Design Considerations User Guide*, Xilinx Inc., Juli 2017.



Lukas Nothhelfer studierte im Bachelorstudiengang Fahrzeugelektronik an der Hochschule Ulm. In seiner Bachelorarbeit bearbeitete er die Thematik „Entwurf und FPGA Implementierung eines CDMA-Transceivers“. Sein Studium führt er an der Universität Würzburg im Bachelorstudiengang Informatik fort.

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 promivierte 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.

Neuronale Netze für die Anomalieerkennung in der Automatisierungstechnik auf feldprogrammierbaren Logikbausteinen (FPGAs)

Patrick Krawczyk, Heinz-Peter Bürkle

***Zusammenfassung*—Neuronale Netze werden im Bereich des Machine Learnings heute häufig eingesetzt. In diesem Beitrag wird die Entwicklung eines neuronalen Netzes auf einem FPGA zur Erkennung von Anomalien in binär digitalen Signalen einer speicherprogrammierbaren Steuerung beschrieben und hinsichtlich Performance und Ressourcen betrachtet. Für künstliche neuronale Netze existieren keine „One-size-fits-all“ Lösungen. Aus diesem Grund wurden verschiedene Modelle erstellt und auf ihre Eignung hin untersucht. Das finale Modell wurde samt Klassifizierung auf dem FPGA Cyclone V 5CSEMA4U23C6 implementiert. Das FPGA ist in der Lage Eingaben durch das implementierte neuronale Netz zu propagieren und anschließend zu klassifizieren. Das entwickelte System wurde auf die Gesamtperformance hin untersucht. Ein externes embedded Linux System (Raspberry Pi 2) fungiert als Testbench um die Klassifikationsergebnisse des FPGA-Modells zu evaluieren. Aus der Performance Evaluierung geht hervor, dass das Gesamtsystem 1369 (9 %) Adaptive Logic Module (ALMs) und 1229 (2 %) Register verbraucht. Weiterhin propagiert das neuronale Netz eine Eingabe innerhalb von 340 ns zum Ausgang. Dies entspricht einer Frequenz von 2,94 MHz. Die maximale Frequenz digitaler Signale einer SPS-Steuerung liegt typischerweise bei 1 kHz. Zudem ist das neuronale Netz auf dem FPGA etwa um einen Faktor 10.000 schneller als auf dem embedded Linux System und um etwa einen Faktor 1000 schneller als ein neuronales Netz in einer virtuellen Maschine (64Bit-Linux Mint, Intel(R) Core(TM) i5-7500 CPU @ 3,40 GHz, 2048 MB Speicher). Die Evaluierung der Klassifikation ergab eine Treffergenauigkeit von 98,52 % für den verwendeten Testdatensatz. Abschließend werden Verbesserungen des Systems erörtert.**

***Schlüsselwörter*—Neuronale Netze, Anomalieerkennung, FPGA**

I. Einleitung

Die Automatisierungstechnik hat das Ziel, dass Maschinen oder Anlagen selbstständig und weitgehend ohne Mitwirkung von Menschen betrieben werden können. Möglichst autonome Systeme werden im Zuge der Ausbreitung von Industrie 4.0 als Grundlage eines jeden Fertigungsprozesses in der industriellen Produktion von zentraler Bedeutung sein. Ein autonomes System muss auch kaum merkliche, aber potentiell bedrohliche Veränderungen im Fertigungsablauf erkennen und bewerten können: Zum einen macht die immer größer werdende Vernetzung der Systeme diese auch anfälliger gegenüber Cyber-Attacken. Computerviren dringen in Systeme ein, die Industrieanlagen und Kraftwerke steuern, und manipulieren deren Steuersignale. Zum anderen ist es auch wünschenswert Verschleiß und Alterung von Maschinenteilen zu erkennen und eine frühzeitige Wartung zu ermöglichen (Predictive Maintenance).

Patrick Krawczyk, Patrick.Krawczyk90@hs-aalen.de,
Heinz-Peter Bürkle, Heinz-Peter.Buerkle@hs-aalen.de

Daher ist es sinnvoll, Maschinelles Lernens zur Anomalieerkennung in der Qualitätskontrolle und in der Überwachung von Industrieanlagen einzusetzen [1]. In der Literatur existiert nur wenig Material bezüglich dem Einsatz neuronaler Netze für diesen Zweck. Dabei erweisen sie sich als vielversprechendes Verfahren [1]. Die Struktur solch neuronaler Netze drängt eine wirklich nebenläufige Implementierung der Algorithmen geradezu auf. Mittels FPGAs kann eine echte Parallelität der Algorithmen in Hardware umgesetzt werden, was zu einer Verbesserung der Performance führt.

II. Anomalieerkennung mittels neuronaler Netze

Basierend auf einem maschinellen Lernverfahren wird ein Modell trainiert, mit welchem das normale Verhalten eines Systems erlernt wird. Anschließend werden neue, unbekannte Daten mit dem trainierten Modell verglichen. Ist das Verhalten des Modells im probabilistischen Sinne schwer oder unwahrscheinlich zu erklären, wird dies als Anomalie betrachtet [2].

A. Autoencoder

Ein Autoencoder ist ein künstliches neuronales Netz, welches im Bereich des maschinellen Lernens für die Anomalieerkennung eingesetzt wird. Das Ziel eines Autoencoders ist die Rekonstruktion eines Eingabevektors [1, 2]. Basierend auf dem normalen Verhalten, erlernt das Modell des Autoencoders eine Identitätsfunktion des Trainingsdatensatzes. Mit der erlernten Identitätsfunktion wird es für das Modell schwierig

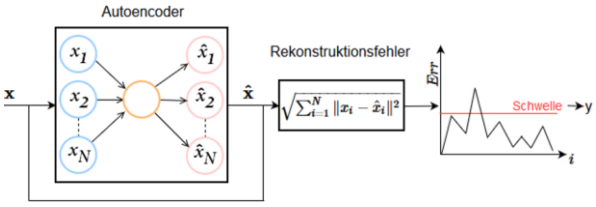


Abbildung 1. Prinzip der Anomalieerkennung mittels Autoencoder.

Anomalien zu rekonstruieren, was anhand eines Rekonstruktionsfehlers beschrieben werden kann [1]–[3]. Abbildung 1 illustriert das Prinzip der Anomalieerkennung mittels eines Autoencoders. Aus dem Eingabevektor $x = \{x_1 \ldots x_N\}$ wird der Ausgabevektor $\widehat{x} = \{\widehat{x}_1 \ldots \widehat{x}_N\}$ rekonstruiert. Hierfür gelten die allgemeinen Gleichungen:

$$a_j^{(s)} = f\left(\sum_{i=1}^{N} w_{ji}^{(s-1)} a_i^{(s-1)} + w_{j0}\right) = f\left(W^{(s-1)} a^{(s-1)}\right)$$

$$\widehat{x} = \begin{pmatrix} \widehat{x}_1 \\ \vdots \\ \widehat{x}_N \end{pmatrix} = \begin{pmatrix} a_1^{(s)} \\ \vdots \\ a_j^{(s)} \end{pmatrix}$$

Hierbei ist $a_j^{(s)}$ das Neuron j der Ausgabeschicht s, $W^{(s-1)}$ die Gewichtsmatrix der Ausgabeschicht, $a^{(s-1)}$ die Aktivierung der vorherigen Schicht und $f()$ die Aktivierungsfunktion der Ausgabeschicht. Anschließend wird der Rekonstruktionsfehler mit der Gleichung

$$\text{Err} = \sqrt{\sum_{i=1}^{N} \|x_i - \widehat{x}_i\|^2}$$

ermittelt. Überschreitet nun der Rekonstruktionsfehler einen bestimmten Schwellenwert, so wird die Eingabe als Anomalie klassifiziert ($y = 1$). Befindet sich der Rekonstruktionsfehler unterhalb des Schwellenwerts, so wird die Eingabe als normal klassifiziert ($y = 0$). Der Schwellenwert kann anhand von Erfahrungswerten gewählt werden.

B. Training

Dem Autoencoder werden Lernmuster als Eingabevektoren überreicht und die Ausgabe wird ermittelt. Der Fehler bzw. die Fehlerfunktion (engl. *error function* oder *loss function*) $J(w)$ wird minimiert, wenn die Ausgabe den gewünschten Ergebnissen entspricht [4]. Zur Minimierung des Fehlers, werden die Gewichte des Autoencoders angepasst. Die Fehlerfunktion $J(w)$ wird als Summe der quadrierten Differenzen zwischen

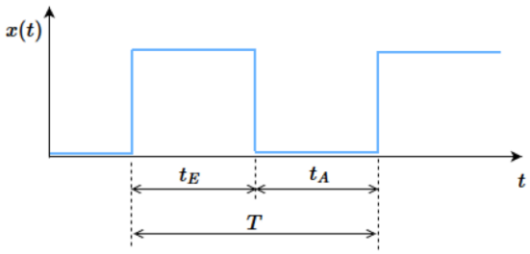


Abbildung 2. Binär digitales Signal. t_E entspricht T_ON; t_A entspricht T_OFF.

der gewünschten Ausgabe t_k und der tatsächlichen Ausgabe a_k beschrieben [4].

$$J(w) = \frac{1}{2}\sum_{k=1}^{c} \|t_k - a_k\|^2 = \frac{1}{2}\|\boldsymbol{t} - \boldsymbol{a}\|^2$$

mit $\boldsymbol{t}$ als Ziel- und $\boldsymbol{a}$ als Ausgabevektor der Länge c; $\boldsymbol{w}$ repräsentiert alle Gewichte. Die Gleichung berechnet den Fehler eines einzelnen Lernmusters. Auf einen Trainingsdatensatz $\left[\left(x^{(1)}, t^{(1)}\right), \left(x^{(2)}, t^{(2)}\right), \ldots, \left(x^{(m)}, t^{(m)}\right)\right]$ mit m Lernmustern erweitert, ergibt sich die folgende Gleichung:

$$J(W) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|t^{(i)} - a^{(i)}\right\|^2\right)$$

III. Experimentelle Ermittlung eines Autoencoder-Modells

Für künstliche neuronale Netze existieren keine „One-size-fits-all“ Lösungen. Aus diesem Grund muss das Modell für die jeweilige Aufgabe experimentell ermittelt werden. Die Tests werden auf einem PC in der Entwicklungsumgebung PyCharm Professional v2017.2 in Python 3.5 programmiert. Als Betriebssystem kommt Linux Mint v18.2 Sonya zum Einsatz. Zur Erstellung der Netzstrukturen wird die Bibliothek Keras v2.0.3 verwendet.

A. Datensatz

Die wichtigsten Signaleigenschaften binär digitaler Signale einer in der Automatisierung eingesetzten Steuerung sind die Periodendauer, Ein- und Ausschaltzeit, sowie der Tastgrad (Abbildung 2).

Die Periodendauer T ergibt sich aus der Summe der Einschaltzeit t_E und der Ausschaltzeit t_A:

$$T = t_E + t_A$$

Der Tastgrad,

$$\text{Tastgrad} = \frac{t_E}{T} = \frac{t_E}{t_E + t_A}$$

gibt die Differenz zwischen der Einschaltzeit und der Periodendauer an. Anhand der oben angegebenen Gleichungen ist erkenntlich, dass alle Parameter des binär

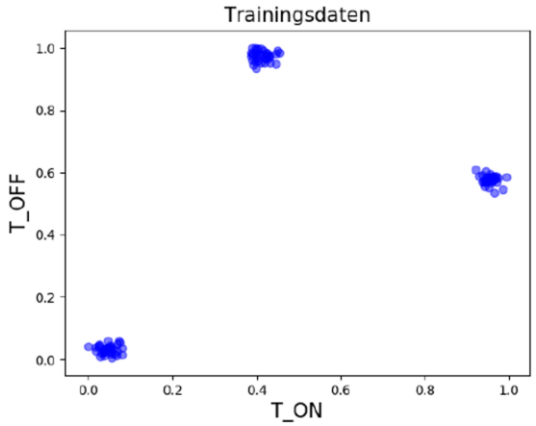


Abbildung 3. Trainingsdaten, blaue Punkte entsprechen normalen Daten.

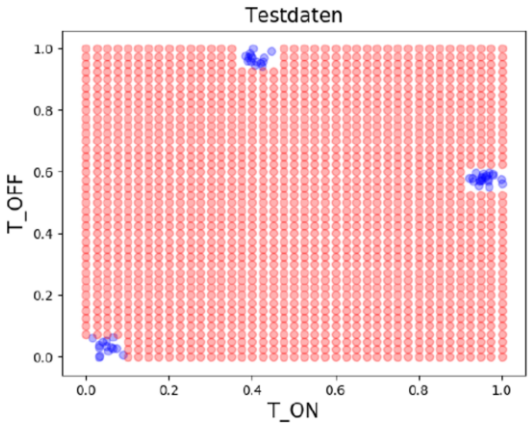


Abbildung 4. Testdatensatz, rote Punkte entsprechen Anomalien, blaue Punkte normalen Daten.

digitalen Signals, sich von der Ein- und Ausschaltzeit ableiten lassen. Somit ergibt sich ein Merkmalsvektor $\boldsymbol{x}$ von

$$x = \begin{pmatrix} \text{T\_ON} \\ \text{T\_OFF} \end{pmatrix}$$

welcher als Eingabevektor des zu ermittelnden Modells fungiert. T_ON und T_OFF sind die normalisierten Werte von t_E und t_A. Anhand des Merkmalsvektors haben wir einen Datensatz mit 1795 Einträgen generiert, wovon 150 Einträge normalen Daten und 1645 Einträge Anomalien entsprechen. Der Datensatz ist wiederum in einen Trainings- und Testdatensatz aufgeteilt. Der Trainingsdatensatz (Abbildung 3) umfasst nur normale Daten, welche 3 Cluster formen. Der Testdatensatz (Abbildung 4) weist neben normalen Daten auch Anomalien auf (rote Punkte). Die Wertigkeit der Einträge ist auf 0 bis 1 skaliert. Die x-Achse entspricht der skalierten Einschaltzeit und die y-Achse der skalierten Ausschaltzeit.

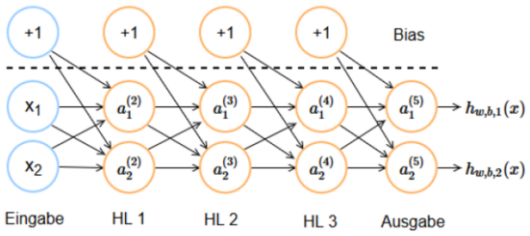


Abbildung 5. Finales Modell zur Anomalieerkennung des generierten Trainings- und Testdatensatzes.

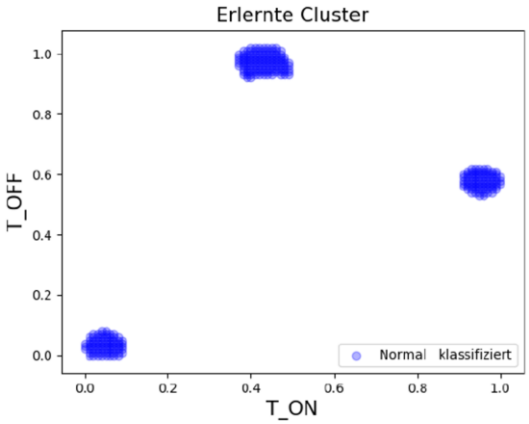


Abbildung 6. Erlernte Cluster, blaue Punkte entsprechen normalen Daten.

B. Performance Evaluierung unterschiedlicher Modelle

Für die experimentelle Ermittlung eines passenden Modells, wurden 120 verschiedene Modelle erstellt. Encoder- und Decodernetz besitzen hierbei stets eine symmetrische Anzahl an Neuronen. Die Anzahl der Neuronen pro Schicht variiert von 1 bis 15, die Anzahl der versteckten Schichten variiert zwischen 1 und 3. Als Aktivierungsfunktion der einzelnen Neuronen kommt die Sigmoidfunktion zum Einsatz. Da das Trainingergebnis der Modelle in Abhängigkeit der Zahl der Trainings schwanken kann, wird jedes Modell hier fünfmal trainiert, was hier zu einem recht stabilen Ergebnis führt. Anschließend dient der Mittelwert der „False Negative“-Werte (FN) als Kriterium zur Wahl eines finalen Modells. Tabelle I veranschaulicht einen Vergleich von drei unterschiedlichen Modellen. In Anbetracht der Mittelwerte erzielt das Modell mit drei versteckten Schichten und jeweils 2 Neuronen, das beste Ergebnis. Abbildung 5 illustriert das finale Modell. Zusätzlich bietet Abbildung 6 eine visuelle Ergänzung der vom Modell erlernten Cluster. Eingabewerte die sich in den blauen Bereichen befinden, werden als normal und Daten außerhalb davon als Anomalie klassifiziert.

IV. Systemarchitektur

Das System umfasst einen Raspberry Pi 2 und den FPGA Cyclone V 5CSEMA4U23C6 (Abbildung 7).

Tabelle I
MITTELWERTE DER „FALSE NEGATIVE“.

Modell	Mittelwert FN
Ein Hidden Layer mit 2 Neuronen	86,2
Hidden Layer 1 und 3 mit 2 Neuronen, Hidden Layer 2 mit 1 Neuron	27,4
Drei HiddenLayer mit jeweils 2 Neuronen	10,8

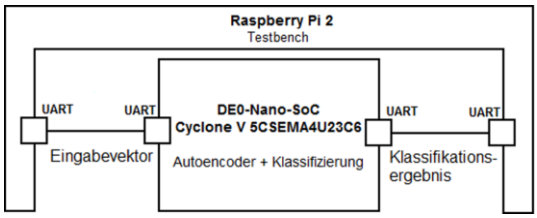


Abbildung 7. Systemarchitektur.

Der Raspberry Pi 2 fungiert als Testbench, um die Klassifikationsergebnisse des auf dem FPGA implementierten Modells zu evaluieren. Hierfür trainiert der Raspberry Pi ein eigenes Modell und sendet anschließend die trainierten Gewichte per UART-Schnittstelle zum FPGA. Während einer Initialisierungsphase speichert das FPGA die trainierten Gewichte ab. Anschließend ist dieser betriebsbereit und kann Eingangssignale durch das implementierte Netz propagieren und klassifizieren. Der Raspberry Pi sendet nun die Testdaten zum FPGA und evaluiert im Anschluss die Klassifikationsergebnisse.

Der Einsatz eines externen embedded Linux Systems kann in einem realen Einsatzfall zur Feinjustierung des Modells verwendet werden. Sollten zum Beispiel neue Sensoren der Maschine hinzugefügt werden oder alte Sensoren erneuert werden müssen, so reicht es aus, nur die Gewichte des Modells neu zu ermitteln. Die neuen Daten müssten in das System eingespeist werden, woraufhin sich das System autonom an die neuen Bedingungen anpasst.

V. IMPLEMENTIERUNG

Bei der Realisierung eines neuronalen Netzes auf einem FPGA müssen Kompromisse eingegangen werden. Ein wesentlicher Kompromiss liegt zwischen der Fläche und der Präzision der Implementierung. Hier liegt das Problem darin, eine Balance zwischen numerischer Präzision und den Verbrauch der FPGA-Ressourcen zu finden. Die einfache Genauigkeit der Gleitkommaarithmetik (engl. *single precisions floating-point*) ist eine ideale numerische Repräsentation, da sie aufgrund eines geringen Quantisierungsfehlers die größte Präzision liefert. Zusätzlich würde sie der Repräsentation auf dem Raspberry Pi 2 entsprechen, womit eine identische Performance resultiert. Allerdings liegt der Ressourcenverbrauch einer Gleitkommaarithmetik höher als bei einer Festkomma-Arithmetik [5, 6]. In [5] wurde die numerische Präzision neuronaler Netze auf einem FPGA bezüglich der Rechenzeit und Flächenverbrauch untersucht. Dabei wurde festgestellt, dass die Festkomma-Arithmetik 12-mal schneller und eine um 13-fach kleinere Fläche verbraucht. Weiterhin werden dort 12 Bit für die Nachkommastelle empfohlen.

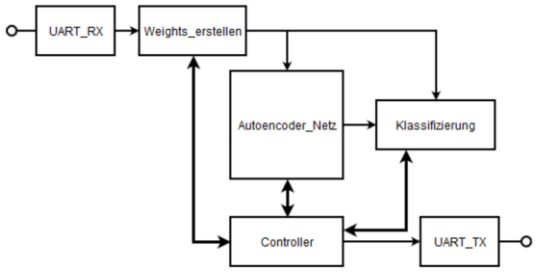


Abbildung 8. Blockdiagramm der FPGA-Implementierung.

Für diesen Beitrag haben wir uns für eine 18 Bit Festkomma-Arithmetik entschieden. Hierbei entsprechen 12 Bit der Nachkommastelle, 5 Bit der Vorkommastelle und 1 Bit dem Vorzeichen. Mit insgesamt 6 Bit für die Vorkommastelle lässt sich ein Wertebereich von $[-32; 32]$ beschreiben. Der benötigte Wertebereich und somit die benötigte Anzahl an Vorkommastellen, hängt von dem Wertebereich der trainierten Gewichte ab. Aufgrund der 12 Bit Nachkommastellen, weist die Repräsentation einen LSB-Wert von $2.44140625 \cdot 10^{-4} \triangleq 2^{-12}$ auf. Negative Werte werden mittels Zweierkomplement dargestellt.

Abbildung 8 veranschaulicht eine Übersicht aller Komponenten. Für die Übertragung der Gewichte werden 3 Byte benötigt, welche mit der Komponente „Weights_erstellen“ zusammengefügt werden.

A. Einzelnes Neuron

Das Neuron ist die grundlegende Recheneinheit in einem künstlichen neuronalen Netz. Abbildung 9 veranschaulicht die Implementierung. Für jedes Neuron müssen drei Gewichte (ein Bias-Gewicht, zwei Eingangsgewichte) gespeichert werden. Aufgrund der 18 Bit Festkomma-Arithmetik benötigt ein Neuron 54 Bit Speicher. Für acht Neuronen (ganzes Autoencoder-Netz) entspricht das 432 Bit Speicherbedarf. Aufgrund des geringen Speicherbedarfs, speichern wir die Gewichte in Registern, welche Logikzellen verbrauchen. Bei deutlich größeren Netzen bzw. bei Neuronen mit deutlich mehreren Eingängen, empfiehlt es sich, spezielle BlockRAMs zu verwenden.

Die ALU multipliziert die Eingänge mit den entsprechenden Gewichten und summiert diese auf. Um eine echte Parallelität zu realisieren, benötigt jeder Eingang eines Neurons einen Multiplizierer. Somit hängt die Parallelität von der Größe des Netzes, sowie von den zur Verfügung stehenden FPGA-Ressourcen ab. Moderne FPGAs stellen eingebettete DSP-Blöcke bereit, um für Rechenoperationen Logikzellen zu sparen. Die

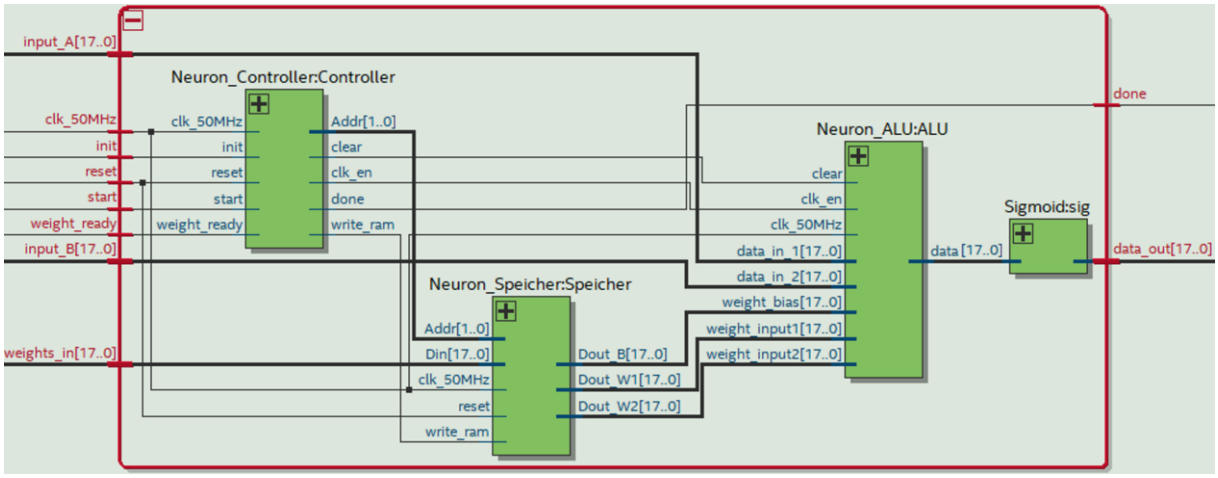


Abbildung 9. Blockdiagramm der FPGA-Implementierung eines Neurons.

Ergebnisse der Multiplikationen müssen aufsummiert werden. Hierfür bietet sich eine Baumstruktur der Addierer an. Bei Bedarf kann diese Baumstruktur mit Pipe-Lining erweitert werden. Für den Bias Eingang wird kein Multiplizierer verwendet, da dieser immer eine „1“ aufweist und sich somit eine Multiplikation erübrigt.

In jedem Neuron fungiert die Sigmoidfunktion als Aktivierungsfunktion. Eine Hardware Implementierung der Sigmoidfunktion benötigt aufgrund der e-Funktion einen hohen Ressourcenverbrauch. In [7] werden diverse Ansätze zur Hardware Implementierung einer Sigmoidfunktion getestet. Hauptaugenmerk liegt auf dem Ressourcenverbrauch, sowie der Abweichung der implementierten Funktion gegenüber der originalen Sigmoidfunktion. Zur Implementierung als Look-Up Tabelle werden keinerlei Angaben zur Abweichung angegeben. Allgemein gilt, je größer die Tabelle, desto geringer ist die Abweichung und desto größer ist der Ressourcenverbrauch. Für unser Netz haben wir uns für die PLAN Funktion (Piecewise Linear Approximation of a Nonlinear function) als Implementierungsansatz entschieden. Mit nur 44 benötigten Logikblöcken, 67 Look-Up Tabellen und einer maximalen Abweichung von 2,14 %, erweist sich dieser Ansatz als annehmbar. Ein Vorteil der PLAN-Funktion ist die Verwendung von Schiebeoperatoren anstelle von Multiplikationen.

B. Autoencoder-Netz

Das Autoencoder-Netz wird mit acht einzelnen Neuronen realisiert. Für die Initialisierungsphase ist ein DEMUX vorgesehen, welcher die Gewichte zu den einzelnen Neuronen weiterleitet. Ein Controller ist für die Steuerung der einzelnen Neuronen, sowie für den DEMUX verantwortlich.

C. Klassifizierung

Der Rekonstruktionsfehler entspricht dem euklidischen Abstand zwischen der Eingabe und der vom Autoencoder berechneten Ausgabe. Dieser wird anschließend mit einem Schwellenwert verglichen und entsprechend klassifiziert. Der euklidische Abstand berechnet sich gemäß der Gleichung

$$\sqrt{\sum_{i=1}^{N} \|x_i - \widehat{x}_i\|^2}$$

mit x_i als Eingabe und $\widehat{x}_i$ als rekonstruierte Eingabe. Für den Term

$$\sum_{i=1}^{N} \|x_i - \widehat{x}_i\|^2$$

wird das Zweikomplement der rekonstruierten Eingaben gebildet. Auf diese Weise kann die Subtraktion mit einem Addierer realisiert werden. Die Differenzen werden anschließend quadriert und zum Schluss addiert.

Anstatt nun die Wurzel aus dem oben berechneten Ergebnis zu ziehen, kann stattdessen der Schwellenwert quadriert werden. Dies erspart die Implementierung der Wurzelberechnung auf dem FPGA und führt zum selben Klassifikationsergebnis.

Der Schwellenwertvergleich wird mit einem Komparator realisiert. Weist der Rekonstruktionsfehler einen Wert kleiner gleich dem Schwellenwert auf, so wird der Ausgang auf LOW gesetzt. Ist der Rekonstruktionsfehler größer als der Schwellenwert, so wird der Ausgang auf HIGH gesetzt.

VI. Ergebnisse

A. Ressourcenverbrauch auf dem FPGA

Die in den Tabellen II bis V vorgelegten Daten des Ressourcenverbrauchs, werden dem Synthese-Report

Tabelle II
GESAMTRESSOURCENVERBRAUCH AUF DEM FPGA.

Resource	Verbrauch
ALMs	1.369 / 15.880 (9 %)
Register	1229 / 60.376 (2 %)
ALUTs	2463
BlockRAM Bits	0 / 2.764.800 (0 %
DSP Blöcke	16 / 84 (19 %)

Tabelle III
RESSOURCENVERBRAUCH EINES EINZELNEN NEURONS.

Resource	Verbrauch
Register	89 / 60.376 (0.1 %)
ALUTs	173
BlockRAM Bits	0 / 2.764.800 (0 %
DSP Blöcke	2 / 84 (2,3 %)

Tabelle IV
RESSOURCENVERBRAUCH DES AUTOENCODER-NETZES.

Resource	Verbrauch
Register	743 / 60.376 (1.2 %)
ALUTs	1533
BlockRAM Bits	0 / 2.764.800 (0 %
DSP Blöcke	16 / 84 (19 %)

Tabelle V
RESSOURCENVERBRAUCH DER KLASSIFIZIERUNG.

Resource	Verbrauch
Register	344 / 60.376 (0.5 %)
ALUTs	830
BlockRAM Bits	0 / 2.764.800 (0 %
DSP Blöcke	0 / 84 (0 %)

der Entwicklungsumgebung Quartus Prime entnommen. In Tabelle 2 ist der Gesamtressourcenverbrauch auf dem FPGA dargelegt. Aus der Tabelle geht hervor, dass das Gesamtsystem 9 % der ALMs, 2 % der Register und 19 % an DSP Blöcken verbraucht.

Der Ressourcenverbrauch eines einzelnen Neurons ist in Tabelle III dargelegt. Aufgrund der zwei verwendeten eingebetteten Multiplizierer, verbraucht ein einzelnes Neuron 2 DSP Blöcke. Dies erklärt den Gesamtverbrauch an DSP Blöcken in der Tabelle II (8 Neuronen · 2 DSP Blöcke). Tabelle IV legt den Ressourcenverbrauch des Autoencoder-Netzes dar und Tabelle V legt den Ressourcenverbrauch der Klassifizierung dar.

B. Rechenzeit

1) Laufzeiten auf dem FPGA: Zur Ermittlung der Laufzeiten wird eine RTL-Simulation in der Software ModelSim 10.5b der Firma Mentor Graphics durchgeführt.

Tabelle VI
LAUFZEITEN EINES EINZELNEN NEURONS.

Phase	Laufzeit in ns
Initialisierung	80
Betrieb	40

Tabelle VII
GESAMTZEIT, VOM AUTOENCODER-START BIS HIN ZUM KLASSIFIKATIONSERGEBNIS.

Funktion	Laufzeit in ns
Gesamt	400
Autoencoder-Netz	340
Klassifizierung	60

Die Laufzeiten eines einzelnen Neurons können der Tabelle VI entnommen werden. Es ist ersichtlich, dass ein Neuron 80 ns benötigt, um die Gewichte zu speichern. Während der Betriebsphase benötigt ein Neuron 40 ns (25 MHz), um einen Eingabevektor zu verarbeiten.

Die Zeit vom Start des Autoencoder-Netzes bis hin zum Klassifikationsergebnis kann der Tabelle VII entnommen werden. Das Autoencoder-Netz benötigt 17 Takte um eine Eingabe zum Ausgang hin zu propagieren. Diese Zeit ergibt sich aus der Rechenzeit eines einzelnen Neurons (2 Takte), der vier Schichten (3 versteckte-, 1 Ausgabeschicht), einem Start- und Ende-Takt (Start- und Done-Flag) jeder Schicht, sowie dem Start-Takt des Netzes ($2 \times 4 + 2 \times 4 + 1 = 17$).

Die Gesamtzeit beträgt 400 ns, was einer Frequenz von 2,5 MHz entspricht. Eingangsfrequenzen schneller als 2,5 MHz können dementsprechend nicht direkt verarbeitet werden, weshalb eine Zwischenpufferung höherer Frequenzen notwendig wäre. Typischerweise liegt die maximale Frequenz binär digitaler Signale einer SPS-Steuerung bei 1 kHz – sie ist also wesentlich kleiner.

2) FPGA vs. PC(VM) vs. Pi: Des Weiteren ist ein zeitlicher Vergleich der Netzgeschwindigkeit auf unterschiedlichen Plattformen interessant. Zur Verfügung steht neben dem FPGA, der Raspberry Pi 2, sowie eine virtuelle Maschine mit 64 Bit-Linux Mint 18.2 Sonya als Betriebssystem, einem Intel(R) Core(TM) i5-7500 CPU @ 3,40 GHz Prozessor und 2048 MB Speicher. Die Netzgeschwindigkeit entspricht der Zeit, die das Autoencoder-Modell benötigt, um eine Eingabe zum Ausgang hin zu propagieren.

Die Netzgeschwindigkeit des FPGAs wird durch eine RTL-Simulation ermittelt. Sowohl auf dem Raspberry Pi 2, als auch in der virtuellen Maschine, wird die Netzgeschwindigkeit nach 10 Durchläufen mit 1000 Eingaben gemittelt. Durch die 1000 Eingaben sollen die betriebssystembedingten Schwankungen der Laufzeiten herausgemittelt werden. Um zu vermeiden das bestimmte Caching-Mechanismen der Betriebssysteme die Zeiten verfälschen, werden die Eingaben jedes Mal

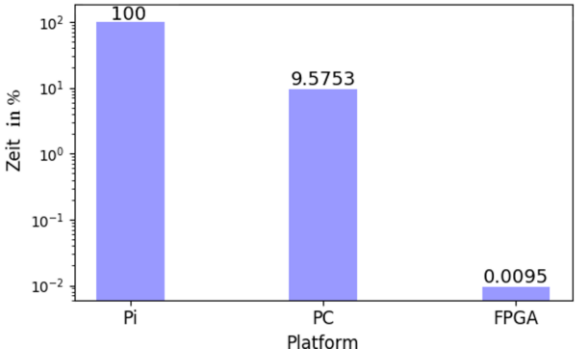


Abbildung 10. Zeitvergleich der Netzlaufzeit zwischen den unterschiedlichen Plattformen (Pi, PC(VM), FGPA).

zufällig neu gewählt.

Des Weiteren ist wichtig zu erwähnen, dass das Autoencoder-Netz in Python mit der Bibliothek Keras realisiert wurde. Jegliche Netzfunktionen werden durch Keras ausgeführt, weshalb an dieser Stelle keine nachträglichen Performanceverbesserungen erzielt werden können. Zusätzlich ist Python eine Interpretersprache, welche gegenüber einer Maschinensprache eine langsamere Ausführungsgeschwindigkeit aufweist.

Der Raspberry Pi 2 benötigt im Schnitt 3,579 ms, die virtuelle Maschine 342,7 us und das FPGA 340 ns. Abbildung 10 veranschaulicht die Zeiten anhand eines Balkendiagramms. Das FPGA ist etwa um einen Faktor 10.000 schneller als der Raspberry Pi 2 und um etwa einen Faktor 1000 schneller als die virtuelle Maschine.

VII. Klassifikation auf dem FPGA

Die Performance wird anhand des Testdatensatzes ermittelt. Die Testdaten werden durch das implementierte Modell propagiert und anschließend klassifiziert. Die Klassifikationsergebnisse werden in Form einer Wahrheitsmatrix angegeben (Abbildung 11). Das Modell erzielt eine Treffergenauigkeit von 98,52 %. Qualitativ betrachtet liefert das FPGA Modell ein brauchbares Ergebnis, welches in der prädiktiven Wartung von SPS-Anlagen durchaus seine Anwendung finden kann.

VIII. Fazit

Wir haben gezeigt, dass neuronale Netze für die Anomalieerkennung durchaus ihre Anwendung im Bereich der Automatisierungstechnik finden können. Das Modell erzielt mit 98,52 % eine sehr gute Treffergenauigkeit des Testdatensatzes. Das auf dem FPGA implementierte Modell arbeitet im Vergleich zu den relativ langsamen Steuersignalen (1 kHz) zur Zeit zu schnell (2.5 MHz) und weist einen für die Modellgröße relativ hohen Ressourcenverbrauch auf.

In kommenden Projekten kann auf eine rein sequentielle Implementierung gewechselt werden (Abbildung 12). Diese weist nur noch eine einzige ALU, mit 2 Addierern und 2 Multiplizieren auf. Die Gewichte

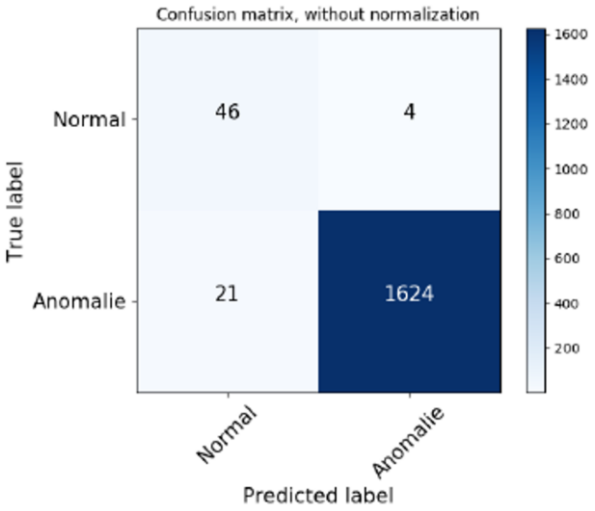


Abbildung 11. Wahrheitsmatrix des Testdatensatzes.

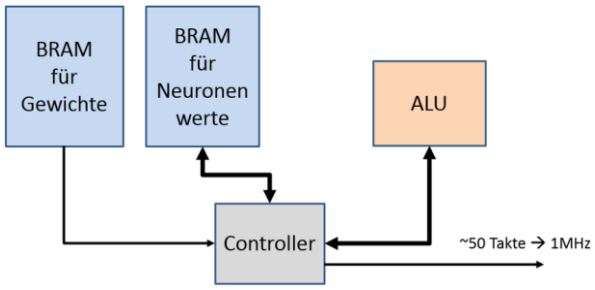


Abbildung 12. Neuronale Netzstruktur als sequentieller Ansatz.

und die Ausgabewerte der einzelnen Neuronen werden in BRAMs gespeichert. Diese Implementierung reduziert deutlich den Ressourcenverbrauch auf dem FPGA und benötigt hochgerechnet etwa 50 Takte (1 MHz) um eine Eingabe durch das Netz zu propagieren und zu klassifizieren. Eine weitere Möglichkeit um das System in Zukunft zu erweitern ist den Raspberry Pi 2 durch das embedded Linux eines SoC-Chips zu ersetzen. Dadurch könnte das System als „stand-alone“ fungieren – Dies führt zu geringeren Kosten und einem reduzierten Bauraumbedarf.

LITERATURVERZEICHNIS

[1] Prof. Dr. Wahlster, Prof. Dr. Kagermann, „Maschinelles Lernen“, *mooc.House* von Hasso-Plattner-Instituts, Online-Kurs 2017.

[2] T. Yairi M. Sakurada, „Anomaly detection using autoencoders with nonlinear dimensionality reduction“, *MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 2014.

[3] V. M. Janakiraman, D. Nielsen, „Anomaly detection in aviation data using extreme learning machines“, *International Joint Conference on Neural Netzworks (IJCNN)*, 2016.

[4] D. G. Stork, R. O. Duda, P. E. Hart, *Pattern Classification*, Second Edition Wiley & Sons Verlag, 2001.

[5] Kristian Moussa, Shawki Areibi, „On the arithmetic precision for implementing back-propagation networks on FPGA: A case study“, *FPGA Implementations of Neural Networks, pp.37-61*, 2006.

[6] José M. Ortega-Zamorano, „Efficient implementation of the backpropagation algorithm in FPGAs and microcontrollers“, *IEEE Transactions on Neural Networks and Learning Systems*, 2015.

[7] D. Mic, A. Tisan, S. Oniga, „Digital implementation of the sigmoid function for FPGA circuits“, Electronic and Computer Department, North University of Baia Mare, 2009.

Patrick Krawczyk erhielt den akademischen Grad des Master of Science von der Hochschule Aalen 2017. Aktuell arbeitet er als wissenschaftlicher Mitarbeiter im Bereich autonomer Systeme an der Hochschule Aalen.

Heinz-Peter Bürkle bekam den akademischen Grad Dipl.-Ing. in Elektrotechnik im Jahr 1991 von der Universität Stuttgart und den Grad Dr.-Ing. 1997 ebenfalls von der Universität Stuttgart. Nach einigen Jahren in Forschung und Entwicklung bei Alcatel ist er seit 2003 Professor für Schaltungstechnik und rechnergestützten Schaltkreisentwurf an der Hochschule Aalen.

Hochschule
Albstadt-Sigmaringen
Albstadt-Sigmaringen University

Multikriterien-Optimierung energietechnischer Komponenten unter Anwendung von Methoden der Künstlichen Intelligenz

Johannes Mast, Stefan Rädle, Joachim Gerlach

***Zusammenfassung*—Die vorliegende Arbeit untersucht die Übertragung und Applizierung von algorithmischen Lösungsansätzen aus dem Bereich der Künstlichen Intelligenz und des Maschinellen Lernens in den Bereich des Systementwurfs. Das betrachtete Anwendungsbeispiel entstammt dem Bereich energietechnischer Systeme. Aufsetzend auf eine im Rahmen von Vorgängerarbeiten [1] geschaffene skalierbare Methodik und Werkzeugumgebung wird ein KI-Ansatz für eine systematische Optimierung der Steuerung von Systemkomponenten und -szenarien hinsichtlich variierender Zielkriterien eingesetzt – im vorliegenden Fall hinsichtlich der optimierten Steuerung eines Blockheizkraftwerk-Szenarios (BHKW). Innerhalb einer experimentellen Untersuchung wird aufgezeigt, wie und in welchem Umfang innerhalb eines KI-basierten Lösungsverfahrens die Berücksichtigung einer komplexen Nebenbedingung – im vorliegenden Fall die Charakteristika der Strombörse („börsenoptimiertes BHKW“) – Vorteile gegenüber der konventionellen („wärmegeführten“) Steuerungsstrategie zu verschaffen in der Lage ist.**

***Schlüsselwörter*—Energietechnische Systeme, BHKW, Börsenoptimierung, Künstliche Intelligenz, Simulated Annealing**

I. Einleitung

Wie verschiedene Anwendungsbereiche eindrucksvoll belegen, liefern Ansätze aus dem Bereich der Künstlichen Intelligenz (KI) und des Maschinellen Lernens (ML) leistungsfähige Lösungsverfahren für Problemstellungen, die durch eine hohe Komplexität, Heterogenität, Dynamik sowie stark variierende Randbedingungen charakterisiert sind. Klassische Beispiele hierfür liefern etwa Anwendungen wie Siri, Google-Now oder Google-Translate, die – basierend auf Neuronalen Netzen – etablierte Systemlösungen im Bereich der automatisierten Sprachanalyse, -erkennung und -übersetzung liefern.

Da beim Entwurf komplexer Systeme ähnliche Voraussetzungen und Randbedingungen gelten, ist der Versuch, KI-Ansätze im Bereich des Systementwurfs anwendbar zu machen und dort zur Beantwortung relevanter Fragestellungen einzusetzen, naheliegend. Dies soll im Folgenden am Beispiel energietechnischer Systeme im Kontext komplexer Randbedingungen, die sich im Rahmen der Energiewende beim Übergang zu erneuerbaren Energien ergeben, untersucht und aufgezeigt werden.

Eine der Schlüsseltechnologien der Energiewende stellen BHKWs dar, die beim Betrieb sowohl Wärme als auch Strom erzeugen. In der Regel werden sie nach dem Wärmebedarf betrieben. Wenn allerdings der Wärmebedarf gedeckt werden soll, bei gleichzeitiger Berücksichtigung der Preise an der Strombörse, entsteht ein hochkomplexes Optimierungsproblem, bei dem deutliche Parallelen zu Anwendungsgebieten erkenntlich sind, in denen sich KI bereits als ertragreich erwiesen hat. Aus diesem Grund wird in diesem Beitrag untersucht, auf welche Weise die Problemstellung in eine für die Anwendung von KI-Ansätzen taugliche Darstellung überführt werden kann und auf welche Weise geeignete Iterationsverfahren und Metriken für eine zielgerichtete Traversierung des Explorationsraums formuliert werden können.

Der folgende Beitrag ist wie folgt gegliedert: II beschreibt als Grundlagen die Besonderheiten von Blockheizkraftwerken sowie den für die Optimierung gewählten KI-Ansatz. In III wird erläutert, wie die Problemstellung auf den KI-Ansatz abgebildet werden kann, so dass ein auf die Strombörse optimiertes Steuersignal für den Folgetag liefert. IV vergleicht die Resultate eines strombörsenoptimierten BHKWs mit einem nicht optimierten BHKW. Abschließend werden im Abschnitt Fazit die Ergebnisse zusammengefasst und es wird ein Überblick über die Einsatzmöglichkeiten der beschriebenen Methodik gegeben.

II. Grundlagen

In diesem Kapitel werden die im weiteren Verlauf benötigten Kenntnisse erläutert. Zu diesem Zweck werden die Eigenschaften eines BHKWs, die intelligente Suchmethodik „Simulated Annealing“, sowie spezifische Randbedingungen der Domäne Energietechnik erläutert.

A. Werkzeugumgebung

Im Rahmen des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Projekts SADE („Simulative Analyse dezentraler Energieversor-

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

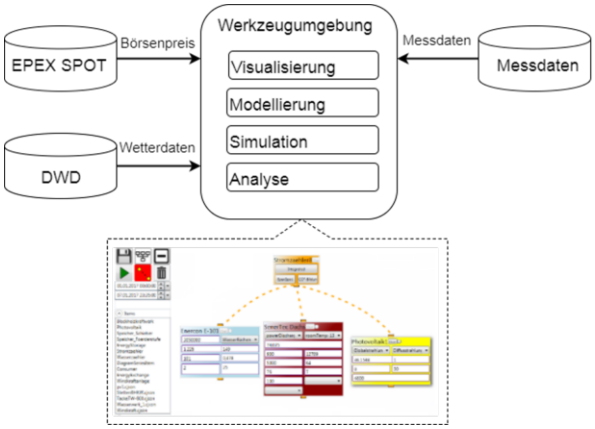


Abbildung 1. Werkzeugumgebung.

gungskonzepte“) ist eine Modellierungs- und Simulationsumgebung entstanden. Wie in Abbildung 1 dargestellt, sind Datenquellen des Servers der Strombörse (EPEX SPOT), dem Server des Deutschen Wetterdienstes (DWD) und weiteren Servern mit Messdaten an die Werkzeugumgebung angebunden.

Durch diese ist es möglich, in der Werkzeugumgebung modellierte, energietechnische Systemszenarien auf bestimmte Standorte und Szenarien anzupassen. Als energietechnische Komponenten sind Simulationsmodelle einer Photovoltaikanlage, einer Windenergieanlage und eines Blockheizkraftwerks in die Umgebung eingebunden. Damit ermöglicht es die Umgebung Daten für den KI-Ansatz zu erstellen und Systemszenarien zu bewerten. [3]

B. Blockheizkraftwerk

Ein BHKW besteht aus einem Verbrennungsmotor der zur Stromerzeugung an einen Generator angeschlossen ist. Neben dem Verbrennungsmotor und dem Generator besitzt ein BHKW zusätzlich häufig einen Pufferspeicher, bei dem es sich um einen Wärmespeicher handelt. Wenn mehr Wärme erzeugt wird, als zum aktuellen Zeitpunkt benötigt, kann der Wärmeüberschuss im Pufferspeicher gelagert werden, bis die Wärme benötigt wird. Um den Verschleiß eines BHKWs gering zu halten, wird es typischerweise immer mit einer Nennleistung betrieben. Somit hat ein BHKW zwei Zustände: Ausgeschaltet und Eingeschaltet (auf Nennleistung). Die am häufigsten verwendete Betriebsweise richtet die Steuerung des BHKWs auf die Wärmeanforderungen des angeschlossenen Verbrauchers aus. Sie wird als wärmegeführte Betriebsweise bezeichnet.

C. Gesetzeslage für KWK-Anlagen

BHKWs stellen eine zunehmend verbreitete energietechnische Komponente, basierend auf der Kraft-Wärme-Kopplung (KWK) dar. Betreiber von KWK-Anlagen können nach dem KWK-Gesetz eine von der elektrischen Leistung abhängige Förderung der Anlage vom Bund erhalten, wodurch sie für jede erzeugte Kilowattstunde (im Folgenden: kWh) eine Zuschlagszahlung erhalten. Bislang hatten alle vom KWK-Gesetz geförderten KWK-Anlagen für ihren erzeugten Strom eine Abnahmegarantie durch den Netzbetreiber, der dazu verpflichtet war, unabhängig von der Netzlast oder dem Börsenstrompreis den Strom zum üblichen Marktpreis mit zusätzlichem KWK-Zuschlag abzunehmen. Das hat dazu geführt, dass ein Großteil der heute existierenden BHKWs die Steuerung des Motors nach dem Wärmebedarf ausgelegen. Der nebenbei erzeugte Strom, wird entweder für den Eigenbedarf genutzt oder zu den beschriebenen, fixen Konditionen in das öffentliche Netz eingespeist. Mit der Neufassung des KWK-Gesetzes von 2016 werden alle Neuanlagen und alle seit Beginn modernisierten KWK-Anlagen, die eine Leistungsgrenze überschreiten, zur Direktvermarktung ihres erzeugten Stroms verpflichtet [1]. Zudem wird im Gesetz festgehalten, dass keine Zuschlagszahlungen in Stunden mit Börsenstrompreisen kleiner oder gleich Null erfolgen. Diese Gesetzesneufassung führt dazu, dass wärmegeführte BHKWs für den Betreiber nicht mehr so rentabel sind wie zuvor. Denn bei der wärmegeführten Betriebsweise wird Strom nahezu durchgehend erzeugt, und zwar auch zu Zeiten, in denen der Strompreis an der Strombörse aufgrund eines Überangebot nicht rentabel ist oder sich gar im Negativen befindet. Dahingegen können energietechnische Komponenten, die als weiteren Parameter die Strompreise berücksichtigen, besser wirtschaften und zusätzlich dazu beitragen den Energiehaushalt des Stromnetzes zu stabilisieren.

D. Bedingungen für einen maximierten Profit

Im Wartungsvertrag eines BHKWs werden meist Zeitbedingungen für einen verschleißarmen Betrieb angeführt. Darunter fällt meist eine Mindestlaufzeit und -kühlzeit. Unabhängig vom Wartungsvertrag sollten beim Umstieg bzw. der Neuinstallation eines börsenoptimierten BHKWs mehrere Faktoren gegeben sein, um den Gewinn gegenüber der wärmegeführten Betriebsweise zu maximieren. Diese werden bei der Implementierung im Abschnitt Methode und Implementierung miteinfließen und werden deshalb in den folgenden Abschnitten erläutert.

1) Kessel für Gewinnoptimierung: Der Wirkungsgrad eines BHKWs ist in der Regel höher als der eines Kessels, weil anders als beim Kessel die Wärme nur ein Nebenprodukt des stromerzeugenden Generators ist. Allerdings ist die Wärmeerzeugung eines Kessels bei gleichem Einsatz von Brennstoff größer die eines BHKWs [4]. Daher rentiert sich die Inbetriebnahme eines BHKWs gegenüber einem Kessel nur, solange der Gewinn durch den erzeugten Strom die zusätzlichen Brennstoffkosten übersteigt. Wenn ein zusätzlicher Kessel vorhanden ist, kann die börsenoptimierte

Steuerung Brennstoffkosten einsparen, indem sie das BHKW bei niedrigen Börsenpreisen trotz Wärmebedarf ausgeschaltet lässt. Zu diesen Zeiten deckt der Kessel den Wärmebedarf alleine ab. Vorzugsweise sollte der Kessel im Heizkreislauf nicht an denselben Pufferspeicher wie das BHKW installiert werden. Ansonsten ist die Temperatur im Rücklauf des Pufferspeichers tendenziell höher und das BHKW muss öfters Kühlpausen einlegen, was die Betriebsstunden des BHKWs einschränkt. Als *Grenzkosten* wird der Preis für elektrische Energie (z.B. 1 kW) bezeichnet, ab dem die Inbetriebnahme des BHKWs profitabler ist als die des Kessels. Abhängig vom spezifischen Fall unterscheiden sich die Grenzkosten, und müssen vom Betreiber des BHKWs im Vorfeld berechnet werden.

2) Mindesttemperatur aufrechterhalten: Bei der börsenoptimierten Steuerung gibt es Situationen, in denen es zu längeren Ausschaltzeiten des BHKWs aufgrund von niedrigen Preisen an der Börse kommt. In diesen Zeiten würde sich die Temperatur des Wassers im Pufferspeicher langsam auf die Umgebungstemperatur abkühlen. Bei der wärmegeführten Betriebsweise tritt diese Situation nie ein, weil das BHKWs seine Schaltzyklen nach vorgegebenen Temperaturgrenzen im Pufferspeicher richtet. Im börsenoptimierten Betrieb führt dieser Umstand allerdings dazu, dass der Kessel und das BHKW häufig parallel im Betrieb sind, wenn die Temperatur im Pufferspeicher nicht heiß genug ist, um den Wärmebedarf zu decken. Der parallele Betrieb lässt den Brennstoffbedarf stark ansteigen. Um dies zu vermeiden, sollten bei einem börsenoptimierten BHKW installierte Pumpen dafür sorgen, dass die Speichertemperatur im ausgeschalteten Zustand dauerhaft nur geringfügig unter die Mindesttemperatur (oft 60 °C) fällt.

E. Algorithmischer Lösungsansatz

Zur Lösung der Problemstellung wird die intelligente Suchmethodik „Simulated Annealing“ gewählt, die sich ins Teilgebiet „Künstliche Intelligenz“ der Informatik eingruppieren lässt. Ihr Einsatzgebiet ist die näherungsweise Lösung schwieriger Optimierungsproblemen mit großem Zustandsraum. Der Algorithmus ist dem Abkühlungsprozess in der Metallurgie nachempfunden. Bei der Suchmethodik wird versucht, bei hoher Temperatur das Problem in der Grundform zu erschließen, um bei fallender Temperatur nur noch den Feinschliff vorzunehmen. Die Temperatur wird durch einen Parameter im Algorithmus dargestellt. Bei der Implementierung wird die Suchmethodik meistens in zwei Funktionen unterteilt. Die *Nachbarschaftsfunktion* passt, angefangen mit einem Startzustand, in jeder Iteration den aktuellen Zustand zufällig an. Die *Trefferfunktion* prüft den derzeitigen Zustand und ermittelt den Trefferwert hinsichtlich des Optimierungsziels. Liefert der neu kreierte Zustand einen besseren Trefferwert als der vorherige Zustand, wird er übernommen. Ansonsten wird er mit einer Wahrscheinlichkeit übernommen, die mit dem Abfallen der Temperatur und mit der Höhe der Abweichung vom vorherigen Zustand geringer wird. Diese Funktionen werden in jedem Durchlauf abwechselnd aufgerufen. Nach jedem Durchlauf wird die Temperatur verringert, wodurch die Sprünge im Zustandsraum kleiner werden. Sobald die Temperatur einen festgelegten Wert erreicht, wird der bislang beste Zustand für die Lösung des Optimierungsproblems vorgeschlagen. [4]

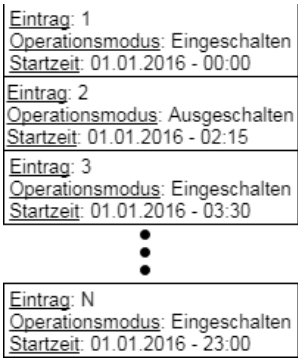


Abbildung 2. Datenstruktur.

III. Methode und Implementierung

Nachdem im vorherigen Kapitel die Grundkenntnisse und Rahmenbedingungen spezifiziert wurden, werden diese in den folgenden Abschnitten bei der Implementierung einer möglichen Vorgehensweise vorausgesetzt. Diese erlaubt es eine börsenoptimierten Steuerung für ein BHKW zu erstellen.

Untersuchungen heuristischer Suchverfahren haben gezeigt, dass Simulated Annealing gut geeignet ist, um nach einer Lösung für die Problemstellung zu suchen. Eine auf das Ziel zugeschnittene Implementierung einer Nachbarschafts- und Trefferfunktion wird in den folgenden Abschnitten beschrieben.

A. Nachbarschaftsfunktion

Für die Nachbarschaftsfunktion muss ein Zustand definiert werden. In diesem Fall wurde für den Zustand eine Datenstruktur gewählt, bei der ein Eintrag aus dem Operationsmodus des BHKWs und dessen Startzeit besteht (vgl. Abbildung 2). Das BHKW kann entweder ein- oder ausgeschaltet sein und besitzt demnach Operationsmodi, die immer abwechselnd durchgeführt werden. Die Startzeit eines Eintrags ist gleichzeitig die Endzeit des vorherigen Eintrags. Auf die Datenstruktur kann die Nachbarschaftsfunktion zwei Operationen ausführen. Sie kann entweder einen Eintrag aus der Datenstruktur entfernen oder hinzufügen. Beim Entfernen eines Eintrags muss unterschieden werden zwischen dem ersten, dem letzten und den restlichen Einträgen (vgl. Abbildung 3).

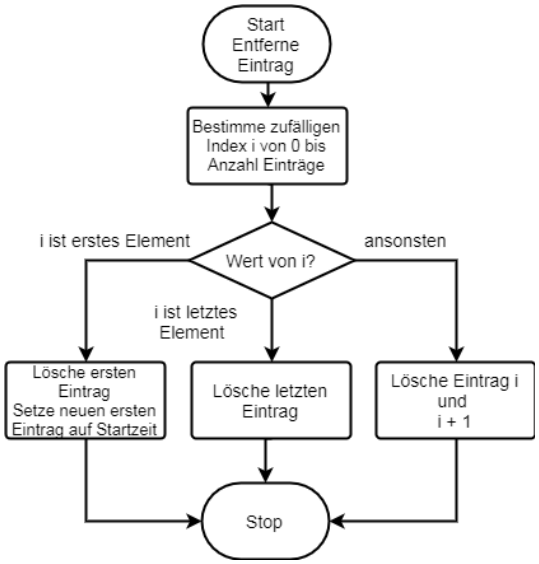


Abbildung 3. Entfernen eines Eintrags aus der Datenstruktur.

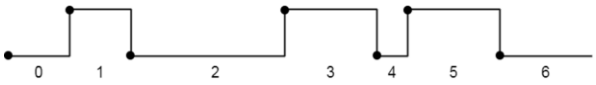


Abbildung 4. Steuersignal vor Löschoperation.

Soll das erste Element aus der Liste entfernt werden, muss neben dem Löschen des Eintrags noch eine Anpassung der Startzeit des zweiten Eintrags auf den Beginn des Zeitintervalls vorgenommen werden. Wenn der letzte Eintrag gelöscht wird, zieht sich dadurch der vorherige Punkt bis zum Ende des Intervalls. Punkte, die zwischen dem ersten und dem letzten Punkt liegen, werden durch Entfernen von zwei aufeinanderfolgenden Punkten realisiert. Dadurch wird der Operationsmodus des vorherigen Eintrags um die entfernte Zeit verlängert. Als Beispiel wird in Abbildung 4 ein Steuersignal dargestellt, auf das eine Löschoperation durchgeführt wird.

Die Punkte stellen einen Eintrag in der Datenstruktur dar. Im Beispiel wird für i der fünfte Eintrag gewürfelt. Dadurch werden die Einträge 5 und 6 entfernt und der vierte Eintrag wird implizit um die Zeit der entfernten Einträge verlängert. Das Ergebnis zeigt Abbildung 5.

Als Gegenspieler zur Löschoperation existiert eine Operation für das Hinzufügen eines Eintrags in die Datenstruktur (vgl Abbildung 6). Hierbei wird ein bestehender Eintrag, der lang genug ist, um einen inversen Eintrag aufzunehmen, ausgewählt.

Aus dem gewählten Eintrag werden schlussendlich durch das Einfügen drei Einträge. Erst ein Eintrag mit dem alten Operationsmodus, anschließend ein neuer Eintrag mit dem inversen Operationsmodus und zuletzt ein Eintrag mit dem originalen Operationsmodus.

Für die Länge eines Eintrages auf den eine Hinzufügeoperation durchgeführt werden soll, müssen also Randbedingungen für die Mindestlauf- und Mindest-

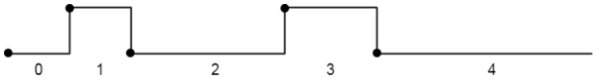


Abbildung 5. Steuersignal nach Löschoperation.

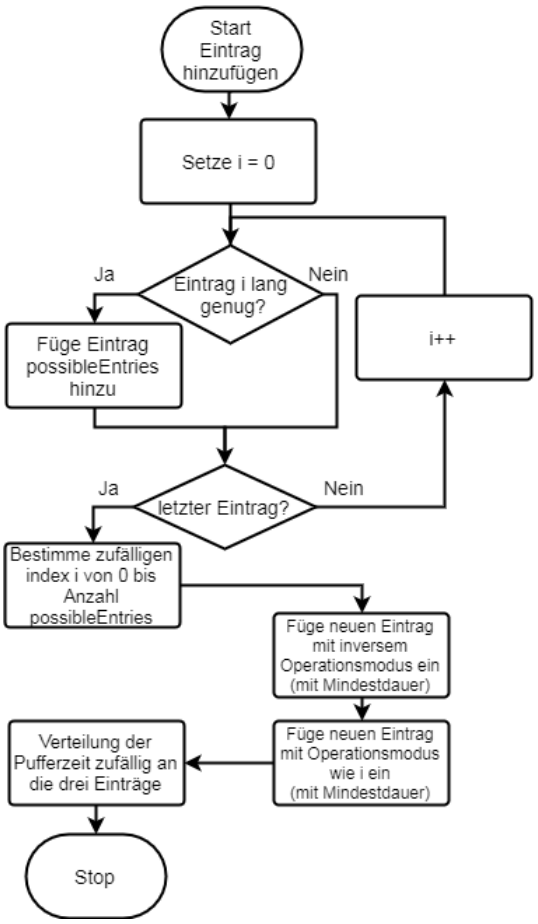


Abbildung 6. Hinzufügen eines Eintrags in die Datenstruktur.

kühlzeit gelten. Um geeignet Einträge zu ermitteln, werden alle Einträge in der Datenstruktur auf folgende Bedingung überprüft:

(i) Wenn gewählter Eintrag einen eingeschalteten Operationsmodus hat:

$$\text{Duration}\left(\text{Entry}_i\right) \geq 2 \cdot \text{Min}_{\text{ON}} + \text{Min}_{\text{OFF}}$$

(ii) Wenn gewählter Eintrag einen ausgeschalteten Operationsmodus hat:

$$\text{Duration}\left(\text{Entry}_i\right) \geq 2 \cdot \text{Min}_{\text{OFF}} + \text{Min}_{\text{ON}}$$

Wenn die Bedingung eingehalten werden kann, wird die zusätzlich vorhandene Zeit zufällig auf die drei Einträge aufgeteilt. Dies ist notwendig, damit keine explizite Operation für das Skalieren und Verschieben von Einträgen benötigt wird.

Auf das im letzten Beispiel resultierende Steuersignal wird im Folgenden eine Hinzufügeoperation durchgeführt. Das Steuersignal vor der Hinzufügeoperation zeigt Abbildung 7.

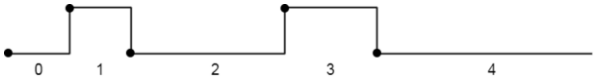


Abbildung 7. Steuersignal vor Hinzufügeoperation.

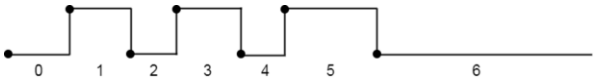


Abbildung 8. Steuersignal nach Hinzufügeoperation.

Die Einträge 2 und 4 erfüllen die Bedingung (ii). Sie sind lang genug, um einen Eintrag einzunehmen. Durch Würfeln wird der Eintrag 2 ausgewählt. Ein inverser Eintrag wird eingefügt, so dass aus dem Eintrag 2 implizit die Einträge 2, 3 und 4 entstehen.

Innerhalb einer Iteration kann entweder eine Entfern- oder Hinzufügeoperation oder eine Kombination aus beiden Operationen ausgeführt werden. Eine Kombination aus beiden hilft dem Algorithmus bei einer fortgeschrittenen Anzahl an Iterationen nicht festzustecken, sobald schlechtere Zustände nur noch mit einer sehr geringen Wahrscheinlichkeit übernommen werden. Wenn dann immer nur eine der beiden Operationen ausgeführt wird, kann es vorkommen, dass diese den Zustand nur noch verschlechtern können.

B. Trefferfunktion

Die Trefferfunktion beurteilt die Qualität des aktuellen Zustandes. Sie wird in zwei Teile unterteilt. Im ersten Teil wird der Temperaturverlauf im Pufferspeicher bewertet und im zweiten Teil der Profit an der Strombörse ermittelt. Um die Teile zusammen zu gewichten, wird von beiden Teilen ein normierter Wert zwischen 0 und 1 produziert.

1) Teil 1: Überprüfung des Pufferspeichers:

Für die Überprüfung des Pufferspeichers werden N Temperaturen in einem Vektor $T = (T_1, \ldots, T_N)$ mit $T_i \in \mathbb{R}$ festgehalten.

Wenn die Temperatur des Heizwassers im Pufferspeicher einen verhängnisvollen Wert (bei SenerTec Dachs 95 °C) erreicht, wird in der Realität eine Notabschaltung durchgeführt. Ein Steuersignal welches in die Gefahr läuft in die Nähe des Wertes zu kommen, sollte deshalb schlecht bewertet werden. Im betrachteten Fall werden alle Temperaturwerte über 90 °C als kritisch bewertet und fließen deshalb negativ in den Trefferwert ein.

Für die Normierung werden die Werte ab 90 °C in eine Hyperbelfunktion gegeben, die als Rückgabe einen Wert zwischen 0 und 1 liefert. Eine Hyperbelfunktion hat gegenüber einer Sprungfunktion den Vorteil, dass der Simulated Annealing Algorithmus leichter aus einem Zustand mit kritischen Temperaturwerten findet, da auch kleine Zustandsänderungen in Richtung der unkritischen Temperaturen zu einem besseren Trefferwert und somit zur sicheren Übernahme des Zustands führen. Es gilt:

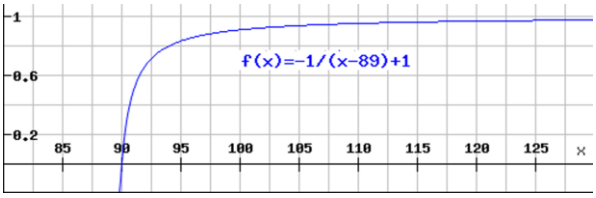


Abbildung 9. Hyperbel für Normalisierung der Temperaturwerte.

$$f(x) = \begin{cases} -\dfrac{1}{x-89} + 1, & x \geq 90 \\ 0, & x < 90 \end{cases} \tag{1}$$

Die normierten Werte werden einem Vektor $v = (f(T_1), \ldots, f(T_N))$ hinzugefügt. Nachdem alle Temperaturen im Pufferspeicher bewertet wurden, wird der Durchschnitt der Werte im Vektor gebildet.

$$\text{Score}_{\text{Pufferspeicher}} = \frac{\sum_{i=0}^{N} v_i}{N} \tag{2}$$

Somit wird neben der Stärke des kritischen Bereichs auch die Anzahl an Temperaturen im kritischen Bereich gewichtet. Im besten Fall liegen alle Temperaturwerte unterhalb des kritischen Bereichs, dann wird dieser Teil der Trefferfunktion mit dem bestmöglichen Wert von 0 bewertet.

2) Teil 2: Bewertung der Wirtschaftlichkeit:

Als zweiter Teil fließt der Profit an der Strombörse in den Trefferwert ein. Zunächst werden initial für den betrachteten Zeitraum die stündlich schwankenden Strompreisen an der EPEX SPOT $ES = (ES_1, \ldots, ES_N)$ mit $ES_i \in \mathbb{R}$ in das Programm geladen. Zusätzlich werden wie in Abschnitt II-D1 erläutert, Grenzkosten $g \in \mathbb{R}$ definiert, ab welchem Preis an der Strombörse sich das Einschalten des BHKW gegenüber dem Kessel rentiert. Mit Hilfe dieser Daten wird eine Zeitreihe $DG = (DG_1, \ldots, DG_N)$ mit $DG_i \in \mathbb{R}$ erstellt, welche die Differenz zwischen den Werten der EPEX SPOT Zeitreihe und den Grenzkosten enthält.

$$DG = ((ES_1 - g, \ldots, ES_N - g) \tag{3}$$

Diese Werte sind negativ, wenn der Preis an Strombörse unter den Grenzkosten liegt, ansonsten sind sie positiv. Damit kann einfach ermittelt werden, ob der Kessel oder das BHKW zu einer Stunde profitabler ist.

Für die Normierung der Wirtschaftlichkeit muss der größt- und kleinstmögliche wirtschaftliche Gewinn für den betrachteten Zeitraum bekannt sein. Der größtmögliche Wert lässt sich bestimmen, indem angenommen wird, dass das BHKW zu den Stunden läuft, an denen der Strompreis die Grenzkosten übersteigt. Hingegen

wird der kleinstmögliche Wert bestimmt, indem das BHKW immer läuft, wenn in ein negativer Wert steht.

Für alle DG_i in DG :

Wenn $DG_i \geq 0$:

$G_{\text{Oben}} += DG_i \cdot \text{LeistungProStunde}_{\text{BHKW}}$

Ansonsten :

$G_{\text{Unten}} += DG_i \cdot \text{LeistungProStunde}_{\text{BHKW}}$

Durch die beiden Grenzen *G* kann später eine Normierung des Zustands stattfinden, die immer zwischen 0 und 1 liegt. Nachdem die initialen Berechnungen geschehen sind, kann der Simulated Annealing Algorithmus gestartet werden. Die elektrische Leistung des BHKWs ist als Zeitreihe $ZR = (x_1, \ldots, x_N)$ mit $x_i \in \mathbb{N}$ bestehend aus N Zeitpunkten geben. Dabei stellt x_i einen Zeitpunkt (in Minuten ab dem Beginn des Zeitbereiches) innerhalb der Zeitreihe dar. Eine diskrete Funktion $f : \mathbb{N} \to \mathbb{R}$, welche den x-Werten die entsprechenden Leistungswerte zuordnet, ist gegeben. Innerhalb jeder Iteration muss zunächst die erzeugte elektrische Energie des BHKWs bestimmt werden, indem die elektrische Leistung stundenweise über den gewählten Zeitraum integriert wird. Die integrierte Zeitreihe $ZRI \in \mathbb{R}^N$ wird mit der Gleichung 4 gebildet.

$$ZRI = \Bigg(\frac{f(x_1) + f(x_2)}{2} \cdot (x_2 - x_1), \ldots, \frac{f(x_{N-1}) + f(x_N)}{2} \cdot (x_N - x_{N-1}) \Bigg) \tag{4}$$

Mit Hilfe der initial erstellten Zeitreihe wird der Profit des Zustands gegenüber dem Kessel ermittelt, indem die erzeugte elektrische Energie stundenweise mit dem Differenzwert multipliziert und anschließend zu einem Wert aufsummiert wird. Daraus ergibt sich ein Wert, der angibt, wie viel Euro Gewinn bzw. Verlust gegenüber dem reinen Kesselbetrieb mit dem aktuellen Zustand gemacht wird.

$$\text{Gewinn}_{\text{ges}} = \sum_{i=1}^{N} (DG_i \cdot ZRI_i) \tag{5}$$

Mit den initial berechneten Grenzen kann die Wirtschaftlichkeit auf einen Wert zwischen 0 und 1 normiert werden.

$$\text{Score}_{\text{Profit}} = \frac{(\text{G}_{\text{oben}} - \text{G}_{\text{unten}}) - (\text{Gewinn}_{\text{ges}} - \text{G}_{\text{unten}})}{\text{G}_{\text{oben}} - \text{G}_{\text{unten}}} \tag{6}$$

Dabei stellt 0 den bestmöglichen und 1 den schlechtestmöglichen Fall dar. Jeder Zustand im Zustandsraum befindet sich von der Wirtschaftlichkeit zwischen den beiden initial bestimmten Grenzen. Aufgrund des Pufferspeichers ist der bestmögliche Wert in der Regel nicht erreichbar. Einen möglichst hohen Profit zu finden unter Einhaltung der unkritischen Speichertemperaturen ist Aufgabe des Algorithmus.

3) Berechnung des gesamten Trefferwerts:

Der gesamte Trefferwert setzt sich zusammen aus den beiden beschriebenen Anteilen. Bei einer Gewichtung von jeweils 0.5 würden beide Teile gleichmäßig einfließen. Allerdings hat sich herausgestellt, dass eine stärkere Sanktionierung des Speichers nötig ist, um nicht Situationen zu generieren, bei denen der Trefferwert fällt, trotz des Anstiegs der Temperaturen im kritischen Bereich. Eine Gewichtung des Speichers mit 0.8 und des BHKWs mit 0.2 hat sich als geeignet erwiesen.

$$\text{Score} = 0.8 \cdot \text{Score}_{\text{Pufferspeicher}} + 0.2 \cdot \text{Score}_{\text{Profit}} \tag{7}$$

IV. Validierung

Im Folgenden werden die Ergebnisse einer Optimierung für den Zeitraum eines Tages (13.08.2016), sowie eines Jahres (2016) aufgeführt und erläutert. Die Betrachtung eines Tages ist beim Handel am Day-Ahead Markt einer Strombörse sinnvoll, da mit einer Prognose der Strompreisen und einer Prognose des Wärmebedarfs, eine Steuerung für den kommenden Tag erstellt werden kann [5]. Dahingegen bietet die Jahresbetrachtung eine Relevanz für die Wirtschaftlichkeit der börsenoptimierten Betriebsweise über einen längeren Zeitraum. Der Vergleich einer strombörsenoptimierten mit einer wärmegeführten Betriebsweise findet innerhalb der in Abschnitt II vorgestellten Werkzeugumgebung statt. Das hochakkurate Simulationsmodell bildet ein SenerTec Dachs ab. Dieses bietet eine elektrische Leistung von 5 kW. Im Heizkreis befindet sich ein Pufferspeicher mit einem Fassungsvolumen von 900 Litern. Für die Grenzkosten wurden 0,018 € pro kWh angenommen und als thermische Lastkurve wird eine reale Last eines Mehrfamilienhauses gewählt.

A. Tagesbetrachtung

In Abbildung 10 ist der Strompreisverlauf an der EPEX SPOT für den 13.08.2016, sowie der darauf optimierte Leistungsverlauf des BHKWs zu sehen. An diesem Tag herrschte an der EPEX SPOT Day-Ahead-Auktion während den Mittagsstunden ein Strompreis, der unter den Grenzkosten des BHKWs liegt. Dieser Umstand wird von der optimierten Steuerung beachtet, sodass das BHKW überwiegend in den Morgen- und Abendstunden läuft.

Wie Abbildung 11 zeigt, sorgt dies dafür, dass der Kessel in den in den Mittagsstunden zwei Mal taktet, sobald der Pufferspeicher „leer“ läuft, um den Wärmebedarf auszugleichen. Die Temperatur des Heizwassers im Pufferspeicher fällt unter den Mindestwert, der für die Abdeckung des Wärmebedarfs notwendig ist.

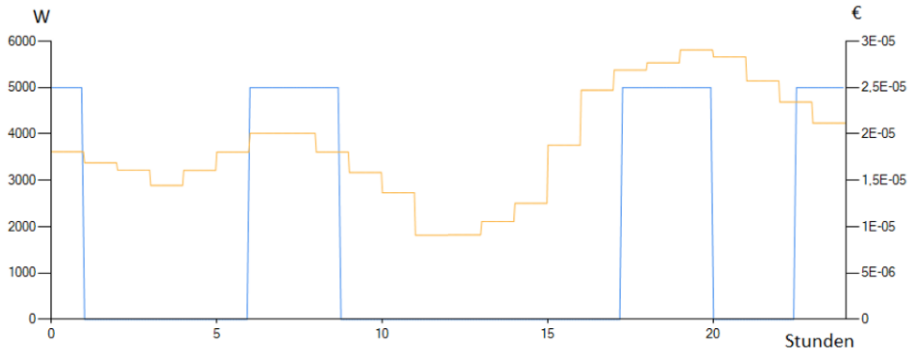


Abbildung 10. Optimierte BHKW Leistung und EPEX SPOT (13.08.2016).

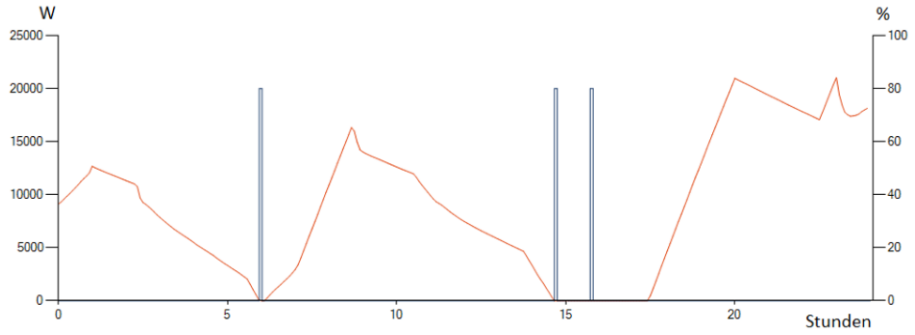


Abbildung 11. Pufferspeicher und Kessel (13.08.2016).

Während den Abendstunden, wenn der Preis an der Strombörse ansteigt, wird der Pufferspeicher an seine Grenze getrieben. Dadurch legt das BHKW um 20:00 Uhr eine Kühlpause ein, die es unterbricht sobald die Mindestlaufzeit wieder eingehalten werden kann, ohne bestraft zu werden aufgrund von kritischen Temperaturen.

Obwohl der Preis an der Strombörse an diesem Tag für mehrere Stunden unter den Grenzkosten liegt, muss der Kessel nur dreimal takten. Demnach liegt die Betriebszeit des BHKWs nur geringfügig unter der der wärmegeführten Betriebsweise. Um die optimierte Steuerung zu finden, hat der Simulated Annealing knapp 40000 Iterationen durchlaufen. Bereits nach 10000 Iterationen kann eine in den meisten Fällen völlig ausreichend optimierte Steuerung gefunden werden. Mehr als 40000 Iterationen führen nur noch zu marginal kleinen Profitsteigerungen, die vernachlässigt werden können.

B. Jahresbetrachtung

Im Folgenden wird die Wirtschaftlichkeit eines wärmegeführten BHKWs mit einem börsenoptimierten BHKW für das gesamte Jahr 2016 verglichen. Der durchschnittliche Strompreis an der EPEX SPOT lag 2016 bei 0,02898 € pro kWh.

Die entstandenen Ergebnisse sind in Tabelle I aufgeführt. Bei der wärmegeführten Betriebsweise erzielt das BHKW im Jahr 2016 einen Gewinn von 292.934 € gegenüber dem Kesselbetrieb. Dieser kommt zustande, da der durchschnittliche Strompreis an der EPEX SPOT höher war als die Grenzkosten und somit die Inbetriebnahme das BHKW im Durchschnitt profitabler als die des Kessels war.

Tabelle I
ERGEBNISSE DER JAHRESBETRACHTUNG.

	Erzeugte elektrische Energie	**Betriebs-stunden**	**Gewinn gegenüber Kesselbe-trieb**
Wärme-geführtes BHKW	24.525.833 Wh	4905,1 h pro 8784 h	292.934 € (100%)
Börsen-optimiertes BHKW	22.831.875 Wh	4566,3 h pro 8784 h	361.106 € (123,27%)

Mit einer börsenoptimierten Steuerung konnte für das Jahr 2016 ein Gewinn von 361.106 € gegenüber dem Kesselbetrieb erzielt werden. Dabei wurde die Betriebszeit des BHKWs um lediglich 6,07 % gegenüber der wärmegeführten Betriebsweise verringert. Im Vergleich mit der wärmegeführten Betriebsweise wurde der Gewinn um 68.172 € gesteigert. Dies entspricht 23,272 % mehr Ertrag. Allgemeiner betrachtet konnte pro kW Leistung des BHKWs 13,634 € mehr Ertrag gegenüber der wärmegeführten Betriebsweise erzielt werden.

C. Diskussion der Ergebnisse

Mit der Erstellung eines börsenoptimierten Steuersignals konnte bei der Simulation unter den beschriebenen Bedingungen und Annahmen ein über 23 % höherer Ertrag im Jahresvergleich erzielt werden als bei der wärmegeführten Betriebsweise. Um diesen Wert zu erzielen, sollten die in Abschnitt II.D beschriebenen Randbedingungen möglichst gut realisiert werden. Darüber hinaus ist zu beachten, dass wenn die Grenzkosten zu hoch angesetzt werden, sich die Laufzeiten des BHKWs über das Jahr gesehen deutlich verringern. Deshalb sollten die Grenzkosten im Voraus berechnet werden. Durch die Neufassung des KWK-Gesetzes von 2016 muss abgewogen werden, welche Betriebsweise unten den spezifischen Umständen am lukrativsten ist. Falls möglichst lange Betriebsstunden mit dem BHKW erreicht werden sollen und eine Direktvermarktung zu aufwendig ist, kann weiterhin eine wärmegeführte Betriebsweise rentabler sein, obwohl in den Stunden eines Börsenstrompreises kleiner oder gleich Null kein Zuschlag mehr erfolgt. Neben der Optimierung auf die Strombörse, kann in Mehrfamilienhäusern ein Mieterstrommodell ertragreich sein, bei dem die Schaltzyklen des BHKWs auf den Stromverbrauch der Mieter angepasst ist. Ferner kann für Stadtwerke eine Verringerung der CO_2-Emissionen interessant sein, wenn davon profitiert wird CO_2-Grenzen bei der Stromerzeugung einzuhalten. Die Optimierung auf weitere Kriterien ist durch Anpassung der vorgestellten Methodik möglich. Allerdings ist durch die steigende Konkurrenzfähigkeit der BHKWs zu den anderen Marktteilnehmern davon auszugehen, dass die Gesamtzahl der BHKWs die zur Direktvermarktung verpflichtet werden, in den nächsten Jahren prozentual zunimmt. Für diese ist dann eine wärmegeführte Betriebsweise in der Regel nicht länger rentabel.

Angesichts des Umstands, dass die Strombörse den Schwankungen für Angebot und Nachfrage unterliegt, wirkt sich die Optimierung nicht nur für Betreiber, sondern auch volkswirtschaftlich wertvoll aus. Anstatt das Stromnetz in Zeiten erhöhten Angebots zusätzlich zu belasten, helfen börsenoptimierte BHKWs dem Markt sich zu stabilisieren und verringern dadurch die Maßnahmen, die getätigt werden müssen.

V. Fazit

In der Arbeit wurde eine Vorgehensweise zur Multikriterien-Optimierung energietechnischer Komponenten beschrieben. Zur Darstellung des Verfahrens wurde als zu optimierendes Kriterium die Strombörse und als energietechnische Komponente ein Blockheizkraftwerk gewählt.

Für das BHKW konnte gezeigt werden, dass durch Anwendung des entwickelten Verfahrens ein über 23 % höherer Ertrag an der Strombörse gegenüber konventionellen Betriebsweisen erreicht wurde. Dabei wurden die Betriebsstunden lediglich um 6 % gesenkt.

Neben der Optimierung auf die Strombörse sind für ein BHKW eine Optimierung auf den Mieterstrom oder eine Optimierung zur Verringerung der CO_2-Emissionen geeignete Optimierungskriterien. Zudem ist eine Übertragung der Methodik auf weitere energietechnische Komponenten möglich.

Auf diese Weise konnte gezeigt werden, dass sich die Konkurrenzfähigkeit von umweltschonenden energietechnischen Komponenten gegenüber konventionellen Kraftwerken weiter steigern lässt, indem eine Steuerung durch systematisches Vorgehen anwendungsorientiert entwickelt wird.

Literaturverzeichnis

[1] Joachim Gerlach, Oliver Bringmann, Andreas Sauter, „Simulative Analyse dezentraler Energieversorgungskonzepte“, 2016.
[2] ASUE Arbeitsgemeinschaft, „Das KWK-Gesetz 2016“, 2016.
[3] ASUE Arbeitsgemeinschaft, „BHKW–Grundlagen“.
[4] David L. Poole, „Artificial Intelligence - Foundations of Computational Agents“, University of British Columbia, Vancouver: Cambride University Press, 2017.
[5] „Day-Ahead-Handel - was ist das?“, Available: https://www.next-kraftwerke.de/wissen/strommarkt/day-ahead-handel. [Zugriff am 21. November 2017].

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.

Stefan Rädle erhielt den akademischen Grad des B.Eng in Technischer Informatik im Jahr 2016 von der Hochschule Albstadt-Sigmaringen und studiert derzeit im Masterstudiengang Systems Engineering an der Hochschule Albstadt-Sigmaringen.

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“ und „Systems Engineering“.

Analog Properties of Printed Electrolyte-Gated FETs based on Metal Oxide Semiconductors

Xiaowei Feng, Gabriel Cadilha Marques, Farhan Rasheed, Mehdi B. Tahoori and Jasmin Aghassi-Hagmann

***Abstract*—Printed electronics (PE) offers attractive attributes such as mechanical flexibility and on-demand printing, which is complementary to the silicon technology. It has been shown that electrolyte-gated FETs (EGFETs) based on oxide semiconductors feature high device mobility, gate capacitance and drain-source current (I_{DS}), which offers benefits for low power and high-performance circuit design. In this work, we study this technology from the analog design perspective. Performance metrics, such as channel conductance (g_{ch}), transconductance (g_m), intrinsic gain (A_V) and efficiency (η), are analyzed for EGFETs, printed organic and silicon technologies. Results show that analog properties of EGFETs are superior compared to organic printed transistors, but currently not as good as in silicon transistors. Furthermore, the EKV-based model developed for EGFETs shows good agreement to the measurement data for these analog metrics in most regions, except where the source-drain current is very low.**

***Index Terms*—printed electronics, analog design, oxide semiconductor, electrolyte-gating**

I. Introduction

Printed electronics (PE) is an emerging technology with many promising application areas. It offers attractive attributes such as mechanical flexibility and on-demand printing which are complementary to the silicon electronics. The market research firm forecasted growth to $73 billion by 2027 [1]. Generally, PE technology can be categorized into two groups based on the semiconductor material, namely organic and inorganic. On one hand, organic field-effect transistors are popular due to advantages such as low processing temperature. However, they have low intrinsic carrier mobility in the range of 0.1-10 $cm^2V^{-1}s^{-1}$ and suffer from stability issues in the air [2]. On the other hand, inorganic materials, despite their disadvantages such as the high processing temperature, exhibits inherently higher mobility and superior stability than their organic counterparts. It is common to see the intrinsic mobility exceeding 100 $cm^2V^{-1}s^{-1}$ for high-quality crystalline oxide semiconductors [3]. In this context, we have developed electrolyte-gated FETs (EGFETS) [4] based on indium oxide (In_2O_3) as the semiconductor material, whose structure is shown in Figure 1. As will be discussed later in this paper, this combination ensures high drain-source current (I_{DS}) and low supply voltage.

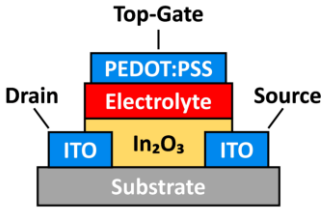


Figure 1. The cross-section view of the EGFET.

Benefiting from the recent development of printed transistor devices, the focus is now shifting towards the circuit level or even application level. Based on EGFET technology, digital circuits like the ring oscillator [5] and the physical unclonable functions (PUF) [6] have already been developed. In addition, there is also analog organic PE design presented by other researchers. For instance, the author of [7] demonstrated a 3-stage organic comparator. Additionally, a flexible surface electromyogram measurement sheet based on OFETs was presented in [8]. By viewing the most recent demonstrations in the field of PE, the following points can be concluded: First, it is expected that a certain amount of analog circuits will be involved in various applications, particularly in sensor applications where PE seems to be very promising. Second, despite the inferior performance, organic transistors still dominate PE analog circuit design, mainly due to the easy fabrication in early years.

Thus, it is of great interest to investigate, how much performance in analog circuits the EGFET technology can deliver, especially in comparison with other state-of-the-art technologies. In this work, we present a comprehensive analysis for this purpose. Important perfor-

Xiaowei Feng, xiaowei.feng@kit.edu, and J. Aghassi-Hagmann, jasmin.aghassi-hagmann@hs-offenburg.de are with the Department of Electrical Engineering and Information Technology at the Offenburg University of Applied Sciences.

X. Feng, G. C. Marques, gabriel.marques@kit.edu , J. Aghassi-Hagmann are with the Institute of Nanotechnology (INT) at the Karlsruhe Institute of Technology (KIT)

G.C. Marques, F. Rasheed, farhan.rasheed@kit.edu, M. Tahoori, mehdi.tahoori@kit.edu are with the Chair of Dependable Nano Computing (CDNC) of Department of Computer Science at the Karlsruhe Institute of Technology (KIT)

mance metrics for analog circuits are calculated from DC measurement or simulation of various technologies. Results show the EGFET technology provides an acceptable performance and its EKV based model [9] predicts the behavior in analog circuits precisely. The rest of the paper is organized as follows: Section II gives background information on organic and silicon technologies. Section III presents technical details of the EGFET technology, measurement and simulation environments as well as the figure of merit for comparison. Section IV shows the results. Conclusions are drawn in section V.

II. Background

In order to perform quantitative and comparative analysis of analog metrics of the EGFET technology, we compare it with the representative silicon technology, namely 65 nm CMOS technology, as well as well-established printed organic counterpart technologies.

For the organic semiconductor technology, we choose the organic thin-film-transistors (OTFTs) [10] developed by University of Minnesota (UMC) VLSI Group, which is representative among organic PE technologies. In this technology, ionic liquid is utilized as the top gate [10]. Poly(3-hexylthiophene), or P3HT, a popular organic semiconductor, is used as the semiconductor material. An organic process design kit (OPDK) is also provided by the same research group, facilitating the entire process of the circuit design by realizing circuit simulation and computer-aided layout design. The PDK is designed to be compatible with the contemporary standard of the silicon industry, which is based on the open source FreePDK [11]. It adopts the HSPICE level 61 transistor model [12] which is designed for the amorphous silicon transistor. In this work, the simulation results of OTFTs are obtained by using the OPDK.

As a representative silicon industry standard, we select the 65 nm silicon transistor, which is widely utilized in the modern silicon integrated circuits design. It represents the current trend in analog/mixed-signal market, whose tape-outs are increasing in recent years. The simulation results in this part are based on the PDK provided by the foundry [13].

III. Methodology

A. Fabrication and modeling of EGFETs

EGFETs we fabricated use precursor-derived indium oxide (In_2O_3) as the semiconductor material in combination with a solid polymer electrolyte as the gate insulator [4]. The later one offers a high gate capacitance C_i, e.g., 4.33 μF/cm^2 for 0.1 Hz @$V_G = 1$ V as reported in [14], higher than the traditional "high-k" dielectric. This results from the formation of an electric double layer (EDL) [15]. The combination of high mobility and gate capacitance brings the supply voltage to around 2 V or lower, while reasonably high drain-source current (I_{DS}) in the order of hundreds μA is achieved. The standard drain-source current I_{DS} equation in the saturation region gives a good explanation of this relationship:

$$I_{DS} = \frac{1}{2}\frac{W}{L}\mu C_i \left(V_{GS} - V_{TH}\right)^2 \qquad (1)$$

where W and L are the channel width and length, μ is the carrier mobility, V_{GS} is the gate-source voltage and V_{TH} is the threshold voltage.

The structure is prepared on a glass substrate, which allows a high processing temperature (300 °C - 500 °C) necessary for the process. The substrate is coated with sputtered indium tin oxide (ITO), which is then structured by e-beam lithography and serves as drain and source terminals. After that, between source and drain, indium oxide precursor is inkjet-printed using Dimatix 2831 printer and then annealed. Experimental results show devices annealed at 400 °C offer optimal performance [5]. Next, two extra layers have to be printed. One is the lithium ion-based composite solid polymer electrolyte (CSPE), which is dried at room temperature. It serves as the insulator, covering the semiconductor material. The other one is the PEDOT:PSS acting as the top gate. After that, EGFETs are ready for the electric characterization.

Table I
Design parameters of transistors.

	EGFET Measured	EGFET Model	OPDK	Silicon
W/L	2.5	2.5	2.5	2.5
W	100 μm	100 μm	100 μm	650 nm
L	40 μm	40 μm	40 μm	260 nm
V_{DS}	0.5 V	0.5 V	0.5 V	0.5 V

B. Measurement and simulation setup

Fabricated EGFET devices are characterized by the semiconductor analyzer Agilent 4156C, which applies independent V_{DS} and V_{GS} to transistors through the probe station. To get the transfer curve of the transistor, the V_{GS} is swept from 0 V to 1 V under different V_{DS}. In the meantime, channel current I_{DS} is measured. Similarly, the output curve is obtained by varying the V_{DS} and keeping the V_{GS} constant. Considering the process variation, the measurement is performed on 88 single transistors repeatedly. Finally, the curve with median value on the DC characteristic, representing the median performance, is selected. Besides, based on the measurement data, a smooth EKV-based model is also established, which enables accurate simulation in EDA tools. Here, in this work, it is verified, how good this prediction is, especially for analog circuits.

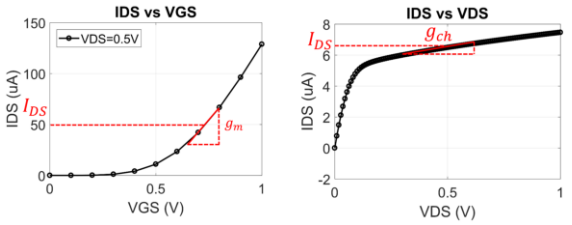


Figure 2. Calculation of transconductance out of transfer curve (left) and channel conductance out of transfer curve (right).

We carry out the simulation using the same schematics in the characterization of EGFETs in Cadence Virtuoso. Then, data are further processed in MATLAB.

Next, design parameters are determined, which are listed in Table 1. First, the length and width of both EGFETs and OTFTs are set to be 40 μm and 100 μm respectively, while those of silicon transistors are set to be 4 and 10 times of their minimum length. The purpose is to avoid short-channel effects for silicon transistors, whereas the W/L ratio for all is kept as 2.5. Secondly, the V_{DS} of 0.5 V is applied, which is half of the full supply voltage of 1 V. This considers the amplifier configuration in the analog circuit, for instance, the common source amplifier. In this case, transistor's DC output is usually configured at half of the supply voltage, such that a maximum headroom for the output swing is acquired. Finally, calculated metrics are plotted against I_{DS}, because I_{DS} is an indication of the device operating mode. Based on that, we defined roughly the I_{DS} of 10^{-7} to be the threshold region.

C. Figures of merit

The most effective way to compare transistors of various technologies is to look at their metrics. The basic ones, such as carrier mobility (μ) and gate capacitance (C_i), describe transistor's electric behavior on the physical level. However, their translations into higher order metrics give a clearer view of their performance in circuits. In the context of the analog design, nine metrics are of great importance, which reflect the performance in circuits directly from various aspects. They are transconductance (g_m), channel conductance (g_{ch}), intrinsic gain (A_V), cut-off frequency (f_c), efficiency (η), linearity, noise, mismatch and gate leakage.

By performing the DC characterization, four of the metrics can be derived at first: transconductance, channel conductance, intrinsic gain and efficiency. They are expressed as follows:

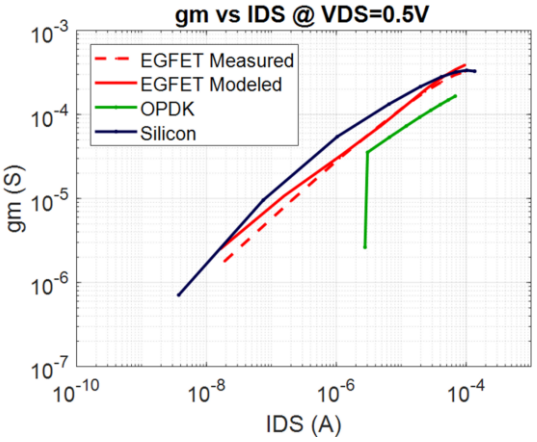


Figure 3. Transconductance v.s. drain-source current.

$$g_m = \left.\frac{\partial I_{DS}}{\partial V_{GS}}\right|_{V_{DS}=const.} \tag{2}$$

$$g_{ch} = \left.\frac{\partial I_{DS}}{\partial V_{DS}}\right|_{V_{GS}=const.} \tag{3}$$

$$A_V = \frac{g_m}{g_{ch}} \tag{4}$$

and

$$\eta = \frac{g_m}{I_{DS}} \tag{5}$$

Considering that all data here are discrete, we calculate transconductance and channel conductance numerically. Figure 2 gives an exemplary illustration. The other two metrics can be then calculated after gaining the first two.

IV. Result and Discussion

A. Transconductance versus drain-source current

Transconductance represents the transistor's ability to convert the small voltage change at the gate terminal to the change in the channel current. By reformulating the original definition of I_{DS} equation, two expressions of transconductance are obtained:

$$g_m = \frac{W}{L}\mu C_i \left(V_{GS} - V_{TH}\right) \tag{6}$$

and

$$g_m = \frac{2I_{DS}}{V_{GS} - V_{TH}} \tag{7}$$

The first expression explains the relationship between transconductance and physical properties of the transistor. The later one is more related to the working point of the transistor, implying that larger transconductance can be gained by setting a higher I_{DS}. From the result plotted in Figure 3, we can

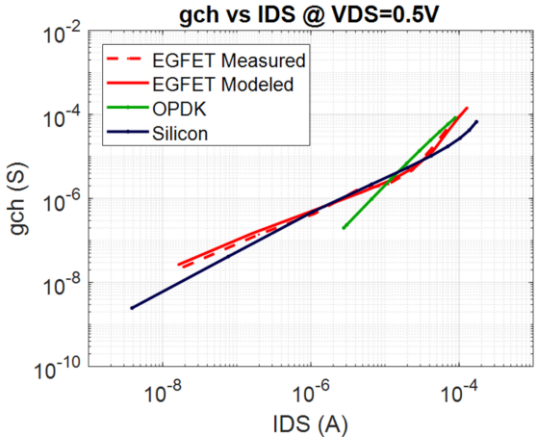


Figure 4. Channel conductance v.s. drain-source current.

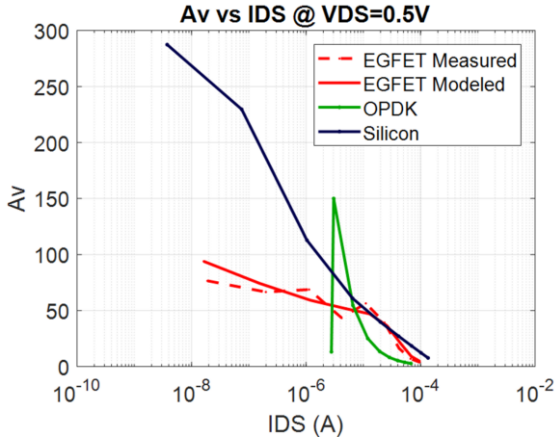


Figure 5. Intrinsic gain v.s. drain-source current.

conclude that transconductance is strongly dependent on technologies. Generally, all types of transistors have the same pattern of changing with I_{DS}. Owing to the good properties, the silicon transistor has the greatest value. Furthermore, the EKV-based model of EGFETs shows a good agreement to the measurement data in the most regions, except for the deep subthreshold region, where the model gives almost a doubled value as the real behavior. However, the curve of the OTFT has an abrupt drop around the subthreshold region, which is assumed to be a modeling failure.

B. Channel conductance versus drain-source current

Channel conductance is a fundamental parameter in transistors' small signal model. It stands for the output characteristic of the transistor, which is essential in circuit design and often interacts with the next stage. In the saturation region, it is impossible to write an expression of channel conductance just by the standard I_{DS} equation, since the channel length modulation is missing there. This implies that channel conductance is less dependent on the physical parameters. Therefore, a similar behavior for all transistors is expected.

Results from Figure 4 verify this hypothesis. Channel conductance of various transistors are close to each other. However, a distinction in the linear region is still to be seen, implying different working mechanisms in this area.

In terms of modeling, the model of EGFETs fits well to its measurement data. In contrast, again, OPDK fails to give a reasonable prediction near and below the subthreshold region.

C. Intrinsic gain versus drain-source current

Intrinsic gain, a combination of transconductance and channel conductance by definition, is one of the most important metrics in the amplifier design. It represents the maximum gain that a single common source stage can achieve by assuming the output is infinitely large.

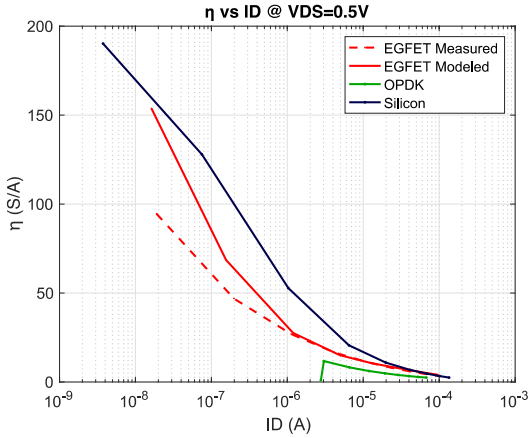


Figure 6. Efficiency v.s. drain current.

From Figure 5 we can conclude that intrinsic gain of silicon transistors and EGFETs are close to each other in the linear region. In contrast, OTFTs have clearly a lower value. When entering the saturation and subthreshold region, intrinsic gain of silicon transistors starts to increase aggressively, whereas the increase of EGFETs is slower.

The behavior of OTFTs in the saturation region, which shows a rapid growth, is a false translation, considering already existing failures in modeling transconductance and channel conductance. Additionally, the model of EGFETs gives a precise prediction in intrinsic gain.

D. Efficiency versus drain-source current

Efficiency $\eta = g_m/I_{DS}$ is a widely employed measure of the ability to translate the current, hence power, to transconductance [16]. In the silicon circuit design process, it also serves as a tool for the calculation of transistor dimensions. Alternatively, it can be written in the following form as well:

$$\frac{g_m}{I_{DS}} = \frac{1}{I_{DS}} \frac{\partial I_{DS}}{\partial V_{GS}} = \frac{\partial \ln (I_{DS})}{\partial V_{GS}} \tag{8}$$

In the linear region, η reaches its maximum value, where I_{DS} depends exponentially on V_{GS}. By entering the saturation region, this ratio starts to decrease almost linearly. For this reason, it is also an indicator of transistor's operation mode.

It can clearly be seen in Figure 6 that all types of transistors have a low efficiency with a similar value in the linear region. When entering the saturation region, the efficiency increases as expected, yet with a various rate for each transistor. Silicon transistors are the most efficient, while EGFETs have a moderate efficiency value. Despite lack of data in the subthreshold region, we still can conclude that OTFTs are the least efficient devices, which is no more than 75% of the efficiency of EGFETs.

The EGFET model shows a considerable error in the deep threshold region, owing to the already occurring modeling error of transconductance there. This will lead to an over-optimistic estimation in analog circuit design in this region, as far as power efficiency is concerned. As before, the model provided by OPDK fails to give value in the deep subthreshold region.

V. Conclusion

In this work, we gave a comprehensive comparison between three different technologies, namely EGFETs, OTFTs and silicon transistors, by focusing on four performance metrics, namely transconductance, channel conductance, intrinsic gain and efficiency. Concerning the analog properties under investigation, the EGFET technology is superior to its organic counterpart, mainly due to advanced physical properties such as mobility and gate capacitance. However, there is still a performance gap between EGFETS and silicon transistors. Additionally, we examined here the quality of models for EGFETs and OTFTs. The level-61 based OTFT model was proved to be less capable for analog circuit purpose, especially when transistors were working in the subthreshold region. The prime cause was that the physical mechanism of amorphous silicon transistors and OTFTs differ. In the analog design where more precision is demanded, this model is not sufficient. In contrast, the EKV-based transistor model for EGFETs was proved to be more accurate. However, it was still problematic in the modeling of transconductance near and below the subthreshold region, which needs more improvement in the future. Performance metrics discussed here were derived from DC simulation and measurement. As mentioned, there are other metrics, which rely on, e.g., AC and noise measurements. Future works will include these topics as well as the experimental characterization and simulation of dedicated test structures for them.

Acknowledgement

This work is supported by the graduate school MERAGEM and is a joint research between University of Applied Sciences Offenburg and Karlsruhe Institute of Technology.

References

[1] MD. Raghu, K. Ghaffarzadeh, G. Chansin, and X. He, "Printed, organic & flexible electronics: forecasts, players & opportunities 2017–2027.", 2017.

[2] K. Suganuma, *Introduction to Printed Electronics*, vol. 74. 2014.

[3] H. Q. Chiang, J. F. Wager, R. L. Hoffman, J. Jeong, and D. A. Keszler, "High mobility transparent thin-film transistors with amorphous zinc tin oxide channel layer", Appl. Phys. Lett., vol. 86, no. 1, pp. 2003–2006, 2005.

[4] G. C. Marques et al., "Electrolyte-gated FETs based on oxide semiconductors: Fabrication and modeling", IEEE Trans. Electron Devices, vol. 64, no. 1, pp. 279–285, 2017.

[5] G. Cadilha Marques et al., "Digital power and performance analysis of inkjet printed ring oscillators based on electrolyte-gated oxide electronics", *Appl. Phys. Lett.*, vol. 111, no. 10, 2017.

[6] A. T. Erozan, M. S. Golanbari, R. Bishnoi, J. Aghassi-Hagmann, and M. B. Tahoori, "Design and Evaluation of Physical Unclonable Function for Inorganic Printed Electronics", in *International Symposium on Quality Electronic Design (ISQED)*, 2018.

[7] R. W. Wanjau, D. Donaghy, and M. Raja, "Design of a high gain organic comparator for use in low-cost smart sensor systems", 2015 *11th Conf. Ph.D. Res. Microelectron. Electron. PRIME 2015*, pp. 37–40, 2015.

[8] H. Fuketa, K. Yoshioka, Y. Shinozuka, and K. Ishida, "1 um-Thickness Ultra-Flexible and High Electrode-Density Surface Electromyogram Measurement Sheet With 2 V Organic Transistors for Prosthetic Hand Control", *IEEE Trans. Biomed. Eng.*, vol. 8, no. 6, pp. 824–833, 2014.

[9] F. Rasheed, M. S. Golanbari, G. C. Marques, M. B. Tahoori, and J. Aghassi-hagmann, "A Smooth EKV-Based DC Model for Accurate Simulation of Printed Transistors and Their Process Variations", pp. 1–7, 2018.

[10] W. Zhang, "OPDK User Manual", 2014.

[11] J. E. Stine et al., "FreePDK: An open-source variation-aware design kit", *Proc. - MSE 2007 2007 IEEE Int. Conf. Microelectron. Syst. Educ. Educ. Syst. Des. Glob. Econ. a Secur. World*, pp. 173–174, 2007.

[12] Synopsys, "HSPICE ® Applications Manual", no. September, 2005.

[13] "65 nm CMOS Technology", Taiwan Semicond. Manuf. Company, Ltd.

[14] S. K. Garlapati et al., "Electrolyte-gated, high mobility inorganic oxide transistors from printed metal halides", *ACS Appl. Mater. Interfaces*, vol. 5, no. 22, pp. 11498–11502, 2013.

[15] S. H. Kim et al., "Electrolyte-gated transistors for organic and printed electronics", *Adv. Mater.*, vol. 25, no. 13, pp. 1822–1846, 2013.

[16] F. Silveira, D. Flandre, and P. G. A. Jespers, "A gm/ID based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA", *IEEE J. Solid-State Circuits*, vol. 31, no. 9, pp. 1314–1319, 1996.

Xiaowei Feng Xiaowei Feng received his B.Sc. degree from Tongji University, China, his B.Sc. degree in Systems Engineering and Automation in the year of 2013. He then started his Master of Electrical Engineering at RWTH Aachen University and completed his degree in July 2016. Since May 2017, he started working as a Scholarship Ph.D. student on the MERAGEM project. His research interest includes noise modeling and analog design in printed electronics.

Gabriel Cadilha Marques received the B.Eng. degree in electrical engineering and the M.Eng. degree in electronic and mechatronic systems from the Technische Hochschule Nürnberg Georg Simon Ohm, Nürnberg, Germany, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Chair of Dependable Nano Computing, Institute of Nanotechnology, Karlsruhe Institute of Technology, Karlsruhe, Germany, with a focus on the field of printed electronics.

Farhan Rasheed received the M.Sc. degree in nanoelectronic systems from the Technische Universität Dresden, Dresden, Germany, in 2016. He is currently pursuing the Ph.D. degreewith the Chair of Dependable Nano Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany, with a focus on process design kit development for printed electronics. His current research interests include design automation and process design kit development.

Mehdi B. Tahoori received the B.S. degree in computer engineering from Sharif University, Tehran, Iran, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2002 and 2003, respectively. He is currently a Professor with the Chair of Dependable Nano Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany. He has authored over 250 conference and journal papers.

Jasmin Aghassi received her M.Sc. in physics from Aachen University (RWTH), Aachen, Germany, and the Ph.D. from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. In 2007 she joined Infineon Technologies AG, in 2011 Intel Mobile Communications GmbH in Munich, Germany. She joined KIT in 2012 as a group leader and was additionally appointed full Professor in electrical engineering at the Offenburg University of Applied Sciences, Offenburg, Germany.

Stand der Technik von Powermanagement-ASICs für gedruckte Energy Harvester

Vy Le, Elke Mackensen

***Zusammenfassung*—Der Entwurf und die Realisierung gedruckter Schaltungen oder Elektronikkomponenten stellt ein intensives Thema der Forschung dar. Forschungsgruppen beschäftigen sich zunehmend mit der Entwicklung von gedruckten Energy Harvestern, weil diese kostengünstig und einfach herstellbar sind. Das Energy Harvesting (EH) oder auch das „Mikro Energy Harvesting“ (MEH) bezeichnet die Gewinnung von elektrischer Energie aus der Umgebung, um elektronische Verbraucher zu versorgen, kontinuierliche Leistungen zu erzeugen, das System energieeffizienter zu machen, sowie die Energiespeicherung im Mikrowattbereich zu gewährleisten. Energy Harvesting-Systeme stellen eine Alternative gegenüber der Energieversorgung autarker Low-Power-Elektronik mit Batterien dar. Das Energiemanagement solcher EH-Systeme ist jedoch eine Herausforderung aufgrund der Energieverfügbarkeit und der im Zeitablauf nicht konstanten Verlustleistung.**

Dieser Beitrag gibt einen Überblick über die derzeit existierenden ultra low-power Energiemanagement-Schaltungen für Energy Harvester. Dabei wird insbesondere der Fokus auf gedruckte Energy Harvester gelegt. Es soll aufgezeigt werden, welche Aspekte der vorgestellten Energieversorgungsschaltungen bei der Entwicklung eines Energieversorgungschips für gedruckte Energy Harvester berücksichtigt werden sollen.

***Schlüsselwörter*—Gedruckte Elektronik (PE), Energy Harvesting (EH), Schaltungsdesign, Energiemanagement.**

I. Einleitung

Mit der fortschreitenden Technik der Embedded Systems, begonnen bei Wearables, über die Unterhaltungselektronik, bis hin zum Monitoring, steigt der Bedarf nach einer autarken Energieversorgung [1].

Es besteht ein fundamentaler Unterschied zwischen Energy Harvestern und Batterien. Batterien stellen Energiequelle, -speicher und -wandler in einem dar. Entscheidend ist der aufsummierte Energiebedarf über die Laufzeit. Dahingegen liefern Energy Harvester ausschließlich Leistungen, dessen Zufuhr diskontinuierlich ist. [2, 3]

Das erforderliche Powermanagement regelt abhängig vom aktuellen Systemzustand den Leistungsanteil der Energiequellen und -speicher.

Das Ziel beim Energy Harvesting ist nicht, dass möglichst viel Energie geerntet wird, sondern dass die bestimmte Anwendung mit der verfügbaren Energie effektiv und autark betrieben wird [4].

Vy Le, vy.le@hs-offenburg.de,
Prof. Dr. Elke Mackensen, elke.mackensen@hs-offenburg.de.

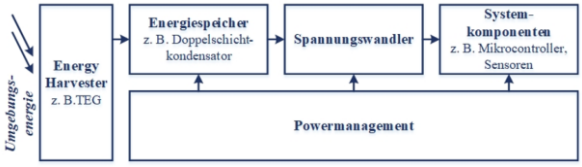


Abbildung 1. Prinzipielle Energieerzeugung und -verarbeitung.

Der grundsätzliche Aufbau eines ganzheitlichen Energy Harvesting-Systems ist in der Abbildung 1 dargestellt.

Unter Ausnutzung des Seebeck- oder Peltier-Effektes gewinnt ein thermoelektrischer Generator (TEG) Energie aus der Umgebung. Typischerweise wird die Ausgangsspannung des Energy Harvesters zunächst in einem Energiespeicher z.B. in Form eines Doppelschichtkondensators oder einer Batterie gespeichert. Mittels eines DC/DC-Wandlers wird die Spannung aufbereitet, sodass die Systemkomponenten mit einem definierten Spannungsniveau versorgt werden. Das Powermanagement charakterisiert nicht nur die Systemleistung. Es lassen sich Spannungssequenzen optimieren, sowie der Stromverbrauch minimieren.

II. Gedruckte Elektronik

Die gedruckte Elektronik ist eine aufstrebende Technologie mit Aussicht auf einen Multimilliarden-Dollar-Markt [5]. Das Forschungsgebiet der organischen Elektronik erstreckt sich von der Materialsynthese und Fabrikation funktionaler Tinten, über die Prozesstechnik, bis hin zum Schaltungsdesign und bildet eine Wertschöpfungskette von der Chemie, über den Maschinenbau, bis hin zur Elektrotechnik.

A. Einführung

Das Drucken von elektronischen Schaltungen mit additiven Verfahren stellt für das Internet der Dinge zunehmend einen Schlüsselfaktor zur Herstellung von Low-Cost-Schaltungen für digitale und analoge Anwendungen dar. Der Druck ist weitgehend komplementär zur konventionellen Siliziumtechnologie und ermöglicht schnelle Designänderungen, mechanisch

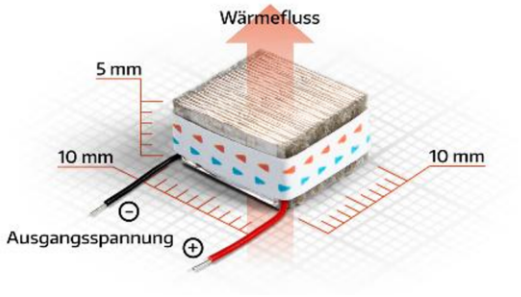


Abbildung 2. Thermoelektrischer Generator von Otego [15].

Abbildung 3. Herstellung einer Heliatek®-Solarfolie [16].

flexible Formfaktoren und die Integration auf unterschiedlichen Substraten, wie z. B. Glas oder Kunststoff [6].

Weiterhin können neuartige Komponenten,wie z.B. rollbare Displays, flexible Sensoren, intelligente Verpackungen, kostengünstige Speicher und einfach herstellbare Energy Harvester realisiert werden, die mit der Siliziumtechnologie nicht möglich oder zu kostenintensiv wären. [7]–[9]

Gedruckte, organische Elektronik besteht aus elektrisch aktiven Materialien wie Leitern, Halbleitern, Dielektrika, Leuchtstoffen, elektrochromen, elektrophoretischen oder Verkapselungsmaterialien. Die Kombination dieser Materialien bestimmt die Leistungsfähigkeit und Stabilität der organischen Elektronikkomponenten. [10, 11]

Die Drucktechnologien für die organische und anorganische Elektronik werden in subtraktive (non-fully printed) und additive (fully-printed) Prozesse klassifiziert. Beim Integrated Circuit (IC) basierten Subtraktionsverfahren wird bei der Herstellung eine Maske benötigt.Vorteilhafter ist das Additivverfahren, da die Funktionsmaterialien maskenlos übereinander gedruckt werden. Zu den am weitesten entwickelten Drucktechniken zählen der Siebdruck, Inkjet-Druck und Aerosol-Druck [6, 12, 13].

B. Gedruckte Energy Harvester

Die Herstellung von gedruckten Energy Harvestern wird seit einigen Jahren intensiv erforscht. Trotzdem sind erst wenige kommerzielle Energy Harvester erhältlich, insbesondere infolge der schnellen Degradierung der funktionalen Tinten [6].

Das Startup-Unternehmen Otego entwickelt einen würfelförmigen Thermoelektrischen Generator (TEG). An der Warm- und Kaltseite des TEGs liegen unterschiedliche Temperaturen an, wodurch eine Thermospannung generiert wird. Dieser Effekt wird Seebeck-Effekt genannt [14].

Der TEG besteht aus einer bedruckten und anschließend im Origamiprozess gefalteten Folie, siehe Abbildung 2. Dadurch wird eine gleichmäßige Struktur erreicht, sodass die Dicke des TEGs einstellbar ist. Die Ausgangsspannung ist kundenspezifisch variabel, da beliebig viele Thermopaare in Serie geschaltet werden können. Die thermoelektrischen Tinten entwickelt das Unternehmen selbst, da auf dem Markt keine produziert werden. Der signifikante Vorteil der TEGs liegt in der Flexibilität, da sie auf gekrümmte Oberflächen, wie z.B. Rohre angepasst werden können. Die Ausgangsspannung und der Kurzschluss-Ausgangsstrom sind abhängig vom jeweiligen Einsatz-Szenario. Die Ausgangsspannung der TEGs ist linear proportional zur angelegten Temperaturdifferenz. Die Kenndaten des TEGs in der Tabelle I gelten bei einer angelegten Temperaturdifferenz von 30 K. In Stickstoff gelagert funktionieren die TEGs ca. drei Jahre an der Luft [15].

Das Unternehmen Heliatek® stellt flexible Solarfolien aus ultradünnen Schichten organischer Moleküle im Rolle-zu-Rolle-Verfahren her, siehe Abbildung 3.

Unter inerter Atmosphäre werden kleine Moleküle auf eine PET-Folie aufgedampft, die auf maximal 120 °C erwärmt wird, sodass die Material kosten niedrig sind. Die Solarzelle weist ein sehr gutes Schwachlichtverhalten auf. Die Effizienz beträgt 13,2 % pro Jahr. Eine klassische Solarzelle weist einen Wirkungsgrad von 14 % bis 15 % auf. Die Solarfolien werden beispielsweise auf Gebäude-Fassaden zur Energiegewinnung genutzt und können in Baumaterialien integriert werden. [16]

Am Innovation Lab Heidelberg wurde eine flexible, organische Photodiode (OPD) durch eine additive, digitale Aerosol-Jet-Drucktechnologie entwickelt. Die miniaturisierten, polymeren Photodetektoren arbeiten wie eine Solarzelle und eignen sich beispielsweise für den Einsatz in Überwachungssystemen oder in der Medizintechnik [17]. Die Abbildung 4 zeigt den schematischen Aufbau einer gedruckten OPD auf Basis von zwei leitfähigen und transparenten PEDEOT:PSS Top- und Bottom-Elektroden, einer aluminiumdotierten Zink-Oxid-Elektronentransportschicht und eines aktiven Materials PTB7:PC70BM zwischen Anode und Kathode. Die Leerlauf-Ausgangsspannung liegt zwischen 0,6 V und 1,2 V bei einem Kurzschluss-Ausgangsstrom von 100 mA/W bis 400 mA/W. Mit geringerer Einstrahlungsintensität sinkt der Strom linear auf 1 μA/cm^2 bis 100 μA/cm^2 [17].

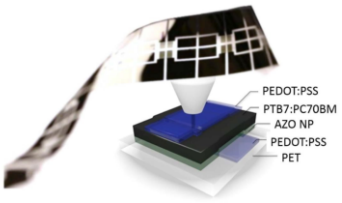


Abbildung 4. Schematischer Aufbau der gedruckten Photodiode [17].

Tabelle I
ZUSAMMENFASSUNG DER ORGANISCHEN ENERGY HARVESTER.

Kenngröße	**Otego TEG**	**Heliatek® Solarzelle [29]**	**iL Photodiode**
Leerlauf-Ausgangsspannung	600 mV	1,733 V	0,6 V bis 1,2 V
Kurzschluss-Ausgangsstrom	6 μA	8,03 mA/cm^2	100 mA/W bis 400 mA/W
Ausgangsleistung	108 μW	4,6 μW @ 500 lux 13,8 mW @ 120000 lux	5 mW/cm^2 bis 100 mW/cm^2

Die Tabelle I fasst die vorgestellten, organischen Energy Harvester mit den spezifischen Kenndaten zusammen.

III. ANFORDERUNGEN AN POWERMANAGEMENT-SCHALTUNGEN

Im Folgenden werden auf Grund der analysierten gedruckten Energy Harvester Anforderungen für den Entwurf von Powermanagement-Schaltungen abgeleitet. Die Anforderungen lassen sich in vier Bereiche gliedern:

- **Spannungsaufbereitung des Energy Harvesters**
  Die Gleichspannung oder Wechselspannung des Energy Harvesters muss über einen Spannungswandler in ein definiertes Gleichspannungsniveau transformiert werden. Das stabile Anlaufen und definierte Abschalten der aktiven Systemkomponenten muss gewährleistet sein. Der Wirkungsgrad muss möglichst groß sein. Zudem muss die Spannungsaufbereitung mit den sehr kleinen Spannungen und Strömen der Energy Harvester arbeiten können.
- **Speicherung der Energie**
  Ein Energiespeicher muss vorhanden sein, der die durch den Energy Harvester gesammelte Energie zwischenspeichert, um dann wiederum für den Betrieb der Systemkomponenten zu bestimmten Zeiten zur Verfügung zu stehen.
- **Intelligentes Energiemanagement**
  Mit dem Powermanagement sollen sich Spannungssequenzen für den Betrieb der Systemkomponenten verbessern, sowie der Energieverbrauch der Gesamtschaltung optimieren lassen.
  Um die Effizienz des Gesamtsystems zu bewerten, ist die Überwachung des Energiebedarfs des Gesamtsystems, beispielsweise durch ein Maximum Power Point Tracking unerlässlich. Der Fokus liegt auf der Steuerung der Sensorfunktionen und Kommunikation in Abhängigkeit der Dienstgüteanforderungen und des Speicherzustandes bzw. des Leistungseintrags.
- **Smarte Low-Power-Multi-Sensorik**
  Bei der Auswahl der Elektronikkomponenten muss die An- bzw. Abschaltung der Sensorik gewährleistet sein, ohne vorher eine Kalibrierung durchführen zu müssen. Dadurch wird der Stromverbrauch reduziert. Es sollen Sensoren eingesetzt werden, deren Leistungsaufnahme durch unterschiedliche Arbeitsmodi einstellbar ist.

Auf Grund der hier aufgeführten Anforderungen werden in den folgenden Abschnitten zum einen kommerziell erhältliche Powermanagement-Schaltungen und zum anderen aus der Forschung bekannte Powermanagement-Schaltungen analysiert und abschließend bewertet.

IV. STAND DER TECHNIK VON POWERMANAGEMENT-SCHALTUNGEN

A. Stand der Technik von kommerziell erhältlichen Powermanagement-ICs

Heutzutage werden verschiedene Powermanagement-Funktionen zumeist auf einem Powermanagement-Chip kombiniert.

Texas Instruments hocheffizienter Lade-IC BQ25504 für Energy Harvesting und Management-Anwendungen zeichnet sich durch einen niedrigen Ruhestrom von ca. 330 nA aus. Der IC hat einen Wirkungsgrad, abhängig von der Eingangsspannung, von bis zu 90 %. Aufgrund der variierenden Licht- und Wärmeverhältnisse bei Solarmodulen und thermoelektrischen Anwendungen beinhaltet der Chip ein programmierbares MPPT. Die Power Management Control Unit (PMU) überwacht kontinuierlich den Ladevorgang, wodurch sichergestellt ist, dass keine Über-, Unterspannung oder Überhitzung am Akku vorliegt [18].

Eine Powermanagement-Lösung für piezoelektrische Energy Harvester bietet der LTC3588-1 von Linear Technology. Der IC beinhaltet einen verlustarmen Vollwellen-Brückengleichrichter und einen hocheffizienten Abwärtswandler. Der Eingangsspannungsbereich beträgt 2,7 V bis 20 V. Die Ausgangsspannung ist programmierbar und beträgt 1,8 V bis 3,6 V. Der Ruhestrom des LTC3588-1 liegt bei abgetrennter Last bei 950 nA [19].

Der ultra low-power Energy Harvester SPV1050 von ST Microelectronics verfügt über ein integriertes Batterieladegerät, einen MPPT-Algorithmus und zwei Low-

Tabelle II
KOMMERZIELLE POWERMANAGEMENT-ICS [16, 17, 19].

Kenngröße	BQ25504	LTC3588-1	SPV1050
$U_{E(DC)}$	$\geq$ 80 mV	2,7 V	180 mV bis 18 V
$U_{A(DC)}$	3 V	1,8 V; 2,5 V; 3,3 V; 3,6 V	1,8 V; 3,3 V
Effizienz	bis zu 90 %	bis zu 92 %	bis zu 92 %
Kosten	4,68 €	5,88 €	2,53 €

Dropout Regler (LDOs). Sie werden über eine interne Logik gesteuert, sodass eine Ausgangsspannung von 1,8 V und 3,3 V anliegt. Der SPV1050 ist für Solarzellen und TEGs geeignet, da ein Eingangsspannungsbereich von 180 mV bis 18 V abgedeckt wird. Außerdem ist ein hoher Wirkungsgrad garantiert, aufgrund der Buck-Boost- bzw. Boost-Topologie. Der Chip bietet einen Überladeschutz und eine Unterspannungssperre, um Schäden zu vermeiden. [20, 21].

Die Kenngrößen der vorgestellten Powermanagement-ICs sind in Tabelle II aufgeführt.

B. Stand der Technik von Powermanagement-ICs in der Forschung

Die Züricher Hochschule für angewandte Wissenschaften (ZHAW)stellt in [22] eine Powermanagement-Schaltung für einen Thermogenerator vor. Der zweistufige Aufwärtswandler verarbeitet Eingangsspannungen von 10 mV bis 160 mV und erzeugt eine Ausgangsspannung bis zu 3,6 V. Die erste Stufe realisiert den Kaltstart und besteht aus einem selbstschwingenden Oszillator mit Transformator und einem nachfolgenden Spannungsverdoppler. Liegt eine Spannung von 36 mV am Eingang an, startet die erste Stufe und lädt den Zwischenspeicher. Sobald der Zwischenspeicher eine definierte Spannung von 2,8 V erreicht, wird die zweite Stufe aktiviert. Eine Power-On-Reset-Schaltung, bestehend aus einer Spannungsreferenz und einem Schmitt-Trigger, überwacht die Spannungen. Die Powermanagement-Schaltung in Kombination mit dem TEG 071-150-22 [23] erreicht einen Wirkungsgrad von 62 % bei einer Temperaturdifferenz von 3 K.

In [24] wird eine voll integrierte, hybride Powermanagement-Unit (HPMU) für passive UHF RFID-Tags vorgestellt, die in einem 130 nm CMOS-Prozess gefertigt wurde. Die HPMU nutzt überschüssige Energie, die über die Betriebsanforderungen eines RFID-Tags hinausgeht und speichert sie extern ab. Das sorgt für die Aufrechterhaltung der Versorgungsspannung des RFID-Tags, wodurch die Kaltstartzeit reduziert wird. Stromausfälle werden verhindert und die Reichweite und Reaktionsfähigkeit des Tags effektiv erhöht. Wenn die geerntete Leistung zu niedrig ist, wird die HPMU ausgeschaltet. Des Weiteren wurde die HPMU für einen niedrigen Stromverbrauch optimiert. Im ausgeschalteten Zustand verbraucht die HPMU ca. 10 nA vom externen Speicherelement und weist während des Ladungsspeicherbetriebs einen Spitzenwandlungswirkungsgrad von ca. 65 % auf.

Leicht et. al. [25] veröffentlichen ein effizientes autonomes Mikro-Powermanagement-System für elektromagnetische Energy Harvester. Es verarbeitet Spannungsbereiche von 440 mV bis 4,15 V bei einem Gesamtwirkungsgrad von ca. 72 %. Das Ziel ist die Optimierung der Ausgangsleistung durch eine Impedanzanpassung. Sie wird durch die Implementierung eines MPPT-Algorithmus erreicht. Die Bestimmung des Maximum Power Points erfolgt über eine Kontrollvariable, den sogenannten Stromflusswinkel α. Er ist das Verhältnis der gleichgerichteten Ausgangsspannung des Energy Harvesters und dessen Sinusspannung. α wird wie folgt berechnet:

$$\alpha = 2\cos^{-1}\left(\frac{V_{DCr}}{V_{SP}}\right)$$

Beim Einsatz eines elektromagnetischen Energy Harvesters liegt der Stromflusswinkel zwischen 65 ° und 67 °. Das Powermanagement-System ermöglicht autonomen Sensorsystemen aufgrund des adaptiven Nachführungsprinzips und der Spannungsstabilisierung einen autarken Betrieb.

Ulusan et. al. stellen in [27] ein Powermanagement-System für elektromagnetische Energy Harvester für Eingangsspannungen von 0,4 V_{pp} vor. Das System wurde im 180 nm CMOS-Prozess gefertigt und besteht aus zwei Stufen. In der ersten Stufe wird die Spannung durch einen energieautarken AC/DC-Wandler und nachgeschalteten Dioden gleichgerichtet. Die Spannung wird in der zweiten Stufe durch eine Niederspannungs-Ladepumpe auf 3 V geregelt. Das System erreicht einen Wirkungsgrad bis zu 86 %.

Fan et. al. zeigen in [28] ein Powermanagement-ASIC für vibrationsbasierte Energy Harvester. Die geerntete Energie wird durch einen Vollbrückengleichrichter in Gleichstrom gewandelt und lädt anschließend einen Filterkondensator. Durch einen Buck-Boost-Wandler wird eine Spannung von 3,3 V erzeugt, die in einem Superkondensator gespeichert wird. Die Spannungswandlungseffizienz beträgt 96 %. Anschließend erfolgt eine Impedanzanpassung über ein MPPT. Ein LDO-Regler mit einem Ruhestrom von 750 pA sorgt für die Versorgung der On-Chip Sensoren.

Liu et. al. [28] verwenden ein sogenanntes „Active Piezoelectric Energy Harvesting Scheme (AEHS)“, siehe Tabelle I. Über aktive Schalter wird die Ausgangsspannung des piezoelektrischen Energy Harvesters gleichgerichtet. Die Funktionsweise der Powermanagement-Schaltung gliedert sich in zwei

Tabelle III
ÜBERSICHT DES STANDES DER TECHNIK VON ERFORSCHTEN POWERMANAGEMENT-ICS.

Kenngröße	**ZHAW [22]**	**Sun et. al [24]**	**Leicht et. al [25]**	**Ulusan et. al [26]**	**Fan et. al [27]**	**Liu et. al [28]**
Systembeschreibung	Zweistufiger Aufwärtswandler	Hybride Powermanagement-Unit	Autarkes Mikro-Powermanagementsystem mit Impedanzanpassung	Zweistufiges Powermanagementsystem	Autarkes Mikro-Powermanagementsystem mit Impedanzanpassung	Active Piezoelectric Energy Harvesting Scheme
$U_{E(DC),min}$	10 mV	bis zu 4 V	440 mV	400 mV	keine Angabe	keine Angabe
$U_{A(DC)}$	bis 3,6 V	anwendungsspezifisch	anwendungs- spezifisch	3 V	3,3 V	bis 2 V
Effizienz	ca. 62 %	ca. 65 %	ca. 67 %	ca. 86 %	ca. 96 %	ca. 78 %

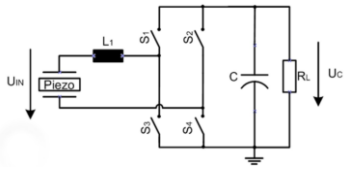


Abbildung 5. Prinzipielles Modell der AEHS-Schaltung.

Schritte: Nachdem die maximale Spannung am Energy Harvester anliegt, wird der Energiespeicher geladen, indem die Schalter S1 und S4 geschlossen und die Schalter S2 und S3 geöffnet werden. Anschließend wird die Spannung am Energy Harvester invertiert, indem die Schalter umgekehrt geöffnet bzw. geschlossen werden. Die Ausgangsspannung des Powermanagement-Schaltkreises beträgt 1 V bis zu 2 V bei einer durchschnittlichen Effizienz von 78 %. Die Optimierung der Ausgangsleistung wird durch eine Impedanzanpassung sichergestellt.

Die Tabelle III fasst alle vorgestellten Powermanagement-Schaltungen zusammen.

V. FAZIT

Im Rahmen dieses Beitrags wurde eine Einführung in das Thema der gedruckten Elektronik gegeben. Es wurden sowohl die Möglichkeiten, als auch die Grenzen der Energiewandlung mit Hilfe von gedruckten Energy Harvestern erläutert. Basierend auf dem aktuellen Stand der Technik wurden die Anforderungen an Powermanagement-Systeme für gedruckte Energy Harvester abgeleitet. Diese Kriterien müssen bei der Entwicklung eines effizienten Powermanagement-Systems berücksichtigt werden, sodass der Energiebedarf so gering wie möglich ist. Anschließend wurden kommerziell erhältliche und derzeit erforschte Powermanagement-ICs für konventionelle, siliziumbasierte Energy Harvester analysiert.

Der Entwicklungsstand der vorgestellten Powermanagement-Schaltungen zeigte, dass es für gedruckte Energy-Harvester noch keine Lösungen gibt, insbesondere auf Grund der Tatsache, dass diese nur kleine Ausgangsleistungen zur Verfügung stellen. Es besteht der Bedarf einer System-On-Chip-Integration von gedruckten Energy Harvestern sowie Energiespeichern in Kombination mit entsprechenden siliziumbasierten Energieaufbereitungs- und Energiemanagement-Lösungen. Die Erforschung von ganzheitlichen Energy Harvesting-Konzepten ermöglicht einen signifikanten Fortschritt im Bereich der energieautarken, hochminiaturisierten Low-Power-Sensorsysteme. Im Hinblick auf die Optimierung von Energy Harvesting-Konzepten für gedruckte Energy Harvester sollen integrierte On-Chip-Sensorsysteme entwickelt, gefertigt und untersucht werden.

LITERATURVERZEICHNIS

[1] K. Lundager, B. Zeinali, M. Tohidi, J. K. Madsen und F. Moradi, „Low Power Design for Future Wearable and Implantable Devices“, *Journal of Low Power Electronics and Applications*, Okt. 2016.

[2] E. Mackensen und T. Wendt, „Energy-Harvesting-basierte Energieversorgungen für drahtlose Sensor-Systeme: Analyse kommerziell verfügbarer Lösungen und darauf abeleitete Design-Konzepte“, *WEKA Fachmedien GmbH (Hrsg.): 1. Elektronik energy harvesting congress 2012, Tagungs-unterlagen. München: WEKA Fachmedien GmbH*, 2012.

[3] B. Leistritz, W. Kattanek, E. Chervakova, S. Krug, S. Engelhardt und A. Schreiber, „Systematischer Entwurf Plug-and-Play-fähiger Funksensoren mit Vibrations-Energy-Harvestern“, *Dresden: Präsentation 9. GMM-Workshop Ener-gieautonome Sensorsysteme*, 2018.

[4] K. Dembowski, „Energy-Harvesting für die Mikroelektronik: energie-effiziente und -autarke Lösungen für drahtlose Sensorsysteme“, *Berlin: VDE Verlag*, 2011.

[5] OE-A, „Who We Are“, Online verfügbar unter:http://www.oe-a.org/who-we-are. [Zugriff am 1 Oktober 2017].

[6] J. Chang, T. Ge und E. Sanchez-Sinencio, „Challenges of printed electronics on flexible substrates“, *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), Boise, ID*, pp. 582-585, 2012.

[7] J. S. Chang, A. F. Facchetti und R. Reuss, „A Circuits and Systems Perspective of Organic/Printed Electronics: Review, Challenges, and Contemporary and Emerging Design Approaches“, *IEEE Journal on emerging and selected topics in circuits and systems, Vol. 7, No. 1*, pp. 7-26, März 2017.

[8] G. A. T. Sevilla und M. M. Hussain, „Printed Organic and Inorganic Electronics: Devices To Systems“, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 7, no. 1*, pp. 147-160, März 2017.

[9] A. Nag, S. Mukhopadhyay und J. Kosel, „Wearable flexible sensors“, *IEEE Journal, vol. 17, no.13*, pp. 829-842, 2017.

[10] E. Cantatore, „Applications of Organic and Printed Electronics“, *London: Springer*, 2013.

[11] OE-A„Organic and Printed Electronics: Application, Technologies and Supplier“, *5th Edition*, 2013.

[12] G. C. Marques, F. Rasheed, J. Aghassi-Hagmann und M. B. Tahoori, „From silicon to printed electroinics: A coherent modeling and design flow approach based on printed electrolyte gated FETs“, *IEEE Design Automation Conference (ASP-DAC), 2018 23rd Asia and South Pacific*, pp. 658-663, 2017.

[13] P. Fastermann, „3D-Drucken: Wie die generative Fertigungstechnik funktioniert“, *Krefeld-Uerdingen: Springer*, 2016.

[14] F. Bernhard, „Handbuch der Technischen Temperaturmessung“, *Heidelberg: Springer Vieweg*, 2014.

[15] Otego, „Powering the Internet of Things“, *Karlsruhe: Interne Quellen*, 2017.

[16] Heliatek®, „Die Zukunft ist leicht: Organische Solarfolien von Heliatek“, Online verfügbar unter: http://www.heliatek.com/de/. [Zugriff am 25 April 2018].

[17] R. Eckstein, „Aerosol Jet Printed Electronic Devices and Systems“, *Karlsruhe*, 2016.

[18] Texas Instruments, „BQ25504 Ultra Low-Power Boost Converter With Battery Management for Energy Harvester Applications“, *Texas*, 2015.

[19] Linear Technology, „LTC3588-1 Nanopower Energy Harvesting Power Supply“, *Milpitas, CA*, 2010.

[20] A. Nicosia, D. Pau, D. Giacalone, E. Plebani, A. Bosco und A. Iacchetti, „Efficient light harvesting for accurate neural classification of human activities“, *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 1-4, 2018.

[21] STMicroelectronics, „SPV1050 Ultralow power energy harvester and battery charger“, *Genf, Schweiz*, 2015.

[22] J.-M. Gruber, S. Mathis, „Aufwärtswandler für Thermogeneratoren: Kleine Temperaturdifferenzen optimal nutzen“, *Elektronik 10/2017*, pp. 34-38, 2017.

[23] Thermalforce, „Thermo Electrical Generator TEG 071-150-22“, Online verfügbar unter: http://thermalforce.de/de/product/thermogenerator/TG71-150-22h.pdf. [Zugriff am 13 November 2017].

[24] M. Sun, S. F. Al-Sarawi, P. Ashenden, M. Cavaiuolo und D. C. Ranasinghe, „A Fully Integrated Hybrid Power Management Unit for Passive UHF RFID in 130-nm Process“, *IEEE Journal of Radio Frequency Identification*, Vol. 1, NO. 1, pp. 90-99, 2017.

[25] J. Leicht, D. Maurath und Y. Manoli, „Autonomous and Self-Starting Efficient Micro Energy Harvesting Interface with Adaptive MPPT Buffer Monitoring, and Voltage Stabilization“, *2012 Proceedings of the ESSCIRC (ESSCIRC)*, Bordeaux , pp. 101-104, 2012.

[26] H. Ulusan, Ö. Zorlu, A. Muhtaroglu und H. Külah, „Highly Integrated 3 V Supply Electronics for Electromagnetic Energy Harvesters with Minimum 0.4 V Peak Input“, *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5460-5467, 2017.

[27] S. Fan, L. Zhao, R. Wei, L. Geng und P. Feng, „An ultra-low quiescent current power management ASIC with MPPT for vibrational energy harvesting“, *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, pp. 1-4, 2017.

[28] Y. Liu, G. Tian, Y. Wang, J. Lin, Q. Zhang und H. Hofmann, „Active Piezoelectric Energy Harvesting: General Principle and Experimental Demonstration“, *Journal of intelligent material systems and structures*, Vol. 20, pp. 575-585, März 2009.

[29] M. A. Green, K. Emery, Y. Hishikawa, W. Warta, „Solar cell efficiency tables (version 37)“ *Wiley Online Library*. *DOI:10.1002/pip.1088*, pp. 84-92, 2011.

Vy Le erhielt den akademischen Grad des B.Eng. in Elektro- und Informationstechnik^{Plus} im Jahr 215 und den M.Sc.-Titel im Jahr 2017 von der Hochschule Offenburg. Derzeit promoviert sie im Fachbereich Elektro- und Informationstechnik in Kooperation mit dem Karlsruher Institut für Technologie. Ihr Forschungsschwerpunkt liegt im Bereich der ganzheitlichen Energy Harvesting-Konzepte unter Verwendung von gedruckten Energiespeichern und –wandlern.

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

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. Koblitz, (0721) 925-2238
marc.ihle@hs-karlsruhe.de

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

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

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

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

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

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

Hochschule Ulm
Prof. Dr. Terzis, (0731) 502-8341
terzis@hs-ulm.de

www.mpc-gruppe.de