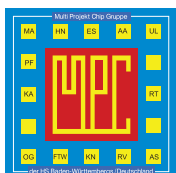


MPC

MULTI PROJEKT CHIP GRUPPE
BADEN - WÜRTTEMBERG

Herausgeber: Hochschule Ulm Ausgabe: 49 ISSN 1868-9221 Workshop: Mannheim Februar 2013

- 1 **Entwicklung von Schaltungen auf Basis organischer Elektronik**
S. Hengen, B. Scherwath, C. Haas, B. Vettermann, J. Giehl, HS Mannheim
- 7 **Design and Implementation of a Mixed-Signal ASIC in Ultra-Thin CMOS Technology for 2-D Stress Measurement**
Y. U. Mahsereci, C. Scherjon, IMS Chips Stuttgart
- 13 **DC Forward Model of High-Power LED's**
V. Lange, B. Sherston, R. Hönl, HS Furtwangen
- 17 **A Current-Mode Buck-Boost DC-DC Converter with Fast Transient Response**
A. Ehrhart, B. Wicht, K.-H. Chen, HS Reutlingen
- 23 **Entwurf eines rauscharmen Vorverstärkers für 60 GHz in SiGe:C 0,25 µm**
P. Kuhn, S. Lensing, G. Forster, R. Münzner, HS Ulm
- 33 **Direktumsetzender großsignalfester IQ-Demodulator für SDR-Anwendungen im UKW-Band mit High-Speed MOSFET Demultiplexer IC**
P. K. Nwabo, J. Mayer, HS Mannheim
- 39 **Optimierung eines 32-Bit-Softcores mit Harvard-Architektur und Dual Cache sowie Memory Management Unit für Embedded Computing**
M. Schmidt, S. Stickel, F. Zowislok, D. Jansen, HS Offenburg
- 45 **Ein präziser Signalgenerator mit 16-Bit-DAC auf einem Low-Power-FPGA**
P. Harrer, S. Kammerer, T. Wucher, I. Schoppa, HS Konstanz
- 51 **VHDL Based Circuit Description to Transmit, Receive and Analyse USB Data Packets**
M. G. S. Suárez, HS Ravensburg-Weingarten
- 57 **Erkennung von Fußgängern in Echtzeit auf FPGAs**
M. Hahnle, F. Saxen, K. Doll, HS Aschaffenburg
- 67 **Headless Android Systems for Industrial Automation and Control**
A. Hassan, A. Sikora, D.-S. Kunz, D. Eberlein, HS Offenburg



Cooperating Organisation
Solid-State Circuit Society Chapter
IEEE German Section



Inhaltsverzeichnis

Entwicklung von Schaltungen auf Basis organischer Elektronik	1
S. Hengen, B. Scherwath, C. Haas, B. Vettermann, J. Giehl, HS Mannheim	
Design and Implementation of a Mixed-Signal ASIC in Ultra-Thin CMOS Technology for 2-D Stress Measurement	7
Y. U. Mahsereci, C. Scherjon, IMS Chips Stuttgart	
DC Forward Model of High-Power LED's	13
V. Lange, B. Sherston, R. Hönl, HS Furtwangen	
A Current-Mode Buck-Boost DC-DC Converter with Fast Transient Response	17
A. Ehrhart, B. Wicht, K.-H. Chen, HS Reutlingen	
Entwurf eines rauscharmen Vorverstärkers für 60 GHz in SiGe:C 0,25 μm	23
P. Kuhn, S. Lensing, G. Forster, R. Münzner, HS Ulm	
Direktumsetzender großsignalfester IQ-Demodulator für SDR-Anwendungen im UKW-Band mit High-Speed MOSFET Demultiplexer IC	33
P. K. Nwabo, J. Mayer, HS Mannheim	
Optimierung eines 32-Bit-Softcores mit Harvard-Architektur und Dual Cache sowie Memory Management Unit für Embedded Computing	39
M. Schmidt, S. Stickel, F. Zowislok, D. Jansen, HS Offenburg	
Ein präziser Signalgenerator mit 16-Bit-DAC auf einem Low-Power-FPGA	45
P. Harrer, S. Kammerer, T. Wucher, I. Schoppa, HS Konstanz	
VHDL Based Circuit Description to Transmit, Receive and Analyse USB Data Packets	51
M. G. S. Suárez, HS Ravensburg-Weingarten	
Erkennung von Fußgängern in Echtzeit auf FPGAs	57
M. Hahnle, F. Saxen, K. Doll, HS Aschaffenburg	
Headless Android Systems for Industrial Automation and Control	67
A. Hassan, A. Sikora, D.-S. Kunz, D. Eberlein, HS Offenburg	
Gefertigte ASICs	
Testchip für ein Energy Harvesting IP in 0,18 μm CMOS-Technologie	75
J. Rechtsteiner, G. Forster, HS Ulm	

Tagungsband zum Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie.

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

Herausgeber: Gerhard Forster, Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

Alle Rechte vorbehalten

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

<http://www.mpc.belwue.de>

Entwicklung von Schaltungen auf Basis organischer Elektronik

Stefan Hengen, Bernd Scherwath, Christian Haas, Bernd Vettermann, Jürgen Giehl

Zusammenfassung—Organische Materialien werden schon heute in Handysdisplays und Solarzellen eingesetzt. Transistoren und Sensoren lassen sich damit ebenfalls realisieren, haben aber den Sprung vom Labor zum Massenmarkt noch nicht geschafft. Durch Gründung des Spitzenclusters Forum Organic Electronics unterstützt das BMBF diese aufkommende Technologie. Von besonderem Interesse sind organische Transistoren (OTFT). Diese lassen sich prinzipiell aus Lösungen heraus verarbeiten (z. B. drucken) und versprechen dadurch extrem günstige Herstellungskosten. Am Beispiel eines Inverters wird der Herstellungsprozess vom Schaltplan zur fertigen Schaltung erläutert. Insbesondere die Extraktion der Transistorparameter, Modellierung, Simulation und die Charakterisierung von Schaltungen werden gezeigt. Zur zielgerichteten Charakterisierung wurden mit dem MENTOR Layouteditor Teststrukturen entworfen. Ebenso wurden Messtechnik und Routinen zur Parameterextraktion konzipiert. Darauf aufbauend wurden Software zur Datenanalyse und Modelle zur Transistorsimulation entwickelt, erweitert und in Simulationsprogramme (MENTOR, LTspice) implementiert. Hier wurden die besonderen Eigenschaften der organischen Materialien berücksichtigt. So mussten z.B. vorhandene Modelle um Kontaktwiderstände und Gate-Leckströme erweitert werden. Zur Verifikation wurden die Simulationsergebnisse mit Messungen an einzelnen Transistoren und aufgebauten Schaltungen verglichen.

Schlüsselwörter—Organische Elektronik, Organic Electronics, OFET, OTFT, Simulation, Modellierung.

I. EINLEITUNG

Gedruckte organische Elektronik verspricht Neuerungen, welche mit herkömmlicher Silizium-Technologie nicht umsetzbar sind. Schaltungen können flexibel, transparent und großflächig hergestellt werden, außerdem könnten die Herstellungskosten deutlich gesenkt werden.

Organische Elektronik basiert auf Halbleitern aus organischen Materialien, diese lassen sich prinzipiell in Lösemitteln lösen und dann flüssig verarbeiten. Dadurch wird es möglich, die Herstellungsprozesse von komplizierten und aufwendigen Verfahren zu einfacheren Druckprozessen umzustellen. Transistoren aus flüssigprozessierten Materialien haben jedoch Nachteile gegenüber der Silizium-Technologie. Durch die einfacheren Prozesse müssen die Strukturen deutlich größer gewählt werden, gleichzeitig ist jedoch die Ladungsträgerbeweglichkeit etwa zwei bis drei Größenordnungen geringer als bei Silizium. Dies führt dazu, dass Schaltungen mit wenigen Transistoren mehrere cm^2 groß werden können. Da es angestrebt ist, elektronische Schaltungen z.B. bei der Verpackungsherstellung im Druckprozess zu erzeugen, spielt jedoch der Flächenbedarf eine untergeordnete Rolle. Es gibt auch Anwendungsfälle, in denen explizit große Flächen gefertigt werden sollen, z.B. organische Photovoltaik (OPV) und organische Leuchtdioden (OLED) zu Beleuchtungszwecken.

A. Transistoraufbau

Bei der Herstellung organischer Transistoren (OTFT= Organic Thin Film Transistor) wird, wie auch bei Dünnschichttransistoren, zwischen grundlegend verschiedenen Ausführungen unterschieden. Sowohl das Gate als auch Drain und Source können entweder direkt auf dem verwendeten Substrat liegen oder erst im späteren Verlauf der Herstellung aufgebracht werden. Unterschieden wird so zwischen Top- oder Bottom- Contact und/oder Gate. Bei den hier gezeigten Transistoren handelt es sich um eine Top-Gate, Bottom-Contact Topologie. Drain und Source liegen direkt auf dem Substrat, darüber Halbleiter und Dielektrikum, das Gate wird als letzte Schicht aufgebracht. Abb. 1 zeigt den Querschnitt des verwendeten Transistortyps. Als Substrate kommen Glas und flexible Folien (z.B. PET (Polyäthylenterephthalat,

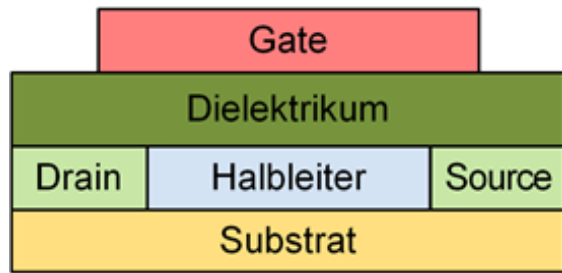


Abbildung 1: Querschnitt der verwendeten Transistoren (nicht maßstabsgetreu).

Hostaphan) oder PEN (Polyäthylennaphthalat)) zum Einsatz. Die Herstellung auf dünnen und flexiblen Substraten ist aufwändiger und fehleranfälliger als auf Glas, daher werden experimentelle Strukturen meistens auf Glassubstraten untersucht. Zur präzisen Charakterisierung der Halbleiter werden teilweise auch Si-Wafer eingesetzt, in diesen können z.B. die Elektroden für Drain und Source schon eingearbeitet sein.

Drain und Source werden auf Glas oder Folie meist aufgedampft, seltener gedruckt. Silber und Gold eignen sich wegen ihrer Austrittsarbeit relativ gut für die Verwendung als Elektroden in OTFT. Für Druckverfahren sind Tinten mit (Nano-) Metallpartikeln erhältlich, auch organische Verbindungen kommen zum Einsatz z.B. PEDOT:PSS.

Oft wird über den Elektroden eine weitere Schicht aufgebracht, sogenannte SAM (Self Assembling Monolayer), welche die Austrittsarbeit verändern und so Potentialbarrieren zwischen Halbleiter und Metall verringern. Eine schlechte Anpassung der Austrittsarbeit resultiert in hohen Kontaktwiderständen zwischen Elektrode und Halbleiter [1]. Als Halbleiter kommen verschiedene organische Verbindungen zum Einsatz, häufig verwendete Materialien sind z.B. Pentacen, P3HT (Poly-3-Hexylthiophen) und PTAA (Polytriphenylamin) [2], [3]. Neben den elektrischen Eigenschaften ist insbesondere die Löslichkeit und Prozessierbarkeit der Halbleiter wichtig. Manche organische Halbleiter lösen sich nur in sehr giftigen und aufwändig zu verarbeitenden Lösemitteln und sind daher für eine Serienfertigung auf Druckmaschinen nicht geeignet.

Als Dielektrikum werden ebenfalls unterschiedliche Verbindungen eingesetzt. Genauso wie der Halbleiter sollen diese ebenfalls flüssig-prozessierbar sein. Häufig verwendete Dielektrika sind z.B. PMMA (Plexiglas[®], Evonik Röhm GmbH) oder Cytos[®] (Asahi Glass) [4], [5]. Auf strukturierten Si-Wafern wird auch SiO₂ verwendet.

Als Gate werden prinzipiell die gleichen Materialien wie für Drain und Source verwendet, für transparente Anwendungen kommt auch das anorganische Indiumzinnoxid (ITO) zum Einsatz. Typische Dicken der einzelnen Schichten sind z.B.: Elektroden: 50-100 nm, Halbleiter: 20-100 nm, Dielektrikum: 500-1500 nm, Gate 50-100 nm.

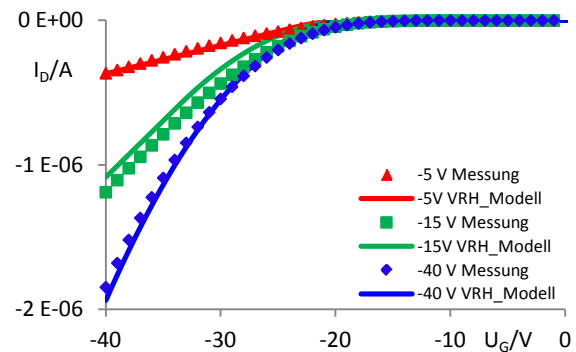


Abbildung 2: Transferkennlinien eines OFET, $U_{DS} = -5 \text{ V} / -15 \text{ V} / -40 \text{ V}$ Punkte sind Messwerte, durchgehende Linien sind simuliert.

B. Unterschied von OTFT zu anorganischen FET

Prinzipiell ist die Funktionsweise von organischen FETs sehr ähnlich zu herkömmlichen Si-Transistoren. Es gibt jedoch einige nennenswerte Unterschiede:

Anders als bei Silizium ist bei organischen Halbleitern die Beweglichkeit der Löcher üblicherweise höher als die der Elektronen. Deshalb beziehen sich die größten gemessenen Beweglichkeiten meistens auf P-Kanal-Transistoren. Organische Transistoren werden üblicherweise in Akkumulation betrieben, Inversion lässt sich nur in Ausnahmefällen erreichen. Auch die hier gezeigten Transistoren basieren auf P-Halbleitern und werden in Akkumulation betrieben.

C. Herstellung

Für die spätere großtechnische Herstellung ist die Verwendung von industriellen Druckmaschinen geplant, wie sie z.B. beim Druck von Zeitschriften und Verpackungen eingesetzt werden. Ausgehend vom Substrat sollen möglichst alle weiteren Schritte auf der gleichen In-Line-Maschine erfolgen. Aktuell wird jedoch im Labor meistens jeder Herstellungsschritt einzeln durchgeführt, was aufwendiges Positionieren nötig macht und die erreichbare Genauigkeit reduziert.

Drain und Source werden üblicherweise aufgedampft, entweder durch eine Schattenmaske, oder anschließend in einem Lithographieprozess strukturiert. Halbleiter und Dielektrikum werden meistens per Spincoating (Rotationsbeschichten) aufgetragen, oder mit kleineren Druckmaschinen aufgebracht. Im Spincoating wird die Lösung mit dem aufzutragenden Material auf das Substrat gegeben, anschließend wird dieses mit definierter Drehzahl und Dauer geschleudert. Die resultierende Schicht ist reproduzierbarer und homogener als aus Druckmaschinen, jedoch kann das Material nicht strukturiert aufgebracht werden. Daher ist dieses Verfahren für das Gate meistens ungeeignet, dieses wird üblicherweise aufgedampft oder gedruckt.

Tabelle 1: Simulationsparameter.

C_{Ox}	0,0232 fF/ μm^2
V_{Th}	-5,08 V
μ	0,02 cm^2/Vs
R_C	610 k Ω (W=10mm)
R_{off}	$5 \cdot 10^{12} \Omega$
λ	0,003 V^{-1}

D. Modellierung

OTFTs sind Feldeffekttransistoren (FETs) und lassen sich prinzipiell mit herkömmlichen Modellen beschreiben, auch wenn streng genommen die Gleichungen nur für Transistoren in Inversion gelten. Dennoch werden auch die in Akkumulation betriebenen OFET durch die Modelle gut beschrieben.

Zum Einsatz kommen das einfache MOS-Modell nach Shichman/Hodges [5] sowie das VRH-Modell (VRH = Variable Range Hopping) [7], welches für amorphe Materialien entwickelt wurde. Die Modelle wurden sowohl in MENTOR als auch in LTspice umgesetzt und liefern in beiden Simulatoren vergleichbare Ergebnisse. Die gezeigten Ergebnisse wurden mit dem VRH-Modell in LTspice simuliert. Erste Versuche haben ergeben, dass die vorhandenen Modelle zwar reale, flüssigprozessierte Transistoren prinzipiell beschreiben, jedoch gibt es nennenswerte Abweichungen zwischen Simulation und Messung [8]. Deshalb mussten die Modelle erweitert werden, um parasitäre Effekte einzuschließen. Vier Modellerweiterungen wurden implementiert:

1. Off Strom
2. Kanallängenmodulation
3. Kontaktwiderstände
4. Gate-Leckströme

Mit diesen Erweiterungen lassen sich Simulationen durchführen, deren Genauigkeit besser ist als die Serienstreuung der hergestellten Transistoren. Die Abweichungen zwischen zwei gleich hergestellten Transistoren sind oft größer als die Abweichungen zwischen Messung und Simulation mit den gemessenen Parametern. Abb. 2 zeigt exemplarisch den Vergleich von simulierter und gemessener Transferkennlinie bei $V_{DS} = 0 \text{ V}$, -15 V und -40 V . Die erreichte Genauigkeit der Simulation wird aktuell als ausreichend betrachtet, jedoch werden die Modelle mit zunehmender Verbesserung der Herstellungsprozesse ebenfalls weiter optimiert werden müssen.

Die Serienstreuung der Prozesse limitiert ebenfalls die erreichbare Komplexität einer Schaltung. Monte-Carlo Simulationen werden verwendet, um komplexe Schaltungen auf die erreichbare Ausbeute hin zu untersuchen.

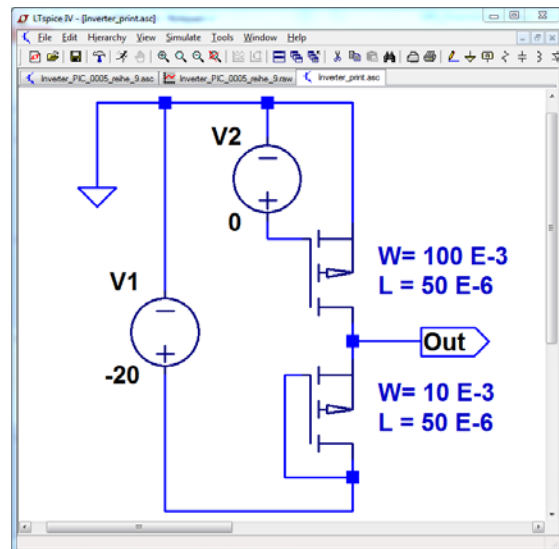


Abbildung 3: Schaltbild zur Simulation eines Inverters in LTspice.

E. Charakterisierung

Die für die Simulation notwendigen Parameter werden durch Messungen an aufgebauten Transistoren gewonnen. Nach einem genau definierten Standard werden Transfer- und Ausgangskennlinien gemessen [9]. Die vorgestellte Routine wird für Transistoren in Akkumulation verwendet und gilt für P-Kanal OTFT. Für N-Kanal OTFT gelten jeweils andere Vorzeichen. Transferkennlinien werden bei $V_{DS} = 0 \text{ V}$, -5 V , -15 V und -40 V aufgenommen, V_{GS} wird von $+5 \text{ V}$ bis -60 V in 1 V Schritten verändert. Ausgangskennlinien werden bei $V_{GS} = 0 \text{ V}$ bis -40 V in 10 V Schritten aufgenommen, V_{DS} wird von $+5 \text{ V}$ bis -60 V verändert. Zur Untersuchung von Hystereseeffekten werden alle Messungen in Vorwärts- und in Rückwärtsrichtung durchlaufen. Die Messungen werden mit SMUs (Source Measure Unit) der Fa. Keithley, Typ 2636A durchgeführt. Aus den Messdaten werden die notwendigen Simulationsparameter extrahiert. Die verwendeten Verfahren zur Ermittlung der Schwellspannung und Ladungsträgerbeweglichkeit μ sind in [9] beschrieben. Der differentielle Ausgangswiderstand wird direkt aus der Ausgangskennlinie ermittelt, weiter wird hierdurch die Early-Spannung ($V_E = 1/\lambda$) [10] berechnet. Tabelle 1 gibt einen Überblick über die extrahierten Parameter.

Um Schwankungen in der Serienfertigung zu untersuchen, wurde ein Reihemessplatz aufgebaut, der es erlaubt, vollautomatisch große Mengen von Transistoren zu charakterisieren. Präzise Einzel- und Temperaturmessungen wurden auf einem herkömmlichen Waferprober mit ThermoChuck durchgeführt.

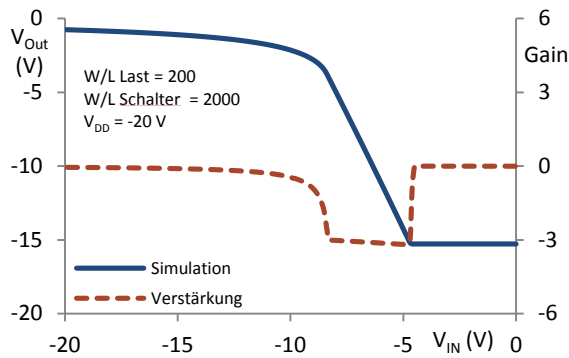


Abbildung 4: Kennlinien des simulierten Inverters aus Abb.3, Übertragungskennlinie und Verstärkung.

II. HERSTELLUNG DER SCHALTUNG

Ausgehend von früheren Messergebnissen wurden Schaltungen entwickelt, an denen die Qualität der Simulation untersucht werden soll. Der Herstellungsprozess und die große Parameterstreuung lassen zunächst keine große Schaltungskomplexität zu, deshalb wurde ein aus drei Transistoren bestehendes NAND-Gatter aufgebaut. In dieser Schaltung kann das Zusammenspiel der Einzeltransistoren weiter untersucht werden.

A. Simulation

Das NAND-Gatter besteht aus zwei Schalttransistoren und einem Lasttransistor, welcher als MOS Diode geschaltet ist. Ein NAND-Gatter kann als Inverter mit zusätzlichem Schalttransistor betrachtet werden. Daher wurden auch die Simulationen anhand eines einfacheren Inverters durchgeführt. Abb. 3 zeigt das Schaltbild der Inverterschaltung in LTspice, das Größenverhältnis zwischen Treiber und Last beträgt 1:10. Anhand dieser Schaltung soll die Simulation mit real hergestellten Schaltungen verglichen werden. Abb. 4 zeigt die simulierte Kennlinie des Inverters aus Abb. 3, die Versorgungsspannung beträgt $V_{DD} = -20$ V, die Eingangsspannung wird von $V_{in} = 0$ V bis -20 V variiert. Die Simulation zeigt, dass die erreichten Pegel bei ca. $V_{Out} = -1$ V und $-15,3$ V liegen. Die Verstärkung wird mit -3 simuliert.

B. Layout

Abb. 5 zeigt das Layout der Schaltung, es wurde in MENTOR gezeichnet, hierfür wurde eigens ein Device Generator eingerichtet, der aus den eingegebenen Parametern automatisch Transistoren passender Geometrie erzeugt [8].

C. Herstellung

Die hergestellten Transistoren sind in Abb. 6 gezeigt, sichtbar sind die Kontaktpads und die größeren Gate-Strukturen.

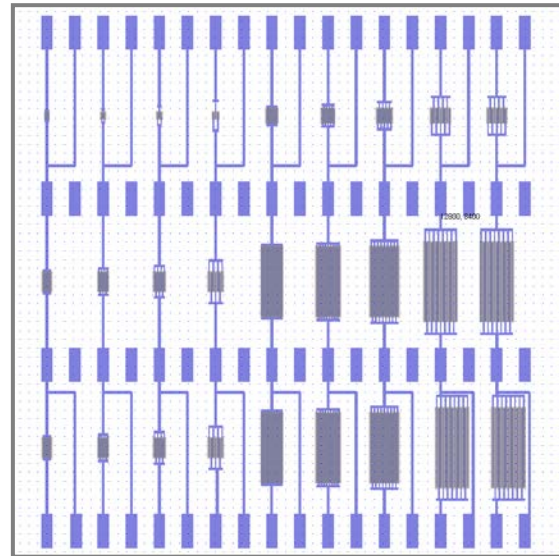


Abbildung 5: Layout mit 9 NAND-Gattern bzw. Invertern.

Ausgangsmaterial für die Fertigung sind $50 \mu\text{m}$ dicke Foliensubstrate mit strukturierten Drain- und Sourcekontakten aus Silber. Der polymere P-Halbleiter wurde in einem Lösungsmittel gelöst und per Spincoating aufgebracht. Das polymere Dielektrikum wurde ebenfalls per Spincoating aufgebracht, die Schichtdicke wurde mit einem DEKTAK Profilometer zu 877 nm bestimmt, die Dielektrizitätskonstante beträgt $\epsilon_r = 2,3$. Hieraus ergibt sich ein Gate-Kapazitäts-Flächenbelag von $C_{Ox} = 23,2 \mu\text{F}/\text{m}^2 = 0,0232 \text{ fF}/\mu\text{m}^2$. Das Gate besteht aus, durch eine Schattenmaske strukturiertem, aufgesputtertem Gold.

D. Messung der hergestellten Schaltung

Der hergestellte Inverter wurde auf einem Waferprober kontaktiert und mit SMUs angesteuert. Als Versorgungsspannung wurde wie bei der Simulation $V_{DD} = -20$ V gewählt. Die Eingangsspannung wurde von $V_{in} = 0$ V bis -20 V in $0,1$ V Schritten verändert. Durch die geringe Treiberfähigkeit der organischen Transistoren kann der Ausgang nicht direkt mit herkömmlichen Messgeräten gemessen werden. Bereits ein einfaches Voltmeter mit $20 \text{ M}\Omega$ Innenwiderstand belastet den Ausgang so stark, dass die Messung stark verfälscht wird. Daher wurde der Ausgang mit einem SMU Kanal gemessen, welcher als Stromquelle mit 0 A konfiguriert wurde. Simulationen zeigen, dass Ströme unter 10 pA das Verhalten der Schaltung nicht messbar beeinflussen. Die Stromquelle regelt den Strom auf die eingestellten 0 A, die sich einstellende Spannung wird gemessen. Hierdurch ergibt sich ein Voltmeter mit fast unendlichem Eingangswiderstand, welches den Inverter nicht belastet. Der Sollwert von 0 A wird nicht erreicht, der gemessene Strom beträgt im Mittel ca. $0,1$ pA, und verändert das Verhalten des Inverters nicht messbar.

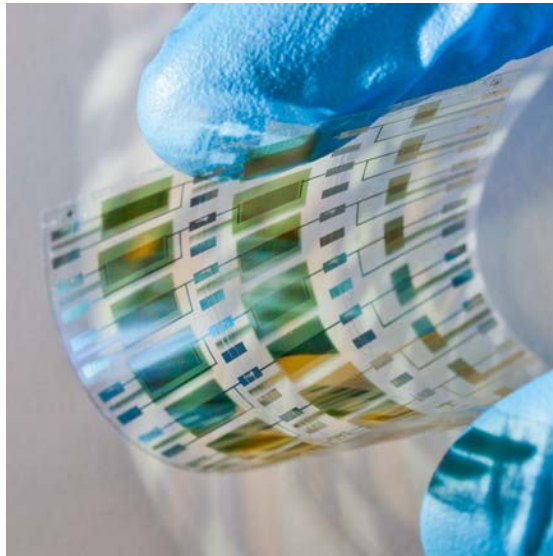


Abbildung 6: Prozessiertes Substrat, vgl. Abb. 5.

E. Auswertung

Abb. 7 zeigt den Vergleich von gemessener (Punkte) und simulierter (durchgehende Linie) Inverterkennlinie. Bei $V_{in} = -20$ V wird der Ausgangspegel mit $V_{out} = -0,75$ V simuliert und mit 0 V gemessen. Bei $V_{in} = 0$ V wird der Ausgangspegel mit $V_{out} = -15,3$ V simuliert und mit $-14,2$ V gemessen. Im aktiven Bereich zwischen $V_{in} = -5$ V und -8 V, in dem die Verstärkung -3 beträgt, zeigt sich eine Verschiebung der gemessenen Kennlinie um ca. $0,8$ V. Diese Unterschiede lassen sich mit Unterschieden in der Schwellenspannung zwischen Simulation und realen Transistoren erklären. Für die Simulation wurden mittlere Werte verwendet. Besonders erwähnenswert ist, dass die Verstärkung im aktiven Bereich sehr genau simuliert wurde.

FAZIT

Organische Elektronik hat das Potential, neue Anwendungen zu ermöglichen, welche bisher nicht durchführbar oder zu teuer waren. Die Transistoren lassen sich mit herkömmlichen Modellen simulieren, es sind jedoch Detailverbesserungen notwendig. Am Beispiel einer Inverterschaltung wurde die Herstellung von der Simulation bis zur späteren Charakterisierung gezeigt. Hierbei wurde deutlich, dass die verwendeten Modelle das Verhalten für den aktuellen Stand der Fertigung mit tolerierbaren Fehlern beschreiben können. Insbesondere die richtig simulierte Verstärkung lässt den Schluss zu, dass auch mehrstufige Verstärker zuverlässig simuliert werden könnten. Der Ausgangspegel zwischen $V_{in} = 0$ V und -5 V zeigt deutliche Unterschiede zur Simulation. In diesem Bereich ist der Schalttransistor ausgeschaltet und wird mit einem konstanten Widerstand simuliert, es wird erwartet, dass eine weitere Modellverbesserung in diesem Bereich die Simulationsergebnisse deutlich verbessert.

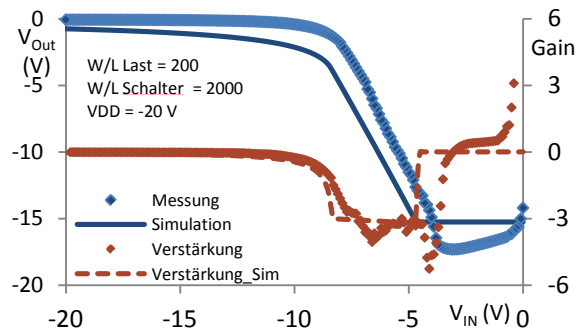


Abbildung 7: Simulierte und gemessene Kennlinien des Inverters aus Abb.3, Übertragungskennlinie und Verstärkung.

DANKSAGUNG

Die Autoren bedanken sich für die finanzielle Unterstützung des Projektes POLYTOS2 durch das Bundesministerium für Bildung und Forschung (Förderkennzeichen 13N12090), im Rahmen des Spitzenclusters „Forum Organic Electronics“, koordiniert von der Innovationlab GmbH.

LITERATURVERZEICHNIS

- [1] H. Klauk, et al. "Contact resistance in organic thin film transistors," *Solid State Electronics* 47 (2), S. 297–301, 2003.
- [2] P. Herwig, K. Müllen, "A Soluble Pentacene Precursor: Synthesis, Solid-State Conversion into Pentacene and Application in a Field-Effect Transistor" *Adv. Mater.* 11 (6), S. 480–483, 2009.
- [3] B. Gburek, V. Wagner, "Influence of the semiconductor thickness on the charge carrier mobility in P3HT organic field-effect transistors in top-gate architecture on flexible substrates" *Organic Electronics* 11 (5), S. 814–819, 2010, doi: 10.1016/j.orgel.2010.01.023.
- [4] J. Veres, et al., "Gate Insulators in Organic Field-Effect Transistors," *Chem. Mater.* 16 (23), S. 4543–4555, 2004.
- [5] Y. Yun, et al., "Pentacene thin film transistors with a poly(methyl methacrylate) gate dielectric: Optimization of device performance," *J. Appl. Phys.* 105 (3), S. 34508, 2009, doi: 10.1063/1.3075616.
- [6] H. Shichman, D.A. Hodges, "Modeling and simulation of insulated-gate field-effect transistor switching circuits," *IEEE Journal of Solid-State Circuits*, vol.3, no.3, pp. 285- 289, 1968, doi: 10.1109/JSSC.1968.1049902.
- [7] O. Marinov, et al., "Organic Thin-Film Transistors: Part I-Compact DC Modeling," *IEEE Transactions on Electron Devices*, vol.56, no.12, 2009, doi: 10.1109/TED.2009.2033308.
- [8] S. Hengen et al., "Modelling and Simulation of Organic Thin Film Transistor Circuits," *Proceedings of LOPE-C 2012*, München.
- [9] S. Hengen et al., "Mass Characterisation of OTFT," *Proceedings of LOPE-C, Frankfurt am Main*, 2011.
- [10] F. A. Lindholm, D. J. Hamilton, "Incorporation of the early effect in the Ebers-Moll model," *Proceedings of the IEEE* 59(9), S. 1377–1378, 1971 doi: 10.1109/PROC.1971.8435.



Stefan Hengen erhielt den akademischen Grad des Dipl.-Ing. (FH) in Nachrichtentechnik /Elektronik im Jahr 2008 von der Hochschule Mannheim. Dort erhielt er im Jahr 2010 den Master of Science in Informationstechnik. Seit 2009 ist er Wissenschaftlicher Mitarbeiter an der Hochschule Mannheim.



Bernd Scherwath erhielt den akademischen Grad des Bachelor of Science von der Hochschule Mannheim 2011. Dort erhielt er 2012 ebenfalls den Master of Science in Informationstechnik.



Christian Haas erhielt den akademischen Grad des Dipl.-Ing. (FH) von der Hochschule Mannheim 2011. Zurzeit arbeitet er an seiner Masterarbeit über den Aufbau eines Messplatzes für transiente Messungen an organischen Transistoren.



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



Jürgen Giehl erhielt den akademischen Grad Diplom-Physiker 1990 von der Universität Mainz und den Grad Dr.-Ing. in Elektrotechnik von der Universität Siegen im Jahr 1997.

Von 1997 bis 2007 hat er bei ITT Semiconductors (seit 1998 Micronas GmbH) in Freiburg i. Breisgau als Projektmanager und Designer für analoge Schaltungen gearbeitet.

Seit 2007 ist er Professor mit Lehrgebiet Entwurf integrierter Schaltkreise an der Hochschule Mannheim.

Design and Implementation of a Mixed-Signal ASIC in Ultra-Thin CMOS Technology for 2-D Stress Measurement

Yigit U. Mahsereci, Cor Scherjon

Abstract—A CMOS sensor for in-plane stress measurement is presented. The device is designed using the IMS 0.5 μm CMOS Gate Array technology and it is to be fabricated as a flexible chip using the ChipfilmTM technology of IMS CHIPS. The sensor consists of four channels, each of which includes a sensor core, a readout circuit, and a 10-bit analog-to-digital converter (ADC). The linearity of the sensor is enhanced by the design of the sensor core and readout. The readout is designed to be insensitive to the induced stress in order to maintain proper operation at all times. The internal sensor timing can be tuned through a serial interface for adjusting the analog performance of the sensor.

Index Terms—Stress sensor, ultra-thin chip, flexible chip, stress-insensitive mixed-signal design

I. INTRODUCTION

Flexible electronics is a contemporary technology that is likely to grow rapidly in near future. The hybrid system-in-foil (SiF) devices, that house an ultra-thin chip, are expected to be an important part of the market [1]. By using the properties of the ultra-thin chips, stress on a surface can be measured or topological information can be acquired from the adjacent surface.

The piezoresistive effect in silicon is a well explained phenomenon which is widely used for stress sensing. Resistors or MOSFET's are used as sensing elements. The information is obtained by measuring the stress induced variation in resistance and mobility for resistors and MOSFET's respectively. The choice differs according to the final application of the sensor. In literature, a number of studies are available and the target applications cover various topics such as dental care, wire bonding, flip chip bonding, and wafer stability [2]-[6].

In this work, a sensor is designed for measuring the in-plane stress or the flex on a bendable surface. Due to the extremely low thickness of the device, the rigid-

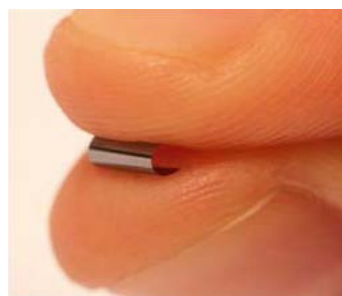


Figure 1: Ultra-thin (thickness $< 20 \mu\text{m}$) silicon chip. The chip is fabricated with ChipfilmTM technology of IMS CHIPS [10].

ity of the silicon is negligible; therefore the data of the surface state can be harvested without distortion. Despite the physical advantage, ultra-thin chips have intrinsic problems that are to be taken care of such as, substantial stress in the devices and vulnerability to thermal variations due to low heat capacity. Also, the out-of-plane stress component must be eliminated so that in-plane data is not corrupted. The readout circuit is designed to be insensitive to stress, thermal fluctuations and the out-of-plane stress component. The sensor provides 10-bit digital output for four different stress sensitive MOSFET pairs. The enhanced linearity of the sensor makes it possible to accurately calculate the 2-D polar coordinates of the stress vector.

II. GATEFORESTTM TECHNOLOGY

The IMS CHIPS GateForestTM technology provides key features such as quick fabrication cycle, mixed-signal integration, on-chip RAMs at a process node of 0.8 μm or 0.5 μm CMOS technology [7], [8]. Various standard die sizes, so-called master chips are offered. Each master includes a different number of digital and analog core cells, which enables mixed-signal integration. All wafers are already processed until the metallization and are residing in the foundry. By processing different metallization layers on top of the pre-processed master wafers, different customer-specific designs can be realized. Hence, the complete process cycle is reduced to only back-end-of-line. Therefore, even small volume fabrications can be made in a low-cost manner.

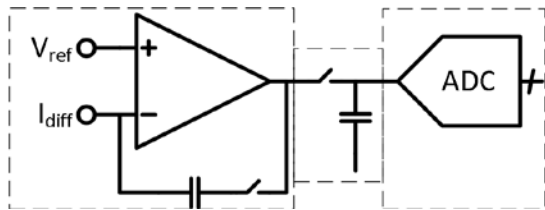


Figure 4: Readout circuit diagram. The chain consists of an integrator, track-and-hold stage and an ADC.

B. Readout

The readout chain consists of an integrator, a track-and-hold stage, and an ADC. Figure 4 shows the readout diagram.

The stability of voltage at the output node of the sensor core significantly affects the linearity performance. Hence, an active stage with feedback is preferred to keep this voltage constant during operation.

The integrator topology has a number of advantages for this task. First, it serves as a low-pass filter and suppresses noise. Second, the voltage of the input node is kept constant due to feedback. Finally and most importantly, the feedback loop only accommodates a capacitor which is stress insensitive. The feedback behavior remains the same under varying stress unlike resistive feedback structures.

A two stage miller op-amp is used for the integrator. The amplifier is designed to remain within acceptable limits under stress values up to 200 MPa. This may cause a mobility change up to $\pm 15\%$, depending on the direction of the stress vector. The fundamental practice is to place all transistors of the same type in the same direction. Therefore, the parameter changes in the differential branches are cancelled out and it is easier to take account of a global mobility change for all transistors of the same type. Further measures are taken for higher stress insensitivity, as gain and phase margin are explicitly affected. The phase margin is pushed above 60° at the cost of higher power consumption. The minimum gain is set to be 80 dB for minimizing the voltage drift at the input node under stress.

The track-and-hold circuit is intrinsically not affected by stress. The size of the capacitor is selected considering the load of the integrator and the kickback of the ADC. Complementary switches with charge injection cancellation are used.

A 10-bit successive approximation register ADC from the IMS CHIPS intellectual property library is used for digitization. The ADC has a rail-to-rail input range and requires 20 clock cycles for one complete conversion. The stress insensitivity of the device is established by careful layout and design measures.

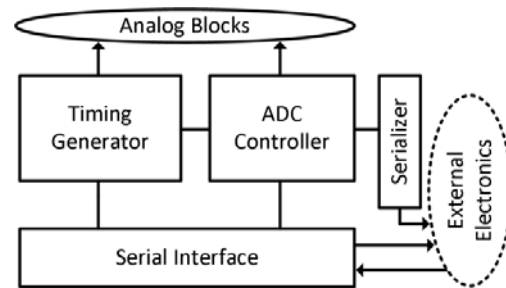


Figure 5: Digital controller block diagram. Conceptual diagram is deduced from VHDL code and simplified for clarity.

C. Digital Controller

The management of the analog blocks and external communication is to be done by the designed controller. It is written in VHDL and realized using silicon compiling tools. Figure 5 shows the controller block diagram.

The readout timing determines the post-readout calculation and affects the overall performance. The controller supports external programming which enables testing different cases on silicon. The timing can be adjusted as precise as one pulse of system clock. Fine tuning of pulse widths and sequence of activities enable optimization of important system performance figures such as sensor bandwidth, noise performance, and dynamic range.

The number of pads is minimized by serializing the sensor I/O's. The data exchange with external electronics is done via two serial synchronous interfaces that are used for sensor configuration and ADC data read-out.

IV. IMPLEMENTATION

Several intrinsic and extrinsic restrictions are present in layout level due to the gate array technology and stress exposure. All circuits except stress sensor cores are constructed with same-oriented transistors due to gate array technology. Differential branches in the analog and digital circuitry benefit from uniform alignment as stress effects are nulled out. The stress sensor cores are built using the special [110], $\bar{1}10$, [100], and [010] oriented transistors in the used gate array master. Figure 6 illustrates the orthogonal transistor pairs in the master used for this work.

Analog and digital circuits are implemented in their indigenous regions resulting from the technology. The wells and supplies of analog and digital domains are separated in order to minimize crosstalk. Chip size is 4.5 mm x 4.5 mm.

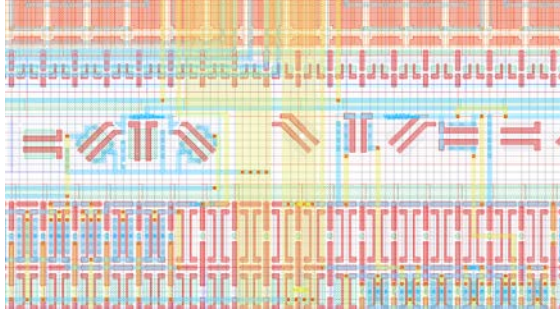


Figure 6: Orthogonal transistors on gate array master GFQ032D.

V. VECTOR CALCULATION

The stress is calculated as a vector on a 2-D polar coordinate system with an angle and a magnitude. The ideal readings from the sensor are given in equation (2). The stress tensor components in this table can be represented by angle and equivalent vector magnitude as here

$$\begin{bmatrix} \sigma_X \\ \sigma_Y \\ \sigma_{XY} \end{bmatrix} = \begin{bmatrix} \cos^2 \varphi \\ \sin^2 \varphi \\ \sin \varphi \cdot \cos \varphi \end{bmatrix} \cdot \sigma \quad (3)$$

Rearrangement of equation (2) using equation (3) results in

$$\begin{aligned} \Delta I_1 &= C_1 \cdot \sin 2\varphi \cdot \sigma \\ \Delta I_2 &= C_2 \cdot \cos 2\varphi \cdot \sigma \end{aligned} \quad (4)$$

where $C_1 = -\pi_{44} \cdot I_{D1}$ and $C_2 = -(\pi_{11} - \pi_{12}) \cdot I_{D2}$. The constant terms and the currents are known in equation (4), hence the stress magnitude is calculated by

$$\sqrt{\left(\frac{\Delta I_1}{C_1}\right)^2 + \left(\frac{\Delta I_2}{C_2}\right)^2} = |\sigma| \quad (5)$$

The angle calculation is done by inverse trigonometric functions after the magnitude is derived. Certain offset and gain errors, due to sensor non-idealities, can be characterized and corrected in the calculation phase.

VI. SIMULATION RESULTS

Overall circuit behavior is mainly affected by stress acting on the sensor. Non-linearity is the main problem as precision must be maintained for the accuracy of post-readout mathematical operations. Figure 7 illustrates the deviation from the ideal characteristics for one channel of readout. The total deviation corresponds to two counts as the LSB is 4 mV. The effect of circuit nonlinearity on the vector calculation

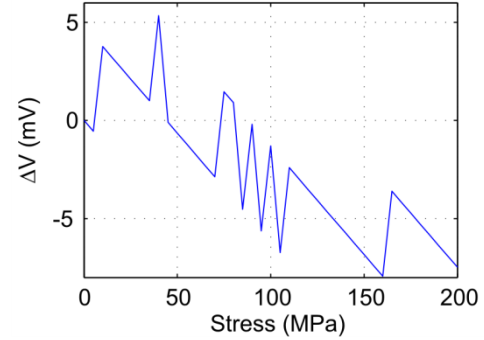


Figure 7: The simulated sensor output deviation from ideal case. The stress effect is taken into account for all components.

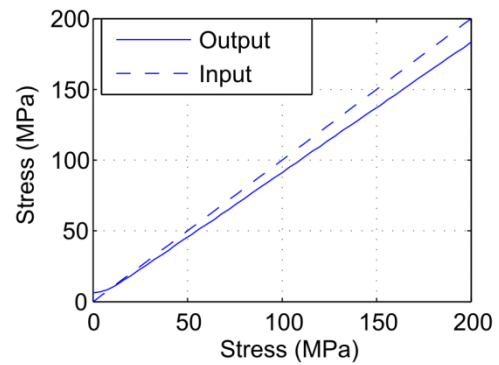


Figure 8: Stress magnitude calculated from the simulated sensor output vs. applied stress. The result is acquired without any offset correction. The stress is applied at a constant 22.5° angle.

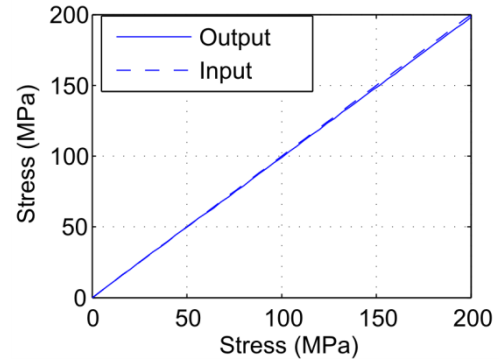


Figure 9: Simulated stress with correction vs. applied stress. The deviation is limited to 2 MPa in the 200 MPa range. Stress angle is 22.5°.

is negligible. Figure 8 shows the applied and the calculated stress on the sensor. The relatively large deviation of the sensor output is due to the offset errors which can be simply corrected in the calculation phase. The corrected data is given in Figure 9 and Figure 10. The nonlinearity of the corrected stress magnitude and angle are 2 MPa and 0.5° respectively.

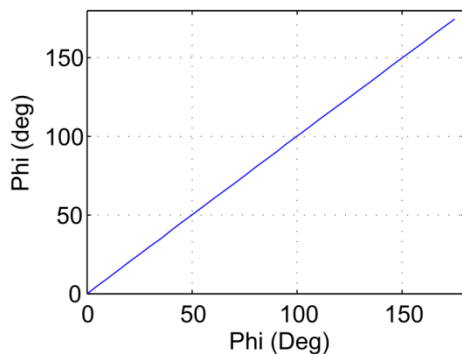


Figure 10: Calculated angle with correction vs. applied angle. A constant stress magnitude of 100 MPa is applied. The total deviation is limited to 0.5° in the 180° span.

VII. CONCLUSION

A stress sensor, for measuring 2D in-plane stress components, is presented. The sensor is to be realized using the chip-film technology; therefore it can be used to detect stress on a flexible surface. The sensor operation is not affected by flexing, due to stress insensitive design. Linearity under stress is maximized for the accuracy of stress vector calculations.

ACKNOWLEDGEMENT

The authors would like to thank H. Richter, C. Burwick, N. Wacker, S. Hirsch and H. Rempp for their support along the way.

REFERENCES

- [1] J. Burghartz, W. Appel, C. Harendt, H. Rempp, H. Richter and M. Zimmermann, "Ultra-thin chips and related applications, a new paradigm in silicon technology," in *Proceedings of the European Solid State Device Research Conference*, 2009.
- [2] Y. Shi, C. Ren, W. Hao, M. Zhang, Y. Bai and Z. Wang, "An Ultra-Thin Piezoresistive Stress Sensor for Measurement of Tooth Orthodontic Force in Invisible Aligners," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1090-1097, 2012.
- [3] M. Kuhl, P. Gieschke, D. Rossbach, S. Hilzensauer, P. Ruther, O. Paul and Y. Manoli, "A telemetric stress-mapping CMOS chip with 24 FET-based stress sensors for smart orthodontic brackets," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2011 *IEEE International*, vol., no., pp.108-110, 2011.
- [4] M. Doelle, C. Peters, P. Gieschke, P. Ruther and O. Paul, "Two-dimensional high density piezo-FET stress sensor arrays for in-situ monitoring of wire bonding processes," in *Micro Electro Mechanical Systems, 2004. 17th IEEE International Conference on MEMS*, vol., no., pp. 829- 832, 2004.
- [5] C. Jiang, F. Xiao, H. Yang and C. Dou, "Application of Silicon Stress Sensor in Flip Chip Packaging System," in *International Conference on Electronic Packaging Technology & High Density Packaging*, 2011.
- [6] X. Zhang, R. Rajoo, C. Selvanayagam, A. Kumar, V. Rao, N. Khan, V. Kripesh, J. Lau, D.-L. Kwong, V. Sundaram and R. Tummala, "Application of Piezoresistive Stress Sensor in Wafer Bumping and Drop Impact Test of Embedded

Ultrathin Device," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 2, no. 6, pp. 935-943, 2012.

- [7] C. Burwick and C. Scherjon, "Structured ASICs für Mixed-Signal Anwendungen," in *Multi Projekt Chip Gruppe Baden-Württemberg*, Künzelsau, 2009.
- [8] Institut für Mikroelektronik Stuttgart, "IMS CHIPS Web Site," 2011. [Online]. Available: http://www.ims-chips.de/content/pdf/text/White_paper_MS_Array_09_11.pdf [Accessed Feb. 2013].
- [9] W. Nicoleta, R. Harald, H. Mahadi-UI, R. Horst and B. Joachim N., "Compact modeling of CMOS transistors under variable uniaxial stress," *Solid-State Electronics*, vol. 57, no. 1, pp. 52-60, 2011.
- [10] J. Burghartz, W. Appel, H. Rempp and M. Zimmermann, "A New Fabrication and Assembly Process for Ultrathin Chips," *IEEE Transactions on Electron Devices*, vol. 56, no. 2, pp. 321-327, 2009.



Yigit U. Mahsereci received the M.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Turkey, in 2012. Between 2009 and 2012, he was working in Mikro-Tasarim, Ankara, Turkey, as an IC designer for CMOS image sensors.

In 2012, he joined IMS CHIPS and he is working towards a Ph.D. degree in the area of stress sensors and stress management in ultra-thin CMOS chips.



Cor Scherjon received the M.Sc. degree in electrical engineering from the University of Twente, The Netherlands, in 1997. He has been with the Institut für Mikroelektronik Stuttgart since 1997. He has participated in several international projects in the field of embedded systems and ASIC development. He has been involved in developing and testing high-dynamic range image sensors since 2000. Currently, he is focusing on the development of mixed-signal ASICs.

DC Forward Model of High-Power LED's

Volker Lange, Brenton Sherston, Robert Hönl

Abstract—Exact models of electronic components are necessary for the development of electronic circuits. Manufacturers of light emitting diodes (LED) make only standard diode Spice models available, particularly for the active region of LED's, the DC forward voltage range. In our work we present a DC-model for the forward active region, which exactly characterizes the behavior of high-power LED's in a large current/voltage range.

Index Terms—High-power LED, DC forward model, Spice model.

I. INTRODUCTION

The development of high-power LED's makes their usage in measurement and signal technology as nearly monochromatic light sources possible. Furthermore, high-power LEDs are used as white light sources in lightning technology, where they have a very high potential due to their high energy efficiency in contrast to conventional light sources [1].

Basically, a LED is nothing else than a diode and can be described with the Spice model of a pn junction [2]. The DC forward Spice parameters are the saturation current I_S , the emission coefficient N and the serial resistance R_S . Only these basic parameters are in general provided by manufacturers of LED's. Neither the temperature behavior, which is described in the Spice model by the saturation current temperature exponent X_{TI} and the energy gap E_G nor deviations from the ideal diode behavior like generation-recombination and high-level injection effects [3] can be modeled by the information given by the manufacturer. The knowledge of the exact electrical description and the temperature behavior for LEDs working under different ambient conditions are necessary to be considered in electric circuit development to increase e.g. energy efficiency.

In the following, we present the development of a general model for the entire DC forward operating range of a LED. We applied the model to different high-power LED's, showing the universality of the

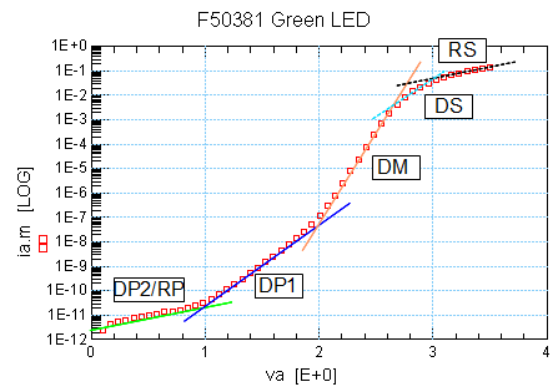


Figure 1: Forward IV- characteristic of a green light emitting LED

model. First results on temperature effects and ambient light influence will be shown.

II. EXPERIMENTS

Measurement and data analysis has been done with Agilent Technologies ICCAP software (rev. 2012.01). With this tool it is possible to drive the measurement equipment, acquire the data and to perform the Spice analysis. For this purpose, several tools like sensitivity analysis and parameter optimization algorithms are available. We used high accuracy source monitor units (HP4142B) to take the current voltage data. Current measurements down to pA with a resolution of tenth of fA with 1% accuracy are possible. We made measurements on different high power LED's from Seoul Semiconductor F50381 RGB-LED. Temperature control was performed with the LED mounted on a Temprotronic TP34 controlled temperature chuck. The temperature was additionally measured with a KTY-10 temperature sensor in direct contact with the LED.

III. MODEL DEVELOPMENT

The development of the DC forward model will be discussed on a measurement example of a LED emitting green light of 533 nm (figure 1). It is obvious, that such an IV characteristic cannot be characterized by a simple diode model. Modeling such curve, we take into consideration the influences of only standard electronic components on the characteristic curve [1] and follow a step by step procedure for the diode

Volker Lange, volker.lange@hs-furtwangen.de, Brenton Sherston, brenton.sherston@hs-furtwangen.de, Robert Hönl, robert.hoenl@hs-furtwangen.de, Hochschule Furtwangen, Computer and Electrical Engineering, Robert Gerwig Platz 1, 78120 Furtwangen.

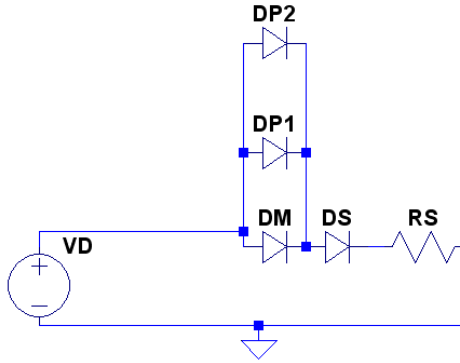


Figure 2: LED model for DC forward characteristic

modeling [4]. In the semi logarithmic plot of the IV characteristic of the LED, each linear segment corresponds to the exponential behavior of one diode. A higher current at a given voltage stands for a parallel diode while a higher voltage at a given current stands for a serial element (diode or resistance). This results in the electrical circuit consisting of ideal electronic components given in figure 2. Three parallel components, the main diode DM and two additional diodes DP1 and DP2. The flattening of the curve for high currents is usually described by the series resistance RS. The adaption of the curve at the transition from the main diode to the series resistance makes an additional serial diode DS necessary.

The sensitivity analysis tool included in ICCAP gives valuable hints on the overall influence of the individual device parameters as well as on the region of influence of the parameters on the complete characteristic curve. In figure 3 we show the sensitivity of the parameters together with the IV characteristic of the LED. We conclude that the emission coefficients N of the main diode DM and the diode DP1 have the strongest influence in well localized regions of the IV characteristic. All other parameters have much less influence. IS of the serial diode DS has the weakest influence of all parameters. Due to the fact that the parameters have a strongly localized influence, all elements must be considered in this example. Consequently, initial values for the parameters can be taken with high accuracy from the region of influence of the component described by the parameters. This is common to the diode parameter determination in the semi logarithmic IV-curve by fitting a linear curve to the data [2]. The intersection with the y-axes gives the logarithm of IS and the slope corresponds to $m = 1/(2.3 N V_t)$ with the thermal voltage $V_t = kT/q$. ICCAP further allows comfortable parameter tuning by simultaneously paying attention to the absolute or relative RMS error. This gives accurate starting values for all parameters. This is necessary for reliable pa-

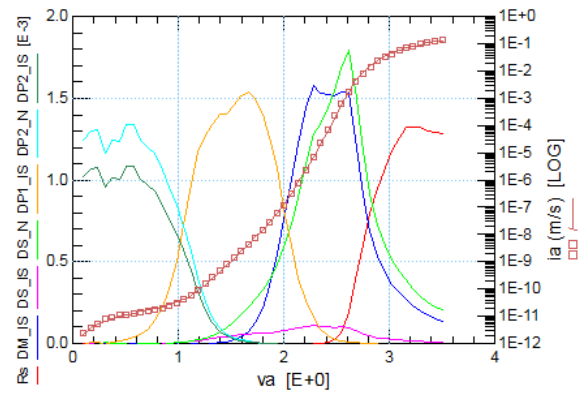
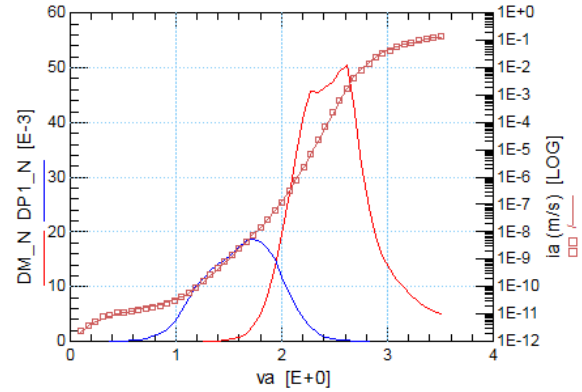


Figure 3: Sensitivity plot for the emission coefficient N of the main diode DM and parallel diode DP1 (upper diagram) and saturation currents IS of all diodes, emission coefficient N of DS and DP2 and Rs (lower diagram) together with the measured IV-curve (dotted line).

parameter optimization in such a multi parameter analysis problem by the Levenberg-Marquardt algorithm available under ICCAP.

IV. RESULTS

A. LED characteristics

We have investigated the DC forward characteristic of three LED's of the Seoul Semiconductor F50381 RGB-LED, the red light emitting (622 nm), the green light emitting (533 nm) and the blue light emitting (460 nm) diode. Guided by the eye and with the help of the sensitivity analysis, it is easy to decide whether all electrical components of the circuit in figure 2 are necessary. We found that the serial diode DS and/or one parallel diode DP2 can be removed from the general model in some cases and that the parallel diode DP2 can be sometimes replaced by a resistor RP as already indicated in figure 1. This will be discussed in the following.

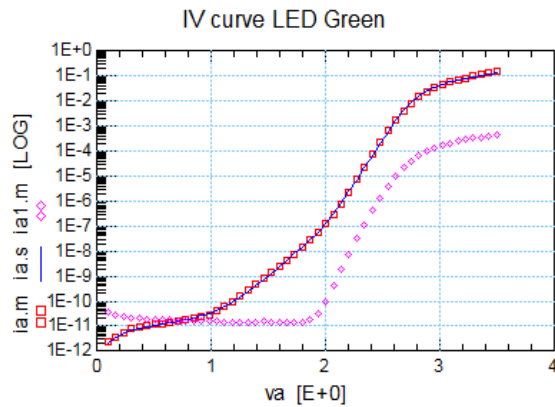


Figure 4: IV-characteristic of green light emitting LED, measured data (red dot) together with simulated curve (full line) after parameter optimization and photodiode data showing the light emission.

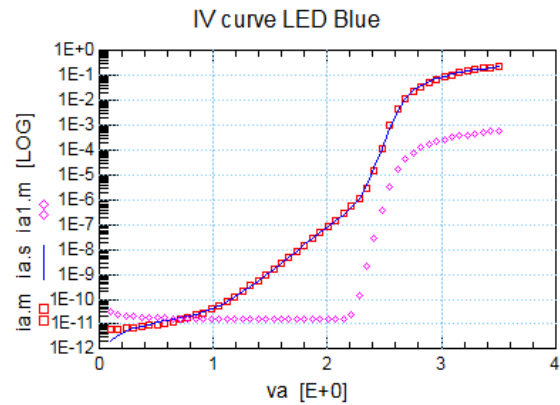


Figure 6: IV-characteristic of blue light emitting LED, measured data (red dot) together with simulated curve (full line) after parameter optimization and photodiode data showing the light emission.

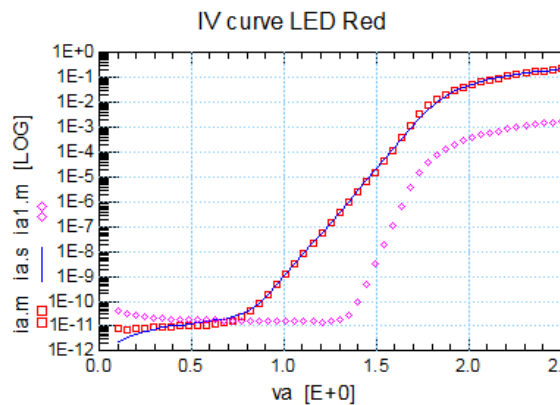


Figure 5: IV-characteristic of red light emitting LED, measured data (red dot) together with simulated curve (full line) after parameter optimization and photodiode data showing the light emission.

The application of the general model (figure 2) on the green light emitting diode gives the curve shown in figure 4. Measured data can only be reproduced by simulation based on the complete model. Due to the fact that the emission coefficient of the parallel diode DP2 amounts to close to $N = 500$, which corresponds to a slope of the curve in this section close to zero, diode DP2 can be replaced by a resistor RP with very similar simulation results. This alternative model has the advantage of less model parameters and can always be applied when the slope of the IV characteristic in the low voltage region is close to zero.

To model the red light emitting diode (figure 5) the general model can be simplified as following. Only the main diode DM together with the serial resistance is necessary to model the active region of the LED. The horizontal sub threshold part of the IV characteristic can be modeled with a parallel resistance RP. An additional serial diode DS will give only small improvement in the reproduction of the measured data by the model. In summary the model consists only in the elements RS, DM and RP.

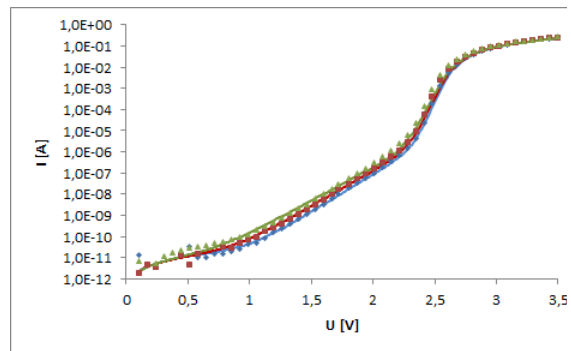


Figure 7: Temperature dependence of the IV-characteristic of blue light emitting LED, measured data (dotted) together with simulated curves (full line) at 30°C (blue), 50°C (red) and 70°C (green).

Finally, data of the blue light emitting diode are given in figure 6. As for the green light emitting diode two parallel electrical components in addition to the main diode DM are necessary. The not vanishing slope of the IV characteristic in the low-voltage region indicates that the parallel diode DP2 cannot be replaced here by a parallel resistor. From the sensitivity analysis we can conclude that the serial diode DS is not necessary to model the IV characteristic of this LED.

B. Temperature dependence

LED's especially used in outdoor technology will work under very different ambient temperature conditions. Consequently it is necessary to verify the validity of the model for different temperatures. Main Spice parameters describing the temperature dependence of the saturation current $IS(T)$ are the energy gap EG and the saturation current temperature exponent XTI [2]. Consequently, for an accurate description of the temperature dependence the knowledge of EG, which is given by the material composition of the LED [1] is necessary. This knowledge is not available for us. However, the energy gap is related to the wavelength

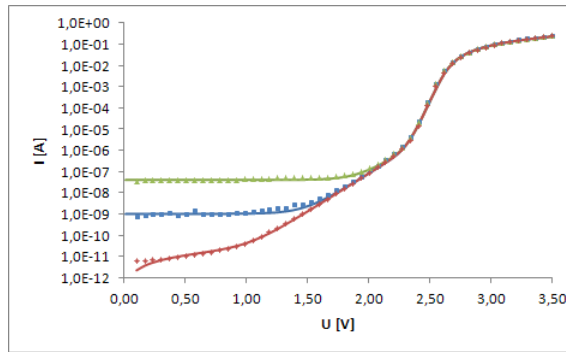


Figure 8: Measured (dots) and simulated (full line) IV-characteristic of blue light emitting LED at different light levels of ambient light.

of the emitted light. Therefore we could take approximate values for the minimum energy gap E_G for the simulation. Results are given in figure 7 showing the correct behavior. Only in the low IV region differences between measured and simulated curves are observed. More detailed temperature analysis will be possible with manufacturer data on the material composition of the LED's.

C. Ambient light influence

LED's are often utilized in environment with ambient light. The pn junction of the LED with applied voltage such that the LED emits light will also be sensitive to ambient light. This means that the ambient light produces some photocurrent in the pn junction of the LED and the LED detects the ambient light [5]. This will influence the IV-characteristic in the region of low IV values strongly. In figure 8 we show measured and simulated data of the IV characteristic of the blue light emitting diode under increasing ambient light level. For the simulation of the generated photocurrent, we introduced a current source in the circuit (figure 9). The curves clearly show that our model still holds for LED's working under ambient light illumination.

V. SUMMARY

In the present work we have developed a general model for high-power LED's based on elementary ideal electronic circuit elements. We have shown that the model

- can be applied to LED's of different type with only minor model modifications,
- describes well the temperature behavior,
- is also suitable to characterize the LED under ambient illumination.

ACKNOWLEDGEMENT

The authors thank the MPC group Baden – Württemberg for financial support.

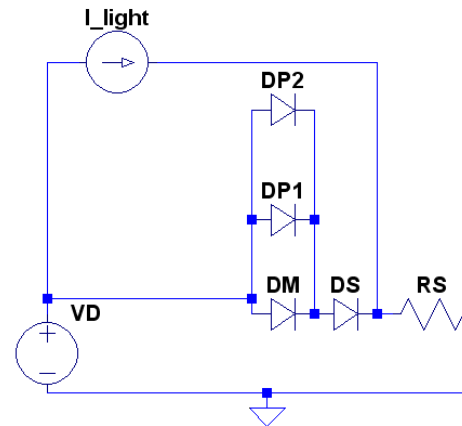


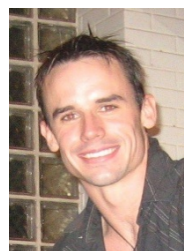
Figure 9: Electric circuit for simulation of LED with ambient illumination.

REFERENCES

- [1] E.F. Schubert, "Light-Emitting Diodes", *Cambridge University Press*, 2010.
- [2] G. Massobrio and P. Antognetti, "Semiconductor Device Modeling with Spice", *McGraw-Hill*, 1998.
- [3] S.M. Sze, "Physics of Semiconductor Devices", *John Wiley & Sons*, 1981.
- [4] F. Sischka, "Diode Modeling", *Agilent Technologies*, 2003.
- [5] V. Lange, F. Lima and D. Kühlke, „Multicolor LED in Luminescence Sensing Application”, *Sensors and Actuators A* 169, 2011, pp. 43-48.



Volker Lange received his diploma degree in physics and his Ph.D. from the University of Freiburg. Since 1991, he is a scientific assistant at Furtwangen University, faculty Computer and Electrical Engineering. His scientific interests are semiconductor measurement technology and optical sensor and measurement technology.



Brenton Sherston studied Electrical and Electronic Engineering at Murdoch University in Perth. He majored in Electrical Power and Industrial Computer Systems Engineering. He performed his Bachelor of Engineering Thesis at Furtwangen University in Germany in 2012. He currently works for First Quantum Minerals as an Electrical Engineer in Zambia, where he is responsible for the mining operations electrical power grid.



Robert Hönl received his diploma degree in Electrical Engineering and his Dr. Ing. from University of Berlin. Until 1993 he was product development manager at Krone AG. Now, he is a professor at Furtwangen University, faculty Computer and Electrical Engineering. His scientific interests are measurement-, sensor-, control engineering and project management.

A Current-Mode Buck-Boost DC-DC Converter with Fast Transient Response

Andreas Ehrhart, Bernhard Wicht, Ke-Horng Chen

Abstract—A fast transient current-mode buck-boost DC-DC converter for portable devices is presented. Running at 1 MHz the converter provides stable 3 V from a 2.7 V to 4.2 V Li-Ion battery. A small voltage under-/overshoot is achieved by fast transient techniques: (1) adaptive pulse skipping (APS) and (2) adaptive compensation capacitance (ACC). The proposed converter was implemented in a 0.25 μm CMOS technology. Load transient simulations confirm the effectiveness of APS and ACC. The improvement in voltage under-shoot and response time at light-to-heavy load step (100 mA to 500 mA), are 17 % and 59 %, respectively, in boost mode and 40 % and 49 %, respectively, in buck mode. Similar results are achieved at heavy-to-light load step for overshoot and response time.

Index Terms—DC-DC converter, buck-boost converter, fast transient, adaptive pulse skipping, adaptive compensation capacitance.

I. INTRODUCTION

Nowadays, there are lots of portable devices, such as mobile phones, notebooks, smartphones etc. and the number of them is still increasing. To have a competitive device, it is necessary to provide efficient power management to ensure a long battery life time. Therefore, DC-DC converters are used to power battery-operated portable systems. These converters can provide a constant output voltage (e.g. 3 V). To be able to use the whole battery voltage (e.g. Li-Ion battery with a voltage from 2.7 V to 4.2 V), buck-boost converters get more and more interesting [1]-[4]. To realize three different operation modes, buck, boost and buck-boost, the H-bridge topology of Fig. 1, consisting of switches M_A to M_D , is widely used in the design of buck-boost converters.

Fig. 2 shows the current paths in the different modes. *Path I* and *II* are used in the buck mode and *path III* and *IV* are used in the boost mode. Besides, the signals of the duty cycle D and its complement D'

Andreas Ehrhart, andreas ehrhart@gmail.com and Bernhard Wicht, Bernhard.Wicht@Reutlingen-University.DE are with Reutlingen University, Robert Bosch Center for Power Electronics, Alteburgstraße 150, 72762 Reutlingen. Ke-Horng Chen, khchen@faculty.nctu.edu.tw, is with National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan 30010, R.O.C.

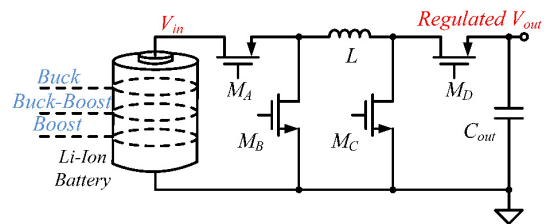


Fig. 1: Topology of the H-bridge buck-boost DC-DC converter.

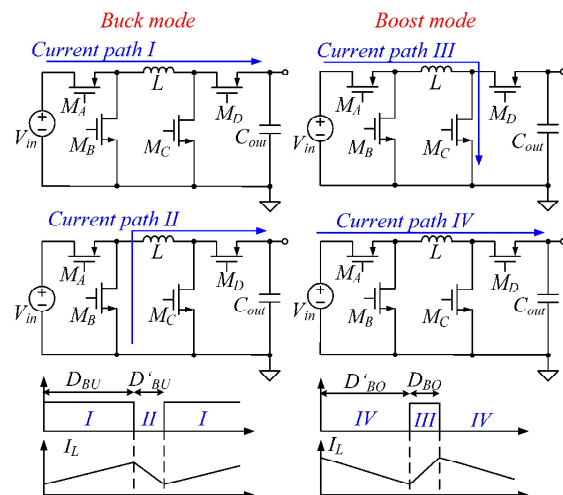


Fig. 2: Current paths and duty cycle in buck and boost mode.

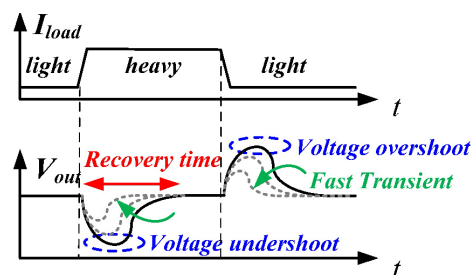


Fig. 3: Output voltage variations at load transition from light-to-heavy and vice versa.

are shown with the inductor current I_L .

The challenge for buck-boost converters is to have a high-quality and stable power management. For example if the supply voltage might get unstable due to load variations, it might cause abnormal operation or deteriorate the performance of the portable device. Therefore, a good transient response is a critical point of the design specifications. Also, it is necessary to

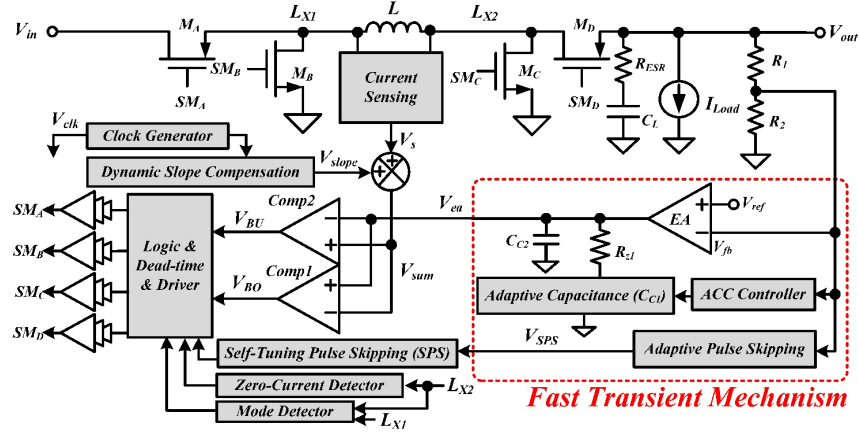


Fig. 4: Structure of the proposed current-mode buck-boost DC-DC converter with APS and ACC.

have a low voltage over-/undershoot at load variations, to improve the transient time.

Fig. 3 shows the effect of a load transition from light-to-heavy load and vice versa. There exist several techniques to improve the transient time, but they are only either for buck or for boost converters [5]-[7]. In buck-boost converters, it is much more complex to achieve fast transient response because the converter can work in three different modes, buck, boost and buck-boost. For each operation mode, the system stability has to be ensured. That means in steady state, the output voltage has to be kept always at its regulated value. Further, if there is a load variation, either from light-to-heavy load or vice versa, the output voltage has to be brought back to its regulated value without causing oscillations.

In [5] and [6], fast transient techniques for buck converters are presented. In these cases, fast transient is reached with an adaptive phase margin (APM) and an adaptive pole-zero position (APZP) technique. The compensation pole-zero pair can be dynamically moved towards high frequencies during transient period. After the output voltage is regulated to its steady-state value, the compensation pole-zero pair is smoothly moved back to the steady-state position. Thus, the overshoot/undershoot voltage can be decreased. Besides, owing to the techniques, the compensation zero is moved to nearly close to the position of the system pole to achieve the pole-zero compensation.

In [7], a fast transient technique for boost converters is presented. This modified hysteretic current control (MHCC) provides the limits for the output voltage. These limits can be adjusted at load changes, to achieve fast transient. Further, the system is able to regulate the output of the error amplifier and provide a good phase margin in steady state. In addition, the compensation technique is also adaptively, with the result that the compensation poles and zeros can be adaptively adjusted in the load transient and steady-state.

In this paper, a current-mode buck-boost DC-DC converter with adaptive pulse skipping (APS) and

adaptive compensation capacitance (ACC) is proposed to realize a fast transient response. In Section II, the structure of the proposed converter with the fast transient techniques is described. Section III shows the circuit implementation. Simulation results are presented in Section IV and finally conclusions are made in Section V.

II. PROPOSED BUCK-BOOST CONVERTER WITH FAST TRANSIENT TECHNIQUES APS AND ACC

A. System Overview of the Proposed Converter

Fig. 4 shows the proposed current mode buck-boost converter with the APS and the ACC (red box). The power stage contains four power switches, M_A to M_D , in an H-bridge structure. The feedback resistors, R_1 and R_2 , are used to regulate V_{out} . In the feedback loop, a high-gain error amplifier, realized with a cascode high-gain transconductance operational amplifier (OTA) with a single-ended output, generates the error signal V_{ea} . This signal is compared to a summation signal V_{sum} to realize the duty cycles for buck, V_{BU} , and boost, V_{BO} , mode. V_{sum} is the sum of the current-sensing signal V_s and the slope compensation signal V_{slope} . For the slope, the dynamic slope compensation is used to avoid sub-harmonic oscillation. This buck-boost converter utilizes the dual-mode operation, where the peak current control (PCC) and the valley current control (VCC) are used for buck and boost mode, respectively. To ensure a smooth transition between the two modes, a master/slave control scheme is implemented. The master is the traditional mode detector, which compares V_{out} with V_{in} . If $V_{in} < V_{out}$, boost mode is required and if $V_{in} > V_{out}$, buck mode is needed. The slave circuit is used if V_{in} comes close to V_{out} . In that case the self-tuning pulse skipping (SPS) mechanism is applied. The clock generator provides the system clock V_{clk} with 1 MHz. The Zero-Current Detector (ZCD) is used to detect, whether the inductor current I_L is negative. Dead time and driver circuits are implemented in order to control switches M_A to M_D , with the driving signals SM_A to SM_D .

B. Adaptive Pulse Skipping -APS

According to Fig. 2, *path I* and *path II* are used in buck mode and *path III* and *path IV* are used in boost mode. Besides, the signals of the duty cycle D and its complement D' are shown with the inductor current I_L . Self-tuning pulse skipping (SPS) is utilized to extend the effective duty cycle. With the SPS, the duty cycle of the converter cannot be larger than 16.7 %. This value is large enough to ensure a smooth transition, when V_{in} comes close to V_{out} and not too large to unnecessarily enlarge the inductor current and output voltage ripple. But this limit slows down the charging or discharging performance of the inductor. If it would be possible to change the duty cycle to higher values, it could be also possible to charge or discharge the inductor for a longer time. Thereby, the required charge in the inductor can be reached faster.

To realize the adaptive duty cycle in case of load transition, the APS is used. To activate the APS a the feedback signal V_{fb} is compared to a higher and lower limit. If V_{fb} exceeds one of these limits the APS starts. Fig. 5 shows the difference with and without the APS. The main interest of the APS is to vary the duty cycle and its complement, since they control the charge and discharge pulse of the inductor by varying the skipping number of the pulses. Fig. 5 shows that the waveform with APS can reach the required inductor current level faster than the waveform without the APS. The reason is that with the APS, the controller is able to remain the system for a longer time in the charge or discharge path.

Further, ω_{RHP} is dependent of D' in boost mode (1),

$$\omega_{RHP} = \frac{(D')^2 R_L}{L} \quad \text{with } R_L = \frac{v_{out}}{i_{out}} \quad (1)$$

which means, with varying the complement of the duty cycle, the ω_{RHP} can be affected. Higher values of ω_{RHP} make the system slower and more stable and lower values make the system faster and improve the transient response. In buck mode, the ω_{RHP} does not occur.

First, there is the load jump from light-to-heavy in boost mode. The pre-defined value of the SPS mechanism is 90 %. With this duty cycle, the system can only remain in the charging path (*path III*) for 10 % of a period T_s . Hence, the controller is not able to vary the duty cycle to remain longer in *path III* and to reach a higher inductor current level faster. When there is the load change, the voltage at the output V_{out} drops down and a voltage controlled current source (VCCS) detects the variation. In that case the SPS signal V_{SPS} changes the effective duty cycle to a value of 60 %. Therefore, the SPS mechanism is able to skip more discharging phases and the inductor can be charged for a longer time. Consequently, more current is stored into the inductor through *path III*. The result is a lower undershoot voltage because ω_{RHP} is brought to lower frequencies and the system is faster. Further, it is possible to reach the required energy level in the in-

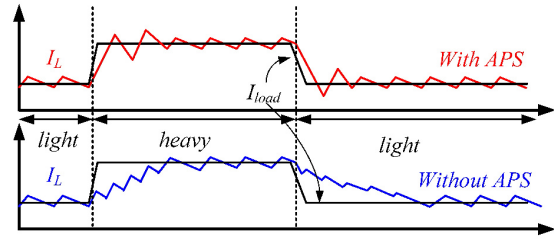


Fig. 5: Effect of the APS on the inductor current.

ductor faster. Second, if there is a load change from heavy-to-light load, the SPS signal also sets the duty cycle to a value of 60 %. In that case, the skipping number is also higher, so the charging phase of the inductor is skipped and the energy in the inductor is brought to the output. Hence, it is possible to use the energy in the inductor first, before recharging it again. But however, in total D' is shorter and ω_{RHP} is also at low frequencies. The result is a lower overshoot, because the system is fast again. Similarly, there is the load jump from light-to-heavy and heavy-to-light in buck mode.

C. Adaptive Compensation Capacitance - ACC

Current mode regulation is used to get a better line rejection and on-chip system compensation. Therefore, the system compensation is a proportional integral (PI) compensator. This PI compensator has a pole-zero pair, where the ω_{zcl} is used to cancel the load-dependent system pole ω_{p1} located at the output. However, ω_{zcl} affects the performance of the current mode buck-boost converter at different load conditions. The fixed compensation pole-zero pair of the PI compensator can only ensure the system stability but not a good transient response. Therefore, an adaptive compensation is proposed. This requires to analyze the relevant transfer functions of the converter. The control-to-output transfer function (2) of the current mode buck-boost converter has a real pole ω_{p1} , located at the output of the regulator.

$$G_{vc} = \frac{\hat{v}}{\hat{i}_c} = \frac{D'R}{2R_i} \frac{\left(1 - s \frac{L}{(D')^2 DR}\right) (1 + sR_{ESR}C_o)}{\left(1 + s \frac{RC_o}{2}\right)} \quad (2)$$

$$\omega_{p1} = \frac{2}{RC_o}, \quad \omega_{z(RHP)} = \frac{(D')^2 R}{L}, \quad (3)$$

$$\omega_{z(ESR)} = \frac{1}{R_{ESR}C_o}$$

The pole ω_{p1} , where C_o is the output filter capacitor and R is the load resistance (3), is proportional to the load current I_{load} . D' is the complement of the duty cycle. R_i is the current sensing gain and R_{ESR} is the equivalent series resistance of C_o . The control-to-output function contains two zeros. One is the right-half-plane zero $\omega_{z(RHP)}$, and the other one is $\omega_{z(ESR)}$.

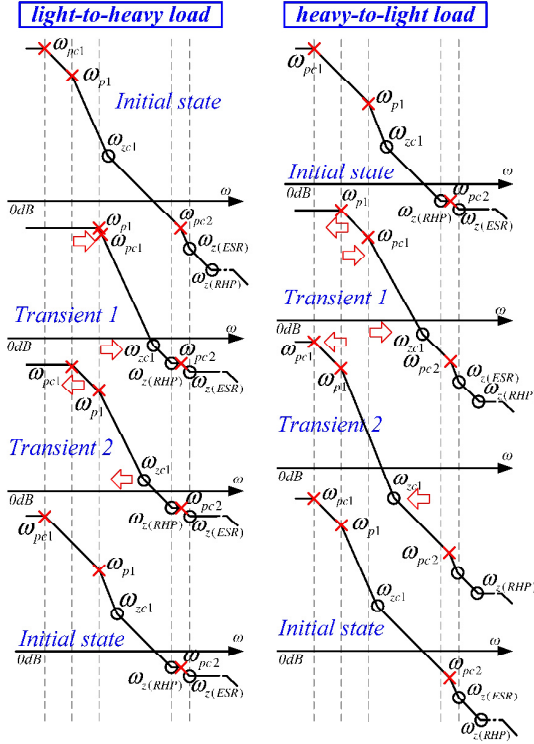


Fig. 6: Compensation poles and zeros with the ACC mechanism and a load change from light-to-heavy and heavy-to-light.

The zero $\omega_{z(RHP)}$ is the critical parameter in boost operation and it is dependent on the load. The zero $\omega_{z(ESR)}$ comes from the output capacitor C_o and it is at high frequencies, which makes it easier to handle than the $\omega_{z(RHP)}$. The transfer function of the PI compensator is

$$G_C(s) = -g_m R_o \frac{1 + sKC_C R_{z1}}{1 + sKC_C (R_{z1} + R_o)} \quad (4)$$

$$\omega_{pc1} = \frac{1}{KC_C R_o}, \quad \omega_{zc1} = \frac{1}{KC_C R_{z1}}, \quad KC_C = C_{C1} \quad (5)$$

R_{z1} and C_{C1} , are the on-chip compensation resistor and capacitor, respectively. R_o is the output resistance of the error amplifier and its value is much higher than that of R_{z1} , hence the simplification in (5). G_m is the transconductance of the error amplifier and K is the multiplication factor of the on-chip capacitor. The current generated by the error amplifier can be redirected to charge/discharge the on-chip capacitor C_{C1} . Further, the fast transient mechanism can be realized in case of a load current change [8]. In (4) and (5), the capacitance, which has to be adaptive, can be found. By varying the value of C_{C1} , a good performance can be implemented. Further, a large system bandwidth and an adequate PM are realized at any load condition.

The frequency response can be described in case of a load change with (2), (4) and Fig. 6. When the load current changes from light-to-heavy load, the output pole ω_{p1} moves from lower to higher frequencies. This happens because ω_{p1} is inversely dependent on the load resistance. The RHP zero, $\omega_{z(RHP)}$, is also depend-

ent on the load resistance. With a lower value of the load resistance, $\omega_{z(RHP)}$ is also at lower frequencies. With the first step of the fast transient mechanism the pole-zero pair ω_{pc1} and ω_{zc1} are moved far away from their initial state towards higher frequencies by adjusting the value of the adaptive capacitance C_{C1} . This value is brought, for example, from 400 pF to 4 pF. As a result the unity gain frequency ω_c is also at higher frequencies and the phase margin PM is deteriorated. Thus the system might get unstable, but it improves the transient response because of the small equivalent compensation capacitance and large system bandwidth. To get the system stable again, the second step of the fast transient mechanism gets initiated. In the second step, the pole-zero pair is pulled back to their initial state, by increasing the value of the adaptive capacitance again, so that the ω_c is decreased. Further, the PM is improved again to get the system stable. After the second step is over, the pole-zero pair is brought back to their initial position by setting the value of the adaptive capacitance back to the steady-state value. The pole ω_{pc2} and the zero $\omega_{z(ESR)}$, are not dependent on the load resistance or the adaptive capacitance. Therefore, their position remains always at the same point. The mechanism is similar for the change from heavy-to-light load.

III. CIRCUIT IMPLEMENTATION

For the Adaptive Pulse Skipping (APS) it is necessary to detect the voltage undershoot/overshoot if there is an output load change. Therefore, a voltage controlled current source (VCCS) circuit is used as shown in Fig. 7. This circuit can be used as a suitable differentiator to improve the sensing accuracy and the transient response time. The on-chip capacitor C_d can be kept small because of the current mirror ration k . The output voltage V_{VCCS} of the VCCS is the differentiated value of V_{fb} and it is used to detect the overshoot or undershoot. With this information, a signal for the SPS mechanism can be provided to realize the right SPS value V_{SPS} . At the beginning, V_{fb} is in steady state. If there is a load change, V_{fb} starts to fall or rise. The VCCS detects the change in the V_{fb} signal and the signal V_{VCCS} falls or rises, too. The slope of the V_{VCCS} depends on the change of the V_{fb} . The larger the change in V_{fb} , the larger the slope of the V_{VCCS} . The V_{VCCS} signal is compared to a lower V_{low} and a higher V_{high} limit. If it exceeds one of the limits, the V_{APS} signal is set to high. When V_{APS} is set high, the SPS value is set to a different value, to allow the controller to skip more charging or discharging phases. The value is set to an effective duty cycle of 60 %.

The adaptive compensation capacitance is used to multiply the small on-chip capacitor C_C , Fig. 8. Two modes are possible, first, to multiply the capacitor to an adequate value for system stability in steady state and second, to adjust it to a small value to realize a fast transient response [8].

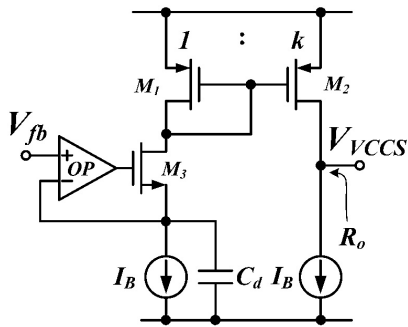


Fig. 7: Schematic of the VCCS.

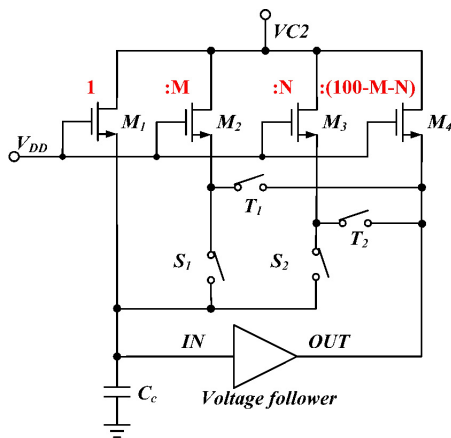


Fig. 8: Compensation Capacitance Multiplier.

In normal operation, the system is in *Steady State*. In that case, switches S_1 and S_2 are off (T_1 and T_2 are on), so that the current cannot flow through the small on-chip capacitor C_c . The current is flowing to the output of the voltage follower, which keeps its input voltage and output voltage at the same level. It is achieved that the small on-chip capacitor is multiplied to an adequate value to stabilize the system. In *Steady State*, the on-chip capacitor is equivalent to an off-chip capacitor, with a larger fixed value. If there is a load change, the fast transient technique gets started and S_1 and S_2 are switched on and T_1 and T_2 are switched off. This state is called the *Transient 1*. Now, it is possible to redirect the current to the output capacitor C_c of the error amplifier, what allows a fast discharge or charge of it. Therefore, the value of the equivalent capacitance is decreased to speed up the transient response. Though, a large current can cause oscillation. Therefore, the *Transient 1* has to have an adequate duration and the current, which is directed to C_c has to be decreased. Simulations showed that a time of $5 \mu\text{s}$ to $8 \mu\text{s}$ is enough to get a good transient response without causing oscillations.

After *Transient 1* is over, the mechanism goes into *Transient 2*. In that step, switches S_2 and T_1 are on and S_1 and T_2 are switched off. This results in a lower current, which is directed into the small on-chip capacitor. Thereby, C_c cannot be discharged or charged as fast as in *Transient 1*. This is done, in order to in-

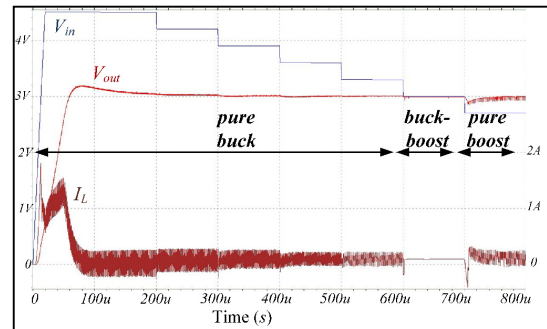
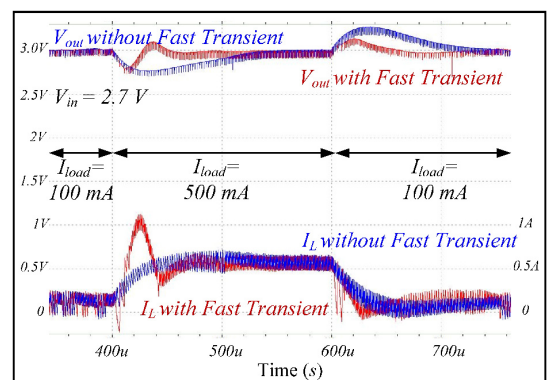
Fig. 9: Input voltage variation from 4.5 V to 2.7 V with an increment of 300 mV @ $I_{load} = 100 \text{ mA}$.

Fig. 10: Load transient (boost), 100 mA to 500 mA and vice versa.

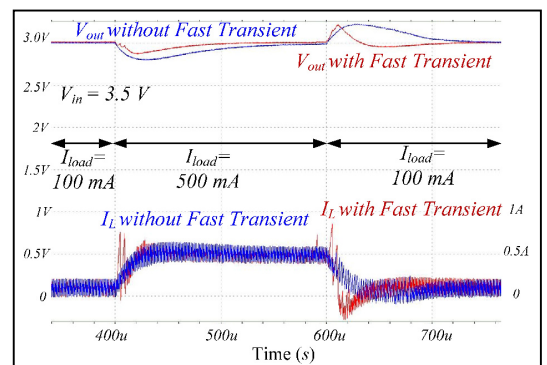


Fig. 11: Load transient (buck), 100 mA to 500 mA and vice versa.

crease the PM and to stabilize the system again, by pulling the compensation zero ω_{zc1} back to its origin state. After V_{out} is brought back to its regulated value, *Transient 2* is over and S_1 and S_2 are off and T_1 and T_2 are on again. Now, the system is in *Steady State* and the equivalent capacitor is at its amplified value.

IV. SIMULATION RESULTS

The proposed fast transient concepts were implemented in a 250 nm CMOS technology for an output voltage of 3 V and a load range of 10 mA to 900 mA at 2.7-4.5 V input. The switching frequency is 1 MHz, the inductor value $2.2 \mu\text{H}$ and the output capacitor $20 \mu\text{F}$ with $30 \text{ m}\Omega$ ESR.

Fig. 9 shows the simulation results at an input voltage variation from 4.5 V to 2.7 V with an increment of

Table 1: Comparison of the performance of the converter, with and without the fast transient techniques

		100mA – 500mA		500mA – 100mA	
		ΔV_{out} in mV	Respon- se time	ΔV_{out} in mV	Respon- se time
Boost	FT	200	54 μ s	130	59 μ s
	no FT	240	131 μ s	270	137 μ s
Buck	FT	120	94 μ s	190	41 μ s
	no FT	200	185 μ s	220	130 μ s
Buck- Boost	FT	180	56 μ s	150	69 μ s
	no FT	200	155 μ s	250	133 μ s

300 mV. Fig. 10 and Fig. 11 show the simulation results at a load transient from 100 mA to 500 mA and vice versa in boost and buck mode, respectively. Table 1 shows the comparison of the performances with and without the new fast techniques in the current mode buck-boost DC-DC converter. In boost mode, the improvements, in the voltage undershoot and the response time from light-to-heavy load, are 16.7 % and 58.8 %, respectively. In buck mode the improvements are 40 % and 49.2 % and in buck-boost mode, 10 % and 63.9 %, respectively. The improvements in the voltage overshoot and the response time from heavy-to-light load are, in boost mode, 51.9 % and 56.9 %, in buck mode, 13.6 % and 68.5 % and in buck-boost mode, 40 % and 48.1 %, respectively. The efficiency of the converter is at 60 % at light load (10 mA) and 91 to 95 % at heavy load (50-900 mA). At light load, the efficiency suffers from the switching losses of the power switches M_A to M_D . Obviously, the proposed fast transient mechanism shows better results.

V. CONCLUSIONS

A fast transient current mode buck-boost DC-DC converter with a good power management and a good transient performance is presented to realize a stable supply voltage for portable devices. The critical points in a buck-boost converter are the stability and a constant output voltage. Therefore, the adaptive pulse skipping APS and the adaptive compensation capacitance ACC are proposed as a new fast transient mechanism. Not only the transient response is improved, but also the overshoot/undershoot voltage of V_{out} . Further, the system stability can be ensured by the ACC, due to its adjustable value. Simulation results demonstrated the potential of the fast transient techniques. With the presented fast transient current mode buck-boost DC-DC converter, a solution for an efficient power management was found. This way, it is possible to lower the risk of deteriorations, e.g. unwanted resets, to the powered micro-controller (μ C).

REFERENCES

- [1] Y.-H Lee, S.-C. Huang, S.-W. Wang, W.-C. Wu and K.-H. Chen, "Power-tracking embedded buck-boost converter with fast dynamic voltage scaling for the SoC system," *IEEE Trans. Power Electronics*, pp. 1271–1282, March 2012.

- [2] C.-L. Wei, C.-H. Chen, K.-C. Wu; I.-T. Ko; , "Design of an average-current-mode noninverting buck-boost dc-dc converter with reduced switching and conduction losses," *IEEE Trans. Power Electronics*, pp.4934-4943, Dec. 2012.
- [3] B. Sahu, and G.A.R. Mora., "A low voltage, dynamic, noninverting, synchronous buck-boost converter for portable applications," *IEEE Transactions on Power Electronics*, vol.19, no.2, pp. 443- 452, March 2004.
- [4] B. Sahu, and G.A.R. Mora, "A high-efficiency dual-mode, dynamic, buck-boost power supply IC for portable applications," *VLSI design*, 2005, pp.858–861.
- [5] Y.-H Lee, S.-C. Huang, S.-W. Wang and K.-H. Chen, "Fast transient (FT) technique with adaptive phase margin (APM) for current mode dc-dc buck converter" *IEEE Trans. On Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 10, pp. 1781-1793, Oct. 2012.
- [6] C.-Y. Hsieh and K.-H. Chen, "Adaptive Pole-Zero Position (APZP) Technique of Regulated Power Supply for Improving SNR", *IEEE Trans. On Power Electronics*, vol. 23, no. 6, pp. 2949-2963, 2008.
- [7] J.-C. Tsai, C.-L. Chen, Y.-H. Lee, H.-Y. Yang, M.-S.Hsu, and K.-H. Chen, „modified hysteretic current control (MHCC) for improving transient response of boost converter”, *IEEE Trans. On Circuit And Systems*, vol.58, no.8, Aug. 2011.
- [8] K.-H. Chen, C.-J. Chang and T.-H. Liu, "Bidirectional current-mode capacitor multipliers for on-chip compensation", *IEEE Trans. On Power Electron.*, vol.23, no.1, pp. 180-188, Jan. 2008.



Andreas Ehrhart received the B.Eng. Degree in Electronics, Vehicle Electronics and Mechatronic Systems, in 2010 from the Baden-Württemberg Cooperative State University (DHBW) Ravensburg and the M.Sc. Degree in Electronics, Power- and Micro Electronics, in 2012 from the Reutlingen University.



Bernhard Wicht received the Dipl.-Ing. Degree from the Technical University Dresden in 1996 and the Ph.D. degree from the Technical University Munich in 2002. 2003 - 2010, he was with the Mixed Signal Automotive business unit of Texas Instruments, in Freising, Germany, responsible for the development of automotive smart power ICs. Since September 2010 he is professor for integrated circuits at Reutlingen University, Robert Bosch Center for Power Electronics.



Ke-Horng Chen received the B.Sc., M.Sc. and the Ph.D degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1994, 1996, and 2003, respectively. From 1996 to 1998, he was a part-time IC Designer at Philips, Taipei. From 1998 to 2000, he was an Application Engineer at Avanti, Ltd., Taiwan. From 2000 to 2003, he was Project Manager at ACARD, Ltd., where he was involved in designing power management ICs. He is currently an Associate Professor in the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan, where he organized a Mixed-Signal and Power Management IC Laboratory.



Entwurf eines rauscharmen Vorverstärkers für 60 GHz in SiGe:C 0,25 μm

Peter Kuhn, Stephan Lensing, Gerhard Forster, Roland Münzner

Zusammenfassung—Für drahtlose Funkssysteme steht ein unlizenzirtes Frequenzband von mehreren GHz Bandbreite bei 60 GHz zur Verfügung, dass z.B. vom IEEE-Standard 802.15.3c genutzt wird. Der Beitrag befasst sich mit dem Schaltungs- und Layoutentwurf eines rauscharmen Vorverstärkers (LNA) für diesen Frequenzbereich auf der Basis des 0,25- μm -SiGe-BiCMOS-Prozesses von IHP. In der Zielspezifikation wurde neben Kenngrößen wie Rauschzahl, Verstärkung und Verlustleistung besonderer Wert auf eine große Bandbreite gelegt. Bandbreite, Verstärkung, Rauschzahl und Verlustleistung wurden in einer gemeinsamen Kennzahl (Figure-of-Merit) zusammengefasst. Nach Voruntersuchungen an mehreren Schaltungstopologien wurde die erfolgversprechendste Verstärkerschaltung ausgewählt, einschließlich der erforderlichen Anpassnetzwerke dimensioniert, in ein Layout überführt und dieses mit einem 2,5D-Simulator verifiziert. Der Vorverstärker erreicht in der Simulation eine maximale Verstärkung von $S_{21} = 21,5$ dB im Frequenzbereich von 55 GHz bis 68 GHz (3-dB-Bandbreite) und ein Rauschmaß von $NF = 7,3$ dB. Die Chipfläche des Verstärkers beträgt 0,21 mm². Die auf Basis der Simulationen ermittelte Figure-of-Merit erreicht den höchsten Wert im Vergleich zu anderen aktuell bekannten Vorverstärkern.

Schlüsselwörter—60-GHz-Band, IEEE 802.15.3c, SiGe, BiCMOS, Low-Noise-Amplifier (LNA), 0,25 μm , Kaskodeschaltung, Smith-Diagramm.

I. EINLEITUNG

Drahtlose Funkssysteme sind in der heutigen Zeit aus nahezu keinem Alltagsbereich mehr wegzudenken. Mobiltelefone und Laptops nutzen z. B. den IEEE-802.11-Standard (Wireless Local Area Network – WLAN), der in Erweiterungen Datenraten von einigen

Peter Kuhn, kuhn peter1@googlemail.com, Gerhard Forster, forster@hs-ulm.de und Roland Münzner, muenzner@hs-ulm.de sind Mitglieder der Hochschule Ulm, Eberhard-Finckh-Str. 11, 89075 Ulm.

Stephan Lensing, stephan.lensing@iis.fraunhofer.de, Fraunhofer-Institut für Integrierte Schaltungen IIS, Am Wolfsmantel 33, 91058 Erlangen.

Tabelle 1: Lizenzfreie Frequenzbänder bei 60 GHz [3].

Land / Region	Frequenzbereich
Europa	57 bis 66 GHz
Nordamerika	57 bis 64 GHz
Japan	59 bis 66 GHz
Korea	57 bis 64 GHz

hundert Mbit/s ermöglicht. Moderne Anwendungen verlangen nach immer höheren Datenraten und sind daher mit WLAN nur eingeschränkt nutzbar. Dazu zählt zum Beispiel die unkomprimierte, hochauflösende Videoübertragung, welche Datenraten von einigen Gigabit pro Sekunde (Gbit/s) benötigt.

Der IEEE Standard 802.15.3c [1] ermöglicht es, unter Verwendung eines mehrere Gigahertz breiten Frequenzbandes bei 60 GHz, die Datenraten zu steigern. Länderspezifisch stehen bis zu neun Gigahertz Bandbreite zur Verfügung (siehe Tabelle 1). Die damit möglichen Datenraten von bis zu fünf Gbit/s lassen sich mit weniger komplexen Modulationsarten realisieren, was eine reduzierte Komplexität in der Signalverarbeitung bewirkt [1]. Ein mögliches Anwendungsfeld ist zum Beispiel der breitbandige Datentransfer für Büro- und Konferenzanwendungen. Für den Consumer-Bereich ist die unkomprimierte, hochauflösende Videoübertragung ein mögliches Einsatzgebiet.

Ziel der diesem Beitrags zu Grunde liegenden Arbeit war es, einen modular einsetzbaren, breitbandigen und rauscharmen Vorverstärker für ein bei 60 GHz arbeitendes Empfängersystem zu entwickeln. Der Vorverstärker soll in einer 0,25- μm -SiGe-BiCMOS-Technologie von IHP realisiert werden. Diese Technologie stellt NPN-HBT-Transistoren mit einer Transit- und maximalen Oszillatorfrequenz von 190 GHz zur Verfügung. Der Verstärker soll mit der typischen Versorgungsspannung von 3 V effizient mit maximal 40 mW arbeiten. Die maximale Verstärkung S_{21} wird auf größer 20 dB angesetzt.

Zunächst wird in Abschnitt 2 der Stand der Technik dargestellt und die für den entworfenen Verstärker erarbeitete Spezifikation diskutiert. In Abschnitt 3 wird auf mögliche Verstärkertopologien eingegangen. Der Verstärkerentwurf und die Dimensionierung der

Tabelle 2: Aktuelle Vorverstärker in 0,25- μm -SiGe-BiCMOS für den Frequenzbereich bei 60 GHz (Auswahl).

Quelle	Grujic et al. [5]	Scheytt et al. [6]	Sun et al. [7]
Topologie	2 x Diff.-Kaskodeschaltung	Diff. Kaskode + Emitterschaltung	3 x Differenzverstärker
Verstärkung S_{21}	24,5 dB	15,3 dB	18 dB
3-dB-Bandbreite B	5 GHz	7 GHz	22 GHz
Rauschmaß NF	5,95 dB	6,2 dB	6,8 dB
Verlustleistung P_V	59,4 mW	9,9 mW	66 mW
Figure-of-Merit FoM	8 GHz/mW	7 GHz/mW	5,7 GHz/mW

Anpassnetzwerke erfolgen in Abschnitt 4. Der Layoutentwurf wird in Abschnitt 5 besprochen. Simulationsergebnisse der Gesamtsimulation werden im sechsten Abschnitt erörtert. In Abschnitt 7 folgen eine Zusammenfassung sowie ein abschließender Ausblick.

II. STAND DER TECHNIK UND SPEZIFIKATION

LNA bei 60 GHz für den oben beschriebenen Einsatzbereich lassen sich mit Hilfe einer Kennzahl (Figure-of-Merit - FoM) zusammenfassend vergleichen [3]. Diese FoM wird für die Bewertung des Standes der Technik und für die Definition der Spezifikation herangezogen. Die in [3] beschriebene FoM setzt sich aus dem linearen Wert der Verstärkung S_{21} , der 3-dB-Bandbreite B in GHz, dem linearen Wert der Rauschzahl F und der Leistungsaufnahme P_V in mW des Verstärkers zusammen:

$$FoM = \frac{S_{21} \cdot B}{(F - 1) \cdot P_V}$$

Die Parameter sind bewusst für einen breitbandigen, rauscharmen Vorverstärker gewählt. Eine hohe Verstärkung und eine geringe Rauschzahl sind am Beginn einer Verstärkerkette besonders wichtig. Dadurch haben die Rauschzahlen F_2 bis F_n der nachfolgenden Stufen einen geringeren Einfluss auf die Gesamt-rauschzahl F_{ges} . Dies ist von Friis [4] mit der Gleichung

$$F_{ges} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \cdot G_2} + \dots + \frac{F_{n-1} - 1}{G_1 G_2 \dots G_{n-1}}$$

beschrieben worden, wobei G_i die Verstärkung der i -ten Stufe bezeichnet. Eine große Bandbreite sorgt neben einer breiten Abdeckung für einen flachen Verstärkungsverlauf über der Frequenz. Weiter wichtig ist eine geringe Verlustleistung des Verstärkers. Im Folgenden wird näher auf den Stand der Technik und die Spezifikation eingegangen.

Tabelle 3: Spezifikation für den Vorverstärker ($V_{Supply} = 3 \text{ V}$).

Verstärkung S_{21}	$\geq 20 \text{ dB}$
3-dB-Bandbreite B	$\geq 11 \text{ GHz}$ (min. 56 bis 67 GHz)
Rauschmaß NF	$\leq 7,5 \text{ dB}$
Verlustleistung P_V	$\leq 40 \text{ mW}$
Figure-of-Merit FoM	$\geq 6 \text{ GHz/mW}$

A. Stand der Technik

Eine Auswahl von in der Literatur beschriebenen Vorverstärkern ist in Tabelle 2 aufgelistet. Der Vorverstärker nach Grujic [5] besteht aus einer zweifachen differenziellen Kaskodeschaltung. Diese erreicht durch die Kaskodeschaltung eine hohe Verstärkung von 24,5 dB. Ihre Verlustleistung ist aber durch den differenziellen Aufbau sehr hoch. In [6] wird dagegen die Verlustleistung durch die Wiederverwendung des Arbeitsstromes niedrig gehalten. Die Verstärkung beträgt dabei nur ca. 15 dB. Die 3-dB-Bandbreiten der Verstärker aus [5] und [6] sind deutlich kleiner als das europäische Frequenzband (Tabelle 1). Dieses wird mit dem dreistufigen Differenzverstärker nach [7] abgedeckt. Die Verlustleistung ist, bedingt durch den dreistufigen, differentiellen Aufbau mit 66 mW deutlich oberhalb der angesetzten 40 mW. Das Rauschmaß der Auswahl erreicht für den Frequenzbereich bei 60 GHz und die verwendeten Technologien typische Werte von kleiner 7 dB. In der folgenden Spezifikation werden die für die FoM wichtigen Parameter des Vorverstärkers festgelegt.

B. Spezifikation

Die Spezifikation für den Vorverstärker ist in Tabelle 3 zusammengefasst. Darin sind die geforderten Wertebereiche für die Parameter S_{21} , B , NF , P_V und die daraus resultierende FoM aufgelistet. Die angestrebte Verstärkung soll größer 20 dB sein, damit

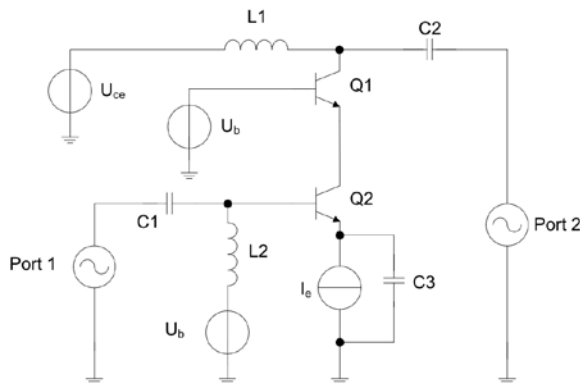


Abbildung 1: Kaskodeschaltung in Testbench in Cadence.

der Einfluss des Rauschens der nachfolgenden Stufen minimiert wird. Zur Abdeckung des kompletten Frequenzbereiches sind nach Tabelle 1 mindestens 9 GHz notwendig. Zusätzlich wird jeweils 1 GHz zusätzliche Bandbreite zur Kompensation möglicher auftretender Prozessschwankungen am unteren und oberen Ende des Frequenzbereiches einkalkuliert. Das Rauschmaß des Vorverstärkers soll ein Maximum von 7,5 dB nicht überschreiten. Bei einer Verlustleistung von 40 mW mit der Versorgungsspannung $V_{\text{Supply}} = 3 \text{ V}$ ist die Stromaufnahme auf 13,3 mA beschränkt. Keiner der Vorverstärker aus Tabelle 2 erreicht all diese Parameter gleichzeitig. Ziel dieser Arbeit war es daher, einen Vorverstärker zu entwerfen, der alle Punkte der Spezifikation erfüllt.

III. VERSTÄRKERENTWURF

Wie im vorigen Abschnitt gezeigt, kommen verschiedene Verstärkertopologien für die Realisierung in Frage. Zuerst wurden zur Ermittlung der Leistungsfähigkeit einstufige Verstärkerschaltungen untersucht. Dazu zählen Emitter-, Basis- und Kaskodeschaltung, sowie ein Differenzverstärker mit und ohne Kaskodetransistor. Mit einer Simulationsumgebung wurden das minimale Rauschmaß NF_{min} und die verfügbare Verstärkung G_A in Abhängigkeit vom Kollektorstrom I_C ermittelt. In Abbildung 1 ist eine solche Simulationsumgebung am Beispiel der Kaskodeschaltung dargestellt.

In Tabelle 4 sind die Ergebnisse für NF_{min} und G_A aufgelistet. Die verwendeten Transistoren haben einen maximalen Kollektorstrom $I_{C,\text{max}} = 8 \text{ mA}$. Die Werte sind bei einem Kollektorstrom $I_C = 4 \text{ mA}$ bei $f = 60 \text{ GHz}$ ermittelt. Mit dem Kollektorstrom $I_C = 4 \text{ mA}$ wird ein Kompromiss zwischen Rauschmaß und Verstärkung getroffen, wobei ein kleines Rauschmaß bei angemessener Verstärkung angestrebt wurde.

Keiner der einstufigen Verstärker in Tabelle 2 erreicht die geforderte Verstärkung von 20 dB. Diese Gesamtverstärkung ist nur durch mehrstufige Topologien erreichbar. Ein zweistufiger Verstärker kann

Tabelle 4: NF_{min} und G_A für die einzelnen Verstärkerschaltungen bei einem Kollektorstrom von 4 mA.

	NF_{min}	G_A
Emitterschaltung	4,7 dB	6,7 dB
Basisschaltung	5,0 dB	7,8 dB
Kaskodeschaltung	5,5 dB	18,5 dB
Differenzverstärker	4,7 dB	6,8 dB
Differenzverstärker mit Kaskodetransistor	5,5 dB	17,4 dB

aus einer Emitter- oder Basisschaltung mit nachfolgender Kaskodeschaltung bestehen. Durch Verwendung der Basisschaltung verringert sich durch Unterdrückung des Miller-Effekts die Rückwirkung vom Ausgang auf den Eingang. Das reduziert die Gefahr, dass der Verstärker instabil wird. Außerdem verringert sich mit der Basisschaltung der Eingangswiderstand der Gesamtschaltung, was oft eine einfachere Anpassung an die Systemimpedanz von 50Ω zur Folge hat. Eine zweistufige Kaskodeschaltung stellt eine weitere mögliche Verstärkertopologie dar. Die Kaskodeschaltung hat ein höheres Rauschmaß als die anderen Verstärker, was sich auf die größere Anzahl Transistoren zurückführen lässt. Vorteile der Kaskodeschaltung liegen dagegen in der geringen Rückwirkung und einer hohen Verstärkung. Kombiniert man reine Basis- und Emitterschaltungen, benötigt man mindestens drei Stufen, um die gewünschte Verstärkung zu erreichen. Dies erhöht, ebenso wie die Verwendung von mehreren Differenzverstärkern, den Strombedarf erheblich. Für Differenzverstärker wird der doppelte Strom für das Erreichen vergleichbarer Werte für Rauschmaß und Verstärkung im Verhältnis zu massebezogenen Verstärkern benötigt.

Wegen ihrer hohen Verstärkung und großen Rückwirkungsfreiheit wird eine zweifache Kaskodeschaltung als erfolgversprechendste Verstärkertopologie ausgewählt. Das höhere Rauschmaß wird zu Gunsten der großen Verstärkung in Kauf genommen.

Im folgenden Abschnitt wird auf die zweistufige Kaskodeschaltung und ihre Umsetzung in ein Layout, sowie die erzielten Simulationsergebnisse eingegangen.

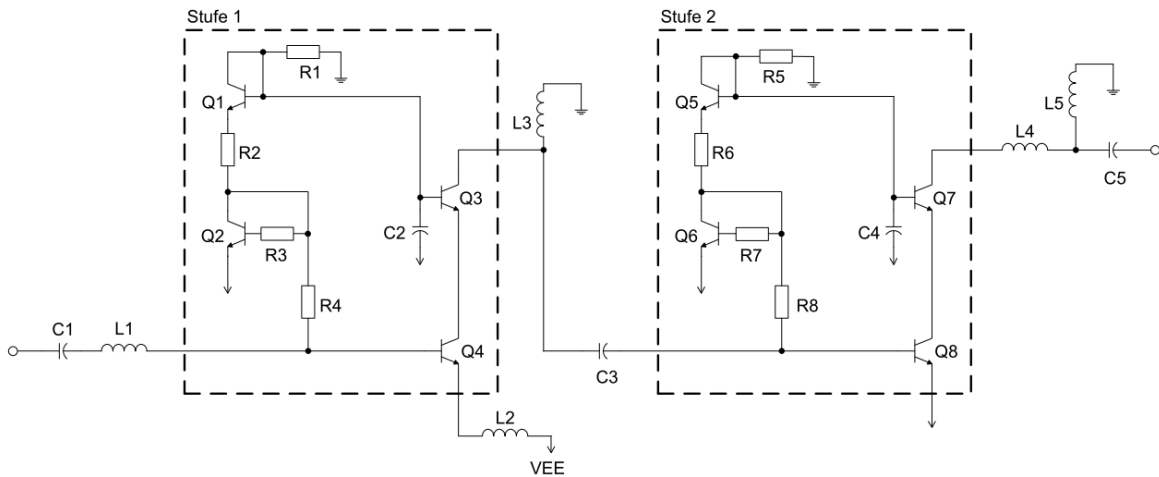


Abbildung 2: Vorverstärker, bestehend aus zwei Kaskodeschaltungen.

Tabelle 5: Eingesetzte Bauteile der aktiven Verstärkerstufen.

Bauteile	Wert	Bauteile	Wert
R1, R5	1 kΩ	Q1, Q2, Q5, Q6	$I_{C,max} = 2 \text{ mA}$
R2, R6	0,75 kΩ	Q3, Q4	$I_{C,max} = 10 \text{ mA}$
R3	2,5 kΩ	Q7, Q8	$I_{C,max} = 16 \text{ mA}$
R4, R8	0,5 kΩ	C2, C4	0,5 pF
R7	4 kΩ		

IV. ZWEIFACHE KASKODESCHALTUNG

In Abbildung 2 ist die entwickelte Verstärkerschaltung, bestehend aus zwei Kaskodeschaltungen, dargestellt. Die aktiven Verstärkerstufen sind jeweils eingrahmt. Die umliegenden Elemente lassen sich als Anpassnetzwerke identifizieren. Die Auslegung der Verstärkerstufen und die Dimensionierung der einzelnen Anpassnetzwerke werden im Folgenden erläutert.

A. Arbeitspunkteinstellung

Für den zweistufigen Verstärker wird ein fester Arbeitspunkt gewählt. Die Transistoren Q3 und Q4 sind nach dem Vorgehen aus [8] dimensioniert. Zuerst wird der Arbeitsstrom der Kaskodeschaltung bestimmt, welcher ein minimales Rauschmaß ermöglicht. Anschließend wurden die Transistoren so gewählt, dass bei gleichbleibender Stromdichte der Realteil der optimalen Rauschimpedanz möglichst 50 Ω entspricht. Das erreicht man beim Einsatz von Transistoren, deren maximaler Kollektorstrom 10 mA beträgt. Die Transistoren der zweiten Stufe sind so gewählt, dass sie ein $I_{C,max} = 16 \text{ mA}$ haben. Diese sind damit absolut stabil. In Tabelle 5 sind die Nennwerte der eingesetzten Bauteile zusammengestellt.



Abbildung 3: Layout der ersten Verstärkerstufe (Ausschnitt).

B. Layout der aktiven Stufen

Die Layouts der Verstärkerstufen werden mit einem 0,25 μm Prozess von IHP realisiert. Der Prozess SG25H1 stellt insgesamt fünf Metalllagen zur Realisierung der Verdrahtung und der Anpassnetzwerke bereit.

Abbildung 3 zeigt das Layout der ersten Verstärkerstufe. Für das Layout der Verstärkerstufe ist darauf geachtet worden, dass Widerstände, Kondensatoren und Transistoren möglichst platzsparend angeordnet sind. Weiterhin sind alle Transistoren in der gleichen Orientierung angeordnet. Nach fehlerfreiem LVS und DRC wurden aus dem Layout die parasitären Kapazitäten und Widerstände extrahiert. Diese parasitären Elemente wurden in die weiteren Simulationen einbezogen. Auf die Extraktion der Induktivitäten wurde wegen ihres zu erwartenden geringen Einflusses aufgrund des kompakten Layouts verzichtet. Das Erstellen des Layouts und die Extraktion der parasitären Elemente der zweiten Verstärkerstufe erfolgten nach

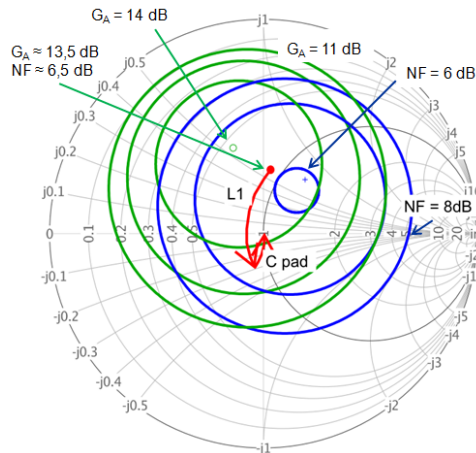


Abbildung 4: Kreise konstanter Verstärkung und konstanten Rauschmaßes der ersten Verstärkerstufe inklusive des Transformationsweges der Komponenten $L1$ und C_{pad} .

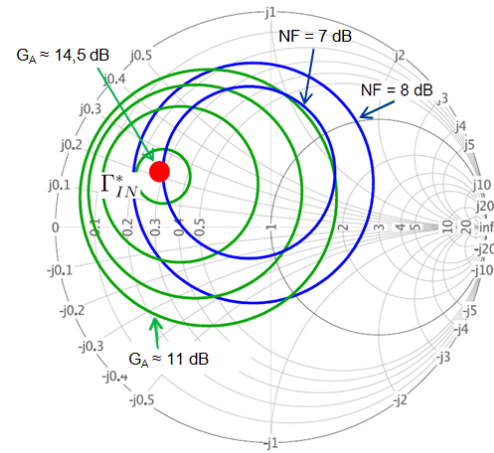


Abbildung 6: Kreise konstanter Verstärkung und konstanten Rauschmaßes der zweiten Verstärkerstufe.

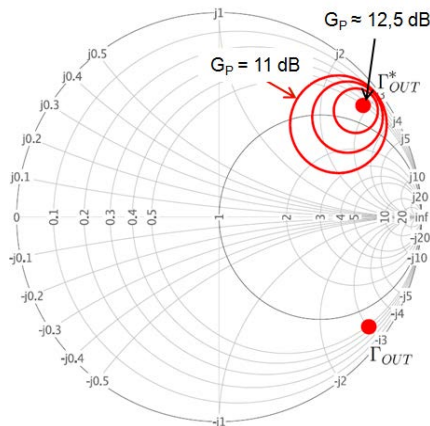


Abbildung 5: Kreise konstanter Leistungsverstärkung der ersten Verstärkerstufe.

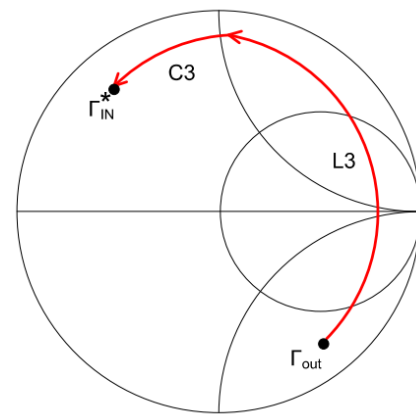


Abbildung 7: Prinzipieller Verlauf der Impedanztransformation durch die Anpasselemente $L3$ und $C3$ zwischen den Verstärkerstufen.

identischem Vorgehen. Der Einfluss der parasitären Elemente wurde anschließend beim Entwurf der Anpassnetzwerke mitberücksichtigt.

C. Entwurf der Anpassnetzwerke

Die Anpassnetzwerke des Verstärkers lassen sich in drei Teile zerlegen. Diese befinden sich am Ein- und Ausgang sowie zwischen den Verstärkerstufen. Eingang und Ausgang sorgen dabei für die Anpassung an die Systemimpedanz von 50Ω . Für die Anpassung und Dimensionierung der Anpasselemente wird das Smith-Diagramm eingesetzt. Mit den Kreisen für konstante verfügbare Verstärkung und Leistungsverstärkung lassen sich die Reflexionsfaktoren für eine bestimmte Verstärkung an Ein- bzw. Ausgang der einzelnen Verstärkerstufen ermitteln. Neben den Verstärkungskreisen des Eingangs werden die Kreise konstantem Rauschmaßes in der Simulationsumgebung berechnet und in das auf den Eingang bezogene Smith-Diagramm eingezeichnet. Die hohe Rückwirkungsfreiheit der Kaskodeschaltung ermöglicht die

schrittweise Entwicklung der Anpassnetzwerke vom Eingang zum Ausgang hin.

1) Anpassnetzwerk am Verstärkereingang

Das Netzwerk am Verstärkereingang ist so dimensioniert, dass eine hohe Verstärkung bei gleichzeitig geringem Rauschmaß erreicht wird. In Abbildung 4 sind Kreise konstanter Verstärkung und konstanten Rauschmaßes für den Eingang der ersten Verstärkerstufe bei einer Frequenz von 60 GHz dargestellt. Als Kompromiss zwischen Verstärkung und Rauschmaß wurde der Punkt mit $G_A \approx 13,5 \text{ dB}$ und $NF \approx 6,5 \text{ dB}$ gewählt. Neben der Serieninduktivität $L1$ ist die Kapazität des Anschlusskontaktes C_{pad} bei der Anpassung an 50Ω mit zu berücksichtigen. C_{pad} entsteht durch die Fläche des $80 \mu\text{m} \times 80 \mu\text{m}$ großen Bondpads über dem Substrat und berechnet sich zu rund 16 fF . $C1$ aus Abbildung 2 ist mit 400 fF so dimensioniert, dass nur ein geringer Einfluss auf das Anpassnetzwerk besteht.

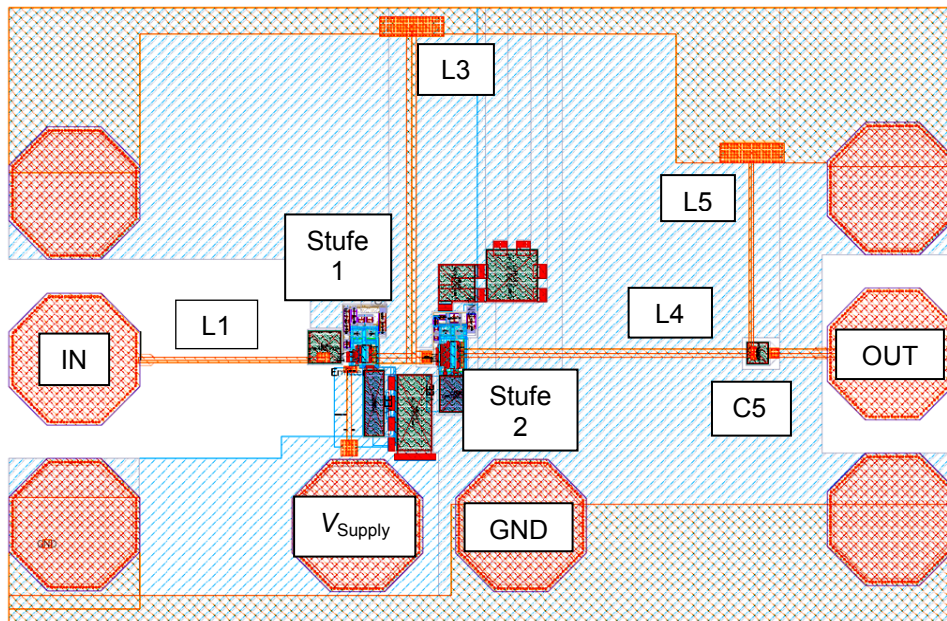


Abbildung 10: Layout des Vorverstärkers mit Anschluss pads.

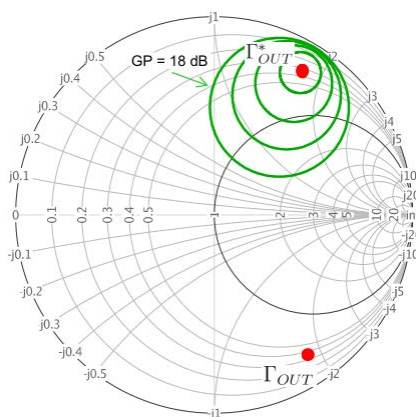


Abbildung 8: Kreise konstanter Leistungsverstärkung des zwei-stufigen Verstärkers ohne Ausgangsanpassung.

2) Netzwerk zwischen den Verstärkerstufen

Das Netzwerk zwischen den Verstärkerstufen hat die Aufgabe, den Ausgang der ersten auf den Eingang zweiten Stufe anzupassen. Bei der hier eingesetzten Anpassung spricht man von einer konjugiert komplexen Anpassung. Diese hat den Vorteil, dass man nicht auf eine Zwischenimpedanz anpassen muss und zusätzliche Anpasselemente gespart werden.

Abbildung 5 zeigt die Leistungsverstärkungskreise am Ausgang der ersten Verstärkerstufe von 11 bis 12,5 dB bei 60 GHz. Für eine hohe Verstärkung der ersten Stufe wird auf Γ_{OUT} angepasst.

Die Kreise für konstantes G_A und NF sind in Abbildung 6 dargestellt. Bei einer hohen Verstärkung der ersten Verstärkerstufe kann, entsprechend der Friis'schen Gleichung, der Eingang der zweiten Stufe auf eine hohe Verstärkung angepasst werden. Das

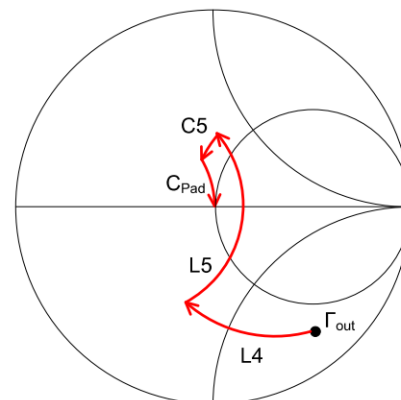


Abbildung 9: Prinzipieller Transformationsweg durch die Anpasselemente am Ausgang des Verstärkers.

Rauschmaß liegt bei Anpassung auf Γ_{IN}^* bei ca. 7 dB. Die Verstärkung G_A der zweiten Stufe erreicht in dieser Konfiguration bei optimaler Ausgangsanpassung ca. 14,5 dB.

Die Anpasselemente $L3$ und $C3$ übernehmen zusätzlich die Aufgabe der Stromversorgung der ersten Verstärkerstufe und der Gleichstromentkopplung zwischen den beiden Stufen. Für die Anpassung von Γ_{OUT} auf Γ_{IN}^* erhält man mit diesen Elementen einen Verlauf wie in Abbildung 7 prinzipiell dargestellt. Die ermittelten Werte für Induktivität bzw. Kapazität sind 90 pH und 60 fF.

3) Anpassnetzwerk am Verstärkerausgang

Der Ausgang des Verstärkers wird wie der Eingang auf die Systemimpedanz von 50 Ω angepasst. In Abbildung 8 sind die Kreise konstanter Leistungsverstärkung der gesamten Verstärkerkette ohne das aus-

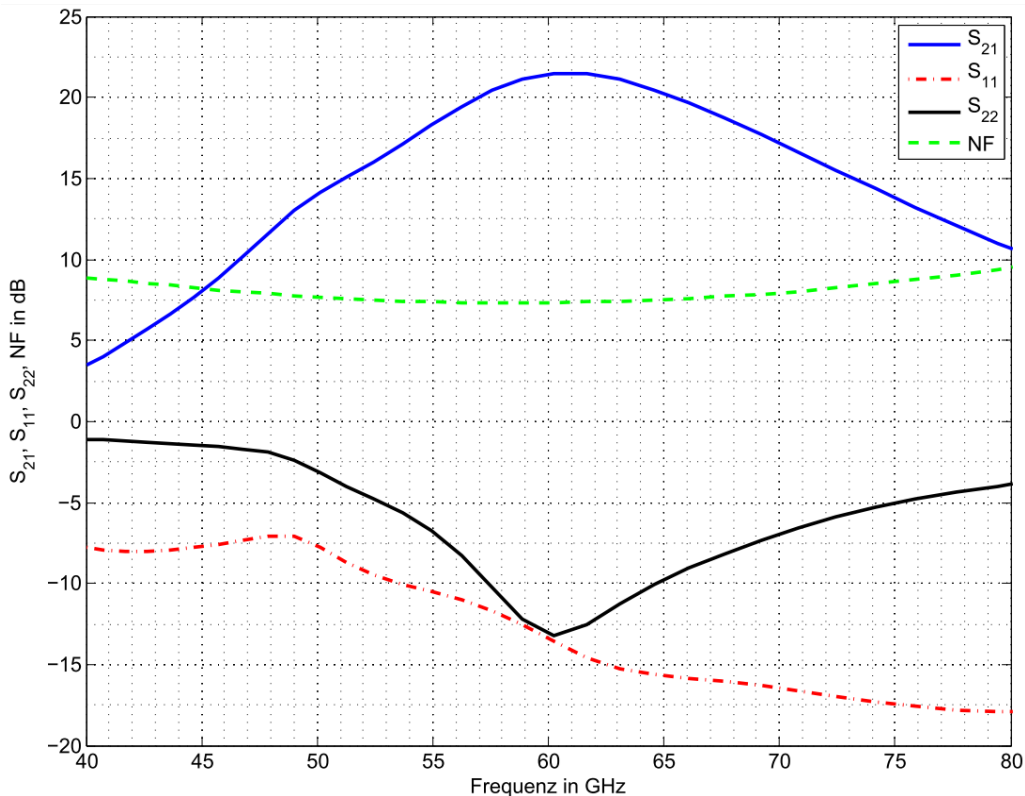


Abbildung 11: Simulationsergebnisse zu S-Parameter und Rauschmaß.

gangsseitige Anpassnetzwerk von 18 bis 21,5 dB bei Γ_{OUT}^* dargestellt. Wie zwischen den Verstärkerstufen wird mit einer Parallelinduktivität $L5$ und einer Serienkapazität $C5$ angepasst. Da die Serienkapazität nur begrenzt klein wählbar ist, wird eine serielle Leitung, hier $L4$, zusätzlich zwischen Ausgang der Verstärkerschaltung und $L5$ eingebracht. Im Anpassnetzwerk wird außerdem, wie bereits am Verstärkereingang, die Kapazität des Anschlusskontaktes in Form von C_{pad} mitberücksichtigt. Abbildung 9 zeigt den prinzipiellen Transformationsweg der eingesetzten Elemente bei 60 GHz.

V. GESAMTLAYOUT

Das Gesamtlayout des Vorverstärkers (Abbildung 10) setzt sich aus den zwei Verstärkerstufen und den drei im vorangegangenen Kapitel entworfenen Anpassnetzwerken zusammen. Letztere werden auf der obersten und stärksten Lage ausgeführt. Dadurch weisen die Bauteile einen geringeren Leitungswiderstand auf und induktive Komponenten erreichen eine höhere Güte. Die Leitung $L1$ am Eingang IN hat eine Länge von 115 μm bei 3 μm Breite. Für eine höhere Güte der Induktivität $L3$ wird eine breitere und längere Leitung gewählt. Sie ist bei 200 μm gegen Masse kurzgeschlossen und 6 μm breit. Die Stichleitung $L5$ ist 180 μm vom Ausgang der zweiten Stufe entfernt. Mit einer Länge von 115 μm und einer Breite von

3 μm wird eine kurze Leitung erreicht. $L4$ hat durch eine größere Breite mit 5 μm einen kleineren Serienwiderstand.

Die benötigte Chipfläche des Vorverstärkers mit Anschluss pads beträgt 370 μm x 580 μm . Die Abmessung der Anschluss pads von Ein- und Ausgang, sowie der Versorgungsanschlüsse betragen 80 μm x 80 μm . Das Gesamtlayout wurde mit Virtuoso XL von Cadence erstellt und hinsichtlich LVS und DRC geprüft.

Alle passiven Layoutbestandteile ohne die aktiven Stufen und Kapazitäten wurden mit ADS Momentum simuliert. Damit stehen deren S-Parameter von 0 bis 250 GHz zur Verfügung. Diese wurden für die Gesamtsimulation des Verstärkers mit dem Cadence Simulator Spectre genutzt.

VI. GESAMTSIMULATION

In den Simulationen für Rauschmaß, S-Parameter und Stabilität sind die parasitären Elemente der Verstärkerstufen und das Frequenzverhalten des passiven Gesamtlayouts mitberücksichtigt. Abbildung 11 zeigt die Verstärkung S_{21} , die Anpassung von Ein- und Ausgang S_{11} und S_{22} , sowie das Rauschmaß NF für den Frequenzbereich von 40 bis 80 GHz. Die Verstärkung erreicht ein Maximum von 21,5 dB bei 61 GHz und hat eine 3-dB-Bandbreite von 13 GHz. Die An-



Tabelle 6: Kenndaten des Vorverstärkers.

Verstärkung S_{21}	21,5 dB
3-dB Bandbreite B	13 GHz
Rauschmaß NF	7,3 dB
$S_{11} < -10$ dB	Ab 54 GHz
$S_{22} < -10$ dB	57,5 bis 64,5 GHz
Verlustleistung P_V	36 mW
Figure-of-Merit FoM	11 GHz/mW
Chipfläche mit Pads	370 μm x 580 μm

passung des Eingangs S_{11} liegt ab 54 GHz unter -10 dB. Die Reflexion am Ausgang ist im Frequenzbereich von 57,5 GHz bis 64,5 GHz kleiner -10 dB. Das Rauschmaß des entworfenen Verstärkers hat sein Minimum bei 58 GHz mit 7,3 dB. Die Verlustleistung des Verstärkers liegt bei 36 mW, was einer Stromaufnahme von 12 mA entspricht. Der Vorverstärker ist für den Bereich von 0 GHz bis 120 GHz absolut stabil. Alle Simulationen wurden mit dem Simulator Spectre von Cadence durchgeführt.

Der Vergleich der erzielten Werte mit den in Tabelle 3 spezifizierten zeigt, dass die Spezifikation vollständig erfüllt wird. Die Figure-of-Merit für den entworfenen und simulierten Vorverstärker liegt bei $FoM \approx 11$ GHz/mW und übertrifft damit alle in Tabelle 2 aufgelisteten Vorverstärker.

VII. ZUSAMMENFASSUNG

Für Anwendungen im Frequenzbereich bei 60 GHz wurde ein Vorverstärker entworfen. Dieser lässt sich als IP-Block in Transceivern einsetzen. Die Schaltung und die ermittelten S-Parametern des Layouts wurden anhand von Simulationen verifiziert. Der Vorverstärker kann im nächsten Schritt als Chip gefertigt werden. Die erzielten Kenndaten des Vorverstärkers sind in Tabelle 6 zusammengefasst. In einer Überarbeitung des Verstärkers könnte bei geeigneter Anordnung der Anpassnetzwerke die Chipfläche noch verringert werden.

In einem weiteren Schritt soll der Vorverstärker gefertigt und auf Waferebene getestet werden. Sollten sich die Simulationsdaten im Wesentlichen bestätigen, so stünde ein attraktiver Baustein für stationäre oder mobile Empfänger im Frequenzbereich um 60 GHz zur Verfügung.

DANKSAGUNG

Die Autoren bedanken sich beim Fraunhofer-Institut für Integrierte Schaltungen IIS, Erlangen, für die Ermöglichung der Masterarbeit, auf der diese Veröffent-

lichung beruht, sowie bei den Mitarbeitern der Abteilung für integrierte Schaltungen und Systeme (ICS) für deren Unterstützung.

LITERATURVERZEICHNIS

- [1] IEEE 802.15.3c-2009, Part 15.3: "Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for High Rate Wireless Personal Area Networks (WPANs)," *IEEE*, 2009.
- [2] A. V. Garcia, P. Xia und K. Yong, "60 GHz Technology for Gbps WLAN and WPAN: From Theory to Practice," 1. Aufl. *John Wiley & Sons*, 2011.
- [3] F. Ellinger u. a., "A low-power DC-7.8 GHz BiCMOS LNA for UWB and optical communications," *IEEE MTT-S International Microwave Symposium (IMS)*, 2004, pp. 13–16.
- [4] H. T. Friis, "Noise Figures of Radio Receivers," *Proceedings of the IRE*, Juli 1944, pp. 419–422.
- [5] D. Grujic et al., "Wide-bandwidth 60 GHz Differential LNA in SiGe:C Technology," *5th European Conference on Circuits and Systems for Communications*, 2010, pp. 71–74.
- [6] C. J. Scheytt et al., "A Low-Power 60 GHz Receiver Front-End with a Variable-Gain LNA in SiGe BiCMOS Technology," *Bipolar/BiCMOS Circuits and Technology Meeting*, 2010, pp. 192–195.
- [7] Y. Sun et al., "A Fully Integrated 60 GHz LNA in SiGe:C BiCMOS Technology," *Proceedings of the Bipolar/BiCMOS Circuits and Technology Meeting*, 2005, pp. 14–17.
- [8] M. Gordon und S. P. Voinigescu, "An Inductor-Based 52-GHz 0.18 μm SiGe HBT Cascode LNA with 22 dB Gain," *Proceeding of the 30th European Solid-State Circuits Conference*, 2004, pp. 287–290.



Peter Kuhn erhielt den akademischen Grad des Dipl.-Ing. (FH) in Nachrichtentechnik im Jahr 2009 von der Hochschule Ulm. Nach seiner zweijährigen Teilzeitbeschäftigung an der Hochschule Ulm und einem zehnmonatigen Auslandsaufenthalt am Rose-Hulman Institute of Technology in Terre Haute, Indiana, USA schloss er sein Masterstudium in Systems Engineering und Management im Oktober 2012 an der Hochschule Ulm ab.



Stephan Lensing studierte Elektrotechnik-Elektronik-Informationstechnik an der Universität Erlangen-Nürnberg und schloss im Jahr 2006 mit dem Grad Dipl.-Ing. (univ.) ab. Seit 2006 arbeitet er als wissenschaftlicher Mitarbeiter am Fraunhofer Institut für integrierte Schaltung IIS im Bereich RF- und Mikrowellen IC-Design.



Gerhard Forster studierte Physik mit dem Schwerpunkt Quantenelektronik an der Universität Heidelberg. Nach seinem Diplomabschluss 1977 befasste er sich als wissenschaftlicher Mitarbeiter am damaligen Forschungsinstitut von AEG-Telefunken (später Daimler Forschungszentrum) mit der Entwicklung und Anwendung neuer Halbleiterprozesse. Zuletzt war er als Teamleiter zuständig für die Entwicklung und den Test anwendungsspezifischer Integrierter Schaltungen aus den Gebieten der Nachrichtentechnik und der Automobilelektronik. Seit 1992 ist er Professor für Elektronik und Mikroelektronische Schaltungen an der Hochschule Ulm. Seine Schwerpunkte liegen auf dem Gebiet des Entwurfs von Mixed-Signal-ASICs. Zwischen 2001 und 2010 hatte er die Leitung der Fakultät Elektrotechnik und Informationstechnik inne. Prof. Forster ist Herausgeber des vorliegenden Tagungsbandes.



Roland Münzner studierte Physik und Philosophie an der Universität Tübingen. Nach Abschluss seiner Promotion wechselte er 2001 an das Forschungszentrum der Alcatel SEL AG in Stuttgart. Bis 02/2007 war er dort in verschiedenen Positionen tätig, zuletzt leitete er ein internationales Kompetenzzentrum für 4G Mobilfunksysteme. Seit 03/2007 ist er Professor für Nachrichtentechnik und Hochfrequenzelektronik an der Hochschule Ulm. Schwerpunkte seiner Arbeit sind Funk-systeme und die elektromagnetische Verträglichkeit. Prof. Münzner ist Leiter des Instituts für Kommunikationstechnik der Hochschule Ulm.

Direktumsetzender großsignalfester IQ-Demodulator für SDR-Anwendungen im UKW-Band mit High-Speed MOSFET Demultiplexer IC

Prisca Kadje Nwabo, Jonathan Mayer

Zusammenfassung—Es wurde ein Tuner für SDR-Anwendungen entwickelt, der mit verschiedener Software flexibel für FM- oder DRM+-Empfang eingesetzt werden kann. Für die direktumsetzende IQ-Demodulation kommt ein großsignalfester High-Speed MOSFET-Demultiplexer zum Einsatz, dessen Schalter durch einen Lokaloszillator angesteuert wird. Vor dem Demultiplexer wird das HF-Signal mittels LNA und AGC verstärkt und geregelt. Eine breitbandige Soundkarte und PC-Software sorgen für die Rückgewinnung des Audiosignals.

Schlüsselwörter—Software Defined Radio, IQ-Demodulation, High-Speed-Schalter.

I. EINLEITUNG

Bei Software Defined Radio (SDR) ist die eingesetzte Hardware unabhängig von der verwendeten Modulationsart. Das wird durch eine möglichst frühe Digitalisierung erreicht. Je nach Software können verschiedene Modulationsarten empfangen und demoduliert werden, wodurch eine hohe Flexibilität erreicht wird [1]. Vorgestellt wird ein SDR-fähiger Tuner entsprechend Abbildungen 1 und 2, der für Voruntersuchungen zur Digitalisierung des Rundfunks im UKW-Band geeignet ist. Der Tuner arbeitet nach dem Prinzip der Direktumsetzung, d. h. ohne ZF-Selektion.

In Deutschland wird neben analogen FM-Sendern seit August 2011 mit DAB+ (Digital Audio Broadcasting+) auch digitales Radio ausgestrahlt. Dazu wird das Frequenzband 174 ... MHz genutzt [2]. Bei DAB+ werden sogenannte Ensembles aus verschiedenen Programmen mit Audio- und Dateninhalten übertragen. Jedes Ensemble wird in Form eines OFDM-Multiplexsignals mit entsprechend großer Bandbreite von 1,536 MHz ausgestrahlt [3]. Im UKW-Band (87,7 ... 108 MHz) werden bei dem heute üblichen FM-Modulationsverfahren hingegen einzelne Programme mit einer Bandbreite von circa 300 kHz ausgestrahlt [2]. Aufgrund der unterschiedlichen Struktur wäre eine zukünftige Umstellung von analogem Rundfunk

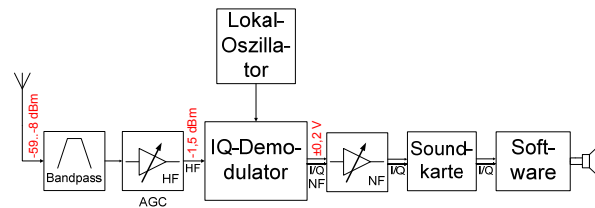


Abbildung 1: Konzept des SDR-Empfängers.

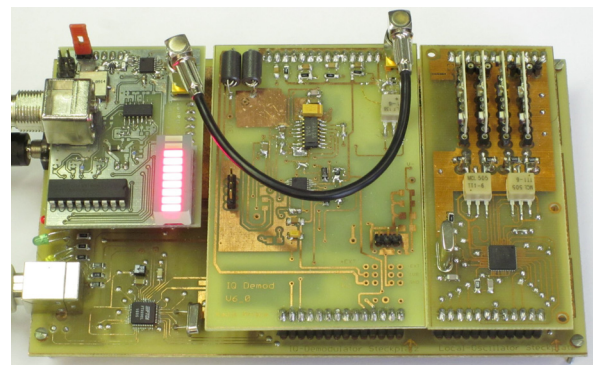


Abbildung 2: Hardware mit Mainboard und Aufsteckmodulen: HF-Modul (links), IQ-Demodulator (Mitte), DDS-Synthesizer (rechts).

auf digitale DAB+-Ausstrahlung im UKW-Band nur schlagartig möglich, da ein ganz anderes Kanalaraster verwendet werden müsste.

Die Alternative besteht darin, einzelne Sender auf digitalen Rundfunk umzustellen und damit nach und nach die analoge Ausstrahlung abzulösen. Das wäre mit DRM+ (Digital Radio Mondial+) im Übertragungsmodus E möglich, einer Weiterentwicklung von DRM für Frequenzen oberhalb von 30 MHz. Bei DRM+ kommt ein OFDM-Verfahren zum Einsatz, bei dem 213 orthogonale Subträger in einer Bandbreite von 100 kHz mit 4QAM oder 16QAM moduliert werden [4].

Der neu entwickelte Tuner führt eine IQ-Demodulation nach dem Prinzip des Tayloe-Detektors [5] (Abbildung 3) durch. Dabei kommt ein schneller Multiplexer/Demultiplexer zum Einsatz, der vom Institut für Entwurf integrierter Schaltkreise an der Hochschule Mannheim entwickelt wurde [6]. Die resultierenden Inphase- und Quadratur-Basisbandsignale werden mit einer Soundkarte digitalisiert und dann per Software digital demoduliert.

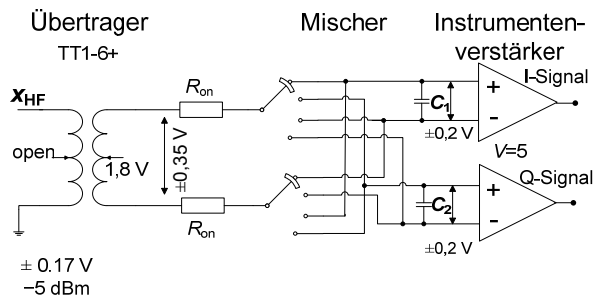


Abbildung 3: Prinzip des Tayloe-Detektors.

II. IQ-DEMULATORPRINZIP

Bei der IQ-Modulation werden zwei niederfrequente informationstragende Basisbandsignale $V_i(t)$ und $V_q(t)$, hier als Inphase- und Quadratur-Signale (I/Q) bezeichnet, jeweils mit einer Trägerschwingung gemischt und anschließend aufaddiert. Die beiden hochfrequenten Mischsignale sind gegeneinander um 90° phasenverschoben und haben die gleiche Trägerfrequenz:

$$U_{HF}(t) = V_i(t) \cdot \cos(2\pi f_0 \cdot t) - V_q(t) \cdot \sin(2\pi f_0 \cdot t)$$

Der IQ-Demodulator extrahiert die beiden informationstragenden Basisbandsignale im gewünschten Frequenzkanal und gibt sie für die Digitalisierung an die Soundkarte weiter.

Die gewählte Schaltungsrealisierung für den IQ-Demodulator entspricht einem Tayloe-Detektor [5] (Abbildung 3). Vom Prinzip her handelt es sich um die schaltungstechnische Interpretation eines zweikanaligen Ringmischers mit RC-Tiefpassfiltern. Die als Stromschalter wirkenden Dioden der Ringmischer sind durch High-Speed-Transistorschalter ersetzt.

Beim realisierten Tuner wird das Spektrum des gesamten UKW-Bands so verschoben, dass der ausgewählte Frequenzkanal auf einer einstellbaren, basisbandnahen Zwischenfrequenz liegt. Die Kanalselektion und die Extraktion seiner IQ-Signale werden direkt im Basisband mittels der RC-Tiefpässe realisiert.

A. Architektur des Demodulators

Abbildung 3 zeigt die Architektur des IQ-Demodulators. Er besteht aus einem Übertrager, einem High-Speed-Schalter und einem Instrumentenverstärker. Der Übertrager sorgt für das Biasing des HF-Eingangssignals auf 1,8 V und gibt es in differentieller Form an die High-Speed-Schalter weiter. Der eingesetzte Übertrager mit Übersetzungsverhältnis 1 wird bezüglich der anliegenden HF-Signale in 50Ω -Anpassung betrieben. Rechtsseitig sorgen die R_{on} -Durchgangswiderstände der beiden Schalter für die Anpassung. Sie betragen jeweils 25Ω , die Kapazitäten wirken bei hohen Frequenzen als Kurzschluss.

Zwischen Übertrager und Instrumentenverstärkern befindet sich die High-Speed-Schaltmatrix. Diese besteht aus zwei Einzelschaltern, die periodisch zwi-

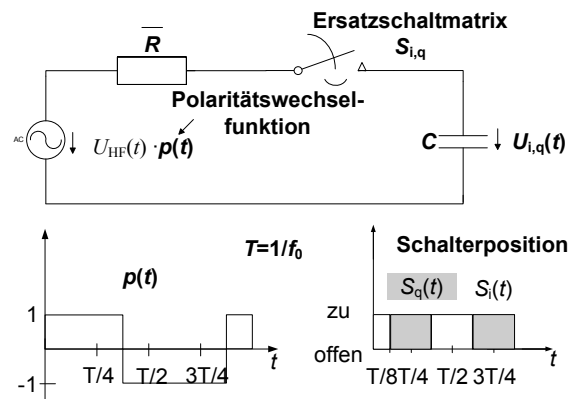


Abbildung 4: Ersatzschaltbild des Demodulators.

schens jeweils vier Schalterstellungen umschalten. Ein in DDS-Technik (direct digital synthesis) ausgeführter frequenzprogrammierbarer Lokal-Oszillator (LO) liefert die Steuersignale, so dass jeder Schalter während eines Viertels der LO-Periode in einer Schalterposition bleibt. Das HF-Signal wird in dieser Zeit an die Schalterausgänge übertragen und mit RC-Tiefpässen gefiltert. Deren Grenzfrequenz hängt von dem Wert des Kondensators C , den zwei R_{on} -Widerständen und vom HF-seitigen 50Ω -Widerstand ab. Die Grenzfrequenz wurde zu $f_g = 1/T_{RC} = 200 \text{ kHz}$ gewählt. Aufgrund des Schalterastverhältnisses erhöht sich die Filterzeitkonstante um den Faktor 4:

$$T_{RC} = 4 \cdot (50\Omega + 2R_{on}) \cdot C$$

B. Funktion des IQ-Schaltdemodulators

Abbildung 4 zeigt das Ersatzschaltbild eines Zweiges (I bzw. Q) des IQ-Demodulators. Dabei lassen sich der HF-seitige 50Ω -Widerstand und die beiden R_{on} -Durchgangswiderstände der Schalter bei geschlossenen Schaltern durch $\bar{R} = 100 \Omega$ darstellen. Die Wirkung der Schalter wird für eine einfache Analyse durch die Polaritätswechselfunktion $p(t)$, die HF-Quellspannung $U_{HF}(t)$ und die Ersatzschalterfunktionen $S_i(t)$ bzw. $S_q(t)$ dargestellt. $U_i(t)$ und $U_q(t)$ stellen die Spannungen an den Kondensatoren $C_1 = C_2 = C$ dar. Die Polaritätswechselfunktion $p(t)$ bildet die Invertierung des Signals nach einer halben Periode ab. In jeder Schalterposition wird immer nur einer der beiden Kondensatoren C_1 oder C_2 durch die Ersatzquellspannung $U_{HF}(t) \cdot p(t)$ aufgeladen. In dieser Zeit fließt kein Strom in den anderen Kondensator, so dass dort die Spannung gehalten wird.

Da $T_{RC} \gg T$ ist, ändert sich die Spannung am Kondensator während eines Schalterzyklus kaum. Daher bilden $U_i(t)$ und $U_q(t)$ in guter Näherung die Mittelwerte der Ersatzquellfunktion $U_{HF}(t) \cdot p(t)$ während des durch $S_i(t)$ bzw. $S_q(t)$ beschriebenen Ladeintervalls. Für den I-Zweig lässt sich dieser Spannungsmittelwert durch Integration über die Kosinus-Träger-

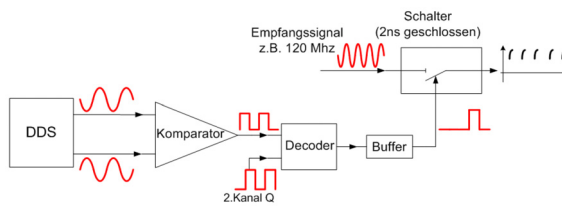


Abbildung 5: Ansteuerung der Schalter im IC [6].

schwingung ermitteln, der Q-Anteil fällt in dem von $S_i(t)$ vorgegebenen Ladeintervall weg:

$$\begin{aligned} U_i(t) &= 4f_0 \int_{-1/(8f_0)}^{1/(8f_0)} V_i(t) \cdot \cos(2\pi f_0 \cdot t) dt \\ &\quad - 4f_0 \int_{-1/(8f_0)}^{1/(8f_0)} V_q(t) \cdot \sin(2\pi f_0 \cdot t) dt \\ &= 4f_0 \cdot V_i(t) \cdot \frac{1}{2\pi f_0} \sin(2\pi f_0 \cdot t) \Big|_{-1/(8f_0)}^{1/(8f_0)} - 0 \\ &= \frac{4 \cdot \sqrt{2}}{\pi} \cdot V_i(t) \end{aligned}$$

Analog lässt sich der Mittelwert für den Q-Zweig in dem von $S_q(t)$ gegebenen Ladeintervall des Kondensators berechnen.

$$\begin{aligned} U_q(t) &= -4f_0 \int_{1/(8f_0)}^{3/(8f_0)} V_q(t) \cdot \sin(2\pi f_0 \cdot t) dt \\ &= -\frac{4 \cdot \sqrt{2}}{\pi} \cdot V_q(t) \end{aligned}$$

Das komplexe Basisbandsignal nach Tiefpassfilterung ergibt sich damit näherungsweise zu

$$U_{NF}(t) = U_i(t) - jU_q(t) = \frac{4 \cdot \sqrt{2}}{\pi} [V_i(t) + jV_q(t)].$$

Daraus lässt sich der theoretische Demodulationsgewinn G ablesen:

$$G = 20 \cdot \log\left(\frac{4 \cdot \sqrt{2}}{\pi}\right) = 5,1 \text{ dB}$$

Der an der aufgebauten Schaltung messtechnisch ermittelte Gewinn beträgt lediglich $G = 4,8$ dB. Die Gewinnreduktion erklärt sich daraus, dass die Schalter, insbesondere bei höheren LO-Frequenzen, wegen realer Schaltflanken nicht über komplette Viertelperioden geschlossen sind.

C. IC-Funktion

Der in [6] speziell für die vorgestellte Anwendung entwickelte High-Speed-Schalter-IC integriert neben der Ansteuerlogik Transmission-Gates, die als Anlogschalter verwendet werden können. Die Ansteuerlogik erzeugt aus zwei differentiellen, um 90° phasenverschobenen LO-Schwingungen Rechtecksignale, die

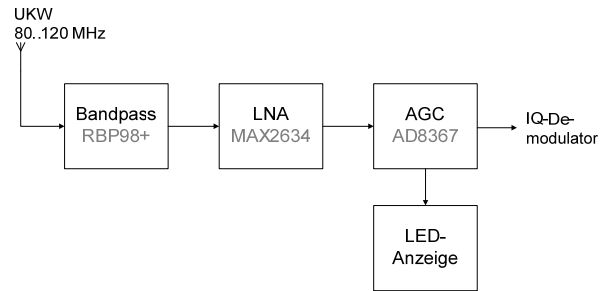


Abbildung 6: HF-Modul.

durch NAND-Gatter und Inverter verknüpft werden, so dass jeder Schalter während eines Viertels der LO-Periode an einer Position steht. Die Schwellspannung der integrierten Komparatoren liegt $\pm 0,3$ V relativ zu einer Referenzspannung von $1,8$ V, so dass die LO-Schwingungen ein entsprechendes Biasing benötigen. Der Schalter kann Schaltperioden von minimal 8 ns (maximal 125 MHz Schaltfrequenz) erreichen und somit in der vorgestellten Demodulatoranwendung das komplette UKW-Band abdecken. Abbildung 5 zeigt den inneren Aufbau des IC-Bausteins.

III. HARDWARE

Die Tuner-Hardware besteht aus einzelnen Modulen entsprechend der Abbildung 2. Das HF-Modul verstärkt und regelt die Leistung des Antennensignals vor der IQ-Demodulation. Der IQ-Demodulator erhält vom frequenzprogrammierbaren Lokal-Oszillator die beiden phasenverschobenen LO-Schwingungen und mischt entsprechend deren Frequenz ab. Nach der Demodulation ins Basisband oder auf eine niedrige Zwischenfrequenz (z. B. 48 kHz) sorgen programmierbare NF-Verstärker für eine gute Aussteuerung der AD-Umsetzer. Zur Digitalisierung der I- und Q-Signale dient eine breitbandige Soundkarte mit 192 kHz Abtast- und circa 85 kHz Grenzfrequenz. Verschiedene Software ermöglicht derzeit den Empfang von FM und DRM+.

A. HF-Modul

Ein Bandpass selektiert nach dem Antenneneingang das gesamte UKW-Band (Abbildung 6). Damit fallen unerwünschte Signale anderer Frequenzbänder schon vor der Verstärkung weg. Als erste Stufe wird ein rauscharmer Verstärker (LNA) eingesetzt. Anschließend wird die Signalleistung mit einem AGC-Baustein (Automatic Gain Control) auf ungefähr $-1,5$ dBm geregelt. Dem IQ-Demodulator wird ein breitbandiges Eingangssignal mit konstantem Pegel zur Verfügung gestellt. Die maximale Verstärkung des HF-Moduls liegt bei circa 58 dB. Wird das Eingangssignal kleiner als -59 dBm, sinkt die Ausgangsleistung entsprechend.



Eine Pegelanzeige greift die Steuerspannung der AGC ab und zeigt mit 10 LEDs den Eingangspegel an. Im Betrieb fällt auf, dass in typischen UKW-Empfangssituationen ein relativ hoher Eingangspegel zur Verfügung steht. Es wird ja nicht auf einen einzelnen Sender geregelt, sondern auf die breitbandige Gesamtleistung aller Sender im UKW-Band.

B. IQ-Demodulator mit Lokal-Oszillator

Der frequenzprogrammierbare DDS-Lokal-Oszillator liefert die phasenverschobenen LO-Schwingungen zur Ansteuerung der Schalter. Seine Funktion ist für die Qualität der Demodulation besonders wichtig. Die Steuersignale sollen für eine saubere Trennung des I- bzw. Q-Signals identische Amplitude haben und genau um 90° phasenverschoben sein. So wird eine hohe IQ-Entkopplung erreicht, ein Maß für die Trennschärfe der IQ-Demodulation. Die Amplitude dieser Steuersignale soll weit über den Schwellspannungen der Komparatoren im Schalter-IC liegen. Bei der Ansteuerung mit zu kleinen Amplituden ist das Verhalten des Schalters undefiniert.

C. Messungen

Zur Charakterisierung der Hardware wurde die Rauschzahl des Tuners gemessen und seine Nichtlinearität durch den IP3-Punkt beschrieben. Die Rauschzahl wurde nur am HF-Modul vermessen, da der Einfluss des nachfolgenden IQ-Demodulators vernachlässigbar ist. Die AGC hat einen fest definierten Ausgangspegel, weshalb der Verstärkungsfaktor durch die Leistung des Eingangssignals festgelegt werden kann. Das thermische Rauschen vom Eingang wird um diesen Verstärkungsfaktor angehoben, dazu kommt das Eigenrauschen der Schaltung. Die Rauschzahl entspricht dann dem Pegelunterschied zwischen verstärktem, thermischem Rauschen und dem Gesamtrauschen. Sie beträgt bei maximaler Verstärkung etwa 5,4 dB.

Der Third Order Intercept Point (IP3) ist ein Maß für die Nichtlinearität eines Tuners. Gemessen wird hier durch Anregung mit einem Zweitonsignal (f_1, f_2) [7]. Die Nichtlinearität des Tuners lässt sich anhand einer Potenzreihenentwicklung seiner Amplitudenkennlinie charakterisieren:

$$\hat{x}(t) = \beta_1 x(t) + \beta_2 x^2(t) + \beta_3 x^3(t) + \dots$$

Für den IP3-Punkt werden nur die Glieder erster und dritter Ordnung berücksichtigt. Mit trigonometrischen Funktionen lässt sich zeigen, dass durch Intermodulation 3. Ordnung zusätzliche, unerwünschte Signalanteile bei Frequenzen von $2f_1 - f_2$ und $2f_2 - f_1$ entstehen. Die Leistung dieser unerwünschten Signale steigt bei 10 dB erhöhtem Eingangspegel um 30 dB.

Wird die Leistung des Eingangssignals variiert, so ergibt sich beim Ausgangssignal theoretisch ein Schnittpunkt zwischen dem Nutzpegel und dem

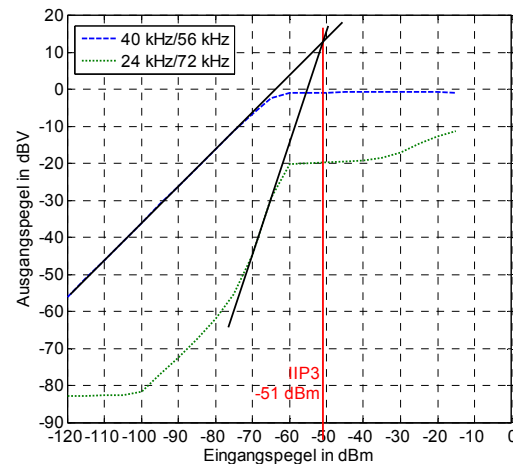


Abbildung 7: IP3-Messung am Gesamttuner.

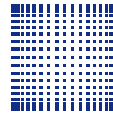
schneller wachsenden Pegel der unerwünschten Intermodulationsprodukte 3. Ordnung. Dieser Schnittpunkt wird als IP3-Punkt bezeichnet und kann auf die zugehörige Eingangs- (Input-IP3, IIP3) oder Ausgangsleistung (Output-IP3) bezogen werden.

Der gemessene IIP3-Punkt des HF-Frontends liegt bei -30 dBm, der des IQ-Demodulators bei 5 dBm. Da die Verstärkung des HF-Moduls bis zu 58 dB beträgt, wird der IP3-Punkt der Gesamtschaltung vom IQ-Demodulator bestimmt. Dessen IP3-Punkt wird um die hochfrequenzseitige Verstärkung verschoben. Erst bei starken Eingangssignalen kommt die Nichtlinearität des HF-Moduls ins Spiel. In Abbildung 7 ist das Messergebnis dargestellt. Der IIP3-Punkt liegt bei -51 dBm. Die Wirkung der Regelung setzt ab einem Eingangspegel von -59 dBm ein. Die unerwünschte Komponente mit Steigung 1 zwischen -100 und -70 dBm wird durch den verwendeten Signalgenerator erzeugt. Die Ermittlung des IP3-Punktes ist durch die durchgezogenen Linien dargestellt, die an die Kurven bei Steigung eins bzw. drei angelegt werden. Ab -40 dBm wirkt sich dann auch die Nichtlinearität der Verstärker aus.

Durch die Ergebnisse der IP3-Messung wird die Großsignalfestigkeit des Tuners für den Fall beschrieben, dass das Intermodulationsprodukt 3. Ordnung (zweier) leistungsstarker Nachbarkanäle als Störer in den zu selektierenden Nutzkanal fällt. Bleibt die Nachbarkanalleistung deutlich unterhalb des gemessenen IIP3 von -51 dBm, so kann die Störwirkung vernachlässigt werden. Der IIP3 bezieht sich also auf die Empfangsleistung einzelner Frequenzkanäle und ist damit für den praktischen Empfängerbetrieb hinreichend groß.

IV. FAZIT

Der entwickelte SDR-Tuner beweist seine Funktionstüchtigkeit im Laborbetrieb sowohl beim Empfang von FM als auch von DRM+. Dabei kann eine hervorragende Audio-Qualität erreicht werden.



Der speziell entwickelte High Speed-Schalter ist für die beschriebene Anwendung hervorragend geeignet und besitzt eine praxistaugliche Großsignalfestigkeit. Die Gesamtschaltung erreicht eine hohe Empfindlichkeit bei kleiner Rauschzahl. Eine Verbesserungsmöglichkeit wäre die Automatisierung der NF-Pegelregelung, so dass die Soundkarte immer optimal angesteuert wird. Bisher muss der Verstärkungsfaktor dazu manuell eingegeben werden.

DANKSAGUNG

Die Autoren bedanken sich beim Institut für Digitale Signalverarbeitung der Hochschule Mannheim für die Möglichkeit, die Bachelorarbeiten zu diesem interessanten Themengebiet anzufertigen, insbesondere bei ihrem Betreuer Herrn Prof. Dr.-Ing. U. Martin.

LITERATURVERZEICHNIS

- [1] T. Roupael, „RF and Digital Signal Processing for Software-Defined Radio“, *Elsevier*, Oxford 2009.
- [2] Bundesnetzagentur, „Frequenznutzungsplan“, August 2011, http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/BNNetzA/Sachgebiete/Telekommunikation/Regulierung/Frequenzordnung/Frequenznutzungsplan/Frequenznutzungsplan2011pdf.pdf?__blob=publicationFile (21.01.13).
- [3] W. Fischer, „Digitale Fernseh- und Hörfunktechnik in Theorie und Praxis“, *Springer-Verlag*, Berlin, Heidelberg 2010.
- [4] ETSI Standard, „Digital Radio Mondial (DRM); System Specification“, www.etsi.org/deliver/etsi_es/5C/201900_201999/5C201980/5C03.02.01_60/5Ces_201980v030201p.pdf (27.11.12).
- [5] D. Tayloe, „Ultra Low Noise High Performance, Zero IF Quadrature Product Detector and Preamplifier“, NorCal Club, www.norcalqrp.org/files/tayloe_mixer_x_3a.pdf (21.01.13).
- [6] C. Rahnke, J. Giehl, B. Vettermann, „Entwurf eines High-Speed Multiplexers/Demultiplexers für einen Mischer in 0,35 μm Technologie“, *Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, Karlsruhe, Juli 2009.
- [7] C. Rauscher, „Grundlagen der Spektralanalyse“, *Rohde&Schwarz*, München 2007.



Prisca Kadje Nwabo wird ihr Bachelorstudium in Nachrichtentechnik/Elektronik an der Hochschule Mannheim voraussichtlich im Februar 2013 abschließen. Sie befasste sich während ihrer Bachelorarbeit mit der Entwicklung des vorgestellten direktumsetzenden IQ-Demodulators.



Jonathan Mayer hat sein Bachelorstudium im Studiengang Nachrichtentechnik/Elektronik an der Hochschule Mannheim im Dezember 2012 abgeschlossen. Die Bachelorarbeit befasste sich mit der Entwicklung des beschriebenen HF-Moduls.

Optimierung eines 32-Bit-Softcores mit Harvard-Architektur und Dual Cache sowie Memory Management Unit für Embedded Computing

Michael Schmidt, Sebastian Stickel, Florian Zowislok, Dirk Jansen

Zusammenfassung—Die SIRIUS-Prozessorfamilie ist eine am Institut für Angewandte Forschung der Hochschule Offenburg entwickelte Prozessorreihe, welche ursprünglich zu Lehrzwecken entwickelt wurde. Sie umfasst zwischenzeitlich drei Softcores, die sich alle in Performance, Funktionalität und damit einhergehend auch in Komplexität unterscheiden. Der Schwerpunkt dieser Veröffentlichung liegt auf der leistungstärksten Variante der drei Softcores, dem SIRIUS HULK. Insbesondere die vorgenommenen Optimierungen an Hardware und Software sowie die Synthesergebnisse werden vorgestellt und erläutert.

Schlüsselwörter—SIRIUS, Prozessorkern, Prozessor, Softcore, Harvard Architektur, FPGA, Benchmarking, Synthese.

I. EINLEITUNG

Als Teil des Instituts für angewandte Forschung (IAF) entwickelt das ASIC Design Center der Hochschule Offenburg seit 1989 anwenderspezifische Schaltungen (ASICs). Ferner wurde vor einigen Jahren mit der Entwicklung eines hauseigenen Prozessorkerns namens SIRIUS für Forschung und Lehre der Studenten begonnen. Dieser verfügte seinerzeit über eine Von-Neumann-Architektur, einen 16 Bit breiten Datenbus sowie über einen beschränkten RISC-Befehlssatz und konnte sämtliche Register-Register Befehle in einem Taktzyklus abarbeiten. Darüber hinaus verfügte er über zwei Betriebsmodi, den 16- und den 32-Bit-Modus, die es ihm ermöglichten, optimal auf den Zieleinsatz abgestimmt zu werden. Diese Prozessorkernphilosophie wurde als Grundlage für eine ganze Prozessorfamilie verwendet, die nun unter dem Familiennamen SIRIUS weiterentwickelt wird. Der bisherige SIRIUS-Kern wurde in *SIRIUS Janus* umbenannt und kommt beispielsweise im hauseigenen PDA in UMC 0.18 μm Technologie zum Einsatz. Als kompaktesten Vertreter gibt es den *SIRIUS Tiny*, der über eine v. Neumann-Architektur und einen 16 Bit breiten Datenbus verfügt.

Michael Schmidt, michael.schmidt@hs-offenburg.de, Sebastian Stickel, sebastian.stickel@hs-offenburg.de, Florian Zowislok (Ehemaliger Mitarbeiter), Dirk Jansen, d.jansen@hs-offenburg.de, Hochschule für Technik, Wirtschaft und Medien – Offenburg, University of Applied Sciences, Badstraße 24, 77652 Offenburg.

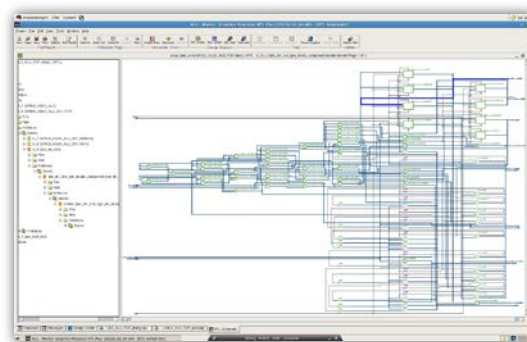


Abbildung 1: RTL Schematic der End-Synthese.

Ferner wurde der Befehlssatz weiter eingeschränkt, sodass dieser Prozessorkern auch zur Lehre der Studenten in einem Kurs eingesetzt werden kann, in dem diese den Core selbst in VHDL programmieren und somit erste tiefere Einblicke in moderne Prozessorarchitektur gewinnen können. Als performantester Vertreter tritt der Softcore *SIRIUS HULK* auf, da bei ihm auf den berühmten Flaschenhals der v. Neumann-Architektur verzichtet und stattdessen eine Harvard-Architektur zu Grunde gelegt wurde.

II. ÜBERSICHT SOFTCORES

Ein Unterscheidungsmerkmal der SIRIUS Softcores liegt neben der v. Neumann-Architektur bei Janus und Tiny und der Harvard-Architektur bei HULK auch in der Breite des Adress- und Datenbusses. Verfügt der Tiny sowohl über einen 16 Bit breiten Adress- als auch Datenbus, ist der Adressbus des Janus bereits 32 Bit breit, womit dieser volle 4 GB Speicher adressieren kann. Der SIRIUS HULK wiederum wurde konsequent als 32-Bit-Softcore entwickelt. Allerdings muss bei ihm auf Grund der Harvard-Architektur zunächst zwischen Daten und Instruktionen unterschieden werden. Beide Busse verfügen demnach über Daten- und Adressleitungen, welche bis auf den Instruktionsdatenbus 32 Bit breit sind. Letzterer verfügt lediglich über 16 Bit, da das Instruction Word Format bei allen drei Softcores einheitlich gehalten ist. Ein weiteres Alleinstellungsmerkmal des SIRIUS HULK ist der 16 kB große Daten- und Befehls-cache. Tabelle 1 fasst die Unterschiede und Gemeinsamkeiten der

Tabelle 1: Vergleich der SIRIUS Softcores.

	Tiny	Janus	HULK
Wortbreite	16 Bit	16/32 Bit	32 Bit
Architektur	Neumann	Neumann	Harvard
Datencache	-	-	16 kB
Befehlsache	-	-	16 kB
Befehlssatz	6 Bit RISC		
Datenbus	16 Bit	16 Bit	32 Bit (Daten)
Adressbus	16 Bit	32 Bit	32 Bit (Daten)
Anzahl Register	16		
Automat	Register-Register		

drei Softcorevarianten nochmals tabellarisch zusammen.

Als Anwendungsgebiete lassen sich beispielsweise eine Verwendung in der Lehre, als Recheneinheit in Taschenrechnern oder zur Sensorauswertung, zur Heimautomation oder auch als CPU in einem PDA aufführen. Der SIRIUS HULK ist darüber hinaus mit Hilfe seiner Hardware-Floating-Point-Einheit (FPU), seiner Hardware-Divisionseinheit und der Hardware-Multiplikationseinheit auch für rechenintensivere Aufgaben wie etwa Grafikausgaben oder, naheliegend durch die FPU, auch für Floating-Point-Berechnungen universell einsetzbar. Weitere detaillierte Informationen und Beschreibungen des SIRIUS HULK finden sich in früheren Veröffentlichungen im Rahmen der MPC Workshops [1], [2].

Während der Entwicklung der Softcores wurde stets Wert auf eine strikte Einhaltung des Hardware-Software-Co-Designs gelegt. Dieses Prinzip, dargestellt in Abbildung 2, beschreibt kategorisch und systematisch den Ablauf und die einzelnen Schritte der Entwicklung sowie deren logische Verknüpfung.

III. BENCHMARKING

Um eine Einschätzung der Leistung der entwickelten Softcores zu erhalten, wurden diese einem Benchmark unterzogen. Hierbei handelte es sich um den frei beziehbaren Benchmark *Coremark*¹. Ein weiterer wesentlicher Vorteil dieses Benchmarks neben der freien Verfügbarkeit ist, dass er eine realitätsnahe „Workload“, u.a. bestehend aus CRC-Berechnungen, Listenberechnungen und Listensortierungsalgorithmen sowie diversen arithmetischen Operationen auf Matrizen enthält. Die CRC-Checksummen werden zugleich als Selbsttest verwendet, um eine korrekte bzw. feh-

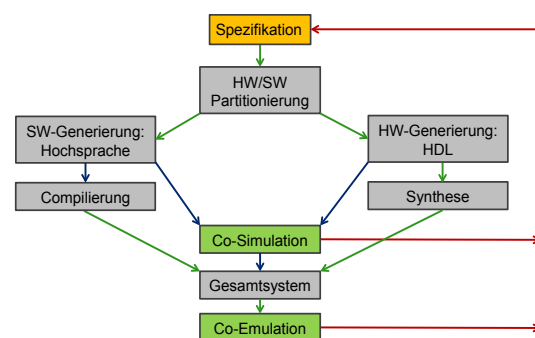


Abbildung 2: Hardware-Software-Co-Design.

lerhafte Codeausführung zu signalisieren. Der Benchmark-Sourcecode liegt in der Programmiersprache C vor und muss vor der Ausführung noch kompiliert werden. Hierzu gehört auch die Angabe, wie viele Iterationen die eigentliche Messschleife durchlaufen soll, denn der Benchmark sollte mindestens zehn Sekunden ausgeführt werden, um verlässliche Informationen liefern zu können. Ist dies garantiert, so erhält man schlussendlich den Benchmark-Wert, welcher in Coremark pro Megahertz (CM/MHz) angegeben wird. Wie ersichtlich ist, wird der erreichte Wert auf 1 MHz normiert, was eine Vergleichbarkeit mit anderen Prozessoren ermöglicht. Die Ergebnisse können auf oben genannter Homepage veröffentlicht werden.

Der Benchmark Coremark liefert, da er als Sourcecode vorliegt und somit vor Ausführung noch kompiliert werden muss, nicht nur Angaben über die Leistungsfähigkeit der Hardware, sondern bedingt durch die Compilierung auch zugleich Angaben über die Effektivität der Codeübersetzung. Bei dem hier getesteten System war dies von großem Vorteil, da als Compiler der frei verfügbare LCC-Compiler² einge-

¹ <http://www.coremark.org/>

² <http://www.cs.virginia.edu/~lcc-win32/>

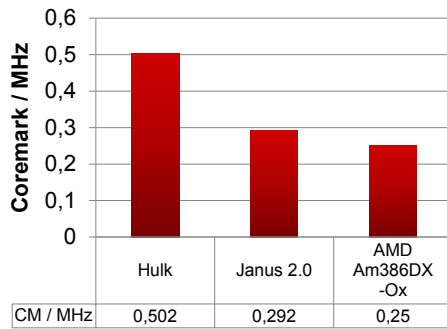


Abbildung 3: Optimierte Benchmark Coremark – Ergebnisse.

setzt wurde, welcher jedoch über keine Optimierungsoptionen, wie z.B. der GCC-Compiler, verfügt. Ein Nachteil besteht darin, dass zum einen nicht nur der verwendete Compiler mitgetestet wird, sondern auch dessen Optimierungsstufe, was einen direkten Vergleich der Benchmark-Ergebnisse erschwert.

Ausgehend von diesen Tatsachen erreichte der SIRIUS HULK einen Benchmark-Wert von 0,502 CM/MHz und der SIRIUS Janus 2.0 einen Wert von 0,292 CM/MHz. Wie in Abbildung 3 ersichtlich, ist der SIRIUS HULK damit bereits doppelt so leistungsfähig wie ein AMD Am386DX und in etwa 72% schneller als der SIRIUS Janus. Ausgehend von diesen Resultaten stellte sich heraus, dass Optimierungspotential vorlag, sowohl auf Seiten der Software, als auch auf Seiten der Hardware.

IV. OPTIMIERUNGEN

A. Software

Die Verbesserung auf Compiler-Seite bezieht sich auf die Einführung sogenannter Registervariablen. Der Unterschied zu den bisher verwendeten Stackvariablen besteht darin, dass eine im Code angelegte lokale Variable nicht mehr auf dem Stack angelegt wird, wodurch der Wert der Variablen jedes Mal bei Verwendung mit Hilfe von Load- und Store-Instructions geladen bzw. abgelegt werden muss, sondern stets in einem freien Prozessorregister gehalten wird. Hierdurch entfallen die Load- und Store-Instructions, was die Codeausführung beschleunigt. Diese Optimierung musste rein auf Seiten des Compilers vorgenommen werden und hat je nach Anzahl der benötigten lokalen Variablen Einfluss auf die Gesamtausführungszeit eines Programmes. Anhand des Benchmarks Coremark zeigte sich, dass es sich vor allem auf die Ausführungszeit der CRC-Berechnungen auswirkte und allgemein wie erwartet auf alle Codeabschnitte mit lokalen Variablen. Eine Begrenzung dieser Optimierung liegt einzig in der Verfügbarkeit freier Prozessorregister zum jeweiligen Zeitpunkt, an dem der Einsatz einer Registervariablen möglich wäre. Abbildung 4 zeigt beispielhaft die Verkürzung des auszuführenden Codes durch den Einsatz von Registervariablen.

```

3392 LDS RA,0+56 ; Wert von Stack laden
3393 XRA RA, RA ; Wert manipulieren (RA = 0)
3394 STB RB, RA ; Wert auf Stack speichern
...
3513 LDS RA,0+56 ; Wert von Stack laden
3514 INC RA ; Wert manipulieren (RA += 1)
3515 STB RB, RA ; Wert auf Stack speichern

...

3393 XRA RA, RA ; Wert manipulieren (RA = 0)
3514 INC RA ; Wert manipulieren (RA += 1)

```

Abbildung 4: Einsatz von Registervariablen.

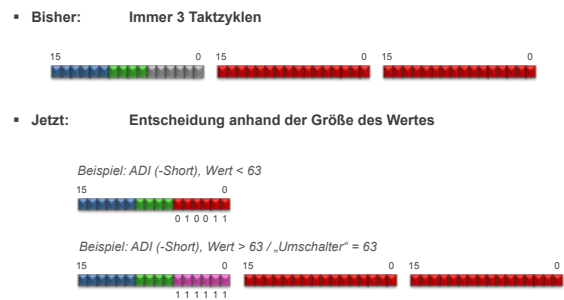


Abbildung 5: Überladung der Instructions.

B. Hardware

Die Optimierung der Hardware bestand darin, verschiedene Instructions zu überladen. Das Prinzip dahinter wird in Abbildung 5 anhand des Befehls ADI (Add Immediate) verdeutlicht. Bisher wurde die zum Quellregister (grün coloriert) zu addierende Konstante stets in zwei getrennten 16-Bit-Worten übertragen (rot), unabhängig von der Größe der Konstanten. Da es sich bei ADI jedoch um einen Immediate-Befehl handelt, sind in diesem Falle jedoch keine Opcode-Bits für ein Zielregister erforderlich. Somit sind insgesamt sechs Bits innerhalb des Opcodes ungenutzt (in Abbildung 5 grau dargestellt). Bei der Überladung werden diese ungenutzten Bits nun dazu verwendet, Konstanten zu definieren. Ist die Konstante größer als sechs Bit, wird anhand eines „Umschalters“ dem Prozessor mitgeteilt, dass die zwei darauffolgenden 16-Bit-Worte die Konstante darstellen. Hieraus ergibt sich eine verkürzte Ausführungszeit für all diejenigen Konstanten, die einen Wert kleiner 63 besitzen. Der Wert 63 dient in diesem Fall als „Umschalter“. Somit können bei der Überladung im Idealfall 2 Zyklen pro überladener Instruction gespart werden.

Bei Sprungbefehlen fällt die ungenutzte Bitbreite und somit die innerhalb eines Opcodes übertragbare Konstante mit 10 Bit noch größer aus, sodass hieraus ein maximaler Sprung von ± 1022 realisierbar ist. Der Wert +1024 dient als „Umschalter“ in diesem Falle. Es gilt zu beachten, dass pro Konstanten-Wert immer um zwei Byte (eine Instruction) gesprungen wird und hieraus der Sprungbereich von ± 1022 resultiert.

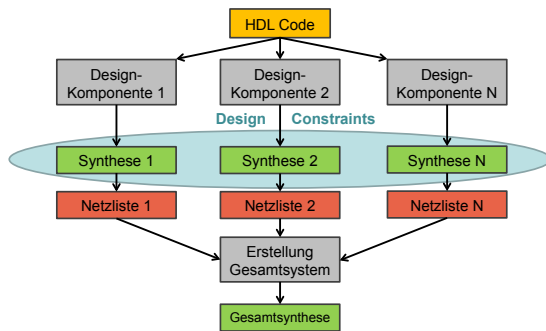


Abbildung 6: Synthese.

V. SYNTHESE

Die Synthese ist ein wesentlicher Bestandteil der Entwicklung und hat großen Einfluss auf die Gesamtfrequenz, mit der ein System betrieben werden kann. Im Idealfall wurde das Gesamtsystem schrittweise in mehrere Module gegliedert, die alle einzeln synthetisiert und somit auch optimiert werden können. Auch bei der Softcore-Entwicklung wurde auf diese Weise vorgegangen. Somit kann jedes Teilsystem einzeln auf maximale Frequenz optimiert werden. Die nach dieser Teilsynthese erhaltenen Netzlisten dienen anschließend als Basis für die Erstellung des Gesamtsystems, welches abermals optimiert werden kann bzw. sollte. Abbildung 6 stellt diesen Ablauf nochmals graphisch dar.

Die eigentlichen Optimierungen werden bei einer Synthese mit Hilfe sogenannter Timing Constraints getroffen, mit denen verschiedenste Einstellungen getroffen werden können. Beispielsweise können ganze Signalpfade von den Optimierungen ausgenommen werden, wie etwa asynchrone Reset-Logiken der Flip-Flops. Alle Constraints an dieser Stelle aufzuführen würde den Rahmen sprengen. Einen guten Überblick gibt [3].

Um korrekte Syntheseergebnisse zu erhalten, sollte der Systemdesigner möglichst alle wichtigen Pfade per Timing Constraints optimieren. Beim SIRIUS HULK zählt hierzu beispielsweise der Datenpfad in der Arithmetisch-Logischen-Einheit (ALU) des Hardware-Divisors. Aufgrund seiner Konfiguration hat dieser für eine Berechnung sieben Systemtakte Zeit, das Ergebnis zu berechnen. Diese sieben Takte müssen auf diesem Pfad der Synthese über einen *set_multicycle_path*-Constraint mitgeteilt werden. Erst mit diesem weiß die Synthese, dass dieser Pfad im Worst-Case-Fall nicht den kritischen Pfad darstellt und kann sich deshalb der Aufgabe der Optimierung anderer, wirklich kritischer Pfade, widmen. Im Allgemeinen muss der Systemdesigner jedoch auch darauf achten, das System nicht zu beschränken. Es sollten stets realistische Vorgaben gemacht werden, insbesondere in Bezug auf gewünschte Zielfrequenzen, da ansonsten negative Effekte auftreten können.

Tabelle 2: Syntheseergebnisse – F_{max} .

	Erreichte Frequenz [MHz]	Zielfrequenz [MHz]
ALU	250,5	250
Control Unit	111,5	
HULK Core	108,0	
Cache	132,8 / 1287	
MMU	112,8	

Tabelle 2 gibt einen Überblick über die erreichten Maximalfrequenzen der einzelnen Systeme. Wie ersichtlich ist, wäre mit dem Gesamtsystem eine Maximalfrequenz von 108 MHz auf einem FPGA des Typs Altera Cyclone IV E – EP4CE115F29C7N möglich. Jedoch muss beachtet werden, dass die MMU über einen rein kombinatorischen Aufbau verfügt und somit deren Delay dem des Cores hinzugefügt werden muss, sodass sich in Kombination eine Maximalfrequenz von etwa 55 MHz ergeben würde. Im Vergleich mit aktuellen Mikrocontrollern wäre dies eine respektable Systemfrequenz (vgl. Microchip dsPIC mit ca. 40 MHz). Allgemein kann gesagt werden, dass eine direkte Gegenüberstellung von SoCs bzw. ICs allerdings immer nur dann Sinn ergibt, wenn sie in gleicher Technologie gefertigt bzw. emuliert wurden.

VI. AUSBLICK

In naher Zukunft wird die vor kurzem implementierte FPU noch genaueren Tests unterzogen und dauerhaft mit in die Synthese eingebunden. Weiterhin kann die maximale Systemfrequenz mit Hilfe präziserer Timing Constraints innerhalb der Synthese erhöht und maximiert werden.

Ebenfalls wäre ein SIRIUS HULK-ASIC denkbar, welcher den eigentlichen Core, den Cache und die MMU enthält. In diesem Zuge wäre ebenso die Implementierung eines (seriellen) Bussystems wie PCI-Express möglich, um das klassische CPU-Northbridge-Southbridge-Prinzip zu erfüllen. Hierdurch könnte Peripherie schnell und problemlos an den HULK-ASIC angebunden werden.

Unabhängig hiervon wäre ein Multicore-System, basierend auf den SIRIUS HULK-Cores, denkbar und durch die Einbindung desselben in Altera Qsys realisierbar. Allerdings wäre somit die sinnvolle Multicore-Unterstützung seitens der Software und des Compilers Voraussetzung.

VII. FAZIT

Der komplexeste der drei SIRIUS Softcores des Instituts für Angewandte Forschung der Hochschule Offenburg wurde einem Benchmark-Test unterzogen. Hieraus wurden anhand der Ergebnisanalyse diverse Optimierungsmöglichkeiten deutlich, die systematisch umgesetzt wurden. Ferner wurden die Timing Constraints für die Synthese präzisiert, wodurch sich eine höhere Gesamtsystemfrequenz ergab.



Michael Schmidt erhielt den akademischen Grad des M. Eng in Elektro- und Informationstechnik im Jahr 2012 von der Hochschule Offenburg und arbeitet derzeit im Institut für Angewandte Forschung der Hochschule Offenburg.

DANKSAGUNG

Besonderer Dank gilt an dieser Stelle allen Mitarbeitern des ASIC Design Centers und des Instituts für Angewandte Forschung (IAF) der Hochschule Offenburg, die mir stets mit Rat und Tat zur Seite standen.

LITERATURVERZEICHNIS

- [1] M. Schmidt, „Integration des Altera Avalon Bus-Systems in einen 32 Bit Softcore mit Harvard Architektur und Dual Cache“, 47. *Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg, Offenburg, Feb.* 2012.
- [2] S. Stickel, D. Jansen, F. Zowislok, M. Durrenberger, „32 Bit Softcore mit Harvard Architektur und Double Cache“, 45. *Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg, Albstadt-Sigmaringen, Feb.* 2011.
- [3] Weng Fook Lee, „VHDL Coding and Logic Synthesis with Synopsis“, *Academic Press, 2000, ISBN: 0-12-388671-6.*

Ein präziser Signalgenerator mit 16-Bit-DAC auf einem Low-Power-FPGA

Philipp Harrer, Sven Kammerer, Thomas Wucher, Irenäus Schoppa

Zusammenfassung—Das hier beschriebene und auf einem Low-Power-FPGA realisierte System besteht aus einem 16-Bit-Digital-Analog-Wandler mit einer Bandbreite von 40 kHz und einem präzisen Sinusgenerator. Die Grundstruktur des Digital-Analog-Wandlers basiert auf einem mehrkanaligen Delta-Sigma-Modulator mit einem geeigneten Spannungsteiler sowie einem aktiven Tiefpassfilter vierter Ordnung. Der Sinusgenerator beinhaltet eine Vorrichtung zur besseren Stabilisierung der Signalfrequenzen auf der Basis eines erweiterten Phasenakkumulators mit einer variablen Modulo-N-Addition. Die Sinusfunktion wird in 16 Intervallen durch passende Regressionsgeraden nachgebildet.

Schlüsselwörter—Low-Power-FPGA, direkte digitale Signalsynthese DDS, Frequenzsynthese, Delta-Sigma-Modulation, Digital-Analog-Wandler.

I. EINLEITUNG

Während der Entwicklung digitaler DSP-Pfade in einem FPGA besteht oft die Notwendigkeit, Teilabschnitte einer DSP-Strecke mit definierten Signalen zu testen. Zu diesen Zwecken verwendet man häufig Signal- oder Vektorgeneratoren, die extern über Pins des FPGAs an den zu testenden Abschnitt angeschlossen werden. Der Einsatz solcher Signalgeneratoren ist allerdings problematisch, wenn im Zielsystem keine freien Pins mehr verfügbar sind, oder wenn keine Anschlusschnittstellen vorgesehen sind. Eine recht einfache Möglichkeit besteht darin, einen FPGA-internen Signalgenerator mit ausreichender Genauigkeit zu realisieren. Einen solchen Signalgenerator kann man nicht nur während der Entwicklungsphase einsetzen. Er ist auch später in der Diagnose und Wartung durchaus hilfreich.

Das hier beschriebene System wurde auf einem Low-Power-FPGA in der Flash-Technologie realisiert, und zeichnet sich durch eine Bandbreite von 40,0 kHz eine Auflösung von 16 Bit, mehrkanalige Delta-Sigma-Modulation und einen minimalen Einsatz ex-

P. Harrer, phharrer@htwg-konstanz.de, S. Kammerer, svkammer@htwg-konstanz.de, und T. Wucher, thwucher@htwg-konstanz.de, sind Studenten an der HTWG Konstanz, I. Schoppa, ischoppa@htwg-konstanz.de, ist Mitglied der HTWG Konstanz, Brauneggerstr. 55, 78462 Konstanz.

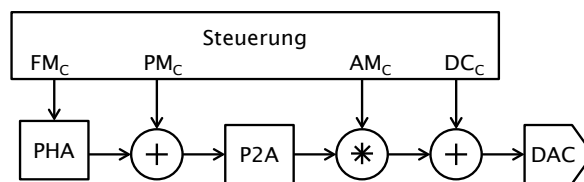


Abbildung 1: Typische Anordnung von Funktionsblöcken für die direkte, digitale Signalsynthese, bestehend aus einer Steuerungseinheit, einem Phasenakkumulator PHA, einem Phase-Amplitudenwandler P2A und einem Digital-Analog-Wandler DAC.

terner Komponenten aus. Der realisierte Sinusgenerator hat eine Amplitudenaufösung von 16 Bit und eine Phasenaufösung von $2\pi/8192$ rad. Diese Phasenaufösung resultiert aus der Tatsache, dass der eingesetzte FPGA-Baustein mit einem 128 Byte Flash-Speicher ausgestattet ist, in dem Korrekturwerte für die Sinusfunktion abgelegt sind.

II. DIREKTE DIGITALE SIGNALSYNTHESE

Die Abbildung 1 zeigt einen allgemeinen Datenpfad der direkten, digitalen Signalsynthese in einem FPGA. Mit diesem Modul lassen sich die drei gängigen Modulationsarten realisieren: die Frequenz-, Phasen- und Amplitudenmodulation. Die Auswahl der jeweiligen Modulationsart (auch in Kombinationen) erfolgt im Steuerungsmodul durch Änderungen von Einstellungswerten. Auf weitergehende Beschreibung der direkten, digitalen Signalsynthese wird an der Stelle verzichtet, und der Leser auf umfassende Literatur zur dieser Thematik z.B. [1], [2] und [3] verwiesen. Auf die Realisierung der drei Funktionsblöcke, nämlich des Phasenakkumulators PHA, des Phase-Amplitudenwandlers P2A und des Digital-Analog-Wandlers DAC wird in diesem Beitrag aber genauer eingegangen.

III. DIGITAL-ANALOG-WANDLER

Für die Umsetzung digitaler Information in analoge Signale stehen dem Anwender heute mehrere Möglichkeiten zur Auswahl:

- Diskrete Digital-Analog-Wandler
- Gewichtete Widerstände
- R-2R-Widerstandsnetze
- Pulsweiten-/Delta-Sigma-Modulation

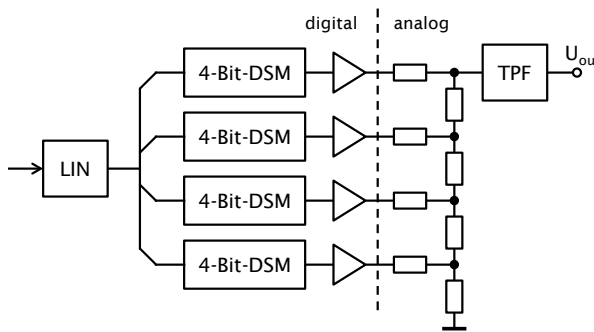


Abbildung 2: Digital-Analog-Wandler, bestehend aus vier parallelen 4-Bit-Delta-Sigma-Modulatoren DSM, einem Linearisierungsmodul LIN, einem Spannungsteiler aus Widerständen und einem analogen Tiefpassfilter TPF.

Vor allem die Pulsweitenmodulation PWM oder auch die Delta-Sigma-Modulation DSM erster Ordnung finden heute in FPGA-basierenden Systemen weit verbreitete Anwendung, siehe [4], [5]. Laut [6] zeichnen sich solche Realisierungen durch einen vergleichsweise relativ geringen Hardware-Aufwand aus. Im FPGA benötigt man einen Modulo-Zähler (und bei PWM auch einen Komparator), außerhalb sind einige analoge Elemente wie Widerstände und Kondensatoren, und bei aktiven Filtern noch Operationsverstärker notwendig. Nachteilig bei einer einkanaligen PWM- oder DSM-Realisierung ist eine hohe Taktfrequenz. Beispielsweise benötigt ein einkanaliger PWM-DAC eine Taktfrequenz von mehr als 2,62 GHz, wenn er eine Bandbreite von 40 kHz und eine Auflösung von 16 Bit unterstützen soll.

In einigen Applikationsberichten sind Realisierungen von Digital-Analog-Wandlern auf der Basis zweikanaliger Pulsweitenmodulation mit Widerstandnetzwerken, z.B. 15-Bit-DAC mit 6- und 9-Bit-PWM [7], 12-Bit-DAC mit zwei 6-Bit-PWM [8], oder 16-Bit-DAC mit zwei 8-Bit-PWM [9] beschrieben.

A. Delta-Sigma-Modulatoren

Damit der hier spezifizierte Digital-Analog-Wandler eine Bandbreite von 40 kHz und eine Auflösung von 16 Bit bei einer Taktfrequenz von 20 MHz erreichen kann, ist es notwendig, den DAC mit vier parallel geschalteten 4-Bit-Delta-Sigma-Modulatoren aufzubauen. Dadurch wäre es möglich, die Taktfrequenz bis auf 640 kHz zu reduzieren. Über die Ausgänge der einzelnen 4-Bit-Delta-Sigma-Modulatoren wird direkt ein Spannungsteiler angesteuert, der von einem R-2R-Widerstandnetzwerk abgeleitet ist. Ein Blockschaltbild dieser Anordnung ist in Abbildung 2 zu sehen. Damit aus einem PWM- oder DSM-Signal ein analoges Signal rekonstruiert wird, ist zusätzlich ein analoges Tiefpassfilter TPF notwendig. In einfachsten Fällen, in denen keine großen Anforderungen an die Qualität des rekonstruierten Signals gestellt sind, z.B. hinsichtlich der Restwelligkeit, können solche Tiefpassfilter aus passiven Elementen, z.B. aus einzelnen RC-Gliedern, aufgebaut sein. Für anspruchsvollere

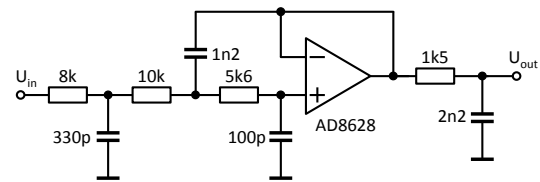


Abbildung 3: Analoges Tiefpassfilter vierter Ordnung, basierend auf der Sallen-Key Architektur.

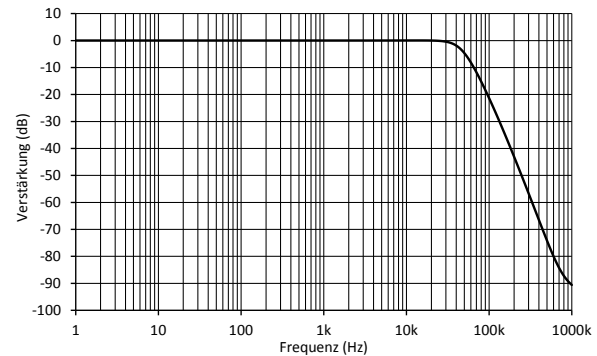


Abbildung 4. Amplitudengang des Tiefpassfilters vierter Ordnung.

Anwendungen sind passive Filter nicht ausreichend. Dazu müssen meistens aktive, oft auch mehrstufige Tiefpassfilter auf der Basis präziser Operationsverstärker eingesetzt werden, so wie in [10] beschrieben ist.

B. Tiefpassfilter

Das hier eingesetzte analoge Tiefpassfilter, dessen Schaltplan in Abbildung 3 zu sehen ist, basiert auf der Sallen-Key-Architektur vierter Ordnung aus [11]. Widerstände und Kondensatoren in diesem Filter wurden für eine Grenzfrequenz von 40,0 kHz nach Design-Regeln aus [12] und [13] berechnet. Anschließend wurde dieses Filter mit dem Spice-Modell des Operationsverstärkers AD8628 mit Hilfe der LTSpice-Simulation weiter optimiert und an gängige Widerstands- und Kondensator-Reihen angepasst. Dadurch hat sich die Grenzfrequenz auf 44,74 kHz erhöht. Der Amplitudengang des Filters ist in Abbildung 4 zu sehen.

Der verwendete Operationsverstärker AD8628 ist ein chopper-stabilisierter Präzisionsoperationsverstärker, der sich durch eine sehr niedrige Eingangs-Offsetspannung von max. 5 μV , eine niedrige Eingangs-Offsetspannungsdrift von max. 0,02 $\mu\text{V}/^\circ\text{C}$, ein geringes Rauschen von 22 $\text{nV}/\sqrt{\text{Hz}}$ und eine Bandbreite von 2,5 MHz auszeichnet. Der AD8628 kann mit einer unsymmetrischen Versorgungsspannung von +2,7V bis +5,0V betrieben werden. Weitere technischen Details findet man in [14].

C. Linearisierung

Eine der aufwändigsten Aufgaben ist die Linearisierung des Digital-Analog-Wandlers, also die Kompensation von Nichtlinearitäten, vor allem unter

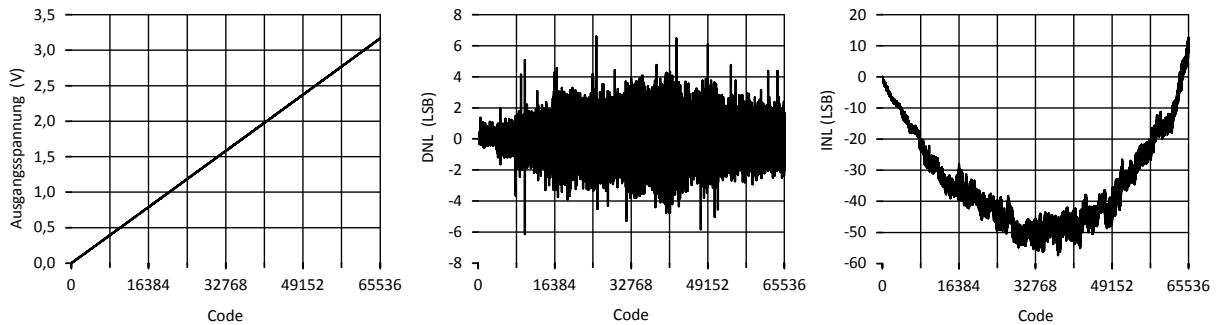


Abbildung 5: Drei für Digital-Analog-Wandler typische Diagramme: die Ausgangsspannung, der differentielle Nichtlinearitätsfehler DNL und der integrale Nichtlinearitätsfehler INL, alle drei in der Abhängigkeit vom digitalen Eingangswert.

dem Aspekt eines möglichst geringen Hardware-Aufwands. Dazu ist es notwendig, die Ausgangsspannung des DACs in der Abhängigkeit von digitalen Eingangswerten im gesamten Wertebereich von 0 bis 65535 zu messen. Die Ausgangsspannung des DACs ist nur eine von mehreren charakteristischen Kenngrößen eines DACs. Zwei weitere Kenngrößen, nämlich der differentielle Nichtlinearitätsfehler DNL und der integrale Nichtlinearitätsfehler INL sind bei der Bewertung eines DACs äußerst hilfreich. Beide Kenngrößen lassen sich nach Vorgaben in [15], [16] und [17] aus der gemessenen Ausgangsspannung z.B. mit Excel berechnen. Nach einer genaueren Analyse der Daten wurde festgestellt, dass die Ausgangsspannung „kleine“ Sprünge in ihrem Verlauf hat. Diese Störungen sind durch Überlagerung von mehreren, periodischen Fehlern entstanden und treten bei Übergängen in 256-, 512- und 4096-Intervallen auf. In der für den DAC implementierten Lösung wurde eine dreistufige Linearisierung vorgenommen, so dass in jeder Stufe ein anderer Fehler kompensiert werden konnte. Die Resultate nach der Kompensation sind in den Diagrammen in Abbildung 5 dargestellt. Durch diese Maßnahmen konnte der DNL recht gut korrigiert werden, denn fast alle DNL-Werte liegen im Bereich ± 5 LSB. Der INL zeigt uns jedoch an, dass im DAC noch weitere Fehler enthalten sind, die mit den hier eingeführten Maßnahmen nicht vollständig korrigierbar sind.

IV. PHASENAKKUMULATOR

Im Rahmen der direkten digitalen Signalsynthese, so wie sie in zahlreichen Literaturquellen z.B. [1], [2], [3] beschrieben ist, erfolgt die Generierung der Phase meistens mit einem Phasenakkumulator, bestehend aus einem „langen“ Register und einem Additionsschaltnetz. Die Länge eines Phasenakkumulators resultiert aus der erwarteten Frequenzauflösung. In kommerziellen Funktionsgeneratoren, z.B. von Agilent, erreichen solche Phasenakkumulatoren eine Länge von 64 Bit [18]. Die Anzahl der Bitstellen im Phasenakkumulator ist gewöhnlich deutlich größer als das bei der Phasen-Amplituden-Umsetzung notwendig

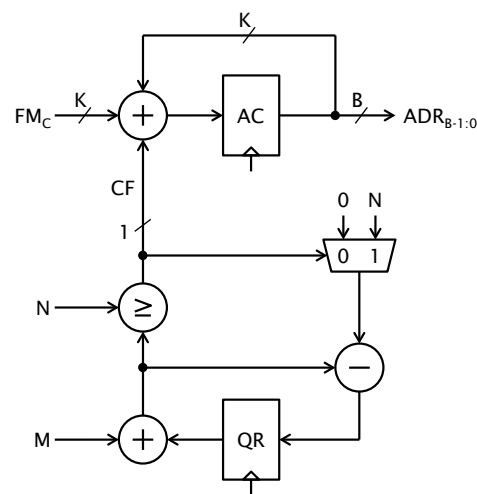


Abbildung 6: Die Registertransferstruktur des Phasenakkumulators AC mit Erweiterung um einen variablen Modulo-N-Zähler mit Hilfsregister QR.

gewesen wäre. Für die Phasen-Amplituden-Umsetzung werden nur die höherwertigen Bitstellen des Akkumulators verwendet. Die Bestimmung des Frequenzteilers FM_C aus der Länge K eines Phasenakkumulators, der Sampling-Frequenz F_S und der Ausgangsfrequenz F_O erfolgt mit folgender Formel:

$$FM_C = \left\lfloor 2^K \frac{F_O}{F_S} \right\rfloor$$

Ein Dezimalwert mit einer endlichen Anzahl von Nachkommastellen lässt sich nicht immer fehlerfrei in einen binären Wert mit einer endlichen Anzahl von Bitstellen konvertieren. Bei einer solchen Umrechnung bleibt oft ein Restfehler übrig. Das folgende Zahlenbeispiel kann diese Problematik gut veranschaulichen: für die Werte $K = 25$, $F_O = 15,0$ kHz und $F_S = 1,25$ MHz ergibt sich ein Frequenzteiler FM_C von 402651 mit einem Restfehler $\varepsilon = 0,184$. Mit dem errechneten Frequenzteiler wird sich eine geringfügig kleinere Ausgangsfrequenz, nämlich 14.999,9186 Hz einstellen lassen. Durch eine weitere Erhöhung der Länge des Phasenakkumulators kann man zwar den Restfehler verkleinern, aber i.d.R. nicht vollständig beheben.

Tabelle 1: Resultate aus Frequenzmessungen mit dem konventionellen 25-Bit-Phasenakkumulator und dem erweiterten 13-Bit-Phasenakkumulator mit einem 12-Bit-Register für den variablen Modulo-N-Zähler.

	PHA-25			PHA-13 + QR-12		
Soll-Frq. [Hz]	15.000,00	9.000,00	1.000,00	15.000,00	9.000,00	1.000,00
FM_C (M, N)	402653	241591	26843	98 (38, 125)	58 (614, 625)	6 (346, 625)
Min [Hz]	14.999,99	8.999,944	999,961	15.000,02	8.999,989	999,986
Max [Hz]	15.000,14	9.000,072	1.000,009	15.000,13	9.000,096	1.000,027
Avg [Hz]	15.000,07	9.000,007	999,985	15.000,07	9.000,045	1,000,005
SD [mHz]	26,962	27,173	7,266	16,013	17,134	6,256

A. Erweiterter Phasenakkumulator

Die Firma Analog Devices hat den DDS-Baustein AD9913 im Programm, in dem eine Erweiterung des Phasenakkumulators zwecks Verbesserung der Frequenzgenauigkeit vorgenommen worden ist [19]. Im Applikationsbericht [20] ist diese Erweiterung ansatzweise beschrieben. Sie basiert auf einer variablen Modulo-N-Operation, wo der Teiler N nicht an Werte in 2er-Potenz gebunden ist. Lässt sich der Restfehler ε aus der Berechnung von FM_C als rationale Zahl $\varepsilon = M/N$ darstellen, dann kann man mit dem Zähler M und Nenner N eine Modulo-Addition realisieren. Dabei wird ein zusätzlicher 1-Bit-Übertrag CF generiert, der als Korrekturwert zum Phasenakkumulator AC addiert wird. Die Berechnung der Phase erfolgt dann nicht durch eine relativ simple Modulo- 2^K -Addition, sondern wird durch die Folge von einzelnen Operationen aufgelöst, wobei QR ein Hilfsregister ist:

$$\begin{aligned} CF &= (QR + M \geq N) \\ QR &= (QR + M) \bmod N \\ AC &= (AC + FM_C + CF) \bmod 2^B \end{aligned}$$

Da eine variable Modulo-N-Operation, die nicht an die 2er-Potenz gebunden ist, sich aus einer VHDL-Beschreibung nicht synthetisieren lässt, ist es notwendig, diese Operation durch eine passende Schaltung, bestehend aus einem Addierer, einem Vergleicher und einem Subtrahierer aufzubauen. Die daraus resultierende Registertransferstruktur ist in Abbildung 6 zu sehen.

Die Darstellung des Restfehlers ε als rationale Zahl kann man relativ einfach mit einem HP-Taschenrechner, z.B. HP32Sii oder HP35S vornehmen. Diese Taschenrechner ermöglichen die Darstellung von Zahlen als Brüche, wobei zu berücksichtigen ist, dass der Nenner mit maximal 12 Bitstellen dargestellt wird.

Sobald die Werte M und N bekannt sind, besteht keine Notwendigkeit mehr, den Phasenakkumulator als „langes“ Register zu halten. Es reicht, wenn die Anzahl der Bitstellen an den Phasen-Amplituden-Wandler angepasst ist. Das folgende Zahlenbeispiel kann diese Thematik verdeutlichen: für die Werte $K = 13$, $F_O = 15,0$ kHz und $F_S = 1,25$ MHz ergibt sich nun

ein Frequenzteiler $FM_C = 98$, und ein Restfehler ε von 0,304, der diesmal aber durch die Werte $M = 38$ und $N = 125$ dargestellt wird. Mit diesen drei Werten erfolgt die Initialisierung des erweiterten Phasenakkumulators.

B. Frequenzmessungen

In der Tabelle 1 sind die Resultate der Frequenzmessung für drei exemplarisch gewählte Frequenzen 15,0 kHz, 9,0 kHz und 1,0 kHz zusammengefasst. Die Frequenzmessungen wurden zuerst mit einem herkömmlichen 25-Bit-Phasenakkumulator (PHA-25) durchgeführt, und anschließend mit dem 13-Bit-Phasenakkumulator mit einer 12-Bit-Erweiterung (PHA-13 + QR-12). Für jede Frequenzmessung wurden insgesamt 1000 Sampling-Werte mit dem 6,5-stelligen Multimeter DMM 4040 von Tektronix aufgezeichnet. Der Multimeter ist mit einer Analysefunktion ausgestattet, die es ermöglicht, innerhalb einer Messreihe den minimalen Messwert (Min), den maximalen Messwert (Max), den arithmetischen Mittelwert (Avg) sowie die Standardabweichung (SD) zu bestimmen. Neben den aus der Messung ermittelten Frequenzwerten enthält die Tabelle auch den jeweiligen Frequenzteiler FM_C sowie die Werte M und N für die Einstellung der Frequenz.

Die Standardabweichung SD selbst ist ein Maß für die Aussagekraft des arithmetischen Mittelwerts Avg. Eine kleine Standardabweichung bedeutet, dass alle Beobachtungswerte nahe am Mittelwert liegen. Bei einer großen Standardabweichung sind die Beobachtungswerte weit um den Mittelwert gestreut. Die Messresultate in der Tabelle 1 zeigen uns, dass die Fluktuation der Frequenz bei dem Phasenakkumulator mit Erweiterung geringer ist als bei dem konventionellen Phasenakkumulator.

V. PHASEN-AMPLITUDEN-WANDLER

Wegen der Symmetrieeigenschaften der Sinusfunktion werden bei deren Digitalisierung nur Funktionswerte aus dem Intervall 0 bis $\pi/2$ berechnet. In [21] und [22] findet man einen guten Überblick über ver-

schiedene Methoden zur Realisierung einer Sinusfunktion in digitalen Systemen. Das Quantisieren der Amplitude A erfolgt iterativ mit Hilfe der unten dargestellten Formel, wobei N die horizontale Auflösung, also die Anzahl der Stützstellen der Sinusfunktion im Intervall $0.. \pi/2$ und B die vertikale Auflösung der Amplitude sind:

$$A_k = \left\lfloor 2^B \sin\left(\frac{\pi}{2N}\left(k + \frac{1}{2}\right)\right) \right\rfloor \quad \text{für } k = 0.. N-1$$

Die so digitalisierte Sinusfunktion lässt sich in 16 Intervalle aufteilen, und in jedem Intervall i durch die Gleichung einer Regressionsgeraden $y_i = m_i x + b_i$ modellieren. Die Steigungen der Regressionsgeraden m_i und die Werte b_i , in denen die Regressionsgerade die Y-Achse schneidet, lassen sich abschnittsweise z.B. mit Excel leicht berechnen. In der Tabelle 2 sind die Ergebnisse dieser Umrechnung zusammengefasst.

Eine Rekonstruktion der Sinusfunktion erfolgt durch eine geeignete Spiegelung der Phase (binäre Negation) und der Amplitudenwerte (arithmetische Negation). In Abbildung 7 sieht man die Registertransferstruktur des Phasen-Amplituden-Wandlers. Eine Besonderheit in dieser Realisierung ist eine „kleine“ Tabelle mit Korrekturwerten. Ein Low-Power-FPGA in der Flash-Technologie verfügt über einen 128-Byte-Flash-Speicher, in dem der Anwender applikationsspezifische Werte ablegen kann. Da die Modellierung der Sinusfunktion mit Hilfe von Geraden mit kleinen Fehlern behaftet ist, lassen sich 128 Korrekturwerte als Durchschnittswerte der Differenzen zwischen der mathematischen Sinusfunktion und deren linearisierten Version berechnen, im Flash-Speicher ablegen und bei der Rekonstruktion der Sinusfunktion berücksichtigen.

VI. EVALUIERUNGSUMGEBUNG

Die Entwicklungs- und Experimentierumgebung umfasste mehrere Softwarewerkzeuge, zwei Entwicklungsplatinen und zwei Messinstrumente. Die Funktionalität des Digital-Analog-Wandlers und des Sinusgenerators wurden in VHDL modular, hierarchisch und synthesesgerecht beschrieben und mit dem VHDL-Simulator ModelSim von MentorGraphics umfassend simuliert. Für die Schaltungssynthese wurde die integrierte Entwicklungsumgebung Libero IDE 9.1 von Microsemi verwendet. Das Softwarewerkzeug LTSpice IV von LinearTechnology kam bei Design und Simulation der selbstentwickelten Aufsteckplatine mit dem Widerstandsnetzwerk und dem Tiefpassfilter zum Einsatz.

Als Basishardware diente das Entwicklungsboard IGLOO-Nano-Starterkit [23] mit dem Low-Power-FPGA AGLN250V2 von Microsemi. Dieses Entwicklungsboard verfügt über mehrere Stiftleisten, über die man die Aufsteckplatine auf dem Board betreiben

Tabelle 2: Nachbildung der Funktion $\sin(x)$ mit Hilfe von Geraden $y_i = b_i x + m_i$ in 16 Intervallen.

i	m	b
0	25	14
1	25	3228
2	24	6411
3	24	9532
4	23	12562
5	22	15471
6	20	18230
7	19	20815
8	17	23198
9	15	25358
10	13	27274
11	11	28928
12	8	30303
13	6	31386
14	4	32166
15	1	32637

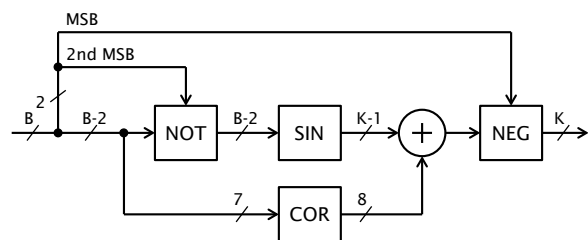


Abbildung 7: Die Registertransferstruktur des Phase-Amplituden-Wandlers, NOT – binäre Negation, NEG – arithmetische Negation, COR - Tabelle mit Korrekturwerten.

kann. Der Low-Power-FPGA wird über einen speziellen Download-Adapter programmiert.

Messungen wurden mit einem präzisen 6,5-stelligen Multimeter DMM 4040 von Tektronix und einem 4-kanaligen Mixed-Signal-Oszilloskop DLM2024 von Yokogawa durchgeführt. Mit dem Multimeter konnte der gesamte Spannungsbereich in einer Auflösung von $10 \mu\text{V}$ gemessen werden. Das Mixed-Signal-Oszilloskop diente zur Visualisierung analoger Signale und dank der eingebauten FFT-Funktion zur Spektralanalyse. Bei der Auswertung der Messdaten kam häufig Excel zum Einsatz.

VII. AUSBLICK

Die Entwicklung des hier vorgestellten Systems ist noch nicht vollständig abgeschlossen. Durch neue, aus Messungen und Analyse gewonnene Erkenntnisse lassen sich verbesserungswürdige Punkte angeben:

- Einsatz einer präziseren Referenzspannung
- Berücksichtigung des Ausgangswiderstandes in I/O-Treibern im FPGA
- Verbesserung der Linearität im DAC durch Polynom 2. Grades
- Untersuchung hinsichtlich Reproduzierbarkeit und Temperaturstabilität
- Höherwertigere Rekonstruktion des Signals durch Tiefpassfilterung

LITERATURVERZEICHNIS

- [1] B.-G. Goldberg: „Digital frequency synthesis demystified“, *LLH Technology Publishing*, 1999.
- [2] Analog Devices: „A Technical Tutorial on Digital Signal Synthesis“, *Analog Devices, Inc.*, 1999.
- [3] W. Kester: „Fundamentals of Direct Digital Synthesis (DDS)“, Tutorial MT-085, *Analog Devices, Inc.*, 2009.
- [4] J. Logue: „XAPP154 Virtex Synthesizable Delta-Sigma DAC“, Applikationsbericht XAPP154, Xilinx, Inc. 1999.
- [5] B. Millier: „16-Bit PWM/PWM-DACs using One Digital PSo Block“, Applikationsbericht AN2266, *Cypress Semiconductor Corp.*, 2005.
- [6] R. Stein und J. Day: „D/A Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms“, Applikationsbericht AN655, *Microchip Technology Inc.*, 1997.
- [7] Epson: „15-Bit D/A Conversion by PWM“, Applikationsbericht PCM15, *Epson Europe Electronics GmbH*, 2005.
- [8] K. Tam und V. Chan: „A Low-Cost 12-Bits Speech CODEC Using the MSP430F13x“, Applikationsbericht SLAA131, *Texas Instruments, Inc.*, 2001.
- [9] S. Woodward und C. Hill: „Combine two 8-Bit outputs to make one 16-Bit DAC“, *EDN*, S. 85-86, Sep. 2004.
- [10] D. M. Alter: „Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x“, *Texas Instruments, Inc.*, 2008.
- [11] J. Karki: „Analysis of the Sallen-Key Architecture“, Applikationsbericht SLOA024B, *Texas Instruments, Inc.*, 2002.
- [12] J. Karki: „Active Low-Pass Filter Design“, Applikationsbericht SLOA049, *Texas Instruments, Inc.*, 2000.
- [13] R. Mancini: „Op Amps for Everyone Design Guide“, Applikationsbericht SLOD006A, *Texas Instruments, Inc.*, 2001.
- [14] Analog Devices: „AD8628/AD8629/AD8630 Zero-Drift, Single-Supply, Rail-to-Rail Input/Output Operational Amplifier“, Datenblatt, *Analog Devices, Inc.*, 2008.
- [15] Maxim: „INL/DNL Measurements for High-Speed Analog-to-Digital Converters (ADCs)“, Tutorial AN283, *Maxim Integrated Products*, 2001.
- [16] Texas Instruments: „Understanding Data Converters“, Applikationsbericht SLAA013, *Texas Instruments, Inc.*, 1995.
- [17] W. Kester: „Analog-Digital Conversion“, *Analog Devices, Inc.*, 2004.
- [18] Agilent: „Agilent 33220A 20 MHz-Funktions-/Arbiträr-signalgenerator“, Benutzerhandbuch 33220-90441, *Agilent Technologies Inc.*, 2007.
- [19] K. Gentile: „Direct Digital Synthesis (DDS) with Programmable Modulus“, Applikationsbericht AN-953, *Analog Devices, Inc.*, 2008.
- [20] Analog Devices: „Low Power 250 MSPS 10-Bit DAC 1.8 V CMOS Direct Digital Synthesizer“, Datenblatt AD9913, *Analog Devices, Inc.*, 2010.
- [21] L. Cordesses: „Direct Digital Synthesis: A Tool For Periodic Wave Generation (Part 1)“, *IEEE Signal Processing*, S. 50-54, July 2004.
- [22] L. Cordesses: „Direct Digital Synthesis: A Tool For Periodic Wave Generation (Part 2)“, *IEEE Signal Processing*, S. 110-112, Sept. 2004.
- [23] Actel: „IGLOO nano Starter Kit User’s Guide“, *Actel Corp.*, 2010.



Sven Kammerer studiert Technische Informatik im 6. Semester an der HTWG Konstanz. Im Rahmen des Studiums befasst er sich u.a. mit Softcore-Prozessoren und Schaltungsdesign für Low-Power-FPGAs. Neben seinem Studium arbeitet er in der Software-Entwicklung im Linux-Embedded-Bereich bei Atos Deutschland.



Philipp Harrer schloss 2008 die Ausbildung zum staatlichen geprüften Informationstechniker ab und studiert jetzt Technische Informatik im 6. Semester an der HTWG Konstanz. Im Rahmen des Studiums befasst er sich u.a. mit Softcore-Prozessoren und Schaltungsdesign für Low-Power-FPGAs. Neben seinem Studium arbeitet er als Softwareentwickler bei CADcoR GmbH.



Thomas Wucher studiert Technische Informatik im 6. Semester an der HTWG Konstanz. Im Rahmen des Studiums befasst er sich u.a. mit Softcore-Prozessoren und Schaltungsdesign für Low-Power-FPGAs. Neben seinem Studium beschäftigt er sich mit hardwarenaher Programmierung in C auf Linux-basierten, eingebetteten Systemen.



Irenäus Schoppa studierte Informatik an der Technischen Universität Berlin und erhielt dort im Jahre 1993 den akademischen Grad Dipl.-Informatiker. Im Jahre 1998 promovierte er dort zum Dr.-Ing.. Seit dem Jahr 2008 ist er Professor für Hardware-Software Codesign an der HTWG Konstanz.

VHDL Based Circuit Description to Transmit, Receive and Analyse USB Data Packets

María Guadalupe Salgado Suárez

Abstract—A peripheral component is described using VHDL to handle USB traffic at full speed (12 Mb/s). Synthesized and downloaded to a FPGA, it accepts, generates or analyses USB packets. The target platform is a Xilinx Spartan 3E, located on a FPGA prototyping board, running at a clock rate of 48MHz. The module supports the calculation or manipulation of CRCs. Furthermore, SOF packets can be activated to act as a host USB device. It could be used as an I/O peripheral of a Microcontroller to perform as an USB device running at full speed.

Index Terms—VHDL, USB, FPGA, Xilinx Spartan 3E.

I. INTRODUCTION

As the usage of the Universal Serial Bus, or USB, popularizes so does the need to retrieve the packets sent through it. All sorts of electronic devices have now implemented this standard and in order to analyse their functionality, the data transmitted needs to be analysed as well. Therefore a peripheral component for a microcontroller, to transmit and receive USB packets, was designed using VHDL.

The USB transfers a differential signal with the encoded clock (D+ and D-) in two wires and power (V+ and GND) over another two, constituting a 4-wire cable working with three different data rates:

- High-speed: with a signaling bit rate of 480 Mb/s
- Full-speed: with a signaling bit rate of 12 Mb/s
- Low-speed: with a signaling bit rate of 1.5 Mb/s

Throughout this document the focus is on the full-speed requirements.

II. USB DESCRIPTION

The USB is a polled bus, this means that there can only be one host and many devices connected to it, and that the host initiates all communications and assigns unique addresses [1], [2].

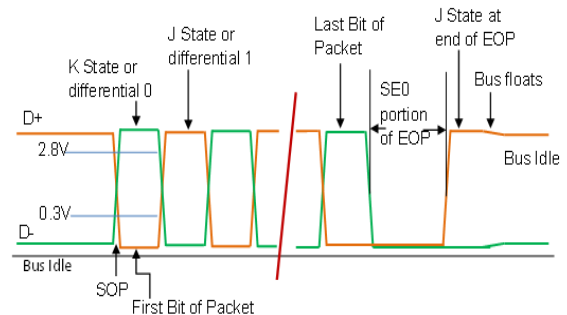


Figure 1: Signaling levels

A. USB Signaling

The devices must comply with different characteristics depending on the data rate they manage. In the case of full-speed devices, they are terminated with a pull-up resistor on the D+ line which has an impact on the signaling. Figure 1 provides the basic signaling levels to be fulfilled.

1) Non-Return-to-Zero Inverted Encoding

The USB uses NRZI to transmit the clock encoded along with the differential data. In NRZI encoding, a “1” is represented by no change in the levels and a “0” is represented by a change.

2) Bit Stuffing Principle

The bit stuffing principle consists of the insertion of a “0” bit after every six consecutive “1” in the data stream. It is used because, when a string of ones is NRZI encoded, there are no transitions in the data line and this ensures a transition at least once every seven bit times to guarantee the data and clock lock at the receiver.

B. USB Protocol

The bytes transmitted are moved over the bus in little endian order, i.e., LSB to MSB; the same applies for bits which are sent out onto the bus least-significant bit first.

Table 1: PID Types

PID Type	PID Name	PID bits	Fields
Token	OUT	0001	ADDR+ENDP
	IN	1001	ADDR+ENDP
	SOF	0101	Frame Number
Data	DATA0	0011	Data
	DATA1	1011	Data
Handshake	ACK	0010	---
	NACK	1010	---

1) SYNC Pattern

The synchronization pattern is a 0x80 byte sent before each packet. The Start-of-Packet SOP delimiter is part of the SYNC field.

2) Packet Identifier Field

The PID indicates the type of packet, its format and type of error detection. It is always the second byte sent and has its own error detection field. After the 4-bit packet type field follows the one's complement of it to ensure the correct decoding of the entire packet.

3) Address and Endpoint Fields

The address field or ADDR, it has 7 bits to specify the address of the function that can be either the source or destination of the packet.

The endpoint field or ENDP has four bits that, except for the control endpoint zero, are function-specific.

4) Frame Number Field

The frame interval used for full-speed devices is 1.00 ± 0.0005 ms and the host needs to send a packet every time a new frame begins. The frame number field is an 11-bit field that is incremented by the host each start of frame.

5) Data Field

The data field may range from zero to 1,024 bytes and must always be an integral number of bytes.

6) Cyclic Redundancy Check Fields

The CRCs provide protection for all single- and double-bit errors on all non-PID fields. They are generated with a specific polynomial before bit stuffing. There are 2 different CRCs used in USB: the CRC-5

Table 2: CRC's

Token Packets	
Generator Polynomial	$G(X) = X^5 + X^2 + 1$
Binary Bit Representation	00101B
Binary Bit Residual	01100B
Data Packets	
Generator Polynomial	$G(X) = X^{16} + X^{15} + X^2 + 1$
Binary Bit Representation	100000000000101B
Binary Bit Residual	1000000000001101B

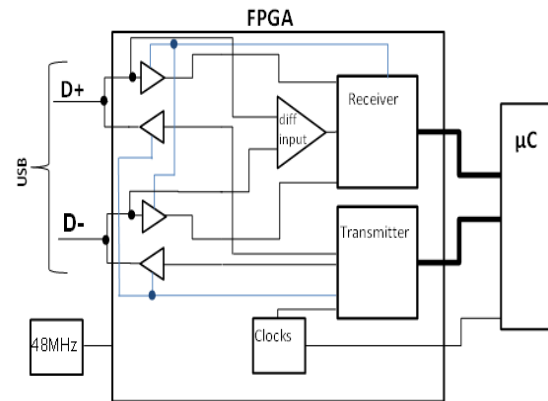


Figure 2: General Solution

used for the token packets and the CRC-16 used only for data packets; both are described in Table 2.

III. DESIGN AND SIMULATION

A. General Solution

The USB transceiver is divided into two main blocks, one for receiving data and another one to transmit it. Each of these blocks work independently from one another, both connected to the microcontroller by several control lines. A special block was created to generate different frequency clocks to use as reference and to keep track of the SOFs (see figure 2). Every block in the structure was simulated before the implementation.

B. Transmitter

The transmitter is in itself divided into 4 different structures: the interface with the microcontroller μC , the block for data packets, the block for token and handshake packets, and finally the block in charge of giving the proper format to the transmitted string. Each of these blocks are shown in Figure 3; the dotted

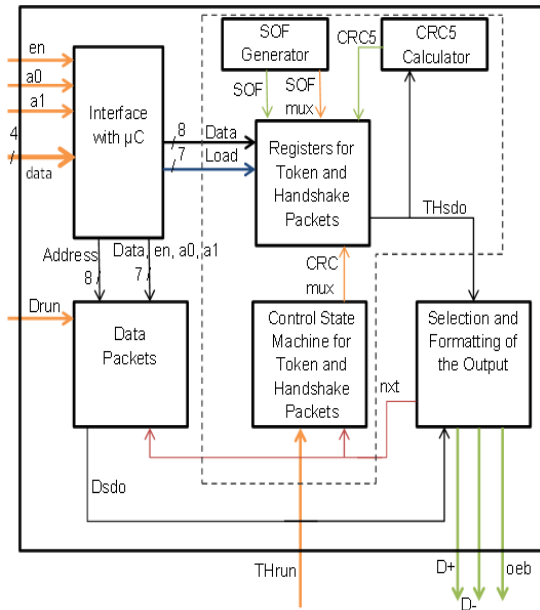


Figure 3: Transmitter

line encloses the whole token and data packet structure.

As inputs of the whole block there are 7 lines to write into the registers what wants to be transmitted, and two “run” lines (Drun & THrun) which enables the process to send either the data or the token and handshake packets. As outputs there are the two differential lines for the USB along with the oeb to control the output driver line if necessary.

1) Interface with the microcontroller

There are 4 inputs dedicated to the parallel data which can be either an address of a specific register to write to, or the data that needs to be written. Furthermore, it can be the upper or a lower nibble of a byte; two lines are used (a0 & a1) to signal which one it is according to Table 3. The address is decoded and the load is sent to the corresponding register.

2) Token and Handshake Packets Structure

The structure has several registers to save the data that will be transmitted; their addresses are stated on Table 4. The CTRL register provides important information about what will be sent out and how. Bit 6, if set, means a token packet (32 bits) will be sent, otherwise only a handshake (16 bits).

The calculated CRC and the SOF are loaded from the CRC calculator and the SOF generator respectively. While the bit 5 from the CTRL register dictates whether the calculated CRC or the stored CRC will be used, the bit 7 activates the host’s functions, i.e., SOF generation.

Table 3: Interface Data Types

a1	a0	Data Type
0	0	Lower nibble of Data
0	1	Upper nibble of Data
1	0	Lower nibble of Address
1	1	Upper nibble of Address

Table 4: Token and Handshake Registers

Address	Register
0x00	SYN
0x01	PID
0x02	ADDR
0x03	ENDP
0x04	CRC stored
0x05	SOF PID
---	SOF
---	CRC calculated
0x0F	CTRL

The SOF generator increments the SOF number and gives a pulse to begin the transmission at the appropriate time. The register structure is designed to be able to change, from sending any token packet to send a SOF packet, easily.

In order to maintain control between sending a normal token, handshake or a data packet and sending the SOF packets two steps are taken. First, in the clock block issues a warning signal to announce the proximity of the new SOF; if the warning signal is high, no transmission should take place. The second step consists of a state machine to avoid conflicts between sending SOF and the two run signals. The highest priority is given to the SOF packets; if a conflict arises and both Drun and THrun signals are activated simultaneously, the sending of token and handshake packets will have priority.

Every block is controlled by the Control State Machine for Token and Handshake Packets. When the run signal is activated, it enables the output and pushes the data out, validating every bit with a data valid signal, and receiving feedback from the bit stuffing block in case a zero bit needs to be inserted. It takes into account if a token or a handshake packet will be sent, by keeping track of the number of bits. Once the PID is sent it enables the CRC calculator and disables

Table 5: Data Registers

Address	Register
0x10	SYN
0x11	PID
0x12	CRC stored first byte
0x13	CRC stored second byte
0x14-0x33	Data Bytes
--	CRC 16 calculated
0xF0	NUM

it after the ENDP field. At the end of the transmission it provides the EOP for the needed 2 bit times.

3) Data Packets Structure

The block for data packets has the same logic behind it as the block for token and handshake packets. It begins with loading the data registers and ends with the control state machine. The addresses for the block of registers for the Data are stated in Table 5. The NUM register tells how many bytes of data will be transmitted.

The calculated CRC 16 is loaded directly from the CRC 16 calculator. Whether the CRC calculated or stored is used, is also stated by the CTRL register in the Token and Handshake registers structure. All the blocks in the Data structure are controlled by the control state machine inside the data structure. Its performance is almost the same as the one for token and handshake packets; it keeps count of how many bytes are sent, enables and clears the CRC16 calculator and provides the time for the EOP.

4) Selection and Formatting of Output

This block is in charge of selecting between the token and handshake packet serial output and the data serial output to perform the bit stuffing, NRZI encoding and the differential output. The selection is done via some multiplexers controlled by the Drun and THrun signals, during every transmission only one of the two lines can be activated.

The bit stuffing sends a next signal to the control state machines to ask for the next bit of the string, when six one bits are encountered, the signal *nxt* is stalled and a zero bit is inserted and endorsed with the valid bit in the output section. After the bit stuffing is done, the NRZI encoding and the differentiation of the output take place.

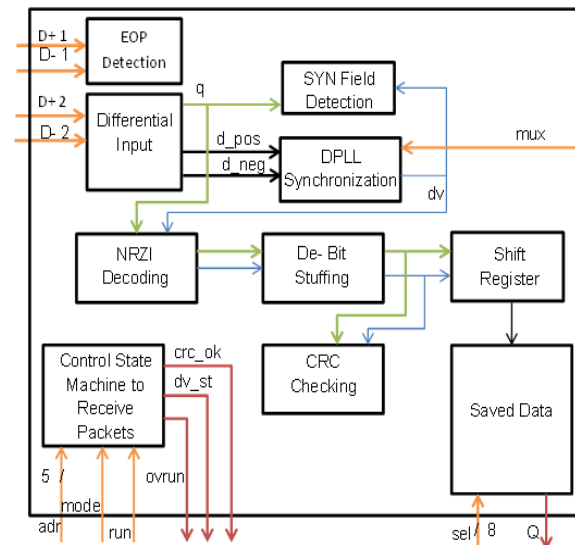


Figure 4: Receiver

C. Receiver

The receiver is in itself divided into several different structures: the differential input block, the blocks for EOP and SYN detection, the block for synchronization, the blocks for NRZI decoding, de-bit stuffing and CRC checking, the block with the shift register and the saved received data, and finally the control finite state machine. The whole receiver structure is shown in Figure 4.

1) EOP Detection and Differential Input Blocks

As inputs of the structure there are 2 D+ lines and 2 D-, one set of them go to the EOP detection block to be able to recognise the SE0 part of the EOP. The other set of lines are inputs of the differential input block, where a primitive from the Xilinx Unified Libraries for the Spartan 3E board was used, the IBUFDS differential signaling input buffer with optional delay, to retrieve the data from the differential lines. Another function of the differential input block is to do two samplings of the data line; first with the positive edge of the clock, the second with the negative.

2) DPLL for Synchronization and SYN Detection Blocks

A digital phase locked loop is implemented through a state machine to keep track of the changes in the data line in the positive and the negative edges of the internal clock. Then some signals for data available *dv* are created to indicate a reliable place of the bit cell to be read, the best *dv* signal can be manually selected by the input mux of the block. The DPLL needs at least 2 changes in the lines to lock the internal clock with the signal.

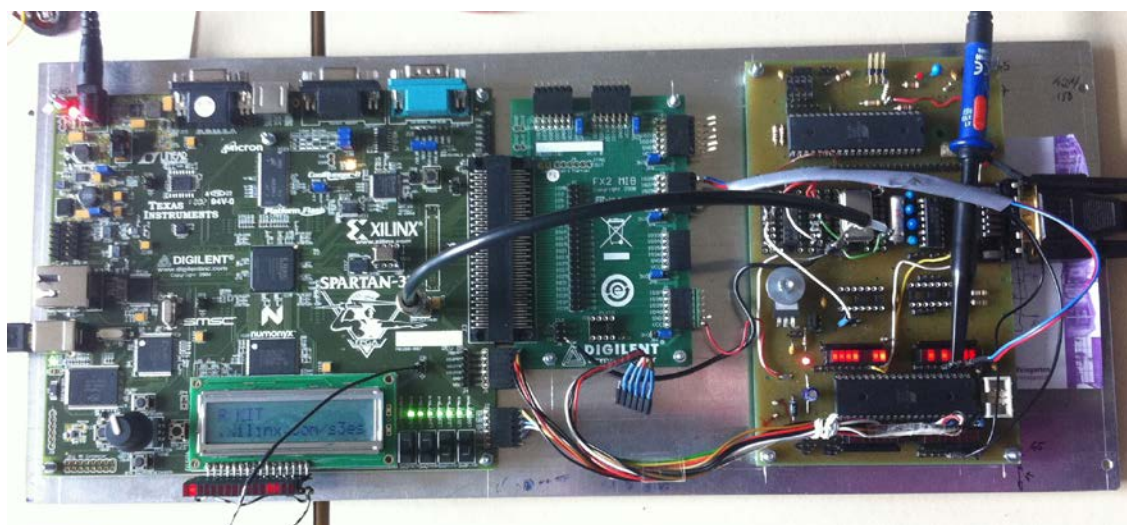


Figure 5: Set-up for Implementation

The dv signal helps the SYN detection block to read the incoming bits. Due to the start-up delay in the DPLL the SYN Detection block only takes into account the last 6 bits of the SYN pattern, that is, if the string “010100” is detected, a SYNC pattern will be considered positive. The SYN detection block also searches for long strings of ones to detect an idle state.

3) NRZI Decoding and De-Bit Stuffing Block

The NRZI decoding block must read a bit when the dv signal is high to guarantee the correct value has been stored. Much in the same way, it must deliver a new data valid signal to compensate the delay along with the decoded data. The de-bit stuffing block is designed as a finite state machine with the purpose to extract the 0 bit whenever 6 consecutive ones are found. This task is done only by not activating the data valid signal output whenever the zero should be extracted.

4) CRC Verification Block

The CRC verification block is activated after the PID field has been read and it is disabled when the EOP is detected. There are 2 internal verification blocks, one to check the CRC-5 and the second for CRC-16, and they both work with the exact logic as the CRC calculators. They calculate the CRC from the data being received and compares it to the residual, mentioned in table 2, to check if there were any errors in the process.

5) Shift Register and Registers for Saved Data

The shift register is a simple 16 bit register to load the incoming data before transferring it to the registers for saved data. It takes the data coming from the de-bit stuffing block when the valid signal is high and saves it. The 16 bits of the shift register are loaded into a word of the registers for saved data structure

when the control state machine gives the write signal in the address provided.

If the microcontroller wishes to read the information stored, it uses the input sel of the receiving block; the four msb chooses the word that will appear in the parallel output $Q<16:1>$; the four lsb runs each of the bits into the serial output $Q<0>$. After the receiving process is over, the control state machine writes the number of words received into the last register, that way when the data wants to be read out, the first step is to read address 1111B for the number of bytes to be read out.

6) Control State Machine for Reception

The Control State Machine takes care of every control line in whole receiving structure. The Receptor Structure offers the possibility to start saving in a specific register, this is indicated by the adr and mode inputs. If mode is one the control state machine loads the address from adr to start saving, otherwise it will start in address zero. If run is activated and either an EOP or a line of ones (idle state) is detected the waiting to detect a valid SYN pattern begins. Once this happens the enable signal for the de-bit stuffing is activated and the process of reception starts.

When the EOP is detected, the control state machine saves the last data into a register along with the number of words received and the CRC Verification is checked. If the data received was only the PID plus 16 bits the CRC5 is taken into account, if there were more bytes received then the CRC16 is checked.

The outputs of the block are flags; if there is new data stored the dv_st flag is set, if the correct CRC was a match the CRC_ok flag is set. The overrun flag is set when the last possible register is written to and there is no more place to save data. The flags are maintained until the run signal is turned off; once the run signal is off the flags are reset and the CRCs cleared.

IV. IMPLEMENTATION

The implementation on the Spartan 3E board was done by testing the receiver structure at one time and the transmitter with the clock blocks on another to simplify it. The set up used can be seen on Figure 5, the middle board is only used for easier access to the pins while the board on the right correspond to the microcontroller used.

Some minor changes were made in the code, particularly for the receptor, in the amount of inputs and outputs of the structures due to a limited number available on the board. Some multiplexers were implemented to have a bigger number of outputs available and some intermediate signals were routed out in order to have a better understanding of the performance of the code. The *.ucf files were done for the allocation of the pins accordingly.

V. CONCLUSION

The desired VHDL code was designed and simulated; it was also synthesized and implemented on a Spartan 3E Board. The system was tested and was able to receive and transmit USB packets. It has also both, the capability to send SOF packets to play host if requested as well as calculate or manipulate the CRCs for almost every packet. The received information was saved and able to be read out when the user decides to. After further testing the manufacturing of an ASIC should be possible.

ACKNOWLEDGEMENT

The author would like to thank Prof. Dr.-Ing. Walter Ludescher for his advice and guidance during the development of this project, as well as for the microcontroller program used for testing.

REFERENCES

- [1] *I. Xilinx, Spartan-3 Libraries Guide for HDL*, 2008.
- [2] *H. P. I. L. M. Compaq, Universal Serial Bus Specification Revision 2.0*, 2000.



María Guadalupe Salgado Suárez obtained the academic degree B. Sc. in mechatronics engineering at the Monterrey Institute of Technology and Higher Education in Mexico in December 2009. She then continued her master studies in electrical engineering at the University of Applied Sciences Ravensburg-Weingarten, having received the M. Eng. degree in March 2013. This publication is part of her Master Thesis.



Erkennung von Fußgängern in Echtzeit auf FPGAs

Michael Hahnle, Frerk Saxen, Konrad Doll

Zusammenfassung—Die Detektion von Fußgängern unter Echtzeitbedingungen ist eine der Kernproblemstellungen bei der Entwicklung von kamerabasierten Fahrerassistenzsystemen. Häufig wird hierfür der von Dalal und Triggs [1] entwickelte Histogramms of Oriented Gradients (HOG) Deskriptor in Verbindung mit einer Support Vector Machine (SVM) für die Klassifikation verwendet, die auch die Basis dieser Arbeit bilden. Es wird ein neuer, ressourcen- und geschwindigkeitsoptimierter Ansatz vorgestellt, bei dem sowohl die Deskriptorenberechnung als auch die Klassifikation sowie die Skalierung auf einem FPGA implementiert sind. Dies ermöglicht, zusammen mit einem Zeitmultiplexverfahren für die Multiskalenberechnung, eine Vollbildklassifikation bei 60 Full-HD-Bildern (1920 x 1080 Pixel) pro Sekunde mit einer Latenz von weniger als 150 μ s. Damit können in einer Skalenstufe pro Sekunde 1,8 Mio. HOG-Deskriptoren sowie die dazugehörigen Klassifikationsergebnisse mit der SVM berechnet werden. Auf einem Xilinx Virtex-5 FPGA sind so im Multiskalenbetrieb bis zu 11 Mio. Klassifikationsfenster pro Sekunde auswertbar, was bestehende FPGA-Implementierungen um den Faktor 10 übersteigt.

Schlüsselwörter—Fußgängererkennung, FPGA, HOG, SVM.

I. EINLEITUNG

A. Motivation

Die automatisierte Detektion von Personen hat in vielen Bereichen, wie z. B. in Überwachungssystemen oder in Fahrerassistenzsystemen, eine große Bedeutung. Während bei manchen Anwendungsfällen die Geschwindigkeit eine untergeordnete Rolle spielt, ist vor allem auch bei modernen Fahrerassistenzsystemen die Echtzeitfähigkeit und Latenz der Detektion entscheidend. CPUs – auch moderne Mehrkern-CPUs – sind hierfür aufgrund ihrer sequentiellen Arbeitsweise nur eingeschränkt geeignet. Durch Parallelisierung der

Berechnungen auf modernen Graphics Processing Units (GPUs) sind die geforderten Echtzeitbedingungen erreichbar. In zunehmendem Maße werden seit einigen Jahren auch Field Programmable Gate Arrays (FPGAs) für die digitale Bildverarbeitung eingesetzt. Sie bieten im Vergleich zu GPUs den Vorteil einer – je nach Algorithmus – deutlichen Geschwindigkeitssteigerung bei gleichzeitig geringerem Strom- und Platzbedarf, da moderne FPGAs für die digitale Signalverarbeitung (Digital Signal Processing – DSP) spezielle Makrozellen enthalten, die große Vorteile bezüglich der Geschwindigkeit und des Ressourcenbedarfs des späteren Designs bieten.

B. Stand der Technik

Der von Dalal und Triggs 2005 vorgestellte HOG-Deskriptor in Verbindung mit einer SVM zur Klassifikation [1], [3] ist im Bereich der Personenerkennung weit verbreitet. Während einige Veröffentlichungen auf die Optimierung der Erkennungsrate durch Änderung des Algorithmus (z. T. für bestimmte Situationen) abzielen (z. B. [8], [13]), steht in anderen die Optimierung der Geschwindigkeit – vor allem durch den Einsatz von GPUs und FPGAs – im Fokus (z. B. [4]-[7], [10]-[12], [14]).

Bislang gibt es im Vergleich zu CPU- und GPU-Implementierungen nur wenige Veröffentlichungen zu FPGA-basierten HOG-Fußgängerdetektoren. Grundlegend kann dabei zwischen reinen FPGA-Implementierungen (z. B. [6], [10]-[13]) und FPGA-beschleunigten Systemen (z. B. [4], [5]), bei denen nur ein Teil der Berechnung auf dem FPGA stattfindet, unterschieden werden.

Die schnellsten GPU- und FPGA-Implementierungen erreichen maximal 0,4 – 0,9 Mio. Klassifikationsfenster pro Sekunde [7], [11] mit Bildraten von 17 Bildern pro Sekunde (fps) bei einer Auflösung von 640 x 480 Pixel [13] bis 30 fps bei 1920 x 1080 Pixel [11].

C. Zielsetzung

Die im Folgenden beschriebene FPGA-Implementierung ist auf die Verarbeitung einer Auflösung von 1920 x 1080 Pixel (Full-HD) bei mindestens 60 Bildern pro Sekunde ausgelegt. Dies stellt eine hohe Anforderung an das Design der Schaltung, um eingehende Pixeldaten schritthaltend verarbeiten zu können. Damit bei dieser Auflösung eine so hohe Geschwindigkeit erreicht werden kann, kommt ein konsequentes Pipelining der Berechnungen und ein zellenbasierter

Michael Hahnle, michael.hahnle@h-ab.de, und Konrad Doll, konrad.doll@h-ab.de, sind Mitglieder der Hochschule Aschaffenburg, Fakultät Ingenieurwissenschaften, Labor für Rechnergestützten Schaltungsentwurf, Würzburger Straße 45, 63743 Aschaffenburg.

Frerik Saxen, frerk.saxen@ovgu.de, ist Mitglied der Otto-von-Guericke-Universität Magdeburg, Lehrstuhl für Technische Informatik, Universitätsplatz 2, 39106 Magdeburg.

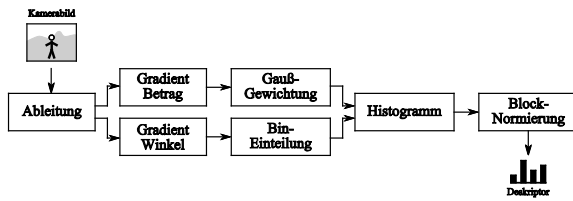


Abbildung 1: Schritte zur Berechnung des HOG-Deskriptors

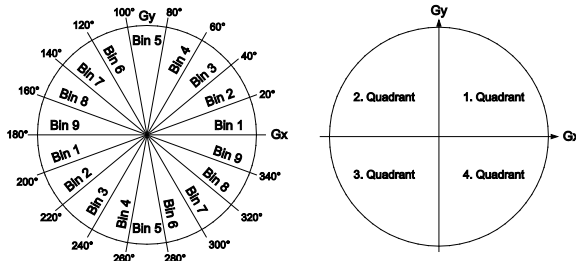


Abbildung 2: Einteilung des Gradientenwinkels in Bins und Quadranten

Ansatz zum Einsatz. Bei diesem wird nicht erst der komplette Deskriptor eines Fensters bestimmt, sondern der Deskriptor wird erst bei der Auswertung durch die SVM aus einzelnen Zellen zusammengesetzt.

Des Weiteren ist auch die Realisierung einer Multiskalenerkennung Gegenstand der Implementierung. Diese ist für eine Fußgängererkennung in der Praxis unerlässlich, da verschiedene Körpergrößen und Entfernungen zur Kamera abzudecken sind. Während die meisten der oben genannten FPGA-Implementierungen auf eine Skalenstufe beschränkt sind, soll hier eine Skalenanzahl von etwa 20 Stufen, wie sie bei CPU- und GPU-Implementierungen üblich ist, ermöglicht werden. Dies wiederum erfordert eine ressourcenschonende Umsetzung des Algorithmus, um die parallele Verarbeitung mehrerer Skalen zu ermöglichen, ohne dabei die oben genannten Echtzeitbedingungen zu verletzen. Einer der Kernpunkte für die Ressourcenschonung ist hier die Verwendung einer Taktverdoppelung, wodurch an vielen Stellen Ressourcen eingespart werden können.

Da jedoch auch große FPGAs nur über begrenzte Ressourcen verfügen, wird ein Zeitmultiplexverfahren vorgestellt, das es ermöglicht, mehr Skalenstufen zu berechnen als parallel mit den vorhandenen Ressourcen implementierbar sind.

D. Überblick

Im Kapitel II wird zunächst grundlegend die Theorie zum HOG- und SVM-Algorithmus erläutert. Anschließend wird im Kapitel III ein kurzer Überblick über die Implementierung und die darin enthaltenen Module gegeben, welche darauf folgend im Detail beschrieben werden. Kapitel IV stellt die Ergebnisse hinsichtlich Ressourcenbedarf, Echtzeitverhalten und

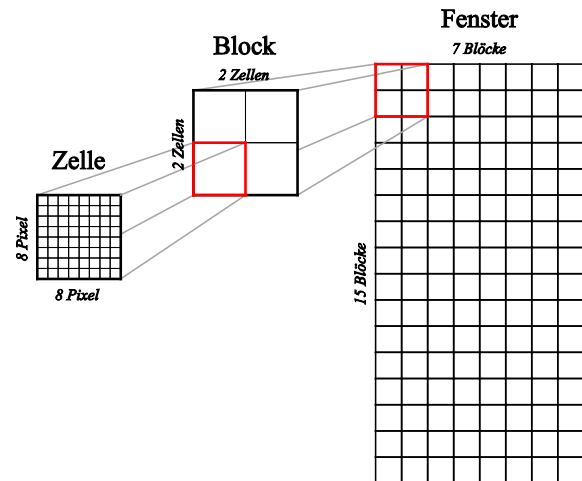


Abbildung 3: Einteilung des HOG-Klassifikationsfensters in Blöcke und Zellen

Erkennungsgenauigkeit dar. Abschließend wird die Arbeit im Kapitel V zusammengefasst.

II. ALGORITHMUS

Die bildbasierte Detektion von Personen erfolgt nach dem sogenannten Sliding Window-Prinzip, bei dem immer nur ein Ausschnitt des Bildes mit fester Größe (hier: 64 x 128 Pixel) betrachtet wird. Durch Verschieben dieses Fensters von links oben nach rechts unten wird nach und nach das gesamte Bild ausgewertet.

Die Erkennung setzt sich dabei aus drei Teilen zusammen: Zunächst wird für jedes Fenster ein Deskriptor berechnet. Dies erfolgt in diesem Fall mit Hilfe des HOG-Algorithmus. Anschließend wird der Deskriptor durch einen Klassifikator bewertet. Für diese Klassifikation wird eine SVM verwendet. Überlappende Erkennungen werden im letzten Schritt mit einer Non-Maxima Suppression (NMS) zusammengefasst. Da die NMS jedoch nicht Teil dieser FPGA-Implementierung ist, wird sie nachfolgend nicht näher erläutert.

A. Histograms of Oriented Gradients-Deskriptor

In Abbildung 1 sind die einzelnen Schritte, die für die Berechnung des HOG-Deskriptors notwendig sind, dargestellt. Zunächst wird das Eingangsbild I durch Faltung mit den diskreten Filtermasken M_x und M_y horizontal und vertikal abgeleitet:

$$G_x = M_x * I \quad \text{mit} \quad M_x = [-1 \ 0 \ 1]$$

$$G_y = M_y * I \quad \text{mit} \quad M_y = [-1 \ 0 \ 1]^T$$

Aus den abgeleiteten Bildern G_x und G_y wird anschließend der zweidimensionale Gradient für jedes Pixel (x, y) berechnet. Betrag $|G(x, y)|$ und Winkel $\phi(x, y)$ des Gradienten für die Pixelposition (x, y) ergeben sich durch:



$$|G(x, y)| = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

$$\tan(\phi(x, y)) = \frac{G_y(x, y)}{G_x(x, y)}$$

Jedes Klassifikationsfenster ist in Blöcke unterteilt (Abbildung 3). Jeder Block besteht dabei aus 4 Zellen und diese wiederum aus 8 x 8 Pixel. Die einzelnen Blöcke überlappen sich im Fenster um jeweils eine Zelle in vertikaler und horizontaler Richtung. Damit ergibt sich für ein 64 x 128 Pixel großes Fenster eine Anzahl von $7 \cdot 15 = 105$ Blöcken.

Für jeden der 105 Blöcke wird nun ein Histogramm erstellt. Dieses besteht aus den konkatenierten Histogrammen der vier Zellen des jeweiligen Blocks. Das Histogramm einer Zelle wird gebildet, indem die Gradientenwinkel in 9 Bins entsprechend Abbildung 2 eingeteilt werden. Jedes Zellhistogramm enthält also neun Elemente – eines für jedes Bin. Dies ergibt eine Länge von $4 \cdot 9 = 36$ Elementen pro Blockhistogramm.

Zur Verbesserung der Erkennungsrate werden die Gradientenbeträge der Pixel im Randbereich der Blöcke weniger stark gewichtet als solche im Zentrum. Dies geschieht durch elementweise Multiplikation des Blocks mit einem 16 x 16 Pixel großen Gaußfilter F_g mit $\sigma = 8$ [1]:

$$|G|_{Gau\beta} = F_g \circ |G|_{Block}$$

$$F_g = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Diese gewichteten Beträge werden dann je Bin zu den einzelnen Histogrammelementen aufsummiert. Dabei kann es zu Aliasing-Effekten kommen, wenn markante Bildinformationen mit signifikanter Auswirkung auf den Deskriptor nahe an Bin- oder Zellgrenzen liegen. Um diesem Effekt vorzubeugen, können die Gradientenbeträge zusätzlich trilinear zwischen den Pixel und den Bins interpoliert werden [3].

Um Helligkeits- und Kontrastschwankungen entgegenzuwirken, werden die Blockhistogramme \underline{v} anschließend normiert. Hierfür wird meist die L2-Norm verwendet:

$$\underline{v} \rightarrow \frac{\underline{v}}{\sqrt{\|\underline{v}\|_2^2 + \varepsilon^2}}$$

ε repräsentiert darin die kleinste mögliche Zahl und verhindert eine Division durch Null.

Die 105 Blockhistogramme werden konkateniert und ergeben dann den $105 \cdot 36 = 3780$ -dimensionalen HOG-Deskriptor.

B. Klassifizierung durch eine Support Vector Machine

Die SVM dient zur Klassifizierung der berechneten HOG-Deskriptoren. Typischerweise unterscheiden SVMs zwischen zwei Klassen – in diesem Fall "Fußgänger" und "kein Fußgänger".

Hierfür werden die Deskriptoren als Punkte im n-dimensionalen Raum betrachtet. Mit Hilfe einer großen Anzahl von positiven (Fußgänger) und negativen (kein Fußgänger) Trainingsdaten wird eine Hyperebene zur Trennung der beiden Klassen bestimmt. Bei der hier verwendeten linearen SVM kann diese Trennebene durch folgende Gleichung beschrieben werden:

$$y(\underline{x}) = \underline{w}^T \cdot \underline{x} + b$$

\underline{x} ist darin der zu klassifizierende Deskriptor. Die Hyperebene selbst wird durch den Gewichtsvektor \underline{w} , der die Orientierung der Ebene im Raum beschreibt, und den Bias b , für die Parallelverschiebung der Ebene, festgelegt. Während das Vorzeichen des Ergebnisses dieser Gleichung $y(\underline{x})$ die Klassenzugehörigkeit von \underline{x} angibt, liefert der Betrag die Entfernung zur Ebene und damit ein Maß für die Sicherheit, dass \underline{x} zur jeweiligen Klasse gehört.

In dem hier nicht näher beschriebenen Training der SVM wird, anhand eines Trainingsdatensatzes (Beispiele beider Klassen) \underline{w} und b so gewählt, dass der Abstand zwischen den begrenzenden Vektoren beider Klassen (Stützvektoren – engl. Support Vectors) und der Hyperebene maximiert wird:

$$\max\left(\frac{1}{\|\underline{w}\|}\right)$$

Bei der Optimierung werden Zuordnungsverletzungen durch Vektoren, die innerhalb der begrenzenden Vektoren und/oder auf der falschen Seite der Ebene liegen, zwar toleriert, die Summe dieser Fehler ist jedoch zu minimieren und geht daher mit in die Optimierungsgleichung ein. Diese Form der SVM wird als lineare Soft-Margin-SVM bezeichnet.

Die Klassifikation von schwer separierbaren Daten kann durch die Verwendung einer Kernel-SVM verbessert werden. Um die lineare Trennung zu erleichtern, hebt diese die Deskriptoren mit einer Kernel-Funktion in eine höhere Dimension. Die Kernel-SVM bedarf hierdurch eines hohen Rechenaufwands und ist – unter den hier gesetzten Echtzeitbedingungen – nicht ohne performanzrelevante Einschränkungen in Hardware implementierbar [9], weshalb für diese Implementierung die lineare SVM gewählt wurde.

III. IMPLEMENTIERUNG

Die hier vorgestellte Implementierung folgt einem zellenbasierten Ansatz. Dabei werden die Deskriptoren nicht sofort für die einzelnen Fenster erstellt, son-

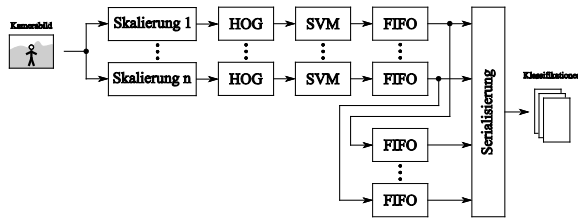


Abbildung 4: Systemüberblick mit mehreren Skalenstufen und zusätzlichen FIFOs für die Zeitmultiplex-Erweiterung

dem es werden zunächst nur die einzelnen Zell- bzw. Blockhistogramme gebildet. Erst im zweiten Schritt werden daraus die einzelnen Fensterdeskriptoren zusammengesetzt. Dies hat den Vorteil, dass Blockhistogramme, die aufgrund des Sliding Window-Prinzips in mehreren Fenstern auftreten, nur einmal berechnet werden müssen. Durch diese Vorgehensweise wird eine FPGA-Implementierung unter den hier vorausgesetzten Echtzeitbedingungen erst möglich.

Um die Flexibilität des Designs für verschiedene Konfigurationen und spätere Erweiterungen zu erhöhen, wurde das Design stark modularisiert aufgebaut. In diesem Kapitel werden die wesentlichen Hauptmodule in Bezug auf ihre Funktion und Implementierung im Detail erläutert.

A. Systemüberblick

In Abbildung 4 ist das komplette Design inklusive der Multiskalen- und Zeitmultiplexererweiterung vereinfacht schematisch dargestellt. Jede Skalenstufe besteht aus je einem Skalierungsmodul für die Bildskalierung, einem HOG-Modul für die Deskriptor-Berechnung und einem SVM-Modul für die Klassifikation. Die nachfolgenden FIFOs sind für die Multiskalen- und Zeitmultiplexererweiterung relevant (siehe Abschnitte III.E und III.F).

Die Berechnung der Zell- und Blockhistogramme erfolgt in dieser Implementierung nicht getrennt für jedes Klassifikationsfenster, sondern für das gesamte Bild. Die Zuordnung der Blockhistogramme zu den einzelnen Fensterdeskriptoren findet erst in der SVM bei der Bildung des Skalarproduktes statt.

B. Skalierung

Da das Fenster für die Deskriptorberechnung eine feste Größe von 64×128 Pixel besitzt, wird im Skalierungsmodul das Eingangsbild für die jeweilige Skalenstufe auf eine in Pixel definierte Auflösung herunterskaliert.

Die Implementierung berechnet die Ausgangspixel mittels bilinearer Interpolation aus dem Eingangsbild [2]. Dies erhöht, im Gegensatz zu einem einfacheren und ressourcenschonenderen Downsampling durch Überspringen jedes zweiten Pixels mit vorangegangener Gaußfilterung, die Flexibilität der Skalierung, da die Skalierungsfaktoren nicht auf Zweierpotenzen

beschränkt sind. Durch die freie Skalierungswahl wird die Erkennungsrate signifikant verbessert.

C. Deskriptorberechnung (HOG)

Der HOG-Deskriptor besitzt eine Reihe von Parametern, deren optimale Werte von Dalal und Triggs [1] ermittelt wurden. Wie auch bei den meisten anderen Implementierungen wurden jedoch auch hier nicht überall die Werte mit der besten Erkennungsrate gewählt, sondern an einigen Stellen Kompromisse zugunsten der Implementierbarkeit und Ressourcenschonung eingegangen. Dies betrifft vor allem die Wahl der Aufteilung des Klassifikationsfensters in Zellen und Blöcke (Abbildung 3). Nach [1] ergibt sich die beste Erkennungsrate bei einer Zellgröße von 6×6 Pixel und 3×3 Zellen pro Block. Hierdurch würde jedoch der Deskriptor sehr lang werden, was die echtzeitfähige Implementierung erheblich erschweren würde. Stattdessen wurde die meist verwendete Einteilung in 2×2 Zellen und 8×8 Pixel gewählt, die nur eine geringfügig schlechtere Erkennungsrate bietet und einen Deskriptor mit nur etwa ein Drittel der Elemente liefert. In Abbildung 5 ist die Struktur der Implementierung dargestellt, wobei die einzelnen Schritte zur Berechnung des HOG-Deskriptors entsprechend Abbildung 1 beschriftet sind. Im Folgenden werden die einzelnen Abschnitte näher erläutert.

1) Ableitung

Die Ableitung setzt sich aus einem Zeilenspeicher und den beiden Subtraktionen für die parallele Berechnung der horizontalen und vertikalen Ableitungen zusammen.

Anschließend werden die Daten im Clock Domain Crossing an den Core-Takt übergeben, dessen Frequenz dem doppelten Pixeltakt entspricht. Diese Frequenzerhöhung dient in den weiteren Abschnitten zur Ressourceneinsparung, da so zwei Operationen nacheinander mit einem Operator durchgeführt werden können.

2) Gradientenberechnung und Bin-Zuordnung

Für die Berechnung des Gradientenbetrags werden die beiden Ableitungen zunächst quadriert und aufsummiert. Anschließend wird aus der Summe die Wurzel gezogen. Hierbei lassen sich durch die Taktverdoppelung nur minimal Ressourcen einsparen – lediglich ein Multiplizierer (DSP-Zellen) kann eingespart werden, indem die beiden Ableitungen nacheinander quadriert werden.

Um auf die ressourcenintensive Division bei der Bestimmung von $\tan(\phi(x, y))$ verzichten zu können, wird der Winkel nicht direkt berechnet, sondern mit der Winkelteilung kombiniert [4]. Die Zuordnung des aktuellen Gradienten zu einem Bin kann durch Multiplikation mit Konstanten und Vergleichen realisiert werden. Um beispielsweise die Zugehörigkeit zu

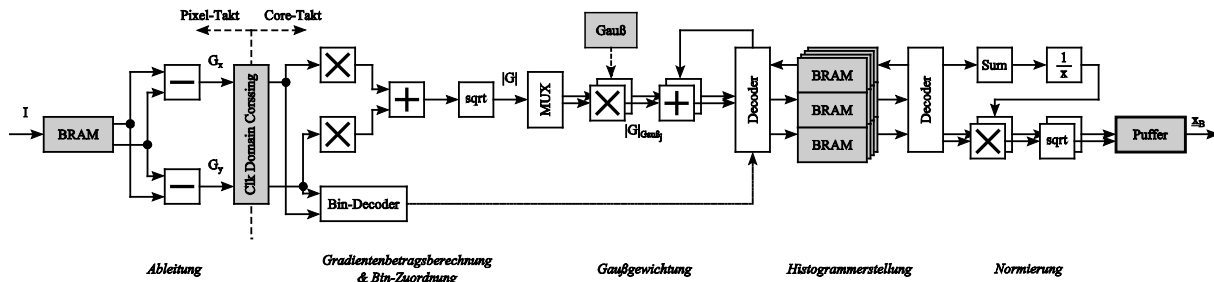


Abbildung 5: Aufbau der HOG-Implementierung

Bin 1 ($0^\circ - 20^\circ$) zu überprüfen, lautet die Ungleichung:

$$\tan(0^\circ) < \frac{|G_y|}{|G_x|} < \tan(20^\circ)$$

$$\tan(0^\circ) < |G_y| < \tan(20^\circ) \cdot |G_x|$$

$$0 < |G_y| < 0.364 \cdot |G_x|$$

Ist diese Ungleichung erfüllt, gehört der Gradient zu Bin 1. Um auf Gleitkommazahlen-Berechnung bei der Hardwareimplementierung verzichten zu können, werden die Konstanten durch Brüche angenähert:

$$\tan(20^\circ) = 0.364 \approx \frac{4}{11}$$

Die Ungleichung kann nun so umgestellt werden, dass nur noch mit Ganzzahlen gerechnet wird:

$$0 < \frac{|G_y|}{|G_x|} < \frac{4}{11}$$

$$0 < |G_y| \cdot 11 < |G_x| \cdot 4$$

Für jedes Bin wird unter Berücksichtigung der Quadranten eine entsprechende Ungleichung berechnet, um die Zugehörigkeit zu überprüfen. Für die Multiplikationen mit den verschiedenen Konstanten stehen durch den höheren Core-Takt zwei Taktzyklen zur Verfügung, was zur Halbierung der benötigten Anzahl an DSP-Zellen genutzt wird.

3) Gaußgewichtung

Die Gewichtung der Pixel innerhalb eines Blocks wird durch Multiplikation mit vorher berechneten Werten des Gaußfilters realisiert, welche in einem ROM als Lookup Table vorliegen. Da jede Zelle aufgrund der Überlappung zu 4 Blöcken gehört und darin jeweils eine andere Position besitzt, wird jede Zelle mit jedem der vier 8×8 Pixel großen Abschnitte $F_{g1} - F_{g4}$ des Gaußfilters F_g elementweise multipliziert:

$$F_g = \begin{bmatrix} F_{g1} & F_{g2} \\ F_{g3} & F_{g4} \end{bmatrix}$$

$$\forall j \in \{1, \dots, 4\}: |G|_{Gauß_j} = F_{g_j} \circ |G|_{Zelle}$$

Auch hier können durch die Taktverdoppelung die 4 Multiplikationen teilweise sequentiell durchgeführt werden und so die Anzahl der notwendigen DSP-Zellen von 4 auf 2 halbiert werden.

4) Histogrammerstellung

Für die Erstellung der Blockhistogramme werden die nun gewichteten Gradientenbeträge binweise für die einzelnen Zellen aufsummiert und die einzelnen Zellhistogramme anschließend zu den Blockhistogrammen konkateniert.

Die Bildung der Zellhistogramme erfolgt gleichzeitig über die gesamte Bildbreite. Hierfür befinden sich in einem BlockRAM-Modul 3 Zeilen von Zellhistogrammen – 2 für die Ausgabe der Blöcke und eine dritte als aktuelle Eingangszeile, in der die aktuellen Eingangsdaten aufsummiert werden. Da jede Zelle zu vier Blöcken gehört und sich die Werte durch die Gaußgewichtung jeweils unterscheiden, werden für jede Zelle vier Histogramme erstellt, was erforderlich macht, dass das Speichermodul entsprechend viermal vorhanden ist.

Der Eingangsdecoder (in Abbildung 5 links neben dem BRAM-Modul) ist bedingt durch die hohe Core-Frequenz als Pipeline konzipiert. Dies bedeutet, dass die Berechnung der Schreib- und Leseadresse sowie die Auswahlmultiplexer in mehreren Taktzyklen erfolgen müssen. Hierbei ist darauf zu achten, dass durch die Latenz der Pipeline keine noch nicht aktualisierte Speicherzelle erneut ausgelesen und der alte Wert der Addition zugeführt wird.

Der Ausgangsdecoder (in Abbildung 5 rechts neben dem BRAM-Modul) liest nacheinander für jeden Block die 4 zugehörigen Zellhistogramme aus und gibt diese an die Normierung weiter. Darüber hinaus übernimmt er auch die Löschung der Speicherzellen im BRAM nach dem Auslesen der Zellhistogramme.

5) Normierung

Bei der Normierung wird im Gegensatz zur ursprünglichen Implementierung von Dalal und Triggs

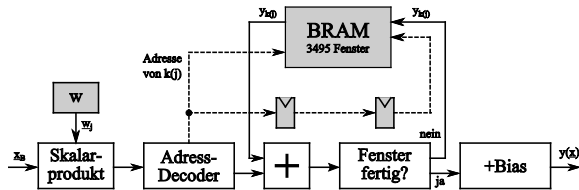


Abbildung 6: Aufbau der SVM-Implementierung mit Fensterangabe für ein Full-HD-Bild (1920 x 1080 Pixel)

nicht die L2-Norm, sondern die L1-Sqrt-Norm verwendet:

$$\underline{v} \rightarrow \sqrt{\frac{\underline{v}}{\|\underline{v}\|_1 + \varepsilon}}$$

Diese bietet bei der Implementierung den Vorteil, dass auf das Quadrieren der Histogrammelemente verzichtet werden kann, ohne dass die Erkennungsrate hierdurch wesentlich beeinflusst wird (siehe Abbildung 4 in [1]).

Durch die Taktverdoppelung stehen für die nachfolgende Histogrammerstellung 128 statt 64 Takte bis zur Fertigstellung der nächsten Zelle (und damit des nächsten Blocks) zur Verfügung. Dies ermöglicht erst eine sequentielle Berechnung der einzelnen Schritte, was in einer erheblichen Ressourceneinsparung resultiert.

Zur Erstellung und anschließenden Normierung der Blockhistogramme wird zunächst die Blocksumme berechnet, d. h. die Summe aller Elemente der vier Zellhistogramme $\|\underline{v}\|_1$ plus ε (entspricht bei der Hardwareimplementierung dem niederwertigsten Bit auf logisch 1).

Um Ressourcen zu sparen, wird anschließend statt der elementweisen Division einmal der Kehrwert von $(\|\underline{v}\|_1 + \varepsilon)$ gebildet und elementweise mit diesem multipliziert:

$$\underline{v} \cdot \left(\frac{1}{\|\underline{v}\|_1 + \varepsilon} \right)$$

Dies hat nicht nur den Vorteil, dass nur ein ressourcenintensiver Dividierer notwendig ist, sondern dieser muss darüber hinaus auch nicht als vollständige Pipeline realisiert werden. Er kann so konzipiert werden, dass nur in jedem achten Takt ein Eingangswert verarbeitet wird, was den Ressourcenbedarf des einen Dividierers auf ca. ein Viertel reduziert.

Um schritthaltend arbeiten zu können, müssen immer zwei Blockhistogrammelemente parallel mit dem Wert von $1/(\|\underline{v}\|_1 + \varepsilon)$ multipliziert werden. Anschließend wird die Wurzel gezogen und die normierten Elemente werden in einem Ausgangspuffer abgelegt, bis das Blockhistogramm vollständig normiert wurde.

D. Klassifikation (SVM)

Die SVM wertet zur Klassifikation eines Fensters die Gleichung

$$y(\underline{x}) = \underline{w}^T \cdot \underline{x} + b$$

aus. Diese Gleichung lässt sich auch in Abhängigkeit der 105 zu einem Fenster gehörenden Blockhistogramme \underline{x}_{Bi} darstellen:

$$y(\underline{x}) = \underline{w}^T \cdot \underline{x} + b = \sum_{i=1}^{105} (\underline{w}_i^T \cdot \underline{x}_{Bi}) + b$$

mit

$$\underline{x}^T = [\underline{x}_{B1}^T \quad \underline{x}_{B2}^T \quad \dots \quad \underline{x}_{B105}^T]$$

$$\underline{w}^T = [\underline{w}_1^T \quad \underline{w}_2^T \quad \dots \quad \underline{w}_{105}^T]$$

Berücksichtigt man, dass diese Gleichung für alle Fenster ausgewertet werden muss, erkennt man, dass jeder Block mit seinem Blockhistogramm \underline{x}_B aufgrund der Überlappung der Fenster (außer in den Randbereichen des Bildes) zu 105 Fenstern gehört. In jedem Fenster und dem dazugehörigen Deskriptor hat das Blockhistogramm \underline{x}_B eine andere Position ($\underline{x}_{B1} - \underline{x}_{B105}$). Die Position innerhalb eines Fensters wird mit j bezeichnet. Der zur Position j gehörige Teil des Gewichtsvektors heiße \underline{w}_j . Insgesamt muss für jedes Blockhistogramm \underline{x}_B ein Teilskalarprodukt mit jedem der 105 Abschnitte des Gewichtsvektors $\underline{w}_1 - \underline{w}_{105}$ gebildet werden. \underline{w} liegt für die Berechnungen auf dem FPGA in einem ROM vor. Die Teilskalarprodukte $\underline{w}_j^T \cdot \underline{x}_B$ werden anschließend zu den aktuellen Zwischensummen $y_{k(j)}$ der zugehörigen Fenster $k(j)$ hinzuaddiert:

$$\forall j \in \{1, \dots, 105\}: y_{k(j)} \leftarrow y_{k(j)} + \underline{w}_j^T \cdot \underline{x}_B$$

Der Vorteil dieser Vorgehensweise besteht darin, dass zur Berechnung von $y(\underline{x})$ nicht Blockhistogramme mit jeweils 36 Elementen, sondern nur noch Zwischensummen gespeichert werden müssen.

Die Zwischensummen liegen dabei in einem BlockRAM-Modul. Da die Berechnung des Skalarproduktes $\underline{w}^T \cdot \underline{x}$ erst mit der Berechnung des letzten Blockhistogramms \underline{x}_{B105} abgeschlossen ist, müssen immer die Zwischensummen aller vertikal überlappenden Fenster über die gesamte Bildbreite im RAM gespeichert werden. Daher hängt die Anzahl der zu speichernden Zwischensummen von der Bildbreite und damit von der jeweiligen Skalenstufe ab. Sie ergibt sich als Produkt der Fenster pro skaliertes Bildbreite und der vertikalen Anzahl an Blöcken in einem Fenster. Bei Full-HD-Auflösung wären dies

Tabelle 1: Ressourcenbedarf der einzelnen Module

	Skalierung	HOG	SVM
Register	1.861	3.642	1.534
LUTs	1.177	3.924	1.264
DSP-Zellen	8	12	37
18k BRAMs	2	52	14

Tabelle 2: Ressourcenbedarf und -auslastung für ein Xilinx Virtex®-5 FPGA (XC5VFX200T) bei 6 parallelen Skalinstufen und 3 Skalensätzen im Zeitmultiplex

	Design	Verfügbar	Prozent
Register	42.987	122.880	34 %
LUTs	38.535	122.880	31 %
DSP-Zellen	357	384	92 %
18k BRAMs	396	912	43 %

Tabelle 3: Vergleich des Ressourcenbedarfs verschiedener FPGA-Implementierungen für eine Skalinstufe (entspricht Ressourcen von HOG + SVM aus Tabelle 1)

	Plattform	Auflösung	fps	LUTs	Register	DSPs	RAM (kBit)	MHz
[6]	Virtex-5	640 x 480	62	17.383	2.181	k.A.	1.327	44
[11]	Cyclone IV	1920 x 1080	30	34.403	23.247	68	645	76
[12]	Stratix II	640 x 480	30	37.940	66.990	120	k.A.	127
Unser	Virtex-5	1920 x 1080	64	5.188	5.176	49	1.188	270

$233 \cdot 15 = 3495$ Zwischensummen. Der Adress-Decoder bestimmt, basierend auf der aktuellen Position j im Gewichtsvektor \underline{w} und der Position des Blockes \underline{x}_B im Bild, die Speicheradresse des zugehörigen Fensters $k(j)$.

Nach dem Aufaddieren wird die neue Zwischensumme $y_{k(j)}$ wieder zurück in den RAM geschrieben oder – falls das Skalarprodukt für dieses Fenster vollständig ist (nach der Berechnung von $\underline{w}_{105}^T \cdot \underline{x}_{B105}$) – weitergereicht und mit dem Bias-Wert b zu $y(\underline{x})$ addiert. Nun kann durch Abfragen des Vorzeichen-Bits die Klassenzugehörigkeit des jeweiligen Klassifikationsfensters bestimmt werden.

Am Ausgang der SVM werden für jede positive Erkennung (Fußgänger) die Position in Pixel im unskalierten Eingangsbild, Breite und Höhe des skalierten Fensters sowie $y(\underline{x})$ ausgegeben.

E. Multiskalenansatz

Für die Multiskalenberechnung werden mehrere Verarbeitungszweige parallel instanziiert, wobei jedes Skalierungsmodul für eine andere Skalinstufe konfiguriert ist. Die positiven Detektionen an den Ausgängen der SVMs müssen für die weitere Verarbeitung oder Übertragung serialisiert werden. Dies übernimmt ein Fusionsmodul (siehe Abbildung 4). Es besteht aus je einem FIFO-Puffer nach jeder SVM, aus denen abwechselnd – um einen Overflow zu vermeiden – die Erkennungen entnommen und durch das Serialisierungsmodul in das gewünschte Ausgangsformat – zum Beispiel einen Byte-Datenstrom – konvertiert werden.

Um eine möglichst hohe Flexibilität des Designs zu gewährleisten, ist das Multiskalen-Modul parametrisierbar gestaltet. So können je nach Anwendungsfall oder vorhandenen Ressourcen unterschiedlich viele Skalinstufen berechnet werden.

trierbar gestaltet. So können je nach Anwendungsfall oder vorhandenen Ressourcen unterschiedlich viele Skalinstufen berechnet werden.

F. Zeitmultiplexansatz

Um wie in der Praxis üblich etwa 20 Skalinstufen für ein Bild zu erhalten, müssten entweder entsprechend viele Verarbeitungszweige instanziiert werden oder das Bild mehrfach durch die Pipeline laufen, was zum einen die Zwischenspeicherung des kompletten Bildes erforderlich machen würde und zum anderen mit den oben genannten Echtzeitanforderungen kollidieren würde. Da hier jedoch von hohen Bildraten ausgegangen wird (50 – 60 fps), kann auf die Speicherung verzichtet werden. Bei diesen Bildraten liegen zwischen zwei Bildern weniger als 20 ms. Selbst bei zügiger Gehweise bewegt sich ein Fußgänger von einem zum nächsten Bild nur einige Pixel weit. Basierend auf dieser Tatsache, wird in dieser Implementierung jedes Bild mit einem anderen Skalensatz berechnet (klassifiziert). Dabei können die Erkennungsergebnisse der letzten Bilder (mit den jeweils anderen Skalensätzen) mit den neuen Klassifikationen überlagert werden.

Das eingangs erwähnte Zeitmultiplexverfahren realisiert dies durch eine Umkonfigurierung der Skalierungsmodule nach jedem Bild. Auf diese Weise können mehrere Skalinstufen nacheinander mit derselben Pipeline berechnet werden. Um für jedes Bild alle Skalinstufen ausgeben zu können, werden die Ergebnisse der letzten Skalen durch eine FIFO-Kaskade geleitet, deren Länge von der Anzahl der Skalensätze abhängt. Die Ausgangsdaten der einzelnen Kaskadenstufen werden für die Ausgabe ebenfalls an das Serialisierungsmodul übergeben.

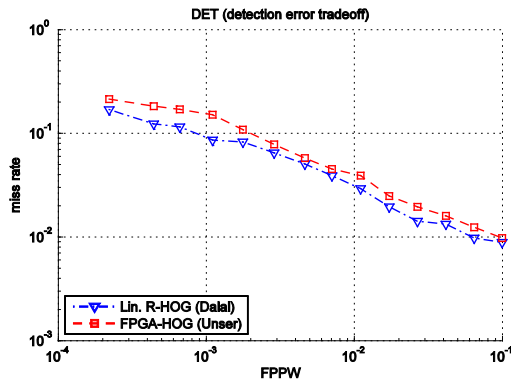


Abbildung 7: Vergleich der Erkennungsrate auf Basis der INRIA-Datenbank [15]

IV. ERGEBNISSE

Dieses Kapitel stellt die Ergebnisse dar, die mit der oben beschriebenen Implementierung erzielt wurden. Für die Tests wurde eine Kamera verwendet, die Bilder mit Full-HD-Auflösung bei 50 fps und 8 Bit Graustufenwerten liefert.

A. Ressourcen

In Tabelle 1 sind die Ressourcen für die einzelnen Module aufgelistet, während Tabelle 2 die Auslastung des zur Evaluierung verwendeten Xilinx Virtex[®]-5 FPGAs (XC5VFX200T) für das komplette Design mit 6 parallelen Skalinstufen und 3 Skalensätzen im Zeitmultiplex zeigt.

Wie erkennbar ist, sind hierbei die DSP-Zellen das begrenzende Element, weswegen bei der Implementierung der Module insbesondere auf deren Einsatz geachtet wurde. Durch die Taktverdoppelung konnte die Anzahl der DSP-Zellen für den HOG erheblich reduziert werden. Vor allem bei der Normierung können damit, wie oben beschrieben, sehr viele Ressourcen eingespart werden, da durch die Taktverdoppelung 128 statt 64 Takte bis zum nächsten Block zur Verfügung stehen, und so ein großer Teil der Normierung sequentiell berechnet werden kann.

Im Vergleich zu der Implementierung aus [11] die auch auf eine Auflösung von 1920 x 1080 Pixel abzielt, konnte der Ressourcenbedarf – trotz doppeltem Durchsatz (30 fps zu 60 fps) – deutlich reduziert werden (Tabelle 3).

B. Echtzeitverhalten

Die Implementierung wurde auf den Betrieb mit einem Pixel-Takt von 133 MHz und einem Core-Takt von 266 MHz optimiert. Die Maximalfrequenz liegt bei etwa 270 MHz (Core-Takt). Daraus ergibt sich eine maximale Bildrate von 64 fps bei 1920 x 1080 Pixel (Tabelle 3). Die maximale Klassifikationsperformance liegt damit bei

$$\left(\frac{1920}{8} - 6\right) \cdot \left(\frac{1080}{8} - 14\right) \cdot 64 = 1.812.096$$

Klassifikationsfenstern pro Sekunde für eine Skalinstufe (bei einem Skalierungsfaktor von 1). Das Design aus Tabelle 2 ist somit in der Lage, eine maximale Anzahl von knapp 11 Mio. Klassifikationsfenstern pro Sekunde (ohne Zeitmultiplex) zu verarbeiten. Dies übersteigt bestehende Implementierungen [5], [6], [7], [11] um mehr als den Faktor 10.

Die Latenz der Implementierung variiert durch die Skalierungsmodule und die diversen Zeilenpeicher je nach gewählter Skalinstufe. Durch eine Hardwaresimulation wurde eine Latenz von unter 150 µs ermittelt.

C. Genauigkeit

Um die Erkennungsrate zu erhöhen, werden im gesamten Design Werte bei Overflow nicht abgeschnitten sondern auf den Maximalwert abgerundet, um den daraus resultierenden Fehler zu minimieren. Des Weiteren wurden die Bitbreiten der einzelnen Signale so gewählt, dass es nur unter Worst-Case-Bedingungen zu einem Overflow kommen kann.

Für die Bewertung der Erkennungsrate wurde die Implementierung basierend auf der INRIA Datenbank [15] mit der CPU-Implementierung (R-HOG mit linearer SVM) verglichen (Abbildung 7). Hierbei zeigt sich eine um 6% höhere Miss Rate zum original R-HOG bei 10⁻³ FPPW. Dies ist vor allem durch das Fehlen der trilinearen Interpolation bei der Histogrammerstellung und die Rundung der einzelnen Berechnungen (v. a. bei der Normierung) erklärbar.

V. ZUSAMMENFASSUNG

Die hier vorgestellte Implementierung beschleunigt die Berechnung und Klassifikation der HOG-Deskriptoren erheblich. Im Vergleich zu bestehenden Veröffentlichungen [5]-[7],[11] ergibt sich ein Geschwindigkeitsunterschied um mehr als den Faktor 10 (bei der Multiskalen-Implementierung aus Tabelle 2). Durch den Einsatz eines doppelt so hohen Core-Taktes und weiterer Optimierungsmethoden konnte der Ressourcenbedarf reduziert werden, ohne die Echtzeitfähigkeit der Implementierung zu beeinträchtigen. Basierend auf dem Zeitmultiplex-Verfahren konnte die Anzahl der berechenbaren Skalinstufen auf die bei CPU- und GPU-Implementierungen übliche Anzahl erhöht werden. Aufgrund der parametrierbaren Multiskalenmodule ist eine Anpassung an unterschiedliche Hardwareplattformen einfach möglich.



LITERATURVERZEICHNIS

- [1] N. Dalal, B. Triggs, „Histograms of Oriented Gradients for Human Detection“, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 886-893, San Diego, Juni 2005.
- [2] S. Bauer, „Konzepte zur Anbindung von Kameras an FPGAs unter Berücksichtigung von Echtzeitbedingungen in der Verkehrssicherheit“, *Master Thesis*, Aschaffenburg, Mai 2012.
- [3] N. Dalal, „Finding People in Images and Videos“, *PhD*, Grenoble, Juli 2006.
- [4] S. Bauer, U. Brunsmann, S. Schlotterbeck-Macht, "FPGA Implementation of a HOG-based Pedestrian Recognition System", *42. Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, p. 49-58, Karlsruhe, Juli 2009.
- [5] S. Bauer, S. Köhler, K. Doll, U. Brunsmann, "FPGA-PGU Architecture for Kernel SVM Pedestrian Detection", *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, p. 61-68, San Francisco, Juni 2010.
- [6] K. Negi, K. Dohi, Y. Shibata, K. Oguri, "Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm", *2011 International Conference on Field-Programmable Technology*, p. 1-8, New Delhi, Dez. 2011.
- [7] K. Lillywhite, D.J. Lee, D. Zahng, "Real-time Human Detection Using Histograms of Oriented Gradients on a GPU", *2009 Workshop on Applications of Computer Vision*, p. 1-6, Snowbird, Dez. 2009.
- [8] N. Dalal, B. Triggs, C. Schmid "Human Detection Using Oriented Histograms of Flow and Appearance", *ECCV 2006: 9th European conference on Computer Vision*, p. 428-441, Graz, Mai 2006.
- [9] M. Berberich, K. Doll, "Highly Flexible FPGA-Architecture of a Support Vector Machine", *45. Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, p. 25-32, Albstadt-Sigmaringen, Feb. 2011.
- [10] J. Brookshire, J. Steffensen, J. Xiao, "FPGA-based Pedestrian Detection", Mai 2010.
- [11] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, M. Yoshimoto, "Architectural Study of HOG Feature Extraction Processor for Real-Time Object Detection", *2012 IEEE Workshop on Signal Processing Systems*, Québec City, Okt. 2012.
- [12] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, "Hardware Architecture for HOG Feature Extraction", *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, p. 1330-1333, Kyoto, Sept. 2009.
- [13] M. Hiromoto, R. Miyamoto, "Hardware Architecture for High-Accuracy Real-Time Pedestrian Detection with CoHOG Features", *12th International Conference on Computer Vision Workshops (ICCV Workshops)*, p. 894-899, Kyoto, Sept. 2009.
- [14] V. Prisacariu, I. Reid, "fastHOG - a real-time GPU implementation of HOG", *Tech. Rep. 2310/09*, Oxford University, Jan. 2012.
- [15] N. Dalal, B. Triggs, "INRIA Person Dataset" 2005, <http://pascal.inrialpes.fr/data/human/>, Datum des Zugriffs: Feb. 2013.



Michael Hahnle erhielt den akademischen Grad des B. Eng. in Elektro- und Informationstechnik im Jahr 2012 von der Hochschule Aschaffenburg, wo er sein Masterstudium in Elektro- und Informationstechnik voraussichtlich im Jahr 2013 abschließen wird.



Frerk Saxen erhielt den akademischen Grad des B. Eng. in Elektro- und Informationstechnik im Jahr 2010 und den Grad des M. Eng. in Elektro- und Informationstechnik im Jahr 2011 von der Hochschule Aschaffenburg. Zurzeit promoviert er an der Otto-von-Guericke-Universität in Magdeburg am Lehrstuhl für Technische Informatik.



Konrad Doll erhielt den akademischen Grad des Dipl.-Ing. in Elektro- und Informationstechnik im Jahr 1989 und den Grad des Dr.-Ing. im Jahr 1994 von der Technischen Universität München. Er ist Professor für Technische Informatik und Entwurf integrierter Schaltungen an der Hochschule Aschaffenburg.

Headless Android Systems for Industrial Automation and Control

Aneeqe Hassan, Axel Sikora, Dominique-Stephan Kunz, David Eberlein

Abstract—Android is an operating system which was developed for use in smart mobile phones and is the current leader in this market. A lot of efforts are being spent to make Android available to the embedded world, as well. Many embedded systems do not have a local GUI and are therefore called headless devices. This paper presents the results of an analysis of the general suitability of Android in headless embedded systems and ponders the advantages and disadvantages. It focuses on the hardware related issues, i.e. to what extent Android supports hardware peripherals normally used in embedded systems.

Index Terms—Android, Accessory Development Kit (ADK), Android Open Source Project (AOSP), Asynchronous Shared Memory (Ashmem), Micro-controllers (MCUs).

I. INTRODUCTION

Android is an operating system for mobile devices developed by the Open Handset Alliance (OHA). Android enjoys being the market leader in the field of smart phones with a market share of more than 70 % [1]. Android is based on the Linux kernel. So the question arises if there are any additional benefits in using Android instead of Linux for applications outside the smart phone or tablet world, i.e. for applications from industrial, process, or building automation.

The objective of this project was to discuss the suitability, i.e. the advantages and disadvantages of Android for headless systems in industrial automation. In the projected case, the anticipated end product was a system without a local user interface, but with the option to add a GUI for a certain percentage of the applications. This paper will only cover the use case that all applications are Android based and there are no Linux applications. A heterogeneous system (Android and Linux applications) would need further investigations.

Aneeqe Hassan, ahassan@stud.hs-offenburg.de, and Axel Sikora, axel.sikora@hs-offenburg.de, are with Hochschule Offenburg, Badstraße 24, D77652 Offenburg.

Dominique-Stephan Kunz, dominique.kunz@ch.sauter-bc.com, and David Eberlein, david.eberlein@ch.sauter-bc.com, are with Fr. Sauter AG, Im Surinam 55, CH4058 Basel.

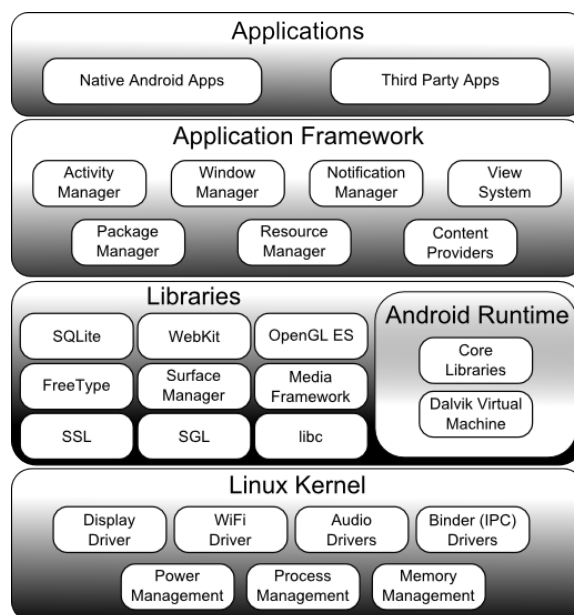


Figure 1: Architecture of Android [2].

A basic question in this context is the support for external hardware peripherals. Specifically built for smart phones Android does not need to provide generic support for external interfaces like SPI, UART, GPIO or similar peripherals as opposed to "normal" embedded systems.

In this paper first the architecture of Android is presented in ch. II and the concept of headless Android is introduced in ch. III. Then the used baseboard is shortly presented in ch. IV. The extension of Android based systems with additional local hardware interfaces was the most important part of the project. The mechanism of hardware support in Android is discussed in ch. V. Then the timing results of the test setup are presented in detail in ch. VII and ch. VIII. The report is concluded by giving an opinion based on the findings during the project.

II. ANDROID ARCHITECTURE

Android is based on the Linux kernel, a number of software libraries written in 'C', a Java-like virtual machine for the execution of application code, and a full suite of mobile phone applications [3]. Fig. 1 shows an overview of the Android architecture.

The Linux kernel in the Android is a modified version of the kernels available from the Linux kernel archives, often also called vanilla Linux kernel [4]. These modifications include Alarm, Asynchronous shared memory (Ashmem), Binder, Power Management, Low Memory Killer, Kernel Debugger and Logger [5]. Android uses the Linux kernel to control the underlying hardware.

III. ANDROID FOR HEADLESS EMBEDDED SYSTEMS

Headless embedded devices do not have a local user interface like mobile devices, which Android was originally created for. Android for headless devices could therefore omit those components that are used to draw and control the user interface. The components, which are not required for the headless-only use case, are SurfaceFlinger, Window-Manager, Wallpaper-Service and InputMethodManager [6].

IV. THE BASE BOARD

The selection of the base board is an important step, as there are a lot of aspects to be considered in this process, especially in the case of porting Android to a new device. The parameters to be taken into account are SoC features, community, cost, features, expandability, availability, licensing, catalogue part, and software support [7].

In the given project the focus was not on the porting of Android, but on the features it offers. The project work was started by testing requirements on the Raspberry Pi [8] and then continued on the "Tsunami Pack" by TechNexion [9]. The Raspberry Pi was not used further because the Android builds available for it are still not stable. Figure 2 shows the board used in the setup of the test environment.

V. HARDWARE SUPPORT MECHANISMS

A. General Architecture

Many embedded devices need to collect data from local sensors, analyze and process the data, and control local actuators accordingly. In some cases, they are connected to backend control and monitoring systems for proper operation. All these embedded scenarios require different hardware interfaces like GPIO, SPI, UART PWM or alike. This connectivity between sensors and actuator devices through local hardware interfaces is normally not required in smart phones. The Android architecture has been optimized for smart phones, and thus, only includes a minimum of hardware drivers required for smart phones [10].



Figure 2: TechNexionTsunami Board running Android.

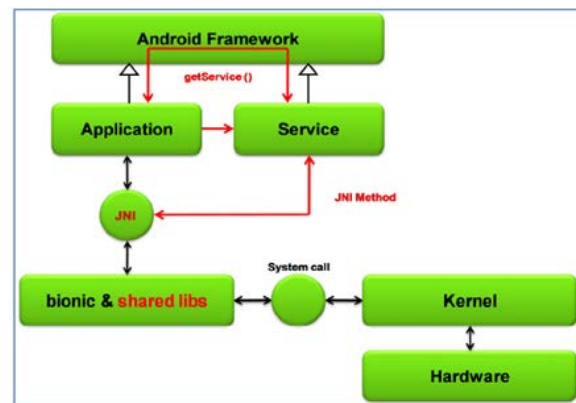


Figure 3: Android hardware access mechanisms [11].

In the changes made to the standard Linux kernel for Android, the approach of hardware support has also been changed. Standard Linux provides access to hardware interfaces via device drivers. In Android hardware access is typically supported through shared libraries provided by manufacturers. Most software stacks typically found in Linux distributions that enable interaction with local hardware are not found on Android [11]. Figure 3 shows how hardware is accessed on Android.

Unlike native Linux applications, Android applications communicate with the underlying hardware through Java APIs, direct system calls are not possible. Android defines APIs which can be utilized by applications to communicate with the underlying hardware. A system service is used to load and interface the shared libraries and provide the API functionality to the application. These system services consist of two parts. One is written in Java and the other one in C. The Java part implements the Android side of the service and the C part communicates with the shared library.

Consequently, there exist multiple ways to add local hardware interfaces to an Android run-time system:

- Add shared libraries to the source code and make a custom port to Android (cf. ch. B.1)).
- Use the Android Open Accessory Mode (cf. ch. B.2)).
- Use external MCUs connected over USB or Bluetooth (cf. ch. B.3)).

B. Accessing local interfaces from Android

1) Android Porting

This solution requires adding the needed hardware drivers to the Linux kernel and applying the Android patch to the kernel as a first step. As a second step, the shared libraries should be made available in the source code of Android. This custom version must then be ported to the target device. The best part of this approach is that after the accomplishment of this task only application level programming would be required to make use of these peripherals like the USB and WiFi APIs are used in Android normally. The downturn of this approach is that detailed knowledge of the hardware would be required, which mostly calls for support from the hardware vendor.

Generally speaking, this approach fundamentally contradicts the overall Android approach, which strives to avoid complex Linux programming and tries to replace it with comparatively easier Java programming. In the end this approach would double the work, because first Linux programmers would build their own build of Android to include the libraries in the OS and then the Java developers would use these libraries in the application development phase. This would also increase the initial development cycle of the product but with a much more stable result.

2) Android Open Accessory Mode

The second option to add hardware interfaces comes from the Android Open Accessory mode, which allows the connection of peripherals to an Android platform where the Android is the USB device and the peripheral (accessory) is the USB host [6]. This avoids the necessity of new hardware drivers.

Google implemented its own hardware by the name Android Accessory Development Kit (ADK) as a reference implementation to build hardware accessories for Android. The ADK is based on the Arduino open source electronics prototyping platform [12]. Android Open Accessory [13] protocol is used to communicate with the Android device over a USB or a Bluetooth connection. This mode is not supported on a lot of devices and as reported by various senior developers on communities it is not so robust [14].

3) External Microcontrollers

The third possible solution is very similar to the one used by Google for their Google ADK. It uses the USB or Bluetooth interfaces of the Android device to

attach an external MCU, where the MCU implements and provides the actual connectivity to the local interfaces. There are external boards already available on the market, like the FT311D from FTDI [15] or the IOIO from Sparkfun [16]. These boards are delivered with a custom API library which may be included into an Android application and provides access to the boards peripherals. The good side is that for these boards the development time decreases considerably, compared to integrating Linux drivers.

C. Evaluation of methods to access local interfaces

If a company develops their own board and ports Android on their own, then they will write their own drivers for their system. This is the best way to offer these interfaces to the application. If a company wants to use an off the shelf board with Android already running on it but the required interfaces are missing, the Android Open Accessory Mode or the external MCU approach will be the approach with less effort.

The latter use case is interesting for this investigation because the TechNexion board does not support the drivers needed for the hardware. Furthermore the ADK is not orderable. Therefore, it was decided to go with the external MCU approach as a solution to add peripheral support to Android. The IOIO board was used for this work.

The board is created by a day-time Google employee as a hobby project and is sold by Sparkfun. The board connects to the Android device in debug mode using USB OTG or Bluetooth. An Android application can utilize the IOIO libraries to access functionalities on the board. No embedded programming is required. Another good thing is that it works with all Android versions above 1.5.

VI. HARDWARE FOR THE TARGET SYSTEM

The target platform requires a lot of hardware interfaces to work. In this section each interface will be discussed one after the other.

A. Tsunami Mainboard

1) SD-Card

As large memory storage is required to store logged data, an SD-Card could easily solve this requirement. The SD-Card support is provided by default in Android and is fully functional on the Tsunami board.

2) UART

A serial UART-interface is still an important way to access hardware peripherals on embedded systems. There are no standard APIs present in Android to read and write to Linux serial ports. However, there is an open source project which makes it possible to read and write data through the serial ports of Linux. The

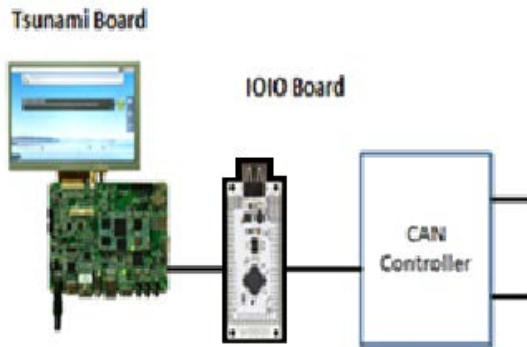


Figure 4: Block diagram of CAN Controller connected to the Android device using IOIO.

project name is "android-serialport-api" [17]. The serial ports on the Tsunami board can be used with this api. Moreover, the Modbus [18] protocol can be used. The application level protocol operates directly on top of a serial interface and serial communication standards [19].

3) GPIO, SPI and CAN

The support for other peripherals is present on the Tsunami board. However, no out of the box solutions or precompiled libraries exist for them. The IOIO board was used to provide the other peripherals required for the system.

B. Using IOIO Board

1) GPIO

All 48 pins on the IOIO board can be configured and used as GPIO's. For this both digital and analogue input/output tests were performed. For the digital output an LED was successfully controlled with the IOIO board. Using the IOIO board implementing the GPIO functionality is quite straight forward.

2) SPI

There is a special class in the IOIO library called "SpiMaster.java". Controlling the IOIO SPI interface is done via the SpiMaster class. An instance of this interface corresponds to a single SPI module on the board, as well as to the pins it uses for CLK, MOSI, MISO and SS (one or more) [20].

In the test setup it was tested to make the SPI work in the Master mode using a fixed buffer of 0x55 and it transmitted successfully. During the tests the transmitted signal was good up to 1 Mbps but it started distorting after that and it worked till 3.2 Mbps. At higher speeds, the output signal degraded significantly.

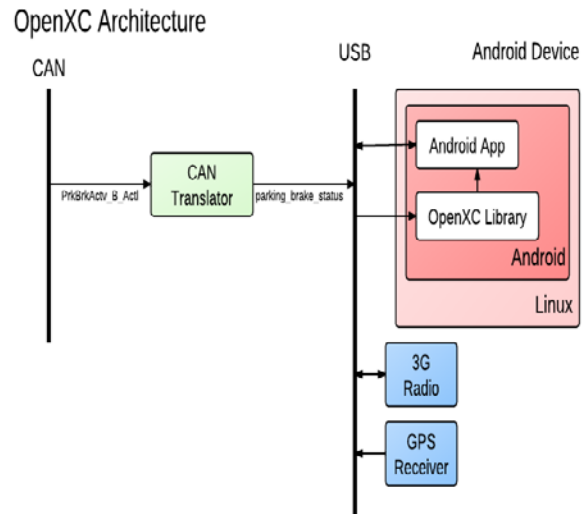


Figure 5: OpenXC architecture [22].

3) CANbus

CAN Bus [21] is also not natively supported on Android. One possible solution is to use the SPI of IOIO board to control a CAN controller from the Android device. Such a CAN controller is provided by Microchip, MCP2515 [23]. Figure 4 shows the block diagram of the proposed solution.

Another interesting solution is the OpenXC architecture [22] developed by Ford and Bug labs. It connects the CAN bus to the Android device over USB using a CAN translator in between. Figure 5 shows the overall architecture. The Android libraries of the project are available on the projects webpage.

4) UART

The IOIO Board also offers a UART interface, which can be used in a similar way as the Tsunami boards UART.

VII. IOIO TIMING TESTS

The purpose of the tests was to test the round trip time of communication from Android to the IOIO board and back. The idea was to measure the delays that incurred during this communication process. These tests were performed to assess the setup's feasibility for target systems communication timing requirements, which are very strict for automation control systems. The goal was to achieve a signal cycle time of less than 12.5 ms because it sufficed the timing requirements of the target automation systems.

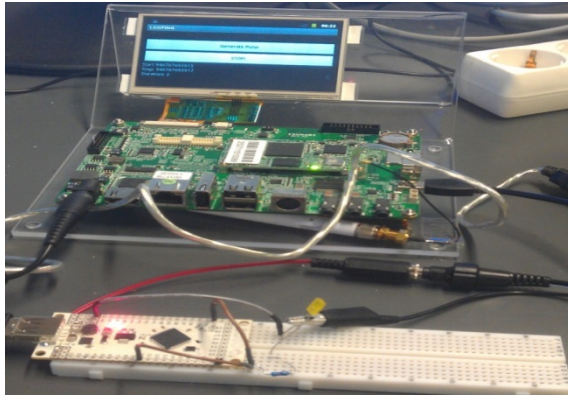


Figure 6: The hardware setup for the tests.

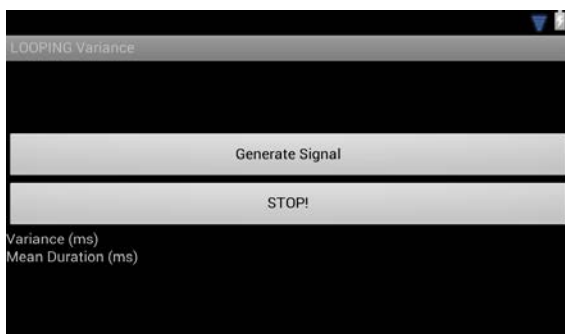


Figure 7: The screen shot of the application.

A. Hardware Setup

The set-up includes the Tsunami board running Android connected to the IOIO board through USB in Android Debug Bridge (ADB) mode. The other possibility would have been a connection over Bluetooth which would additionally include its own delays. Therefore the direct physical connection via USB was used instead.

For the hardware, one pin (#10) of the IOIO board was set as digital output in the application, which turns on an LED, when set to true (on), and turns it off, when set to false (off). The output of this pin (#10) is given as input to another pin on the board (pin #33) which is configured as digital input. With this setup a hardware loop is created on the IOIO board, which forms a cycle of on and off signals like a ring oscillator. An oscilloscope was used to monitor the signal across the LED, which was the desired repeating digital signal. Fig. 6 shows the hardware setup for the tests.

B. Software Application

The software uses the IOIO library and turns on the digital output pin when the "Generate Signal" button is pressed in the application. If the logical state of the input pin is changed, then this state is inverted and set to the output pin. The process of changing the logical state of the output pin until the input pin represents

this logical state is one cycle. By pressing the "Generate signal" button this process runs in a loop until 1000 if these cycles are run through and subsequently the mean value of the readings and the variance of the read values are displayed using the TextView widget available in the Android API. Fig. 7 shows a screen shot of the application that was developed to perform the tests.

C. Measurements

Two types of measurements were taken for the tests. One was software timing in the application itself for one complete cycle off on and off (which means two cycles of the IOIO board). The other one was using an oscilloscope across the LED in the hardware setup to measure the time period of one signal cycle.

The software measurements could be halved and compared to the duration of one signal cycle. Alternatively two signal cycles (on and off) could be considered as one software cycle duration and the timings could be compared. The second option of taking two consecutive on and off signals was chosen. The results are discussed below.

D. Test Cases

1) Different Load Conditions

One important aspect is the dependency on the load conditions of the CPU, i.e. from other processes running quasi-simultaneously in the Android device. Three load conditions were tested to generate a set of results. The first one was to run the test application by itself, with no other applications running. The second one was with a medium load condition with some other applications running in the background. The third one was a high load condition with even more applications running in the background.

2) Sleep Function

For another test scenario the sleep function was introduced in the loop test. In each operation the looping thread of the IOIO test application is put to sleep for different amounts of milliseconds. After the time expires the thread is available again for scheduling and is executed [24]. This was done to get an idea of the delays that are incorporated by releasing and reacquiring the processor.

3) Run as Service

Another test scenario was to do the timing tests by communicating with the IOIO board in an Android service. A service is just like a foreground application (activity). The only difference is that it does not have a user interface and runs only in the background without any user interaction. The looping test was the same as before, this time only the application was run as a

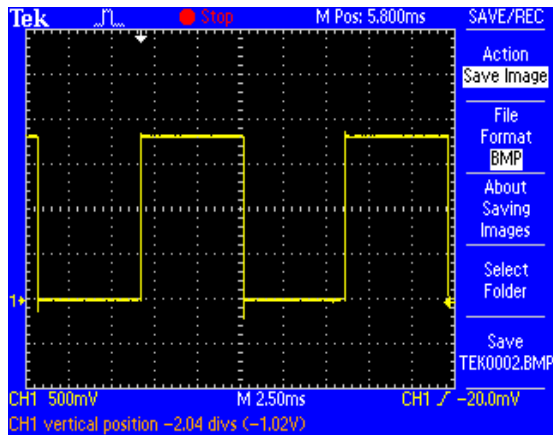


Figure 8: Oscilloscope measurements for the low load scenario.



Figure 9: Software application readings for the low load scenario test.

service, everything else was the same. Further three sub scenarios for different load conditions were tested which included the following:

- Run only the IOIO service.
- Run the service plus the music player application continuously running in background as a service.
- Use the service, the music player playing songs plus playing a game at the same time

VIII. RESULTS

A. Tests without sleep

On average a delay of 12 ms was experienced for the two signal cycles. From the measurements it was seen that adding the load didn't really affect the average duration a lot (at least in milliseconds) but the variance did increase when increasing the number of applications running. Using a web server during the IOIO operation increased the mean duration as well, which was obvious due to processor scheduling. But apart from only those specific times the oscilloscope readings were the same. This explains Android scheduling as well. The foreground activities have most (~90 %) of the CPU time whereas background activities share only the remaining resources (~10 %) [25]. Fig. 8 shows the oscilloscope image of one of the tests being carried out for the low load scenario.

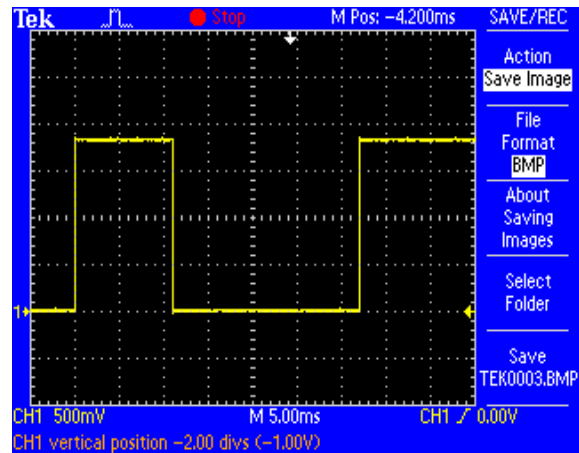


Figure 10: Oscilloscope measurements for the high load scenario and 10 ms sleep time.

Table 1. Repeated readings for the low load test scenario.

Application with no sleep (low load)			
test #	mean duration (ms)	variance (ms)	oscilloscope reading (ms)
1	12.09	0.28	12
2	12.21	0.49	12
3	12.23	0.54	12

Fig. 9 shows the software results of the timing test for the same scenario presented above. Table 1 shows the timing measurements for three repeated tests for the same scenario, which are consistent enough to deduce results. Readings were noted for the other test scenarios as well, but are omitted here due to limited space.

B. Tests with sleep

For this set of measurements a huge variation between the oscilloscope readings was observed. Also when compared to the readings of the software application the readings don't match as for the test measurements in the "no sleep scenarios". However, the mean values are consistent between different tests. This indicates that the oscilloscope reading depends a lot on the point in time when the oscilloscope was stopped to take a measurement. Whereas the difference between the mean value in the software application and the measured oscilloscope values presumably are caused by the increased variance. The mean values didn't change a lot in the computed results but the variance increased a lot.

The results also depict that the delays added by adding more load to the processor are small enough to remain in the given timing requirements. Adding the sleep functionality increases the timing of the whole setup by almost the same amount as the sleep function puts the thread on a hold.

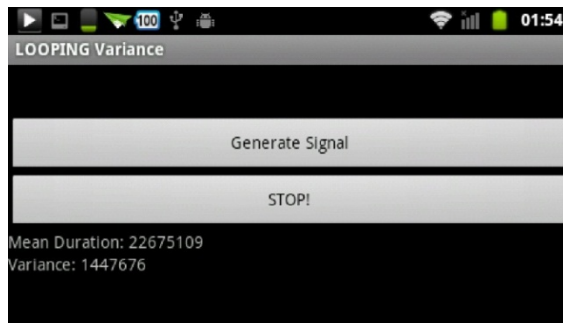


Fig 11: Software application readings for the high load and 10 ms sleep scenario test.

Table 2: Repeated readings for the high load test scenario.

Application with 10 ms sleep (high load)			
test #	mean duration (ms)	variance (ms)	oscilloscope reading (ms)
1	22.67	1.45	32 (varying)
2	22.55	1.28	32 (varying)
3	22.28	0.91	32 (varying)

Fig. 10 shows the oscilloscope image of one of the tests being carried out for the high load scenario and a sleep time of 10 ms. Fig. 11 shows the software results of the timing test for the same scenario presented above. Table 2 shows the timing measurements for three repeated tests for the same scenario.

In summary if a process is released by the CPU and it comes back after some time, the delays incorporated are still bearable according to the timing requirements. Adding the load also doesn't affect the communication in an adverse manner. The sleep function only adds delays roughly of the passive time of the sleep function.

C. Test of Running as a Service

The same logic of testing was applied in another Android program which was run as service under different load conditions. From the oscilloscope images it was clear that the variance is so high at different times that the mean delay of communication has also increased to a high number. These high numbers will not meet the desired communication timing requirements so running the application in the foreground as an activity would be advisable to stay under the communication delay requirements. Fig. 12 shows the oscilloscope image of one of the tests being carried with the music service and game being run simultaneously. Fig. 13 shows the software results of the timing test for this very scenario.

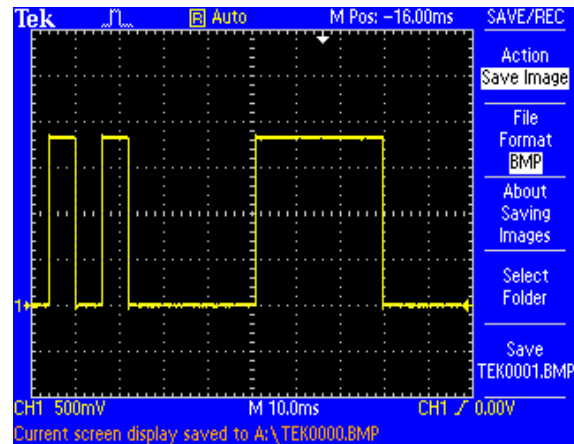


Fig 12: Oscilloscope measurements for the service, music service and game run together.

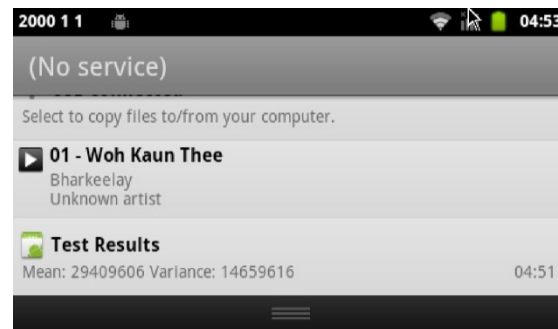


Fig 13: Software application readings for the service, music service and game run together.

It is interesting that when the service is running alone, it experiences significantly larger mean values and variances in comparison to the situation when other services are simultaneously executed on the Android system. For the moment, it can just be assumed that the power management has an impact also on the scheduler. Perhaps the scheduler calls the background services more often if there is more than one service active, which causes this unpredicted behavior.

IX. CONCLUSION

The target systems are headless systems with the option to add a local GUI to the system if required. These headless systems require different hardware interfaces than the ones used in Android smartphones. The standard procedure of integrating hardware interfaces in Android is a bit complex but there are workarounds available which were presented in this paper. The delays these elements add to the whole communication setup are still bearable in most scenarios as proved from the tests performed during this project.

From the work it was seen that Android is set to join the embedded world. If the target systems are headless with a small percentage being GUI enabled as well, it is recommendable to use Android as the OS. This will give the advantage of having a single platform with

slightly different features. This would reduce the development phase and costs. To summarize we think that using Android in headless devices in industrial automation is attractive, as soon as they have optional GUIs.

REFERENCES

- [1] <http://mashable.com/2012/11/14/android-72-percent/>. Retrieved 3 April 2013.
- [2] http://www.techotopia.com/index.php/An_Overview_of_the_Kindle_Fire_Android_Architecture. Retrieved 3 April 2013.
- [3] <http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Porting-Android-to-a-new-device/>. Retrieved 3 April 2013.
- [4] <http://www.kernel.org/>. Retrieved 3 April 2013.
- [5] F. Maker, Y.-H. Chan, *A Survey on Android vs. Linux*, http://handycodeworks.com/wp-content/uploads/2011/02/linux_versus_android.pdf. Retrieved 3 April 2013.
- [6] <http://www.opersys.com/blog/headless-android-1>. Retrieved 3 April 2013.
- [7] K. Yaghmour, *Embedded Android Porting, Extending, and Customizing*, O'Reilly Media, 2011.
- [8] <http://www.raspberrypi.org/quick-start-guide>. Retrieved 3 April 2013.
- [9] <http://technexion.com/index.php/products/arm-cpu-modules/ti-omap3530/tsunami>. Retrieved 3 April 2013.
- [10] FTDI Whitepaper, *Connecting peripherals to an Android Platform*. Retrieved 3 April 2013.
- [11] <http://sujaiantony.wordpress.com/category/android-2/>. Retrieved 3 April 2013.
- [12] <http://www.arduino.cc/>. Retrieved 3 April 2013.
- [13] <http://source.android.com/tech/accessories/aoap/aoa2.html>. Retrieved 3 April 2013.
- [14] <https://groups.google.com/forum/?pli=1#!searchin/ioio-users/frequency/ioio-users/EIEGF2h8Y8c/uAutFXUPvjYJ>. Retrieved 3 April 2013.
- [15] White paper, FTDI, USB Android Host Module.
- [16] S. Monk, *Making Android Accessories with IOIO*, O'Reilly Media, 2012.
- [17] <http://code.google.com/p/android-serialport-api/>. Retrieved 3 April 2013.
- [18] <http://www.modbus.org/>. Retrieved 3 April 2013.
- [19] http://jamod.sourceforge.net/kbase/protocol.html#sub_serial. Retrieved 3 April 2013.
- [20] <https://github.com/ytai/ioio/wiki/SPI>. Retrieved 3 April 2013.
- [21] <http://www.can-cia.org/index.php?id=46>. Retrieved 3 April 2013.
- [22] <http://openxcplatform.com/getting-started/index.html>. Retrieved 3 April 2013.
- [23] Data Sheet, MCP2515, Microchip.
- [24] <https://github.com/keesj/gomo/wiki/AndroidScheduling>. Retrieved 3 April 2013.
- [25] <http://stackoverflow.com/questions/8081042/sleep-function>. Retrieved 3 April 2013.



Aneeqe Hassan was awarded the degree of Bachelors of Science from FAST-NUCES University in Lahore (Pakistan) in 2009. He is currently a student of M.Sc. in Communication and Media Engineering in the Hochschule Offenburg and works in the Embedded Communications Lab with Prof. Dr. Axel Sikora.



Axel Sikora holds a diploma of Electrical Engineering and a diploma of Business Administration, both from Aachen Technical University. He has done a Ph.D. in Electrical Engineering at Fraunhofer Institute of Microelectronics Circuits and Systems, Duisburg. After various positions in the telecommunications and semiconductor industry, he became a professor at the Baden-Wuerttemberg Cooperative State University Loerrach in 1999. In 2011, he joined Offenburg University of Applied Sciences, where he holds the professorship of Embedded Systems and Communication Electronics. His major interest is in the system development of efficient, energy-aware, autonomous, secure, and value-added algorithms and protocols for wired and wireless embedded communication. He is also founder and head of Steinbeis Transfer Center Embedded Design and Networking (stzedn). Dr. Sikora is author, co-author, editor and co-editor of several textbooks and numerous papers, as well as conference committee member to various international conferences in the field of embedded design and wireless and wired networking.



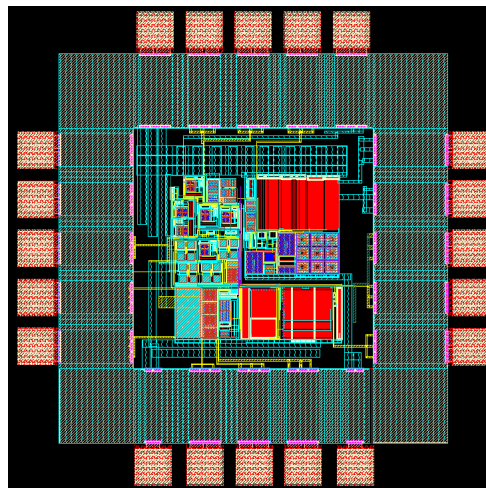
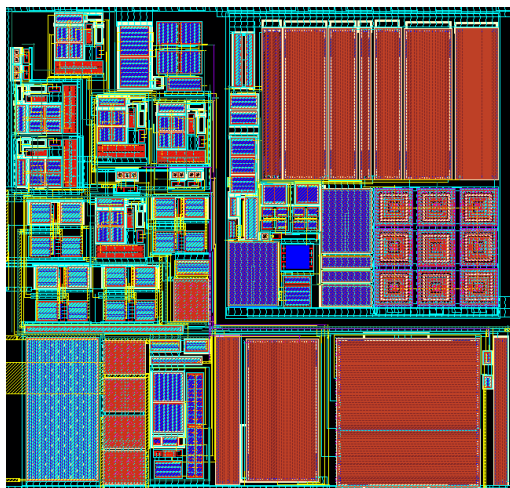
Dominique Stephan Kunz studied electrical engineering (with a focus on control and automation engineering) at the University of Applied Sciences in Basel. Since 1999, he has been working for the company Fr. Sauter AG in Basel. He has over 10 years of experience in industrial product development and project management in the heating, ventilation and air conditioning market with respect to building automation. Currently, he heads the Technologies department, which deals with research projects, and he is responsible for cooperation with universities.



David Eberlein graduated as Bachelor of Engineering in Information Technology at the Cooperative State University Lörrach in 2009, winning the prize of the city Lörrach for the best bachelor thesis of the year. Since then, he has been working for Fr. Sauter AG as Software Engineer at the Technologies department. His main interests are web based graphical user interfaces, as well as mobile applications.



Testchip für ein Energy Harvesting IP in 0,18 μm CMOS-Technologie



ENTWURF	Jens Rechtsteiner, Gerhard Forster Hochschule Ulm, Institut für Kommunikationstechnik, Prittitzstraße 10, 89075 Ulm
ENTWURFSVERFAHREN	Full Custom Design
TECHNOLOGIE	UMC L180 Mixed-Mode/RF 180 nm CMOS 6M/1P 1.8V/3.3V
CHIPFLÄCHE	0,89 mm x 0,89 mm (Core 0,35 mm x 0,35 mm)
GEHÄUSE	QFN 24
FUNKTIONSBLOCKE	Sperschwinger, Aktivgleichrichter, Bandgap-Referenz, Schaltregler, Linearregler, Logik. Extern wird ein Miniaturtransformator benötigt.
FUNKTION	Der Chip enthält den IP-Core einer Versorgungseinheit, die eine Versorgungsspannung von wahlweise 1,8 V oder 3,3 V zum Betrieb eines ASICs zur Verfügung stellt. Die Versorgungseinheit kommt ihrerseits mit einer Speisespannung von weniger als 100 mV aus. Wird dieser IP-Core in ein ASIC eingesetzt, so kann der Chip mit einer sehr geringen Versorgungsspannung betrieben werden, die sich z.B. mit Peltier-Elementen aus geringen Temperaturdifferenzen erzeugt lässt. Die Eigenstromaufnahme der Versorgungseinheit liegt bei 20 μA , der maximale Laststrom bei 1 mA.
HERSTELLDATUM	III. Quartal 2012
KOSTENTRÄGER	MPC-Gruppe Baden-Württemberg
VERÖFFENTLICHUNG	J. Rechtsteiner, G. Forster, „Ein Energy Harvesting IP für den Einsatz in einem 0,18 μm CMOS ASIC“, <i>Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg</i> , Offenburg, Tagungsband ISSN 1868-9221, S. 1 – 12, Ausgabe 47, Februar 2012.

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
rue@hs-furtwangen.de

Hochschule Heilbronn

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

Hochschule Karlsruhe

Prof. Dr. Koblitz, (0721) 925-2238
rudolf.koblitz@hs-karlsruhe.de

Hochschule Konstanz

Prof. Dr. Schoppa, (07531) 206-644
irenaeus.schoppa@htwg-konstanz.de

Hochschule Mannheim

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

Hochschule Offenburg

Prof. Dr. Jansen, (0781) 205-267
d.jansen@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. Hohlfeld (07121) 271-7081
dennis.hohlfeld@reutlingen-university.de

Hochschule Ulm

Prof. Dipl.-Phys. Forster, (0731) 50-28338
forster@hs-ulm.de

www.mpc.belwue.de