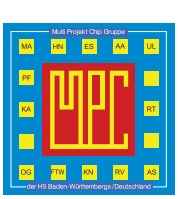
# MPC

MULTI PROJEKT CHIP GRUPPE
BADEN-WÜRTTEMBERG

**Herausgeber:** Hochschule Ulm **Ausgabe:** 54 **ISSN** 1868-9221 **Workshop:** Ulm Juli 2015







Cooperating Organisation
Solid-State Circuit Society Chapter
IEEE German Section

# Inhaltsverzeichnis

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

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie.

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

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

Mitherausgeber (Peer Reviewer):
Jürgen Giehl, Hochschule Mannheim, Paul-Wittsack-Straße 10, D-68163 Mannheim
Frank Kesel, Hochschule Pforzheim, Tiefenbronner Straße 65, D-75175 Pforzheim
Axel Sikora, Hochschule Offenburg, Badstraße 24, D-77652 Offenburg



Diesen Workshopband und alle bisherigen Bände finden Sie im Internet unter:
http://www.mpc.belwue.de

# EUV Lithografie – optische Spitzentechnologie als Grundstein moderner Chipfertigung

Tilmann Heil

***Zusammenfassung*—Optische Nanolithografie ist ein Schlüsselverfahren der modernen Chipfertigung, bei dem optische Systeme höchster Performanz zur Strukturierung der Chips eingesetzt werden. Das „Moore'sche Gesetz" der Halbleiterindustrie, das eine fortlaufende Miniaturisierung der Chip-Strukturen postuliert, steht dabei in direktem Zusammenhang mit Fortschritten in optischen Technologien, da die minimal erreichbare Strukturgröße auf einem Chip direkt mit dem Auflösungsvermögen der verwendeten Optik verbunden ist. Der nächste große Schritt zu noch höherer optischer Auflösung und weiterer Verkleinerung der Chipstrukturen steht mit der Einführung der Extreme Ultra Violet (EUV) Lithografie in die Chip-Volumenfertigung unmittelbar bevor. Aufgrund der kurzen Belichtungswellenlänge von 13,5 nm ermöglicht die EUV-Lithografie eine signifikante Steigerung des Auflösungsvermögens gegenüber der heute eingesetzten Immersionslithografie mit 193 nm Belichtungswellenlänge. Dieser Beitrag stellt die optischen Systeme der EUV-Lithografie vor und diskutiert die besonderen Herausforderungen und Lösungen dieser Technologie. Ein Ausblick auf zukünftige EUV-Optiken mit weiter gesteigerter Auflösung zeigt, dass die EUV-Lithografie über mehrere Technologieknoten hinweg erweiterbar ist und damit eine Fortsetzung der Halbleiter-Roadmap bis unter 9 nm Auflösung hinaus ermöglicht.**

***Schlüsselwörter*—Optische Lithografie, EUV.**

## I. Einleitung

Bereits 1965 machte der Intel-Mitgründer Gordon Moore die Beobachtung, dass sich die Anzahl der Transistoren eines integrierten Schaltkreises auf einem Chip etwa alle 2 Jahre verdoppelt [1]. Als Treiber dieser Entwicklung identifizierte Moore die Tatsache, dass die relativen Herstellkosten pro Komponente mit der Verkleinerung der Strukturgrößen, also steigender Integrationsdichte, sinken. Dieses sogenannte „Moore'sche Gesetz" ist seit 50 Jahren gültig, ein einmaliger Vorgang in der Industriegeschichte. Ein durch Kostenvorteile getriebener Kreislauf ermöglicht fortlaufende Innovationsschübe: neue Produktionsmaschinen ermöglichen die Verkleinerung der Chip-Strukturen, damit sinken die Kosten pro Funktionseinheit, womit auch die Preise von High-Tech-Geräten stark fallen. Dadurch erschließen sich neue und größere Märkte. Die dabei erzielten Gewinne werden wiederum in Technologie und letztlich in noch leistungsfähigere Produktionsmaschinen investiert, was den Innovationskreislauf von Neuem beginnen lässt. Es ist daher nicht verwunderlich, dass sich ein Multi-Milliarden Markt für Geräte zur Herstellung von Halbleitern entwickelt hat. Das weltweite Marktvolumen für Halbleiterkomponenten wird 2015 etwa 340 Mrd. $ umfassen [2]. Über 30 Mrd. $ davon gehen in die Geräte zur Herstellung der Halbleiter, wovon ein sehr großer Anteil mit knapp 7 Mrd. $ auf die Lithografiegeräte, die sogenannten Wafer-Scanner, entfällt. Dieser hohe Anteil begründet sich durch die besondere Wertigkeit des Lithografieschritts in der hochkomplexen Herstellungskette eines Chips.

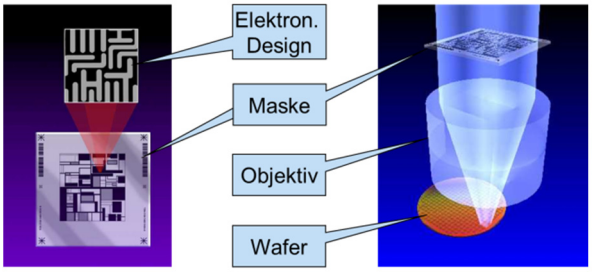



Abbildung 1: Illustration des Funktionsprinzips der optischen Lithografie.

Abbildung 1 zeigt schematisch das optische Lithografie Verfahren zur Strukturierung der Chips. Das Design des zu erzeugenden elektronischen Schaltkreises wird in viele Ebenen zerlegt (typischerweise > 50). Das Bild jeder Ebene wird physisch auf eine sogenannte Maske aufgebracht. Die Maske wird durch ein optisches System im Wafer-Scanner auf den zuvor mit lichtempfindlichem Lack beschichteten Wafer abgebildet. Dieses Verfahren ermöglicht eine kontaktfreie Vervielfältigung der Maskeninformation bei höchst möglicher Produktivität. Die optische Lithografie ist daher die Methode der Wahl für die Strukturierung

Dr. Tilmann Heil, tilmann.heil@zeiss.com, Carl Zeiss SMT GmbH, Rudolf-Eber-Straße 2, 73447 Oberkochen.

aller modernen Chips. Die Schlüsselfunktion des optischen Systems in diesem Prozess ist unmittelbar einsichtig. Ein höheres Auflösungsvermögen der Optik ermöglicht kleinere Strukturen auf dem Wafer und liefert damit dem „Moore'schen Gesetz" folgend signifikante Kostenvorteile.

Das folgende zweite Kapitel beleuchtet die zentrale Rolle der Optik für die Fortsetzung der Halbleiter näher und gibt einen Überblick über die wichtigsten optischen Innovationen der letzten Jahre bis hin zur EUV-Lithografie. Das dritte Kapitel stellt das optische System eines EUV-Wafer-Scanners vor und diskutiert die größten Herausforderungen bei der Entwicklung und Fertigung dieser Optiken. Das vierte Kapitel stellt verschiedene Optionen für neuartige EUV-optische Systeme mit nochmals deutlich gesteigerter Auflösung vor, die eine Fortsetzung der Halbleiter-Roadmap über die nächste Dekade hinaus ermöglichen sollen. Das abschließende Fazit fasst die wesentlichen Botschaften dieses Artikels zusammen.

## II. Optiken für die Fortsetzung der Halbleiter-Roadmap

Ernst Abbe, der die wissenschaftliche Grundlage für die Produkte der Firma Carl Zeiss legte, entwickelte 1873, also fast 100 Jahre vor Gordon Moore, seine Theorie der optischen Abbildung, die auch eine Formulierung für die Auflösungsgrenze eines Objektivs beinhaltet:

$$d = \frac{\lambda}{2n\sin\alpha} = \frac{1}{2}\frac{\lambda}{NA}$$

Die Auflösungsgrenze formulierte er dabei mit $d$ als den minimal noch auflösbaren Abstand zweier Linien eines Gitters, wobei $\lambda$ die Belichtungswellenlänge, $n$ der Brechungsindex, $\alpha$ der halbe objektivseitige Öffnungswinkel und $NA$ die numerische Apertur des Objektivs ist. Ernst Abbes Formulierung wird bis heute in der optischen Nanolithografie der Halbleiterindustrie in nur leicht abgewandelter Form verwendet. Die kleinste mögliche Struktur auf einem Chip, die Critical Dimension CD, ist gegeben durch

$$CD = k_1 \cdot \frac{\lambda}{NA}$$

Der Prozessparameter $k_1$ ist dabei ein Maß für die Schwierigkeit des Herstellungsprozesses. Je kleiner $k_1$, desto schwieriger und teurer der Herstellungsprozess, wobei die Grenze in einem Einzelbelichtungsprozess bei $k_1$ = 0,25 liegt. Dem „Moore'schen Gesetz" folgend verlangt die Halbleiter-Roadmap [3] eine stetige Reduktion der CD. Offensichtlich bieten sich drei Optionen an.

Erste Option ist die Erhöhung der NA des optischen Systems. Dies ist die seitens der Chiphersteller bevorzugte Option, da sowohl Prozesse als auch Technologieplattform weiter verwendet werden können. Zweite Option ist die Reduktion von $k_1$. Sollte eine höhere NA nicht rechtzeitig oder gar nicht mehr zur Verfügung stehen, wird der Faktor $k_1$ gesenkt. Dies erhöht zwar die Komplexität und Kosten der Prozesse, vermeidet aber den mit einer neuen Belichtungswellenlänge verbundenen Sprung zu einer neuen Technologieplattform. Als dritte Option bleibt der Wechsel zu einer kürzeren Belichtungswellenlänge. Diese Option ist mit dem höchsten Aufwand verbunden und erfolgt typischerweise erst, wenn Optionen 1 und 2 ausgereizt sind.

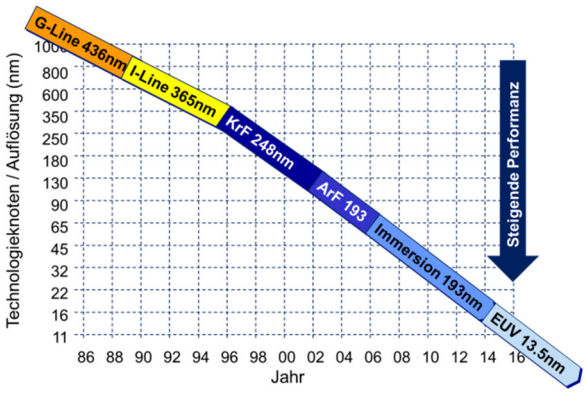



Abbildung 2: Schematische Darstellung, welche Betriebswellenlänge über welche Zeiträume und Technologieknoten in der optischen Lithografie führend eingesetzt wurde. Die zeitliche Steigerung der Auflösung innerhalb einer Betriebswellenlänge wird dabei durch eine Steigerung der NA der optischen Systeme und einer Reduktion von $k_1$ getrieben.

Abbildung 2 zeigt, dass sich die Sequenz aus Steigerung der NA, Absenken von $k_1$ und Sprung zu einer neuen Belichtungswellenlänge in den letzten 25 Jahren mehrfach wiederholt hat. Aktuell steht der Halbleiterindustrie erneut ein Sprung zu einer kürzeren Belichtungswellenlänge bevor. Die heute in Volumenproduktion eingesetzte Immersionstechnologie mit $\lambda$ = 193 nm soll von der EUV-Technologie mit $\lambda$ = 13,5 nm abgelöst werden. Da sich mit Wasser als Immersionsflüssigkeit die NA ökonomisch vernünftig nicht über 1,35 steigern lässt, haben die Chiphersteller in den letzten Jahren den Faktor $k_1$ ihrer Prozesse immer weiter abgesenkt, über Mehrfachbelichtungsverfahren sogar weit unter das für Einzelbelichtungen gültige Limit von 0,25. Die damit verbundene massive Steigerung der Komplexität führte aber zu immer ungünstigeren Kostenstrukturen. Die EUV-Technologie soll nun neben einer Verbesserung der Auflösung eine Rückkehr zu Einzelbelichtungsverfahren und damit eine deutlich reduzierte Prozesskomplexität ermöglichen. Der Übergang zu EUV war ursprünglich deutlich früher geplant. In diesem Jahr verdichten sich positive Konferenzberichte von Chipherstellern über die Produktionsreife von EUV und eine erste Volumenbestellung von ASML EUV-

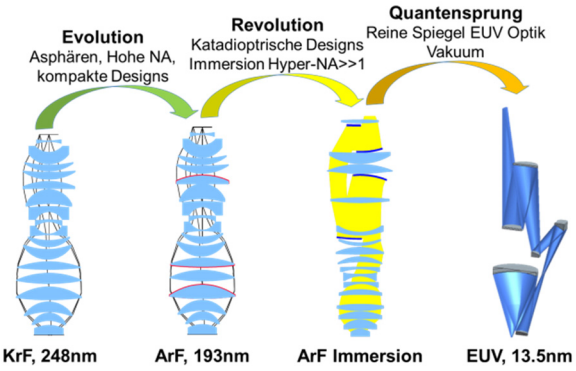



Abbildung 3: Schematische Darstellung einiger optischen Innovationen, die eine Fortsetzung der Halbleiter-Roadmap ermöglicht haben.

Scannern [4] zu einem Gesamtbild, das heute mit einer unmittelbaren Einführung von EUV in die Chip-Volumenfertigung rechnen lässt.

Die in Abbildung 2 skizzierten Innovationsschübe wurden wesentlich durch Fortschritte in optischen Technologien und den verwendeten optischen Konzepten geprägt.

Abbildung 3 zeigt einige Beispiele, wie Innovationen der Firma ZEISS bei optischen Konzepten und Technologien über den Einsatz in den ASML Wafer-Scannern neue Optionen für die Halbleiter-Roadmap eröffnet haben. Der Übergang zur 193 nm-Technologie ging einher mit einem verstärkten Einsatz von Asphären-Technologie, was kompaktere und damit ökonomisch darstellbare Designs hoher NA ermöglichte. Der Übergang zur Immersionstechnologie konnte nur durch neuartige katadioptrische Designs realisiert werden, die herkömmliche Linsenoptiken mit Spiegeloptiken kombinieren und so den Hyper-NA-Bereich bis $NA = 1{,}35$ eröffnen. Der Übergang zur EUV-Technologie erfordert disruptiv neue optische Technologien, die in den folgenden Abschnitten näher erläutert werden sollen. Abbildung 3 verdeutlicht damit, wie sehr optische Systeme Schlüsselelemente der modernen Nanoelektronik sind und eine Fortführung des „Moore'schen Gesetzes" mit ermöglichen.

## III. Besonderheiten und Herausforderungen EUV-optischer Systeme

EUV ermöglicht aufgrund seiner kurzen Betriebswellenlänge nie dagewesene Abbildungsleistungen. Allerdings führt die kurze Wellenlänge von 13,5 nm dazu, dass das optische System mehreren Randbedingungen genügen muss.

Erstens wird EUV-Licht von allen Medien so stark absorbiert, dass das gesamte System in Vakuum betrieben werden muss. Dementsprechend müssen auch alle optischen Elemente inklusive der Maske Spiegel sein.

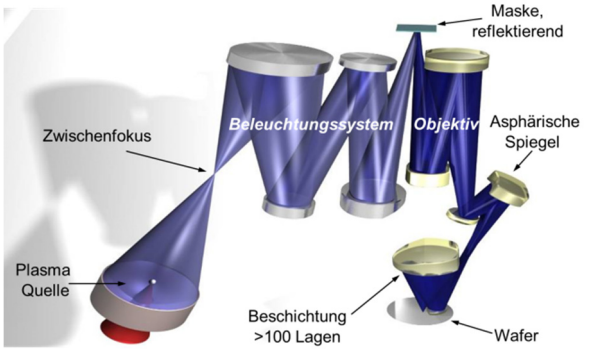



Abbildung 4: Design-Beispiel für ein EUV-optisches System. Das in den ASML Scanner NXE:3300 integrierte optische System Starlith™ 3300 weist als Kenngrößen eine Betriebswellenlänge von 13,5 nm, eine NA von 0,33, eine volle Feldgröße von 26 mm x 33 mm und eine Verkleinerung von -4 x auf.

Zweitens ist der Brechungsindex $n$ aller Materialien bei 13,5 nm nahe 1. Die Beschichtungen der Spiegel müssen daher als Bragg-Reflektoren aus sehr vielen Lagen ausgeführt werden.

Drittens begrenzt die Absorption der zur Beschichtung der Spiegel eingesetzten Materialien die maximal mögliche Reflektion von EUV-Spiegeln auf etwa 70 %. Um also eine möglichst hohe Transmission des EUV-optischen Systems erreichen zu können, muss die Anzahl der eingesetzten Spiegel minimiert werden. Dies ist am effektivsten durch den Einsatz von asphärischen Oberflächen möglich, was allerdings höchste Ansprüche an die Fertigungstechnologien stellt.

Viertens ist die Lichtemission der Plasmaquelle sehr breitbandig. Dies hat zur Folge, dass die Selektion der Betriebswellenlänge des Scanners erst durch die Schichten der Spiegel des Beleuchtungssystems erfolgt. Insbesondere die ersten Spiegel werden daher thermisch sehr stark belastet.

In Summe führen diese Randbedingungen der EUV-Lithografie zu einem völlig neuartigen optischen System, das in Abbildung 4 skizziert ist. Das von einer Plasmaquelle erzeugte EUV-Licht wird durch einen Kollektor auf einen Zwischenfokus gebündelt. Dieser Zwischenfokus definiert den Eingang in das optische System. Dort formt das Beleuchtungssystem das Licht so, dass die in Reflektion betriebene Maske homogen und aus wohldefinierten Winkeln beleuchtet wird. Die Maske erzeugt durch die auf ihr befindlichen Strukturen ein komplexes Muster an Beugungsordnungen. Die Projektionsoptik „fängt" dieses Beugungsmuster auf und bildet so die Maske auf den Wafer ab.

### *A. Herausforderungen bei der Fertigung von EUV-optischen Systemen*

Die große Herausforderung bei der Fertigung von EUV-Systemen liegt darin, einzelne an sich bereits höchst anspruchsvolle Teile und Verfahren zu einem funktionierenden Ganzen zu integrieren. Zudem lassen

sich einige Kernpunkte identifizieren, die im Folgenden besprochen werden.

*1) Die Oberflächen der Spiegel*

EUV-Spiegel erfordern eine sehr genaue Kontrolle der Oberfläche über mehrere Größenordnungen an Raumfrequenzen. Die Messtechnik ist hier von zentraler Bedeutung. Alle Raumfrequenzbereiche müssen mit höchster Präzision vermessen werden, um eine zielführende Fertigung zu ermöglichen und die Abbildungsperformanz des Systems sicher zu stellen.

Langwellige Fehler führen zu Aberrationen der Abbildung. Dies würde zu nicht formtreuer Abbildung der Maskenstrukturen auf den Wafer (CD Uniformity) und zu einer Verschiebung der Strukturen auf dem Wafer (Overlay) führen. Um funktionierende elektronische Schaltkreise erzeugen zu können, sind hier sehr enge Toleranzen vorgegeben, die sich in extreme Anforderungen an die optischen Oberflächen übersetzen. Fehler im mittleren Frequenzbereich führen zu Streulicht, das aber den Wafer noch erreicht. Da dieses Licht nicht zur Abbildung beiträgt, führt dies zu Kontrastverlust und entsprechend schlechterer Kontrolle der Strukturen auf dem Wafer. Dieser Effekt skaliert mit dem Inversen des Quadrats der Betriebswellenlänge. Verglichen mit den Spiegeln in Immersionssystemen ist für EUV-Spiegel diese Anforderung also mehr als 100 mal schärfer. Sehr kurzwellige Fehler der Spiegeloberfläche führen zu Streulicht, das nicht mehr auf dem Wafer landet. Auch diese unerwünschten Lichtverluste sind zu vermeiden.

Das folgende Beispiel soll die extremen Anforderungen an die Oberflächen der EUV-Spiegel verdeutlichen. Die Passen der Spiegeloberflächen des Starlith™ 3300 werden so genau gefertigt, dass die Abweichungen von der Idealform ~ 75 pm Root Mean Square nicht überschreiten. Würde man einen solchen EUV-Spiegel auf die Größe Deutschlands skalieren und die Abweichungen als Berge ansehen, so wäre der höchste Berg im Bundesgebiet niedriger als 0,2 mm.

*2) Die Beschichtung*

EUV-Spiegel werden mit hochgenauen Nano-Multi-Lagen beschichtet. Typischerweise werden dabei mehr als 50 Doppellagen, jeweils bestehend aus Mo und Si, auf den Spiegel aufgebracht. Dabei ist eine sehr genaue Kontrolle der Schichtdicken im Sub-Nanometer-Bereich erforderlich, zum einen, um die Zentralwellenlänge aller Schichten des optischen Systems aufeinander abzustimmen, zum anderen, um dem theoretischen Maximum der Reflektivität möglichst nahe zu kommen. Zudem müssen die Schichten sehr hohen Thermallasten standhalten.

*3) Mechatronik und Positionierung der Spiegel*

EUV-optische Systeme verlangen eine nahezu perfekte Ausrichtung aller Spiegel, da sich der Kipp eines Spiegels aus Idealposition unmittelbar in einen Bildversatz auf dem Wafer übersetzen würde, was gleichbedeutend mit einer Verschiebung der Strukturen des Schaltkreises wäre. Die Overlay-Spezifikation setzt hier sehr enge Grenzen. Typische Werte für das Gesamtsystem liegen in der Größenordnung von ~ 1 nm. Bei einer optischen Weglänge in der Größenordnung von ~ 1 m ergibt sich damit, dass die Spiegel im Sub-Nano-Rad-Bereich positioniert und während des gesamten Belichtungsvorgangs gehalten werden müssen. Um dies sicher zu stellen, sind aufwändige mechatronische Systeme Teil der EUV-optischen Systeme.

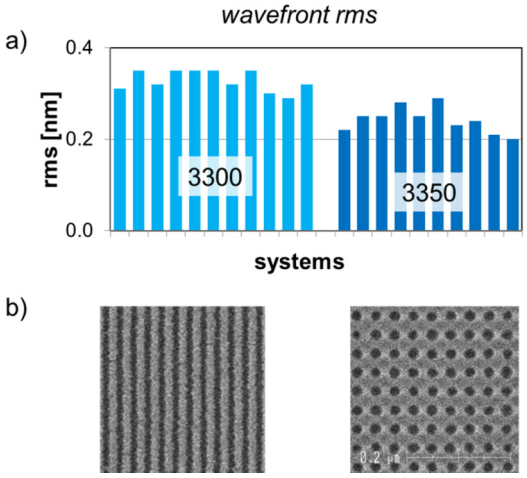



Abbildung 5: a) Wellenfront RMS der bisher gefertigten ZEISS Starlith™ 3300 Serie. Jeder Balken stellt die Wellenfront Performanz eines Systems dar. b) Beispiele von Wafer-Strukturen, gefertigt mit einer ASML NXE:3300. Links: Linien mit 13,5 nm half-pitch-Auflösung, Dosis 31,5 mJ/cm$^2$. Rechts: Kontaktlöcher mit 18 nm half-pitch-Auflösung, Dosis ~ 49 mJ/cm$^2$.

*4) Systemmesstechnik*

Für die Qualifikation von EUV-optischen Systemen ist die Entwicklung speziell abgestimmter Messmaschinen erforderlich. Eine Qualifikation mit sichtbarem Licht ermöglicht es, die Position und Form der Oberflächen im System sicher zu stellen. Um jedoch auch energetische Effekte und Effekte innerhalb der Beschichtungen qualifizieren zu können, ist die Verwendung einer aktinischen Messtechnik notwendig. Dies erfordert den Aufbau einer entsprechenden EUV-tauglichen Interferometrie, was zum Beispiel große Vakuumkammern beinhaltet.

Abbildung 5a) zeigt die Ergebnisse der Wellenfrontmessungen der bisher gefertigten ZEISS Starlith™ 3300 Serie. Es handelt sich um EUV-Projektionsobjektive mit $NA = 0{,}33$, die in den EUV-Scanner NXE:3300 von ASML integriert werden. Abbildung 5 demonstriert eine sehr gute und konsistente Performanz der Wellenfront über die gesamte bisher gefertigt Population. Diese gute Performanz des

Tabelle 1: Auflösung eines EUV-Projektionsobjektivs in Abhängigkeit von der NA.

| NA | 0.25 | … | 0.33 | … | 0.5 | … | 0.6 |
|---|---|---|---|---|---|---|---|
| Auflösung @ k1=0.3 Einzelbelichtung/ nm | 16.2 | … | 12.3 | … | 8.1 | … | 6.8 |

optischen Systems übersetzt sich in bislang unerreichte Abbildungsleistungen auf dem Wafer. Abbildung 5b) zeigt einige Beispiele von Wafern, die mit dem ASML EUV Wafer-Scanner NXE:3300 belichtet wurden [6]. Strukturen von bis zu 13 nm Auflösung in Einzelschussbelichtung verdeutlichen das enorme Potential der EUV-Technologie.

## IV. Hoch-aperturige EUV-Optiken zur Fortsetzung von "Moore's Law"

Der Übergang zur EUV-Technologie ermöglicht der Halbleiterindustrie die Rückkehr zur „NA-Roadmap". Kleinere Strukturen auf dem Chip lassen sich durch eine Steigerung der NA des optischen Systems erzielen. Tabelle 1 zeigt dieses Potential auf. Eine Steigerung der NA eines EUV-*Syst*ems auf ~ 0,5 eröffnet den Bereich von 8 nm Auflösung für Einzelschussbelichtungen.

Dieser Abschnitt diskutiert, welchen Randbedingungen eine solche hoch-aperturige EUV-Optik unterliegt und wie mögliche Lösungen aussehen könnten. Die erste Randbedingung ist die NA selbst. Höhere NA ermöglicht Abbildungen mit größerem Öffnungswinkel. Dies bedeutet größere Winkel und Winkelvariation auf den Oberflächen des optischen Systems. Dabei spielt die EUV-Maske eine besondere Rolle. Abbildung 6a) illustriert die Situation an Maske und Wafer, wie es in der NA-0.33-Optik des NXE:3300 Scanners umgesetzt ist. Über jedem Punkt der Maske ist ein vom Beleuchtungssystem eingehender Lichtkegel definiert. Der Öffnungswinkel dieses Lichtkegels entspricht der beleuchtungsseitigen numerischen Apertur $NA_{Maske}$. Der zur Trennung der ein- und ausgehenden Lichtkegel notwendige Winkel zwischen Maskennormale und dem Hauptstrahl der Lichtkegel wird als „Chief Ray Angle at Object" (CRAO) bezeichnet. In der NXE:3300 beträgt er CRAO = 6°. Der Lichtkegel wird von der Maske reflektiert und tritt in das Projektionsobjektiv ein. Das Objektiv hat eine Vergrößerung von -4 x. Das heißt, dass das 104 mm x 132 mm große Feld der Maske mit einer vierfachen Verkleinerung auf 26 mm x 33 mm auf den Wafer abgebildet wird. Die Konstanz des Lichtleitwerts eines abbildenden Systems hat zur Folge, dass sich dabei die numerische Apertur entsprechend der Verkleinerung vervierfacht. Einer $NA = 0{,}33$ auf Wafer-Ebene entspricht also eine $NA_{Maske} = 0{,}0825$ auf Maskenebene.

Abbildung 6b) zeigt, dass die oben skizzierte Konfiguration für höhere NA nicht übernommen werden

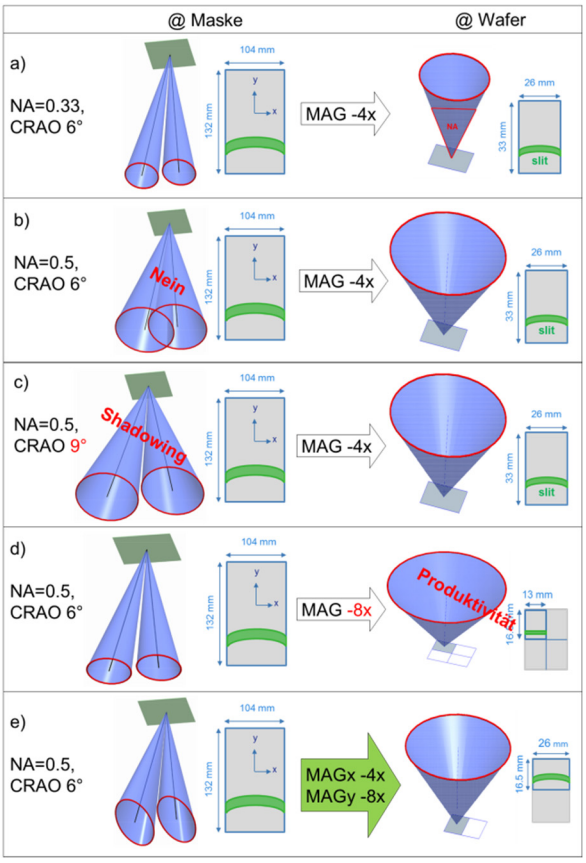



Abbildung 6: Schematische Darstellung verschiedener Optionen zur Realisierung einer hoch-aperturigen EUV-Optik im Vergleich zu der realisierten NA-0.33-Konfiguration.

kann. $NA = 0{,}5$ führt bei einem -4 x Objektiv zu einer entsprechenden Vergrößerung von $NA_{Maske}$ auf 0,125. Der halbe Öffnungswinkel des auf der Maske ein- und ausgehenden Lichtkegels übersteigt den CRAO = 6°. Die Lichtkegel schneiden sich. Ein abschattungsfreies System ist so nicht möglich.

Abbildung 6c) zeigt die naheliegende Option, die durch die hohe NA weit geöffneten Lichtkegel durch eine Erhöhung des CRAO auf 9° wieder zu trennen. Leider ist dies nicht möglich. Abschattungseffekte der Strukturen auf der Maske, das sogenannte „Shadowing", verhindern dies: da die Maske aus Multi-Lagen besteht, auf denen Absorberstrukturen aufgebracht sind, erfolgt die Reflektion bildlich gesprochen aus den Multi-Lagen heraus. Die durch die Absorber entstehende Abschattung wird damit abhängig vom Einfallswinkel. Je steiler der Winkel, desto stärker die Abschattung. In Richtung des Scans ist der Lichtkegel symmetrisch um 0° verteilt, was die Winkel auf ± 7,2° = arcsin(0,5/4) limitiert, so dass die Abschattungseffekte auch bei der höheren NA noch unproblematisch sind. Senkrecht dazu wäre der Lichtkegel um 9° verteilt, was einem Winkelbereich von 1,8° bis 16,2° entspricht. Es entstehen Asymmetrien in der Intensität der bildgebenden Beugungsordnungen, die

den Kontrast des Bildes auf dem Wafer reduzieren. In der in Abbildung 6c) gezeigten Konfiguration ist dieser Kontrastverlust so stark, dass die Bildqualität auf dem Wafer unzureichend ist. Da es sich hierbei um einen fundamentalen Effekt handelt, bleibt als Ausweg nur, $NA_{Maske}$ zu reduzieren. Dies lässt sich im optischen System beispielsweise durch eine stärkere Verkleinerung der Abbildung erreichen.

Abbildung 6d) zeigt die Konfiguration eines NA 0.5 Objektivs mit einer Vergrößerung von -8 x. Der CRAO kann bei 6° gehalten werden. Die Öffnung der Lichtkegel ist klein, so dass die Abschattungseffekte unproblematisch sind. Allerdings führt die Vergrößerung von -8 x dazu, dass sich das Feld auf dem Wafer auf ein Viertel, d. h. auf 13 mm x 16,5 mm verkleinert. Die resultierenden Einbrüche in der Produktivität des Scanners sind so groß, dass auch diese Konfiguration nicht attraktiv ist. Die denkbare Alternative, bei einer Vergrößerung des Objektivs von -8 x die Maskengröße zu vervierfachen, um die Chipgröße auf dem Wafer zu halten, ist aufgrund der notwendigen völlig neuen Maskeninfrastruktur ebenfalls unattraktiv.

Abbildung 6e) zeigt den Ausweg aus diesem Dilemma. Ein anamorphotisches Design mit richtungsabhängiger Vergrößerung ermöglicht es durch eine Vergrößerung von -8 x in Faltungsrichtung das Shadowing zu kontrollieren und durch eine Vergrößerung von -4 x senkrecht dazu die Feldbreite auf 26mm zu belassen. Ergebnis auf Wafer-Ebene ist ein Halbfeld von 26 mm x 16,5 mm Größe. Nur diese Konfiguration ermöglicht sowohl eine Abbildung mit maximaler NA und Auflösung als auch eine befriedigende Produktivität. Diese für die Lithografie neuartige optische Lösung wird in [6] näher beschrieben.

Die mit der hohen NA inhärent verbundene hohe Winkelbelastung führt aber nicht nur auf der Maske zu Schwierigkeiten. Auch innerhalb des Objektivs treten auf den Spiegeln hohe Winkel und hohe Winkelbereiche auf. Eine reine Skalierung des NA-0.33-Systems auf ein NA-0.5-System würde die Akzeptanzbereiche vorhandener Schichtsysteme auf den Spiegeln deutlich überschreiten und zu nicht akzeptablen Transmissionsverlusten führen. Ein Treiber für die hohe Winkelbelastung in Spiegelsystemen ist die Notwendigkeit, den Lichtkegel um die Spiegel herum zu führen. Der Lichtkegel muss also von der optischen Achse weg gekippt werden. Diese Kippwinkel führen zu zusätzlicher Winkelbelastung auf den Oberflächen. Die Lösung für dieses Problem besteht darin, den Lichtkegel nicht an den Spiegeln vorbei, sondern durch die Spiegel hindurch zu führen. Dazu müssen Löcher in die Spiegel gebohrt werden, was zu einer teilweisen Abschattung, einer sogenannten Obskuration, führt. Eine zentrale Obskuration in der Pupille des Objektivs kann bezüglich der Abbildungsperformanz toleriert werden, wenn sie eine Größe von etwa 20 % Radius nicht überschreitet. Durch diese Maßnahme lässt sich die Winkelbelastung der Spiegel trotz der signifikant höheren NA so weit minimieren, dass die Transmission des NA-0.5-Systems die des NA-0.33-Systems sogar übersteigt. Dies wird eine entsprechend höhere Produktivität des NA-0.5-Scanners ermöglichen.

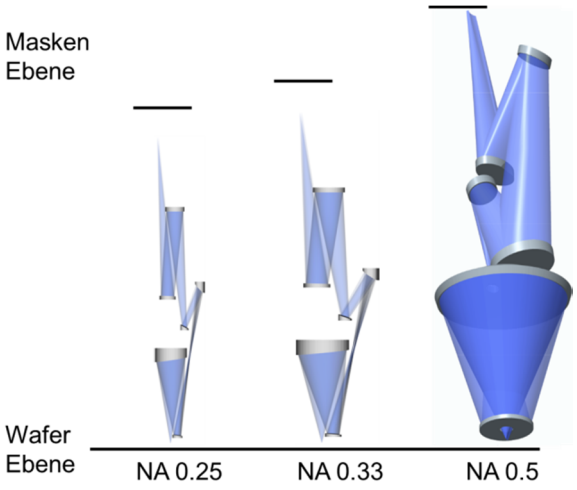



Abbildung 7: Design-Beispiele für EUV-Projektionsobjektive für NA 0.25, NA 0.33 und NA 0.5.

Abbildung 7 zeigt Design-Beispiele für optische Systeme mit NA 0.25, NA 0.33 und NA 0.5. Während das NA-0.33-System sich noch als eine Skalierung des NA-0.25-Systems darstellen lässt, weist das NA 0.5 deutliche Änderungen auf. Auffallend ist die starke Zunahme der Größe. Auch die zentrale Obskuration der letzten Spiegel ist deutlich sichtbar. Eine weitere Besonderheit ist der sehr große letzte Spiegel. Die Größe dieses Spiegels wird direkt durch die NA des optischen Systems getrieben: die direkt mit der NA verbundenen Öffnungswinkel der auf den Wafer auftreffenden Lichtkegel bestimmen über den Abstand zwischen Spiegel und Wafer direkt die minimale Größe des Spiegels. Auch diese größer werdenden optischen Oberflächen müssen deutlich engere Spezifikationen bei weiter steigender Asphärizität erfüllen, weil mit den zukünftigen Technologieknoten auch die Anforderungen an Abbildungsperformanz, Kontrast und Streulicht relativ zu der NA-0.33-Optik steigen.

In Summe stellt das optische System NA 0.5 große Herausforderungen für die Optik-Herstellung und -Technologie dar. Andererseits zeigen sich keine fundamentalen Limits. Daher kann man davon ausgehen, dass hoch-aperturige EUV-Lithografie-Systeme die Fortführung der Halbleiter-Roadmap über die nächste Dekade hinaus ermöglichen werden.

## V. FAZIT

EUV-Lithografie ist die Strukturierungstechnologie zukünftiger Chip-Generationen und ermöglicht eine Fortsetzung der Halbleiter-Roadmap bis unter 9 nm Auflösung hinaus. Innovationen und Fortschritte der optischen Systeme sind hierbei von entscheidender Bedeutung. Die Industrialisierung der EUV-

Technologie zeigt mittlerweile große Fortschritte. Die ersten Scanner NXE:3300 von ASML mit integrierter ZEISS EUV-Optik sind bei den führenden Chipherstellern im Einsatz, und eine große Volumenbestellung für weitere EUV-Scanner ist erfolgt.

## DANKSAGUNG

Der Autor dankt dem Bundesministerium für Bildung und Forschung für Förderung in den Projekten 13N8088, 13N8837, 13N9112 und 13N10567 im Rahmen der MEDEA+/CATRENE-Programme sowie den EUV-Teams bei ZEISS, ASML und unseren Partnern. Insbesondere danke ich Bernhard Kneer, Martin Lowisch und Sascha Migura für Bildmaterial und wertvolle Diskussionen.

## LITERATURVERZEICHNIS

[1] G.E. Moore, “Cramming more components on electronic circuits”, *Electronics Volume 38, No 8*, 1965.

[2] Gartner Dataquest (April 2015).

[3] International Technology Roadmap for Semiconductors, 2013 Edition, *www.itrs.net*.

[4] ASML Press Release, “ASML reaches agreement for delivery of minimum of 15 EUV lithography systems”, April 2015, *www.asml.com*.

[5] A. Pirati et al., “Performance overview and outlook of EUV lithography systems”, *Proc. SPIE 9422, Extreme Ultraviolet (EUV) Lithography VI, 94221P* (13 March 2015).

[6] B. Kneer et al., “EUV lithography optics for sub-9nm resolution”, *Proc. SPIE 9422, Extreme Ultraviolet (EUV) Lithography VI, 94221G* (16 March 2015).

Dr. Tilmann Heil ist Director Lead System Engineering bei der Carl Zeiss SMT GmbH. Er studierte Physik an der TU Darmstadt und der KTH Stockholm und promovierte 2001 an der TU Darmstadt. Nach einem Post Doc bei ATR Kyoto begann er als Wissenschaftlicher Mitarbeiter bei der Carl Zeiss SMT GmbH und arbeitet seither in System Engineering, Technischem Marketing, Forschungsförderung und Konzeptentwicklung.

# CMOS Image Sensors

Harald Neubauer

***Abstract*—CMOS Image Sensors (CIS) are used in a lot of (mobile) devices. An overview of current CMOS technology enhancements for image sensors is summarized and challenges for the ASIC implementations are described. Methods to optimize the design of analog to digital converters (ADC) are shown.**

***Index Terms*—CMOS Image Sensors, ASIC design, ADC.**

## I. Introduction

CMOS Image Sensors (CIS) are used in many applications, mainly in mobile devices (see fig. 1). CMOS image sensors are dominating the market for imaging devices, where the former used CCD sensors have about 6.5 % remaining market share. The change of the last 10 years from CCD towards CIS devices is mostly driven by cost and features:

- Signal processing on the same device
- Similar image quality compared to CCD
- Higher speed
- Lower cost
- Advantages from general technology scaling

Imagers for mobile devices are driven by a race for more pixels. More pixels lead to smaller pixels due to optical boundary conditions (mostly the height of the lens stack). Currently pixels around 1 µm pitch are used for mobile devices. The market segmentation of CIS devices in 2014 is depicted in fig. 2. In total 3.8 B devices were built in 2014 with a revenue of 10.2 B $. The annual growth rate is around 10 %.


Figure 1: Number of CIS devices for different applications [1].

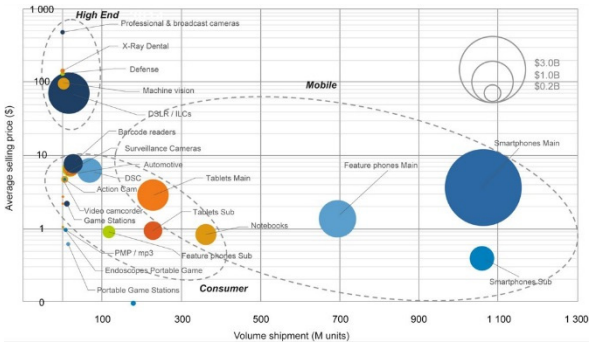



Figure 2: Number and average selling price of CIS devices for different applications [2].

## II. Architecture of CMOS Image Sensors

An image sensor usually is built of a pixel array and readout electronics. The pixels are connected by row and column lines. The row connections are used to select one row, where each of the pixels is connected in the selected row to the column line, as shown in fig. 3. The readout is done by reading the sensor out row by row. The information of the pixels is processed with the readout circuitry. This usually has a circuit for noise reduction through correlated double sampling. Often the value is already digitized on the sensor with an analog to digital converter.

Harald Neubauer, harald.neubauer@iis.fraunhofer.de, is with Fraunhofer Institut IIS, Am Wolfsmantel 33, 91058 Erlangen.

### *A. Rolling and Global Shutter*

Usually, imagers are read out in a rolling shutter configuration. That means that the pixel is reset after it was read out. By this the maximum exposure can be reached and this readout scheme is very simple. As a drawback this creates unwanted image artifacts with moving objects. A rotating fan is often used as a target to show the effect (fig. 4).

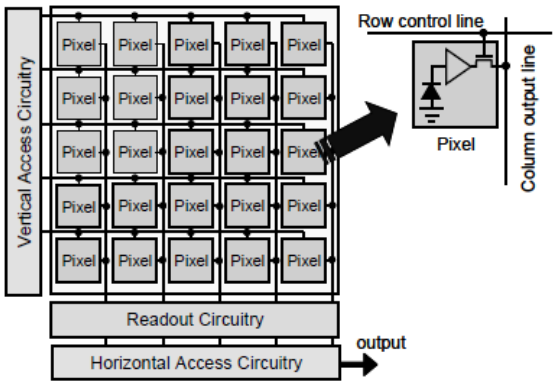



Figure 3: Architecure of a CMOS image sensor.

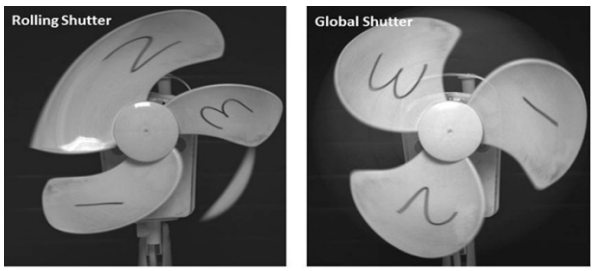



Figure 4: Moving fan imaged with rolling shutter [4].

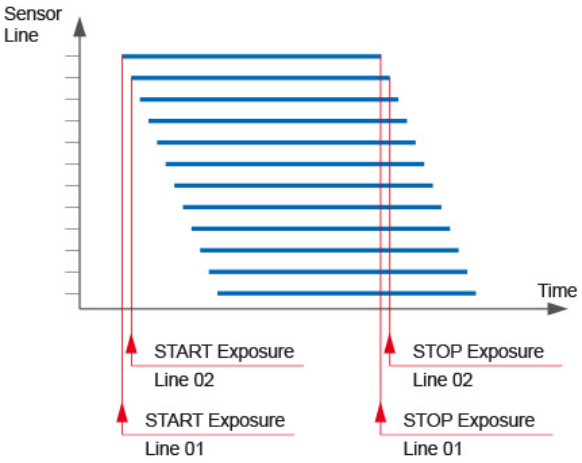



Figure 5: Rolling readout [5].

Although all rows have the same exposure length, the start and end of the exposure is shifted from one row to the next by the readout speed. The fan moves while the image is read out. This is further shown in fig. 5.

To circumvent these artifacts, the exposure interval needs to be the same for all pixels. The exposure start and stop become global signals, so every pixel is exposed in the same interval. During this exposure the previous acquired value is read out of an additional memory in a sequential manner (fig. 5).

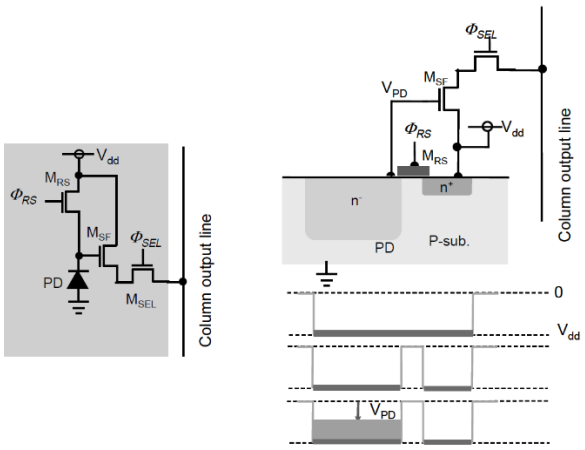



Figure 6: Basic 3 transistor pixel [3].

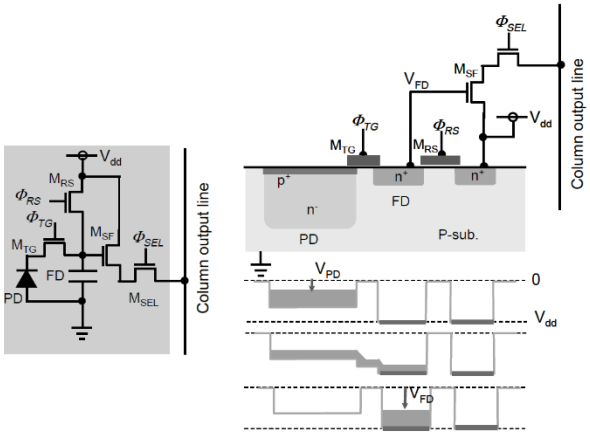



Figure 7: “Pinned” 4 transistor pixel [3].

## B. Pixel Circuits

In image sensors different pixel electronics are used. The pixels are usually named according to the average number of transistors per pixel.

### 1) 3T Pixel

The basic cell is shown in fig. 6. It contains 3 transistors per pixel (reset, source follower, select). For the exposure, first the pixel is reset, so that the photodiode (usually an n diffusion in the p substrate) is charged to a positive voltage. To compensate for the threshold voltage of the reset transistor, either the supply voltage of the pixel is lowered, or the reset voltage is around one threshold voltage above the pixel supply voltage. After the reset the incoming light discharges the photodiode to a lower voltage. When the pixel is read out the row select signal connects the source-follower to the column line. The first element at the end of the column line is a current sink.

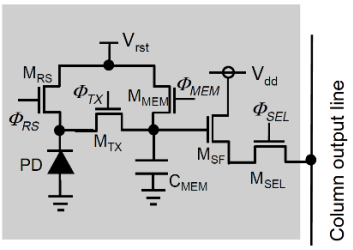



Figure 8: 5T Global shutter pixel [3].

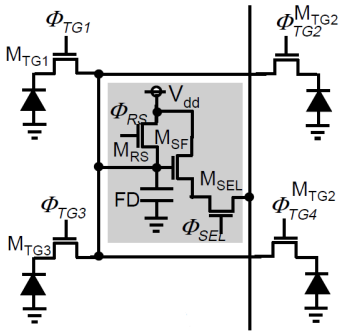



Figure 9: 1.75T (shared) pixel [3].

*2) 4T (pinned) Pixel*

In order to increase pixel performance a modification of this pixel is commonly used as shown in fig. 7. The modified pixel enables reduction of noise and dark current. The pixel separates the photodiode from the readout node. By this a conversion gain between these two nodes can be enabled. Several process steps need to be added to a conventional process, so these pixels cannot be used in standard processes. The operation is different from the described 3T-pixel.

The readout node called floating diffusion (FD) is reset through the reset transistor. The value of this reset is sampled in the readout circuit at the column line. Then the transfer of the charge is done by switching the transfer transistor to the ON state. The charge is fully transferred out of the photodiode to the floating diffusion. Also this value is acquired by the readout circuit. The difference of these two readings is generated which eliminates the reset noise and the source follower offset. As the capacity of the floating diffusion can be chosen smaller than the capacity of the photodiode, a conversion gain can be established. The voltage change on the floating diffusion is higher than on the photodiode.

*3) 5T (global shutter) Pixel*

To enable global shutter, for some applications (mainly in machine vision) a storage node and a global reset is introduced in the pixel (fig. 8). The pixel in fig. 8 is based on the basic pixel of fig. 6 with the addition of a global reset for the photodiode and a pixel storage ($C_{MEM}$). All pixels are reset at the same time and integration of light is stopped by transferring the information to the capacitor at the same time. As 2 additional transistors are needed, this pixel is usually only used if features are extracted for quality control, or fast moving objects are recorded. Some global shutter pixels work similarly to the 4T (pinned) pixels with full charge transfer, although further additional process steps are needed.

*4) 1.75T (shared) Pixel*

For small pixels (< 3 µm) the readout transistors are shared between different pixels to reduce the area consumption (fig. 9). Configurations of 2 or 4 pixels are common. Only half of the column lines are needed and they have twice the pixel pitch, which reduces the area consumption for the readout electronics.

*C. Readout Circuits*

Column readout circuits process the pixel information and feed them off the sensor. This can be a simple multiplexer or an analog to digital converter (ADC) with a chain of image processing algorithms. There are several possibilities for the implementation of the ADC.

- Sensor ADC. The column signals are multiplexed and digitized in one ADC (or only a few). This concept requires high throughput ADCs and high analog bandwidth, which results in a higher noise bandwidth.
- Pixel ADC. Only suitable for pixels, where an ADC can be fit into without losing too much photodiode area. This is usually used with pixel pitches over 50 µm.
- Column ADC. One ADC is used per column. This is the most frequently used architecture, as it permits flexible choice of different ADC types.

For the ADC several requirements have to be met. For a column ADC this could result in the following specifications:

- Usually a sensor has a frame rate from 25 up to several hundred frames per second (fps). The speed of the ADC is the frame rate multiplied by the number of rows. For a 2 MP sensor of 1920 x 1080 pixels, this gives 1080 · 25 fps = 25 kS/s.
- The ADC-resolution is often around 10 bit.
- Spacing of the ADC is around 2 x pixel pitch (if 4-shared pixels are used) or 4 x pixel pitch if the ADC are distributed to the top and bottom of the sensor. Therefore spacing in the example is 4 µm or 8 µm.
- Power consumption is also a challenge as in the example 540 ADCs are needed.

Several architectures for the column level ADC are suitable.

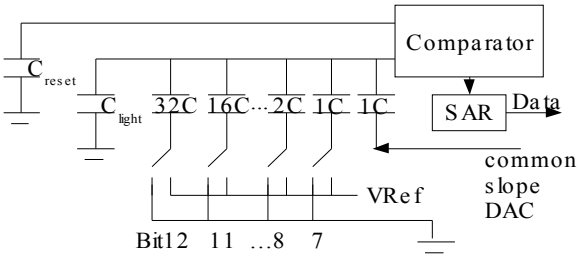



Figure 10: Hybrid SAR slope converter.

Table 1: Comparison of SAR, slope and hybrid converter.

| Type | DAC Elements | Cycles |
|---|---|---|
| 12 bit SAR | 4096 | 12 |
| 12 bit slope | 0 | 4096 |
| 6 SAR – 6 slope | 64 | 70 |

*1) Single Slope Ramp ADC*

This is the most popular architecture as there is very little circuitry per column needed. Only a comparator and memory is needed per column. A global common ramp for all ADCs is implemented once on the sensor. However, this concept becomes challenging in mega-pixel applications where row lines increase which makes high clock rates necessary.

*2) Successive Approximation Converter (SAR)*

This type of converter is very power efficient. Unfortunately a lot of implementations require considerable area as the needed digital to analog converter is built using a capacitor or resistor array. A new type of converter can be built out of the combination of a slope and a SAR converter. The upper half of the resolution is built out of a SAR and the lower half is done using slope conversion (fig. 10). This implementation mixes the two concepts. It requires less hardware than the SAR and less clock cycles than the slope converter. A comparison is shown in table 1.

*3) Pixel Level Preconversion*

If the pixels are big enough and the application demands for high dynamic range a preconversion in the pixels can be carried out. With the preconversion the first most significant bits are already digitized in the pixel extending the dynamic range of the remaining system.

An example is a sensor working in an x-ray application with pixels of 200 µm pitch. A schematic is shown in figure 11. The first 2 bits are digitized in the pixel and the residuum is converted at 12 bit level in the system. In the pixel an operational amplifier

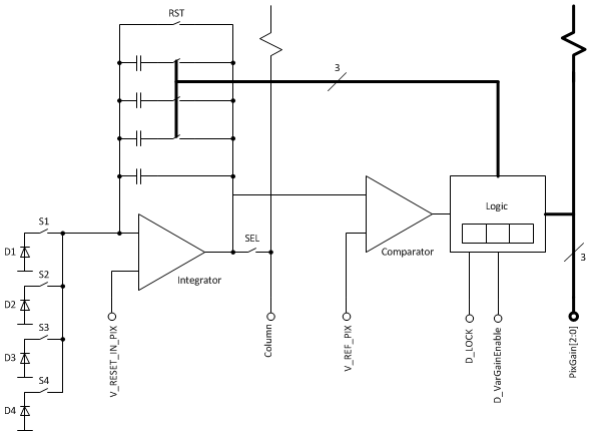



Figure 11: Pixel preconversion.

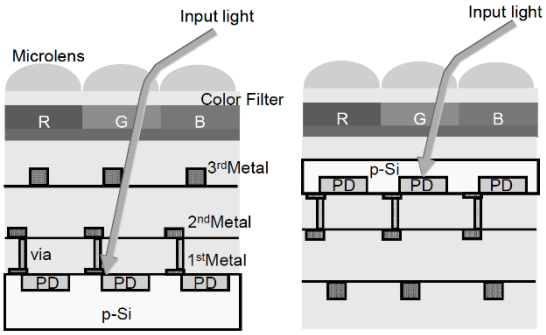



Figure 12: Backside illumination [3].

charges the switchable capacitors, which are binary scaled. A comparator decides if the threshold level is reached and then a small logic block switches to the next capacitor. The digital information is read out by 2 digital lines. The residuum is transferred as usual by an analog column line to the read circuit to digitize the required resolution.

## III. Trends in CMOS Image Sensors

There have been a lot of improvements from the first CMOS image sensors to current state of the art. Around 10 years ago imaging was dominated by charge coupled device imagers (CCD). These need a specific technology and usually higher voltages than CMOS. The CMOS image sensors enabled complex digital processing on the same device and used mainstream and cheap processes. To improve the CMOS performance, a lot of techniques already used in CCD were introduced to CMOS image sensors like the complete charge transfer or the backside illumination. The conception of backside illumination (BSI) is depicted in figure 12.

With BSI a much greater angle of incident light can be accepted. Further the use of more metals in the CMOS process is possible resulting in a higher density of the digital circuits. After processing the wafer is

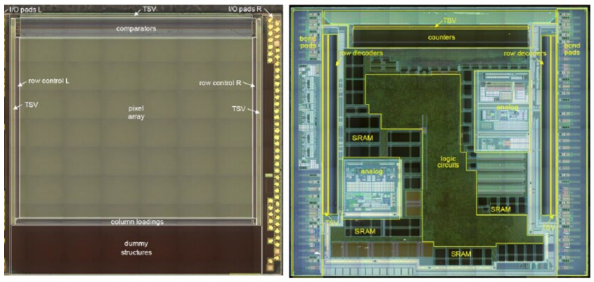



Figure 13: 3D stacked image sensor [6].

attached to a handling structure and thinned down to a few µm. The color filters and microlenses are put on the backside. The electrical connection is realized with through silicon vias.

Recently sensor systems were developed, which separate the acquisition of light and the processing. A sensor die with little circuitry is attached to a processing die where digitization and digital processing is done. In figure 13 the sensor is realized in a 90 nm process, while the processing wafer is manufactured in 65 nm. Together an 8 megapixel image sensor with a pixel pitch of 1.12 µm is formed.

## IV. Summary

CMOS image sensors are mixed signal circuits which process several thousand input signals in parallel. The analog to digital conversion is a challenging task in the implementation of such systems. Several enhancements to CMOS processes have been introduced to make these processes more suitable for the implementation of image sensors. New techniques for the sensors as backside illumination and stacked systems using different processes have been recently introduced.

## References

[1] Yole Developpment: http://www.yole.fr/iso_album/illus_cisapplications_yole_oct2014.jpg.

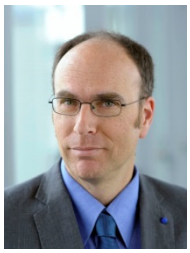
[2] Yole Developpment: http://www.yole.fr/iso_album/illus_cismarket_landscape_yole_jan2015.jpg.

[3] Jun Ohta, „Smart CMOS Image Sensors and Applications“, CRC Press 2007 ISBN: 978-0-8493-3681-2.

[4] Andor:http://www.andor.com/learning-academy/rolling-and-global-shutter-exposure-flexibility.

[5] Basler http://www.baslerweb.com/de/vision-campus/kamera-auswahl.

[6] Chipworks:http://image-sensors-world.blogspot.de/2015/06/iftle-on-stacking-technology-progress.html.



Harald Neubauer received the Dipl.-Ing. in Electrical Engineering in 1997 from Friedrich-Alexander University Erlangen-Nürnberg. He heads the group CMOS Image Sensors at the Fraunhofer Institute for Integrated Circuits in Erlangen.

# Entwurf und Inbetriebnahme einer PLL in 0,35 µm CMOS-Technologie

Christian Eschenbach, Bernd Vettermann, Jürgen Giehl

***Zusammenfassung*—Die „Phase-Locked Loop“ ist ein elektronischer Regelkreis, der zur Erzeugung stabiler Frequenzen eingesetzt wird. Dieser Schaltungstyp stellt eine wichtige Komponente in frequenzabhängigen Systemen dar, die für eine korrekte Funktionsweise stabile Betriebsfrequenzen benötigen. Anhand eines Phasenvergleichs des Ausgangssignals der PLL-Schaltung gegenüber einer vorgegebenen Referenz kann durch eine Schleifenschaltung eine fortwährende Synchronisierung der Ausgangsfrequenz auf die Referenzfrequenz stattfinden. Somit wird die ausgegebene Frequenz in Abhängigkeit von der Eingangsfrequenz stabil gehalten. In dieser Arbeit wird die Entwicklung und die Inbetriebnahme einer Phasenregelschleife mit der Erweiterung durch einen Frequenzteiler in 0,35 µm CMOS-Technologie präsentiert. Die PLL ist auf eine Signalfrequenz von 80 MHz bei einer Referenzfrequenz von 10 MHz ausgelegt. Sie erlaubt eine Frequenzabstufung über 3 Bit bei einer Einschwingzeit von 3,8 µs.**

***Schlüsselwörter*—PLL, Chipentwurf analog, Chipentwurf digital, Platinenentwurf.**

## I. Einleitung

In der integrierten Schaltungstechnik ist die Phasenregelschleife aufgrund ihrer frequenzregulierenden Funktionsweise ein Schaltungstyp, der nicht mehr wegzudenken ist. Die hauptsächlichen Anwendungsgebiete betreffen Frequenzmodulation/-demodulation, Frequenzstabilisierung von Eingangssignalen für andere Systemelemente und Frequenzsynthese [1]-[3].

In ihrer Ausführungsart gibt es unterschiedliche Varianten, was sowohl den grundlegenden Gesamtaufbau, als auch einzelne Komponenten der PLL-Schaltung betreffen. PLLs können vollständig mit analogen oder mit digitalen Schaltungskomponenten entwickelt werden, wobei auch Mischformen möglich sind. Mit digitalen Signalprozessoren besteht prinzipiell auch die Möglichkeit, eine PLL mit Hilfe von Software (z. B. VHDL) zu realisieren. Was die feinere Struktur der Phasenregelschleife angeht, können verschiedene Varianten der Phasendetektion sowie unterschiedliche Filterstrukturen eingesetzt werden, die sich maßgeblich auf die Verhaltensweise des Phasenregelkreises auswirken.

Christian Eschenbach, christian.eschenbach1@hs-mannheim.de, Jürgen Giehl, j.giehl@hs-mannheim.de, Bernd Vettermann, b.vettermann@hs-mannheim.de, Hochschule Mannheim, Paul-Wittsack-Str. 10, 68163 Mannheim.

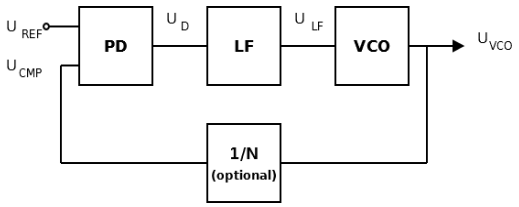



Abbildung 1: PLL-Blockschaltbild.

Im Falle der Phase-Locked Loop, über die in diesem Beitrag berichtet wird, handelt es sich um eine Mischform, die aufgrund des in digitaler Logik arbeitenden Phasendetektors den „Digital PLLs“ zuzuordnen ist. Zur Phasendetektion wird ein sogenannter Phasen-Frequenz-Detektor verwendet, der gegenüber einem herkömmlichen XOR-Phasendetektor Vorteile im Zieh- sowie im Fangbereich vorweisen kann.

## II. Stand der Technik und anfängliche Spezifikation

Die Funktionsweise einer Phase-Locked Loop ist relativ unkompliziert. Sie ist in ihrer Grundkonfiguration mit nur drei notwendigen Komponenten ausgestattet (Abb. 1). Das Hauptmerkmal einer PLL ist eine Rückkopplung ihres Ausgangs auf eine Art Komparator, der Phasendetektor genannt wird. In diese Vergleicherschaltung wird ein Referenzsignal mit einer bestimmten Frequenz gespeist, gegen welches das Ausgangssignal auf Phasendifferenzen verglichen wird. Es ergibt sich wiederum ein hochfrequentes Signal, das abhängig von diesen Differenzen schwingt und als Steuersignal eines spannungsgesteuerten Oszillators fungieren soll. Damit dieser einen von der Höhe seiner Eingangsspannung abhängigen PLL-Ausgangstakt generieren kann, müssen die hochfrequenten Anteile des Komparatorsignals über einen Tiefpass herausgefiltert werden, bevor der VCO (spannungsgesteuerter Oszillator, voltage-controlled oscillator) sein Steuersignal erhält. Durch die Rückkopplung des Ausgangstakts auf den Komparator für einen Phasenvergleich kann somit eine Nachregelung und gegenüber Tempe-

Tabelle 1: Spezifikation der PLL.

| Parameter | Symbol | Spezifikation |
|---|---|---|
| Versorgungsspannung | $V_{DD}$ | 3,3 V |
| Referenzfrequenz | $f_{REF}$ | 10 MHz |
| Max. Ausgangsfrequenz | $f_{max}$ | 80 MHz |
| Teilerfaktor | $N$ | 2-8 |
| Ziehzeit | $T_P$ | 1 µs |
| Fangzeit | $T_L$ | 2 µs |
| Einschwingzeit | $T_P + T_L$ | 3 µs |
| Frequenzabweichung (1σ) | $\Delta f_{CMP}$ | < 5 kHz |
| Jitter bei 10 MHz | | < 100 ps |

ratur oder sonstigen Einflüssen zuverlässige Stabilisierung der Ausgangsfrequenz erreicht werden.

Ziel dieses Projektes war es, eine Phasenregelschaltung für zukünftige Designs bereitzustellen. Die PLL soll die Referenzfrequenz stabilisieren und weitere Frequenzen synthetisieren. Dazu wurde die PLL-Grundschaltung neben den 3 Hauptkomponenten Phasendetektor, Schleifenfilter und spannungsgesteuerter Oszillator um einen Frequenzteiler mit den Teilungsfaktoren 2 bis 8 erweitert (Abb. 1). Zur Verbesserung der Frequenzeigenschaften wurde ein Phasen-Frequenz-Detektor (PFD) verwendet.

Der PFD ermöglicht einen genaueren Phasenvergleich des Ausgangssignals $U_{CMP}$ der Frequenzteilerstufe gegenüber dem Referenzsignal $U_{REF}$, indem er anstatt eines logischen „HIGH“- und „LOW“-Signals (wie es bei einem XOR-Detektor der Fall ist) ein Tristate-Signal generiert, das 3 Regulierungsrichtungen der Frequenz ermöglicht: Erhöhen, Verringern oder Halten. Dieses Tristate-Signal, in dem sich die Phasendifferenzen zwischen PLL-Ein- und Ausgang widerspiegeln, schwingt mit hoher Frequenz und muss somit über ein Tiefpassfilter gemittelt werden, damit sich eine niederfrequente Steuerspannung für den spannungsgesteuerten Oszillator ergibt. Die Höhe dieser Steuerspannung bestimmt die Ausgabefrequenz des durch den VCO generierten Taktsignals $U_{VCO}$. Der VCO erzeugt die Eingangsfrequenz des frequenzbestimmenden Elements der PLL-Schaltung, also des Frequenzteilers. Das in der Frequenz um den Faktor $N$ heruntergeteilte Signal $U_{CMP}$ wird auf den Eingang des Phasendetektors rückgekoppelt, wodurch die Schleifenwirkung der PLL entsteht.

Der angestrebte Betriebszustand einer PLL, also der „eingerastete Zustand“, wird dann erreicht, wenn Rückkoppelfrequenz $f_{CMP}$ und Referenzfrequenz $f_{REF}$ sich gleichen:

$$f_{CMP} \cong f_{REF}$$

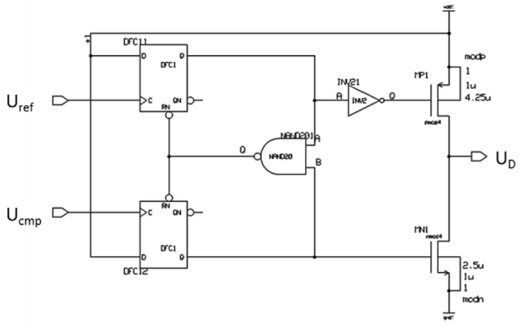



Abbildung 2: Phasen-Frequenz-Detektor.

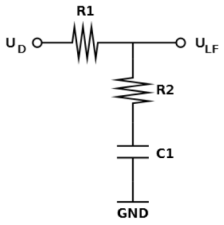



Abbildung 3: PLL-Loop-Filter.

In Kombination mit einem Frequenzteiler des Teilerfaktors $N = 8$ ergibt sich damit die VCO-Frequenz $f_{VCO}$ zu

$$f_{VCO} = N \cdot f_{ref}.$$

Bei einer Referenzfrequenz von 10 MHz und einem Teilerfaktor $N = 8$ liefert der VCO eine Frequenz von 80 MHz.

## III. Entwurf und Dimensionierung

Phase-Locked Loops können mit unterschiedlicher Motivation, die vor allem vom Anwendungszweck abhängig ist, dimensioniert werden. Hier wurde ein eher genereller Ansatz verfolgt, da es keine genauen Einschränkungen oder Vorgaben gab, was bspw. Aufstartgeschwindigkeit oder Rauschen der Schaltung angeht. Deswegen wurde ein Ansatz speziell im Hinblick auf eine kurze Einschwingzeit gewählt, welcher nach Vorbild einer beispielhaften Entwicklungsrichtlinie für Phasenregelschaltungen [1, S. 126-134] durchgeführt wurde. Als Ziehzeit (pull-in time) wurde $T_P = 1$ µs und als Fangzeit (lock-in time) wurde $T_L = 2$ µs festgelegt. Tabelle 1 zeigt die angestrebte Spezifikation der PLL. Als Referenz sollte ein 10 MHz-Quarzoszillator dienen, der bereits in einer früheren Arbeit [4] umgesetzt wurde. Die maximale Ausgangsfrequenz sollte 80 MHz betragen. Mit Teilerfaktoren 2-8 sollten 7 weitere Frequenzen zwischen 80 MHz und 10 MHz ermöglicht werden.

Abbildung 2 zeigt den verwendeten Phasen-Frequenz-Detektor. Die beiden D-Flip-Flops werden mit einem HIGH-Signal kontinuierlich durchgeschaltet. Abhängig von den logischen Pegeln der Eingangssignale können sich zur Übermittlung an die PMOS-

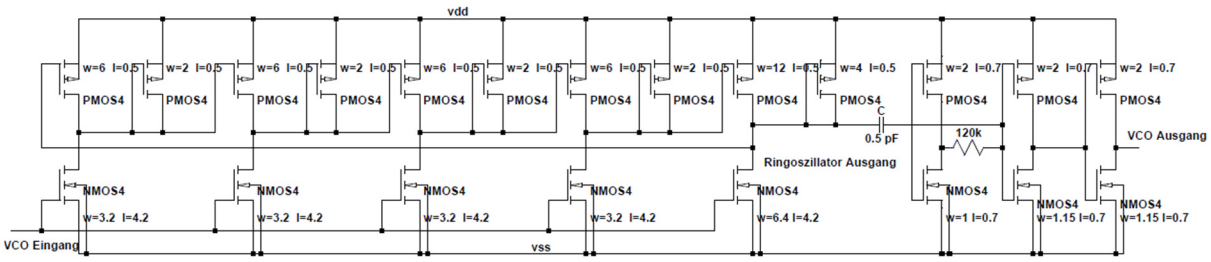



Abbildung 4: Schaltbild des VCO mit Dimensionierung (Einheit µm).

und NMOS-Transistoren 3 Zustände einstellen: 1/0, 0/1 und 0/0. Der vierte Zustand 1/1 wird durch eine Aktivierung des Resets über ein NAND-Gatter unterbunden. Die Zustände bestimmen das Schaltverhalten der Transistoren. 1/0 bewirkt ein Durchschalten des PMOS-Transistors, wodurch das PFD-Ausgangssignal $U_D$ auf $V_{DD}$ gezogen wird. Dies hat zur Folge, dass die Frequenz der PLL erhöht wird. Mit 0/1 wird das PFD-Ausgangssignal auf GND gezogen, während 0/0 eine Sperrung beider Transistoren bewirkt. Dadurch stellt sich ein hochohmiger Zustand ein, der die Frequenznachführung unterbricht und die aktuelle Frequenz beibehält.

Abbildung 3 stellt das Loop-Filter für die Mittelung des PFD-Signals $U_D$ dar. Es wird ein passives Tiefpassfilter verwendet. Die Dimensionierung des Filters beginnt mit der Festlegung der Dämpfungskonstante ζ. Bei einem konstanten Skalierungsfaktor $N$ ist es sinnvoll, diese auf einen Wert zwischen 0,5 und 1,5 [1, S. 129] festzulegen, um die Stabilität der Regelschleife zu gewährleisten. Hier wurde 0,7 gewählt. Die Übertragungscharakteristik des passiven Filters erster Ordnung wird durch die beiden Zeitkonstanten $\tau_1 = R_1 \cdot C_1$ und $\tau_2 = R_2 \cdot C_1$ bestimmt. $\omega_2 = 1/\tau_2$ definiert den Punkt im Bodediagramm, bei dem die Übertragungsfunktion wieder horizontal verläuft, also eine Nullstelle [1, S. 107].

$$\tau_2 = \frac{2\,\zeta}{\omega_n}$$

Die Bedingung $\omega_0 = 1/(\tau_1 + \tau_2)$ bestimmt die -3dB-Eckfrequenz der Übertragungsfunktion. Die natürliche Frequenz $\omega_n$ bestimmt sich aus dem Verhältnis $2\pi/T_L$. Für $T_L$ = 2 µs benötigt man somit $\tau_2$ = 446 ns. Die Zeitkonstante $\tau_1$ bestimmt sich unter Zuhilfenahme von $\tau_2$ und den Verstärkungsfaktoren $K_0$ des VCOs und $K_D$ des PFDs zu:

$$\tau_1 = \frac{K_0 K_D}{N\omega_n^2} - \tau_2$$

Mit geeigneten Werten für $K_0$ und $K_D$ erhält man daraus $\tau_1$ = 1,23 µs. Mit einer Festlegung der Kapazität $C_1$ auf 15 pF ergeben sich die Widerstände $R_1$ = 82 kΩ und $R_2$ = 30 kΩ.

Das tiefpassgefilterte Ausgangssignal $U_{LF}$ steht dem spannungsgesteuerten Oszillator als Steuersignal zur Verfügung. Dessen Schaltplan ist in Abbildung 4 dargestellt. Das niederfrequente Steuersignal $U_{LF}$ kontrolliert über den VCO-Eingang den maximalen Laststrom in den Invertern des Ringoszillators und damit deren Verzögerungszeit. Die Inverter bewirken jeweils eine Phasendrehung um 180° der Eigenschwingung des VCO, wodurch sich mit dem Durchlauf der einzelnen Inverterstufen eine hochfrequente Schwingung $U_{VCO}$ ergibt, deren Frequenz von der Schaltgeschwindigkeit bzw. dem Stromfluss abhängig ist. Der Ringoszillator liefert an seinem Ausgang keinen vollen logischen Pegel von 0 bis 3,3 V. Deshalb ist dessen Ausgangssignal auf einen durch 2 MOS-Dioden vorgespannten Inverter kapazitiv gekoppelt, so dass hier wiederum der volle logische Pegel erreicht wird.

Die HF-Rechteckschwingung aus dem VCO gibt den Takt des PLL-Frequenzteilers vor. Der Frequenzteiler mit einstellbarem Skalierungsfaktor $N$ wird durch eine einfache digitale Zählerschaltung realisiert. Diese besteht aus D-Flip-Flops und bildet ein 3-Bit-Zählsystem der Zustände 000 bis 111. Bis Zustand 111 verstreichen 8 Taktzyklen, womit sich also der Teilerfaktor $N$ = 8 der PLL ergibt. Die zwischen 2 und 8 liegenden Faktoren werden durch zusätzliche Teilerstufen mit entsprechenden Zählern erzeugt, womit sich verschiedene Frequenzen entsprechend der Faktoren ausgeben lassen. So zählt der Zähler beispielsweise für $N$ = 3 drei Takte des VCOs.

## IV. Simulation

Die Simulationen sowie auch die Auswertungen nach Inbetriebnahme der Phasenregelschleife bezogen sich vor allem auf die PLL-Signale $U_{REF}$, $U_{CMP}$ und $U_{VCO}$ und die Bestimmung der Ziehzeit $T_p$ und Fangzeit $T_L$. Es wurden Temperatur- und Spannungsschwankungen sowie Toleranzen der CMOS-Parameter (durch Monte-Carlo-Simulation) miteinbezogen. Die Abbildungen 5 und 6 zeigen Simulationsergebnisse der Signale $U_{VCO}$ und $U_{CMP}$ bei normalen Betriebsbedingungen (keine Prozessschwankungen, $T$ = 25 °C). Es sind jeweils das Taktsignal, der entsprechende zeitabhängige Frequenzverlauf und eine Messung des Cycle-to-Cycle-Jitter von oben nach unten abgebildet.

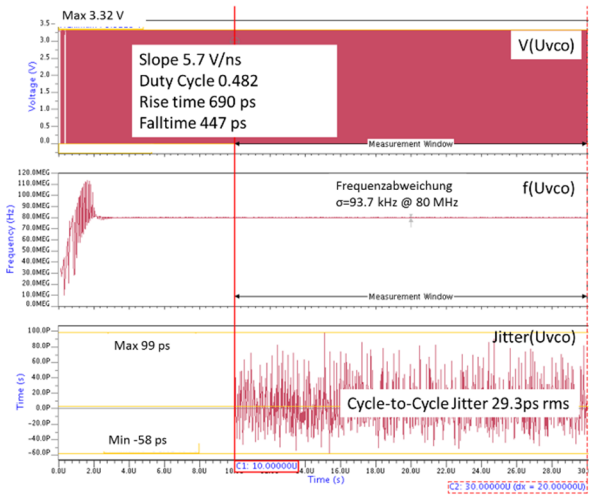



Abbildung 5: Simulation von $U_{VCO}$ mit 80 MHz Taktfrequenz.


Abbildung 6: Simulation von $U_{CMP}$ mit 10 MHz Taktfrequenz.

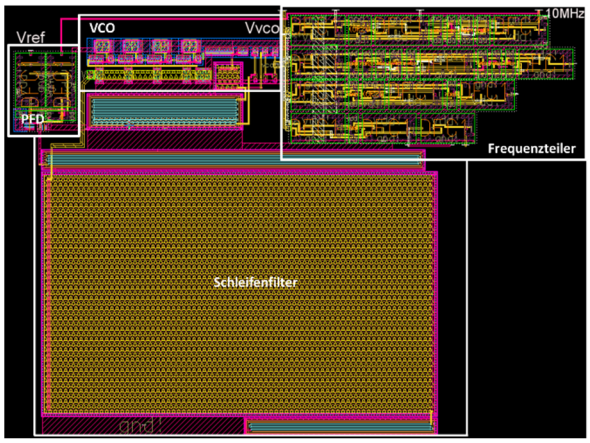



Abbildung 7: PLL-Layout.

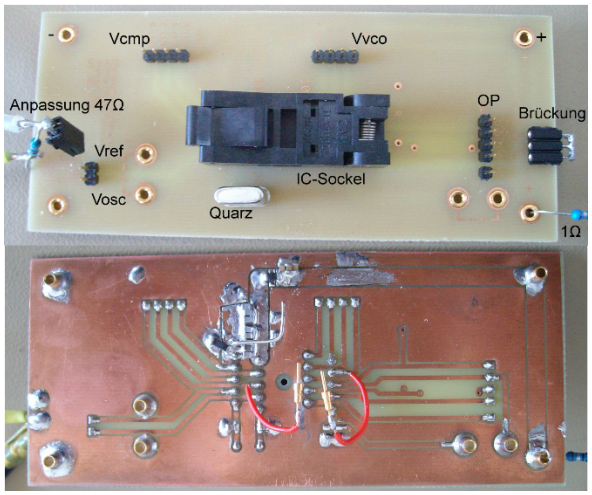



Abbildung 8: Platine mit Wechselsockel.

Anhand Abbildung 5 kann ein Vergleich zwischen den Werteverhältnissen von Vergleichs- und Oszillatorsignal durchgeführt werden. Das Oszillatorsignal $U_{VCO}$ schwingt im Mittel um die gewünschte Oszillatorfrequenz $f_{VCO}$ = 80 MHz. Dabei beträgt die mittlere Standardabweichung 93,72 kHz, also 0,12 % von 80 MHz. Es ist also zu erkennen, dass die Abweichungen keinen linear proportionalen Zusammenhang zu den ausgegebenen Frequenzen haben. Bei der gegenüber dem Referenzsignal von 10 MHz 8-fachen Frequenz des Ausgangstakts $U_{VCO}$ ist die Standardabweichung mehr als 36-mal höher als beim auf die Referenz synchronisierten Vergleichssignal $U_{CMP}$. Das Taktzittern ist dagegen mit 29,3 ps geringer als beim Vergleichssignal.

In Abbildung 6 lässt sich am Vergleichssignal $U_{CMP}$ der charakteristische Einschwingverlauf der PLL beobachten, welche den eingerasteten Zustand erreicht hat, sobald nach dem Ziehvorgang auf die Sollfrequenz von 10 MHz die Vergleichsfrequenz $f_{CMP}$ innerhalb einer Ober- und Unterschwingung auf die Sollfrequenz einrastet. Der ganze zeitliche Ablauf lässt sich in Zieh- und Fangzeit unterteilen, die sich zu jeweils 1,07 µs bzw. 2,43 µs aus der Simulation ergeben. Damit ist das Dimensionierungsziel der Fangzeit $T_L$ = 2 µs relativ genau erreicht. Im eingerasteten Zustand schwingt das Rückkopplungssignal der PLL mit einer geringen Standardabweichung von 2,55 kHz um die Sollfrequenz 10 MHz, was in Relation etwa 0,026 % entspricht. Der Cycle-to-Cycle-Jitter beträgt dabei rund 70 ps.

## V. Layout

Abbildung 7 zeigt das Gesamtlayout der Phase-Locked Loop. Digital arbeitende Schaltungsteile wie PFD und Frequenzteiler sind mit ihren Gatterzellen und innenliegender Verdrahtung links oben bzw. rechts oben angeordnet. Die analogen Schaltungskomponenten, also VCO und Schleifenfilter sind mittig oben bis unten angeordnet. Die Ein- und Aus-

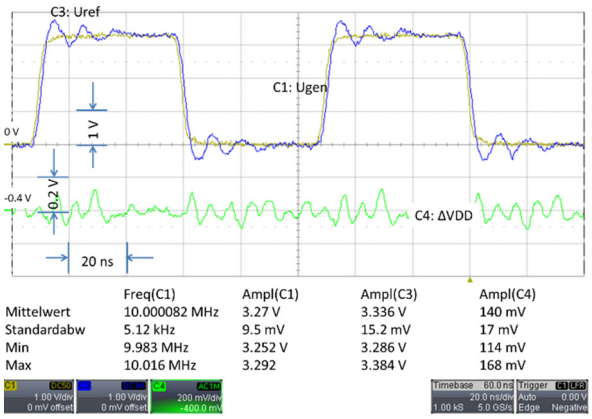



Abbildung 9: Verbleibende Störungen in der 3,3V-Versorgung.

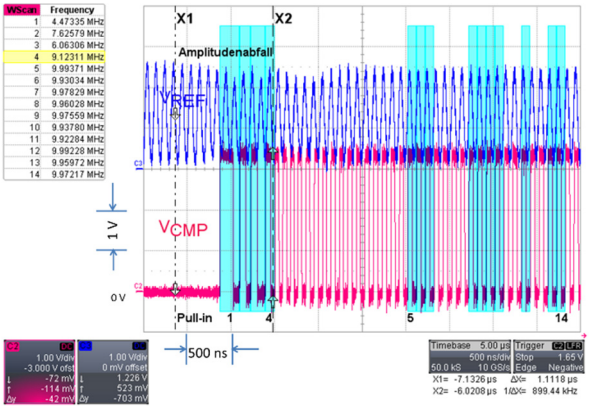



Abbildung 10: Messung der Ziehzeit $T_P$.

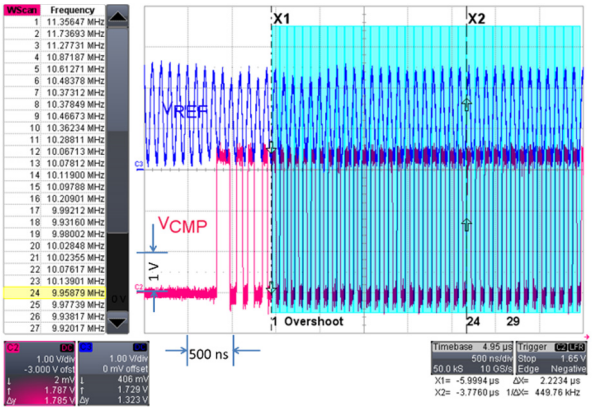



Abbildung 11: Messung der Einrastzeit $T_L$.

gangssignale $U_{REF}$, $U_{CMP}$ (10 MHz) und $U_{VCO}$ (80 MHz) liegen am oberen Rand. Die Verdrahtung befindet sich mit wenigen Ausnahmen auf den ersten beiden Metalllagen des Mikrochips, sodass die zwei übrigen Lagen für eine globale Chipverdrahtung mit Spannungsversorgung und anderen Schaltungen auf dem Chip nutzbar sind. Insgesamt misst das Schaltungslayout der PLL alleine 247 µm x 188 µm.

## VI. PCB-Entwicklung

Zu Testzwecken wurde die PLL zusammen mit weiteren Projekten gefertigt und in ein SOIC20 Gehäuse gepackt. Die Platine für die Laboruntersuchungen ist in Abbildung 8 dargestellt. Mittig auf dem PCB ist ein Wechselsockel für SOIC20-Gehäuse angeordnet, so dass Reihenmessungen über mehrere ICs möglich sind. Von dort führen Leitungen an die jeweiligen Steckerleisten und Buchsen zum Anschluss oder Abgriff von Signalen. Die Anschlüsse für die PLL befinden sich in der Draufsicht oben (Signale der Frequenzteilerstufen) und links (Buchsen sowie Steckerpins für $U_{REF}$ und den Ausgang eines chipinternen Quarzoszillators). An die Buchsen am oberen Rand wird die Spannungsversorgung angeschlossen.

## VII. Inbetriebnahme

Bei Anschluss eines Takts an den Referenzeingang der PLL wurde eine deutlich erkennbare Überkopplung des 80 MHz-Ausgangs $U_{VCO}$ auf die Spannungsversorgung beobachtet. Der Grund für diese HF-Einstrahlung auf die Stromversorgung konnte noch nicht vollständig festgestellt werden. Verschiedenste Untersuchungen ergaben, dass bereits chipintern geringe 80 MHz-Störschwingungen auf der Versorgungsspannung vorliegen, die durch parasitäre Leitungskapazitäten und Induktivitäten überkoppeln.

Eine deutliche Verringerung der Störungen wurde durch die Einbindung entsprechender Blockkondensatoren und Ferritkerne in die beiden Versorgungsleitungen auf der Platine bewirkt. Das Resultat ist mitsamt der Störung in Abbildung 9 zu sehen. C3 ist das am PLL-Eingang gemessene Referenzsignal $U_{REF}$ und C4 die AC-gekoppelte Versorgungsspannung $V_{DD}$ mit der Störung. C1 ist das vom Pulsgenerator gelieferte Signal. Die Amplitudenwerte der Störschwingungen konnten auf ±139 mV reduziert werden. Trotzdem liegt hier noch Verbesserungspotential vor, z. B. durch Einfügen eigener Versorgungspads und die Verbesserung der chipinternen Anschlussleiterbahnen. Als Treiber der Ausgangssignale wurden fertige Pads aus der Library verwendet, die dazu ebenfalls genauer untersucht werden müssen.

Die hier vorgestellten Messergebnisse konzentrieren sich auf die Betrachtung der Parameter $T_L$ und $T_P$ sowie des Vergleichssignals $U_{CMP}$ im eingerasteten Zustand. Die Ziehzeit gibt dabei die Zeitdauer nach einer Frequenzstörung bis zum Erreichen der Sollfrequenz an.

In den Abbildungen 10 und 11 sind die Signalverläufe des Referenzsignals $U_{REF}$ und des Ausgangssignals $U_{CMP}$ dargestellt. Zur Erfassung des Einschwingvorgangs der PLL können verschiedene Messansätze gewählt werden, wie z. B. das Aufstarten eines externen Referenzgenerators oder der Spannungsversorgung bei schon aktivem Referenzsignal $U_{REF}$. Hier wird eine Variante gezeigt, die mit Hilfe eines Chip-

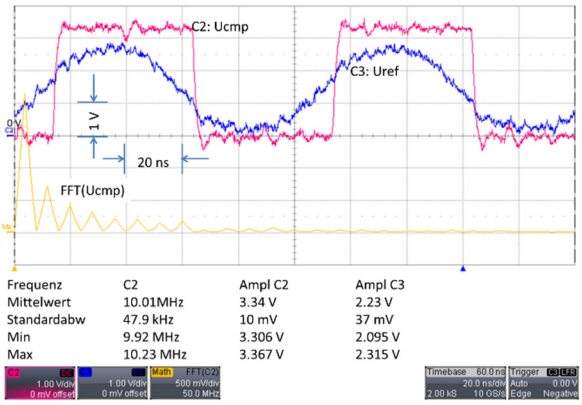



Abbildung 12: Messung im eingerasteten Zustand.

internen Kristalloszillators durchgeführt wurde und hinsichtlich der Auswertung des PLL-Einschwingvorgangs anschauliche Messungen und damit auch sinnvolle Ergebniswerte liefert. Dazu wird das vom Kristalloszillator generierte 10 MHz-Referenzsignal an den Referenzanschluss der PLL angeschlossen und auf die Flanken des Referenztakts Single-Shot getriggert.

Abbildung 10 zeigt die Erfassung der Ziehzeit $T_P$. Die Zeitmessung wird anhand der Momentanfrequenzen der Signalpulse durchgeführt. Dazu wird zur Messung der Ziehzeit $T_P$ abgeprüft, welche Taktpulse Momentanfrequenzen von unter 10 MHz aufweisen (grau hinterlegt). Dies trifft auf die Pulse 1-4 zu Beginn des Anschwingens zu. Danach wird die 10 MHz-Sollfrequenz überschritten, was den Einrastvorgang einleitet. Anhand des schon vor Puls 1 eintretenden Amplitudenabfalls kann die Ziehzeit $T_P \approx 1{,}11$ µs abgelesen werden.

Zur Messung der Fangzeit werden in Abbildung 11 Pulse mit Momentanfrequenzen über 9,9 MHz hervorgehoben, was auf den gesamten Bereich nach dem Ziehvorgang zutrifft. Anhand der Betrachtung von Abweichungen gegenüber der 10 MHz-Frequenz lässt sich feststellen, wann die PLL eingerastet ist. Als Schwellwert, den die Frequenzabweichungen nicht überschreiten dürfen, damit die PLL als eingerastet gilt, wurde die Standardabweichung herangezogen. Diese Bedingung wird ab Puls 29 erfüllt, was einer gemessenen Fangzeit $T_L = 2{,}72$ µs entspricht. Addiert man die Ziehzeit, so ergibt sich eine Gesamteinschwingzeit von 3,83 µs.

Abbildung 12 zeigt die Messwerte des Vergleichssignals $U_{CMP}$ im eingerasteten Zustand. Im abgebildeten Vergleichssignal $U_{CMP}$ sind die Restungenauigkeiten in der Signalamplitude, die durch die 80 MHz-Störungen in der Spannungsversorgung verursacht werden, zu erkennen. Diese haben aber keine größeren Auswirkungen auf die Richtigkeit des 3,3 V-Signalpegels, der im Mittel bei 3,338 V liegt. Die Standardabweichung der Frequenz von $U_{CMP}$ bei 10 MHz beträgt 47,91 kHz und der Clock-Jitter 804 ps.

Tabelle 2: Ergebniszusammenfassung.

| Parameter | Symbol | Spezifikation | Simulation | Messung |
|---|---|---|---|---|
| Versorgungsspannung | $V_{DD}$ | 3,3 V | | |
| Referenzfrequenz | $f_{REF}$ | 10 MHz | | |
| Teilerfaktor | $N$ | 2-8 | | |
| Ziehzeit | $T_P$ | N/A | 1,07 µs | 1,11 µs |
| Fangzeit | $T_L$ | 2µs | 2,43 µs | 2,72 µs |
| Einschwingzeit | $T_P + T_L$ | N/A | 3,5 µs | 3,83 µs |
| Frequenzabweichung (1σ) | $\Delta f_{CMP}$ | N/A | 2,56 kHz | 47,9 kHz |
| Jitter | | N/A | 70 ps | 804 ps |

Diese Werte übersteigen deutlich die Ergebnisse der Simulation, was in erster Linie auf die realen Betriebsverhältnisse und auch auf die Ungenauigkeiten in der PLL-Stromversorgung und vor allem auf die Signalverhältnisse der Referenzquelle zurückzuführen ist, die einen geringeren und deutlich verschliffeneren Taktpegel ausgibt als beabsichtigt. Tabelle 2 fasst die wichtigsten Ergebnisse zusammen.

## VIII. Fazit

Es wurde eine PLL mit 10 MHz Referenztakt und Ausgangstakten im Bereich 10 MHz bis 80 MHz realisiert und charakterisiert. Im Verlauf verschiedener Messungen konnte weitgehend die korrekte Funktionsweise der PLL festgestellt werden. Das Ziel der Dimensionierung auf eine Einrastzeit von 2 µs wird mit geringer Abweichung erreicht. Anhand der bestehenden Schaltung können Ansätze zur Weiterentwicklung der PLL-Funktionalität für weitere Hochschulprojekte ausgearbeitet werden.

## Danksagung

Die Autoren bedanken sich bei Frau Ninja Koetsier, die das Toplevel-Layout umgesetzt hat und bei Goran Bratek, der den Quarzoszillator entworfen hat. Besonderer Dank gilt der MPC-Gruppe für die Unterstützung bei der Entwicklung und der Chipherstellung.

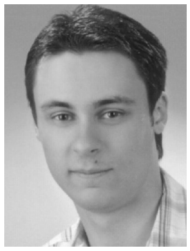
## LITERATURVERZEICHNIS

[1] R. E. Best, *Phase-Locked Loops; Design, Simulation, & Applications*, McGraw-Hill, 3. Auflage, 1997.

[2] B. Razavi, *Monolithic Phase-Locked Loops and Clock-Recovery Circuits*, IEEE Press, 1996.

[3] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill, 2001.

[4] G. Bratek, J. Giehl, B. Vettermann, „Design eines Quarzoszillators in 0,35 µm CMOS-Technologie“, *Tagungsband Workshop der Multiprojektchip-Gruppe*, S. 9 – 14, Göppingen 2010, ISSN 1862-7102.



Christian Eschenbach erhielt 2014 den akademischen Grad des Bachelor of Science im Fachbereich Informationstechnik mit Schwerpunkt Nachrichtentechnik/Elektronik der Hochschule Mannheim. Dort wird er 2015 ebenfalls den Master of Science in Informationstechnik abschließen.

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 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.

# Ein Aufwärts-Wandler-IP im 180 nm CMOS-Prozess zur Versorgung von ASICs mittels Energy Harvesting

Michael Hiller, Marc Locherer, Gerhard Forster

***Zusammenfassung*—Mit der zunehmenden Vernetzung (Internet of Things) wächst die Zahl der autonomen elektronischen Systeme stark an, so dass die dezentrale Stromversorgung dieser Systeme an Bedeutung gewinnt. Der anhaltende Trend zur Reduktion der Verlustleistung in sogenannten Micropower-ASICs ermöglicht zunehmend deren Betrieb durch Energie, die unmittelbar der Umwelt entnommen wird (Energy Harvesting). Häufig ist die verfügbare Energie ausreichend, jedoch die vom Wandler gelieferte Spannung zu niedrig für den Betrieb elektronischer Schaltungen. Ziel ist es daher, ein IP verfügbar zu haben, mit dessen Einsatz ein Chip betrieben werden kann, auch wenn die Versorgungsspannung noch unterhalb der Schwellenspannung der Transistoren liegt. Im vorliegenden Beitrag wird der Entwurf eines Aufwärtswandler-IPs beschrieben, das ab einer externen Versorgungsspannung von 170 mV eine interne Versorgungsspannung von 1,8 V erreicht und dabei 10 µA abgeben kann. Mit höherer Versorgungsspannung nimmt die verfügbare Leistung zu. Der Wandler ist vollständig in einem 180 nm CMOS-Prozess integriert. Er beinhaltet einen Low-Voltage-Oszillator mit Resonator sowie eine Ladungspumpe. Externe Bauelemente sind somit nicht erforderlich.**

***Schlüsselwörter*—Aufwärtswandler, Oszillator, Ladungspumpe, Energy Harvesting.**

## I. Einleitung

Das Internet of Things (IoT) ist schon heute ein wichtiges Thema der Mikroelektronik-Entwicklung und wird in Zukunft eine noch größere Rolle spielen. Es bringt eine große Zahl autonomer elektrischer Systeme mit sich, welche separat mit elektrischer Energie versorgt werden müssen. Energie ist in ihren verschiedenen Erscheinungsformen allgegenwärtig, allerdings nur bedingt nutzbar. Wird die Energie aus der Umwelt in eine nutzbare Energieform überführt (Energy Harvesting), so lässt sich eine autarke Energieversorgung von Verbrauchern realisieren. Damit sind für einen fortlaufenden Betrieb keine Batteriewechsel oder das Laden des Gerätes erforderlich. Problematisch beim Energy Harvesting ist, dass häufig nur geringe Leistungen und Spannungen unterhalb der Transistorschwellenspannung zur Verfügung stehen. So sind bei der Nutzung von Körperwärme am Menschen mittels Peltier-Elementen in der Regel nur Spannungen unter 200 mV erzielbar.

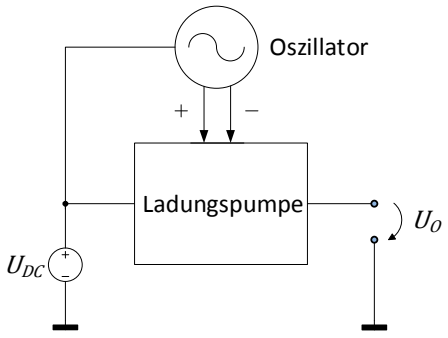



Abbildung 1: Schaltungskonzept mit Ladungspumpe und symmetrischem Oszillator als Taktquelle.

Ziel dieser Arbeit war es, eine Schaltung zu entwickeln, welche den Betrieb eines ASICs mit Eingangsspannungen $U_{DC} < U_{TH}$ zulässt und damit eine Versorgung des ASICs über Energy Harvesting erlaubt. Da es sich sowohl bei der Ein- als auch bei der Ausgangsspannung um eine Gleichspannung handelt, entspricht das System einem DC/DC Wandler, wobei folgende Zielspezifikationen angesetzt wurden:

- Eingangsspannung $U_{DC} \leq 150$ mV
- Ausgangsspannung $U_o \geq 1{,}6$ V (typ. 1,8 V)
- Laststrom $I_L \geq 10$ µA

Die Schaltung soll vollständig integriert sein und ohne externe Bauelemente auskommen.

Mit ähnlichen Zielsetzungen haben sich bereits unterschiedliche Arbeitsgruppen befasst. Gemeinsames Merkmal dieser Arbeiten ist jedoch, dass entweder eine höhere Eingangsspannung vorausgesetzt wird [1]-[4] oder zusätzliche Bauelemente erforderlich sind (z.B. Transformatoren mit großer Übersetzung), die sich nicht monolithisch integrieren lassen [5]-[9]. H. Fukuta et al. berichten hingegen erstmals von einem vollständig integrierten Aufwärtswandler, realisiert in einem 65 nm CMOS-Prozess [10].

Der vorliegende Beitrag befasst sich mit dem Systementwurf und der Umsetzung eines vollintegrierten

Michael Hiller, mhiller@mail.hs-ulm.de, Mark Locherer, locherer@mail.hs-ulm.de und Gerhard Forster, forster@hs-ulm.de sind Mitglieder der Hochschule Ulm, Prittwitzstraße 10, 89075 Ulm.

Aufwärtswandlers in einem 180 nm CMOS-Prozess der Fa. UMC. Das folgende zweite Kapitel gibt eine kurze Einführung in das System- und Schaltungskonzept. Das dritte Kapitel befasst sich mit dem Schaltungsentwurf des Gegentaktoszillators und der dazu erforderlichen Modellierung des vollintegrierten Transformators. Im vierten Kapitel wird, ausgehend von einer theoretischen Analyse der Prinzipschaltung, die Entwicklung der Ladungspumpe beschreiben. Das Gesamtsystem und die ermittelten Ergebnisse werden schließlich im fünften Kapitel vorgestellt. Der Beitrag schließt mit einer Zusammenfassung und dem Ausblick.

## II. Schaltungskonzept

Für eine Spannungerhöhung ist es notwendig, dass die DC-Eingangsspannung in eine AC-Spannung umgewandelt wird. Dazu ist ein Oszillator erforderlich, welcher auch mit Spannungen unter der Schwellenspannung $U_{TH}$ arbeiten muss. Dafür eignen sich, aufgrund der Anforderung einer Integration, nur analoge Oszillatoren mit LC-Resonator. Der Gegentaktoszillator ist dabei besonders geeignet, da er wegen seines Differenzverstärkers eine hohe Schleifenverstärkung und damit ein sicheres Anschwingverhalten, auch unter Belastung, gewährleistet. Außerdem ist die Mitkopplung durch eine einfache Kreuzkopplung realisierbar. Mit Hilfe des Oszillators lässt sich die Spannung des Eingangssignals erhöhen, allerdings ist maximal eine Amplitude erreichbar, die der doppelten (bzw. differentiell der vierfachen) Eingangsspannung entspricht. Da dies immer noch nicht ausreichend ist, um die geforderte Ausgangsspannung zu erzeugen, kommt noch eine nachgeschaltete Ladungspumpe zum Einsatz, für welche der Oszillator als Taktquelle dient. Das sich ergebende Gesamtsystem ist als Blockschaltbild in Abbildung 1 dargestellt.

Am Ausgang muss die Ladungspumpe eine gepulste Gleichspannung liefern, welche mit Hilfe von Chipressourcen geglättet werden kann und nicht erst eine Gleichrichtung erfordert. Dabei steht die Effizienz der Ladungspumpe im Mittelpunkt. Als Arbeitsfrequenz für das System werden 200 – 500 MHz angestrebt, da lediglich kleine Bauteilwerte durch die Integration möglich sind.

Für die aktive Ladungspumpe muss ausreichend Ansteuerspannung zur Verfügung stehen, damit diese effizient arbeiten kann. Dafür reicht die Ausgangsspannung des Oszillators allerdings bei kleinen Eingangsspannungen noch nicht aus. Deshalb soll seine Ausgangsspannung mit Hilfe eines integrierten Transformators weiter erhöht werden.

## III. Gegentaktoszillator

Ein analoger Oszillator basiert auf einer frequenzselektiven Mitkopplung des Ausgangs- auf das Eingangssignal. Damit das System schwingt, muss der

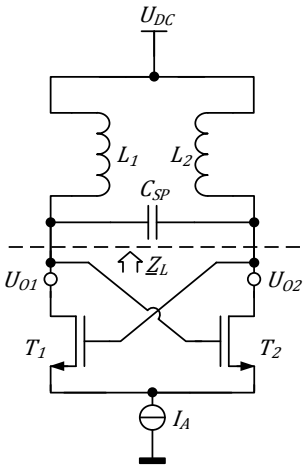



Abbildung 2: Gegentaktoszillator mit Stromquelle zur Arbeitspunkteinstellung und LC-Resonator.

Betrag des Produkts aus Rückkopplungsfaktor $\underline{k}_R$ und Verstärkung $\underline{V}$ größer eins sein [11]:

$$\left|\underline{k}_R \underline{V}\right| > 1 \tag{1}$$

Weiterhin muss die Phase $\varphi$ des Produkts folgende Bedingung erfüllen:

$$\varphi\left(\underline{k}_R \underline{V}\right) \bmod 2\pi = 0 \tag{2}$$

### A. Verstärker

Für den Gegentaktoszillator gilt, dass der Rückkopplungsfaktor $\underline{k}_R$ ideal angesetzt werden kann und somit den reellen Wert eins annimmt. Daraus resultiert für den Betrag der Verstärkung die Bedingung

$$\left|\underline{V}\right| > 1\,. \tag{3}$$

Aus der Darstellung des Gegentaktoszillators in Abbildung 2 wird ersichtlich, dass $T_1$ und $T_2$ einen Differenzverstärker bilden, welcher für die Verstärkung verantwortlich ist. Allgemein gilt für die Verstärkung $\underline{V}$ eines Differenzverstärkers, dass diese von dem Lastwiderstand und der Transkonduktanz $g_m$ abhängt [11]. Der Lastwiderstand entspricht beim Gegentaktoszillator einer Parallelschaltung aus Transistorausgangswiderstand $r_o$ und Schwingkreiseingangswiderstand $\underline{Z}_L/2$. Somit gilt

$$\underline{V} = -g_m\left(r_o \left\| \left(\underline{Z}_L/2\right)\right.\right). \tag{4}$$

Für den idealen Schwingkreis gilt, dass dieser in Resonanz einen unendlichen Eingangswiderstand besitzt. Damit ist die Verstärkung ausschließlich von den Parametern der MOSFETs abhängig:

$$V = -g_m r_o \quad (\text{für } \left|\underline{Z}_L\right| \to \infty) \tag{5}$$

Die Stromquelle $I_A$ in der Abbildung 2 dient beim Gegentaktoszillator zur Arbeitspunkteinstellung der

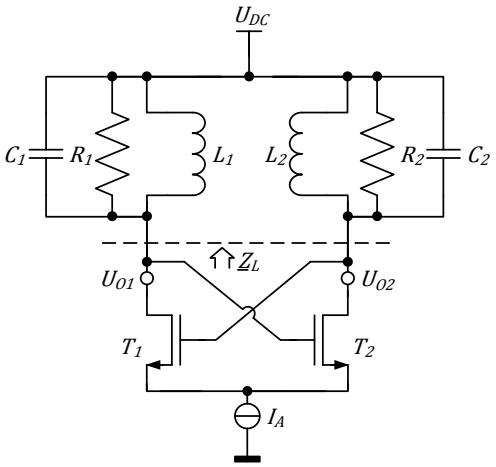



Abbildung 3: Gegentaktoszillator mit verlustbehafteten Parallelschwingkreisen und entferntem Schwingkreiskondensator.

Transistoren. Für geringe Spannungen $U_{DC}$ kann allerdings keine Stromquelle realisiert werden, so dass diese entfallen muss. Da im Ruhearbeitspunkt die Spannungen an Drain und Source gleich sind, also

$$U_{DS} = U_{GS} = U_{DC}, \tag{6}$$

müssen die Transistoren die Anforderung erfüllen, dass in der „Diodenschaltung“ (bei der Gate und Drain kurzgeschlossen sind) das Produkt $g_m r_o > 1$ ist. Bei den verwendeten Low VT-Transistoren des 180 nm Prozesses von UMC ist dies auch noch unterhalb der Schwellenspannung der Fall. Dieser Sachverhalt ist lediglich abhängig von Technologieparametern und gilt im Wesentlichen unabhängig von der Dimensionierung von $W$ und $L$. Mit einem realen Schwingkreis, insbesondere aber mit einem belasteten Schwingkreis, wird $\underline{Z}_L$ niederohmig. Um dennoch die Schwingbedingung zu erfüllen, muss dann die Transkonduktanz $g_m$ entsprechend vergrößert werden, woraus schließlich die erforderliche Größe von $W/L$ resultiert.

### *B. Resonanzkreis*

Für reale Schwingkreise gilt, dass deren Eingangsimpedanz $\underline{Z}_L$ bei der Resonanzfrequenz einen endlichen reellen Wert annimmt. Die Konsequenz dieses Zusammenhangs kann am besten durch eine getrennte Betrachtung bezüglich der Symmetrieebene des Gegentaktoszillators erläutert werden. Dazu wird der Resonanzkreis aus Abbildung 2 durch zwei unabhängige und verlustbehaftete Parallelschwingkreise ersetzt, was zu Abbildung 3 führt. Bei der Resonanzfrequenz $\omega_0$ kompensieren sich die Spulen und Kapazitäten, wodurch lediglich die Widerstände $R_1$ und $R_2$ wirksam sind. Dies ist der Fall bei

$$\omega_0 = \frac{1}{\sqrt{LC}}. \tag{7}$$

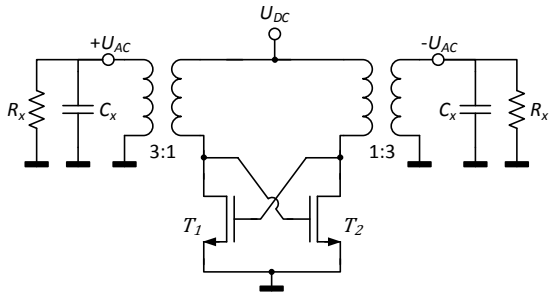



Abbildung 4: Gegentaktoszillator mit Transformator-Auskopplung.

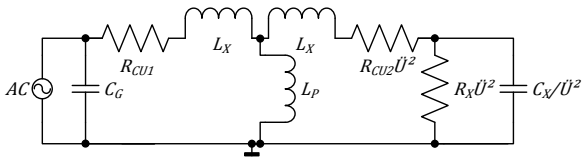



Abbildung 5: Ersatzschaltung eines der beiden Resonatoren.

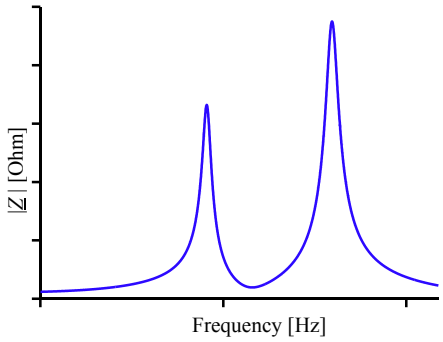



Abbildung 6: Eingangsimpedanz des Resonators.

Die Widerstände $R_1$ respektive $R_2$ entsprechen dem Parallelwiderstand $R_P$ eines allgemeinen Parallelschwingkreises und hängen über den Kennwiderstand $R_K$ mit der Güte $Q$ des Schwingkreises zusammen:

$$Q = \frac{R_P}{R_K} \tag{8}$$

Da der Lastwiderstand des Differenzverstärkers sich bei Resonanz aus der Parallelschaltung von $R_{1,2}$ mit dem Transistorausgangswiderstand $r_o$ ergibt, folgt somit, dass die Güte des Resonanzkreises einen Einfluss auf die Verstärkung hat.

### *C. Netzwerk*

Wie in Kapitel II erwähnt, soll die Signalauskopplung aus dem Oszillator mittels Transformator erfolgen. Abbildung 4 zeigt die zugehörige Schaltungstopologie mit nachgebildeten Lastelementen $R_x$ und $C_x$ in der angestrebten Version ohne Stromquelle. Aufgrund der Tatsache, dass anstelle eines einfachen LC-

Resonators ein Netzwerk mit Transformator zum Einsatz kommt, resultieren Konsequenzen für die Oszillation. Die Abbildung 5 zeigt eine Darstellung des Netzwerks, welches nun als Resonator fungiert. Bei den Widerständen $R_{CU1,2}$ handelt es sich um die Serienwiderstände der Transformatorwicklungen und bei $L_P$ um die Induktivität der Primärwicklung. Die Kapazität $C_X$ und der Widerstand $R_X$ repräsentieren dagegen die Last. Mit Hilfe des Übersetungsverhältnisses $\ddot{U}$ des Transformators kann diese Last von der Sekundärseite auf die Primärseite übersetzt werden. Der Transformator kann mit Hilfe verschiedener Ersatzschaltbilder beschrieben werden. Durch die Darstellung als T-Glied mit den beiden Spulen $L_X$ kann die nichtideale Kopplung zwischen den beiden Transformatorseiten berücksichtigt werden. Zur weiteren Untersuchung wurde zusätzlich die Gatekapazität $C_G$, die aufgrund der Kreuzkopplung an der Primärseite anliegt, eingefügt. Der Betrag des Eingangswiderstands dieses Netzwerks über der Frequenz ist in Abbildung 6 schematisch dargestellt. Es ist ersichtlich, dass sich aufgrund der Struktur der Schaltung zwei Maxima ergeben. Diese Maxima stellen potentielle Resonanzstellen für den Oszillator dar.

### *D. Transformator*

Die On-Chip-Umsetzung des Transformators wurde als Stacked Transformer angesetzt. Bei diesem Typ liegen die Primär- und Sekundärwicklung in verschiedenen Lagen übereinander. Deshalb ergibt sich eine hohe Kopplung und variable Übersetzungsverhältnisse sind möglich [12]. Zur simulationstechnischen Verifikation und zur Oszillatordimensionierung muss der On-Chip-Transformator modelliert werden. Dabei sind folgende Parameter von besonderer Bedeutung:

- Übersetzungsverhältnis $\ddot{U}$
- Kopplungsfaktor $k$
- Überlappungskapazität $C_{ov}$
- Spulenparameter der Primär- und Sekundärspulen

Da integrierte Spulen große Chipflächen erfordern, ist die Windungszahl begrenzt. Das Übersetzungsverhältnis wurde deshalb auf $\ddot{U}$ = 1:3 festgelegt. Für die Kopplung wurde eine Annahme, basierend auf Werten aus der Literatur getroffen [13], wonach mit $k$ = 0,85 gerechnet werden kann. Die Transformator-Überlappungskapazität zwischen Primär- und Sekundärwicklung entspricht der eines Plattenkondensators, bei dem die Plattenfläche (Produkt aus Länge $l$ und Breite $w$), der Plattenabstand $t_{ox}$ und die Permittivität $\varepsilon_{ox}$ eingehen:

$$C_{ov} = \frac{1}{2} l \cdot w \cdot \frac{\varepsilon_{ox}}{t_{ox}} \tag{9}$$

$C_{ov}$ ist dabei zwei mal vorhanden, nämlich zwischen den Eingängen der Primär- beziehungsweise Sekundärspule und zwischen den Ausgängen. Aus dieser Aufteilung rührt auch der Faktor 0,5 in Gleichung (9) her.

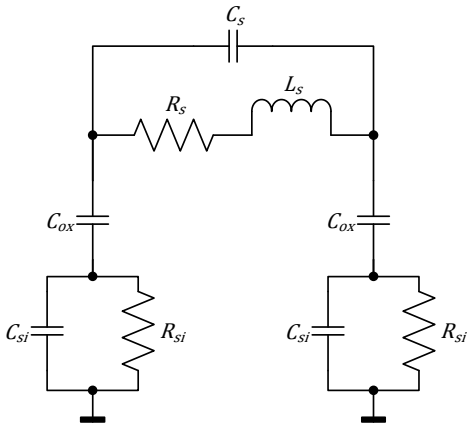



Abbildung 7: π-Modell einer planaren integrierten Spule nach Greenhouse [14].

### *E. Spulenmodellierung*

Abbildung 7 zeigt ein Modell aus konzentrierten Elementen, mit dem die integrierten Spulen des Transformators modelliert wurden. Aufgrund der Struktur wird dieses als π-Modell bezeichnet. Die Gesamtinduktivität $L_S$ der Spule befindet sich dabei in Serie mit dem parasitären Leitungswiderstand $R_S$. Zur Bestimmung von $L_S$ bei rechteckigen Spulen liefert die Methode von Greenhouse [14] eine hohe Genauigkeit. Das Verfahren unterteilt die Induktivität einer Leitung in Eigen- und Kopplungsinduktivität. Als Kopplungsinduktivität wird dabei die positive und negative magnetische Kopplung $M_+$ und $M_-$ jedes Leiterstücks mit allen anderen Leiterstücken verstanden. Für einen Teilabschnitt $L_{teil}$ der Spule ergibt sich die Induktivität damit wie folgt

$$L_{teil} = L_{eigen} + \sum M_+ + \sum M_- \,. \tag{10}$$

Eine Kopplung findet dabei immer statt, außer wenn die Leitungen rechtwinklig aufeinander stehen. Das ist auch der Grund für die Beschränkung des Verfahrens auf rechteckige Spulen. Die Stromrichtung in den beteiligten Leitern bestimmt das Vorzeichen der Kopplung. Für parallele Leiter gleicher Länge besteht für die Kopplungsinduktivität $M_{koppel}$ folgender Zusammenhang:

$$M_{koppel} = 2 \cdot 10^{-7} \, l \cdot Q_{koppel} \tag{11}$$

$$Q_{koppel} = \ln\left( x + \sqrt{1 + x^2} - \sqrt{1 + \frac{1}{x^2}} + \frac{1}{x} \right) \tag{12}$$

$$\text{mit } x = \frac{l}{GMD} \tag{13}$$

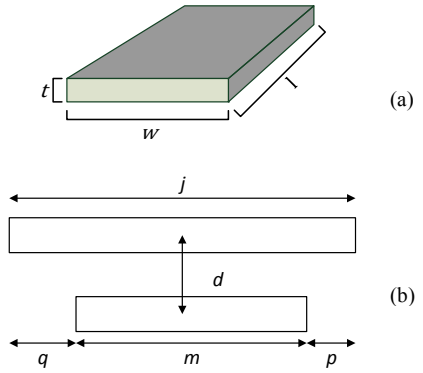



Abbildung 8: (a) Geometrie einer Leiterbahn, (b) unvollständig überlappende Leiterbahnen.

Dabei wird zur Berücksichtigung der realen Leitergeometrie, also der räumlichen Ausdehnung des Leiters, der mittlere geometrische Abstand *GMD* anstelle des Abstands *d* der Leitermitten verwendet. Berechnet wird der *GMD* dabei mit:

$$\ln GMD = \ln d - \frac{1}{12\left(\frac{d}{w}\right)^2} + \frac{1}{60\left(\frac{d}{w}\right)^4} + \frac{1}{168\left(\frac{d}{w}\right)^6} + \frac{1}{360\left(\frac{d}{w}\right)^8} + \frac{1}{660\left(\frac{d}{w}\right)^{10}} + \ldots \quad (14)$$

Abbildung 8 skizziert den Fall, dass die Koppelinduktivität für Leiter unterschiedlicher Längen *j* und *m* bestimmt werden muss. In den folgenden zwei Gleichungen sind die zwei möglichen Fälle beschrieben:

$$2M_{j,m} = \left(M_j + M_m\right) - M_q\big|_{p=0} \quad (15)$$

$$M_{j,m} = M_{m+p} - M_p\big|_{p=q} \quad (16)$$

Durch Fallunterscheidung können somit alle Koppelinduktivitäten bestimmt werden. Unter Einbeziehung von Grover [5] ergibt sich die Eigeninduktivität für jeden Teilabschnitt zu

$$L_{eigen} = 2\cdot 10^{-9}\left(1\cdot 10^2 l\right)\left(\ln(2x) + 0{,}50049 + \frac{1}{3x}\right) \quad (17)$$

$$\text{mit } x = \frac{l}{(w+t)}. \quad (18)$$

Dabei macht Greenhouse die Näherungen, dass zum einen die Permeabilität des Leiters eins ist und zum anderen die Leiterbahndicke *t* gering und die Frequenz hoch ist. Die Gesamtinduktivität der Spule ergibt sich dann aus der Summe der Teilinduktivitäten aller Spulenkanten zu $L_S = \sum L_{teil}$ .

Für den Serienwiderstand $R_S$ gilt, dass sich dieser aus dem Gleichstromwiderstand und einer frequenzabhängigen Komponente zusammensetzt. Der Gleichstromwiderstand $R_{S,DC}$ kann mit Hilfe des Schichtwiderstandes $R_\square$ ausgedrückt werden:

$$R_{S,DC} = \frac{l}{w}\cdot R_\square \quad (19)$$

Ursächlich für den frequenzabhängigen Anteil sind Wirbelströme, die durch den Skineffekt, den Proximity-Effekt oder Substratwirbelströme zu Verlusten führen. Mit Ausnahme des Skineffektes lassen sich diese Effekte allerdings nicht hinreichend genau mit konzentrierten Modellen bestimmen und werden deshalb normalerweise durch Feldsimulation ermittelt. Unter der Annahme, dass bei niedrigen Frequenzen eine ausreichende Genauigkeit erreicht wird, wenn beide Effekte vernachlässigt werden [16], soll lediglich der Skineffekt berücksichtigt werden. Die Skintiefe $\delta$ lässt sich in Abhängigkeit von der Kreisfrequenz $\omega$, dem spezifischen Widerstand $\rho$ und der absoluten Permeabilität $\mu$ des Leiters nach folgender Gleichung ermitteln:

$$\delta = \sqrt{\frac{2\cdot\rho}{\omega\cdot\mu}} \quad (20)$$

Der damit frequenzabhängige Leitungswiderstand $R_S$, welcher durch eine Reduktion des effektiven Leitungsquerschnitts verursacht wird, ermittelt sich dann wie folgt:

$$t_{eff} = \delta\cdot\left(1 - e^{-\frac{t}{\delta}}\right) \quad (21)$$

$$R_S = \frac{t}{t_{eff}}\cdot\frac{l}{w}\cdot R_\square \quad (22)$$

Bei der Spulenrealisierung ergeben sich zwangsläufig auch parasitäre Kapazitäten, welche durch $C_{ox}$ und $C_S$ im π-Modell repräsentiert sind. $C_S$ stellt eine direkte Kopplung zwischen Ein- und Ausgang dar. Ursächlich sind die Plattenkondensatoren zwischen den Leiterbahnen und zwischen dem Mittelanschluss und den Spulenwindungen. Ersterer Effekt spielt dabei eine untergeordnete Rolle, da zum einen der Leiterabstand groß ist bei einer geringen Leiterdicke und zum anderen der Potentialunterschied gering ist. Für den zweitgenannten Effekt lässt sich die Kapazität $C_S$ damit über einen Plattenkondensator berechnen:

$$C_s = n\cdot w^2\cdot\frac{\varepsilon_{ox}}{t_{oxM1-M2}} \quad (23)$$

Die Fläche ergibt sich mit Hilfe der Spulenparameter Windungsanzahl *n* und Leiterbahnbreite *w*, während der Plattenabstand von der Oxiddicke zwischen den Leitern abhängt. Aufgrund der Breite der Leiterbahn

Tabelle 1: Modellparameter der Transformatorspulen nach Dimensionierung.

| | Primärspule | Sekundärspule |
|---|---|---|
| $N$ | 3 | 9 |
| $L_S$ | 9,2 nH | 85,5 nH |
| $C_{ox}$ | 633 fF | 1,3 pF |
| $R_{si}$ | 648 Ω | 642 Ω |
| $C_{si}$ | 77 fF | 80 fF |
| $C_S$ | 80 fF | 80 fF |
| $R_S$ | 6,2 Ω | 97 Ω |
| $C_{ov}$ | 2,3 pF | |

Tabelle 2: Transistorparameter nach Dimensionierung.

| | |
|---|---|
| $W$ | 3,5 mm |
| $L$ | 240 nm |
| $N_{Finger}$ | 350 |

und dem endlichen Abstand zum Bulk existiert eine Kapazität $C_{ox}$:

$$C_{ox} = \frac{1}{2} l \cdot w \cdot \frac{\varepsilon_{ox}}{t_{ox}} \tag{24}$$

Da die parasitären Elemente gegen Masse über die gesamte Ausbreitung der Spule verteilt sind, sind beim π-Modell $C_{ox}$, $C_{si}$ und $R_{si}$ auf den Ein- und Ausgang aufgeteilt, was die Ursache für den Faktor 0,5 ist. Als Näherungsmodell für das Bulkverhalten dient die RC-Parallelschaltung von $C_{si}$ und $R_{si}$, deren Werte sich wie folgt berechnen:

$$C_{si} = \frac{1}{2} l \cdot w \cdot C_{sub} \tag{25}$$

$$R_{si} = \frac{2}{l \cdot w \cdot G_{sub}} \tag{26}$$

### *F. Dimensionierung*

Die Simulation zeigt, dass der Serienwiderstand $R_S$ besonders kritisch für die Schwingfähigkeit des Systems ist. Für die Primärspule wird deshalb eine Leiterbahnbreite $w$ = 30 µm angesetzt. Sie besteht aus drei Windungen und befindet sich in Metalllage sechs. Da die anderen Lagen deutliche dünnere Metalldicken besitzen, besteht die Sekundärspule mit ihren neun Windungen aus einer Parallelschaltung identischer

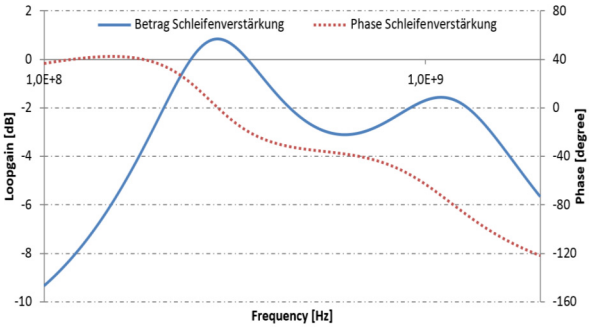



Abbildung 9: Schleifenverstärkung des Oszillators nach Betrag und Phase.

Spulen in den Lagen drei und vier. Daraus ergeben sich die in Tabelle 1 gezeigten Parameter für das Transformatormodell. Für die Bestimmung von $R_S$ wurde dabei eine Frequenz von 300 MHz zugrunde gelegt. Mit diesem Transformatormodell wurde am Simulator Spectre® die Dimensionierung der Transistoren anhand der Schleifenverstärkung durchgeführt. Dabei wurden die in Tabelle 2 dargestellten Transistordaten ermittelt. Für diese Dimensionierung des Übertragers und der Transistoren ergibt sich die in Abbildung 9 dargestellte Schleifenverstärkung nach Betrag und Phase. Dabei ist als Eingangsspannung $U_{DC}$ = 150 mV und als Belastung eine Parallelschaltung aus $R_x$ = 10 kΩ und $C_x$ = 500 fF angesetzt. Um die erforderliche Schleifenverstärkung $|k_R \cdot V| > 1$ zu erreichen, muss das $W/L$-Verhältnis größer als 14.000 gewählt werden. Dem Diagramm ist zu entnehmen, dass nur bei der ersten Resonanzfrequenz $f \approx$ 300 MHz sowohl die Amplituden- als auch die Phasenbedingung erfüllt ist.

## IV. LADUNGSPUMPE

Die Möglichkeiten zur Realisierung einer Ladungspumpe sind sehr vielfältig. Bei der eingesetzten Variante handelt es sich um eine Ladungspumpe mit symmetrischer Ansteuerung. Der Schaltplan der Prinzipschaltung ist in Abbildung 10 dargestellt. Für die weitere Betrachtung werden zunächst ideale Schalter mit geringen Durchlasswiderständen $R_{ON}$ angenommen. Zur Ansteuerung der Schalter dienen idealerweise Rechteckspannungen, wobei deren Phasen abhängig von der Spannung $U_{AC}$ sind. Dabei gilt, dass $\Phi_A$ gleichphasig und $\Phi_B$ gegenphasig zu $U_{AC}$ ist. Die Amplitude von $U_{AC}$ beträgt $\pm U_P$. Damit gibt es für das System zwei mögliche Zustände, welche im Folgenden als Phase A und Phase B bezeichnet werden. In Phase A ist $U_{AC} = -U_P$ und $C_P$ wird auf $U_{DC} + U_P$ geladen. Dies erfolgt über den geschlossenen Schalter A. Anschließend findet in Phase B ein Ladungsausgleich zwischen $C_P$ und $C_L$ statt, wobei $U_{AC} = +U_P$ ist. Dabei ist Schalter A geöffnet und Schalter B geschlossen.

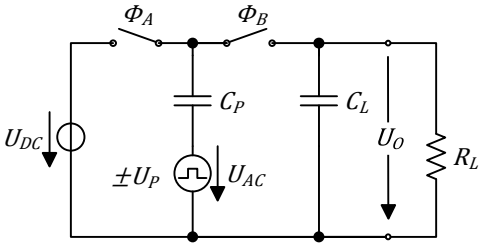



Abbildung 10: Ladungspumpe (Prinzipschaltung).

### *A. Funktionsweise*

Durch eine detaillierte Betrachtung der Phasen lassen sich die Zusammenhänge zwischen den Ein- und Ausgangsspannungen herleiten. Abbildung 11 zeigt die skizzierten Signalverläufe der verschiedenen Spannungen, wobei Phase A grau hinterlegt ist. Als Anfangsbedingung für den Zeitpunkt $t = 0$ wird angenommen, dass über $C_P$ keine Spannung vorhanden ist, jedoch $C_L$ eine Restladung besitzt. Im Zeitbereich $0 \le t \le t_1$ ($t_1 = ½T - \Delta t$ mit $\Delta t << T$) wird $C_P$ über den Durchlasswiderstand $R_{ON,A}$ mit der Zeitkonstanten $\tau_0 = R_{ON,A}C_P$ auf $U_{DC} + U_P$ geladen (Abbildung 11b):

$$U_{CP}(t) = (U_{DC} + U_P)\left(1 - e^{-\frac{t}{\tau_0}}\right) \tag{27}$$

Wenn $\tau_0 << T$ gilt, ergibt sich für den Zeitpunkt $t_1$ innerhalb der Taktperiode $[n]$, dass $C_P$ voll geladen ist,

$$U_{CP1} = U_{DC} + U_P, \tag{28}$$

und damit folgende Ladung enthält:

$$Q_{P1}[n] = C_P(U_{DC} + U_P) \tag{29}$$

Unabhängig davon wird die Lastkapazität $C_L$ über den Lastwiderstand $R_L$ mit der Zeitkonstanten $\tau_1 = R_L C_L$ entladen (Abbildung 11c). Für die Funktion der Ladungspumpe ist es unabdinglich, dass $\tau_1 > \tau_0$ ist, wobei eigentlich $\tau_1 >> \tau_0$ sein sollte. Die Spannung am Ausgang zum Zeitpunkt $t_1$ innerhalb der Taktperiode $[n]$ ergibt sich damit zu

$$U_{o1}[n] = U_o[n-1]e^{-\frac{T/2}{\tau_1}} \tag{30}$$

und damit die verbleibende Ladung auf $C_L$ zu

$$Q_{L1}[n] = C_L U_o[n-1]e^{-\frac{T/2}{\tau_1}}. \tag{31}$$

Die Gesamtladung im System zum Zeitpunkt $t_1$ ergibt sich aus der Summe der Teilladungen:

$$Q_{ges1}[n] = Q_{P1}[n] + Q_{L1}[n] \tag{32}$$

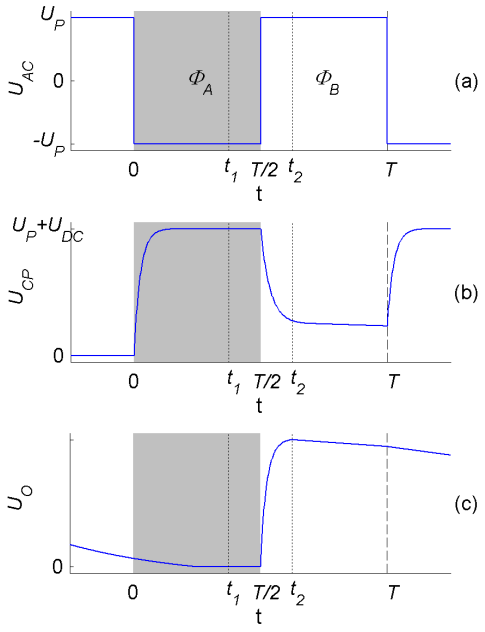



Abbildung 11: Zeitlicher Verlauf der Spannungen während der Taktperiode $[n]$. (a) Idealisierte Wechselspannung $U_{AC}$ mit den Phasen $\Phi_A$ und $\Phi_B$. (b) Spannung am Pumpkondensator $C_P$. (c) Ausgangsspannung $U_o$.

Durch Einsetzen von (29) und (31) erhält man

$$Q_{ges1}[n] = C_P(U_{DC} + U_P) + C_L U_o[n-1]e^{-\frac{T/2}{\tau_1}}. \tag{33}$$

Aufgrund der angenommenen rechteckförmigen Ansteuerspannung findet der Übergang von Phase A nach B in unendlich kurzer Zeit statt. Zur Betrachtung können die Vorgänge über den Zeitbereich $t_1 \le t \le t_2$ ($t_2 = ½T + \Delta t$) ausgedehnt werden. Wegen der Ladungserhaltung muss die Ladung im System im Übergang konstant bleiben. Durch die Änderung der Spannung $U_{AC}$ von $-U_P$ auf $+U_P$ wechseln auch beide Schalter ihre Zustände. Daraus resultiert ein Ladungsausgleich zwischen $C_P$ und $C_L$ mit der Zeitkonstanten

$$\tau_3 = R_{ON,B}\frac{C_P C_L}{C_P + C_L}. \tag{34}$$

Zum Zeitpunkt $t_2$ gilt für $\tau_3 << T$, dass der Ladungsausgleich abgeschlossen ist. Damit befindet sich das System in Phase B. In dieser Phase ist $C_P$ mit dem Ausgang $U_o$ verbunden. Für die Spannung an $C_P$ zum Zeitpunkt $t_2$ gilt in diesem Fall

$$U_{CP2} = U_{o2}[n] - U_P \tag{35}$$

und für die Ladung

$$Q_{P2}[n] = C_P(U_{o2}[n] - U_P). \tag{36}$$

Unter der Voraussetzung $\tau_1 >> 2\Delta t$ ist die Gesamtladung des Systems zum Zeitpunkt $t_2$ gleichzusetzen der Gesamtladung zum Zeitpunkt $t_1$:

$$Q_{ges2}[n] = Q_{ges1}[n] \tag{37}$$

Unter Berücksichtigung von (36) und (33) erhält man:

$$\begin{aligned} &C_P\left(U_{o2}[n] - U_P\right) + C_L U_{o2}[n] = \\ &C_P\left(U_{DC} + U_P\right) + C_L U_o[n-1] e^{-\frac{T/2}{\tau_1}} \end{aligned} \tag{38}$$

Daraus resultiert für die Ausgangsspannung zum Zeitpunkt $t_2$ innerhalb der Taktperiode $[n]$:

$$\begin{aligned} U_{o2}[n] = &\frac{C_L}{C_P + C_L} U_o[n-1] e^{-\frac{T/2}{\tau_1}} + \\ &+ \frac{C_P}{C_P + C_L}\left(U_{DC} + 2U_P\right) \end{aligned} \tag{39}$$

Im letzten Zeitabschnitt $t_2 \le t \le T$ werden die parallel geschalteten Kondensatoren $C_P$ und $C_L$ über die Last $R_L$ entladen. Die Zeitkonstante für diese Entladung ist $\tau_2 = R_L \cdot (C_L + C_P)$ und die Ausgangsspannung $U_o$ am Ende der Phase ergibt sich zu

$$U_o[n] = U_{o2}[n] e^{-\frac{T/2}{\tau_2}}. \tag{40}$$

Setzt man (39) in (40), so lässt sich die Ausgangsspannung in Abhängigkeit von den Kapazitäten $C_L$ und $C_P$, den Eingangsspannungen $U_{DC}$ und $U_P$, der Anfangsbedingung $U_o[n\text{-}1]$ und der Periodendauer $T$ berechnen:

$$U_o[n] = \left[ \begin{aligned} &\frac{C_L}{C_P + C_L} U_o[n-1] e^{-\frac{T/2}{\tau_1}} + \\ &+ \frac{C_P}{C_P + C_L}\left(U_{DC} + 2U_P\right) \end{aligned} \right] e^{-\frac{T/2}{\tau_2}} \tag{41}$$

Dabei ist $\tau_1 = R_L C_L$ und $\tau_2 = R_L (C_L + C_P)$. Bei geringer Belastung, also großem Lastwiderstand $R_L$ wird $\tau_1 >> T$ und $\tau_2 >> T$. In diesem Fall strebt die Ausgangsspannung am Ende der Taktperiode $[n]$ gegen den Wert:

$$\begin{aligned} U_o[n] = &\frac{C_L}{C_P + C_L} U_o[n-1] + \\ &+ \frac{C_P}{C_P + C_L}\left(U_{DC} + 2U_P\right) \end{aligned} \tag{42}$$

Mit zunehmender Periodenzahl $n$ resultiert daraus der idealisierte Endwert der Ausgangsspannung:

$$\lim_{n \to \infty} U_o = U_{DC} + 2U_P \tag{43}$$

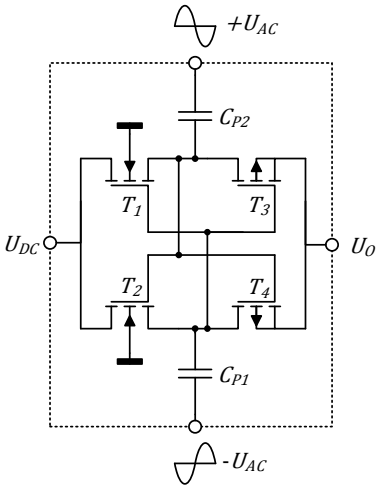



Abbildung 12: Symmetrische Ladungspumpe nach Mandal [17]. Die Phasen A und B werden durch NMOS- und PMOS-Schalter definiert und die Ansteuerung der Schalter erfolgt über das Signal.

*B. Realisierung*

In Abbildung 12 ist die Umsetzung der Ladungspumpe nach Mandal [8] mit 4 Transistoren und 2 Kapazitäten $C_P$ zu sehen. Im Gegensatz zur Prinzipschaltung besteht diese aus doppelt so vielen Schaltern, wodurch die Wechselspannung symmetrisch genutzt wird und sich die Pulszahl an $C_L$ verdoppelt. Um größere Ausgangsspannungen zu erreichen, kann die Ladungspumpe kaskadiert werden, wobei allerdings der Wirkungsgrad abnimmt. Die benötigte Gegenphasigkeit der Schalter erfolgt durch den Einsatz von NMOS- respektive PMOS-Transistoren. Zur potentialfreien Ansteuerung der Transistoren und gleichzeitig als Pumpkapazitäten dienen $C_{P1}$ und $C_{P2}$.

Aus der Art und Weise, wie die Ladungspumpe eingesetzt wird, resultieren Nichtidealitäten. Zunächst einmal handelt es sich bei der Ansteuerspannung nicht um eine Rechteckspannung, sondern um die Sinusspannung des Oszillators. Damit die Ladungspumpe auch mit einer Sinusspannung funktioniert, müssen die Schaltphasen der Transistoren verkürzt sein, da ansonsten die Pump- beziehungsweise der Lastkondensator nicht nur geladen, sondern auch wieder entladen werden. Grund hierfür ist, dass die Kondensatoren aufgrund der abfallenden Sinusspannung wieder entladen werden, wenn die Transistoren nicht bereits an Scheitelpunkt wieder sperren. Dadurch sinkt der Wirkungsgrad bei größeren Signalamplituden. Auch zu kleinen Signalamplituden hin sinkt der Wirkungsgrad. Ursache dafür ist die Abhängigkeit der Durchgangswiderstände der Transistoren von der momentanen Signalspannung $U_{AC}$. Folglich werden die Kondensatoren bei zu kleiner Signalamplitude nicht mehr ausreichend geladen.

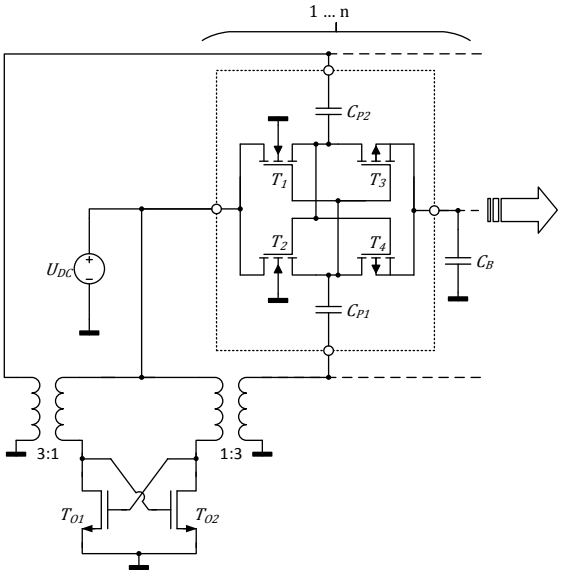



Abbildung 13: Gesamtschaltung, bestehend aus Oszillator mit Transformatorauskopplung und 5-stufiger Ladungspumpe (nur erste Stufe dargestellt).

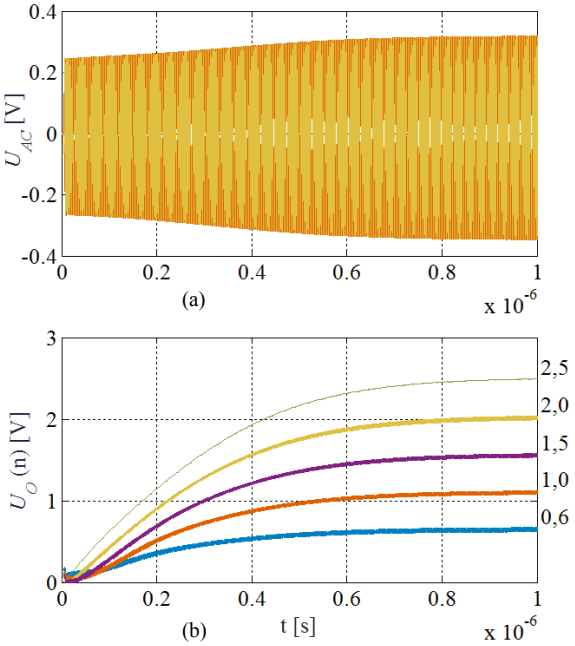



Abbildung 14: Einschaltvorgang des Systems, (a) Zeitlicher Verlauf der Differenzspannung am Oszillatorausgang, (b) Zeitlicher Verlauf der Ausgangsspannungen der 5-stufigen Ladungspumpe.

Aufgrund von Vorgaben, die sich aus der eingesetzen Technologie ergeben, resultieren weitere Nichtidealitäten:

- Das Bulk der NMOS-Transistoren muss auf Ground liegen, wodurch der Body-Effekt mit steigender Ausgangsspannung den Durchschaltewiderstand $R_{ON}$ der NMOS-Transistoren erhöht.
- Bei zu großen Amplituden kann die Bulk-Diode der NMOS-Transistoren leitend werden. Dies ist allerdings nicht hinderlich, sondern wirkt sich eher unterstützend aus.
- Das Bulk der PMOS-Transistoren kann auf ein beliebiges Potential gelegt werden. Wegen der Hochfrequenz an $C_{P1}$ und $C_{P2}$ muss das Bulk mit $U_o$ verbunden werden, um die kapazitiven Verluste zum Substrat hin zu vermeiden.
- Die PMOS-Bulk-Diode kann leitend werden, was aber in dieser Anwendung ebenfalls nicht schadet.

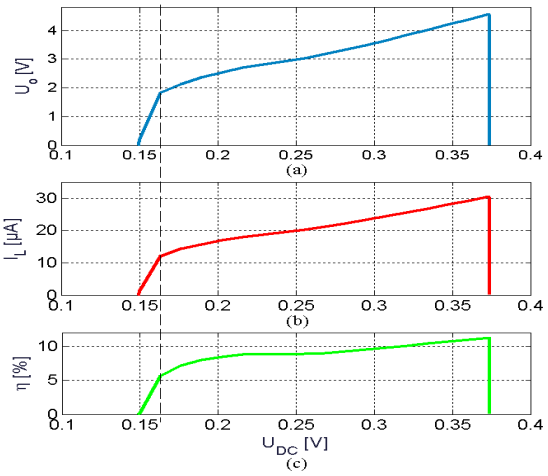



Abbildung 15: Stationärer Betrieb, (a) Ausgangsspannung, (b) Laststrom, (c) Wirkungsgrad.

## V. Gesamtsystem

Die Zusammensetzung der Gesamtschaltung, bestehend aus Oszillator, Transformatorauskopplung und Ladungspumpe ist in Abbildung 13 dargestellt. Da die Ausgangsspannung der Ladungspumpe unter Belastung zurückgeht, ist eine Kaskadierung von fünf Ladungspumpen erforderlich, um die Anforderungen im Gesamtsystem zu erfüllen. Die Simulationsergebnisse des Einschwingvorgangs im Zeitbereich sind in Abbildung 14 visualisiert. Dabei sind zum einen die über den Transformator ausgekoppelte Ausgangsspannung des Oszillators dargestellt und zum anderen die Ausgangsspannungen der einzelnen Stufen der Ladungspumpe. Zur Ermittlung dieser Werte wurden eine Eingangsspannung $U_{DC}$ = 0,2 V, ein Lastwiderstand $R_L$ = 150 kΩ und eine Lastkapazität $C_L$ = 10 pF verwendet. Aufgrund der größeren Belastung während der Anlaufphase der Ladungspumpen ist die Oszillatorspannung zu Beginn geringer und nimmt mit der Zeit leicht zu. Die eigentliche Einschwingzeit des Oszillators liegt unter 20 ns. Mit Ausnahme der ersten Stufe erhöhen alle weiteren Stufen die Spannung um etwa den gleichen Betrag. Ursächlich für die größere Erhöhung der ersten Stufe ist, dass an ihrem Eingang die Spannung $U_{DC}$ anliegt und an den Eingängen der weiteren Stufen jeweils die Ausgangsspannung der vorhergehenden Stufe.

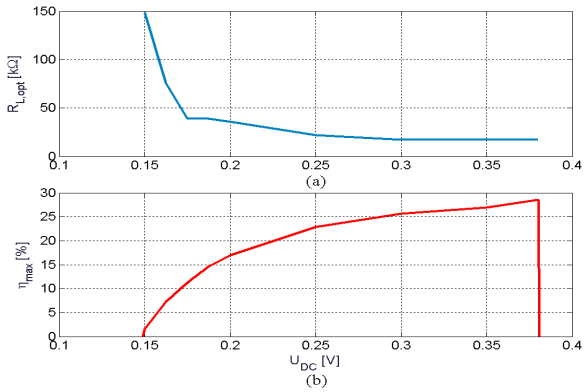



Abbildung 16: (a) Lastwiderstand für Leistungsanpassung, (b) Wirkungsgrad bei Leistungsanpassung.

Bei gleicher Belastung zeigt Abbildung 15 die Abhängigkeit der Ausgangsspannung, des Laststroms und des Wirkungsgrads von der Eingangsspannung. Jeder Punkt in diesem Diagramm ist das Ergebnis einer Transient-Simulation bis zum stationären Zustand und deren Auswertung über mehrere Perioden. Die gestrichelte Linie markiert den Punkt, bei dem die $U_o = 1{,}8$ V und $I_L = 10$ µA aus der Zielspezifikation erreicht werden. Der Wirkungsgrad beträgt dabei $\eta = 6$ %, allerdings werden diese Werte erst bei einer Eingangsspannung $U_{DC} = 170$ mV erreicht. Das anvisierte Ziel wurde somit um 20 mV verfehlt. Nach oben hin ist die zulässige Eingangsspannung auf ca. 370 mV begrenzt, da bei höheren Spannungen im Oszillator die Schwingbedingung nicht mehr erfüllt werden kann. Grund dafür ist die fehlende Stromquelle. Um den maximalen Wirkungsgrad des Systems zu bestimmen, wurde eine Leistungsanpassung für verschiedene Eingangsspannungen durchgeführt. Die Ergebnisse sind in Abbildung 16 dargestellt. Aus den Graphen geht hervor, dass der maximale Gesamtwirkungsgrad bei $\eta_{max} = 27$ % liegt, was einen guten Wert für einen integrierten Wandler darstellt. Tabelle 3 zeigt eine detaillierte Zusammenstellung der durch Simulation ermittelten Ergebnisse.

In Abbildung 17 ist das Layout dargestellt. Den mit Abstand größten Platzbedarf haben die beiden Transformatoren, welche sich links und rechts befinden. Im Gegensatz dazu sind die fünf Stufen der Ladungspumpe im Zentrum beinahe vernachlässigbar. Unterhalb der Transformatoren befinden sich die beiden Transistoren $M_1$ und $M_2$ der Differenzstufe des Gegentaktoszillators, welche trotz der großen Weite $W$ eine vergleichsweise geringe Fläche einnehmen.

## VI. Zusammenfassung und Ausblick

Für die autarke Versorgung von ASICs mittels Energy Harvesting bei geringen Temperaturdifferenzen wurde ein vollständig integrierbarer Aufwärtswandler entwickelt. Der Wandler beruht auf einer

Tabelle 3: Simulationsergebnisse.

| | |
|---|---|
| Betriebsspannungsbereich | $U_{DC} = 150$-$350$ mV |
| Eingangsspannung ($I_L = 10$ µA) | $U_{DC} = 170$ mV |
| Ausgangsleistung ($U_{DC} = 170$ mV, $U_o = 1{,}8$ V) | $P = 18$ µW |
| Maximaler Laststrom ($U_{DC} = 200$ mV, $U_o = 1{,}8$ V) | $I_{L,max} = 50$ µA |
| Wirkungsgrad ($U_{DC} = 200$ mV, $U_o = 1{,}8$ V, $I_{L,max} = 50$ µA) | $\eta = 17$ % |
| Maximale Ausgangsleistung ($U_{DC} = 350$ mV, $U_o = 2{,}5$ V) | $P_{max} = 400$ µW |
| Maximaler Wirkungsgrad ($U_{DC} = 350$ mV, $U_o = 2{,}5$ V) | $\eta_{max} = 27$ % |
| Chipfläche | $A = 2{,}1$ x $0{,}9$ mm$^2$ |

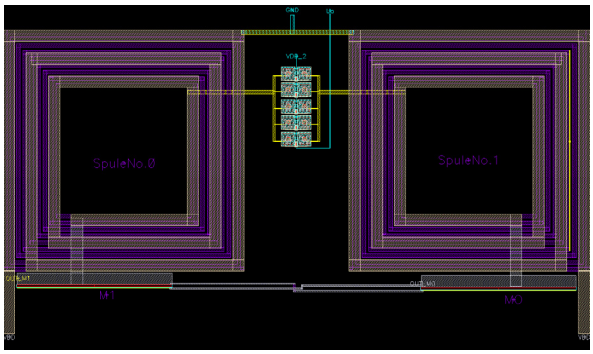



Abbildung 17: Layout der Gesamtschaltung.

Ladungspumpe, die über einen differentiellen Taktgenerator betrieben wird. Dieser besteht aus einer kreuzgekoppelten Differenzstufe mit LC-Oszillator und Transformatorauskopplung und kommt ohne Stromquelle aus.

Ladungspumpe und Oszillator wurden symbolisch analysiert und mit Hilfe umfangreicher Simulationen dimensioniert. Für die planaren Spulen der Transformatoren wurden elektrische Ersatzschaltbilder zur Simulation ihres Hochfrequenzverhaltens und der Verluste angesetzt und parametrisiert. Schließlich wurde ein Layout für den 180 nm CMOS-Prozess der Fa. UMC erstellt.

Die Simulationsergebnisse lassen erwarten, dass der vorliegende IP, eingebettet in ein ASIC, dessen Betrieb mit Versorgungsspannungen ermöglicht, die noch unterhalb der Schwellenspannung der Transistoren liegen. Für den Betrieb einer Nutzlast mit 1,8 V

und 10 µA wird eine Betriebsspannung von 170 mV benötigt. Diese Ergebnisse sind nunmehr an der Hardware zu verifizieren. Noch kleinere Betriebsspannungen sollten mit einem noch moderneren Prozess, insbesondere aufgrund der kleineren kapazitiven Verluste im Leiterbahnsystem, möglich sein. Die Grundlagen für eine entsprechende Weiterentwicklung konnten mit dem vorliegenden Beitrag gelegt werden.

## DANKSAGUNG

Die Autoren bedanken sich bei Herrn Alexander Höffler für die Durchführung der Layout-Arbeiten.

## LITERATURVERZEICHNIS

[1] I. Doms et al., “Integrated capacitive power-management circuit for thermal hervesters with output power 10 to 1000 µW,” *IEEE ISSCC Dig. Tech. Papers*, pp. 300 – 301, 2009.

[2] H. Shao et al., “The design of a micropower management system for applications using photovoltaic cells with the maximum output power control,” *IEEE Trans VLSI Syst.*, vol. 17, pp 1138 – 1142, 2009.

[3] J. Kim et al., “A regulated charge pump with a low-power integrated optimum power point tracking algorithm for indoor solar energy harvesting,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol 58, pp. 802 – 806, 2011.

[4] M. Dini et al., “A 40 nA/source energy harvesting power converter for multiple and heterogeneous sources,” *Proc. ESSCIRC*, pp. 259 – 262, Venice Sep. 2014.

[5] LTC3108 Datasheet, Linear Technology, Aug. 2013.

[6] J. Rechtsteiner, G. Forster, „Ein Energy Harvesting IP für den Einsatz in einem 0,18 µm CMOS ASIC, *Tagungsband Workshop der Multiprojektchip-Gruppe*, Offenburg 2012, ISSN 1868-9221.

[7] J. Kim et al., „A 0.15 V input energy-harvesting charge pump with switching body biasing and adaptive dead-time for efficiency improvement,“ *IEEE ISSCC Dig. Tech. Papers*, pp. 394 – 396, 2014.

[8] J. P. Im et al., „A 40 mV Transformer-Reuse Self-Startup Boost Converter With MPPT Control for Thermoelectric Energy Harvesting,“ *JSSC*, vol 47, no. 12, pp. 3055 – 3067, Dec. 2012.

[9] P. S. Weng et al., „50 mV-Input Batteryless Boost Converter for Thermal Energy Harvesting,“ JSSC, vol. 48, no. 4, pp 1031 – 1041, 2013.

[10] H. Fuketa et al., “An 85-mV Input, 50 µs Startup Fullly Integrated Voltage Multiplier with Passive Clock Boost Using On-Chip Transformers for Energy Harvesting,” *Proc. ESSCIRC*, pp. 263 – 266, Venice Sep. 2014.

[11] G. Forster, "Mikroelektronische Schaltungen," Vorlesungsmanuskript, Hochschule Ulm, 2013.

[12] H. Gan, "On-Chip Transformer Modeling, Characterization, and Applications in Power and Low Noise Amplifiers," PhD thesis, Stanford University, 2006.

[13] S. S. Mohan, "The Design, Modeling and Optimization of On-Chip Inductor and Transformer Circuits," PhD Thesis, Stanford University, 1999.

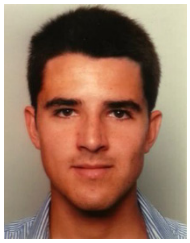
[14] H. M. Greenhouse, "Design of Planar Rectangular Microelectronic Inductors," *IEEE Transactions on Parts, Hybrids, and Packaging*, vol. PHP-10, no. 2, pp. 101ff, Jun. 1974.

[15] F. W. Grover, *Inductance Calculations*. New York, N.Y.: Dover Publications, 1962.

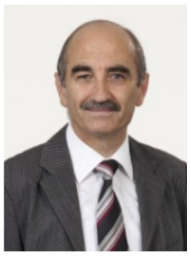
[16] S. S. Patrick Yue, "Physical Modeling of Spiral Inductors on Silicon," *IEEE Transactions on Electron Devices*, vol. 47, no. 3, 2000.

[17] R. S. Mandal, "Low-power CMOS rectifier design for RFID applications," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 54, no. 6, pp. 1177-1188, Jun. 2007.

Michael Hiller erhielt den akademischen Grad des B.Eng. in Elektrotechnik im Jahr 2010 von der DHBW. Seit 2014 studiert er an der Hochschule Ulm im Studiengang Systems Engineering and Management mit der Vertiefungsrichtung Electrical Engineering. Von 2010 bis heute ist er außerdem als Ingenieur im R&D Bereich bei der Advantest Europe GmbH tätig. Themengebiete sind dabei FPGA Design, Embedded Software, automatisierte Codegenerierung und die Automatisierung von RF-Messungen mit Hilfe von Algorithmen.



Mark Locherer studiert seit 2011 Nachrichtentechnik an der Hochschule Ulm. Er absolviert sein duales Studium nach dem Ulmer Modell in Kooperation mit der Netze BW GmbH. Im Rahmen seiner Studienarbeit hat er sich mit dem Thema Energy Harvesting, speziell mit der Entwicklung und Optimierung einer Ladungspumpe, befasst.



Gerhard Forster studierte Physik mit dem Schwerpunkt Quantenelektronik an der Universität Heidelberg. Nach seinem Diplom-Abschluss 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.

# Flächenoptimierte Bandgap-Referenz für Low-Power-Anwendungen mit 2,5 – 5,5 V Versorgung

Ismail Yasar, Robin Staudt, Cedric Leonel Jiago Teffo, Benjamin Schoch, Thomas Stoof,
Jürgen Wittmann, Bernhard Wicht

***Zusammenfassung*—In dieser Arbeit wird eine optimierte Bandgap-Referenz zur Erzeugung einer temperaturstabilen Spannung und eines Referenzstroms vorgestellt. Für Low-Power-Anwendungen wurde die Bandgap-Referenz, basierend auf der Brokaw-Zelle, mit minimaler Stromaufnahme und optimierter Chipfläche durch Multi-Emitter-Layout der Bipolartransistoren implementiert. Zusätzliches Merkmal ist ein verbreiteter Versorgungsspannungsbereich von 2,5 bis 5,5 V. Simulationen zeigen, dass eine stabile Ausgangsspannung von 1,218 V und ein Referenzstrom von 1,997 µA realisiert wird. Im Temperaturbereich -40 °C … 50 °C sowie dem gesamten Bereich der Versorgungsspannung beträgt die Genauigkeit der Referenzspannung ± 0,04 % mit einer Gesamtstromaufnahme zwischen 3,5 und 10 µA. Es wird eine Temperaturdrift von 2,18 ppm/K erreicht. Durch das elektronische Trimmen von Widerständen wird der Offset der Ausgangsspannung, bedingt durch Herstellungstoleranzen, auf ±3,5 mV justiert. Die Referenz wird in einer 0,18 µm BiCMOS-Technologie implementiert.**

***Schlüsselwörter*—Bandabstandreferenz, Bandgap Voltage Reference, Bandgap, Spannungsreferenz, Low Power.**

## I. Einleitung

Viele integrierte Schaltungen benötigen eine präzise Referenz, um deren Funktionalität zu gewährleisten. Im digitalen Bereich gehören hierzu beispielsweise Analog-Digital-Converter (ADC). Diese wandeln analoge Spannungen in einen digitalen Wert um. Dabei müssen ADCs „wissen" wie groß eine digitale Einheit ist. Das wird durch den Vergleich mit einer Spannungsreferenz erreicht. Erst dadurch ist es möglich, Pegel für digitale Vorgänge zu definieren. Aber auch in analogen Systemen kommen sie zum Einsatz. Wird ein Spannungsregler betrachtet, so vergleicht dieser seine Ausgangsspannung über einen Fehlerverstärker ebenfalls mit einer Referenzspannung. Deshalb ist es dem Regler möglich, auf eine Änderung am Ausgang angemessen zu reagieren. Die genannten Bauelemente und dadurch auch Spannungsreferenzen finden ebenfalls im Automotive-Bereich ihre Anwendung. Sie kommen in Steuergeräten, Sensorapplikationen und im Multi-Media-Bereich vor. Da die genannten und weiteren Bauelemente zwingend eine Referenz voraussetzen, muss die Funktionalität der Bandgap-Referenz zu jeder Arbeitsbedingung gegeben sein. Durch diese Abhängigkeiten von der Referenz hat nicht nur dessen Funktionalität, sondern auch die Genauigkeit direkten Einfluss auf die Gesamtsysteme. Deshalb sind Genauigkeit und Fehleranfälligkeit der Bandgap-Referenz von großer Wichtigkeit. Die fortschreitend kleiner werdenden Schaltungsausmaße und die auf Effizienz ausgelegten Systeme stellen dabei besondere Spezifikationen. Die Herausforderung hierbei ist es, trotz geringer Ausmaße und der dadurch bedingten Herstellungstoleranzen sowie der geringen Stromaufnahmen eine dennoch stör- und temperaturunempfindliche Ausgangsspannung zu erhalten. Des Weiteren wird die hier vorliegende Optimierung für den Automotive-Bereich betrachtet, dessen typische Anforderungen ein Temperaturbereich von -40 °C bis 150 °C und ein Niederspannungsbereich von 2,5 V bis 5,5 V sind. Der weite Spannungsbereich ist erforderlich, da bei einem Einbruch der Batteriespannung, z.B. bei einem Kaltstart des Fahrzeugs, die Niederspannungsdomäne und somit die Versorgung der Bandgap-Referenz deutlich absinkt. Für Bandgap-Referenzen für den Automotive-Bereich existieren bereits Entwicklungen [1], die zwar auf eine hohe Genauigkeit ausgelegt sind, aber deren Stromaufnahme für Low Power Anwendungen nicht akzeptabel ist.

Spannungsreferenzen können auf verschiedenste Art und Weise entworfen und hergestellt werden. Im Jahr 1971 entwickelte Robert Widlar die erste Bandgap-Referenz, das LM113. Hierfür verwendete er die herkömmliche junction-isolated Bipolar-IC Technologie, um mit Bipolartransistoren eine stabile Ausgangsspannung (~1,220 V) zu erzeugen [2]. Weiterhin kommen heutzutage für Referenzen auch Buried-Zener- und XFET-Prinzipien zum Einsatz [4][5][6]. Aber um eine ausreichend hohe Präzision zu gewährleisten, den Flächen- und Energiebedarf zu reduzieren und die Möglichkeiten der BiCMOS-Technologie auszunutzen,

Ismail Yasar, Robin Staudt, Cedric Leonel Jiago Teffo, Benjamin Schoch, Thomas Stoof, {ismail.yasar, robin.staudt, cedric_leonel.jiago_teffo, benjamin.schoch, thomas.stoof}@student.reutlingen-university.de, Jürgen Wittmann, juergen.wittmann@reutlingen-university.de and Bernhard Wicht, bernhard.wicht@reutlingen-university.de are with Hochschule Reutlingen, Alteburgstraße 15, 72762 Reutlingen.

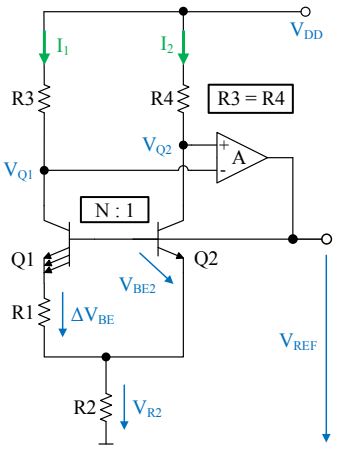



Abbildung 1: Prinzip einer Brokaw-Zelle.

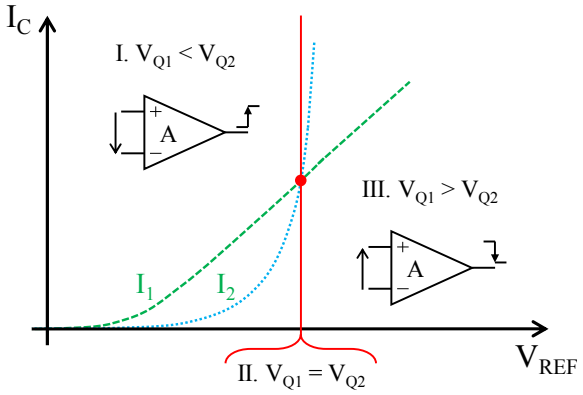



Abbildung 2: Stromverlauf am Kollektor von $Q_1$ und $Q_2$.

wird deshalb auf das Prinzip der Brokaw-Zelle in Abbildung 1 zurückgegriffen [7]. Bei dieser Art werden in zwei Zweigen exakt dieselben Ströme erzeugt, welche proportional zur Temperatur sind. Dieser Strom sorgt dafür, dass an Basis-Emitter-Strecken integrierter Bipolartransistoren und an integrierten Polysiliziumwiderständen eine Spannung abfällt. Durch die Kombination dieser beiden über der Temperatur veränderlichen Spannungen wird eine stabile Ausgangsspannung realisiert, da beide zwar betragsmäßig denselben Temperaturkoeffizienten (TK) besitzen, aber mit unterschiedlichen Vorzeichen behaftet sind. Der Vorteil einer Brokaw-Zelle ist der einfache Grundaufbau aus Bipolartransistoren und Widerständen.

Das Ziel dieser Arbeit ist es, eine hochgenaue Bandgap-Referenz basierend auf einer Brokaw-Zelle zu implementieren. Die Toleranz der Referenzspannung soll dabei über Temperaturbereich, Prozesstoleranzen und Schwankungen der Versorgungsspannung deutlich unter ±10 mV liegen, bei gleichzeitig minimaler Fläche. Abgeleitet von der Referenzspannung mit einer Widerstandslast soll zusätzlich ein Referenzstrom von 2 µA erzeugt werden. Bedingt durch den vorgesehenen Einsatz in Low-Power Anwendungen, wird eine Gesamtstromaufnahme der Bandgap-Referenz von weniger als 10 µA gefordert.

## II. Einführung Brokaw-Zelle

Abbildung 1 verdeutlicht den Grundaufbau einer Brokaw-Zelle. Die Widerstände $R_3$ und $R_4$ sind gleich. Die Emitterfläche von $Q_1$ ist um den Faktor $N$ größer als die Fläche des Bipolartransistors $Q_2$. Dies wird erreicht, indem $Q_1$ aus mehreren parallel geschalteten Bipolartransistoren realisiert wird und somit die Einsatzspannung von $Q_1$ niedriger ist als von $Q_2$. Ist die Spannung $V_{REF}$, und somit die Basisspannungen von $Q_1$ und $Q_2$, zu niedrig, zum Beispiel im Einregelvorgang, so fließt aufgrund der geringeren Einsatzspannung ein größerer Strom durch $Q_1$ (Abbildung 2, I.). Diese unsymmetrische Verteilung der Ströme verursacht unterschiedliche Kollektorspannungen $V_{Q1} < V_{Q2}$. Das treibt den Operationsverstärker (OP) an, durch Erhöhen von $V_{REF}$ diesen Unterschied auszugleichen. Dadurch erhöhen sich wiederum die jeweiligen Zweigströme und der Arbeitspunkt wandert in Abbildung 2, I. nach rechts. Der Strom $I_1$ durch $Q_1$ jedoch wird durch den Widerstand $R_1$ begrenzt, wodurch der Strom $I_2$ durch $Q_2$ stärker ansteigt. Ein stabiler Arbeitspunkt ist erreicht, wenn die beiden Ströme $I_1$ und $I_2$, und somit $V_{Q1}$ und $V_{Q2}$ gleich sind (Abbildung 2, II.). Ist die Spannung $V_{REF}$ höher als der erstrebte Wert und somit $V_{Q1} > V_{Q2}$ (Abbildung 2, III.), so verhält es sich umgekehrt. Der OP verringert $V_{REF}$, bis wiederum der stabile Arbeitspunkt erreicht wird. Mit der Gleichheit der beiden Ströme durch $Q_1$ und $Q_2$ ergibt sich eine Stromdichtenverteilung von $N$ zu 1, gemäß dem Verhältnis der Emitterflächen. Diese Verteilung hat zur Folge, dass die Spannungen $V_{BE1}$ und $V_{BE2}$ unterschiedlich sind. Deren Differenz $\Delta V_{BE}$ ist die Spannung über $R_1$. Somit ergibt sich [7]

$$\Delta V_{BE} = V_{BE2} - V_{BE1} = V_{R1} = \frac{kT}{q} \ln\left(\frac{N \cdot I_2}{I_1}\right). \quad (1)$$

Dabei ist $k$ die Boltzmann-Konstante, $T$ die absolute Temperatur und $q$ die Elementarladung. Mit

$$I_1 = I_2 = \frac{\Delta V_{BE}}{R1} \quad (2)$$

können die beiden Zweigströme $I_1$ und $I_2$ über den Widerstand $R_1$ und das Verhältnis $N$ eingestellt werden. Aus Gleichung (1) und (2) ist zu erkennen, dass $I_1$ und $I_2$ proportional zur absoluten Temperatur sind ($I_{PTAT}$). Der Strom durch $R_2$ entspricht der Summe aus $I_1$ und $I_2$. Damit ist die Spannung über $R_2$

$$V_{R2} = 2 \cdot \frac{R_2}{R_1} \cdot \frac{kT}{q} \ln\left(\frac{N \cdot I_2}{I_1}\right). \quad (3)$$

Somit ist auch $V_{R2}$ proportional zur absoluten Temperatur. Für den TK von $V_{BE2}$ gilt [8]

$$\frac{\partial V_{BE2}}{\partial T} \approx -2 \cdot 10^{-3} \frac{\mathrm{V}}{\mathrm{K}}. \quad (4)$$

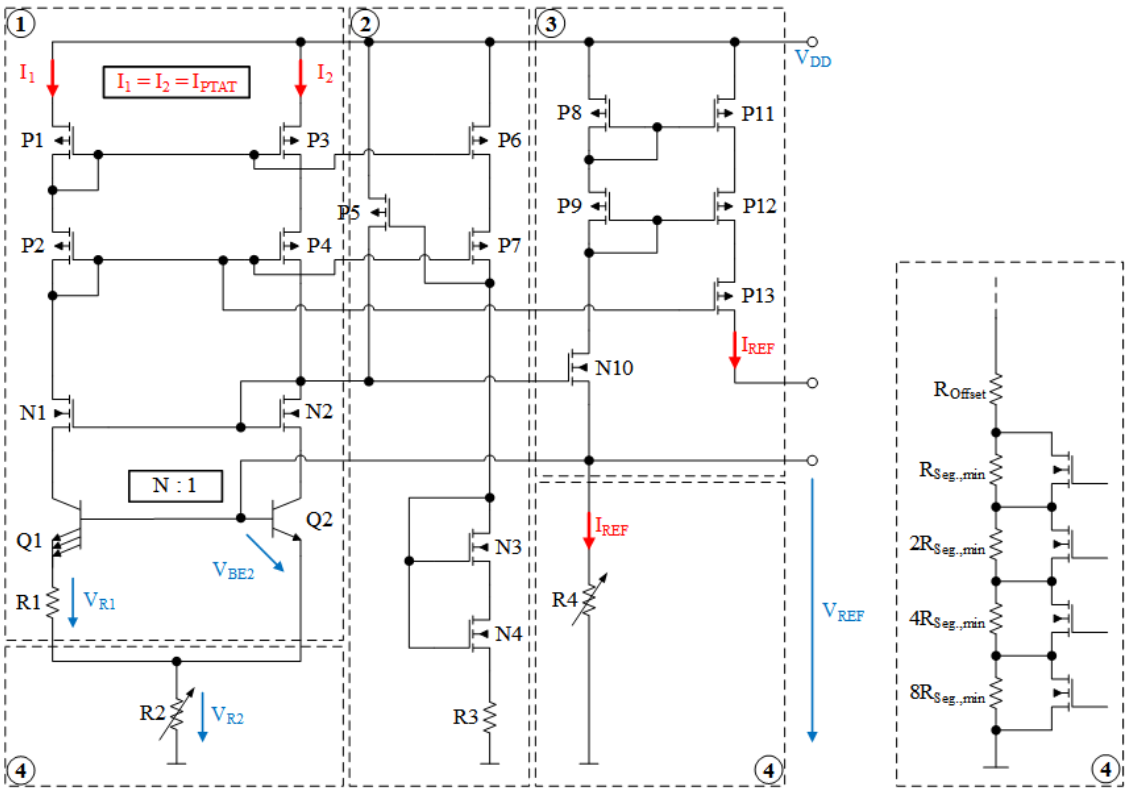



Abbildung 3: Schaltungsaufbau der entwickelten Bandgap-Referenz.

Damit $V_{REF}$ temperaturunabhängig ist, müssen sich die TK von $V_{BE2}$ und $V_{R2}$ aufheben, womit folgendes gelten muss:

$$\frac{\partial V_{REF}}{\partial T} = \frac{\partial V_{BE2}}{\partial T} + \frac{\partial V_{R2}}{\partial T} = 0\,. \tag{5}$$

Für den TK von $V_{R2}$ gilt hingegen

$$\frac{\partial V_{R2}}{\partial T} = 2 \cdot \frac{k}{q} \cdot \frac{R_2}{R_1} \cdot \ln(N). \tag{6}$$

Mit $2 \cdot k/q = 0{,}172 \cdot 10^{-3}$ V/K und $I_1 = I_2$ kann über das Verhältnis von $R_2/R_1$ und dem Verhältnis $N$ der Emitterflächen der TK von $V_{BE2}$ kompensiert werden, woraus sich eine Referenzspannung $V_{REF}$ ergibt, die konstant über der Temperatur ist. In Gleichung (6) ist ersichtlich, dass durch das Verhältnis von $R_2/R_1$ sich deren Herstellungstoleranzen und TKs aufheben.

Der Wert, welcher sich nun an $V_{REF}$ einstellt, ist vom verwendeten Halbleitermaterial abhängig. Bei Silizium beläuft sich der Wert auf ca. 1,2 V, welches in erster Näherung dem Wert der Bandabstandsenergie von Silizium (~ 1,12 eV) entspricht. Dieser Zusammenhang ist durch den Temperaturverlauf der Basis-Emitter-Spannung zu erklären. Wenn die Temperaturverläufe von $V_{BE1}$ und $V_{BE2}$ auf den absoluten Nullpunkt extrapoliert werden wie in Abbildung 4 skizziert, liegen die Schnittpunkte beider Geraden mit der Ordinate bei ungefähr 1,2 V [9].

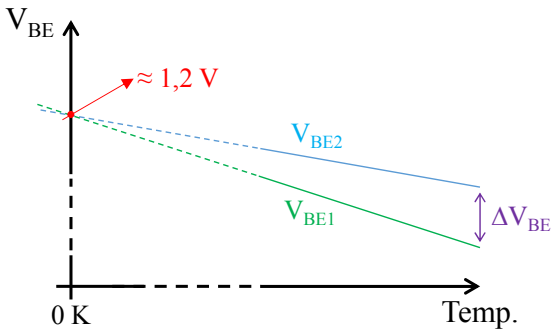



Abbildung 4: Extrapolation der Basis-Emitter-Spannung.

## III. Entwurf der Bandgap-Referenz

Die optimierte Bandgap-Referenz basiert in dieser Arbeit auf der in Kapitel II erläuterten Brokaw-Zelle. Abbildung 3 zeigt die Gesamtschaltung, unterteilt in verschiedene Teilschaltungen. Die Brokaw-Zelle ist in Teilschaltung ① zu sehen, die zugehörige Startschaltung in Teilschaltung ②. Die Beschaltung zum Erzeugen des Referenzstroms befindet sich in ③. Die elektronisch trimmbaren Widerstände werden zusammengefasst als Ersatzschaltbild in ④ dargestellt.

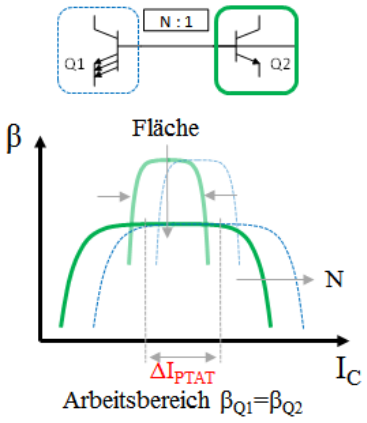



Abbildung 5: Verlauf der Verstärkung $\beta$ in Abhängigkeit des Kollektorstroms $I_C$. Zusätzlich wird der Einfluss der Bipolartransistorfläche und -anzahl ($N$) dargestellt.

### *A. Schaltungsentwurf*

Der Stromspiegel setzt sich aus den Transistoren $P_1$ und $P_3$ zusammen und die dazugehörige Kaskode besteht aus $P_2$ und $P_4$. Diese werden in Abbildung 3, Teilschaltung ① dargestellt. Da das Prinzip der Brokaw-Zelle die Gleichheit der beiden Ströme $I_1$ und $I_2$ zwingend voraussetzt, ist hierzu ein gutes Matching des Stromspiegels $P_1$ und $P_3$ nötig. Deshalb setzt sich dieser aus Niedervolttransistoren zusammen, die sowohl eine deutlich geringere Einsatzspannung als auch ein wesentlich besseres Matchingverhalten aufweisen. Dadurch ist eine Funktion der Bandgap-Referenz bei niedrigeren Einsatzspannungen gegeben und die Fläche des Stromspiegels kann verringert werden, bei gleichbleibender Matchinggenauigkeit. Die Kaskode besteht aus den Hochvolttransistoren $P_2$ und $P_4$. Damit wird zum einen die Genauigkeit des Stromspiegels erhöht, indem die Spannung $V_{DS}$ von $P_1$ und $P_3$ konstant gehalten wird. Zum anderen wird auch der Einsatz bei höheren Versorgungsspannungen gewährleistet, da eine zu hohe Spannung an $P_1$ und $P_3$ durch die Kaskode verhindert wird.

In Teilschaltung ① ist eine zusätzliche Kaskode zu erkennen, bestehend aus den Transistoren $N_1$ und $N_2$. Diese stabilisieren die Spannung an den Kollektoren von $Q_1$ und $Q_2$ und halten sie auf demselben Potential. Durch die zusätzliche Gleichheit der beiden Ströme $I_1$ und $I_2$ wird gewährleistet, dass die Stromdichten $J_{Q1}$ und $J_{Q2}$ unterschiedlich sind und sich damit ein $\Delta V_{BE}$ bildet, siehe hierzu Gleichung (1). Die Kombination aus $N_1$, $N_2$ und $P_{10}$ in Teilschaltung ③ ersetzt den OP aus der Grundschaltung der Brokaw-Zelle in Abbildung 1. Ändert sich der Strom eines Zweiges, wird zwar die Spannung über $Q_1$ bzw. $Q_2$ konstant gehalten, aber die Spannung über der Kaskode variiert. Diese Änderung der Spannung über der Kaskode steuert $P_{10}$ an, wodurch dieser die Spannung an der Basen von $Q_1$ und $Q_2$ ändert. Bei der Dimensionierung der Kaskode $N_1$ und $N_2$ werden eine minimale Länge und eine große Weite festgelegt. Dabei wird die Weite soweit erhöht, bis die Overdrive-Spannung $V_{OVD}$ über der Kaskode einen geringen Einfluss auf den Eingangsspannungsbereich hat. $V_{OVD}$ ist definiert durch

$$V_{OVD} = V_{GS} - V_{TH} . \tag{7}$$

Dabei ist $V_{GS}$ die Gate-Source-Spannung und $V_{TH}$ die Schwellenspannung des jeweiligen MOSFETs.

Die Stromdichten $J_{Q1}$ und $J_{Q2}$ unterscheiden sich, wenn sich die Emitterflächen der Transistoren unterscheiden. Dies wird durch eine Anzahl parallelgeschalteten Bipolartransistoren erreicht. Das Emitterflächenverhältnis wird zu 6:1 gewählt. An dieser Stelle lässt sich durch Variieren des Verhältnisses die Fläche der gesamten Schaltung optimieren. Für die Genauigkeit der Bandgap-Referenz spielt die Schwankung von $V_{R1}$ eine wichtige Rolle. Für $V_{R1}$ gilt neben Gleichung (1) auch

$$V_{R1} = V_{BE2} - V_{BE1} = V_T \cdot \ln\left(\frac{I_{C2} \cdot I_{S1}}{I_{C1} \cdot I_{S2}}\right). \tag{8}$$

Dabei gilt $I_{C1} = I_{C2} = I_{PTAT}$ und $V_T$ ist die Temperaturspannung. Anhand Gleichung (8) ist zu erkennen, dass $V_{R1}$ bzw. $\Delta V_{BE}$ abhängig vom Sättigungsstrom $I_S$ ist. $I_S$ ist gegeben durch [9]

$$I_S = \frac{q \cdot A \cdot D_n \cdot {n_i}^2}{N_A \cdot W_B} . \tag{9}$$

Dabei ist $q$ die Elementarladung, $D_n$ die Diffusionskonstante, $n_i$ die intrinsische Ladungsträgerkonzentration, Dotierkonzentration in der Basis $N_A$ und die Weite der Basis $W_B$. Diese Parameter sind Prozessbedingt und werden als konstant angenommen. Wird nun Gleichung (9) in (8) eingesetzt, und beachtet, dass $\beta$ eine Funktion von $N_A$ und $W_B$ ist, erhalten wir die Beziehung

$$V_{R1} \propto \ln\left(\frac{\beta_1}{\beta_2}\right). \tag{10}$$

Die Spannung $V_{R1}$ ist daher abhängig von dem Verhältnis der Stromverstärkungen $\beta_1$ und $\beta_2$ der beiden Transistoren $Q_1$ und $Q_2$.

Da beide Transistoren bei unterschiedlichen Stromdichten arbeiten, muss eine genaue Betrachtung der Abhängigkeit zwischen Stromdichte und Stromverstärkung erfolgen. Abbildung 5 zeigt die $\beta$-Verläufe in Abhängigkeit der Kollektorströme $I_C$ der beiden Bipolartransistoren. Für eine erhöhte Anzahl von parallelen Transistoren verschiebt sich die Kurve zu größeren Kollektorströmen. Ändert sich die gesamte Fläche des Bipolartransistors, so variiert die Verstärkung $\beta$. Für große Verstärkungen wird der Bereich kleiner, in dem die Verstärkung konstant ist. Um das Verhältnis $\beta_1/\beta_2 = 1$ zu erreichen, müssen die Transistorgrößen so gewählt werden, dass die Stromverstärkungen von $Q_1$ und $Q_2$ über den spezifizierten Bereich, in welchem sich $I_{PTAT}$ bewegt, gleich sind.

Die Verstärkung beeinflusst die beiden Ströme $I_1$ und $I_2$. Da die Basisströme der Transistoren unterschiedlich

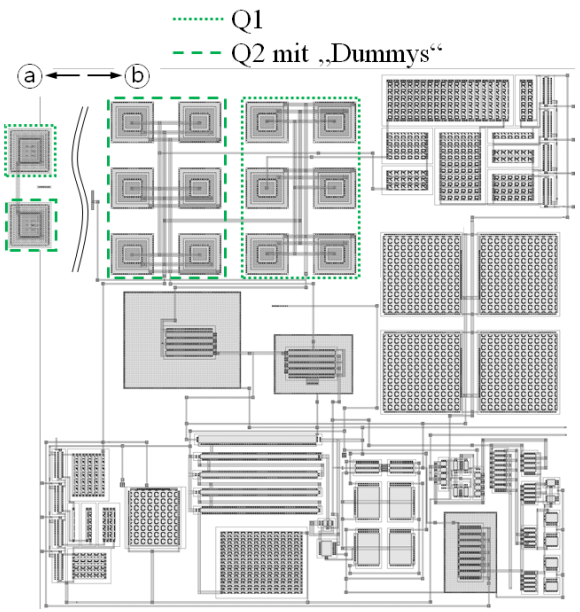



Abbildung 6: Vorläufiges Layout der Bandgap-Referenz. Zusätzlicher Vergleich der Bipolartransistoren mit Standardbauelementen (b) und Multi-Emitter Elementen (a).

hoch sind, sind dadurch auch ihre Emitterströme verschieden. Ist die Verstärkung ausreichend hoch, dann verringern sich die Basisströme und haben einen vernachlässigbaren Einfluss auf die Ströme $I_1$ und $I_2$ und somit auf den Fehler der Referenzspannung.

Wird die Schaltung bei hohen Temperaturen untersucht, sind die Leckströme zum Substrat nicht mehr vernachlässigbar klein. Eine größere aktive Fläche verursacht größere Leckströme. Wenn Leckströme in beiden Zweigen in gleichem Maße vorkommen, hat dies keine Auswirkung auf die Genauigkeit der Referenzspannung. Da $Q_1$ die sechsfache Fläche von $Q_2$ besitzt, müssen die größeren Leckströme von $Q_1$ kompensiert werden. Dies geschieht mit parallelen „Dummy“-Transistoren zu $Q_2$, wobei die Emitter der „Dummy“-Transistoren nicht kontaktiert sind. Durch diese Maßnahme werden in beiden Zweigen identische Leckströme erzielt.

Abweichungen der Widerstandswerte $R_1$ und $R_2$ und der Basis-Emitter-Spannungen von $Q_1$ und $Q_2$, hervorgerufen durch Prozessschwankungen und Missmatch, besitzen eine lineare Temperaturabhängigkeit und können daher als lineare Fehler betrachtet werden. Diese linearen Fehler wirken sich negativ auf die Spannungsreferenz aus und müssen daher kompensiert werden. Diese Kompensation kann durch das Trimmen des Widerstands $R_2$ realisiert werden. Die Schaltung des trimmbaren Widerstands wird in Abbildung 3, Teilschaltung ④ gezeigt. Der Widerstand $R_2$ wird in einen Offsetwiderstand $R_{Offset}$ und weitere Segmentwiderstände $R_{Seg}$ aufgeteilt. Die binär gewichteten Segmentwiderstände können über parallel geschaltete Transistoren kurzgeschlossen werden.

Die Brokaw-Zelle besitzt zwei stabile Arbeitspunkte, den eingeschalteten und den ausgeschalteten Zustand, wenn der Strom $I_{PTAT}$ fließt oder gleich Null ist. Beim Einschalten der Eingangsspannung würde somit die Zelle den ausgeschalteten Zustand nicht verlassen. Um dieses Verhalten zu verhindern, wird eine Startschaltung eingesetzt. Diese ist in Abbildung 3, Teilschaltung ② dargestellt. Bei steigender Versorgungsspannung schaltet sich $P_5$ ein und hebt die Spannung über $Q_2$ auf $V_{DD}$. Gleichzeitig wird auch $P_{10}$ geschaltet, wodurch ein Strom in die Basen von $Q_1$ und $Q_2$ injiziert wird. Hierdurch verlässt die Brokaw-Zelle den stabilen ausgeschalteten Zustand. Durch den nun initiierten Strom $I_{PTAT}$ wird dieser auch in den Zweig von $P_6$ und $P_7$ gespiegelt. Dieser Strom fließt durch die aktive Last aus $N_3$ und $N_4$ und durch den hochohmigen Widerstand $R_3$. Dadurch fällt ungefähr $V_{DD}$ am Gate von $P_5$ ab und wird somit wieder abgeschaltet. Dadurch wechselt die Startschaltung in den „sleep mode“, bei dem sich die Stromaufnahme auf weniger als 300 nA reduziert. Dieser Strom ergibt sich aus der Kombination aus aktiver Last, bestehend aus den Transistoren $N_3$ und $N_4$, und einer Widerstandslast $R_3$. Im Vergleich zu einer reinen Widerstandslast kann dadurch die Fläche verringert werden.

Der Referenzstrom wird, wie in Abbildung 3, Teilschaltung ③ gezeigt, mit Hilfe der Referenzspannung $V_{REF}$ und dem Widerstand $R_4$ erzeugt. Da $V_{REF}$ temperaturstabil ist, ist die Schwankung des Referenzstroms nur von der Schwankung des Widerstands $R_4$ abhängig. Um die Prozessschwankungen von $R_4$ zu kompensieren, ist $R_4$ ebenfalls als trimmbarer Widerstand implementiert. Die verbleibenden Schwankungen des Referenzstroms werden hauptsächlich durch den nicht trimmbaren TK von $R_4$ verursacht. Für einen Referenzstrom $I_{REF}$ von 1,997 µA und einer Referenzspannung $V_{REF}$ von 1,218 V errechnet sich ein Wert für $R_4$ mit

$$R_4 = \frac{V_{REF}}{I_{REF}} = \frac{1{,}218\,\mathrm{V}}{2\,\mu\mathrm{A}} = 609{,}91\,\mathrm{k\Omega}\,. \qquad (11)$$

Zur weiteren Verwendung dieses Stroms, wird dieser über einen Stromspiegel gespiegelt und steht als Stromquelle für periphere Applikationen zur Verwendung.

### *B. Layout der Bandgap-Referenz*

Abbildung 6 zeigt das vorläufige Layout der Bandgap-Referenz. Dessen Ausmaße mit Multi-Emitter-Bauelementen (Abbildung 6 ⓐ) belaufen sich auf 157 µm x 210 µm und damit eine Fläche von 0,033 mm². In einem ersten Entwurf sind die beiden Bipolartransistoren $Q_1$ und $Q_2$ sowie dessen Dummy-Transistoren als 12 separate Standardbauelemente

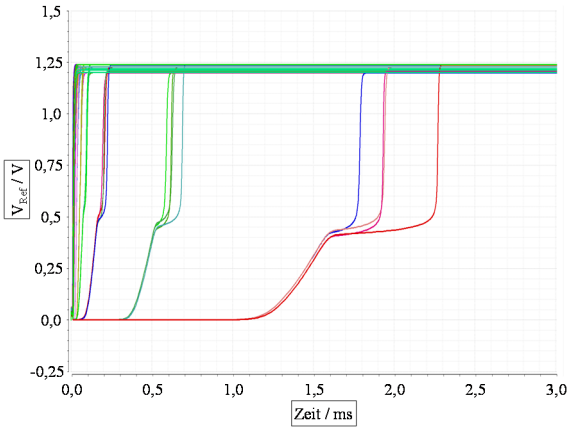



Abbildung 9: Simuliertes Startverhalten über mehrere Corner.

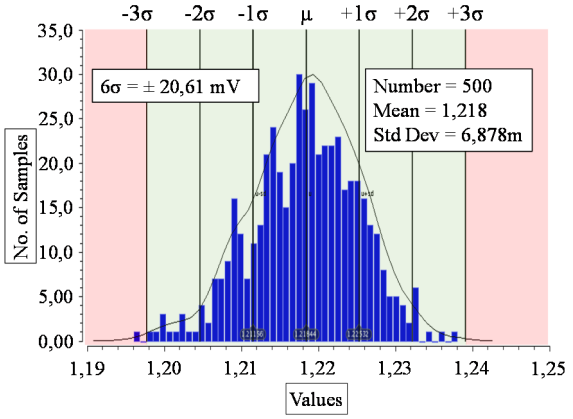



Abbildung 8: Monte-Carlo-Simulation von $V_{REF}$.

implementiert (Abbildung 6 ⓑ). In einem zweiten Entwurf, (Abbildung 6 ⓐ) werden die Standardbauelemente ersetzt durch ein manuell erstelltes Multi-Emitter-Layout. Hierbei werden die jeweiligen Emittergebiete von $Q_1$ und $Q_2$ in jeweils ein gemeinsames Kollektor- und Basis-Gebiet implementiert. Damit kann die Layoutfläche von $Q_1$ und $Q_2$ von 54 µm x 78 µm mit Standardbauelementen (Abbildung 6 ⓑ) auf 44 µm x 16 µm mit Multi-Emitter-Modellen (Abbildung 6 ⓐ) reduziert werden, was einer Flächenreduzierung der Bipolartransistoren von etwa 80 % entspricht. Die Annahme, dass die Multi-Emitter-Modelle eine ähnliches Verhalten wie die Standardbauelemente aufweisen, beruht darauf, dass sich die Dotierungen und die Flächen der Basis-Emitter-Übergänge in den Multi-Emitter-Modellen nicht wesentlich ändern.

## IV. Simulationsergebnisse

Abbildung 9 zeigt die Referenzspannung $V_{REF}$ über der Zeit in Millisekunden. Jede Kurve stellt dabei ein Corner dar. Im Worst Case benötigt die Schaltung nach einem Sprung der Versorgungsspannung auf 2,5 V, 5,0 V oder 5,5 V maximal 2,3 ms, bis der nominelle Wert der Referenzspannung am Ausgang erreich ist. Diese Verzögerung ist hauptsächlich bedingt durch die parasitären Kapazitäten der Transistoren, welche in der Startschaltung zum Einsatz kommen.

Das Ergebnis einer Monte-Carlo Simulation ist in Abbildung 8 dargestellt. Es ist die Anzahl der simulierten Proben über $V_{REF}$ (hier als „Values“) dargestellt. Der Mittelwert von $V_{REF}$ ist bei 1,218 V und dessen Standardabweichung $\sigma$ beträgt 6,878 mV. Hierdurch befinden sich bis $6\sigma$ alle Werte in einem Bereich von ± 20,61 mV.

Eine Kurvenschar von $V_{REF}$ über alle Corner und einer Versorgungsspannung von 2,5 V, 5,0 V und 5,5 V wird in Abbildung 10 dargestellt. Jede Kurve stellt dabei $V_{REF}$ über Temperatur von -40 °C bis 180 °C dar. Diese Kurven zeigen den nicht getrimmten Zustand. Wird die Schaltung getrimmt, so bewegen sich die Kurven in Richtung der nominellen Kurven (Abbildung 10 Mitte). Dadurch ändert sich nicht nur der Offset, sondern auch der TK. Das bedeutet, dass die Kurven zusätzlich kippen und flacher werden. In Abbildung 7 werden alle relevanten Werte und deren Schwankungen zusammengefasst. Nach dem Trimmen werden die Referenzspannung $V_{REF}$ = 1,218 V ± 5 mV und der Referenzstrom $I_{REF}$ = 1,997 µA ± 0,6 µA erzielt.

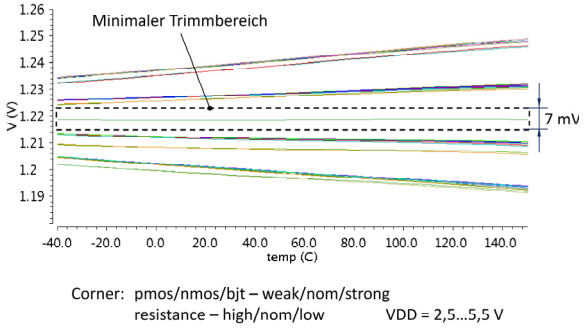



Abbildung 10: Simulation von $V_{REF}$ über Temperatur und verschiedene Corner.

| | Nominell | Prozess | Temperatur |
|---|---|---|---|
| $V_{REF}$ | 1,218 V | ± 20,64 mV | + 2,18 ppm/K |
| $I_{REF}$ | 1,997 µA | ± 89,7 nA | + 0,30 %/K |
| $I_{GES}$ | 7,406 µA | ± 194,4 nA | + 0,39 %/K |
| $I_{PTAT}$ | 1,007 µA | ± 36,6 nA | + 0,70 %/K |

Nach dem Trimmen

$V_{REF}$ = 1,218 V ± 5 mV
$I_{REF}$ = 1,997 µA ± 0,6 µA
$I_{GES}$ = 7,406 µA

Abbildung 7: Auflistung der simulierten Parameter mit nominellem Wert und Schwankung über Prozess und Temperatur.

## V. FAZIT

Es wird ein Entwurf für eine hochgenaue Bandgap-Referenz für Low-Power-Anwendungen, basierend auf der Brokaw-Zelle, vorgestellt. Durch eine genaue Abstimmung der Bipolartransistoren mit hoher Verstärkung, dem Einsatz von kaskodierten Niedervoltstromspiegeln und elektronischem Trimmen werden eine stabile Referenzspannung von 1,218 V mit einer Abweichung von ± 5 mV über einen weiten Versorgungsspannungsbereich von 2,5 bis 5,5 V und einen Temperaturbereich von -40 °C bis 150 °C erreicht. Die Temperaturgenauigkeit entspricht dabei 2,18 ppm/K. Zusätzlich wird ebenfalls ein Referenzstrom von 1,997 µA realisiert, mit einer Abweichung von ± 0,6 µA. Durch die Kombination von aktiver und resistiver Last konnte die Startschaltung flächeneffizient implementiert und die Gesamtstromaufnahme der Bandgap-Referenz auf $I_{ges}$ = 7,406 µA gesenkt werden. Dadurch eignet sich die Referenz optimal für den Einsatz in Low-Power-Anwendungen. Durch die Implementierung von Multi-Emitter-Bipolartransistoren konnte deren Layoutfläche um 80 % reduziert und somit die Bandgap-Referenz mit einer Chipfläche von 0,033 mm² realisiert werden.

## LITERATURVERZEICHNIS

[1] W. Horn, H. Zitta, „A robust Smart Power Bandgap Reference Circuit for Use in an Automotive Environment," in *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 7, July 2002.

[2] Pease, R.A., "The design of band-gap reference circuits: trials and tribulations," *Bipolar Circuits and Technology Meeting*, Minneapolis, September 1990.

[3] Texas Instruments, Datenblatt: "LM113/LM313 Reference Diode", www.ti.com , Mai 2013.

[4] D. P. Laude, J. D. Beasom, "5 V temperature regulated voltage reference," *IEEE Journal of Solid-State Circuits*, December 1980.

[5] P. Miller, D. Moore, „Precision voltage references", Texas Instruments Incorporated, November 1999.

[6] F. J. Franco, Y. Zong, J. A. Agapito, A. H. Cachero, "Radiation effects on XFET voltage references," *Radiation Effects Data Workshop*, July 2005.

[7] A. P. Brokaw, „A Simple Three-Terminal IC Bandgap Reference", *IEEE Journal of Solid-State Circuits*, Vol. SC-9, No.6, December 1974.

[8] P. R. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer, „Analysis and Design of Analog Integrated Circuits", 5th ed., John Wiley and Sons Ltd, May 2009.

[9] A. P. Brokaw, „How to Make a Bandgap Voltage Reference in one easy lesson", A. Paul Brokaw and Integrated Device Technology, Attend Corporate Info Session-week, 2011.

*Ismail Yasar* erhielt den akademischen Grad des B.Eng. in Mechatronik mit dem Schwerpunkt Mikroelektronik im Jahr 2014 von der Hochschule Reutlingen und studiert derzeit den Master Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen.

*Robin Staudt* erhielt den akademischen Grad des B.Eng. in Mechatronik im Jahr 2014 von der Hochschule Reutlingen und studiert derzeit den Master Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen.

*Cedric Leonel Jiago Teffo* erhielt den akademischen Grad des B.Eng. in Mechatronik mit dem Schwerpunkt Mikroelektronik im Jahr 2014 von der Hochschule Reutlingen und studiert derzeit den Master Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen.

*Benjamin Schoch* erhielt den akademischen Grad des B.Eng. in Elektrotechnik/Informationstechnik im Jahr 2013 von der Hochschule Pforzheim und studiert derzeit den Master Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen.

*Thomas Stoof* erhielt den akademischen Grad des B.Eng. in Technische Informatik im Jahr 2014 von der Hochschule Esslingen und studiert derzeit den Master Leistungs- und Mikroelektronik am Robert Bosch Zentrum für Leistungselektronik der Hochschule Reutlingen.

*Jürgen Wittmann* erhielt den akademischen Grad des Dipl.-Ing. im Jahr 2006 von der technischen Universität München. Von 2006 bis 2011 war er in der Abteilung Mixed-Signal Automotive bei Texas Instruments in Freisig als Analog Design Ingenieur tätig. Februar 2011 begann er am Robert Bosch Zentrum für Leistungs- und Mikroelektronik an der Hochschule Reutlingen als Forschungsassistent. Derzeit promoviert er im Bereich der Leistungs- und Mikroelektronik.

*Bernhard Wicht* erhielt den akademischen Grad des Dipl.-Ing. im Jahr 1996 von der technischen Universität Dresden und promovierte im Jahr 2002 an der technischen Universität München. Von 1996 bis 1998 war er als Analog Designer für ASICs für optische Messsysteme tätig. 1998 begann er als Forschungsassistent an der technischen Universität München. Von 2003 bis 2010 war er in der Abteilung Mixed-Signal Automotive bei Texas Instruments in Freisig verantwortlich für die Entwicklung von Automotive Smart Power ICs. September 2010 wurde er Professor für integrierte Schaltungen an der Hochschule Reutlingen, Robert Bosch Zentrum für Leistungs- und Mikroelektronik.

# Design and Verification of a Mixed-Signal SoC for Biomedical Applications

Mayukh Bhattacharyya, Benjamin Dusch, Dirk Jansen, Elke Mackensen

***Abstract*—In this paper an RFID/NFC (ISO 15693 standard) based inductively powered passive SoC (system on chip) for biomedical applications is presented. A brief overview of the system design, layout techniques and verification method is discussed here. The SoC includes an integrated 32 bit microcontroller, sensor interface circuit, analog to digital converter, integrated RAM, ROM and some other peripherals required for the complete passive operation. The entire chip is realized in CMOS 0.18 µm technology with a chip area of 1.52mm x 3.24 mm.**

***Keywords*—RFID (Radio Frequency Identification Device), RF–field passive system, energy harvesting, analog to digital converter, sensor interface circuitry, microcontroller, biotelemtry.**

## I. Introduction

Near field communication (NFC) along with RFID technology has gained significant importance in recent times which in turn aids in development of passive sensor systems by harvesting energy from the induced electromagnetic field [1]. NFC provides a new opportunity for the development of ultra-low power sensor systems for biotelemetry applications. This is because a commercially available hand held device like a smart phone or a tablet is good enough to interact with such a system.

For this reason a SoC with NFC interface, sensor interface circuitry, SAR ADC, microcontroller core and memory have been developed. Although the end application of this SoC is not fixed, the one intended application is to develop a completely passive biotelemetry system to monitor blood pressure in arterial system (femoral) for the patients suffering from Peripheral Arterial Disease (PAD) [2], much similar to the work presented in [3]. Due to availability of features like programmable instrumentation amplifier, integrated temperature sensor and availability of virtual ground compatible with human body model it can also be used for measuring ECG signal, body temperature [4] etc.

Mayukh Bhattacharyya, mayukh.bhattacharyya@hs-offenburg.de, Benjamin Dusch, benjamin.dusch@hs-offenburg.de, Dirk Jansen, d.jansen@hs-offenburg.de, and Elke Mackensen, elke.mackensen @hs-offenburg.de are with Hochschule Offenburg, Badstraße 24, 77652 Offenburg.

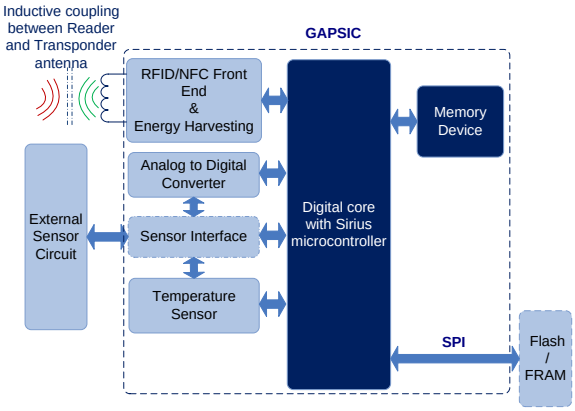



Figure 1: System overview for the proposed SoC (GAPSIC).

A system overview will be presented in section II. Section III and IV contain the description for the analog and the digital part and some basic layout techniques are described in brief in section V.

## II. System Overview

The basic system overview of the developed SoC – GAPSIC (*General Application Passive Sensor Integrated Circuit*) is shown in the figure 1 with all the necessary important blocks. The antenna serves the dual purpose which is used for RFID/NFC communication as well as for harvesting energy by using inductive coupling. The sensors can be externally connected to the system depending on the intended area of applications. The Flash/FRAM contains the application software or the firmware which can be updated depending on the requirement for the application. The internal digital core can communicate with the external device by using SPI interface. The analog to digital converter is a charge redistribution type based on Successive Approximation Register logic (SAR). The sensor interface includes the programmable instrumentation amplifier along with a channel selector and a virtual ground amplifier. The internal bandgap reference is modified to act as a temperature sensor whose purpose is to measure the internal temperature of the chip which is helpful in doing temperature dependent calibrations. The digital core consists of the core of

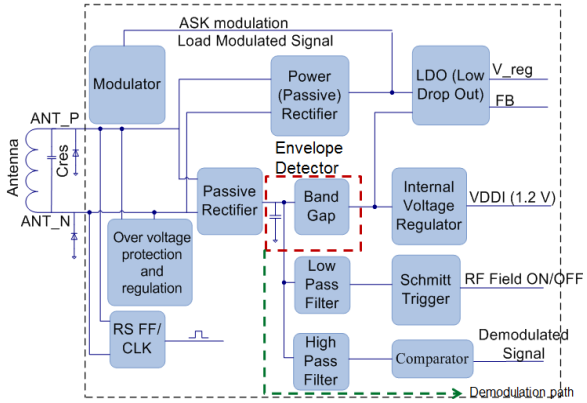



Figure 2: Simplified block diagram of the RFID/NFC analog frontend.

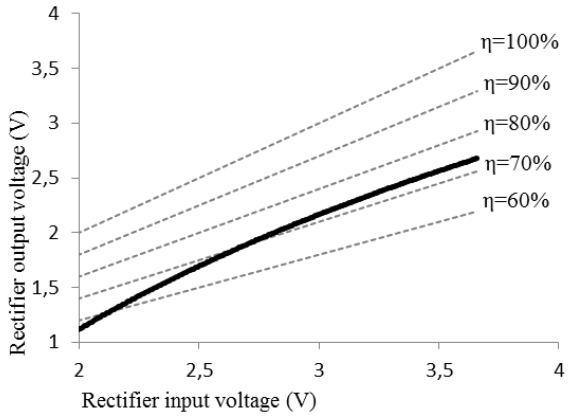



Figure 3: Measured efficiency of the energy harvesting rectifier (shown in solid line).

the SIRIUS microcontroller along with other peripherals like timer, interrupt controller etc. The internal memory devices include a 16 kB ROM and a 16 kB RAM. Being a mixed signal IC it consists of an analog part and a digital part, so each of the system blocks is described further accordingly.

## III. Analog Part

The analog part consists of an RFID/NFC block along with an energy harvesting part and the sensor interface part including the analog to digital converter.

### *A. RFID/NFC and Energy Harvesting Block*

This block consists of the analog circuitry required for RFID/NFC communication and for energy harvesting as shown in the figure 2 [5]. The energy harvesting block consists of a full wave bridge rectifier and a low drop out voltage regulator which provides a stable output voltage. As shown in the figure 3, the rectifier has an average efficiency $\eta$ of ~ 66 % which provides sufficient energy required for complete passive operation. The low drop out regulator (LDO) uses the harvested energy available from the rectifier to provide a

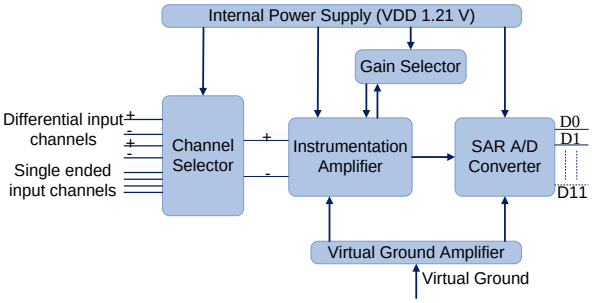



Figure 4: Simplified block diagram of the Sensor interface circuit along with SAR ADC analog frontend.

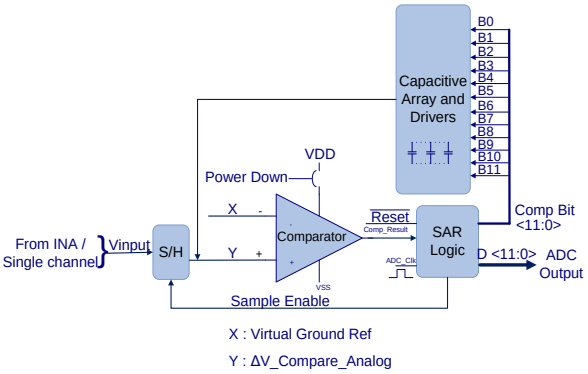



Figure 5: Block diagram showing the sample and hold circuit, comparator, capacitive array and the SAR logic for the Analog to Digital converter (ADC).

stable output voltage ranging from 1.21 V to 2.0 V. The internal bandgap reference voltage provides a stable reference voltage 1.21 V ± 20 mV. The bandgap circuit also act as a load required for the envelope detector circuit (shown in figure 2) which is required for demodulation. Further this bandgap circuit is modified as a temperature sensor in order to get an idea of the in chip temperature, which can be useful for calibration purposes.

The clock is extracted from the RFID field by a clock regenerator circuit which is then further used as system clock. The demodulation is carried out by a comparator which has a predefined offset voltage which sets the dc working condition required for proper demodulation of the RFID signal. The reply is sent back via the modulator circuit which uses ASK (Amplitude Shift Keying) modulation for this purpose.

### *B. Sensor Interface Circuit and SAR ADC*

As shown in figure 4 the sensor interface circuit consists of a channel selector to which two differential channels and four single ended channels are available to which different kinds of sensors can be connected depending on the requirements. The internal low dropout voltage regulator supplies the entire sensor interface block with a constant voltage supply of 1.21 V which in turn aids in lowering the power consumption. In this design, the instrumentation amplifier used is a three operational amplifier instrumentation ampli-

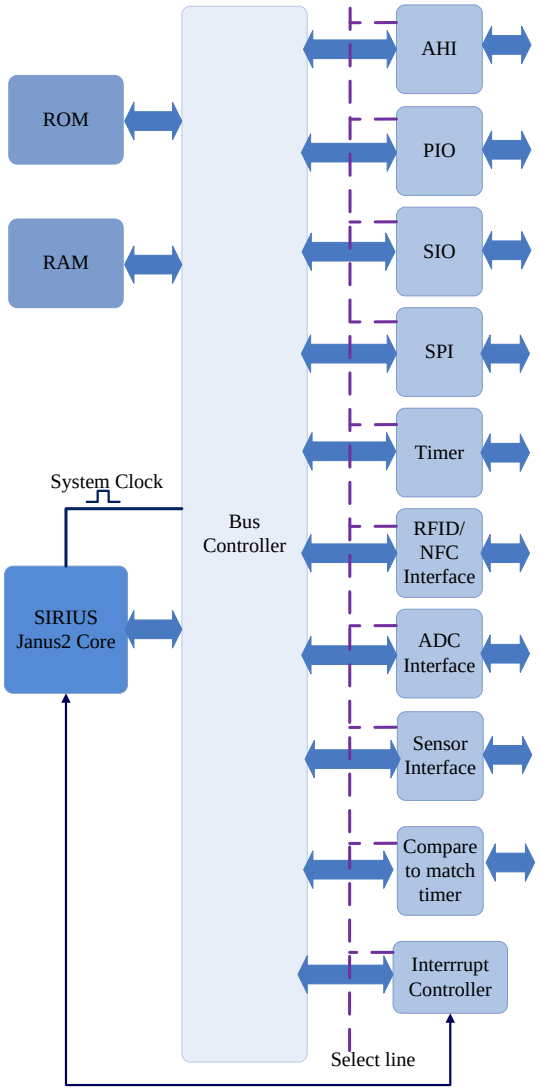



Figure 6: Block diagram showing the digital core including the microcontroller, integrated memory (RAM and ROM) and other peripheries.

fier with a programmable gain. It has a maximum offset voltage of ~2 mV depending upon the temperature range (-30 °C to 85 °C). The gain selector can be controlled by software in order to choose the required gain ranging from 1 to 100. Due to availability of a digital part and microcontroller no offset cancellation method is used rather the offset is measured before every measurement and the value is stored for cancellation. The virtual ground amplifier is designed in such a way that it is compatible with the human body model so that one can use it as the reference electrode in case of an ECG measurement [6].

As shown in the figure 5 the SAR ADC block consists of a sample and hold circuit, comparator, capacitive array with drivers and SAR logic [7]. The sample and hold circuit switch is designed in such a way that it has minimum ohmic resistance as well as the charge injection is also low. The comparator is an open loop comparator with high input impedance but very low input capacitance as it will affect the measurement results. The input offset voltage of the comparator is kept as low as possible (60 µV to 150 µV depending on the temperature). The effect of the random mismatch is reduced effectively by adopting effective layout techniques. The maximum input frequency is around 20 kHz which is well within the specified range of operation. The capacitive array consists of unit capacitances arranged in binary form representing each bits from 1 to 12. In order to avoid the effect of line impedances each of the unit capacitances has their own driver stage. The SAR logic consists of a binary logic required for the measurement in successive approximation form which is actually provided in the digital block. In general, except the voltage regulator, the entire sensor interface circuitry block can be isolated or kept switched off when not in use thereby reducing the static power consumption as much as possible. The layout of the capacitive array is very important and successful layout is necessary for proper operation which is discussed later.

## IV. Digital Part

The digital part includes the SIRIUS JANUS 2.0 microcontroller, memory devices (RAM and ROM) and other peripheries as shown in the figure 6. Some of the individual blocks are discussed below [8]. The select line shown in the figure helps to select each of the peripheries as and when required. The other signal lines which are not shown in the figure for the sake of simplification are read, write, reset, address bus and system clock. Each of the peripheries is selected by using their corresponding addresses and the select lines.

### *A. SIRIUS JANUS 2 Core*

The microcontroller used here is SIRIUS JANUS 2.0 which is developed by ASIC design center (University of Applied Sciences, Offenburg), based upon Von Neumann architecture. It has 16 bit data bus and 32 bit address bus. It consists of 16 registers of which 12 are universal and 4 are special registers. The arithmetic and logical operations are based on Reduced Instruction Set Computing (RISC). The internal control unit can handle 16 bit command format along with an internal 8 bit control signal.

The microcontroller is designed to operate at much higher frequency (~ 50 MHz), but here it is operated at a frequency of 6.78 MHz. The extracted RFID clock is divided to half and then further used as the system clock. The intended application doesn't require a higher clock frequency hence no external oscillator or an internal PLL (Phase locked loop) is required which helps to keep the overall size smaller and also reduces the power consumption which is important for passive operation. Also a slower system clock means lower

power consumption which is again important for power consumption.

*B. Audio Human Interface (AHI)*

This is not required for the actual operation of the chip but only used for easy debugging purposes. Different frequencies can be set at different point of operation which is useful to identify the current state of the system. This is more of a use in the development phase rather than the actual application.

*C. Parallel Input Output (PIO)*

Similar to the AHI block the PIO block is also used for debugging by using the 8 bit data bus to read or to write.

*D. Serial Input Output (SIO)*

Like the AHI block and PIO block this is also used for debugging purposes as text messages used for detecting errors can be read over Hyper Terminal. This is a very useful debugging tool which can be extensively used in the development phase.

*E. Serial Parallel Interface (SPI)*

This is used mainly for the external memory device which can be a Flash or FRAM, to load the firmware from the external memory into the internal RAM. It uses the standard SPI signals MISO (Master input slave output), MOSI (Master output and slave input), SPI clock and chip selects (two are available). The first chip select is by default set for the selection of the external memory device and one more chip select is free to be used for some other SPI device.

*F. Timer*

There are in total four timers available out of which one is used for boot up routine and the other one is used by the RFID communication. The other two timers are completely free and can be used at any given time.

*G. RFID/NFC Interface*

This block is very important as it is responsible for the digital logic required for the RFID and NFC communication as well as the generation of the system clock. It generates four interrupt signals required for the RFID/NFC communication.

*H. ADC Interface*

The ADC interface consists of the SAR logic, ADC clock prescaler, data registers and control registers. The SAR logic block contains the binary logic required for the ADC operation. The clock prescaler is useful as one can chose the clock frequency required for the ADC operation (~96 kHz to ~1 kHz). When

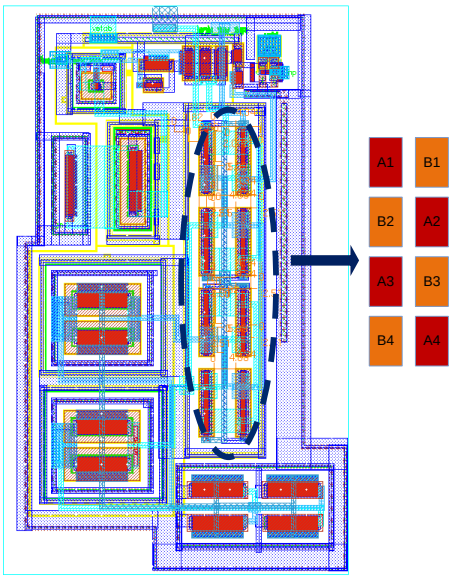



Figure 7: SAR ADC comparator layout showing the common centroid layout for the differential pair which is divided into two sub pairs each.

the ADC conversion is finished it generates an interrupt. The ADC conversion result is stored in a data register which can be read as and when required till the next conversion starts or a reset occurs.

*I. Sensor Interface*

The sensor interface consists mainly of control and data registers required to control the sensor interface port of the analog part except the ADC. It controls the power on and off for the entire sensor interface part, selection of the channel required for measurement and also the gain of the instrumentation amplifier.

*J. Compare to Match*

The compare to match is used as a pulse generator which can be useful for some measurement purposes for example in case of impedance measurement. A pulse wave is useful as it reduces the galvanic corrosion in electrodes due to the DC (Direct Current) effect.

*K. Interrupt controller*

The interrupt controller consists of several registers which can be controlled via software such as control, edge, mask and base address registers. The control register has two bits, one of them being responsible to choose the mode (16 or 32 bit) and the other one is used to reset any pending interrupt. For this design the 16 bit mode is chosen. In the edge register one can decide between rising edge and falling edge to trigger the respective interrupt. The mask register decides which interrupt source will trigger the interrupt and

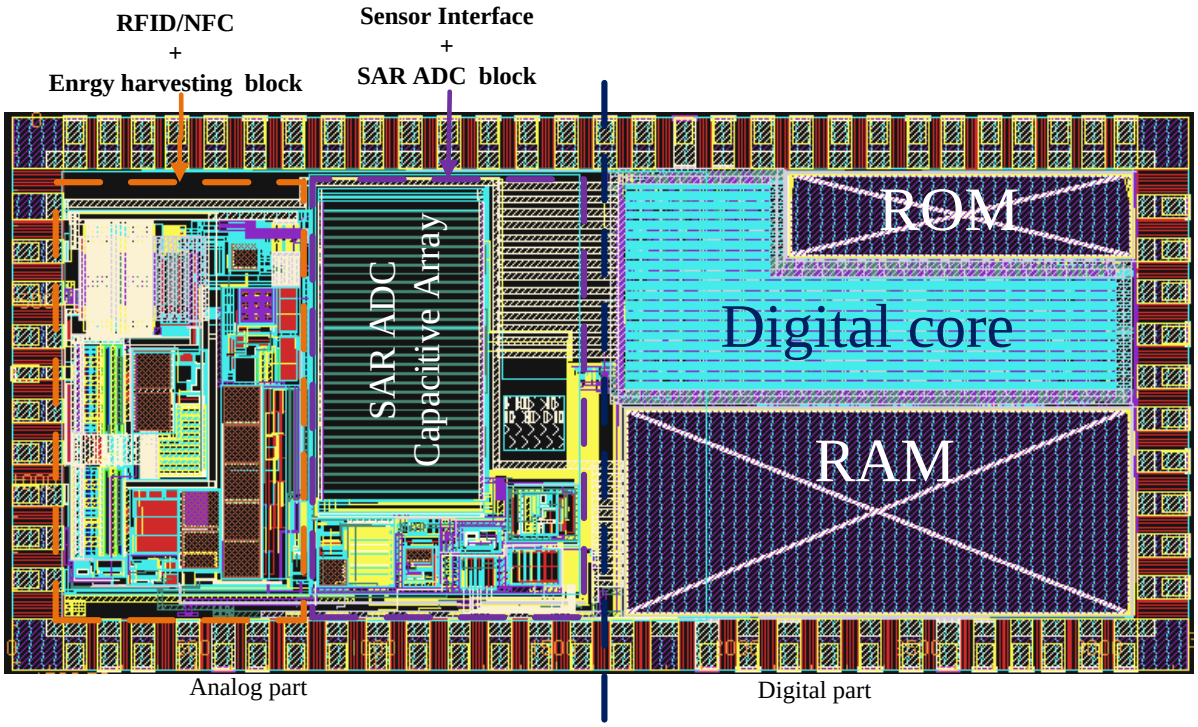



Figure 8: Complete layout of the SoC along with the bond pads.

the bit 0 is non maskable as it is only readable. The base address register contains the base address for the hardware interrupts which is further organized in a vector table.

*L. Random Access memory (RAM)*

A 16 kB RAM is provided inside the chip, the entire firmware along with the initial routines is organized inside the RAM.

*M. Read only memory (ROM)*

The ROM is also 16 kB which contains the initial boot up routine which is mainly the SPI routine along with the hardware initialization required for the operation. The SPI routine in the ROM interacts with the SPI block and starts the SPI communication required for SPI read operation which uploads the firmware from the external memory device into the RAM. Moreover it also contains a small program which responds every time there is a RFID/NFC inventory command by sending a unique identification number. This is also a part of the initial boot up check for the system.

## V. Layout Techniques

As it is a mixed signal system the digital layout is accomplished by using Cadence Encounter program and the analog part is done in Cadence Virtuoso. Finally both parts are combined together in Cadence Virtuoso program along with the bond pads. Here some of the layout techniques adopted for the analog part are discussed in brief.

In general for critical circuits like the comparator (as shown in figure 7) for ADC or instrumentation amplifier, special care has been taken while doing the layout. The layouts of the differential pairs are done by using a common centroid approach in order to reduce the error due to random offsets occurring due to random mismatches. Also guard rings are provided around the transistors for better noise immunity. In figure 7 A1, A2, A3 and A4 represent one of the differential pair and B1, B2, B3 and B4 represent another differential pair which are then arranged in zig – zag form in order to reduce the effect due to random mismatches.

A partial common centroid layout approach is adopted for the layout of the capacitive array in order to reduce the effect due to gradients caused by temperature as well as oxide thickness [9]. For the digital circuit layout, the clock trees are designed for much higher frequencies than required to operate, so that the delays are minimal. Figure 8 shows the complete layout of the chip where on the left hand side is the analog part and the right hand side is the digital part along with the memory devices (RAM and ROM).

## VI. Design Simulation and Evaluation

The analog part is verified by using different kinds of simulations (e.g. corner and Monte Carlo simulations) as well as post layout extraction which is important to take into account all the parasitics which

Table 1: Overall features of the frontend.

| | |
|---|---|
| Technology | UMC CMOS 0.18 µm |
| Size of the chip | 1.52 mm × 3.24 mm |
| Bandgap reference voltage | 1.21 V ± 20 mV |
| Temperature range | -30 °C − 85 °C |
| Output range of Low drop out regulator | 1.21 V − 2.2 V |
| System clock | 6.78 MHz |
| RFID / NFC standard | ISO 15693 |
| ADC | SAR , 12 bit resolution (max) |
| ADC channel | Two differential and four single ended |
| Sensor Interface power consumption (max) | ~ 570 µW |
| Mode of operations | Passive or semi-passive |
| Application area | Biotelemetry, industrial sensors |

play a dominant role in the sensor interface circuit and ADC. A mixed-signal simulation has been used for the simulation of the SAR ADC in order to verify the complete functionality. For the digital part other than simulation one can realize the entire design in an FPGA (Field Programmable Gate Array). For this purpose a FPGA emulation board is used along with an analog circuitry interface consisting of RFID/NFC in order to realize the complete system. A post layout simulation is also done with the generated netlist from encounter. The external memory devices are also included while doing the simulation in order to realize the exact one to one system performance.

## VII. Conclusions

An ultra-low power consuming SoC for biotelemetry applications has been discussed and a brief overview of the overall features of the SoC is presented in table 1. The SoC consists of an RFID/NFC frontend which is used for communication as well as for energy harvesting which is important for passive operation. A 12 bit SAR ADC sensor interface circuit along with a 32 bit microcontroller opens up a wide range of applications. Although the intended area of application is in the field of biotelemetry it can also be used for passive industrial sensors [10]. The availability of the RFID/NFC communication standard (ISO 15693) makes it easy to interact with such a system as a standard hand held device like a smart phone or a tablet containing application software can be used by the end user to interact. This opens up new opportunities for next generation passive medical health care systems [11].

## Acknowledgement

The authors like to thank all colleagues from ASIC design center and Institute for Applied Research Offenburg for their support and coordination.

## References

[1] T. Volk, M. Bhattacharyya, W. Gruenwald, L. Reindl, D. Jansen, "Formal Description of Inductive Air Interfaces using Thévenin's theorem and numerical analysis", *IEEE Trans on Magnetics*, vol.50, no.6, June 2014.

[2] K. Wolf-Maier et al, "Hypertension prevalence and blood pressure levels in 6 European countries, Canada and the United States", *JAMA*, vol. 289, no.18, 2003, p.2363-2369.

[3] J. H. Cheong, S. S. Yan Ng, X. Liu, R. Xue, H. J. Lim, P. B Khannur, K. L. Chan, A. A. Lee, K. Kang, L. S. Lim, C. He, P. Singh, W. Park, "An Inductively Powered Implantable Blood Flow Sensor Microsystem for Vascular Grafts", *IEEE Trans. on Biomedical Eng*, vol. 59, no. 9, pp 2466 – 2475, September 2012.

[4] W. Jeon, J. Melngailis, R. W. Newcomb, "Disposable CMOS passive RFID transponder for patient monitoring", *ISCAS*, Greece, May 2006.

[5] M. Bhattacharyya, T. Volk, A. Kreker, B. Dusch and D. Jansen, "Realization of a RFID Front End IC for ISO 15693 standard in UMC CMOS 0.18 µm technology", *MPC workshop*, Aalen, July 2012

[6] B. Dusch, M. Bhattacharyya and D. Jansen, „Entwicklung und Layoutentwurf eines Analog-Digital-Wandlers mit 12 Bit Auflösung in einer 180 nm CMOS-Technologie", *MPC Workshop*, Esslingen, Februar 2015.

[7] D. Venuto, E. Stikvoort and D. Castro, "Ultra low-power 12 bit SAR ADC for RFID Applications", *Design, Automation & Test in Europe* Conference & Exhibition, 2010 , ISBN - 978-1-4244-7054-9.

[8] D. Jansen, N. Fawaz, D. Bau, M. Durrenberger, "A small high performance microprocessor core SIRIUS for embedded low power designs, demonstrated in a medical mass application of an electronic pill", ISBN 978-0-387-72257-3, pp. 363-372.

[9] D. Zhang, A. Bhide and A. Alvandpour, "A 53-nW 9.1-ENOB 1-kS/s SAR ADC in 0.13 um CMOS for Medical Implant Devices," 2012, *IEEE Journal of Solid-State Circuits*, (47), 7, 1585-1593.

[10] R. Jedderman, L. R. Garcia, W. Lang, "Spatial temperature profiling by semi-passive RFID loggers for perishable food transportation", *Journal Elsevier*, August 2008.

[11] M. Bhattacharyya, W. Gruenwald, B. Dusch, J. Aghasssi-Hagmann, D. Jansen and L. Reindl, "An RFID/NFC Based Programmable SoC for Biomedical Applications", Published in *SoC Design Conference* (*ISOCC*), 2014 International, Jeju, South Korea, 3-6 Nov. 2014, ISBN 978-1-4799-5126-0.

# CAPABLE: A Layout Automation Framework for Analog IC Design

Daniel Marolt, Jürgen Scheible, Göran Jerke, Vinko Marolt

***Abstract*—In practice, the use of layout PCells for analog IC design has not advanced beyond primitive devices and simple modules. This paper introduces a Constraint-Administered PCell-Applying Blocklevel Layout Engine (CAPABLE) which permits PCells to access their context, thus enabling a true "bottom-up" development of complex parameterized modules. These modules are integrated into the design flow with design constraints and applied by an execution cockpit via an automatically built layout script. The practical purpose of CAPABLE is to easily generate full-custom block layouts for given schematic circuits. Perspectively, our results inspire a whole new conception of PCells that can not only act (on demand), but also react (to environmental changes) and interact (with each other).**

***Index Terms*—Analog IC design, layout automation, parameterized cells, design constraints, bottom-up design.**

## I. Introduction

Semiconductor products continue to revolutionize modern life in the 21st century, and analog IC content plays an essential role for the ongoing functional diversification of integrated circuits. Unfortunately, analog IC design still represents an economic bottleneck for the microelectronics industry. In particular, the design of physical layouts becomes more and more critical, facing the increasingly intricate challenges of advanced semiconductor technology nodes.

### *A. The Automation Gap in Analog Layout Design*

In the digital domain, the layout creation task of integrated circuit design is highly automated. *Optimization algorithms* are successfully employed to place and route millions of devices per IC. In contrast, three decades of substantial research in Electronic Design Automation (EDA) have not yet achieved a large-scale adoption of such algorithmic approaches in the analog domain, where the number of devices is much smaller but the design requirements are significantly more complex. For that reason, optimization algorithms still struggle to find their way into industrial environments.

In practice, analog layouts are still handcrafted by expert designers in a time-consuming manual fashion today, with parameterized cells being the main source of automation. These so-called *PCells* represent layout *generators* and are fundamentally different from optimization-based approaches because they do not work in an algorithmic but in procedural way. PCells are mainly used to generate customizable layout variants of primitive devices such as transistors and resistors. From a scientific point of view, a PCell's automation abilities are comparably trivial, but for a design expert's daily layout work, PCells are indispensable. Thus it can be observed that – opposite to the ongoing pursuit of algorithmic solutions in academia – industrial flows rather drive the development of more powerful *module PCells* which are able to create layouts for entire circuits. In practice however, the advancement of PCells has not yet really proceeded beyond simple modules such as current mirrors and differential pairs.

The industrial reluctance to employ design automation for analog layout is rooted in several reasons. In particular, we consider the following three problems, which will be discussed in greater detail in Section III:

(A) The development and/or usage of automatisms is not intuitive: the setup of optimization algorithms is demanding due to their abstract nature; the usage of parameterized cells is more intuitive during design, but the programming of powerful hierarchical module PCells is quite challenging.

(B) The commonly available formal representations of design constraints are not suited to express complex functional circuit requirements with sufficient semantical conciseness.

(C) The inner workings of an automatism are usually not transparent to the user and give only little insight and control during its execution.

On the whole, all of these three problems add to one basic issue: analog layout automation does not adequately meet with the mentality of expert designers.

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

Göran Jerke, goeran.jerke@de.bosch.com, Vinko Marolt, vinko.marolt@de.bosch.com, Robert Bosch GmbH, Tübinger Straße 123, 72762 Reutlingen.

### B. Our Contribution

This paper introduces a *Constraint-Administered PCell-Applying Blocklevel Layout Engine* for layout automation (CAPABLE). CAPABLE is a framework allowing layout engineers to easily combine various automatisms into dedicated, custom-made, cohesive, executable and traceable scripts which can be used to automatically create block layouts for given schematic circuits. While there are no restrictions concerning the nature of the includable automatisms (i.e., algorithmic or procedural), CAPABLE is primarily meant to realize a generator approach which focuses in particular on the application of PCells. With respect to this intention, CAPABLE decidedly targets the three particular problems mentioned above by implementing the following features, as will be covered in Section IV:

(A) CAPABLE facilitates a new style of hierarchical PCell composition to create higher-level modules. That way, the PCells can be successively imposed onto each other in a truly "bottom-up" fashion that is much closer to a layout engineer's manual design style than the usual conception of module PCells which employ other sub-PCells internally.

(B) CAPABLE is integrated into the design flow via formally expressed constraints. For that purpose, new constraint types can be introduced to indicate the overall function of certain circuit structures. The inherent design requirements are then meant to be *implicitly* taken care of by respectively provided module PCells during the script execution.

(C) CAPABLE provides a convenient graphical user-interface (GUI) for the execution of the developed layout scripts. The GUI facilitates different pacing modes which allow the user to run a script step by step and thus to precisely track every single action that is thereby being performed in the layout.

Altogether, CAPABLE is a strongly designer-oriented engine that means to mimic a layout engineer's manual design style as closely as possible. With this objective, the execution of a layout script in CAPABLE is supposed to give users the impression of replaying a "recorded session" of manual layout design.

Our paper is organized as follows: Section II discusses the characteristics of optimization algorithms and parameterized cells. Section III details the three limitations described in Section I.A, while Section IV illustrates the respective solutions (see above) as put into effect by CAPABLE. Section V demonstrates our approach with a practical example and finally, Section VI concludes with a summary and an outlook.

## II. Related Work

### A. Optimization Algorithms

Since layout design is – from a mathematical perspective – an optimization problem, optimization algorithms are a natural choice to address that problem for automation. Characteristically, optimization algorithms translate the problem into an abstract representation and cycle through a repetitive loop of optimization and evaluation [1] to find an optimal layout "solution" with respect to certain optimization goals. Due to the complexity of the layout problem, it is usually divided into several steps such that an optimization algorithm can focus on one specific design task. The two main tasks in layout design are placement and routing, both of which have put forth a vast variety of algorithmic approaches. Routing can be further split into two consecutive steps called global routing and detailed routing. While this has become common practice in the digital domain, the routing of analog circuits is rather performed in one single step called area routing. Two of the first developed area routers are Lee's maze router [2] and Hightower's line router [3]. Algorithmic placement is, due to the huge variability of the devices, enormously challenging in the analog domain, compared to the standard cell approach taken for digital systems. Popular placement algorithms are min-cut placement [4], force-directed placement [5] and the widely spread Simulated Annealing [6].

Around the 1990s, EDA research – inspired by the huge success of optimization algorithms in the digital domain – has led to a plethora of works in which algorithmic approaches were combined into full-fledged tools for automated analog layout synthesis at block level, such as ILAC [7], LADIES [8], ALSYN [9] and INALSYS [10]. However, none of suchlike tools is known to have found evident industrial acceptance.

Optimization algorithms have the characteristic ability to *explicitly* take design constraints into consideration, but to do so they require all these constraints to be comprehensively expressed in a *formal* way. Unfortunately, it is enormously difficult to express complex analog design requirements via formal expressions [11]. This restrains an incorporation of valuable expert knowledge into the automatism and represents a major weakness of algorithmic approaches.

### B. Parameterized Cells

In contrast to optimization algorithms, PCells are not meant to self-intelligently *find* good layout solutions. Instead, a PCell executes a pre-defined series of operations in order to merely *reproduce* a customizable layout "result". On this basis, a module PCell, designed to create a layout for a particular analog basic circuit, has the natural ability to *implicitly* consider all inherent design constraints without the need to formalize them. This characteristic trait allows PCells to encapsulate valuable expert knowledge in an *informal* fashion and to produce layouts in full-custom quality. PCells are especially feasible to implement layout modules for which best-practice layout solutions are already known from experience.

Powerful module PCells covering multiple hierarchy levels have already been presented, e.g. [12]. However, their development still implies a trade-off between module variability and programming effort which is far from viable for industrial demands. Over the past years, a couple of sophisticated commercial PCell programming tools such as 1Stone [13], PyCell Studio [14] and GOLF [15] have been developed. In particular, the PCell Designer tool [16] facilitates a visual programming approach resembling a layout expert's manual design style. The intention behind that approach is to provide an intuitive platform with which design groups can easily create their own appropriate automatisms. With such tools, simple module PCells like current mirrors and differential pairs have become state-of-the-art in the industry, but the development of more complex PCells is obviously still not profitable enough in terms of layout productivity.

## III. Limitations Addressed By CAPABLE

In this section, each of the three limitations (A, B, C) briefly described in Section I.A is subsequently discussed in a subsection of its own.

### *A. Hierarchical Module PCells*

As already stated, CAPABLE does not represent an algorithmic approach like the synthesis frameworks mentioned in Section II.A, but focuses on the application of PCells. For that reason, this subsection examines the common conception of PCells and the inherent difficulties when composing them into more complex hierarchical module PCells in the traditional way. Generally speaking, a module PCell is a PCell that internally instantiates other PCells, and then creates additional layout shapes according to its dedicated purpose. Concretely, the considerations in this paper focus on module PCells which instantiate a set of layout devices and connect them by creating wire shapes and vias which make up a module-specific routing. In Section V, other types of PCells will also be shown.

The usage of a PCell requires data to flow through various "channels" which are given by the design environment (see Fig. 1). In principal, each channel can provide write access to modify a design object, and read access to retrieve information from a design object. One such channel is the design editor itself, by which a user can instantiate and customize a module PCell in a design ($a_1$). Naturally, the design editor also allows for editing the context of such a PCell, i.e., the other objects around that PCell in its design ($a_2$). To generate a certain layout, a PCell has to be evaluated. During this evaluation, the PCell performs an internal series of operations, thereby accessing its internal device instances or routing objects in a programmatic way (b). For example, this can include measuring the distance between two instances or setting the width of a routing wire. In contrast to these *internal* operations,

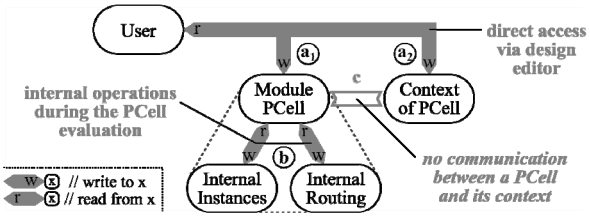



Fig. 1: Data channels required during the usage of a module PCell.

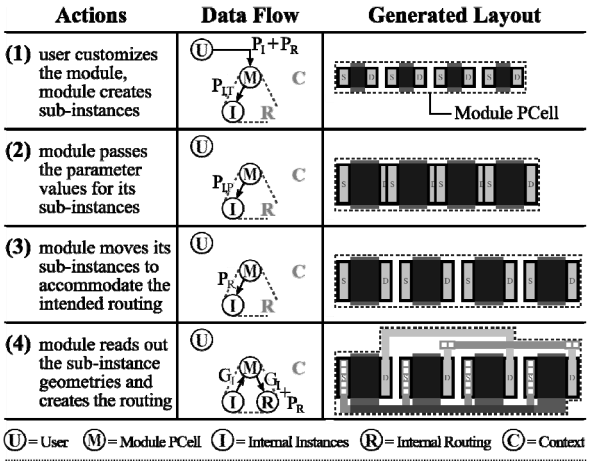



Fig. 2: Flow of data during the usage of a common module PCell.

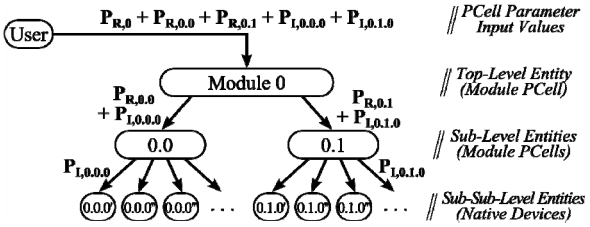



Fig. 3: Passing of parameter values in a traditional module PCell.

a PCell – in its traditional conception – has no ability to access its *external* context (c). Hence, a module PCell cannot actively "communicate" with its surrounding design, which in turn implies that *all* data a PCell requires must be passed to it via PCell parameters ($a_1$).

Fig. 2 illustrates the usage and evaluation of a module PCell to examine the data flow in detail. First of all (step 1), the user *U* instantiates and customizes the module PCell *M* by setting its parameter values. The parameters of a module PCell, as focused on in this paper, can be logically divided into a parameter set $P_I$, which specifies the module's internal instances, and a parameter set $P_R$, which specifies its internal routing. $P_I$ further consists of topology parameters $P_{I,T}$ which define the types and the number of the internal instances. With $P_{I,T}$ the module can initially create the internal instances *I* in their default configuration (1). $P_I$ also contains the device parameters $P_{I,P}$ which are directly passed through to the internal instances to set their dimensions as desired (2). Now that the number,

types, and sizes of the internal instances are set, they must be positioned in an arrangement that accommodates the intended routing. The arrangement is directly or indirectly defined by the routing parameters $P_R$, so this information must somehow be utilized to move each instance to its designated position (3). Then, the module can read-out geometrical data $G_I$ about its instances (e.g., their pin positions) and use that information in conjunction with $P_R$ to generate the appropriate routing $R$ (4). During all these actions, the context $C$ of the module PCell is never taken into account.

For simple modules, this common PCell conception is quite adequate, but it doesn't suit the traditional way of combining various PCells level by level to facilitate more complex hierarchical modules. As an example, two simple PCells can be implemented, each of which instantiates a couple of native transistors and connects them into a *current bank* or a *cascode*, respectively. A current bank and a cascode may be combined into a *cascode current mirror* PCell, which can then be employed for the realization of an *operational transconductance amplifier* (OTA). Such an OTA would then already span a total of four hierarchy levels.

At first glance, this approach is a natural strategy of using PCells as building bricks to form higher-level entities in a bottom-up fashion. But, since *every* parameter for *each* internal instance throughout the *entire* sub-hierarchy needs to be provided at the top-most module level, the flow of information in fact proceeds top-down. As shown in Fig. 3, all instance parameters $P_I$ and routing parameters $P_R$ for all internal entities must be given to the enclosing module by the user and are then internally distributed to the respective recipients. Technically, there is no limit to such a hierarchical composition of module PCells, but the cumulative amount of parameters at top level soon makes this approach virtually impractical. On one hand, the need to provide all parameters at module level escalates the development effort, and on the other hand, the unmanageable mass of parameters detracts from a PCell's usability. Furthermore, potential clashes of parameter names may require a cumbrous renaming scheme which in turn opposes the execution of device-specific validation mechanisms which are required to check and – if necessary – correct a user-entered parameter value. Even if all these problems can somehow be circumvented, the long-term maintenance of such a module PCell remains a critical issue. If, for example, a native transistor is equipped with a new parameter, then that parameter specification also needs to be added to the module PCell and to all its sub-modules throughout the module's entire hierarchy. Altogether, these drawbacks make the development of complex module PCells laborious, error-prone and inflexible.

### B. Formulation of Design Constraints

From an abstract perspective, a design constraint is a piece of information that supplements a schematic

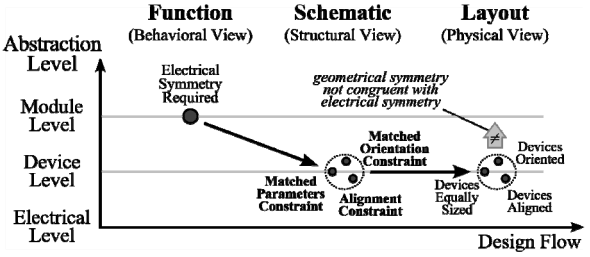



Fig. 4: Example of typical constraint usage in the design flow.

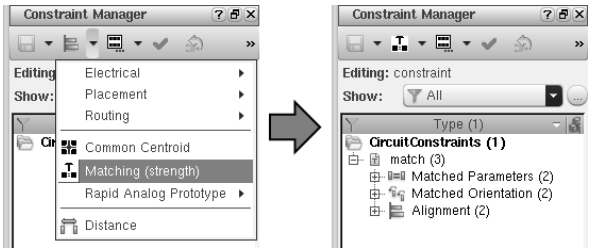



Fig. 5: In Virtuoso, matching is expressed with three constraints.

circuit to help attain an electrically functioning layout design. Initially, design constraints are informal pieces of expert knowledge in a designer's perception, but to be explicitly considered by an automatism, they have to be expressed in a formalized way.

Architecturally, circuit designers tend to think not of individual transistors, but of larger circuit structures that constitute functional units, keeping in mind the essential layout requirements that must be satisfied to ensure their proper electrical functioning. One of the most fundamental duties in analog layout design is the achievement of *matching*. Matching denotes a symmetric placement of belonging-together layout devices to ensure functional robustness against process variations, parasitic effects and physical influences.

Equivalent to the depictions in [17], Fig. 4 shows the three views of the Y diagram [18] side by side and the different abstraction levels as parallel lines. From a functional perspective (a), a circuit designer may be well aware of the need for electrical symmetry concerning a certain circuit structure. For the schematic design (b), that matching requirement may be formalized, but usually this is done by resorting to an alternative set of more concrete constraints. For example, in the design environment Cadence Virtuoso, a *Matching* constraint is indeed available for assignment (Fig. 5, left), but eventually it produces a set of three different, geometrical constraints (Fig. 5, right): *Matched Parameters*, *Matched Orientation*, and *Alignment*. In the layout design (Fig. 4), compliance with these constraints is supposed to achieve the desired matching, but that goal is not necessarily achieved, because the geometrical symmetry imposed by the three concrete, formal constraints is not entirely congruent with the electrical symmetry denoted by matching.

On one hand, the three constraints are missing two further matching criteria: the distance between the devices and their interdigitation. Large spaces between the devices or an unbalanced interdigitation pattern can substantially compromise the overall matching even if the other constraints are satisfied. On the other hand, the static nature of the constraints is insufficient to express geometric variability. In particular, *Alignment* requires that all devices share a common edge. For a single-row layout of a current mirror, this constraint would be satisfied. However, the devices of a current mirror can just as well be placed in two rows. Such an arrangement can even improve the matching, but is not tolerated by the alignment constraint because the alignment edge of the top transistors is different from the alignment edge of the bottom transistors. Furthermore, the routing of the devices can also severely impair the matching, but is not at all covered by the formally expressed design constraints above.

The advancement of *constraint engineering* in academia and in practice is promising, but it is still in its infancy, as shown by the above example. The choice of translating abstract design requirements (such as matching) into more concrete geometrical constraints is comprehensible because it allows for a formal verification of constraints and also makes them amenable to design automation. But, despite the ongoing developments on the consideration of constraints for verification and automation, the described shortcomings of contemporary constraint formulation represent a central limitation that still obstructs the evolvement of long-envisioned constraint-driven design flows.

### C. *Opacity of Layout Automatisms*

The potential acceptance of analog layout automation concepts not only depends on their mere technical merit, but also on a human factor that should not be underestimated. In general, engineers are rather skeptical about processes they cannot easily retrace or influence. Unfortunately, this is the case with many layout automation approaches, taking a set of input values and producing a respective layout output with only little means (or none at all) to let a user follow or steer the course of the automatism's execution.

For algorithmic approaches, tracking an automatism is difficult anyway since an algorithm may perform millions of cycles of optimization and evaluation (e.g., the random perturbations in the placement algorithm Simulated Annealing). The stochastic nature and non-deterministic behavior found in the majority of algorithmic layout automatisms detach an understanding of their actions even farther from the human grasp.

In the digital domain, these problems are of no concern due to the more quantitative quality of the design problem. But in analog design, the success of algorithmic automation is bound to a comprehensive and precise description of all relevant design constraints, which is in turn enormously intricate if the relevance

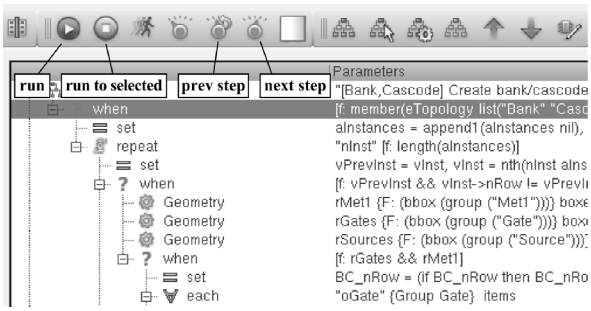



Fig. 6: Execution modes in the Cadence PCell Designer tool.

of each individual constraint, as well as their aggregate effect on the overall course of the automatism cannot be easily and intuitively understood by the designer.

In contrast, the inner workings of a PCell are easier to comprehend than those of an algorithm because PCells implement a pre-determined series of layout operations. On that basis, many PCell development tools offer different PCell execution modes which allow PCell programmers to trace the evaluation of a PCell in detail. For example, the PCell Designer tool displays a PCell's series of operations as a tree and supports running the PCell up to a selected operation call, as well as a step-by-step execution (see Fig. 6).

During PCell development, stepwise PCell evaluations are helpful features, but unfortunately, they are not available for the actual PCell usage. During design, a PCell is always executed in one single stroke and immediately reflects any user-made modifications of parameter values. For simple device PCells, this is quite adequate, but for rather complex modules more detailed control over the evaluation of a PCell would be immensely helpful to understand the structure of the module and the influence of the individual PCell parameters on the finally generated layout result.

## IV. The CAPABLE Approach

As will be individually discussed in each of the following subsections, CAPABLE specifically targets the three limitations (A, B, C) previously detailed in Section III. The combination of these efforts leads to a practical automation flow as depicted in Fig. 7.

Primarily, CAPABLE facilitates an *interface fabric* (A), with which design teams can implement *context-enhanced PCells*. Context-enhanced PCells are PCells that get equipped by the interface fabric with the ability to read and modify their design context. This allows module PCells to be hierarchically imposed onto each other in an intuitive, bottom-up fashion.

To generate the layout for a certain schematic circuit, CAPABLE provides a *constraint interpreter* (B) which allows designers to map a sequence of PCell-applying CAPABLE script commands to a particular constraint type. When assigning these constraints to components of a schematic circuit, the interpreter is

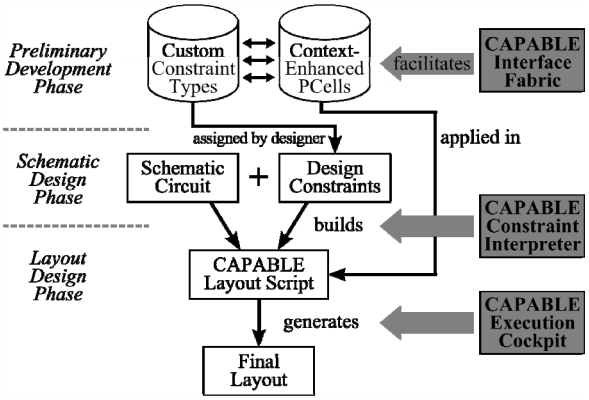



Fig. 7: Overview of the CAPABLE layout automation flow.

used to build the layout script from the respective script commands by which corresponding context-enhanced PCells are then automatically imposed on the constrained components in the layout. If desired, the layout script can also be manually edited and extended to perform further layout actions.

For carrying out a layout script, CAPABLE features a dedicated graphical *execution cockpit* (C). The execution cockpit is responsible for converting the used script commands into low-level code of the design environment's native programming language. To run the created code, CAPABLE's execution cockpit provides various different pacing modes which facilitate (amongst others) a stepwise execution of the script.

### *A. Interface Fabric*

Reprising the illustration from Section III.A, Fig. 8 again shows the common data channels employed for the usage of PCells: the direct editing of a PCell ($a_1$) and its design context ($a_2$) as well as a PCell's access to its internal design entities (b). The previously missing communication abilities between a PCell and its context are now facilitated with CAPABLE's interface fabric (c), that can be considered as being "wrapped" around a PCell during its instantiation. This enables read and write access to the context and can be utilized by PCell developers via providing three dedicated context-related functions per PCell:

- An *adapt* function (mandatory) is required prior to the PCell instantiation to analyze the designated PCell context and turn it into parameter values that can then be passed by CAPABLE to the PCell when instantiating it. This allows a PCell to dynamically adapt itself to the design it is placed in.
- A *modify* function (optional) can be implemented in order to let a PCell alter its context after the initial PCell instantiation. This allows a PCell to modify its surroundings to make them suitable for the PCell's own design requirements.
- An *update* function (optional) is only necessary for PCells that modify their context, and can be used to trigger a re-evaluation of the PCell. This allows a PCell to update itself according to the previously modified context.

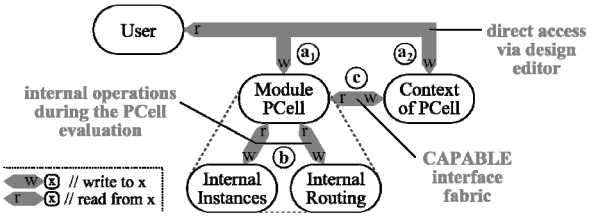



Fig. 8: Data channel enhancement via CAPABLE interface fabric.

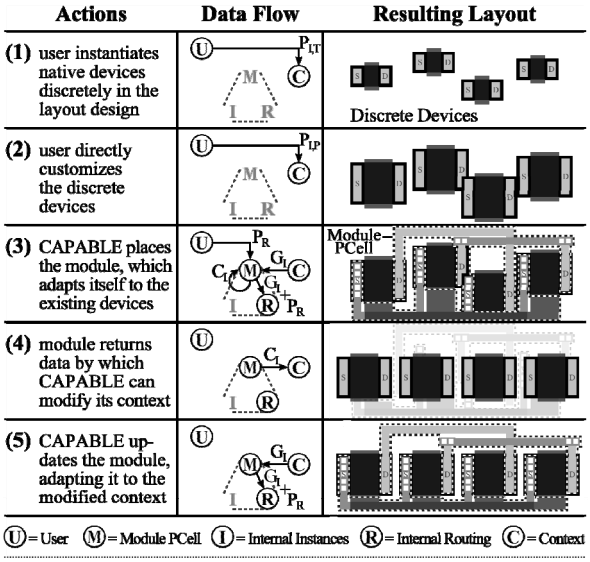



Fig. 9: Flow of data during the usage of a context-enhanced PCell.

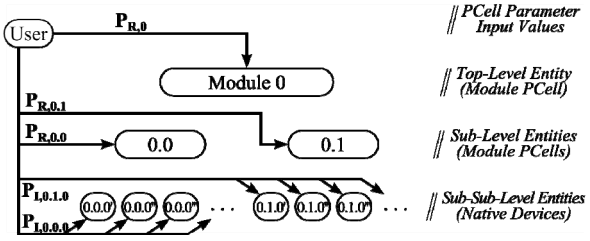



Fig. 10: Passing of parameter values with context-enhanced PCells.

With the above functions, CAPABLE can feasibly split a PCell instantiation into three consecutive steps: adapt, modify, update. After the execution of these three steps, the PCell and its context are supposed to be in perfect conformance with each other.

Fig. 9 illustrates the use of CAPABLE's interface fabric with a practical module PCell equivalent to the example presented in Section III.A. In contrast to the traditional PCell approach, the native devices need not be placed internally by the module PCell, but can be discretely instantiated by the user in the layout design (step 1). This has the benefit, that the devices can also be *directly* customized to set their dimensions (2). With the native devices prepared, CAPABLE can be

used to impose a context-enhanced module PCell on them (3): here, the *adapt* function reads out the pin positions of the discrete devices and passes this geometrical data $G_I$ to the PCell via pre-defined parameters, so the PCell can generate the desired routing considering the routing parameters $P_R$. During this PCell evaluation, the PCell internally computes the correction data $C_I$, which – in this example – specifies a translational move for each individual device. The correction data is stored on the PCell itself and can thus be used after the initial PCell instantiation by the *modify* function to displace all devices such that they are in line with the PCell's intended best-practice arrangement (4). Finally, CAPABLE executes the *update* function so the module PCell adjusts its routing to the newly positioned devices (5). In the end, the resulting layout is equal to that of the traditionally implemented module PCell shown in Fig. 2.

The benefit of this approach becomes obvious when more complex modules are considered. As illustrated in Fig. 10, the interface fabric of CAPABLE eliminates the need to pass all parameter values for all of a module's internal instances throughout the entire sub-hierarchy of the module. Instead, the instance parameters $P_I$ can be directly set on the native devices onto which the module (or, in this case: its sub-modules) are imposed. Equivalently, the routing parameters $P_R$ for the module and its sub-modules are also directly customizable. In that way, each building brick of the overall module PCell can be individually customized *before* it is measured by an adapting PCell on the next-higher hierarchy level. This is not possible with the traditional PCell approach, where the internal instances are really nested inside other modules: it implies that an internal instance is entirely evaluated during the evaluation of its enclosing PCell, so the internal instance cannot be autonomously customized *before* its enclosing PCell begins its own evaluation.

### B. Constraint Interpreter

The application of the context-enhanced module PCells is facilitated via design constraints. But, instead of resorting to low-level geometric constraints as described in Section III.B, CAPABLE employs dedicated higher-level constraints. These are supposed to be assigned to common circuit structures which represent functional entities and for which adequate module PCells are available on the layout side. Thereby it is the responsibility of the design team – depending on their focus (e.g., automotive applications) – to implement the desired module PCells, and also to specify the respective custom constraint types for a seamless integration into the CAPABLE flow. Of course, this presumes that the IC design environment allows for the specification of custom constraint types.

For every custom constraint type, a sequence of CAPABLE commands must once-only be declared in the constraint interpreter. Then, during the application

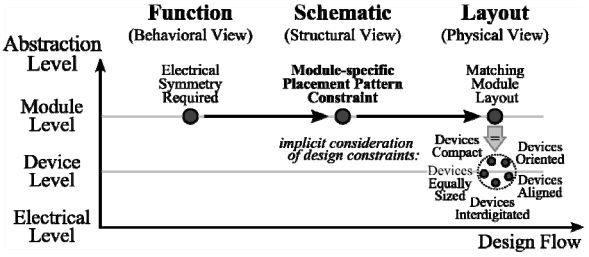



Fig. 11: Example of convenient constraint usage with CAPABLE.

of CAPABLE to a specific schematic circuit, the constraint interpreter turns every occurrence of that constraint type into a concrete call to that command sequence for applying it to the actual design entities the constraint has been assigned to. The final layout script results from the concatenation of the command sequences for all constraints encountered in the schematic design. The layout script can further be manually edited and extended, if desired.

CAPABLE provides a couple of script commands. The most important one of these is a *PCell* command that performs the instantiation of a context-enhanced module PCell according to the three-step approach described in Section A. Another command named *Group* allows to store a collection of design entities in a local variable. This is particularly useful if multiple PCells are to be imposed on the same set of devices. Further commands provided by CAPABLE are beyond the scope of this paper. Anyway, the set of provided commands is rather small, since the substantial automation powers are supposed to be covered by the PCells themselves, or other automatisms that may also be integrated into the overall CAPABLE framework (e.g., the *Modgen* tool, as will be shown in Section V).

As explained in Section II, PCells have the ability to consider intricate, low-level design constraints implicitly. Under that provision, the utilization of constraints is much more concise than in flows such as the one depicted in Section III.B. Fig. 11 shows how – instead of resorting to primitive geometric constraints – the need for electrical symmetry of a particular circuit structure can be simply expressed with a single custom constraint type in the schematic. That constraint should allow for the direct specification of a placement pattern by which the detailed device interdigitation can be explicitly defined. In the physical domain, a dedicated module PCell, specifically designed to respect the custom interdigitation pattern and all other requirements of the respective circuit structure, produces a module layout which achieves the desired matching (c). Thereby, the inherent low-level constraints are implicitly satisfied by the module PCell without the need to explicitly formulate them.

### C. Execution Cockpit

The CAPABLE approach has been implemented for the Cadence Virtuoso design environment. The

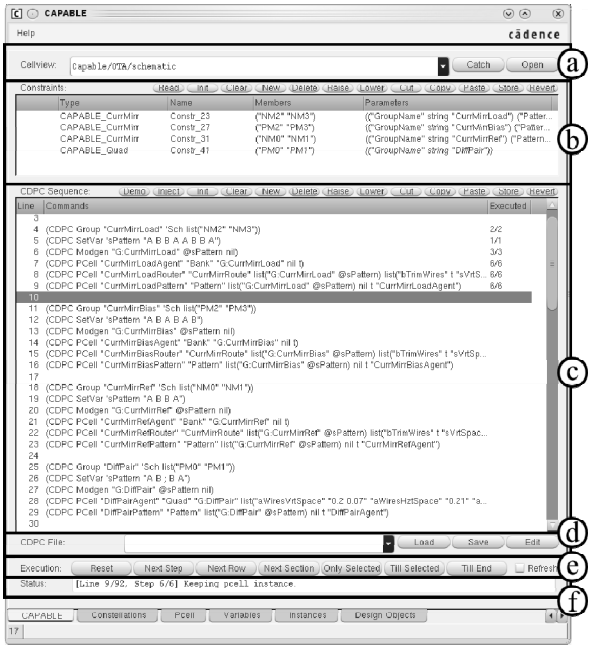



Fig. 12: The execution cockpit – CAPABLE's user-interface.

execution cockpit (CAPABLE's graphical user-interface) is shown in Fig. 12. With the input field (a), the user can enter or select a schematic circuit name. The circuit's design constraints are then read from the schematic and listed in table (b). The constraint interpreter builds the command sequences for the read constraints and concatenates them into the layout script which is then displayed in table (c). Using the elements of (d), layout scripts can be saved as text files and loaded from the file system at a later time. Also, a layout script text file can be manually edited with a plain text editor.

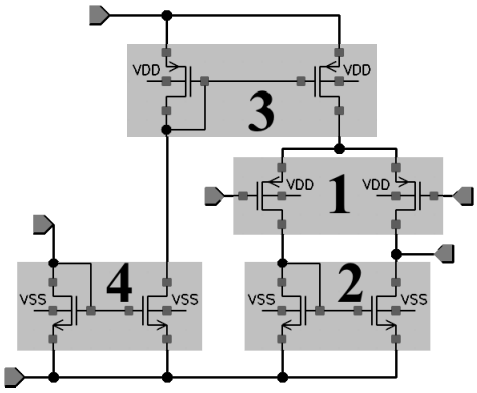
Each line of the layout script represents one call to a CAPABLE command. As already mentioned, the execution cockpit has to convert each command call into basic code of the design environment's native programming language (in our implementation: SKILL). Based on this conversion, the execution of a command is split into multiple fine-grain steps to let the user precisely track every single action that is being performed in the layout. The buttons (e) give control over the execution and allow the user to carry out

- a single step,
- one command,
- a section of commands,
- all commands up to the selected command,
- all commands till the end of the script is reached.

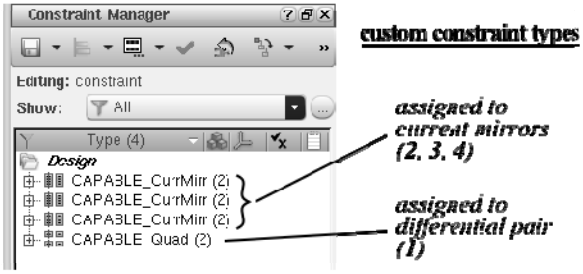
For every single step, the status field (f) displays textual messages that are generated during the execution.

## V. Example And Results

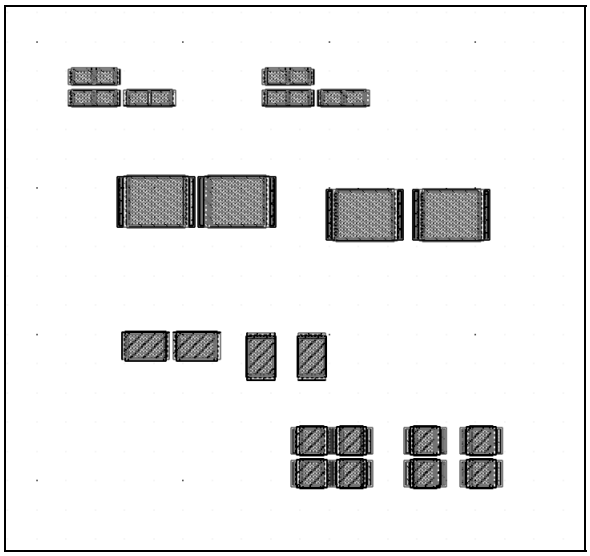
As an example for the application of the CAPABLE approach, Fig. 13 (a) shows the schematic diagram of




(a) Schematic diagram of the p-input OTA circuit example.




(b) Custom types of constraints, as assigned to the OTA circuit.



(c) Initially generated layout instances for the circuit transistors.

Fig. 13: An OTA circuit to exemplify the application of CAPABLE.

a p-input OTA. The circuit consists of one differential pair (1) and three current mirrors (2, 3, 4). As displayed in subfigure (b), a custom *CurrMirr* constraint is assigned to each current mirror, and a custom *Quad* constraint is assigned to the differential pair. The CurrMirr constraints allow the designer to define specific interdigitation patterns, whereas the Quad con-

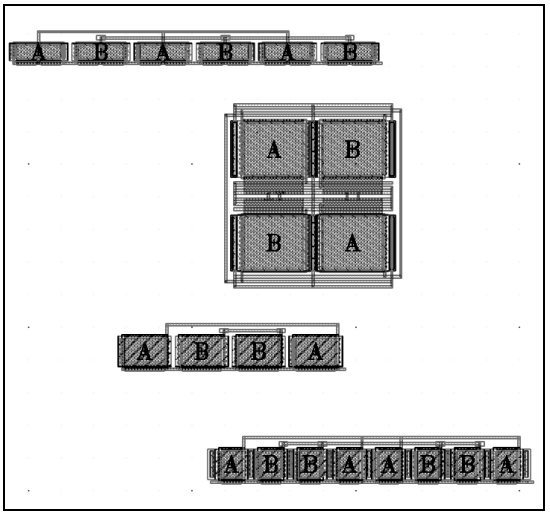
straint inherently denotes a common-centroid AB/BA placement. For the layout creation, CAPABLE initially instantiates the circuit transistors in the layout by calling *Generate from Source*. This is a native schematic-driven-layout functionality by which the dimensions of the layout devices are directly taken from the schematic circuit. The initial constellation is shown in subfigure (c) and represents the starting point for the subsequent constraint-administered imposition of context-enhanced module PCells (illustrated in Fig. 14).

First, the layout devices for each of the four circuit structures are interdigitated by CAPABLE. This is conveniently done using the native *Modgen* tool (but could have also been achieved with a PCell). Then, dedicated *quad* and *current mirror PCells* perform the detailed placement and module routing, which leads to the intermediate layout result shown in Fig. 14 (a).

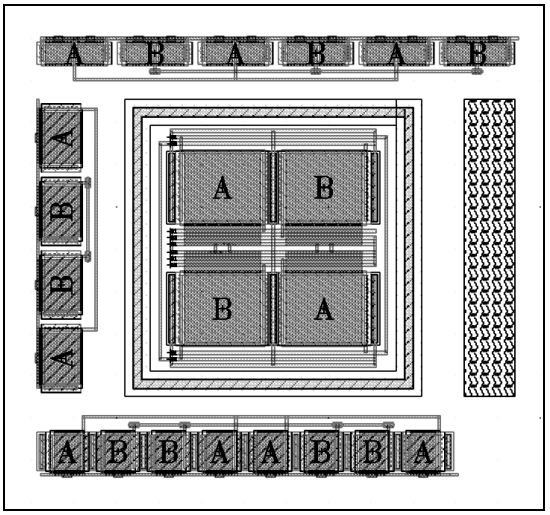
The layout script was manually extended to account for the further tasks of the layout creation. Thus, the differential pair is enveloped in a guard ring as created by a context-enhanced *isolation PCell* which smoothly clasps itself around the devices of the quad. Afterwards, a *placement PCell* moves the OTA modules to a feasible arrangement according to a pre-defined layout template. As shown in subfigure (b), the PCell also generates a blocking cap for reasons of symmetry and leaves sufficient space for the subsequent routing.

The routing between the modules is performed with a *wire PCell*, realizing the creation of multi-segment routing paths. While the paths can be explicitly specified via point lists, the PCell is (thanks to context-enhancement) also capable of snapping the routing wire to existing device terminals. The PCell furthermore supports transitions of the metal layer and automatically creates the required vias at the respective transition locations. In the given example, all inter-module connections are achieved with the above wire PCell, and finally another isolation PCell is put around the entire OTA. The resulting layout is presented in subfigure (c).

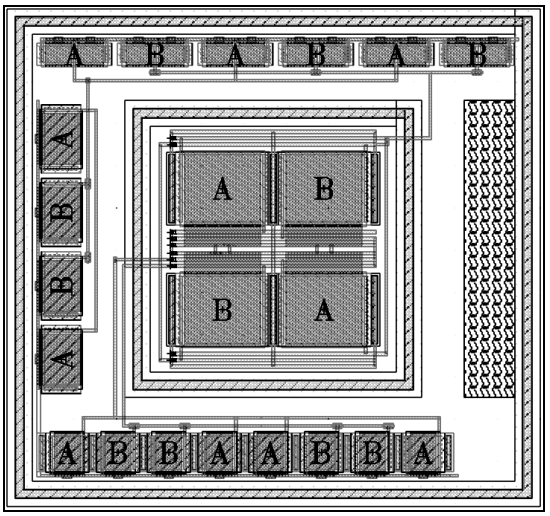
One might be skeptical about the efficiency benefit of CAPABLE's script approach compared to the mere manual creation of an analog block layout. However, a particular bottleneck are design *iterations* where even small modifications to the schematic circuit require laborious adjustments in the layout. In that regard, CAPABLE is supposed to be an especially profitable concept, allowing to easily re-generate the layout with just as little modifications. In general terms, every finished layout design represents one individual determined *solution*, whereas a CAPABLE script rather encapsulates a solution *strategy* that can be executed to generate finalized solutions. Conceptually, this approach is equivalent to the idea of PCells, but on a higher level of abstraction: a PCell performs native layout operations to create a design, while CAPABLE employs PCells to produce more complex results.



(a) Intermediate layout result with quad and current mirror modules.



(b) Intermediate placement with quad guard ring and blocking cap.



(c) Final layout result with full routing and enclosing guard ring.

Fig. 14: Layout results of applying CAPABLE to the OTA circuit.

## VI. Summary And Outlook

This paper presents a Constraint-Administered PCell-Applying Blocklevel Layout Engine for analog layout automation (CAPABLE). It implements three central features: (A) an interface fabric which facilitates the development of context-enhanced module PCells, (B) a constraint interpreter for transforming custom design constraints into command sequences that apply dedicated context-enhanced module PCells, and (C) an execution cockpit to carry out such command sequences as a concatenated layout script.

By implementing appropriate module PCells and assigning respective constraints to a schematic circuit, designers can use CAPABLE to build and run a layout script that generates the block layout in a bottom-up fashion by successively imposing module PCells onto each other. This approach is closer to a layout expert's manual design style and more intuitive than the traditional conception of complex module PCells covering multiple levels of hierarchy. In particular, CAPABLE eliminates the need to pass low-level device parameters across multiple hierarchy levels, it is able to consider intricate design requirements implicitly, and it allows the user to transparently track every single step that is performed during the layout creation.

In the long run, the CAPABLE approach should encourage design experts to abandon the common, manual style of layout creation in favor of rather capturing their invaluable solution strategies. For that purpose, future work on CAPABLE targets the automation of the script creation, e.g. by recording manual layout actions and converting them into script commands. Apart from that, the idea of context-awareness gives rise to an entirely new species of PCells that can not only *act* (on demand), but also *react* (to environmental changes) and even *interact* (with each other) – a vision that adds fundamentally new conceptions of flexibility and intelligence to the classical PCell concept.

## Acknowledgement

We would like to thank Andreas Gerlach for providing the OTA example circuit, as well as Thomas Burdick and Peter Herth for their support on PCells.

## References

[1] R. Rutenbar, "Analog CAD: Not done yet," *presented at NSF Workshop*, Arlington, Virginia, Jul. 8-9, 2009.

[2] C. Lee, "An Algorithm for Path Connections and Its Applications," *IRE Trans. on Electr. Comput.*, vol. EC-10, issue 3, pp. 346-365, 1961, doi:10.1109/TEC.1961.5219222.

[3] D. Hightower, "A Solution to Line-Routing Problems on the Continuous Plane," *Proc. 6th Design Automation Conference*, pp. 1-24, 1969, doi:10.1145/800260.809014.

[4] M. Breuer, "A Class of Min-Cut Placement Algorithms," *Proc. 14th Design Automation Conference*, pp. 284-290, 1977.

[5] N. Quinn Jr., "The Placement Problem as Viewed from the Physics of Classical Mechanics," *Proc. 12th Design Automation Conference*, pp. 173-178, 1975.

[6] S. Kirkpatrick, C. Gelatt Jr. and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, issue 4598, pp. 671-680, May 1983, doi:10.1126/science.220.4598.671.

[7] J. Rijmenants, J. Litsios, T. Schwarz and M. Degrauwe, "ILAC: An Automated Layout Tool for Analog CMOS Circuits," *IEEE Journal of Solid-State Circuits*, vol. 24, issue 2, pp. 417-425, Apr. 1989, doi:10.1109/4.18603.

[8] M. Mogaki, N. Kato, Y. Chikami, N. Yamada and Y. Kobayashi, "LADIES: An Automatic Layout System for Analog LSI's," *Int. Conference on Computer-Aided Design*, pp. 450-453, Nov. 1989, doi:10.1109/ICCAD.1989.76989.

[9] V. M. zu Bexten, C. Moraga, R. Klinke, W. Brockherde and K. Hess, "ALSYN: Flexible Rule-based Layout Synthesis for Analog IC's," *IEEE Journal of Solid-State Circuits*, vol. 28, issue 3, pp. 261-268, Mar. 1993, doi:10.1109/4.209992.

[10] Y. Kim, H. Cho and K. Yoon, "INALSYS: A Layout Automation System Based on Analog Layout Constraints," *Proc. of the 40th Midwest Sympos. on Circuits and Systems*, vol. 2, pp. 1209-1212, Aug. 1997, doi:10.1109/MWSCAS.1997.662297.

[11] J. Scheible and J. Lienig, "Automation of Analog IC Layout – Challenges and Solutions," *Proc. of the ACM 2015 Int. Symposium on Physical Design (ISPD'15)*, pp. 33–40, Mar. 2015, doi:10.1145/2717764.2717781.

[12] T. Reich, U. Eichler, K. Rooch and R. Buhl, "Design of a 12-bit Cyclic RSD ADC Sensor Interface IC Using the Intelligent Analog IP Library," *Proc. of ANALOG 2013*, vol. 239, Mar. 2013, ISBN:978-3-8007-3467-2.

[13] IPGen 1Stone Developer, [online] *http://ipgenme.de/eda-and-ip-products/1stone-developer.html* (accessed 2015-06-23).

[14] Synopsys PyCell Studio, [online] *http://www.synopsys.com/cgi-bin/pycellstudio/req1.cgi* (accessed 2015-06-23).

[15] Anaglobe GOLF, [onl.] *http://www.anaglobe.com/web/wp-content/ uploads/2014/03/PCell_brochure2010.pdf* (accessed 2015-06-23).

[16] G. Jerke, T. Burdick, P. Herth, V. Marolt, C. Bürzele *et al.*, "Hierarchical Module Design with Cadence PCell Designer," *pres. at CDNLive! EMEA*, Munich, Apr. 2015, session CUS02.

[17] S. Gohm, D. Marolt and J. Scheible, "Parametrisierte Layout-Module im analogen IC-Entwurf" (transl. "Parameterized Lay-out Modules in analog IC Design"), *MPC-Workshop*, vol. 48, pp. 57-63, Jul. 2012, ISSN:1868-9221.

[18] D. Gajski and R. Kuhn, "Guest Editor's Introduction: New VLSI Tools," *IEEE Computer*, Dec. 1983.

Daniel Marolt studied mechatronics at Reutlingen University, where he received the B.Eng. degree in 2008 and the M.Sc. degree in 2009. Since 2009 he works as an academic employee at Reutlingen University, where he pursues his Ph.D. degree in electrical engineering at the Robert Bosch Center for Power Electronics since 2011. His research interests focus on PCell-based automation of full-custom analog circuit and layout design.

Jürgen Scheible got his diploma in 1987 and the Ph.D. (Dr.-Ing.) degree (both in electrical engineering) in 1991 both from the University of Karlsruhe. From 1992 on, he was with the automotive electronics division of Robert Bosch GmbH. Since 2010, he is full EDA professor at the Robert Bosch Center for Power Electronics. His research interests include the automation of analog IC design with a special emphasis on physical design and methods for electro-thermal simulation.

Göran Jerke received his diploma degree in electrical engineering from the Dresden University of Technology. Since 1999, he is with the automotive electronics division of Robert Bosch GmbH, where he is responsible for constraint-driven design methodologies. His research interests include design flow concepts in general, as well as methods for physical design implementation, verification and robustness validation of IC designs.

Vinko Marolt accomplished a dual vocational training from 1973 to 1976 and worked in a developmental laboratory until 1979. Then he attended a technical school in Reutlingen and got the degree of certified technician in 1981. Since 1981 he works as a layout engineer and tool expert for the automotive electronics division of Robert Bosch GmbH. His motivation is the advancement of the layout design flow and of PCell development tools.

HOCHSCHULE
KONSTANZ
TECHNIK, WIRTSCHAFT
UND GESTALTUNG

# Synthese eines CRC-Number-Crunchers auf einem FPGA

Stefan Gebhart, Irenäus Schoppa

***Zusammenfassung*—Das hier beschriebene und auf einem FPGA vom Typ Spartan-3A DSP realisierte System dient dazu, auf besonders effiziente Weise die Häufigkeitsverteilung nicht erkannter fehlerhafter Nachrichten mit verschiedenen CRC-Polynomen zu berechnen. Damit die Berechnung in möglichst kurzer Zeit stattfindet, wurde das System aus 64 parallel arbeitenden Instanzen von CRC-Findern in mehrstufiger Fließbandorganisation aufgebaut. In der hier beschriebenen Ausbaustufe erreicht das System eine Gesamtleistung von 6,4 $\cdot 10^9$ Operationen in der Sekunde.**

***Schlüsselwörter*—FPGA, Spartan-3A DSP, CRC-Polynom, Cyclic Redundancy Check, Prüfsumme, PicoBlaze, Pipelining, Parallelisierung.**

## I. Einleitung

In kommunizierenden IT-Systemen werden Nutzdaten im Sender vor der eigentlichen Übertragung häufig um zusätzliche Informationen erweitert, anhand derer der Empfänger den fehlerfreien Empfang solcher Nachrichten feststellen kann. Eine der wichtigsten und heute weit verbreiteten Maßnahmen ist die Erweiterung der Nutzdaten um zyklische Codes (cyclic redundancy check, CRC), die auf der Modulo-2-Arithmetik mit Polynomen beruhen.

Es gibt zahlreiche standardisierte CRC-Polynome, z. B. CRC-16, CCITT-16 oder CRC-32, die für lange Nachrichten (> 1 kB) gut geeignet sind, aber weniger gut für kurze Nachrichten (≤ 5 Bytes). Einen Überblick über CRC-Polynome findet man z. B. in [1]- [3]. Kurze Nachrichten kommen jedoch häufig bei proprietären Kommunikationsprotokollen in messwertverarbeitenden Systemen (data aquisition system, DAS) vor. Sie bestehen meistens aus einem kurzen Statusfeld und einem oder einigen wenigen Messwerten. Eine mit einer CRC-Prüfsumme versehene Nachricht kann in der Regel keinen 100%igen Schutz garantieren. Es kommt gelegentlich vor, dass fehlerhafte Nachrichten empfangen, aber nicht als solche erkannt werden. In diesem Beitrag wird auf diese Besonderheiten genauer eingegangen.

S. Gebhart, stefan.gebhart@htwg-konstanz.de, und I. Schoppa, ischoppa@htwg-konstanz.de, sind Mitglieder der HTWG Konstanz, Brauneggerstr. 55, 78462 Konstanz.

## II. CRC-Berechnung mit der Papier-Bleistift-Methode

Das mathematische Prinzip der Polynomdivision auf der Basis der Modulo-2-Arithemtik (bitweises XOR) wird hier an vier vereinfachten Beispielen erörtert. In den Beispielen werden Nutzdaten durch das 10-stellige Bitmuster 1100001010 repräsentiert, und sie werden um eine 5-stellige CRC-Prüfsumme erweitert.

Anhand der Abbildung 1 kann man die Polynomdivision mit der Modulo-2-Arithemtik, wie sie im Sender vorgenommen wird, leicht nachvollziehen. Vor der Bildung der Prüfsumme werden Nutzdaten um ein Bitmuster mit einer Initialisierungssequenz erweitert. Diese Initialisierung kann im Allgemeinen mit einem beliebigen Bitmuster erfolgen. Häufig wird sie aber vollständig mit Nullen oder Einsen belegt. In diesem Bespiel erfolgt die Initialisierung mit dem 5-stelligen Bitmuster 00000. Der so gebildete 15-stellige Bitvektor 110000101000000 wird durch das 6-stellige Bitmuster 100101 dividiert. Dieses Bitmuster stellt die binäre Repräsentation des Polynoms $P(x) = x^5 + x^2 + 1$ dar, oder genauer formuliert, die seiner Koeffizienten $P(x) = 1 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$.

Das 10-stellige Resultat der Division 1101111110 ist nach der CRC-Berechnung nicht weiter von Interesse und kann verworfen werden. Viel interessanter ist der Rest der Division 00110, der hier den gesuchten CRC-Code, also die Prüfsumme, darstellt. Im Sender werden also die Nutzdaten um die berechnete CRC-Prüfsumme zu einem (10+5)-stelligen Bitmuster 110000101000110 erweitert und an den Empfänger übertragen.

Der Empfänger führt dieselbe Berechnung durch wie der Sender, indem er die empfangene Nachricht inklusive der CRC-Prüfsumme ebenfalls durch das Polynom $P(x) = x^5 + x^2 + 1$ teilt. Diese Division ist in Abbildung 2 dargestellt. Auch bei dieser Berechnung ist ausschließlich der Rest der Division relevant. Wenn dieser Rest mit der Initialisierungssequenz übereinstimmt, wird die Nachricht vom Empfänger als fehlerfrei akzeptiert.

In Abbildung 3 ist ein Beispiel einer fehlerhaften Nachricht gezeigt, bei der drei Bitstellen „umgekippt sind. Die fehlerhaften Bitstellen sind durch senkrechte Striche markiert. Mit Hilfe der Polynomdivision ermittelt der Empfänger den Restwert 01100. Stimmt

```
110000101000000 : 100101 = 1101111110
100101
 101011
 100101
  011100
  000000
   111001
   100101
    111000
    100101
     111010
     100101
      111110
      100101
       110110
       100101
        100110
        100101
         000110
         000000
          00110
```

Abbildung 1: CRC-Berechnung mit der Papier-Bleistift-Methode im Sender.

```
110000101000110 : 100101 = 1101111110
100101
 101011
 100101
  011100
  000000
   111001
   100101
    111000
    100101
     111010
     100101
      111110
      100101
       110111
       100101
        100101
        100101
         000000
         000000
          00000
```

Abbildung 2: CRC-Berechnung mit der Papier-Bleistift-Methode im Empfänger.

```
110000101000110
  |      |    |
100000111000010 : 100101
100101
 001011
 000000
  010111
  000000
   101111
   100101
    010100
    000000
     101000
     100101
      011010
      000000
       110100
       100101
        100011
        100101
         001100
         000000
          01100
```

Abbildung 3: Beispiel einer gelungenen Fehlererkennung.

```
110000101000110
  |      |    |
100000111001110 : 100101
100101
 001011
 000000
  010111
  000000
   101111
   100101
    010100
    000000
     101000
     100101
      011011
      000000
       110111
       100101
        100101
        100101
         000000
         000000
          00000
```

Abbildung 4: Beispiel einer misslungenen Fehlererkennung.

der Rest der Polynomdivision mit der Initialisierungssequenz (hier 00000) nicht überein, wird dies so interpretiert, dass die empfangene Nachricht fehlerbehaftet ist.

Das Beispiel in Abbildung 4 zeigt ebenfalls eine fehlerhafte Nachricht, in der auch drei Bitstellen „defekt“ sind. Im Vergleich zum vorherigen Beispiel liegt eine der fehlerhaften Bitstellen nur um eine Position weiter links als zuvor. Auch in diesem Fall kann der Empfänger mit Hilfe der Polynomdivision den Restwert ermitteln, der in diesem Fall 00000 ist. Dieser Restwert stimmt mit der Initialisierungssequenz überein, was bedeutet, dass die Nachricht fehlerfrei ist. In diesem besonderen Fall wurde eine fehlerhafte Nachricht nicht als solche erkannt. In diesem Zusammenhang stellen sich folgende Fragen:

1. Kommt es häufiger vor, dass fehlerhafte Nachrichten nicht als solche erkannt werden?
2. Wie sieht die Häufigkeitsverteilung nicht erkannter fehlerhafter Nachrichten bei verschiedenen Polynomen aus?
3. Gibt es optimale Polynome für kurze Nachrichten (16b … 32b)?

## III. CRC-Implementierung in Software und Hardware

Bevor man sich mit der Beantwortung der gestellten Fragen befasst, sollte man sich anschauen, wie eine Prüfsumme in der Software und in der Hardware berechnet werden kann. In der Software kommen prinzipiell zwei Implementierungen zum Einsatz: ein algo-

```
unsigned short icrc1(unsigned short crc,
                     unsigned char onech) {
  int i;
  unsigned short ans=(crc ^ onech << 8);

   for (i=0;i<8;i++) {
     if (ans & 0x8000)
       ans = (ans <<= 1) ^ 0x1021;
     else
       ans <<= 1;
   }
   return ans;
}
```

Abbildung 5: Algorithmische Nachbildung der Papier-Bleistift-Methode aus [3].

```
const unsigend short LUT[256] = {
    0, 151, 185,  46, 229, 114,  92, 203,
   93, 202, 228, 115, 184,  47,   1, 150,
  186,  45,   3, 148,  95, 200, 230, 113,
  231, 112,  94, 201,   2, 149, 187,  44,
  227, 116,  90, 205,   6, 145, 191,  40,
  190,  41,   7, 144,  91, 204, 226, 117,
   89, 206, 224, 119, 188,  43,   5, 146,
    4, 147, 189,  42, 225, 118,  88, 207,
   81, 198, 232, 127, 180,  35,  13, 154,
  ...

};
```

Abbildung 6: Auszug aus der tabellarischen Methode nach [7].

rithmisches und ein tabellarisches Verfahren. Das algorithmische Verfahren, das in Abbildung 5 als C-Funktion icrc1() dargestellt ist, beruht auf der algorithmischen Nachbildung der Papier-Bleistift-Methode. Dieses Verfahren hat leider den Nachteil, dass es auf einer Standard-CPU wegen eines hohen Anteils an Bitoperationen nicht besonders effizient abläuft. Deshalb geht man häufig zu einer tabellarischen Methode über, in der partielle Prüfsummen (z. B. für 256 Werte) vorausberechnet und in einer Look-Up-Tabelle abgelegt werden. In Abbildung 6 ist ein Auszug aus einer solchen Tabelle zu sehen. Der Algorithmus selbst reduziert sich auf eine einfache Schleife, in der auf die Tabelle zugegriffen wird und in der die Zwischenwerte byteweise verrechnet werden. Diese Art der Berechnung ist wesentlich effizienter als die algorithmische Methode und wird heute häufig in Embedded-Systemen und auf Mikrocontrollern eingesetzt.

Damit Berechnungen von Prüfsummen auf Mikrocontrollern möglichst in Echtzeit ausgeführt werden, lassen einige Chip-Hersteller sowohl spezielle Peripheriekomponenten [4], [5] als auch spezielle Maschinenbefehle im Befehlssatz von Mikrocontrollern [6] einbauen.

In der Hardware lässt sich die Prüfsumme mit Hilfe eines rückgekoppelten Schieberegisters besonders ressourcensparend implementieren. Diese bitserielle Implementierung kommt häufig dann zum Einsatz, wenn Daten bereits als Bitstream vorliegen und als solche verarbeitet werden. In Abbildung 7 ist die Realisierung des CRC-Polynoms $P(x) = x^6 + x^3 + x^1 + x^0$ dargestellt. Diese Realisierung besteht aus einer seriellen Anordnung von sechs Flipflops und drei XOR-Elementen mit einer Rückkopplung. Die Platzierung der XOR-Elemente ergibt sich aus dem zu realisierenden CRC-Polynom. In der bitseriellen Implementierung kann nur ein Bit der Prüfsumme pro Takt berechnet werden, was sich in manchen Anwendungen als Nachteil erweisen kann. Die zweite Möglichkeit, eine Prüfsumme in der Hardware zu implementieren, beruht auf parallelen Schaltnetzen [8], [9]. Diese Realisierung ist häufig ressourcenintensiv, ermöglicht aber die Berechnung der Prüfsumme für eine vorgegebene Nachricht innerhalb eines einzelnen Taktes. Einige Chip-Hersteller haben spezielle integrierte Bauteile (z. B. 74F401, [10]) im Programm, mit denen sich Prüfsummen für gängige CRC-Polynome berechnen lassen.

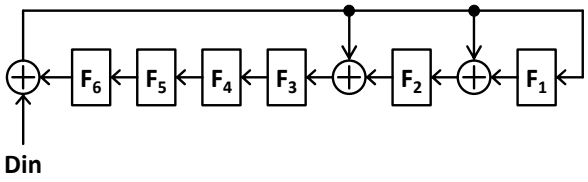



Abbildung 7: Bit-serielle Realisierung mit einem rückgekoppelten Schieberegister.

```
constant POLY: std_logic_vector :=
"110010111";
constant N: natural := 12;
constant K: natural := POLY'length-1;
...
for i in 0 to 2**N-1 loop
   msg1 := conv_std_logic_vector(i, N);
   crc1 := math_crc(msg1&crc0, POLY);
   msg2 := msg1 & crc1;
   for j in 1 TO 2**(N+K)-1 loop
      tmp  := conv_std_logic_vector(j, N+K);
      msg3 := msg2 xor tmp;
      crc3 := math_crc(msg3, POLY);
      if crc3=0 then
         idx := sum(tmp);
         tab(idx) := tab(idx) + 1;
         err := err + 1;
      end if;
   end loop;
end loop;
```

Abbildung 8: Hauptfunktion des CRC-Suchalgorithmus.

## IV. Suchalgorithmus und erste Untersuchungen

Um die zuvor gestellten Fragen beantworten zu können, ist es notwendig, einen möglichst universellen und leicht skalierbaren Suchalgorithmus zu implementieren. Der Suchalgorithmus, der in Abbildung 8 auszugsweise zu sehen ist, und bei dem auf den Abschnitt mit Deklarationen lokaler Variablen der Übersichtlichkeit halber verzichtet wurde, besteht im Wesentli-

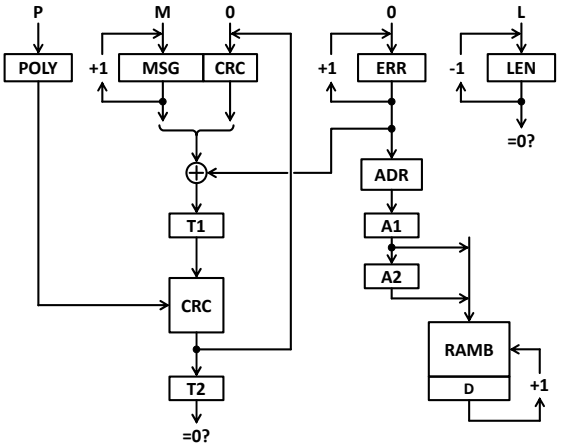



Abbildung 9: Rechenwerk in Fließbandorganisation.

chen aus zwei ineinander verschachtelten Schleifen. Die äußere for-Schleife durchläuft mit ihrem Index *i* alle Werte von 0 bis $2^N$-1. Diese Werte werden im nächsten Schritt in N-stellige Nachrichten umgewandelt. Zu jeder Nachricht wird die passende K-stellige Prüfsumme mit der Funktion math_crc() berechnet und an die ursprüngliche Nachricht angehängt. Die innere for-Schleife durchläuft mit ihrem Index *j* alle Werte von 1 bis $2^{N+K}$-1. Diese Werte werden in Bitvektoren umgewandelt und mit der ursprünglichen Nachricht per XOR verknüpft. Diese Bitvektoren stellen alle möglichen Fehlerkonstellationen dar, von einfachen einstelligen Bitfehlern bis zu mehrstelligen, sog. Burst-Bitfehlern. Für fehlerbehaftete Nachrichten wird erneut eine Prüfsumme mit der Funktion math_crc() berechnet. Ergibt die Berechnung dieser Prüfsumme den Wert Null, dann liegt der Sonderfall vor, dass eine fehlerhafte Nachricht nicht als solche erkannt wurde. Die Häufigkeit dieser Fälle wird in einer Tabelle protokoliert. Dazu bestimmt die Funktion sum() die Anzahl der fehlerhaften Bitpositionen, und in Abhängigkeit davon wird ein passender Eintrag in der Tabelle inkrementiert.

Der Suchalgorithmus selbst ist recht kompakt notiert, aber seine Ausführung auf einem modernen Rechner nimmt selbst bei kleinen Werten *N* und *K*, wie in diesem Beispiel bei $N = 12$ und $K = 8$, ca. 3 Stunden in Anspruch. Die lange Berechnungszeit resultiert daraus, dass der Suchalgorithmus insgesamt $2^N * 2^{N+K} = 2^{2N+K}$ Untersuchungen vornehmen muss, wobei *N* die Nachrichtenlänge und *K* die Prüfsummenlänge in Bits sind. Die Laufzeit des Suchalgorithmus steigt mit *N* und *K* exponentiell an, so dass Untersuchungen längerer Nachrichten oder längerer CRC-Polynome unmöglich erscheinen. In dem hier präsentierten Beispiel wurden insgesamt $2^{32} = 4\,294\,967\,296$ Fälle untersucht, in denen 16 773 120 fehlerhafte Nachrichten nicht als solche erkannt wurden. Das ergibt einen Anteil von 0,39053 % aller untersuchten Nachrichten.

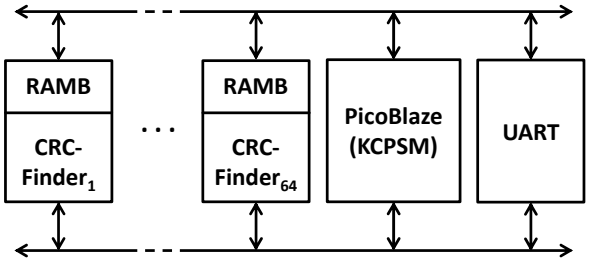



Abbildung 10: Blockschaltbild des Systems.

## V. Effiziente Implementierung des Suchalgorithmus

Es gibt vier Ansatzpunkte, den Suchalgorithmus effizient zu implementieren.

1. Der Suchalgorithmus beinhaltet viele Bitoperationen, die auf einem universellen Prozessor nur suboptimal ausgeführt werden können. Es bietet sich also an, den Suchalgorithmus direkt in Hardware auf einem modernen FPGA zu implementieren und ihn mit höchstmöglicher Taktfrequenz auszuführen.
2. Die Berechnung einer CRC-Prüfsumme mit einem rückgekoppelten Schieberegister ist laufzeitmäßig nicht optimal und benötigt mehrere Takte. Durch den Einsatz spezieller paralleler Schaltnetze ist es möglich, Signallaufzeiten von weniger als 10 ns zu erreichen und so die Berechnung einer Prüfsumme auf einen Takt zu reduzieren.
3. Die einfache Struktur des Suchalgorithmus ermöglicht eine parallelisierte Implementierung. Insbesondere die Ausführung der äußeren for-Schleife kann auf viele parallel arbeitende Instanzen (sog. CRC-Finder) verteilt werden.
4. Die einzelnen Schritte der inneren for-Schleife im Suchalgorithmus sind stark voneinander datenabhängig, so dass sie eigentlich nur sequentiell ausgeführt werden können und ihre Parallelisierung nicht möglich ist. Allerdings lassen sich diese Operationen in einem fließbandorganisierten Rechenwerk (Pipelining) besonders effizient umsetzen. Das Rechenwerk, dessen Registertransferstruktur in Abbildung 9 zu sehen ist, besteht aus einer 3-stufigen Fließbandstruktur zur Berechnung der Prüfsumme und aus einer 4-stufigen Fließbandstruktur zur Protokollierung der Fehler. Durch diese Organisationsform kann das Rechenwerk in jedem Takt eine neue Nachricht einschließlich der Prüfsumme auswerten.

Alle vier Ansätze wurden bei der Implementierung systematisch und vollständig umgesetzt. Als Zieltechnologie wurde ein XC3SD1800A-4FGG676C Spartan-3A DSP 1800A von Xilinx [11] gewählt. Dieser Baustein ist auf dem XtremeDSP Starter Kit [12] verfügbar.

Das gesamte System, dessen Blockschaltbild in Abbildung 10 schematisch dargestellt ist, setzt sich im

Tabelle 1: Verteilung und Häufigkeit von nicht erkannten fehlerhaften Nachrichten.

| | Polynome | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| #Bitstellen | 0x07 | 0x0E | 0x31 | 0x97 | 0x16 | 0x80 | 0xBB | 0xF9 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 16384 | 36864 | 20480 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 278528 | 40960 | 0 |
| 4 | 159744 | 241664 | 176128 | 151552 | 180224 | 724992 | 40960 | 73728 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1425408 | 204800 | 294912 |
| 6 | 1175552 | 1777664 | 1155072 | 1261568 | 1855488 | 2560000 | 503808 | 581632 |
| 7 | 0 | 0 | 0 | 0 | 0 | 3440640 | 1204224 | 1253376 |
| 8 | 4214784 | 4829184 | 4141056 | 3969024 | 4784128 | 3256320 | 2318336 | 2121728 |
| 9 | 0 | 0 | 0 | 0 | 0 | 2457600 | 2809856 | 2547712 |
| 10 | 5713920 | 5816320 | 5881856 | 6004736 | 5877760 | 1585152 | 2797568 | 2867200 |
| 11 | 0 | 0 | 0 | 0 | 0 | 737280 | 2646016 | 2834432 |
| 12 | 4116480 | 3362816 | 4018176 | 3977216 | 3244032 | 208896 | 1908736 | 2039808 |
| 13 | 0 | 0 | 0 | 0 | 0 | 49152 | 1171456 | 1196032 |
| 14 | 1232896 | 663552 | 1220608 | 1245184 | 765952 | 12288 | 741376 | 614400 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 303104 | 237568 |
| 16 | 155648 | 81920 | 180224 | 155648 | 45056 | 0 | 53248 | 86016 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 8192 | 24576 |
| 18 | 4096 | 0 | 0 | 8192 | 4096 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Summe | 16773120 | 16773120 | 16773120 | 16773120 | 16773120 | 16773120 | 16773120 | 16773120 |

Wesentlichen aus 64 parallel arbeitenden CRC-Findern, einer Instanz des Softcore-Prozessors PicoBlaze [13], einer Instanz der asynchron-seriellen Schnittstelle UART und einem Digital-Clock-Manager zusammen. Jeder CRC-Finder ist direkt an einen FPGA-internen RAM-Block angeschlossen, auf dem die Protokollierung von nicht erkannten, fehlerhaften Nachrichten vorgenommen wird. Alle CRC-Finder werden durch den PicoBlaze gesteuert. Der PicoBlaze hat auch die Aufgabe, Daten aus den einzelnen RAM-Blöcken abzuholen, diese zusammenzufügen und anschließend über den UART an einen PC weiterzusenden. Alle 64 CRC-Finder werden FPGA-intern mit 100 MHz getaktet. Die notwendige Taktfrequenz stellt der Digital-Clock-Manager zur Verfügung.

Die hohe Taktfrequenz und das fließbandorganisierte Rechenwerk ermöglichen einem CRC-Finder, bis zu 100 Millionen Berechnungen in der Sekunde durchzuführen. Durch die massive Parallelisierung erreicht das gesamte System eine Leistung von $6{,}4 \cdot 10^9$ Berechnungen in der Sekunde und übersteigt damit bei weitem die Leistung moderner PCs.

## VI. Experimente und Diskussion der Messresultate

Basierend auf dem implementierten System wurden die ersten Untersuchungen vorgenommen. Dabei wurden Häufigkeitsverteilungen nicht erkannter, fehlerhafter Nachrichten für alle 12-stelligen Nachrichten mit allen CRC-Polynomen 9-ten Grades (also für 8-stellige CRC-Prüfsummen) ermittelt. Das System musste insgesamt $2^{40}$ Suchvorgänge vornehmen. Die Berechnungen auf dem XtremeDSP-Board haben lediglich 172 Sekunden in Anspruch genommen. Tabelle 1 zeigt eine repräsentative Auswahl an Resultaten, die im Rahmen dieser Untersuchungen ermittelt wurden. Bei allen untersuchten CRC-Polynomen 9-ten Grades war die absolute Anzahl der nicht entdeckten, fehlerhaften Nachrichten immer gleich 16.773.120. Unterschiedlich ist jedoch bei jedem CRC-Polynom die Verteilung der nicht entdeckten, fehlerhaften Nachrichten in Abhängigkeit von der Anzahl der fehlerhaften Bitstellen. Bei manchen CRC-Polynomen (z. B. 0x07 oder 0x31) konnten alle ungeraden Fehler entdeckt werden, bei anderen (z. B. 0xBB oder 0xF9) nicht. Manche CRC-Polynome (z. B. 0x0E oder 0x31) können alle 1-, 2- und 3-stelligen Bitfehler sowie alle 17-, 18-, 19- und 20-stelligen Burst-Fehler detektieren, andere wiederum (z. B. 0x16 oder 0x80) sind nicht in der Lage, 2- und 3-stellige Bitfehler festzustellen.

## VII. Ausblick

Die Entwicklung des vorgestellten Systems ist zwar abgeschlossen, aber die gestellten Fragen konnten noch nicht umfassend beantwortet werden. Insbesondere wäre es notwendig, längere Nachrichten mit längeren CRC-Polynomen zu untersuchen. Die Leistungsfähigkeit des entwickelten Systems ist zwar sehr hoch, aber für Untersuchungen von 20-Bit-Nachrichten mit 8-stelligen CRC-Prüfsummen ist sie leider noch nicht ausreichend. Dazu müsste das System weiter parallelisiert werden und die FPGA-interne Taktfrequenz müsste weiter erhöht werden.

## LITERATURVERZEICHNIS

[1] P. Koopman, T. Chakravarty: „Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks", *The International Conference on Dependable Systems and Networks*, pp. 145–154, 2004.

[2] P. Koopman: „32-Bit Cyclic Redundancy Codes for Internet Applications", *Conference on Dependable Systems and Networks*, pp. 459–468, 2002.

[3] E. Stavinov: „A Practical Parallel CRC Generation Method", *Circuit Cellar Inc*., Jan. 2010.

[4] Renesas: „Using the M16C/62 CRC calculation periperial", Applikationsbericht, REU05B0007-0100Z, *Renesas Technology Corp*., Juni 2003.

[5] STMicroelectronics: „Using the CRC peripheral in the STM32 family", Applikationsbericht AN4187, *STMicroelectronics N.V.*, Juni 2013.

[6] Texas Instruments: „Cyclic Redundancy Check Computation: An Implementation Using the TMS320C54x", Applikationsbericht SPRA530, *Texas Instruments Inc.*, April 1999.

[7] Dallas: „Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor iButton™ Products", Applikationsbericht 27, *Dallas Semiconductor*, 1999.

[8] Cypress: „Parallel Cyclic Redundancy Check (CRC) for Hotlink", Applikationsbericht AN1089, *Cypress Semiconductor*, Nov. 2014.

[9] E. Stavinov: „A Practical Parallel CRC Generation Method", *Circuit Cellar Inc*., Jan. 2010.

[10] Fairchild: „74F401 CRC Generator/ Checker", Datenblatt DS009534, *Fairchild Semiconductor Corp.*, Aug. 1999.

[11] Xilinx: „Spartan-3A DSP FPGA Family Data Sheet", DS610, *Xilinx Inc*., Oct. 2010.

[12] Xilinx: „Spartan-3A DSP Starter Platform User Guide", UG454, *Xilinx Inc*., Jan. 2009.

[13] Xilinx: „PicoBlaze 8-bit Embedded Microcontroller User Guide", UG129, *Xilinx Inc*., Jun. 2011.

Stefan Gebhart studiert angewandte Informatik in der Vertiefung Computer and Network Engineering im 7. Semester an der HTWG Konstanz. Im Rahmen des Studiums befasst er sich u.a. mit der Synthese digitaler Systeme für FPGAs.

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.

# Analyse von Hardware/Software-Varianten einer Bildverarbeitungsapplikation auf Basis eines FPGA-SoCs

Dennis Sebastian Rieber, Joachim Gerlach

***Zusammenfassung*—Die vorliegende Arbeit untersucht Varianten für die Partitionierung einer konkreten Anwendung in Hardware und Software. Als Entwicklungssystem kommt das ZedBoard der Firma AVNET zum Einsatz, welches einen Zynq-7000 AP-SoC mit einem Dual-Core ARM Cortex A9 und einer Artix-7 FPGA-Einheit beinhaltet. Als Evaluierungsbeispiel aus dem Bereich der Bildverarbeitung wird eine dreigliedrige Filterkette verwendet. Für die Gegenüberstellung wurden Hardware/Software-Implementierungen der Filterelemente erstellt und in unterschiedlichen Varianten als Hardware-Variante auf dem FPGA sowie als Software-Varianten auf dem ARM-Hardcore sowie auf einem MicroBlaze-Softcore auf dem Board realisiert. Ferner wurde ein Hardware/Software Co-Design-Szenario realisiert, welches ein flexibles Testen und Konfigurieren verschiedener Hardware/Software-Partitionierungsvarianten ermöglicht.**

***Schlüsselwörter*—FPGA, Bildverarbeitung, Hardware/Software-Partitionierung, Zynq-7000, Hardware/Software Co-Design.**

## I. Einleitung

Moderne FPGA-Bausteine stellen neben Ressourcen zur Implementierung von Hardwareanteilen auch Prozessorkerne zur Verfügung, die es ermöglichen, Softwareanteile nativ auszuführen. Dies ermöglicht es, eine zu realisierende Funktionalität auf Hardware und Software zu partitionieren und durch deren Zusammenspiel zu realisieren. Das in dieser Arbeit verwendete ZedBoard bietet einen dieser modernen FPGA-Bausteine, zusammen mit einer breiten Auswahl an Peripherie und Anschlüssen für die Entwicklung von Anwendungen. Auf Basis dieser Hardware soll ein Evaluierungsbeispiel erstellt werden, anhand dessen die Partitionierung von Anwendungen veranschaulicht werden kann und die unterschiedlichen Varianten analysiert werden können. Das Evaluierungsbeispiel dieser Arbeit entstammt dem Bereich der Bildverarbeitung, da dies ein für FPGAs typisches Anwendungsszenario darstellt: die Verarbeitung von kontinuierlichen Datenströmen. Das Beispiel besteht aus drei einzelnen Filtern, was die Partitionierung der Anwendung auf dem ZedBoard ermöglicht. Für eine Partitionierung der Anwendung auf unterschiedlichen Teilsystemen des FPGA-Bausteins wird das Evaluierungsbeispiel in unterschiedlichen Varianten auf dem ZedBoard umgesetzt. Bei den verschiedenen Varianten werden neben Implementierungen auf dem FPGA auch Software-Implementierungen auf den verfügbaren ARM-Kernen genutzt.

Dennis Sebastian Rieber, rieberdennis@gmail.com und Joachim Gerlach, gerlach@hs-albsig.de, sind Mitglieder der Hochschule Albstadt-Sigmaringen, Poststraße 6, 72458 Albstadt.

## II. Grundlagen

Das als Entwicklungsplattform verwendete ZedBoard der Firma AVNET basiert auf einem „Xilinx RXC7Z020-1CLG484C Zynq-7000 All Programmable SoC (AP SoC)“ der Firma Xilinx. Dieser besteht aus einem ARM Cortex-A9 Processing System (PS) mit zwei Kernen und einer FPGA Programmable Logic (PL)[1] der Artix-7 Serie [1] [2]. Zur Verfügung stehen 512 MB DDR3 als Arbeitsspeicher sowie zusätzliche Cache-Speicher für den ARM Cortex A9 [3]. Eine schematische Darstellung der für diese Arbeit wichtigsten Komponenten ist in Abbildung 1 zu sehen. Die dort gezeigten Komponenten wurden für die Realisierung dieser Arbeit genutzt.

In den folgenden Abschnitten werden noch einige der verwendeten Technologien näher erläutert, die während der Arbeit an diesem Projekt verwendet wurden.

### *A. MicroBlaze*

Der MicroBlaze ist ein von der Firma Xilinx entwickelter Softcore und gehört zur Klasse der Reduced Instruction Set Computer (RISC) Prozessoren. Er ist optimiert für Xilinx FPGA-Systeme. Unter anderem sind folgende Eigenschaften für jedes MicroBlaze-System fest definiert:

- Zweiunddreißig 32 Bit-Register
- 32 Bit Instruktionen mit drei Operanden und zwei Adressierungsarten
- 32 Bit Adressbus

[1] Wird im späteren Verlauf von PS gesprochen, so ist das ARM Cortex-A9 Processing System gemeint, PL bezieht sich auf das FPGA-System.

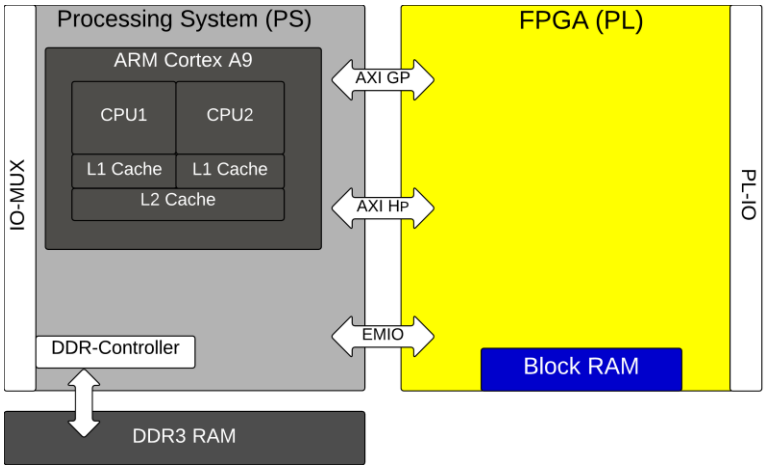



Abbildung 1: Vereinfachter schematischer Aufbau des Zynq-7000 AP-SoCs.

Der MicroBlaze verfügt über keine dedizierten Adressregister. Darüber hinaus lassen sich viele Parameter des Prozessors vom Benutzer an die Gegebenheiten anpassen. Es ist zum Beispiel möglich, den Prozessor sowohl als Little-Endian als auch als Big-Endian-System zu betreiben [4]. Die maximale Taktrate ist dabei von der verwendeten FPGA-Technologie und der Konfiguration des Softcores abhängig. Der Hersteller Xilinx gibt für den in dieser Arbeit genutzten FPGA eine maximale Taktrate von 226 MHz an [1] [2].

### *B. Advanced Extensible Interface (AXI)*

Das AXI-Protokoll ist ein Teil der ARM AMBA-Familie, einer Gruppe von Mikrocontroller-Bussen. AXI gliedert sich in zwei verschiedene Interfaces:

- AXI4(-Lite) wird für die bidirektionale Kommunikation von Memory-Mapped-Interfaces verwendet. Die Kommunikation läuft in Transaktionen ab und wird immer vom Master initiiert. Der Unterschied zwischen AXI4 und AXI4-Lite ist die Anzahl an Datentransfers pro Transaktion. Das AXI4-Lite-Interface ist auf einen Datentransfer pro Transaktion beschränkt. AXI4 hingegen kann bis zu 256 Datentransfers pro Transaktion durchführen [5].
- AXI-Stream ist ein unidirektionales Master/Slave-Protokoll, das unbegrenzt viele Daten während einer Transaktion übermitteln kann. Die Kommunikation ist beendet, wenn der Master mit dem TLAST-Signal anzeigt, dass er keine weiteren Daten zu versenden hat. Es besitzt nicht dieselbe Art von Memory-Mapped-Interface wie AXI4-(Lite). Beim Datentransfer ist zu beachten, dass Master und Slave wissen müssen, wie sie die Daten zu interpretieren haben, die übertragen werden, deshalb kann nicht jeder AXI4-Stream-Master mit jedem AXI4-Stream-Slave arbeiten, was die Wiederverwendbarkeit von AXI4-Stream-IPs einschränkt [5].

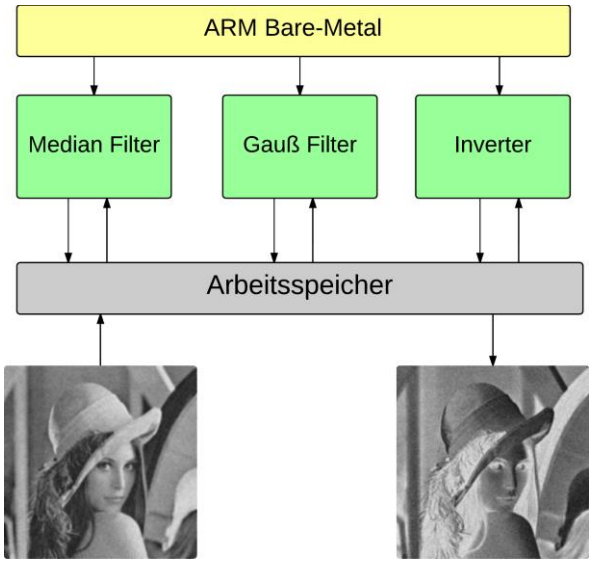



Abbildung 2: Schematischer Aufbau des Evaluierungsbeispiels.

## III. ENTWICKLUNG DES EVALUIERUNGSBEISPIELS

Das Evaluierungsbeispiel ist aus dem Bereich der Bildverarbeitung und reiht eine Kette von Filtern für die Manipulation eines Bildes hintereinander.
Das zu verarbeitende Bild wird im Arbeitsspeicher abgelegt und dort von den einzelnen Filtern verarbeitet. Für die Evaluierung werden Graustufenbilder verwendet, bei denen jedes Pixel durch einen 8-Bit-Grauwert beschrieben wird. Das Bild wird vor der Verarbeitung in eine Byte-Matrix umgewandelt. Dies vereinfacht das Implementieren der Filter, ohne die Eigenschaften der Filter einzuschränken oder zu verändern. Abbildung 2 zeigt ein Schema des Evaluierungsbeispiels. Das Evaluierungsbeispiel ist dabei nicht für die direkte Bildakquisition (durch eine Kamera oder ähnliches) verantwortlich. Im Folgenden werden die verwendeten Filter vorgestellt und deren Arbeitsweise erläutert.

### A. Inverter

Der Inverter führt eine Grauwerttransformation durch und erzeugt ein Negativ des ursprünglichen Bildes. Dies kann die Wahrnehmung von feinen Strukturen im Bild verbessern [6]. Der Inverter wurde gewählt, da die eigentliche Filteroperation von geringem Aufwand ist und seine Laufzeit von der I/O-Geschwindigkeit abhängt. Gleichung 1 zeigt die Berechnung des neuen Grauwerts eines Pixels in einem Graustufenbild. Hierbei beschreibt $G_{\text{neu}}$ den Grauwert des Pixels nach der Berechnung, $G_{\text{now}}$ den aktuellen Grauwert und $G_{\text{max}}$ den maximal möglichen Grauwert, den ein Pixel annehmen kann.

$$G_{\text{neu}} = G_{\text{max}} - G_{\text{now}} \tag{1}$$

### B. Gauß'scher Filter

Es handelt sich dabei um einen nicht-linearen Filter, der eine zweidimensionale Gauß'sche Glocke annähert. Die Annäherung wird durch Gewichtung der Grauwerte in der momentan betrachteten Maske erzeugt. Die Maske enthält immer das momentan betrachtete Pixel und dessen Nachbarn. Nach der Berechnung des neuen Grauwerts mithilfe der Nachbarn wird der Grauwert des Pixels im Bild ersetzt und das nächste Pixel betrachtet.
Der Gauß'sche Filter gehört zur Klasse der lokalen Operatoren und die Maske des Filters wird in dieser Arbeit mit einer Maskengröße von 3x3 verwendet. Mit diesem Filter kann die Leistung der verschiedenen Varianten bei arithmetischen Operationen gemessen werden. Er wird für das Entrauschen von Bildern verwendet [7] und kann durch Gleichung 2 beschrieben werden. $G_{neu}$ ist der neue Grauwert des momentan betrachteten Pixels. $W$ und $H$ sind die Breite und die Höhe der Maske und $w$ und $h$ die Laufvariablen zum Iterieren durch die Maske. $K$ ist die Matrix mit den Gewichten des Gauß'schen Filters und $M$ die Maske, die das aktuell betrachtete Pixel und dessen Nachbarn erfasst.

$$G_{\text{neu}} = \frac{1}{16} \sum_{w=0}^{W} \sum_{\text{h}=0}^{H} M(w,h) \cdot K(w,h) \tag{2}$$

$$\text{mit} \quad K = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

### C. Median-Filter

Der Median-Filter ermittelt den Median aller Werte innerhalb der Maske und nutzt diesen als neuen Wert für das betrachtete Pixel. Dadurch können Pixel, die einen extremen Wert im Vergleich zu den umliegenden Pixeln haben, eliminiert werden. Das Vorhandensein dieser spontanen Extremwerte wird auch *Salt and Pepper Effect* genannt [8]. Der Median-Filter zählt zur Klasse der lokalen Operatoren. In dieser Arbeit wird ein Filter mit der Maskengröße 3x3 verwendet. Der Median-Filter wurde gewählt, da das Sortieren von Werten oft ein wichtiger Bestandteil von Anwendungen ist und viele Vergleichsoperationen benötigt.

Tabelle 1: Implementierungsvarianten des Evaluierungsbeispiels.

| Name | Beschreibung |
|---|---|
| ARM Bare-Metal | Das PS des ZedBoard verfügt über eine ARM Cortex A9 CPU, auf der das Evaluierungsbeispiel als Software ohne Betriebssystem implementiert wird. |
| OS | Das Evaluierungsbeispiel wird als Software auf einem Betriebssystem (Operating System, OS) ausgeführt. Das OS wird dabei auf dem PS des ZedBoards ausgeführt. |
| MicroBlaze | Das Evaluierungsbeispiel wird als Software auf einem MicroBlaze Softcore ausgeführt. |
| AXI-Stream | Die Filter werden als Schaltung für die Konfiguration auf der PL entwickelt und verwenden dabei das AXI4-Stream Protokoll. Der Programmfluss wird durch Software gesteuert. |

## IV. VARIANTEN

Die in dieser Arbeit untersuchten Implementierungen des Evaluierungsbeispiels werden in Tabelle 1 kurz vorgestellt, bevor diese in den kommenden Abschnitten näher erläutert werden. Abbildung 3 zeigt schematisch, wo die verwendeten Varianten auf dem ZedBoard realisiert sind und auf welche Ressourcen diese zugreifen. Die Verbindungen zwischen den Komponenten sind Daten- sowie Steuerungsverbindungen. Auf der linken Seite von Abb. 4 ist das Processing System (PS) zu sehen, dass den ARM Cortex A9 enthält und rechts ist die Programmable Logic (PL) mit der FPGA-Einheit abgebildet. Die Datenverbindungen zwischen den IPs auf der PL und zwischen PS und PL werden als verschiedene Varianten des AXI4-Protokolls genutzt. Der Block <*AXIS-Filter*> steht für die Filter, die als Schaltung für die Variante AXI-Stream auf der PL realisiert werden. Die AXI-DMA ist das Interface zwischen Arbeitsspeicher und AXI-Stream IP. Über einen High-Performance Port kommuniziert die DMA mit dem DDR3-Controller und kann Daten blockweise aus dem DDR3-RAM lesen und schreiben. Alle Varianten, die eine Softwareimplementierung verwenden – ARM Bare-Metal, OS und MicroBlaze – greifen auf dieselbe Implemen-

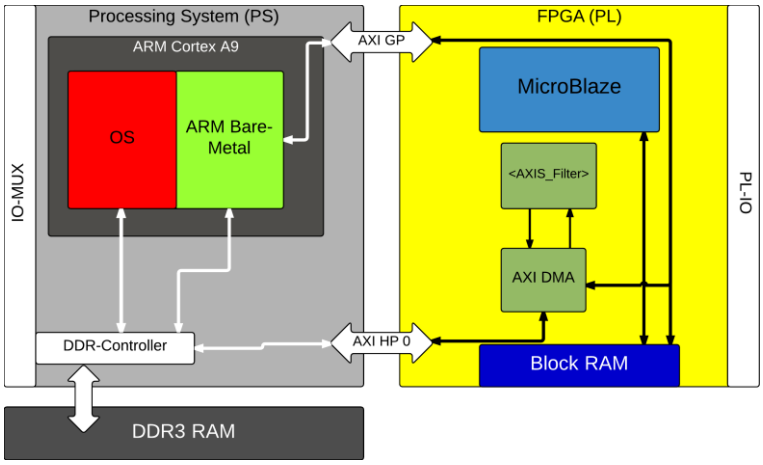



Abbildung 3: Vereinfachte Architektur und Verteilung aller Varianten auf dem ZedBoard.

tierung für die Filter in C-Code zurück, kompiliert für die entsprechende Variante. Diese iterieren über die Dimensionen x und y des Bildes und wählen jedes Pixel und ggf. dessen Nachbarn für die algorithmische Verarbeitung aus. Der Speicherzugriff wird durch get/set-Methoden maskiert, um eine Portierung der Filter zwischen den einzelnen Varianten zu vereinfachen. Für die Ermittlung des Median beim Median-Filter wird wegen seiner stabilen Laufzeit ein Quicksort verwendet.

### *A. Variante ARM Bare-Metal*

Die Variante ARM Bare-Metal führt das Evaluierungsbeispiel als Software aus, welche auf einer dünnen Treiberschicht ohne Betriebssystem auf einem der ARM Cortex A9-Kerne ausgeführt wird. Bei dieser Variante können zwei Arbeitsspeicher verwendet werden, zum einen der DDR3-RAM und zum anderen der Block-RAM, der auf der PL verbaut ist. Wenn die Software nur das DDR3-RAM verwendet, ist keine Konfiguration der PL nötig. Wenn der Block-RAM (BRAM) verwendet werden soll, muss der Zugang zu diesem auf der PL konfiguriert werden. Mit einem BRAM-Controller können dabei nur 128 kByte gleichzeitig angesprochen werden, was die mögliche Größe der verarbeiteten Bilder einschränkt. Der Block-RAM wird über einen AXI-Interconnect an einen General-Purpose-Port (GP-Port) des Processing Systems angeschlossen.

### *B. Variante OS*

Das Betriebssystem Xilinux, eine Linux-Distribution basierend auf Ubuntu 12.04, wird auf dem ARM Cortex A9 ausgeführt und nutzt als Arbeitsspeicher den DDR3-RAM des ZedBoards. Als Festspeicher wird eine EXT4-partitionierte SD-Karte verwendet. Auf dieser SD-Karte befindet sich das Boot Image und das Dateisystem der Xilinux-Installation. Als Bildschirmausgang wird der VGA-Port verwendet und Peripherie (Maus, Tastatur etc.) wird an den USB-OTG-Port über einen USB-Hub angeschlossen. Da hier ein Dateisystem und Treiber vorhanden sind, konnten die Bilder über einen USB-Stick direkt auf dem System abgelegt werden. Die Software des Evaluierungsbeispiels wird vor Ort mit dem g++ Compiler kompiliert. Das Speichern der Bilder erfolgt hier über ein mit malloc() angelegtes, zweidimensionales Array. Für die Zeitmessung wird auf dem Betriebssystem die Funktion gettimeofday() verwendet und die erhaltenen Werte werden in Millisekunden umgerechnet. Die Dekompression der JPEG Dateien wird bei dieser Variante direkt im Evaluierungsbeispiel durchgeführt mithilfe der Bibliothek libjpeg [9]. Die Implementierung für das Lesen und Schreiben einer JPEG-Datei basiert auf Beispiel-Code [10], der für diese Arbeit angepasst wurde.

### *C. Variante MicroBlaze*

Der MicroBlaze Softcore auf der PL wird ebenfalls Bare-Metal, also ohne OS, betrieben. Als Speicher, in dem sich das Bild befindet, wird ein sich auf der PL befindender Block-RAM (BRAM) verwendet. Dabei können, wie auch bei der Variante ARM Bare-Metal, 128 kByte von einem BRAM-Controller angesprochen werden. Die Implementierung der Filter und des Speicherzugriffs ist hier dieselbe wie bei der Variante ARM Bare-Metal. Der MicroBlaze benötigt seinen Takt von einer externen Quelle. In diesem Fall wird Clock vom PS gesteuert. Der MircoBlaze taktet genau so wie die AXI-Stream-Variante mit 100 MHz. Nach dem Start wird der MicroBlaze über eine Steuerungsflag im Block-RAM gesteuert. Über die Flag wird dem MicroBlaze angezeigt, welcher Filter verwendet werden soll und der MicroBlaze zeigt an, dass er mit der Verarbeitung der Daten fertig ist. Der MircoBlaze läuft in einer Hauptschleife und wartet auf Befehle. Das Bild, das vom MircoBlaze verarbeitet werden soll, wird von der Steuerung im Block-RAM abgelegt. Die Anfangsadresse des Datenbereichs wird vor der Kompilierung definiert. Für die Steuerung des

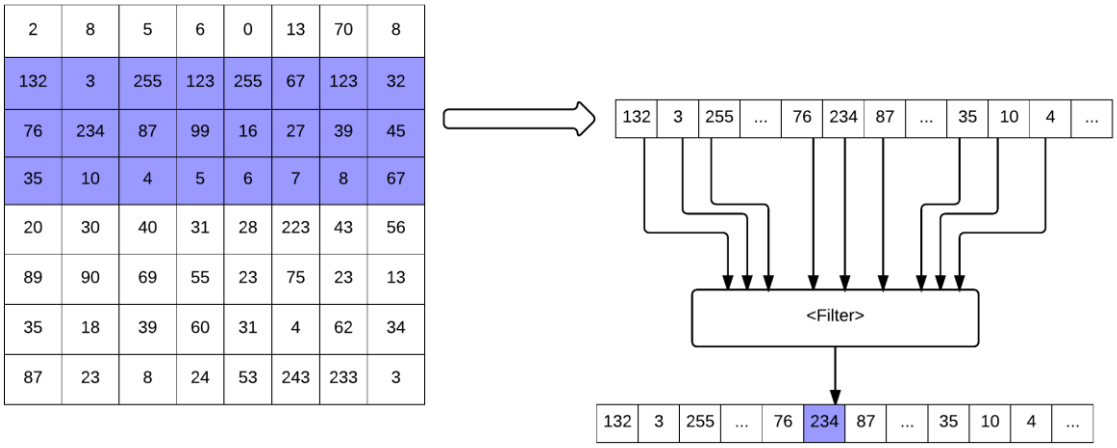



Abbildung 4: Sliding Window.

MicroBlaze ist der Zugriff auf denselben Abschnitt des Block-RAMs nötig, auf den auch der MicroBlaze Zugriff hat.

*D. Variante AXI-Stream*

Die Filter des Evaluierungsbeispiels werden als AXI4-Stream IPs implementiert. Hierbei werden nur die Filter an sich als Schaltung implementiert, deshalb ist diese Variante keine reine Hardware-Lösung, sondern ein Hardware/Software Co-Design. Die Umsetzung der Filter als IPs geschieht mithilfe von High-Level Synthese (HLS). Die Daten werden von einer AXI-DMA IP aus dem Arbeitsspeicher gelesen, an die Filter-IP geleitet und die verarbeiteten Daten zurück in den Arbeitsspeicher gelegt. Es wird jeweils eine DMA pro Filter für den Transfer der Daten zwischen Speicher und Filter-IP verwendet. Die Steuerung der AXI-DMAs und der Filter-IP erfolgt in dieser Arbeit durch Software auf einen ARM-Kern des PS. Die Datenleitung der DMAs wird über einen High Performace Port des Zynq-7020 mit dem DDR3-Controller verbunden. Die Steuerleitungen werden über einen AXI-Interconnect mit einem GP-Port an das Processing System angebunden. Über diese Kontrollleitungen wird der Datentransfer der DMAs gesteuert.

Die DMA kann in zwei verschiedenen Modi betrieben werden: dem Simple Mode und dem Scatter-Gather Mode. Für diese Arbeit ist der Simple Mode ausreichend, da die Daten im Speicher zusammenhängend untergebracht sind. Dabei wird über den AXI-4 Slave Port die Anfangsadresse der zu übertragenden Daten in ein Register geschrieben und in ein weiteres Register wird die Anzahl der zu übertragenden Bytes geschrieben [11]. Die Implementierung des Hardware/Software Co-Designs erfolgt in mehreren Schritten:

- Entwicklung der Filter IP mithilfe des High-Level Synthese (HLS) Tools von Xilinx. Hier wird die Synthese von C nach VHDL oder Verilog durchgeführt.
- Einbinden der IP in ein Block Design, aus dem ein Bitstream erstellt werden kann. Dieser Schritt wird mit Vivado 2014.3 umgesetzt.
- Entwicklung der Software mit der Xilinx SDK für die Steuerung der PL und damit der entwickelten IP.

Da bei den lokalen Operatoren Zugriff auf 3 Zeilen des Bildes erfolgt, wird ein Linebuffer mit Kapazität für 3 Zeilen implementiert. Dieser wird mit einem FIFO-Register von dreifacher Zeilenlänge realisiert. Das Bild wird durch den Linebuffer „geschoben“ und dabei werden die Pixel zur Verarbeitung an den entsprechenden Stellen abgegriffen. Dieses Vorgehen wird auch Sliding Window genannt, veranschaulicht in Abbildung 4 [12]. Die Dimensionen des zu verarbeitenden Bildes werden beim Median-Filter und beim Gauß’schen Filter über separate Datenleitungen an die Filter-DMA übertragen. Hierbei ist vor allem die Breite des Bildes von Bedeutung, da dieses die Länge des Linebuffers bestimmt. Bei der Synthese für die Filter werden verschiedene Direktiven verwendet. Die geschachtelten for-Schleifen werden durch Pipelines in mehrere Stufen unterteilt.

Durch den Linebuffer und die Pipelines entsteht eine Verzögerung zwischen dem Eingang und dem Ausgang eines Pixels. Der Median-Filter verwendet ein Sortiernetzwerk für die Ermittlung des Medians der aktuell betrachteten Pixel, dieses ist ebenfalls durch eine Pipeline gestuft. Die Ermittlung des Median benötigt bei der verwendeten Implementierung neun Taktzyklen. Die gewählte Implementierung erreicht nach Schätzungen des HLS-Tools eine maximale Taktrate von 199,2 MHz. Beim Gauß’schen Filter

Tabelle 2: Laufzeiten aller Varianten.

| | Laufzeiten bei Auflösung 300 x 300 | | | |
|---|---|---|---|---|
| Variante | Median (ms) | Gauß (ms) | Inverter (ms) | Gesamt (ms) |
| ARM Bare-Metal (DDR3) | 217 | 57 | 11 | 286 |
| ARM Bare-Metal (BRAM) | 378 | 220 | 44 | 643 |
| OS (ohne Last) | 229 | 54 | 10 | 295 |
| OS (unter Last) | 492-556 | 99-204 | 20-47 | 672-714 |
| MircoBlaze 100 Mhz | 5205 | 1432 | 291 | 6928 |
| AXI-Stream 100 Mhz | 2,6 | 2,5 | 2,6 | 10,6 |
| | Laufzeiten bei Auflösung 800 x 600 | | | |
| Variante | Median (ms) | Gauß (ms) | Inverter (ms) | Gesamt (ms) |
| ARM Bare-Metal (DDR3) | 1015 | 307 | 59 | 1382 |
| ARM Bare-Metal (BRAM) | - | - | - | - |
| OS (ohne Last) | 1180 | 296 | 58 | 1534 |
| OS (unter Last) | 2428-2893 | 589-642 | 118-204 | 3148-3741 |
| MircoBlaze 100 Mhz | - | - | - | - |
| AXI-Stream 100 Mhz | 9,0 | 9,0 | 9,3 | 27,5 |

Tabelle 3: Hardwareverbrauch der Varianten.

| Variante | Look-Up Tables (%) | Block-RAM (%) | Flip Flops (%) |
|---|---|---|---|
| ARM Bare-Metal (BRAM) | 0,39 | 22,86 | 0,88 |
| OS | 10 | 1 | 4 |
| MicroBlaze | 7,08 | 68,57 | 7,10 |
| AXI-Stream | 9,13 | 2,86 | 5,45 |

werden die Werte über eine *for*-Schleife addiert. Diese Schleife wird mit der Unroll-Direktive parallelisiert, was eine Latenz von nur zwei Taktzyklen ermöglicht. Es wird ebenfalls eine geschätzte maximale Taktrate von 199,2 MHz erreicht. Die Schätzung zeigt, dass die maximale Taktrate nicht von der Filteroperation, sondern von der „umliegenden“ Logik des Sliding Window bestimmt wird. Durch ihre einfache Funktionsweise benötigt die Grauwerttransformation nur einen Taktzyklus, ohne dass mit Direktiven gearbeitet werden muss. Hier kann eine theoretische Taktrate von 515,5 MHz erreicht werden.

## V. Adaptierbare Hardware/Software-Analyseumgebung

Um ein flexibles Testen unterschiedlicher Partitionierungen zu ermöglichen, wurden die Varianten ARM Bare-Metal, MicroBlaze und AXI-Stream in einem Hardware/Software Co-Design untergebracht. Auf einem ARM-Kern wird dabei eine Anwendung für die Koordination und Kopplung der Realisierungsvarianten ausgeführt. Diese Anwendung realisiert einen Server für das Empfangen, Verarbeiten und Senden der Bilder. Der Server ist über Ethernet mit einem Host-Rechner verbunden, der Instruktionen und Bilder an die Anwendung schickt. Beide Parteien kommunizieren mit dem TCP-Protokoll, dass auf dem ZedBoard mit Hilfe der lwip-Bibliothek implementiert wird.

Wenn der Host-Rechner ein Bild verarbeiten möchte, so sendet dieser zuerst einen Request, in dem die Dimensionen des Bildes und die Varianten der Filter definiert sind. Danach wird das eigentliche Bild übertragen. Da Bilder meistens in komprimierten Formaten wie JPEG vorliegen, müssen die Bilder auf dem Host-Rechner vor der Übertragung in eine Pixelmatrix konvertiert werden. Das empfangene Bild wird entsprechend der Angaben im Request verarbeitet und die Laufzeiten der einzelnen Filter werden gemessen. Die Reihenfolge der Filter ist folgendermaßen festgelegt worden: Median Filter – Gauß’scher Filter – Grauwerttransformation. Die Laufzeitmessung wird mit dem ARM Performance Monitor Registers Set durchgeführt. Daraus wird über den Takt des ARM die Laufzeit in Millisekunden errechnet. Nach der Verarbeitung wird das Bild an den Host-Rechner gesendet. Die Ausgabe der Laufzeitmessung erfolgt über ein serielles Terminal, das mit UART an den Host-Rechner angeschlossen ist.

## VI. Analyse und vergleich der Varianten

Nach ihrer Implementierung wurden die Varianten hinsichtlich der Kriterien Laufzeit und Ressourcenverbrauch untersucht. Die Laufzeit, also die Zeit, die zur Bearbeitung eines Bildes benötigt wird, wird in Millisekunden gemessen. Die Laufzeiten, die im Folgenden für die Varianten ARM Bare-Metal, OS und MicroBlaze angegeben werden, zeigen Schwankungen im Bereich von ca. 2 ms. Bei der Variante AXI-Stream liegen diese im Bereich von 0,2 ms. Dies ist auf das Cache-Verhalten des jeweiligen Prozessors und den Scheduler des Betriebssystems zurückzuführen. Die Laufzeiten aller Varianten sind in Tabelle 2 festgehalten. Die Laufzeitmessungen wurden mit der in Kapitel V vorgestellten Analyseumgebung durchgeführt.

Der Hardwareverbrauch der verfügbaren Ressourcen auf der PL für jede Variante wurde ebenfalls analysiert. Dazu zählt nicht nur die Variante an sich, zum Beispiel der MicroBlaze alleine, sondern auch alles, was zusätzlich auf die PL muss, damit die Variante funktioniert, wie zum Beispiel AXI-Interconnects oder AXI-DMAs. Der Verbrauch wird in Prozent der jeweiligen Ressource angegeben. Die in diesem Abschnitt verwendeten Daten wurden nach der Synthese des Block-Designs, jedoch vor der Optimierung bei der Implementierung, von Vivado ermittelt. Es werden keine optimierten Daten verwendet, da diese sich je nach Optimierungsstrategie für dieselbe Schaltung unterscheiden können. Die Messergebnisse aller Varianten sind in Tabelle 3 festgehalten.

Tabelle 2 zeigt alle Laufzeiten für die Verarbeitung eines Bildes aller Varianten im direkten Vergleich. Besonders die Variante AXI-Stream hebt sich deutlich von den anderen Varianten ab, da die Gesamtlaufzeit für die Verarbeitung eines Bildes bei der Auflösung 300 x 300 um den Faktor 26,9 geringer ist als die der nächstschnellsten Variante. Für die Auflösung 800 x 600 wächst der Unterschied auf den Faktor 50,25 an. Die starke Differenz in der Laufzeit kann auf den geringeren Speichertakt des Block-RAMs (100 MHz Block-RAM vs. 533 MHz DDR) sowie dessen örtliche Verteilung auf der PL zurückgeführt werden. Wird der Block-RAM als Speicher verwendet, sind die Daten über die gesamte PL in unterschiedlichen Blöcken verteilt, während diese beim DDR3-Speicher in einem zusammenhängenden Block gespeichert sind. Einer der Gründe für die Effizienz der AXI-Stream-Variante liegt an den vergleichsweise wenigen Lese-Schreibzyklen auf den DDR3-Speicher. Durch das Design mit der AXI-DMA wird im Arbeitsspeicher (DDR3) bei jedem Pixel nur einmal gelesen und geschrieben. Die sich wiederholenden Zugriffe in der IP erfolgen auf lokale Speicher der IP. Bei einer Softwareimplementierung muss bei einem lokalen Operator jedes Pixel 9-mal aus dem Arbeitsspeicher gelesen werden, was einen starken Lese-Overhead erzeugt. Neben dem Speicherzugriff ist auch die schnellere algorithmische Verarbeitung der Bilder ein Faktor für die höhere Effizienz. Die Variante AXI-Stream benötigt durch die Parallelisierung der Operationen für die Berechnung des neuen Pixelwertes beim Gauß'schen Filter zwei Taktzyklen, wohingegen die Softwareimplementierungen neun sequentielle Additionen und eine Division durchführen müssen.

Bei den Varianten ARM Bare-Metal (DDR3) und OS (ohne Last) liegen die Gesamtlaufzeiten mit 285 ms und 296 ms bei der Auflösung 300 x 300 sehr nahe zusammen. Dies gilt auch für die Auflösung 800 x 600. Dies liegt an der Verwendung derselben Hardware, der ARM Cortex A9 CPU und dem DDR3-RAM des Boards. Die geringe Differenz der Gesamtlaufzeit zwischen OS (ohne Last) und ARM Bare-Metal (DDR3) zeigt, dass das Scheduling und damit der Overhead des OS einen sehr geringen Einfluss auf die Gesamtlaufzeit hat, so lange keine Last auf dem System ist. Bei längerer Laufzeit macht sich der Overhead des Betriebssystems deutlicher bemerkbar, wie man an der größeren Laufzeitdifferenz erkennt.

Der MicroBlaze zeigt mit 6928 ms die größte Gesamtlaufzeit aller Varianten. Für das Bild wird derselbe Speicher wie bei der Variante ARM Bare-Metal (BRAM) verwendet. Diese schafft die Verarbeitung des Bildes jedoch in 643 ms, also um den Faktor 10,77 schneller. Eine Ursache dafür ist der unterschiedliche Takt von 667 MHz bei der Variante ARM Bare-Metal gegen 100 MHz bei der Variante MicroBlaze. Dadurch kann man erkennen, dass die Ursache des Laufzeitunterschiedes nicht allein der Prozessortakt sein kann, sondern dass es auch in der Effizienz der Instruktionen Unterschiede geben muss.

Die Skalierung bei unterschiedlichen Aufgabengröße verläuft nicht bei allen Varianten gleich. Die Varianten ARM Bare-Metal (DDR3) und OS (ohne Last) verzeichnen einen Laufzeitanstieg um den Faktor 4,64 bzw. 5,20 bei einem Anstieg der Datenmenge um den Faktor 5,33. Beim AXI-Stream dagegen ist beim gleichen Anstieg der Pixelmenge nur eine Verlängerung der Gesamtlaufzeit um dem Faktor 2,59 festzustellen. Der Unterschied zwischen Software und AXI-Stream ist durch die Arbeitsweise des Treibers der AXI-Stream-Variante zu erklären. Dieser verarbeitet die Bilder zeilenweise, und bei einem AXI-DMA Transfer wird immer eine Zeile bearbeitet. Da die Zeilenlänge stärker gestiegen ist als die Zeilenanzahl, wird bei jeder Transaktion ein größerer Anteil des Bildes bearbeitet.

## VII. Beispiel eines hardware/software co-designs

Nach der Analyse der einzelnen Varianten soll anhand dieses Beispiels eine mögliche Partitionierung demonstriert werden. Bei der Aufteilung der Filter auf die Varianten stützte man sich auf die Analyse in Tabelle 2 und ordnete den Filter mit der größten Laufzeit (Median) der Variante zu, die diesen am schnells-

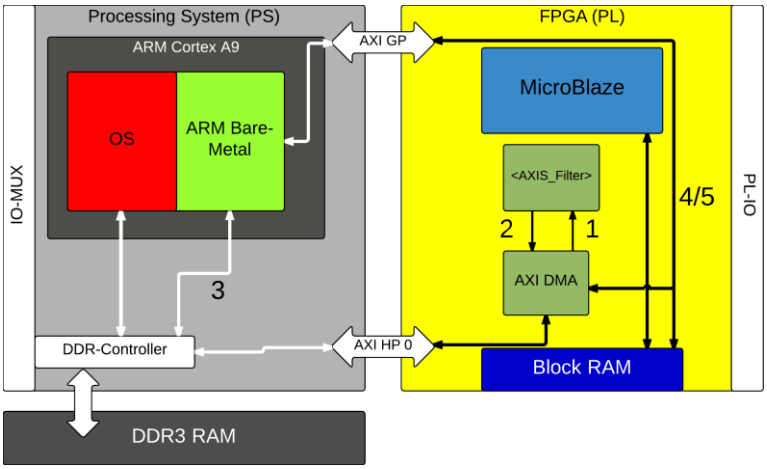



Abbildung 5: Ablauf der Partitionierung.

ten ausführen kann (AXI-Stream). Analog dazu wurde der Filter mit der kürzesten Laufzeit (Inverter) der langsamsten Variante zugeordnet (MicroBlaze). Der letzte Filter (Gauß) wurde der Variante ARM Bare-Metal zugeordnet. Mit dieser Partitionierung werden nicht nur alle in dieser Arbeit analysierten Komponenten für die Verarbeitung des Bildes verwendet, sondern auch beide Arbeitsspeicher. Abbildung 5 illustriert durch die Nummerierung, in welchem Teilsystem des SoCs die im Folgenden erläuterten Teilschritte der Verarbeitung ablaufen. Nach der Erläuterung der partitionierten Bildverarbeitung erfolgt eine Laufzeitanalyse der Teilschritte.

Nach dem Start der Anwendung wartet der ARM auf eine Nachricht des Host-Rechners, in dem ein neues Bild angekündigt wird. Die Daten, die vom Host-Rechner übertragen werden, werden als erstes im DDR3-RAM abgelegt. Bei der Verarbeitung ist der erste Schritt der Median Filter, der in der Variante AXI-Stream ausgeführt wird. Dazu wird das Bild zeilenweise vom Treiber über die DMA in den Filter gestreamt (1). Die Steuerung der DMA wird durch Software auf einem der ARM-Kerne realisiert. Die zu verarbeitenden Daten werden von der DMA zurück in den DDR3-RAM (2) geschrieben. Die Software wartet nach jeder Zeile, bis die DMA die Daten vollständig zurück geschrieben hat, bevor die nächste Zeile verarbeitet wird. Auf diesen Schritt folgt die Variante ARM Bare-Metal, die den Gauß'schen Filter anwendet. Hier erfolgt der Zugriff der Software direkt auf den DDR3-RAM (3). Bevor der MicroBlaze die Grauwerttransformation durchführen kann, muss das Bild über den AXI-GP Port in den Block-RAM verschoben (4) werden. Dabei wird als erstes das Bild über den AXI GP-Port, gesteuert von einem ARM-Kern, in den BRAM transferiert. Bei diesem Datentransfer wird keine DMA verwendet.

Nachdem die Übertragung abgeschlossen ist, setzt der ARM-Kern die Flag, um den MicroBlaze anzuweisen, die Grauwerttransformation durchzuführen. Wenn der MicroBlaze durch Zurücksetzen der Flag signalisiert, dass die Verarbeitung abgeschlossen ist, kann das Bild zurück in das DDR3-RAM transferiert werden (5). Die Verarbeitung ist dann abgeschlossen und das Bild wird wieder zurück an den Host-Rechner gesendet.

Tabelle 4: Laufzeitanalyse des Partitionierungsbeispiels bei einem Bild mit der Auflösung 300 x 300.

| Abschnitt | Laufzeit (ms) |
|---|---|
| AXI-Stream (Median) | 2,1 |
| ARM Bare-Metal (Gauss) | 58,2 |
| Bild von DDR3 nach BRAM | 15,1 |
| MicroBlaze (Inverter) | 285,2 |
| Bild von BRAM nach DDR3 | 14,8 |
| Gesamt | 375,4 |

Die Analyse der Partitionierung wird in Tabelle 4 veranschaulicht. Sie zeigt an, welcher Abschnitt der Verarbeitung wie viel Zeit in Anspruch genommen hat. Diese Analyse zeigt, dass 76 % der Laufzeit auf den Inverter auf dem MicroBlaze entfallen, obwohl der Inverter bei allen Varianten der Filter mit der schnellsten Laufzeit ist. Der Datentransfer zwischen BRAM und DDR3-Arbeitsspeicher hat einen Anteil von 7,9 % an der Gesamtlaufzeit. Ob die Übertragungszeit einen großen Anteil an der Laufzeit hat, hängt von der verwendeten Variante und dem verwendeten Filter ab. In diesem Fall ist der Anteil gering, da der Datentransfer im Kontext des Inverters auf dem MicroBlaze ausgeführt wird. Vergleicht man die Laufzeit der Partitionierung mit der Gesamtlaufzeit der einzelnen Varianten, so lässt sich nur bei den Varianten ARM Bare-Metal (BRAM) und MicroBlaze eine Verbesserung der Laufzeit feststellen. Diese Verbesserung ist auf die, im Vergleich zu den anderen Varianten geringe Laufzeit der Variante AXI-Stream zurückzuführen. In diesem speziellen Fall ist eine

Partitionierung aus reiner Performance-Sicht nicht praktikabel, da die Gesamtlaufzeit durch den langsamen MicroBlaze und den Datentransfer zwischen den Arbeitsspeichern verlängert wird.

## VIII. Fazit

Die Analyse zeigt, wie stark sich die Laufzeit von Hardware und Software unterscheiden. Zusätzlich zeigt sich, dass die Hardware eine bessere Skalierung bei steigender Auflösung ermöglicht. Durch das AXI4-Protokoll und vorgefertigte IPs ist die Realisierung einer Kommunikation von Hardware und Software ohne großen Aufwand möglich. Die modernen, für diese Arbeit verwendeten Softwarewerkzeuge verringern durch ihren hohen Grad an Automatisierung die Einstiegshürde und die Entwicklungszeit. Auch wenn ARM-Prozessoren sich in den letzten Jahren stark verbreitet haben, so stellen selbst einfache Bildverarbeitungsalgorithmen eine große Herausforderung für diese dar. Auch mit einer Parallelisierung der Anwendung auf beiden ARM-Kernen könnte die Leistungsfähigkeit der AXI-Stream-Variante nicht erreicht werden.

Die exemplarische Partitionierung des Evaluierungsbeispiels macht in ihrer gewählten Form keinen Sinn, da die Laufzeit im Vergleich zu fast allen Einzel-Varianten verlängert wird. Im Hinblick auf diese Arbeit stand jedoch nicht die Verbesserung der Leistungsfähigkeit im Vordergrund, sondern eine Untersuchung von Möglichkeiten zur Partitionierung einer Anwendung. Bei einer Weiterführung der Arbeit wäre das Implementieren einer Möglichkeit für die Bildakquisition ein sinnvoller Schritt, sodass sich das Evaluierungsbeispiel selbst mit Daten versorgen kann. Es ist anzunehmen, dass auch hier Unterschiede zwischen möglichen Varianten deutlich werden. Sollten weitere Varianten getestet werden, so wäre der Vergleich der AXI-Stream-Filter mit einer dedizierten GPU interessant, da hier massiv parallelisiert werden kann. Auch das Implementieren weiterer Filter kann interessante Resultate bringen.

## Literaturverzeichnis


[1] Xilinx Inc., “MicroBlaze Soft Processor Core.” http://www.xilinx.com/tools/microblaze.htm. [Online; Access Jan. 2015]

[2] Xilinx Inc., “Zynq-7000 All Proframmable SoC Overview.” http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf , Oct. 2014. [Online; Access Jan. 2015].

[3] AVNET, “ZedBoard Hardware User’s Guide.” http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf , Jan. 2014. [Online; Access Dec. 2014].

[4] Xilinx Inc., “UG-984: MicroBlaze Reference Guide.” http://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_2/ug984-vivado-microblaze-ref.pdf, Apr. 2014. [Online; Access Jan. 2015].

[5] Xilinx, “UG 761: AXI Reference Guide.” http://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf, Mar.2011. [Online; Access Dec. 2014].

[6] Wikipedia, “Negativtransformation.” http://de.wikipedia.org/wiki/Punktoperator_%28Bildverarbeitung%29#Negativtransformation . [Online; Access Feb. 2015].

[7] A. Nischwitz, M. Fischer, P. Haberäcker, G. Socher, *Computergrafik und Bildverarbeitung* Band II: Bildverarbeitung. Vieweg + Teubner, 2011.

[8] D. Bagni, “Median Filter and Sorting Network for Video Processing with Vivado HLS.” http://ens.ewi.tudelft.nl/Education/courses/et4351/Median. pdf, 2014. [Online; Access Feb. 2015].

[9] Open Source, “libjpeg.” http://sourceforge.net/projects/libjpeg/, July 2012. [Online; Access Feb. 2015].

[10] Bourke, Paul, “JPEG Libraries.” http://paulbourke.net/libraries/, Feb.2011. [Online; Access Feb. 2015].

[11] Xilinx, “LogiCORE IP AXI DMA v7.1.” http://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf, Apr.2014. [Online; Access Feb. 2015].

[12] Xilinx, F. M. Vallina, “Implementing Memory Structures for Video Processing in the Vivado HLS Tool.” http://www.xilinx.com/support/documentation/application_notes/xapp793-memory-structures-video-vivado-hls.pdf, Sept. 2012. [Online; Access Feb. 2015].


Dennis Sebastian Rieber studierte Technische Informatik an der Hochschule Albstadt-Sigmaringen und schloss sein Studium 2015 mit dem akademischen Grad eines Bachelor of Engineering ab. Seit dem Wintersemester 2015 absolviert er ein Master-Studium an der Universität Heidelberg im Fach Technische Informatik.

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

Hochschule Ulm

# High-Level-Synthese eines OFDM-Funkkommunikationssystems für eine auf den Einsatz in der Lehre ausgelegte Software Defined Radio-Plattform

Steffen Moll, Marius Welk, Matthias Düll, Roland Münzner

***Zusammenfassung*—Vorgestellt wird die Entwicklung eines Funkkommunikationssystems auf Basis des Mehrträgerübertragungsverfahrens Orthogonal Frequency Division Multiplexing (OFDM) für eine an der Hochschule Ulm entwickelte Software Defined Radio (SDR) Plattform. Das System arbeitet mit 16 Unterträgern und verfügt über die Modulationsarten 4-QAM und 16-QAM. Für die Träger-, Symboltakt- und Sampletakt-Rückgewinnung sowie für die Kanalschätzung kommen Präambelgestützte Verfahren zum Einsatz. Neben der Entwicklung der digitalen Signalverarbeitungsalgorithmen für die Synchronisation stellt eine möglichst ressourcensparende Implementierung des Gesamtsystems eine entscheidende Herausforderung dar, um das System auch mit der Methode der High-Level-Synthese für FPGAs mit relativ geringer Performance zugänglich zu halten.**

***Schlüsselwörter*—Software Defined Radio (SDR), Rapid Prototyping, Orthogonal Frequency Division Multiplexing (OFDM), Synchronisation, High-Level-Synthese, Field Programmable Gate Array (FPGA), ressourcensparende Implementierung, Laborsystem für die Lehre.**

## I. Einleitung

Die Methode des so genannten Rapid Prototyping, d.h. des möglichst schnellen und direkten Umsetzens von Systemkonzepten und deren Modellierung in einen ersten Prototypen, spielt in der Entwicklung von Kommunikationssystemen eine stetig steigende Rolle. Dies gilt insbesondere für so genannte Software Defined Radio (SDR) Systeme, welche zumindest für die Signalverarbeitung im Basisband eine auf Software basierende und damit sehr flexible Lösung bieten.

Um einerseits die Methode des Rapid Prototyping direkt in der Lehre einsetzen zu können und andererseits den Studierenden in Laborübungen einen möglichst umfassenden Zugang zur Analyse digitaler und analoger Funkkommunikationssysteme – von der digitalen Signalverarbeitung im Basisband bis hin zur RF-Performance-Analyse – zu ermöglichen, wurde an der Hochschule Ulm in mehreren Studien- und Abschlussarbeiten eine Hardwareplattform [1] sowie ein auf dem Prinzip der High-Level-Synthese mittels des Matlab/Simulink-HDL-Coders basierender Workflow [2] für das Rapid Prototyping von SDR-Systemen entwickelt. Bisher wurden auf der SDR-Plattform analoge Funkübertragungssysteme auf Basis von Amplituden- und Frequenzmodulation sowie ein Einträger-BPSK (Binary Phase Shift Keying) System als digitales Funkübertragungssystem realisiert. Neben direkt im Basisband des Senders generierten Ton-Signalen erlaubt das System die Übertragung beliebiger Audiosignale, so dass die Auswirkung von im System auftretenden Verzerrungen auch akustisch durch die Studierenden beurteilt werden können.

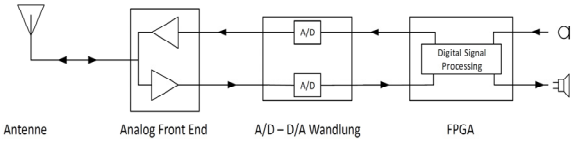



Abbildung 1: Blockschaltbild der eingesetzten Hardware-Plattform.

Rund 30 Jahre, nachdem OFDM erstmals für Anwendungen in der digitalen Funkübertagung vorgeschlagen wurde [3], stellt OFDM eines der dominierenden Übertragungsverfahren für breitbandige digitale Funksysteme dar und kommt z. B. in DAB, DVB, WiMAX, LTE oder Wireless LAN-Systemen entsprechend IEEE 802.11 [4] zum Einsatz.

Die Zielsetzung des hier vorgestellten Projekts ist einerseits die Realisierung eines OFDM-Funkübertragungssystems für die Lehre auf Basis der bestehenden SDR-Plattform, um im Rahmen studentischer Laborübungen die grundlegenden Eigenschaften von OFDM-Systemen an einem einfachen, zugleich aber möglichst weitgehend konfigurierbaren System verdeutlichen zu können. Andererseits wird auch der Einsatz von schmalbandigen OFDM-Systemen mit wenigen hundert kHz Bandbreite im Rahmen von eingebetteten Systemen immer wieder diskutiert, insbesondere für Anwendungen im Bereich der Power Line Communication (PLC), z.B. für Smart Grid Anwendungen [5]. Daher ist die Untersuchung einer möglichst ressourcensparenden digitalen Implementierung eines OFDM-Systems auch von generellem Interesse.

Steffen Moll, Marius Welk, Matthias Düll, Roland Münzner, muenzner@hs-ulm.de, Hochschule Ulm, Eberhard-Finckh-Straße 11, 89075 Ulm.

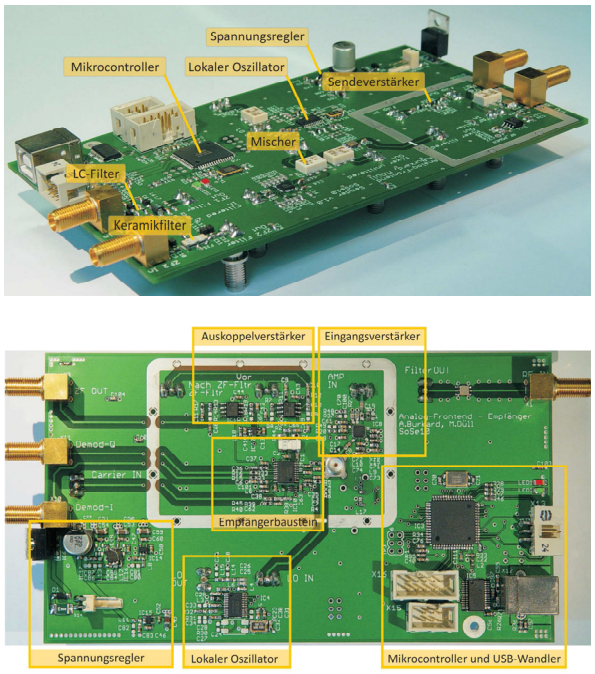



Abbildung 2: Hardware-Realisierung der Analog-Front-Ends für Sender (oben) und Empfänger (unten).

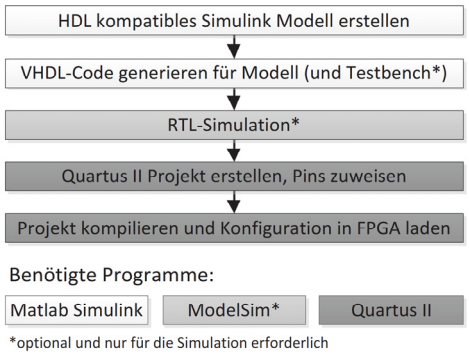



Abbildung 3: Workflow für die High-Level-Synthese aus Matlab/ Simulink heraus [2].

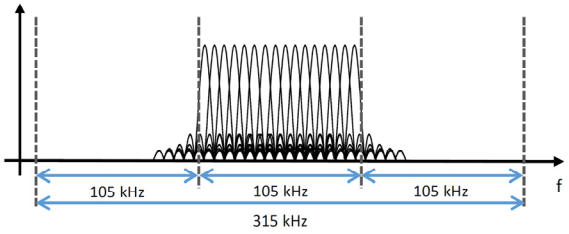



Abbildung 4: Spektrum des OFDM-Übertragungssystems mit 16 Unterträgern.

## II. SDR Hardwareplattform und Workflow für die High Level Synthese

Abbildung 1 zeigt das Blockschaltbild der eingesetzten Hardware-Plattform. Die verwendeten Analog-Front-Ends sind in Abbildung 2 dargestellt. Die digitale Signalverarbeitung erfolgt auf DE2-Boards mit Altera Cyclone® II FPGAs (50 MHz Systemtakt, 35 000 Logikelemente, 70 komplexe Multiplizierer). Die D/A- und A/D-Wandlung erfolgt bei einer Zwischenfrequenz von $f_{ZF}$ = 10,7 MHz auf an der Hochschule Ulm entwickelten Boards ($f_S$ = 50 MHz, D/A-Wandlung mit 8 Bit, A/D-Wandlung mit 10 Bit) [1]. Ebenfalls an Hochschule Ulm entwickelte Analog-Front-Ends [1] für die Übertragung im ISM-Band bei 433 MHz vervollständigen das Funkübertragungssystem. Die Analog-Front-Ends arbeiten mit einer Zwischenfrequenzbandbreite von $B_{ZF}$ = 330 kHz, die gleichzeitig auch der maximalen Kanalbandbreite entspricht und verfügen über eine Vielzahl von Messpunkten zu Analysezwecken im Signalpfad.

Der für die High-Level-Synthese, in der mittels des Mathworks® HDL-Coders zunächst direkt aus Matlab/Simulink heraus synthetisierbarer VHDL-Code für die Implementierung im FPGA erzeugt wird, verwendete Workflow [2] ist in Abbildung 3 dargestellt. Im Rahmen der Laborübungen wird den Studierenden ein vorgefertigtes Quartus II-Rahmenprojekt zu Verfügung gestellt, das insbesondere die Zuweisung der benötigen Pins und die VHDL-Blöcke für die Anbindung der Audioschnittstelle bereits beinhaltet. Dies erlaubt den Fokus der Laborübungen auf das Design der digitalen Signalverarbeitung und die damit erzielte Performance, auf den Ressourcenbedarf der digitalen Implementierung sowie auf die Hardware-Performance des OFDM-Gesamtsystems zu legen.

## III. Orthogonal Frequency Division Multiplexing (OFDM)

Die OFDM-Übertragung [6] erfolgt in paralleler Weise auf einer festgelegten Anzahl von so genannten Unterträgern mit Trägerfrequenzen $f_n$, wobei sich die Trägerfrequenzen alle als Vielfache des Unterträgerabstandes $\Delta f$ darstellen lassen, was die Orthogonalität der Unterträger sicherstellt. Bei gegebener Übertragungsbandbreite $B$ und festgelegter Anzahl der Unterträger $N$ ergibt sich ein Unterträgerabstand von

$$\Delta f = \frac{B}{N}. \qquad (1)$$

Wird für die Unterträger ein Pulse Shaping mittels einfacher Rechteck-Pulse der Dauer

$$T_S = \frac{1}{\Delta f} \qquad (2)$$

verwendet, wobei $T_S$ die Dauer eines OFDM-Symbols darstellt, so ergibt sich das typische Spektrum eines OFDM-Mehrträgerübertragungssystems (siehe auch Abbildung 4) mit sinc-förmigem Verlauf der Spektren der einzelnen Unterträger und der Orthogonalitätsbedingung, dass für die Frequenz jedes Unterträgers die

Tabelle 1: Randbedingungen für die Systemauslegung.

| Audio-Interface und Datenrate | |
|---|---|
| Abtastrate | 8 kHz |
| Quantisierung | 16 Bit |
| Coderate FEC | 1/2 |
| → Erforderliche Datenrate | 256 kBit/s |
| Übertragungskanäle | |
| Anzahl Kanäle | 3 |
| Bandbreite pro Kanal | 105 kHz |
| Kanaleigenschaften | |
| Max. Dopplershift (35 m/s) | 50 Hz |
| Kohärenzzeit | 20 ms |
| Delay Spread | 2 µs |

Spektren aller anderen Unterträger eine Nullstelle aufweisen [7].

In der Realisierung von zeit-diskreten OFDM-Systemen lässt sich das diskrete Zeitsignal $s(k \cdot t_s)$ zum Abtastzeitpunkt $k \cdot t_s$ mittels der diskreten Fourier-Transformation darstellen und schließlich durch die IFFT berechnen [6].

$$s(k \cdot t_s) = \frac{1}{\sqrt{T_S}} \sum_{n=0}^{N-1} c_n \exp(j2\pi \cdot n \cdot \Delta f \cdot k \cdot t_s)$$

$$= \frac{1}{\sqrt{T_S}} \sum_{n=0}^{N-1} c_n \exp\left(j2\pi \cdot n \cdot \frac{B}{N} \cdot k \cdot \frac{1}{B}\right) \qquad (3)$$

Hierbei beachten wir, dass der Abstand $t_s$ zwischen zwei Abtastzeitpunkten durch den Kehrwert der Übertragungsbandbreite $B$ gegeben ist. Die Koeffizienten $c_n$ bezeichnen die komplexen Symbole des Modulationsverfahrens (im Folgenden 16-QAM).

Durch die Möglichkeit der zyklischen Verlängerung des OFDM-Symbols, z.B. durch einen Cyclic Prefix, weisen OFDM-Systeme eine geringe Empfindlichkeit gegenüber Inter-Symbol-Interferenz aufgrund von Mehrwegeausbreitung auf. Sie sind jedoch, insbesondere bei geringem Unterträgerabstand, sehr empfindlich gegenüber Inter-Kanal-Interferenz, wobei z. B. durch einen zu großen Dopplershift oder eine ungenaue Trägerrückgewinnung die Orthogonalität der Unterträger verloren gehen kann [6], [7].

## IV. Systemauslegung

Die Auslegung des OFDM-Systems erfolgt für die gleichzeitige Übertragung von 3 OFDM-Kanälen innerhalb der ZF-Bandbreite des Systems von 330 kHz (siehe Abbildung 4) und für die in Tabelle 1 dargestellten Randbedingungen. Hieraus ergibt sich das in Tabelle 2 dargestellte Design des OFDM-Systems.

Tabelle 2: OFDM System Design.

| | |
|---|---|
| Unterträger | 16 |
| Guard Träger | 3 (+ DC) |
| Cyclic Prefix | 1/16 |
| Modulation | 16-QAM |
| Symboldauer $T_S$ | 153,6 µs |
| Datenrate | 277 kBit/s |
| Synchronisation | Präambel, keine Pilottöne (in den Datensymbolen) |
| Rahmenstruktur | Präambel + Header + 34 Datensymbole |

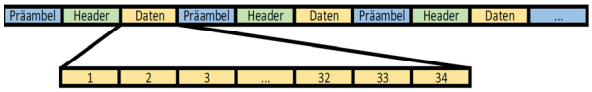



Abbildung 5: Rahmenstruktur das OFDM-Systems.

## V. Herausforderungen für die Implementierung im FPGA

Das Blockschaltbild der im FPGA für Sender und Empfänger des OFDM-Systems realisierten digitalen Signalverarbeitung ist in Abbildung 6 dargestellt, wobei die Punkte der Rahmenerzeugung und Synchronisation, die jeweils eine besondere Herausforderung für das Systemdesign bedeuten, besonders hervorgehoben sind.

Kanalschätzung und Entzerrung stellen unter den für das System vorausgesetzten Kanalbedingungen dagegen keine allzu großen Herausforderungen dar. Die Kanalschätzung kann aufgrund der langen Kohärenzzeit, die sich über ungefähr die vierfache Rahmendauer erstreckt, und der über der Kanalbandbreite von 105 kHz liegenden Kohärenzbandbreite durch Mittelung über die Pilottöne aufeinanderfolgender Präambeln im Frequenz- und im Zeitbereich erfolgen. Die Zeitdauer für die Mittelung ist hierbei ein einstellbarer Systemparameter. Für die Entzerrung wird ein so genannter One-Tap Zero-Forcing Equalizer [7] eingesetzt.

### *A. Einfügen von Präambel und Header-Symbol*

Für die Zwecke der Kanalschätzung und Synchronisation wird zu Beginn eines jeden OFDM-Rahmens ein Präambel-Symbol eingefügt. In den OFDM-Datensymbolen werden keine zusätzlichen, gesonderten Pilottöne verwendet. Um die Datenrate des OFDM-Systems (277,76 kbit/s bei 16-QAM) an die Datenrate des Audio-Interface anzupassen, ist ein zusätzliches Symbol für den Rahmen-Header vorzusehen, da nicht in jedem Rahmen alle OFDM-Symbole

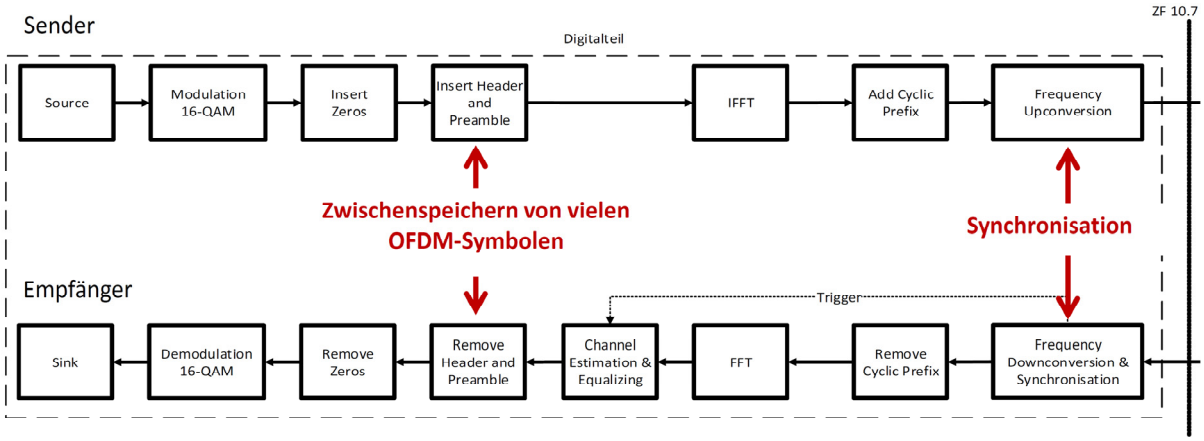



Abbildung 6: Blockschaltbild der im FPGA für Sender und Empfänger des OFDM-Systems realisierten digitalen Signalverarbeitung unter Hervorhebung der Punkte, die eine besondere Herausforderung für das Systemdesign darstellen.

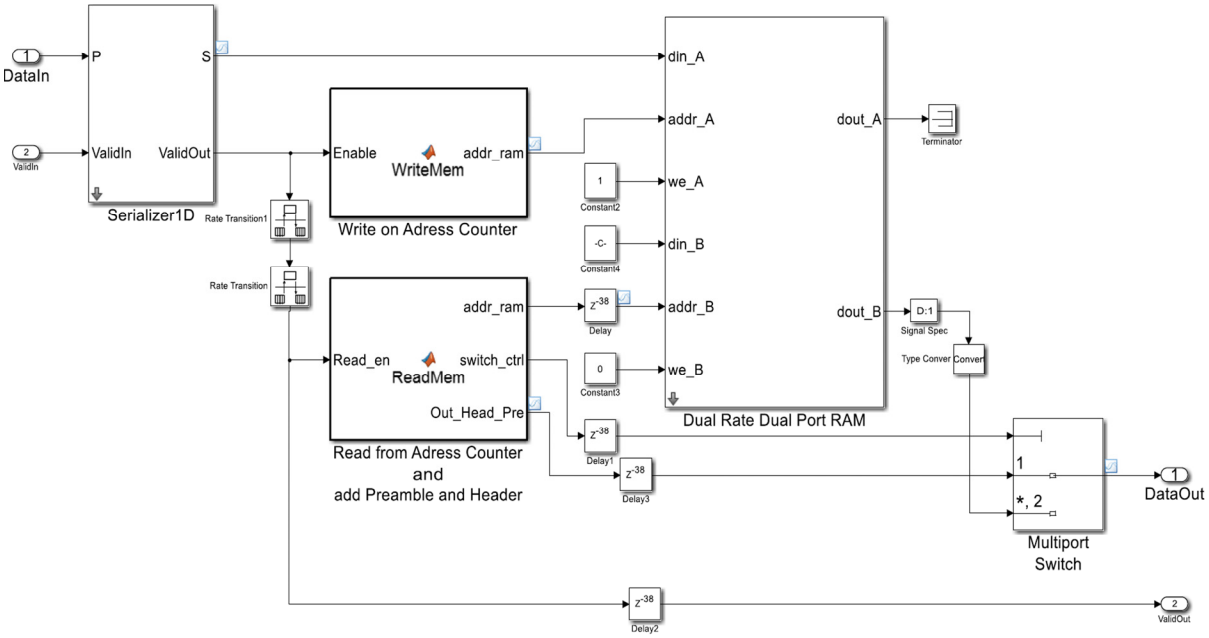



Abbildung 7: Ressourcensparende Realisierung des Einfügens der OFDM-Symbole für Präambel und Header in den Strom der OFDM-Datensymbole unter Verwendung eines Dual Port RAM mit unterschiedlichen Zugriffszyklen für Lesen und Schreiben.

mit Daten gefüllt werden, bzw. nicht alle Rahmen mit derselben Modulationsart gesendet werden. Die sich ergebende Rahmenstruktur ist in Abbildung 5 schematisch dargestellt.

Das Einfügen von Präambel und Header stellt eine Herausforderung für eine ressourcenschonende Implementierung im FPGA dar, da für das Einfügen Daten zwischengespeichert werden müssen. Eine parallele Implementierung führt neben einer Verzögerung von bis zu einer Rahmenlänge zu einem hohen Verbrauch an Logikelementen im FPGA. Eine serielle Verarbeitung der einzelnen OFDM-Symbole erfordert dagegen ein temporäres Zwischenspeichern einzelner Symbole und den Zugriff auf diesen Zwischenspeicher mittels unterschiedlicher Raten. Hierfür eignet sich das im Rahmen des Matlab/Simulink-HDL-Coders zur Verfügung gestellte Dual Port RAM als Schnittstelle zu externen Speicherbausteinen.

Der Ressourcenverbrauch im FPGA wird durch den Einsatz des externen Speichers stark reduziert und beschränkt sich auf die für die Speicheradressierung notwendigen Memory Bits. Darüber hinaus vermeidet diese Implementierung aufgrund der seriellen Verarbeitung der einzelnen OFDM-Symbole zusätzliche Latenzzeiten und lässt sich mittels der RAM-Unterstützung durch den Matlab/Simulink-HDL-Coder einfach im Rahmen einer Modellierung des OFDM-Systems für die High-Level-Synthese umsetzen. Abbildung 7 zeigt die gewählte Umsetzung.

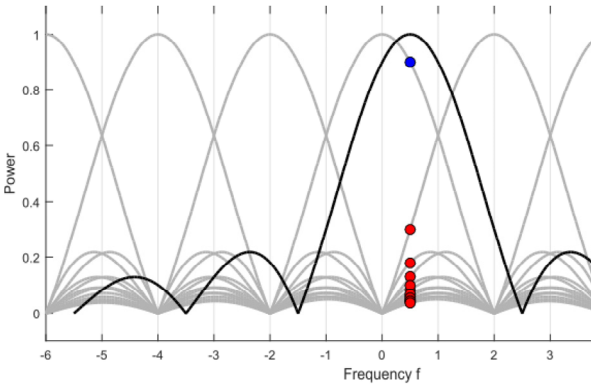



Abbildung 8: Interferenz zwischen den einzelnen Unterträgern bei nicht korrigiertem Träger-Offset in einem OFDM-System.

*B. Träger- und Taktrückgewinnung (Synchronisation)*

Die Rückgewinnung von Träger und Symboltakt kann unter der Voraussetzung, dass eine Implementierung im Basisband erfolgen soll, für OFDM-Systeme nicht unabhängig voneinander durchgeführt werden. Die für das hier vorgestellte OFDM-System gewählte Realisierung basiert auf der Wahl eines zeitlich periodischen Präambel-Symbols, das sich nach der Hälfte der OFDM-Symboldauer $T_S$ zyklisch wiederholt. So kann auf Basis dieses Präambel-Symbols sowohl der OFDM-Symboltakt, mittels einer Bildung der Korrelation über die Präambel, als auch der Träger-Offset, mittels Berechnung des sich über die zyklische Präambel einstellenden mittleren Phasenversatzes, zurückgewonnen werden [8].

Im Rahmen der vorliegenden Implementierung wurde auf den Einsatz einer Phase-Locked-Loop (PLL) für die Synchronisation verzichtet. Vielmehr wird, nachdem der OFDM-Systemtakt durch Korrelation über die Präambel in der Genauigkeit eines OFDM-Zeitsamples zurückgewonnen wurde, der Träger-Offset durch Entzerrung im Zeitbereich mittels des sich durch den Träger-Offset ergebenden zeitabhängigen Phasenversatzes korrigiert und anschließend der verbleibende Timing Offset durch Korrektur des sich durch den Timing-Offset ergebenden frequenzabhängigen Phasenversatzes im Rahmen der Entzerrung mittels des One-Tap Equalizers entfernt [8].

Diese Vorgehensweise erlaubt die Korrektur eines Träger-Offsets von maximal einem Unterträgerabstand $\Delta f = 6{,}5625$ kHz, was ungefähr 15 ppm der Trägerfrequenz von 433 MHz entspricht, sowie die Korrektur eines maximalen Timing-Offsets von einem OFDM-Zeitsample $t_s = 9{,}52$ µs. Für die gewählten Randbedingungen des Systems (Genauigkeit der Lokaloszillatoren auf den DE2-Boards, maximaler Dopplershift, maximale Umweglaufzeitdifferenz in der Mehrwegeausbreitung) sind diese Korrekturbereiche ausreichend und es kann auf den Einsatz einer PLL verzichtet werden.

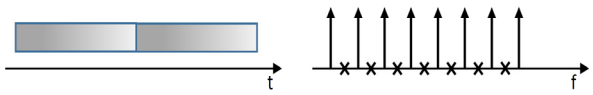



Abbildung 9: Realisierung eines zeitlich periodischen Präambel-Symbols durch Ausdünnung der genutzten Unterträger. Die Verwendung nur jedes zweiten Unterträgers führt zu einer Periodizität mit der halben OFDM-Symboldauer.

Die Entzerrung im Zeitbereich für die Kompensation des Träger-Offsets muss dabei mit ausreichend großer Wortbreite im FPGA erfolgen, um sicherzustellen, dass die sich aus dem Träger-Offset ergebende Interferenz zwischen den einzelnen Unterträgern (siehe Abbildung 8) in ausreichendem Maße unterdrückt wird und so die Orthogonalität der Unterträger erhalten bleibt.

In der konkreten Realisierung wurde das zeitlich periodische Präambel-Symbol im Frequenzbereich realisiert, indem, wie in Abbildung 9 dargestellt, nur jeder zweite Unterträger mit einem QAM-Symbol ungleich Null besetzt wurde. Für die Präambel wurde die Modulationsart BPSK gewählt. Zusätzlich wurde eine Präambel mit besonders geringer Peak-to-Average-Power-Ratio (PAPR) gewählt [9], so dass die Präambel mit höherer Amplitude gesendet werden kann, als die OFDM-Datensymbole, die eine deutlich größere PAPR aufweisen können.

Grundlage der Zeitsynchronisation und der Frequenzkorrektur ist die fortlaufende Berechnung der Korrelation des am Empfänger eintreffenden Stromes der Zeitsamples $s(k \cdot t_s)$ konsekutiver OFDM-Symbole mit einer um die Hälfte der Dauer $T_S$ eines OFDM-Symbols, und damit genau um die Periode der zyklischen Präambel versetzten Kopie seiner selbst. Die Berechnung der Korrelation erfolgt dabei ebenfalls über die Hälfte der Dauer $T_S$ eines OFDM-Symbols, d. h. über $N/2 = 8$ Zeitsamples [8].

$$\mathrm{corr}(l \cdot t_s) = \frac{2}{N} \sum_{k=l}^{l+\frac{N}{2}} s(k \cdot t_s) \cdot s^* \left( \left[ k + \frac{N}{2} \right] \cdot t_s \right) \qquad (4)$$

Da nur das Präambel-Symbol eine Periodizität mit der halben Dauer des OFDM-Symbols aufweist, kann der Beginn der Präambel durch das Maximum des Betrages der Korrelation in Gl. (4), d. h. über den Index $l_{\max}$ mit

$$\left|\mathrm{corr}(l_{\max} \cdot t_s)\right| = \max\left(\left|\mathrm{corr}(l \cdot t_s)\right|\right) \qquad (5)$$

bestimmt werden. Die Genauigkeit beträgt dabei ein OFDM-Zeitsample. Mit Hilfe der Phasenablage $\Delta\Phi$ über die Periodendauer der Präambel

$$\Delta\Phi = \arg\left(\mathrm{corr}(l_{\max} \cdot t_s)\right) \qquad (6)$$

lässt sich im Frequenzbereich ein Träger-Offset von bis zu einem Unterträgerabstand $\Delta f$ entsprechend Gl. (7) korrigieren [8].

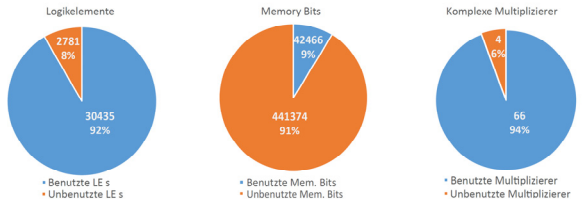



Abbildung 10: Ressourcenverbrauch für die Implementierung von Sender und Empfänger im FPGA.

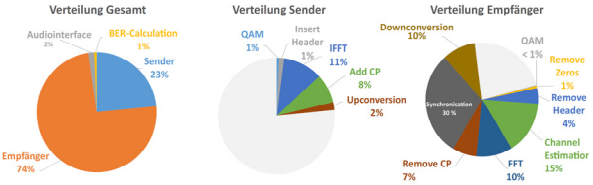



Abbildung 11: Ressourcenverbrauch der Logikelemente aufgeschlüsselt in die einzelnen Blöcke von Sender, Empfänger und Audio-Interface.

$$\tilde{s}(l \cdot t_s) = s(l \cdot t_s) \cdot \exp\left(-j \cdot l \cdot \frac{2}{N} \cdot \Delta\Phi\right) \qquad (7)$$

Aufgrund der sich über mehrere Rahmendauern erstreckenden Kohärenzzeit kann das Rauschen in der Bestimmung der Phasenablage $\Delta\Phi$ durch Mittelung über mehrere Präambeln verringert werden. Die Dauer der Mittelung kann dabei als Systemparameter variiert werden.

Da die Berechnung der Korrelation und der Frequenzkorrektur eine große Anzahl komplexer Multiplizierer erfordert, war letztendlich unter Berücksichtigung der deutlich eingeschränkten verfügbaren Ressourcen auf dem eingesetzten FPGA eine Implementierung nur aufgrund der geringen Anzahl von lediglich 16 Unterträgern im realisierten OFDM-System möglich.

## VI. Ergebnisse (Ressourcenbedarf im FPGA)

Der gemeinsame Ressourcenbedarf für Sender und Empfänger des OFDM-Systems ist in Abbildung 10, aufgeschlüsselt nach Logikelementen, Memory Bits und komplexen Multiplizierern dargestellt. Wie erwartet, werden die verfügbaren Ressourcen für Logikelemente und komplexe Multiplizierer nahezu vollständig ausgeschöpft.

Betrachtet man entsprechend Abbildung 11 nur den Ressourcenbedarf an Logikelementen, so entfallen 74 % der aufzuwendenden Ressourcen auf den Empfänger, wobei die Synchronisation mit 30 % des Ressourcenbedarfs den größten Anteil aufweist.

## VII. Zusammenfassung und Ausblick

In der Arbeit wurde gezeigt, dass die Implementierung eines OFDM-Übertragungssystems bei entsprechender Dimensionierung auch auf für Hardware-Plattformen mit beschränkten Ressourcen möglich ist. In der aktuellen Realisierung bleibt jedoch kein Spielraum, um auf der verwendeten Hardware-Plattform Erweiterungen der Signalverarbeitung, insbesondere den Einsatz einer Forward Error Correction (FEC), zu implementieren. In gewissem Umfang können weitere Logikelemente durch alternative Realisierungen mittels RAM-Nutzung für das Einfügen des Cyclic Prefix und der Guard Carrier eingespart werden. Durch eine sequentielle Implementierung der Korrelationsbildung für die Synchronisation können schließlich in begrenztem Umfang komplexe Multiplizierer eingespart werden.

Als nächster Schritt steht die eingehende Überprüfung der bisher lediglich für den Fall eines AWGN-Kanals verifizierten Performance der Luftschnittstelle unter realistischen Kanalbedingungen an. Gegebenenfalls sind die Kanalschätzung und Synchronisation, insbesondere die hierbei eingesetzten Mittelwertbildungen anzupassen.

## Literaturverzeichnis

[1] A. Burkhard, “Redesign und Charakterisierung des Analog-Frontends für ein auf Software Defined Radio basierendes Funksystem”, Bachelorarbeit, *Hochschule Ulm*, 2010.

[2] M. Düll, “High Level Synthese eines Software Defined Radio Systems zur digitalen Audioübertragung”, Bachelorarbeit, *Hochschule Ulm*, Ulm 2010.

[3] L. J. Cimini, “Analysis and Simulation of a Digital Mobile Channel using Orthogonal Frequency Division Multiplexing”, *IEEE Transactions on Communications*, 33, 665-675, 185.

[4] IEEE Computer Society, “IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements” Part 11: “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, *IEEE Std 802.11*, New York 2012.

[5] A. M. Tonello, S. D'Alessandro, F. Versolatto, C. Tornelli, “Comparison of narrow-band OFDM PLC solutions and I-UWB modulation over distribution grids”, *2011 IEEE International Conference on Smart Grid Communications* (*SmartGridComm*), 149-154, 2011.

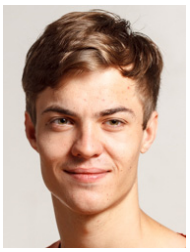
[6] Y. Li, “Basic Concepts” in Y. Li, G. L. Stüber, *Orthogonal Frequency Division Multiplexing for Wireless Communications*, Springer, New York 2006.

[7] A. F. Molisch, *Wireless Communications*, John Wiley & Sons, Chichester 2005.

[8] S. K. Wilson, “Synchronization” in Y. Li, G. L. Stüber, *Orthogonal Frequency Division Multiplexing for Wireless Communications*, Springer, New York 2006.

[9] C. Tellambura, M Friese, “Peak Power Reduction Techniques”, in Y. Li, G. L. Stüber, *Orthogonal Frequency Division Multiplexing for Wireless Communications*, Springer, New York 2006.

Steffen Moll studiert im siebten Semester des Bachelorstudiengangs Nachrichtentechnik an der Hochschule Ulm und arbeitet derzeit an seiner Abschlussarbeit zum Thema Performance Analyse von IEEE 802.11 Wireless Mesh Netzwerken.



Marius Welk studiert im siebten Semester des Bachelorstudiengangs Nachrichtentechnik an der Hochschule Ulm und absolviert derzeit ein Auslandssemester an der Napier University Edinburgh.

Matthias Düll erhielt den akademischen Grad des Bachelor of Engineering in der Fachrichtung Nachrichtentechnik im Jahr 2011 von der Hochschule Ulm. Seit Abschluss seines anschließenden Masterstudiums an der Universität Ulm arbeitet er als Entwicklungsingenieur bei Tesat-Spacecom in Backnang.

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 Funksysteme, Hochfrequenztechnik und die elektromagnetische Verträglichkeit. Prof. Münzner ist Leiter des Instituts für Kommunikationstechnik der Hochschule Ulm.

# Untersuchung maschineller Lernverfahren und Realisierung eines selbstlernenden Algorithmus zur zuverlässigeren Gestenerkennung

David Heese, Karlheinz Blankenbach, Frank Kesel

***Zusammenfassung*—Das GestIC-System von Microchip ist in der Lage, Gesten über ein elektrisches Feld zu erkennen. Mit Hilfe eines selbstlernenden Algorithmus soll die Erkennung von Wischgesten verbessert werden. „Selbstlernend" bedeutet die Fähigkeit, während der Benutzung dazuzulernen und sich an den Benutzer anpassen zu können. Zum Erreichen der gesteigerten Erkennungsrate werden des Weiteren auch Verbesserungen in der Vorverarbeitung durchgeführt.**

***Schlüsselwörter*—Microchip GestIC, MGC3130, maschinelles Lernen, Selbstlernen, Signalaufbereitung.**

## I. Einleitung

Gestenerkennung ist ein aktuelles, doch keineswegs neues Thema. Bereits in den 70-80er Jahren wurden Datenhandschuhe entwickelt, die sowohl Hand- als auch Fingerbewegungen erfassen konnten. Die Vorstellung, ein Computersystem intuitiv mit Bewegungen steuern zu können, klingt vielversprechend. Jedoch ist ein solcher Datenhandschuh vermutlich nicht das komfortabelste Bediengerät. Für eine kurze Bedienung ist der An- und Ausziehaufwand zu groß. Bei längerer Benutzung wird ein solcher Handschuh unangenehm schwer. Daher werden Systeme angestrebt, die Gesten erkennen können, ohne dass der Benutzer ein Gerät führen muss. Eine Möglichkeit, dies zu erreichen, ist die Verwendung von Kameras. Heute bekannte Vertreter dieser Methode sind beispielsweise Microsoft Kinect oder Leap Motion. Sie können einzelne Gliedmaßen eines Menschen erkennen und diese zur einfacheren Auswertung als Skelettmodell wiedergeben. Jedoch sind solche Systeme verfahrensbedingt sehr lichtabhängig. Zudem fühlen sich viele Benutzer durch Kameras beobachtet.

Im Rahmen dieser Thesis wird ein relativ neues System von Microchip eingesetzt – das so genannte GestIC. Dieses ist in der Lage, über ein elektrisches Feld die räumliche Position einer Hand zu erfassen. Zudem kann es vorgegebene Gesten selbstständig erkennen.

David Heese, david.heese@gmail.com, Karlheinz Blankenbach, karlheinz.blankenbach@hs-pforzheim.de, und Frank Kesel, frank.kesel@hs-pforzheim.de, Hochschule Pforzheim, Tiefenbronner Str 65, 75175 Pforzheim.

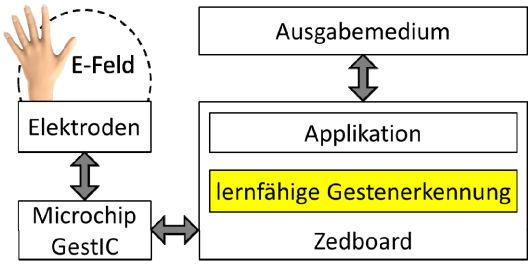



Abbildung 1: Schematischer Aufbau des Gesamtsystems.

Es bietet eine Reihe von Vorteilen gegenüber kamerabasierten Systemen: es ist relativ unabhängig von Umgebungseinflüssen, deutlich günstiger, sehr stromsparend und es erweckt nicht das Gefühl, beobachtet zu werden. Durch diese Eigenschaften eignet sich das System besonders für günstige oder mobile Applikationen.

Was generell von allen Eingabegeräten erwartet wird, ist eine hohe Zuverlässigkeit. Für einen Benutzer ist es äußerst frustrierend, wenn Eingaben nicht oder sogar falsch erkannt werden. Während dieser Arbeit wurde ein Probandentest mit dem GestIC durchgeführt. Dabei wurden nur ca. 64 % der ausgeführten Gesten erkannt. Kaum ein Benutzer dürfte ein System mit einer solchen Erkennungsrate als gut bedienbar empfinden.

Ziel dieser Arbeit war es daher, die Zuverlässigkeit des Systems durch eine gesteigerte Erkennungsrate bei Wischgesten zu verbessern. Dazu wurden Verbesserungen in zwei wesentlichen Bereichen durchgeführt, zum einen in der Signalaufbereitung, wodurch auch schwächere Signale noch ausgewertet werden können, zum anderen in der Erkennung selbst, wobei ein lernfähiges Verfahren angestrebt wurde. Die Idee ist, dass das System während der Benutzung dazulernt und sich auf den Benutzer einstellt. Dabei wurden verschiedene Verfahren aus dem Bereich des maschinellen Lernens untersucht. Bei der Wahl des Verfahrens wurde großen Wert auf die Lernfähigkeit während des Betriebs gelegt. Natürlich wurden auch die Erkennungsrate, der Berechnungs- und Speicheraufwand berücksichtigt. Abbildung 1 zeigt schematisch das hier verwendete Gesamtsystem. In diesem Artikel wird im Wesentlichen nur auf die „lernfähige Gestenerkennung" eingegangen.

Die genannten Verbesserungen sollten auf einem Zedboard implementiert werden. Das Zedboard zeichnet sich durch einen Xilinx Zynq Chip aus, welcher sowohl Mikrocontroller als auch FPGA beinhaltet. Somit bleibt die Möglichkeit offen, aufwendigere Verfahren auf dem FPGA als programmierte Hardware zu implementieren.

## II. Grundlagen

Erste Vorarbeiten mit dem Microchip GestIC wurden von A. Delovski im Rahmen seiner Bachelorthesis geleistet, der verschiedene Elektrodengeometrien und –materialien untersuchte und eigene Auswertesoftware erstellte [1] [2].

### A. Was sind Gesten?

Gesten sind eine nonverbale Kommunikationsweise des Menschen. Häufig werden diese mit den Händen, Armen oder dem Kopf ausgeführt. Einige Gesten, wie Zeigegesten, werden schon früh in der Kindheit erlernt und das Leben lang eingesetzt. Daher stellen sie eine für den Menschen äußerst natürlich und intuitive Kommunikationsform dar. Aus diesem Grund ist eine Gestensteuerung als Methode zur Interaktion mit einem Computer auch so wünschenswert. Es vereinfacht die Bedienung, indem Kommunikationsmethoden verwendet werden, mit denen jeder aus den alltäglichen zwischenmenschlichen Interaktionen vertraut ist.

Das Microchip GestIC System ist in der Lage, drei verschiedene Gestentypen zu erkennen: berührungslose Wischgesten, berührungslose Kreisgesten (s. Abbildung 2) und Touchgesten. Im Rahmen dieser Arbeit liegt das Hauptaugenmerk auf den vier Wischgesten und deren verbesserten Erkennung.

### B. Microchip GestIC

Das GestIC ist ein Chip (MGC3030/3130) von Microchip, mit dem 3D-Positionen und Gesten erkannt werden können. Im Gegensatz zu heute üblichen Erkennungssystemen von berührungslosen Gesten nutzt es ein elektrisches Feld zur Detektion der Hand (Abbildung 3).

Um dieses Feld zu erzeugen, benötigt es eine großflächige Sendeelektrode. Feldänderungen kann es über 4-5 Empfangselektroden erfassen. Die Elektroden können aus einem beliebigen leitenden Material, z. B. Kupfer, bestehen. Dadurch ist das System sehr kostengünstig und einfach an eigene Applikationen anzupassen. Microchip empfiehlt die vier Empfangselektroden entsprechend der Himmelsrichtungen im Rechteck anzuordnen, während die fünfte, optionale Elektrode im Zentrum des Rechtecks aufgebracht ist. Die Sendeelektrode sollte großflächig in einer Lage unter den Empfangselektroden positioniert sein. Eine typische Elektrodenanordnung wird durch Abbildung 4 veranschaulicht. Mit Hilfe des elektrischen Feldes kann eine Hand im Abstand von bis zu 10 cm erfasst

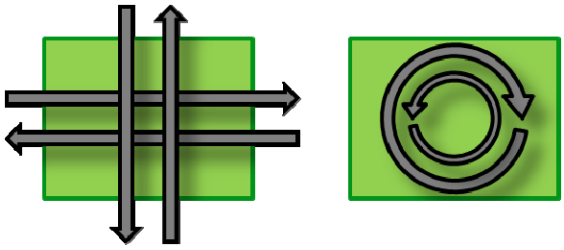


Abbildung 2: Von GestIC erkannte berührungslose Gesten: Wischgesten in vier Richtungen (links) und Kreisgesten in beide Richtungen (rechts).

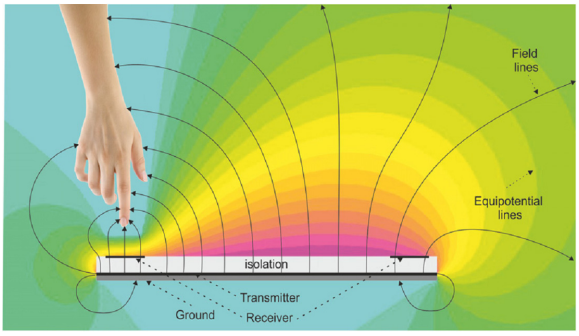



Abbildung 3: E-Feld des GestIC unter Einfluss einer menschlichen Hand [3].

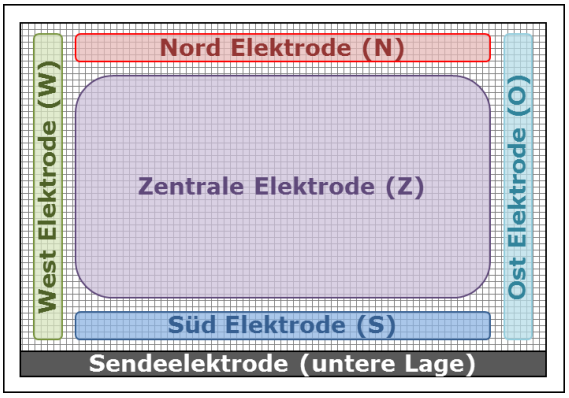



Abbildung 4: Typische Anordnung der GestIC Elektroden.

werden. Weitere technische Details sind in Tabelle 1 aufgeführt [3].

### C. Avnet Zedboard

Das Zedboard ist ein Evaluations- und Entwicklungsboard, welches mit einem Xilinx Zynq-7000 Chip (XC7Z020) ausgestattet ist (Zed bedeutet Zynq™ Evaluation and Development). Dieser Chip macht das Board besonders universell einsetzbar, denn neben zwei ARM Cortex-A9 Prozessorkernen besitzt er einen FPGA mit 85.000 programmierbaren Logikzellen [4]. In Abbildung 5 ist ein Zedboard mit seinen wesentlichen Schnittstellen abgebildet.

Tabelle 1: Technische Details des Microchips GestIC [3].

| | |
|---|---|
| Positionsauflösung | bis zu 150 dpi |
| Detektionsabstand | bis zu 10 cm |
| Elektrodenfläche | max. 14 cm x 14 cm |
| Abtastrate | 200 Hz |
| Ext. Schnittstelle | I²C |
| Stromverbrauch (bei 3,3 V) | aktiv: 20 mA<br>self wake-up: 110 µA<br>deep sleep: 9 µA |

Mit einer maximalen Taktfrequenz von 667 MHz bieten die Prozessorkerne ausreichend Leistung und die 512 MByte DDR3-Arbeitsspeicher bieten auch für komplexere Programme genügend Platz [4] [5]. Zudem ist über die vielen Steckverbinder das Anschließen weiterer Komponenten einfach möglich. Beispielsweise kann das Prozessorsystem über einen der Pmod-Verbinder verschiedene Peripherieeinheiten wie I²C, GPIO, SPI, UART usw. nach außen führen.

### *D. Verbindung der Boards*

Das Zedboard kann über eine I²C-Schnittstelle und zwei GPIO-Pins an den GestIC-Chip angeschlossen werden (s. Abbildung 6). Die Daten- (SDA) und Taktleitung (SCL) verbinden die I²C-Schnittstellen der beiden Kommunikationspartner. Außerdem werden zwei weitere Leitungen benötigt: TS und MCLR. Über die TS-Leitung (Transfer Status Line) kann der GestIC-Chip den Host über abholbereite Daten informieren. Während der Übertragung der Daten zieht der Host diese Leitung auf Masse und verhindert somit, dass der GestIC-Chip die Daten aktualisiert. Mit der Leitung MCLR kann der Host den GestIC-Chip neu starten.

### *E. Verfügbare Signale des GestIC*

Über die I²C-Schnittstelle kann man eine Vielzahl an Information des GestIC abrufen, darunter die Elektrodensignale und die 3D-Positionsdaten. Die Elektrodensignale werden sowohl als Rohsignale als auch in aufbereiteter Form zur Verfügung gestellt. Nachfolgend soll auf diese drei Informationen näher eingegangen werden. Dazu wurden diese Informationen während einer Wischgeste von links nach rechts aufgezeichnet.

In Abbildung 7 sind die Rohsignale der fünf Elektroden abgebildet. Am Anfang und am Ende befindet sich das System im Ruhezustand – die Signalpegel bleiben konstant. In der Mitte findet die Ausführung der Geste statt. Mit Eindringen der Hand steigen die Signalpegel. Es ist deutlich zu sehen, dass die Signale unterschiedliche Offsets aufweisen, was eine Auswertung erschwert. Deshalb führt das GestIC bereits selber eine Aufbereitung der Signale durch. Die aufbereiteten Signale der Elektroden sind in Abbildung 8 dargestellt.

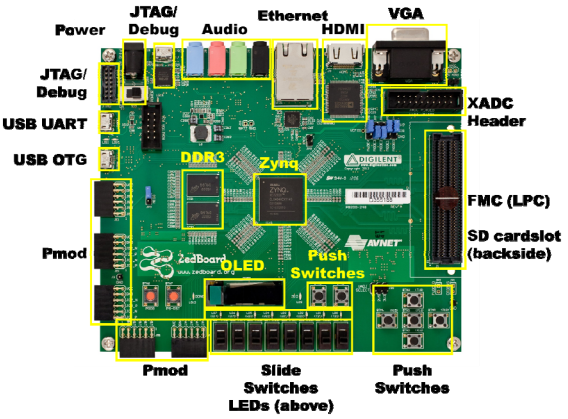



Abbildung 5: Foto des Zedboards mit Hervorhebung wesentlicher Komponenten/Schnittstellen (Quelle: zedboard.org).

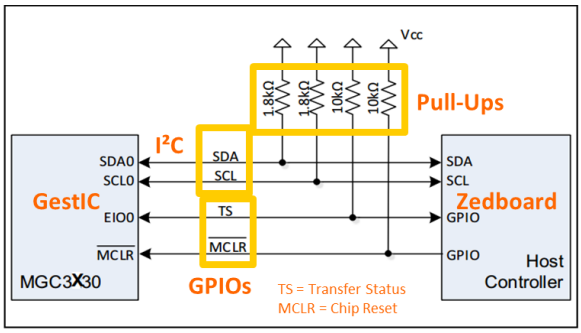



Abbildung 6: Von Microchip empfohlene Applikationsschaltung, um den GestIC-Chip an einen Host anzuschließen [3].

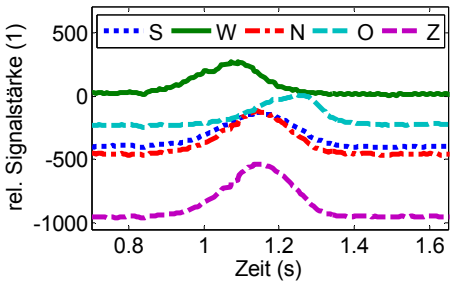



Abbildung 7: Rohsignale („Uncalibrated Signals“ – CIC) der fünf Elektroden während einer Wischgeste von links nach rechts. Die Signalstärke ist ein relativer Wert (einheitslos).

Zunächst einmal soll die zeitliche Beziehung der Signale betrachtet werden. Zwar ist diese natürlich auch in den Rohsignalen vorhanden, jedoch nicht so deutlich ersichtlich. Anhand dieser zeitlichen Beziehung kann eine Wischgeste und ihre Ausführungsrichtung identifiziert werden. Eine Wischgeste von links nach rechts beginnt bei der West-Elektrode und endet bei der Ost-Elektrode. Diese zeitliche Folge ist auch in

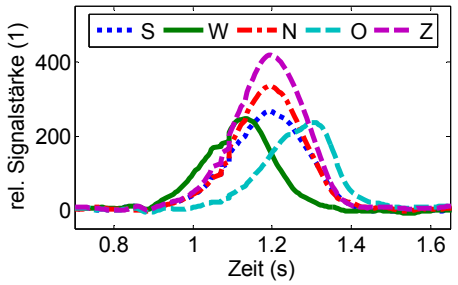



Abbildung 8: Aufbereitete Signale („Signal Deviation“ – SD) der fünf Elektroden während einer Wischgeste von links nach rechts. Die Signalstärke ist ein relativer Wert (einheitslos). Achtung: Skalierung gegenüber Abbildung geändert.

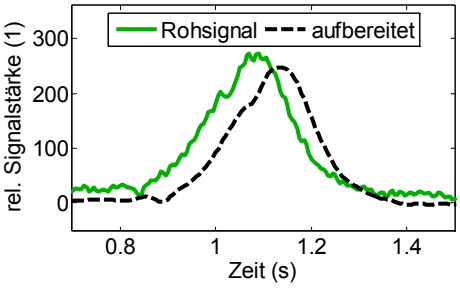



Abbildung 9: Vergleich von Rohsignal und aufbereitetem Signal der West-Elektrode während einer Wischgeste von links nach rechts. Die Signalstärke ist ein relativer Wert (einheitslos).

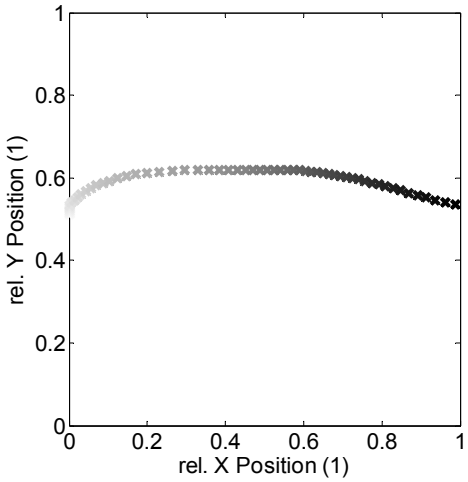



Abbildung 10: XY-Positionsverlauf der Wischgeste von links nach rechts (entspricht einer Draufsicht). Positionsangaben sind relativ zu entsprechenden Elektrodenabständen.

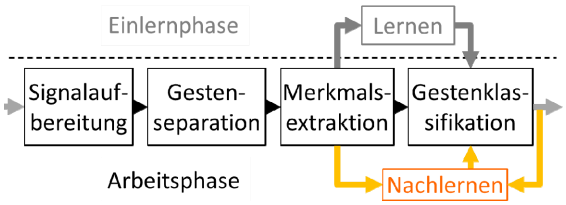



Abbildung 11: Blockdiagramm des Gestenerkennungsprozesses.

den Signalen wiederzuerkennen. Dass dies bei den aufbereiteten Signalen besser ersichtlich ist, liegt an den korrigierten Offsets. Im Ruhezustand befindet sich der Signalpegel bei etwa 0. An dieser Stelle soll angemerkt werden, dass das System die Offsetkorrektur nur zu bestimmten Zeitpunkten ausführt. In Testaufzeichnungen hat sich gezeigt, dass die Signale mit der Zeit durch Signaldrift Offsets entwickeln können.

Um weitere Unterschiede deutlicher zu machen, sind in Abbildung 9 sowohl das Rohsignal als auch das aufbereitete Signal einer Elektrode aufgetragen. Hierbei wurde das Signal der West-Elektrode gewählt, weil dieses auch als Rohsignal kaum einen Offset aufweist. Dadurch lässt es sich gut zusammen mit dem aufbereiteten Signal in einem Diagramm darstellen. Alle daraus gewonnenen Erkenntnisse lassen sich gleichbedeutend auf die restlichen Elektroden übertragen.

Neben den korrigierten Offsets fallen zwei weitere Dinge auf. Zum einen ist das aufbereitete Signal deutlich glatter. Von daher ist davon auszugehen, dass ein Filter zur Rauschminimierung verwendet wurde. Zum anderen tritt das aufbereitete Signal mit einer Verzögerung von etwa 50 ms auf. Diese Verzögerung wird vermutlich durch die Filterung verursacht. Neben den Signalen der Elektroden wird auch die räumliche Position der erfassten Hand geliefert. In Abbildung 10 ist der Positionsverlauf derselben Wischgeste dargestellt. Die dritte Dimension (Z - Abstand zur Elektrodenfläche) wurde der Übersichtlichkeit halber weggelassen.

Es fällt auf, dass die Wischgeste in einem leicht nach unten geöffneten Bogen ausgeführt wurde. Das ist typisch für eine menschliche Wischbewegung mit dem Arm bzw. der Hand. Diese Bewegung wird üblicher Weise als Rotation des Arms mit dem Ellenbogen als Drehpunkt ausgeführt.

Die Positionssignale stehen jedoch nur zur Verfügung, wenn die Elektrodensignale groß genug sind, also die Hand nah genug im Feld ist. Da es Ziel dieser Arbeit ist, Gesten zuverlässiger zu erkennen, müssen auch schwächere Signale noch ausgewertet werden. Das ist mit dieser Positionsinformation nicht möglich. Es muss auf die Elektrodensignale zurückgegriffen werden.

### *F. Gestenerkennung*

Die Kernaufgabe bei der Gestenerkennung ist eine Klassifikation. Eine neue Information soll einer von mehreren gegebenen Klassen zugeordnet werden. In diesem Fall gibt es vier Klassen. Jede Gestenvariante bzw. Richtung der Wischgeste entspricht einer Klasse. Vor der Klassifikation müssen die Daten jedoch entsprechend aufbereitet werden. Der gesamte Gestenerkennungsprozess ist in Abbildung 11 als Blockdiagram dargestellt.

Im ersten Schritt, der Signalaufbereitung, werden die Signale möglichst von Störeinflüssen wie Offsets oder Rauschen befreit. Der zweite Schritt, die Gestenseparation, entspricht der Detektion der Zeitbereiche, in denen eine Geste ausgeführt wurde. Aus dem kontinuierlichen Signalstrom müssen nur die Abschnitte betrachtet werden, in denen eine Geste stattgefunden hat. Vor der letztendlichen Klassifikation findet noch ein weiterer Schritt statt, die Merkmalsextraktion. In diesem Schritt werden innerhalb der Signale Merkmale extrahiert, die möglichst aussagekräftig für die zu klassifizierenden Gesten sind. Dieser Schritt hat starken Einfluss auf die Klassifikation. Zum einen gibt die Art der Merkmale (z. B. wertkontinuierlich oder wertdiskret) vor, welche Klassifikationsverfahren überhaupt eingesetzt werden können. Zum anderen wirken sich der Umfang der Merkmale auf den Klassifikationsaufwand und die Aussagekraft der Merkmale auf die erreichbaren Erkennungsraten aus. Zur eigentlichen Klassifikation werden üblicherweise Verfahren des maschinellen Lernens verwendet. Solche Verfahren müssen vor Einsatz anhand eines Trainingsdatensatzes auf die jeweilige Problemstellung eingelernt werden. Im Fall dieser Arbeit ist es des Weiteren erforderlich, dass die Klassifikation auch während des Einsatzes nachlernen kann.

### *G. Maschinelles Lernen*

Das maschinelle Lernen hat seine Ursprünge in der Forschung im Bereich der künstlichen Intelligenz. Wie der Begriff „Lernen“ andeutet, wird bei Verfahren dieser Art Wissen durch Erfahrung gewonnen. In einer Einlernphase erkennt das System Gesetzmäßigkeiten innerhalb von Trainingsbeispielen und ist nach Abschluss des Einlernens fähig, ähnliche Daten zu erkennen bzw. zuzuordnen. Diese Verfahren ermöglichen es somit Probleme zu lösen, ohne dass man die darin verborgenen Gesetzmäßigkeiten kennen muss. Häufig ist deshalb der Einsatz eines Verfahrens des maschinellen Lernens einfacher, als für das jeweilige Problem eine Lösung explizit zu implementieren.

Das maschinelle Lernen bietet Verfahren für unterschiedliche Verwendungszwecke. Für diese Arbeit wurden nur Verfahren untersucht, die zur Klassifikation verwendet werden können. Nachfolgend sollen die vier in dieser Arbeit verwendeten Verfahren grundlegend erläutert werden.

#### *1) k-Means (KM)*

k-Means ist eigentlich ein Clusteringverfahren, dessen Prinzip jedoch auch zur Klassifikation genutzt werden kann. Das k-Means Verfahren gruppiert die Daten eines Datensatzes in $k$ Gruppen (Cluster) $\{c_1, \ldots, c_k\}$. Der Parameter $k$ wird vorgegeben und sollte der Anzahl Klassen entsprechen. Der Datensatz besteht dabei aus $m$ Daten $\{g^{(1)}, \ldots, g^{(m)}\} \in \mathbb{R}^n$. Jedes Datum weist demnach $n$ Merkmale auf. Somit lässt sich der Datensatz mit $m$ Punkten in einem n-dimensionalen Merkmalsraum abbilden. Die Gruppierung der Punkte erfolgt abstandsbasiert. Zunächst wird für jede Gruppe ein Schwerpunkt $\{\mu_1,\ldots,\mu_k\} \in \mathbb{R}^n$ in diesem Raum vorgegeben. Diese initialen Schwerpunkte werden häufig willkürlich gewählt, jedoch gibt es auch Verfahren, um geeignetere Initialschwerpunkte zu finden. Iterativ werden die Abstände der Punkte zu allen Gruppenschwerpunkten berechnet. Die Punkte werden jeweils der Gruppe des nächsten Gruppenschwerpunktes zugeordnet und der Gruppenschwerpunkt angepasst. Dieses Vorgehen wird solange wiederholt, bis sich die Zuordnung der Punkte nicht mehr ändert. Somit minimiert dieses Verfahren die Varianzen der Daten innerhalb der jeweiligen Gruppe [6]. Abbildung 12 veranschaulicht das Verfahren in einigen Schritten anhand einer Datenmenge im 2D-Raum und der Einteilung in zwei Gruppen.

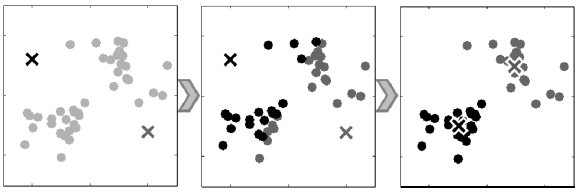


Abbildung 12: Vorgehen des k-Means Verfahrens zur Gruppierung von Punkten innerhalb eines 2D-Raums in zwei Gruppen. Hellgraue Punkte (im linken Bild) sind noch nicht zugeordnet. Grau und schwarz entspricht den zwei Gruppen, deren Schwerpunkte durch X gekennzeichnet sind.

Im linken Bild von Abbildung 12 ist noch keine Zuordnung erfolgt (alle Punkte hellgrau). Die X entsprechen den Gruppenschwerpunkten (grau und schwarz). Das mittlere Bild zeigt die erste Einteilung der Punkte anhand der initialen Gruppenschwerpunkte. Im rechten Bild ist die finale Einteilung und die letztendliche Position der Gruppenschwerpunkte gezeigt.

Soll das Verfahren zur Klassifikation eingesetzt werden, so entspricht die Einlernphase der Gruppierung wie oben beschrieben. Danach sind die Schwerpunkte für jede Gruppe bzw. Klasse bekannt. Zur Klassifikation eines neuen Datums $g^{(i)}$ werden die Abstände zu allen Schwerpunkten berechnet. Der Schwerpunkt, der am nächsten zum Datum liegt, entspricht dann der Klasse des Datums. Zum Nachlernen könnte der entsprechende Schwerpunkt in Richtung des neuen Datums verlagert werden.

#### *2) k-nächste-Nachbarn (KNN)*

Der k-Nearest-Neighbor-Algorithmus (zu Deutsch k-nächste-Nachbarn) ist ebenfalls ein abstandsbasiertes Klassifikationsverfahren, das auch auf Merkmalen in Form eines n-dimensionalen Vektors realer Zahlen zurückgreift. Der zugrundeliegende Datensatz besteht also auch aus $m$ Daten $\{g^{(1)}, \ldots, g^{(m)}\}$ mit $g^{(i)} \in \mathbb{R}^n$.

Bei diesem Verfahren liegen alle Daten im Merkmalsraum vor. Kommt ein neues Datum hinzu, das klassifiziert werden soll, werden die Abstände von diesem Datum zu allen anderen bereits vorliegenden

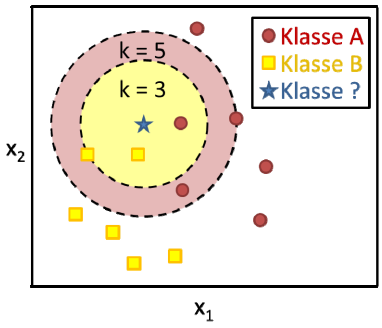



Abbildung 13: Klassifikation eines neuen Datums (Stern) in einem 2D-Raum durch KNN bei unterschiedlich gewähltem *k*. Bei *k* = 3 wird das Datum als Klasse *B* (Quadrat) und bei *k* = 5 als Klasse *A* (Kreis) klassifiziert.

Daten berechnet. Für die Klassifikation sind nur die *k* am nächsten liegenden Daten, die *k* nächsten Nachbarn, relevant. Anhand deren bekannter Klassen wird das neue Datum der in der Mehrheit vorliegenden Klasse zugeordnet [7].

Abbildung 13 zeigt beispielhaft, wie ein neues Datum in einem zweidimensionalen Raum aufgrund einer vorliegenden Datenmenge und unterschiedlichem *k* klassifiziert würde. Der Parameter *k* ist der einzige Parameter dieses Verfahrens und sollte geschickt gewählt werden, da dieser sich auf die Klassifikationsgüte (und geringfügig auf den Rechenaufwand) auswirken kann. Durch ein kleines *k* könnte die Klassifikation bei schlecht klassifizierten Daten ebenfalls falsch klassifizieren. Ein größeres *k* macht das Verfahren robuster gegen einzelne schlecht klassifizierte Daten. Wird *k* jedoch zu groß gewählt, wird ein größeres Umfeld im Merkmalsraum betrachtet und es könnten zunehmend Daten anderer Klassen ins Gewicht fallen.

*3) Naiver Bayes (NB)*

Der naive Bayes-Klassifikator ist ein stochastischer Klassifikator, der auf dem Bayes-Theorem basiert. In dieser Arbeit wird der naive Bayes auf Basis von Merkmalen in Form einer Symbolfolge $<\alpha_1, \alpha_2, \ldots>$, also wertdiskret, verwendet. Die Länge einer Folge ist nicht festgelegt und variiert mit jeder Gesteninstanz. Wenn die Klassen durch $C = \{c_1, \ldots, c_k\}$ definiert sind, kann die Klassifikation anhand der nachfolgen Formel durchgeführt werden:

$$c = \arg\max_{c_j \in C} P(\alpha_1, \alpha_2, \ldots \mid c_j) P(c_j)$$

Die Formel setzt voraus, dass für jede Klasse die Auftrittswahrscheinlichkeit jeder beliebigen Symbolfolge bekannt ist. Diese Wahrscheinlichkeit wird mit der Grundwahrscheinlichkeit, dass die jeweilige Klasse auftritt, multipliziert. In diesem Fall ist davon auszugehen, dass jede Klasse gleich wahrscheinlich auftritt. Somit kann $P(c_j)$ vernachlässigt werden. Zur Klassifikation wird letztendlich die Klasse herangezogen, die die größte Auftrittswahrscheinlichkeit für die Symbolfolge bietet. Da es jedoch praktisch unmöglich ist, für jede mögliche Symbolfolge die Auftrittswahrscheinlichkeit zu wissen, trifft der naive Bayes eine vereinfachende Annahme. Es wird dem Auftreten der Symbole eine bedingte Unabhängigkeit unterstellt. Somit ist die Auftrittsreihenfolge quasi irrelevant. Diese Annahme ist zwar nicht korrekt (deshalb „naiv“), liefert in der Praxis jedoch häufig gute Ergebnisse [8]. Der betroffene Term und damit die oben genannte Formel lassen sich somit folgendermaßen umformen:

$$P(\alpha_1, \alpha_2, \ldots \mid c_j) = \prod_{alle\ s_i} P(\alpha_i \mid c_j)$$

$$c = \arg\max_{c_j \in C} \prod_{alle\ s_i} P(\alpha_i \mid c_j)$$

Die Auftrittswahrscheinlichkeit einzelner Symbole lässt sich deutlich einfacher ermitteln: durch Zählungen innerhalb der Trainingsdaten. Um eine schwache Abhängigkeit aufrecht zu erhalten, können Teilfolgen definierter Länge innerhalb der Symbolfolge zu Symbolen zusammengefasst werden. Sei die Folge z. B. <*A*, *A*, *B*, *A*, *B*>, könnte man diese auch als Folge von Teilfolgen wie <*AA*, *AB*, *BA*, *AB*> betrachten. Das Symbolalphabet hat sich somit von {*A*, *B*} zu {*AA*, *AB*, *BB*, *BA*} geändert. Dabei sollten jedoch zwei Dinge berücksichtigt werden. Je länger die Teilfolge, desto schwieriger wird es, deren Auftrittswahrscheinlichkeiten zu ermitteln. Außerdem wächst der Speicherbedarf exponentiell mit der Länge der Teilfolge und der Anzahl zugrunde liegender Symbole.

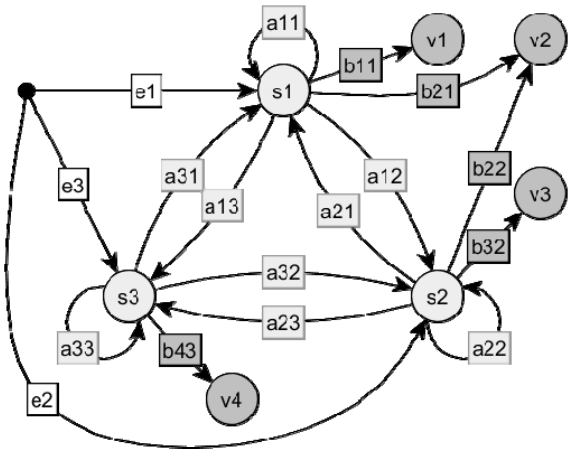



Abbildung 14: Zustandsgraph eines HMM mit drei Zuständen und vier möglichen Emissionen.

*4) Hidden-Markov-Model (HMM)*

Das Hidden-Markov-Model ist ebenfalls ein stochastisches Verfahren, das Symbolfolgen als Merkmale verwendet. Der GestIC-Chip nutzt dieses Verfahren zur Gestenerkennung. In Abbildung 14 ist ein bei-

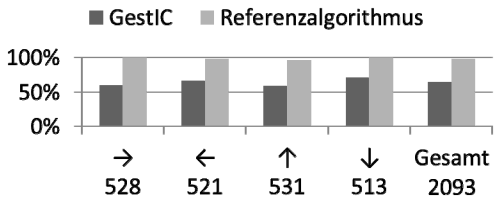



Abbildung 15: Erkennungsraten beider Erkennungsverfahren für die vier Wischgesten in Prozent. Zusätzlich sind unten die Anzahlen der pro Gestentyp erhobenen Daten angegeben.

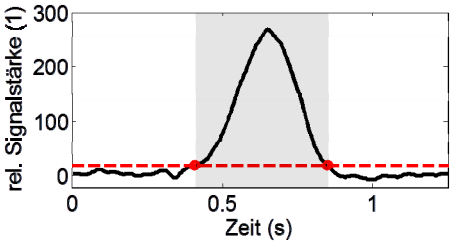



Abbildung 16: Separation des Zeitbereichs, in dem eine Geste stattfindet, durch eine Pegelschwelle (gestrichelt).

spielhaftes HMM als Zustandsgraph dargestellt. Ein HMM besteht aus einer Menge von versteckten („hidden“) Systemzuständen $S = \{s_1, s_2, \ldots, s_N\}$ und einer Menge von Emissionen $V = \{v_1, v_2, \ldots, v_M\}$, die den möglichen Symbolen entsprechen. Der erste Zustandseintritt wird durch den Vektor $e$ mit der Länge $N$ bestimmt, welcher die Eintrittswahrscheinlichkeiten aller Zustände beinhaltet. Die weiteren Zustandsübergänge werden von der Übergangswahrscheinlichkeitsmatrix A (Größe $N$ x $N$) bestimmt. Beim Übergang in einen neuen Zustand wird eine Emission gemäß der Emissonswahrscheinlichkeitsmatrix B (Größe $N$ x $M$) ausgestrahlt [9].

Liegt nun eine Symbolfolge vor, kann berechnet werden, wie wahrscheinlich es ist, dass diese von einem HMM emittiert wurde. Diese Wahrscheinlichkeit wird Produktionswahrscheinlichkeit genannt. Zur Klassifikation wird für jede Geste ein separates HMM benötigt und das HMM herangezogen, das für die Symbolfolge die größte Produktionswahrscheinlichkeit liefert.

## III. Anwendung

### A. Probandentest – Gewinnung von Testdaten

Zur Evaluierung des GestIC-Systems und Gewinnung von Testdaten wurde ein Probandentest durchgeführt. Bei diesem haben 14 Testpersonen die vier Wischgesten in vier verschiedenen Varianten jeweils 10 mal ausgeführt. Die Varianten unterschieden sich in der ausführenden Hand (links oder rechts) und der Lage der Elektroden (horizontal unter der Hand oder vertikal vor der Hand). Dies sollten die üblichsten Varianten sein, in denen das GestIC-System eingesetzt wird. Neben der Erkennung durch das GestIC war zusätzlich ein eigener Erkennungsalgorithmus als Referenz aktiv. Dabei handelt es sich um ein einfaches Verfahren, das die Geste anhand der Signalreihenfolge erkennt. Die Signalzeitpunkte werden dabei durch ihre Maxima bestimmt. Damit werden auch relativ schwache Signale noch als Geste ausgewertet.

Insgesamt wurden dabei 2093 brauchbare Gestendaten gewonnen. Eigentlich wären es mehr gewesen, jedoch mussten einige aussortiert werden, zum einen aufgrund einer schlechten Ausführung oder starken Rauschens. Zum anderen kam es vor, dass das System während der Geste den Offset korrigierte und die Geste somit nicht auswertbar war.

In Abbildung 15 sind die erreichten Erkennungsraten beider Erkennungsverfahren für die vier Wischgesten aufgetragen. Insgesamt hat das GestIC-System nur ca. 64 % aller Wischgesten korrekt erkennen können. Das Referenzverfahren konnte hingegen 98 % richtig erkennen. An dieser Stelle sollte erwähnt werden, dass das GestIC-System Gesten in größerem Abstand häufig nicht auswertet. Demnach liegt die geringe Erkennungsrate nicht zwangsläufig am Erkennungs- bzw. Klassifikationsalgorithmus. Dennoch zeigt der Referenzansatz, dass höhere Erkennungsraten möglich sind. Jedoch ist mit diesem Ansatz ein Nachlernen nicht sinnvoll möglich. Des Weiteren ist er für komplexere Gesten ungeeignet. Die Klassifikationsverfahren aus dem Bereich des maschinellen Lernens sind dafür geeigneter. Der Probandentest zeigt zudem, dass auch in der Signalaufbereitung und der Gestenseparation Verbesserungspotentiale vorhanden sind. Für eine erhöhte Zuverlässigkeit bei der Gestenerkennung sollten daher auch in diesen Bereichen Optimierungen durchgeführt werden.

### B. Signalaufbereitung

Bereits im GestIC-System wird eine Signalaufbereitung durchgeführt, sowohl eine Rauschfilterung als auch eine Offsetkorrektur zu diskreten Zeitpunkten. Um auch schwächere Signale auswerten zu können, müssen diese zwei Maßnahmen nach Möglichkeit weiter verbessert werden.

Zur Rauschfilterung wurden FIR-Tiefpassfilter mit Grenzfrequenzen von 10 und 20 Hz und Ordnungen von 8, 16 und 32 getestet. Der Filter 16. Ordnung mit 20 Hz kam der Rauschfilterung des GestIC am nächsten. Eine Steigerung der Ordnung führte zu einer größeren Verzögerung des Signals und nur bei 10 Hz Grenzfrequenz zu einer deutlichen Minderung des Rauschens. Die Grenzfrequenz von 10 Hz führte jedoch bereits dazu, dass auch die Gesteninformation selbst gedämpft wurde. Im gegebenen Zeitrahmen wurde für die Rauschfilterung keine deutlich bessere Lösung gefunden.

Tabelle 2: Gegenüberstellung der unterschiedlichen Klassifikationsverfahren. In Klammern bei den Aufwänden ist deren Abhängigkeit angegeben. Der Ausdruck „* Iterationen“ bedeutet, dass das Verfahren eine nicht statisch bestimmbare Anzahl an Iterationen durchläuft.

| | k-Means (KM) | k-Nächste-Nachbarn (KNN) | Naiver Bayes (NB) | Hidden-Markov Modelle (HMM) |
|---|---|---|---|---|
| Methode | Abstand | Abstand | Stochastisch | Stochastisch |
| Merkmale | Merkmalvektor | Merkmalvektor | Symbolfolge | Symbolfolge |
| Struktur | 4 Merkmale „Zeitpunkte“ | 4 Merkmale „Zeitpunkte“ | 4er Quantisierung 2er Teilfolgen | 4er Quantisierung 2 Zustände |
| Einlernaufwand | linear * Iterationen | einfach | linear | quadratisch * Iterationen |
| Klassifikations-aufwand | linear (Klassen, Merkmale) | linear (Daten, Merkmale) | linear (Folge, Klassen) | quadratisch (Zustände, Folge) |
| Nachlern-aufwand | linear (Klassen, Merkmale) | einfach | linear | quadratisch * Iterationen |
| Speicher | gering | hoch | gering - hoch | gering - hoch |
| Erkennungsrate | 99,2% | 99,4% | 99,9% | 99,7% |

Störend an der Offsetkorrektur des GestIC ist, dass diese nur zu vereinzelten Zeitpunkten stattfindet und dies in manchen Fällen auch während einer Geste. Zwischen den Zeitpunkten der Offsetkorrekturen entwickelt das Signal jedoch bereits wieder einen Offset aufgrund von Signaldrifts. Als Verbesserung soll deshalb eine Offsetkorrektur eingesetzt werden, die kontinuierlich während der Ruhephase arbeitet. Dabei werden die Signale mit jedem neuen Wert um einen Bruchteil des neu entstandenen Offsets korrigiert.

Da die Rauschfilterung des GestIC nicht deutlich verbessert werden konnte, werden für die weitere Verarbeitung die „aufbereiteten Signale“ des GestIC, jedoch mit deaktivierter Offsetkorrektur, verwendet. Die Offsets werden durch die oben beschriebene gleitende Methode korrigiert.

### *C. Gestenseparation*

In der Gestenseparation sollen die Zeitbereiche der Signale separiert werden, in denen eine Geste ausgeführt wurde. Dazu gibt es zwei Möglichkeiten: anhand der Signalpegel oder der Steigung der Signale. Abbildung 16 zeigt anhand eines Elektrodensignals, wie mit Hilfe einer Pegelschwelle die Geste separiert werden kann. Mit der Pegelschwelle werden die Zeitbereiche separiert, in denen die Signale diese Schwelle überschreiten. Dafür ist eine gute Offsetkorrektur notwendig.

Bei der Separation der Geste anhand der Steigung ist der Offset nahezu unbedeutend. Der Gestenbeginn wird mit Überschreiten einer positiven Steigungsschwelle detektiert und deren Abschluss durch Unterschreiten einer negativen Steigungsschwelle. Jedoch kann das Rauschen Steigungen aufweisen, die zu einem Überschreiten der Schwellen führt und somit falsche Zeitbereiche separiert würden. Aus diesem Grund sollen die Gesten durch eine Pegelschwelle separiert werden.

### *D. Merkmalsextraktion*

Wie bereits erwähnt ist die Art, der Umfang und die Aussagekraft der Merkmale von signifikanter Bedeutung für die Klassifikation. In dieser Arbeit wurden zwei Arten von Merkmalen untersucht, zum einen ein Merkmalsvektor mit den Zeitpunkten der Elektrodensignale und zum anderen eine Folge von Richtungssymbolen.

Beim Merkmalsvektor steht man vor dem Problem, den Elektrodensignalen einen Zeitpunkt zuzuordnen. Hier wurden verschiedene Möglichkeiten erprobt. Die besten Erkennungsraten hat das zugleich einfachste Verfahren geliefert: die Ermittlung anhand der Zeitpunkte der Maxima. Zudem hat sich gezeigt, dass der Zeitpunkt des zentralen Elektrodensignals keinen Mehrwert bringt. Somit reicht ein vierdimensionaler Merkmalsvektor. Diese Merkmale haben jedoch nur für die Wischgesten eine hohe Aussagekraft. Deutlich flexibler hingegen ist die Folge von Richtungssymbolen als Merkmal.

Die Berechnung der Richtungssymbole benötigt mehrere Schritte. Zunächst einmal müssen Koordinaten berechnet werden. Für Wischgesten reichen die X- und Y-Koordinaten. Näherungsweise kann die X-Koordinate durch die Differenz aus den Signalen der Ost- und West-Elektrode, welche zur Normierung

durch die Summe aller Elektrodensignale geteilt wird, berechnet werden. Gleichermaßen kann die Y-Koordinate berechnet werden, wobei jedoch die Differenz aus den Signalen der Nord- und Süd-Elektrode gebildet wird. Zwar entspricht dies keiner exakten Position, jedoch reicht sie zur Berechnung der Richtungssymbole aus. Aus aufeinanderfolgenden Koordinaten können dann durch Differenzbildung Richtungsvektoren berechnet werden. Quantisiert man diese noch in eine vorgegebene Anzahl möglicher Symbole, erhält man die gesuchten Richtungssymbole. Es wurden unterschiedliche Quantisierungsanzahlen erprobt, von 2 bis zu 12. Eine Quantisierung in vier Symbole hat sich am geeignetsten erwiesen. Auch zur Klassifikation von Kreisgesten zeigten sich die vier Quantisierungszustände nur unbedeutend schlechter als eine höhere Anzahl Quantisierungszustände. Zudem werden die Klassifikationsverfahren mit höheren Anzahlen komplexer.

### *E. Wahl des Klassifikationsverfahrens*

In Tabelle 2 ist eine Übersicht der untersuchten Klassifikationsverfahren gegeben. Die Wahl eines Verfahrens erfolgte aufgrund der erreichten Erkennungsraten, des Rechen- und des Speicheraufwandes. Aber auch die Art der Merkmale beeinflusste letztendlich die Wahl. Wie bereits erwähnt, wurde die Richtungssymbolfolge als Merkmal bevorzugt, da sich mit ihr auch komplexere Gesten gut abbilden lassen.

Alle Verfahren erzielten mit den Testdaten hervorragende Erkennungsraten von über 99 %. Die zwei abstandsbasierten Verfahren, KM und KNN, nutzten den Merkmalvektor mit den Zeitpunkten der vier Elektrodensignale (ohne zentrale Elektrode). Hingegen verwendeten die stochastischen Verfahren, NB und HMM, die Folge von Richtungssymbolen als Merkmal. Dabei wurden die Richtungen in vier mögliche Symbole quantisiert.

Ein hoher Einlernaufwand ist hinnehmbar, da dieser nur einmalig und vor dem eigentlichen Einsatz des Verfahrens anfällt. Der Klassifikations- und Nachlernaufwand ist hingegen von signifikanter Bedeutung, da dieser in Echtzeit von dem eingebetteten System bewältigt werden muss. Hierbei schneiden insbesondere KM und NB sehr gut ab. KNN scheint zwar noch effizienter zu sein, aber durch die große Datenmenge, die sich mit jedem Einlernen weiter vergrößert, fällt der effektive Klassifikationsaufwand sehr groß aus. Beim HMM zeigen die Rechenaufwände bei effizienten Verfahren eine quadratische Ordnung [9].

Wie bereits angedeutet, vergrößert sich die Datenmenge beim KNN mit jedem eingelernten Datum, was sich auch negativ auf den Speicherbedarf auswirkt. Das KM merkt sich hingegen nur die Gruppenschwerpunkte, wodurch nur wenig Speicher benötigt wird. Beim NB ist der Speicheraufwand exponentiell von der Struktur abhängig. Für die gewählte Struktur fällt dieser jedoch sehr gering aus. Das HMM kommt prinzipiell ebenfalls mit wenig Speicher aus. Sollen zum Nachlernen jedoch alle Daten mitgeführt werden, so zeigt sich das gleiche Speicherproblem wie bei KNN.

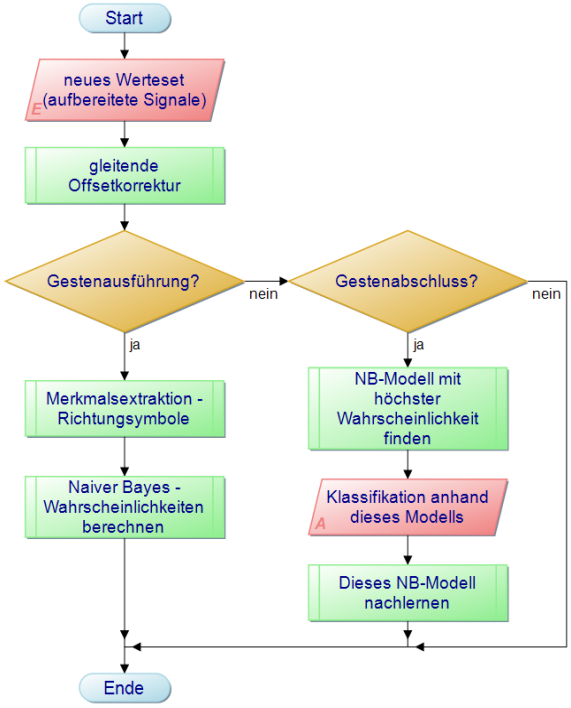



Abbildung 17: Programmablaufplan der implementierten Gestenerkennung.

Somit fiel die Wahl auf das naive Bayes-Verfahren zur Klassifikation der vier Wischgesten. Es erzielte hervorragende Erkennungsraten, ist sehr recheneffizient und bei der verwendeten Struktur auch speichereffizient. Außerdem ist es theoretisch in der Lage, mit der Richtungssymbolfolge als Merkmal auch komplexere Gesten zu erlernen und zu klassifizieren. Wie gut solche Gesten mit diesem Verfahren in der Praxis wirklich erkannt werden können, ist aufgrund der bedingten Unabhängigkeit unklar und müsste weitergehend untersucht werden.

## IV. IMPLEMENTIERUNG

Aufgrund der Effizienz der ausgewählten Verfahren ist eine Implementierung in Hardware nicht notwendig und wird daher in C als Software durchgeführt. Zudem stellt Microchip eine in C geschriebene Software-API zur Verfügung, welche die Kommunikation mit dem GestIC vereinfacht. Als Ausgangslage wird auf dem Zedboard eine Ubuntu Linux-Distribution mit grafischer Oberfläche (HDMI) aufgespielt. Dadurch wird eine größere Flexibilität geboten und es ist zukünftig einfacher, diverse Anwendungen für dieses System zu erstellen[1]. Die GestIC-API muss natürlich an einigen Stellen an die verwendete Hardware und das Betriebs-

[1] Eine Anleitung zum Aufsetzen dieser Linux-Distribution findet sich auf www.zedboard.org.

system angepasst werden. Insbesondere die Sende- und Empfangsroutinen müssen selbst implementiert werden.

In Abbildung 17 ist das Prinzip der implementierten Gestenerkennung als grober Programmablaufplan veranschaulicht. Über die GestIC-API wird ca. alle 5 ms (200 Hz) ein neuer Wertesatz der aufbereiteten Elektrodensignale (ohne Offsetkorrektur) gelesen. Diese werden dann durch die gleitende Offsetkorrektur weiter aufgebessert. Danach erfolgt die Gestenseparation durch die Signalpegelschwelle. Überschreiten alle Signale diese Schwelle, wird die Gestenausführung erkannt. Überschreiten nicht mehr alle Signale die Schwelle, während dies im Zyklus davor noch der Fall war, so wird der Gestenabschluss detektiert. Während der Gestenausführung wird das nächste Richtungssymbol berechnet und anhand von diesem werden die Gesamtwahrscheinlichkeiten aller möglichen Gesten nach dem naiven Bayes-Verfahren aktualisiert. Beim Gestenabschluss wird die Geste mit der größten Gesamtwahrscheinlichkeit ermittelt und zur Klassifikation herangezogen. Danach werden die Auftrittswahrscheinlichkeiten der klassifizierten Geste um die neue Symbolfolge angepasst und somit nachgelernt.

## V. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein eigener Algorithmus zur Gestenerkennung entwickelt. Dieser erzielte für den Testdatensatz mit 99,9 % richtig erkannter Gesten eine deutlich höhere Erkennungsrate als das GestIC System, welches lediglich ca. 64 % der Gesten richtig erkennen konnte. Der Testdatensatz umfasste dabei insgesamt über 2000 Gestenausführungen der Wischgesten in die vier Richtungen.

Diese deutliche Verbesserung war nicht allein durch ein „besseres“ Klassifikationsverfahren möglich. Prinzipiell funktioniert das originale Klassifikationsverfahren für die Wischgesten hervorragend, vorausgesetzt die Signalstärke ist groß genug. Daher wurden auch im Bereich der Signalaufbereitung und der Gestenseparation eigene Ansätze eingebracht. Es wurde eine gleitende Offsetkorrektur eingesetzt, die während der Ruhephasen des Systems die Signaloffsets kontinuierlich anpasst. Die Gesteninformationen werden durch eine Signalpegelschwelle separiert. Durch diese zwei Maßnahmen ist es möglich, auch schwächere Signale noch als Geste auszuwerten.

Zur Klassifikation wird ein naiver Bayes-Klassifikator verwendet. Dieser klassifiziert die Gesten anhand einer Richtungssymbolfolge, die aus den Signalen berechnet wird. Mit einer solchen Richtungssymbolfolge können auch komplexere Gesten gut abgebildet werden. Der naive Bayes-Klassifikator zeichnet sich durch seinen geringen Rechenaufwand aus und auch der Speicheraufwand fällt in der eingesetzten Form gering aus. Des Weiteren ist auch ein Nachlernen mit diesem Verfahren einfach möglich. Jedoch ist der Nutzen des Nachlernens für diesen Fall ungewiss. Bei einer Erkennungsrate von 99,9 % ist kaum eine weitere Verbesserung möglich. Evtl. verschlechtert sich diese sogar, falls das System unsauber ausgeführte Gesten erlernt. Die Verfahren sind aufgrund ihrer Effizienz gut als Software für ein eingebettetes System implementierbar. Eine Umsetzung als programmierbare Hardware (FPGA) war daher nicht erforderlich.

## Danksagung

Die Autoren bedanken sich bei den Herren Manuel Gaiser und Christoph Ußfeller für ihre hilfreichen Anstöße bei den maschinellen Lernverfahren.

## Literaturverzeichnis

[1] A. Delovski, *Touchless Touch - Evaluierung eines 5 Elektroden Touchless 3D-Gestenerkennungs-ICs, sowie Untersuchung alternativer Elektrodengeometrien und Entwicklung von Algorithmen zur adaptierten Gestenerkennung*. Bachelorthesis, Dez. 2013.

[2] A. Delovski und K-H. Blankenbach, „Touchless Touch - Evaluierung eines Gesten-ICs mit (flexiblen) Elektroden und Aufbau eines kompletten Systems“, *Proceedings Entwicklerforum "HMI – Komponenten & Lösungen", WEKA Fachverlag München* (2014), Mai, S. 20ff.

[3] Microchip (Hrsg.): MGC3030/3130 *3D Tracking and Gesture Controller Data Sheet*. Microchip, 2015. http://ww1.microchip.com/downloads/en/DeviceDoc/40001667D.pdf. Zugriff: 09.03.2015.

[4] AVNET (Hrsg.): *ZedBoard - Hardware User’s Guide*. AVNET, 01 2014. http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf. Zugriff: 03.02.2015.

[5] Xilinx (Hrsg.): Zynq-7000 *All Programmable SoC* (Z-7010, Z-7015, and Z-7020): *DC and AC Switching Characteristics*. Xilinx, 02 2015. http://www.xilinx.com/support/documentation/data_sheets/ds187-XC7Z010-XC7Z020-Data-sheet.pdf. Zugriff: 05.03.2015.

[6] A. Ng, *Standford University, Machine Learning, Lecture notes 7a*. http://cs229.stanford.edu/notes/cs229-notes7a.pdf. Zugriff: 31.03.2015.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006. ISBN 0387310738.

[8] A. Ng, *Stanford University, Machine Learning, Lecture notes 2*. http://see.stanford.edu/materials/aimlcs229/cs229-notes2.pdf. Zugriff: 02.03.2015.

[9] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition”, *Proceedings of the IEEE* (1989), Feb., pp. 257–286. http://dx.doi.org/10.1109/-5.18626. – DOI 10.1109/5.18626. ISSN 0018–9219.

David Heese absolvierte 2011 sein Studium an der Dualen Hochschule (DHBW) in Karlsruhe als B.Eng. in Mechatronik. Als Ingenieur war er 2011-2013 bei der Schneeberger GmbH in Höfen/Enz an der Entwicklung und Produktion von Längenmesssystemen beteiligt. Im Rahmen seines Masterstudiums in Embedded Systems (M.Sc.) an der Hochschule Pforzheim arbeitet er derzeit an seiner Thesis, welche Thematik dieses Artikels ist.

Prof. Dr. Karlheinz Blankenbach ist seit 1995 an der Hochschule Pforzheim. Seine Forschungsaktivitäten liegen im Bereich der elektronischen Displays und Graphical User Interfaces (GUI, HMI). Er ist seit 1999 Vorsitzender des Konferenzbeirates der Electronic Displays Conferernce (www.electronic-displays.de) und seit 2000 im Vorstand des Deutschen Flachdisplay Forums (www.displayforum.de), seit 2011 Vorsitzender.

Frank Kesel erhielt den akademischen Grad des Dipl.-Ing. in Elektrotechnik im Jahr 1988 von der Universität Karlsruhe (TH) und den Grad des Dr.-Ing. im Jahr 1994 von der Universität Hannover. Er ist seit 1999 Professor für Integrierte Schaltungstechnik an der Hochschule Pforzheim.

# A Web-Based Monitoring Tool for Metering Bus (EN13757-3)

Thomas Matt, Manuel Schappacher, Axel Sikora

***Abstract*—The Metering Bus, also known as M-Bus, is a European standard EN13757-3 for reading out metering devices, like electricity, water, gas, or heat meters. Although real-life M-Bus networks can reach a significant size and complexity, only very simple protocol analyzers are available to observe and maintain such networks. In order to provide developers and installers with the ability to analyze the real bus signals easily, a web-based monitoring tool for the M-Bus has been designed and implemented. Combined with a physical bus interface it allows for measuring and recording the bus signals. For this at first a circuit has been developed, which transforms the voltage and current-modulated M-Bus signals to a voltage signal that can be read by a standard ADC and processed by an MCU. The bus signals and packets are displayed using a web server, which analyzes and classifies the frame fragments. As an additional feature an oscilloscope functionality is included in order to visualize the physical signal on the bus. This paper describes the development of the read-out circuit for the Wired M-Bus and the data recovery.**

***Index Terms*—Bus analyzer, signal recording, ADC, M-Bus.**

## I. Introduction

The Metering Bus, also known as M-Bus, is a communication protocol used for various meters and data collectors, e.g. for metering of gas, heat, or electricity. It is defined in the European standard EN13757 [1]. The M-Bus is widely used, mainly for apartment buildings, but it is also deployed in the industrial sector. The standard has some issues because of unclear or missing definitions, which might result in inconsistencies during the setup of an M-Bus network. To reduce the problems, companies founded the working group 4 (WG4) [2] in the Open Metering Systems (OMS) group to revise the current standard. The Laboratory "Embedded Systems and Communication Electronics" (ESK) from Offenburg University of Applied Sciences is an active contributor to this group. To overcome existing problems, the authors developed an M-Bus bus analyzer which is shown in figure 1. The M-Bus bus analyzer is a tool that can help to create a properly-working M-Bus network. Therefore, the analyzer (also called sniffer) records the complete signal flow of an M-Bus network and monitors and analyzes the data flow by use of an integrated web server.

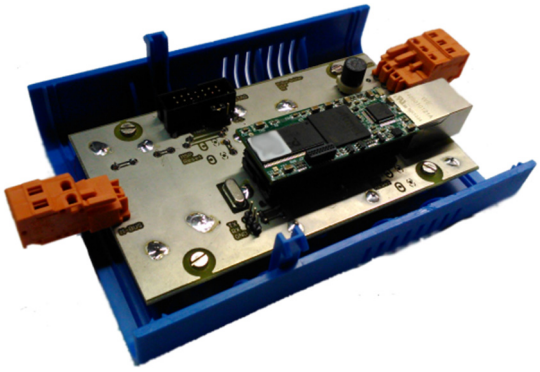


Figure 1: M-Bus protocol analyzer prototype.

The main platform for the analyzer is the Wireless M-Bus analyzer capt2web [3] which uses an ARM9 controller[4], running a Linux distribution and an embedded web server to illustrate the bus data. Since the ARM controller was originally developed as a head unit for a Wireless M-Bus RF module, one of the main tasks in this project was to replace the existing RF module by a read-out circuit for Wired M-Bus and to adapt the head unit's frontend to accommodate to the wired M-Bus protocol. Furthermore, the read-out circuit had to be developed, since no other solution had been available.

This paper is organized as follows: Chapter II explains the Physical (PHY) and Data Link Layer (DLL) of the M-Bus protocol. After this, chapter III discusses the possibilities to tap the signal with minimum influence on the measured system. Then, chapter IV describes the hardware architecture of the analyzer. The software functionality will be presented in chapter V. Finally, chapter VI describes the head unit, and chapter VII provides a summary and an outlook.

Thomas Matt, thomas.matt@hs-offenburg.de, Manuel Schappacher, manuel.schappacher@hs-offenburg.de, Axel Sikora, axel.sikora@hs-offenburg.de, are with Hochschule Offenburg, Badstraße 24, 77652 Offenburg.

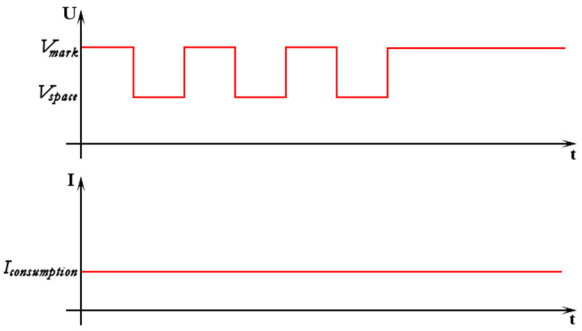



Figure 2: M-Bus physical levels during a master to slave transmission.

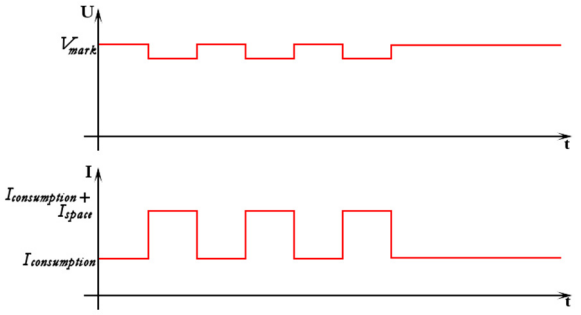



Figure 3: M-Bus physical levels during a Slave to Master transmission.

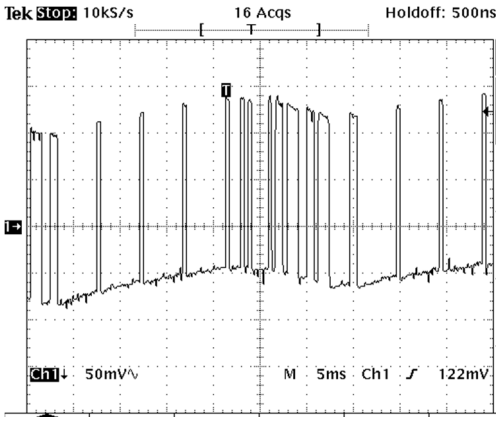



Figure 4: Real slave signals in a screenshot from an oscilloscope.

## II. Metering Bus

### A. General

The Metering Bus is based on a common UART interface that uses a fixed parameter set to transmit a byte. Therefore, the overhead is composed of a single start bit, a parity bit, and a stop bit. For the transmission speed, different baud rates in the range from 300 to 38,400 Baud can be used. However, baud rates above 9,600 Baud are neither recommended, nor widely used due to the characteristics of the physical layer.

The M-Bus does not provide a separate wire as bus clock, but uses self-clocking. To reduce the number of wires and to avoid collisions on the common bus, the M-Bus is based on a single master to control the data flow. Both send and receive operations can be done over a single pair of wires. In addition to the bus arbitration, the master provides the power supply for each slave, resulting in a constant voltage offset on the bus. Since this offset is not clearly specified in the standard, the voltage can be in the range of several volts, while typical values range is from 24 V to 50 V. The M-Bus provides both the downlink communication from master to slave (M2S) and the uplink communication from slave to master (S2M).

### B. Master to Slave Communication (M2S)

For the communication, a logical one is represented by a fixed offset voltage $V_{mark}$ and a logical zero by an offset voltage $V_{space}$ of -12 V [1]. Therefore, the communication is defined more or less by voltage drops of 12 V [1]. The current $I_{consumption}$ is constant during the transmission, as shown in figure 2.

### C. Slave to Master Communication (S2M)

For the communication, the slave modulates its own current consumption. In this case, the current will not be constant anymore and the master that supplies the power to the slaves can decode the communicated data through the total current consumption on the bus.

In its idle state, a slave consumes a constant current of 1.5 mA [1], which is also known as unitload or $I_{mark}$. Depending on the current consumption, a device can consume a single or integer multiples of a unitload. For example, if a device needs 5 mA to run, it will burden the bus with four unitloads (6 mA). In the S2M direction, $I_{mark}$ represents a logical one sent by a slave. The maximal possible unitload within an M-Bus network depends on the master providing the power supply. Typically, a master can provide up to 250 unit loads. The sum of all unitloads is called $I_{consumption}$.

The logical zero or $I_{space}$ is represented by a current consumption in the range between 11 mA and 20 mA [1]. The modulation of the current consumption also has an effect on the offset voltage of the M-Bus master. As shown in figure 3, a slave's logical zero will always cause a voltage drop on the bus. In order to verify that, a part of this project was to record the signals on the transmission lines. The oscillogram in figure 4 shows a record of an S2M communication. It can be seen that the current transmission implies a small voltage swing on the bus according to theory. This swing depends on the internal resistance of the master and on the resistance of the transmission line. Since the internal resistance of a master is not unambiguously specified in the standard, it cannot be predicted. Typical values are between 1 Ω and 68 Ω. These values result in a voltage

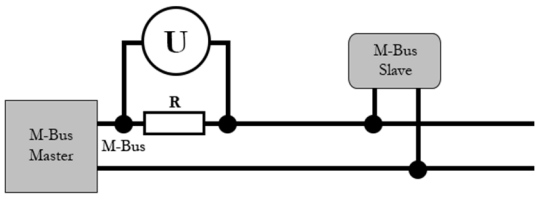



Figure 5: Signal tapping over a shunt.

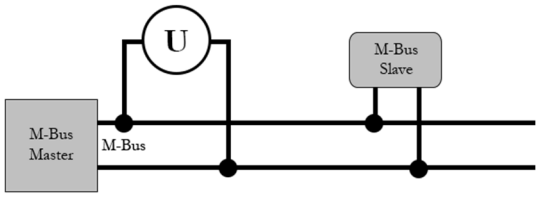



Figure 6: Signal tapping parallel to a slave.

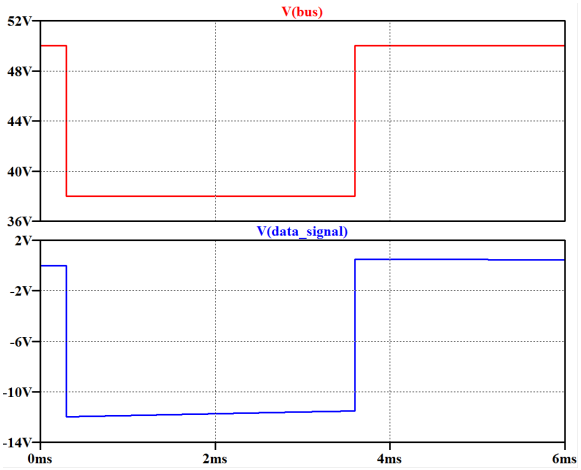



Figure 7: DC cancelation using a passive 1st order high pass filter with $f_c = 1$ Hz at 300 Baud.

swing at the master clamp between 7 $mV_{pp}$ and 1.1 $V_{pp}$. Figure 3 shows an example of some slave signals taken from a test application given as reference. The voltage swing in this case is $V_{pp} \approx 200$ mV. For these measurements, the oscilloscope must be set to AC mode, because the offset voltage is about 200 times higher than the signal.

## III. Signal Tapping and Acquisition

A bus or protocol analyzer needs to interpret the electrical signals of the bus. In the case of the M-Bus, this becomes even more difficult because of the two different communication methods. Existing M-Bus sniffers only interpret parts of the physical signals, since they are designed as master devices without the abstraction of the application and data link layer. Furthermore, they only consider S2M direction, but not the M2S, since this communication is always originated by the master itself. This makes the approach unsuitable for a real analyzer.

For the M2S direction, various compatible driver ICs are available as commercial-off-the-shelf (COTS) devices, e.g. TSS721a [5] from Texas Instruments. However, it is not possible to use such a transceiver for the planned analyzer, as it would increase the current consumption of the bus and therefore would affect the bus itself. Since this is an unwanted behavior for an analyzer, the usage of such components should be avoided. In addition, there is also a circuit needed for the S2M direction, because the previous described solution cannot be used. The project goal was to create an analyzer and not a bus master. To be able to analyze both directions different methods of the signal acquisition have been investigated. However, since there were no suitable methods or existing sniffers available, an own approach had to be developed.

### *A. Current Methods*

The S2M direction uses the current modulation for data transmission, allowing these signals to be grabbed with a current clamp or a shunt. The shunt method is shown in figure 5. Both alternatives can only be used for the S2M communication, because in M2S direction, the bus current is constant. Also a problem of this method is that the total S2M communication can only be regained at the master's clamp, because of Kirchhoff's current law. Each slave creates a new junction in the network, and so the sum of all currents can only be measured at the master's clamp.

### *B. Voltage Methods*

The simplest way to retrieve M2S and S2M signals with a single circuit is to use the bus voltage as shown in figure 6, since both directions have effects on the bus voltage. However, the offset voltage of the M-Bus remains an unknown parameter, as it depends on the implementation of the devices. Since also the offset itself does not have any informational content, the basic idea of this approach is to separate the data signals from the offset.

### *C. DC Cancelation*

The UART interface uses rectangular pulses to transmit the data, meaning the data has a direct current (DC) part as well. However, if an offset blocker is used to split up the data signal from the bus signal, it will have some impact on the frequency characteristics of the data signal, i.e. it will lead to a distortion of the signal. This is because of the system function of the offset blocker. The data signal will be convolved with this function, but the influence of the blocker will be marginal if the cutoff frequency is much smaller than the transmission frequency. Figure 7 shows what a data signal will look like after a passive 1st order high pass. The

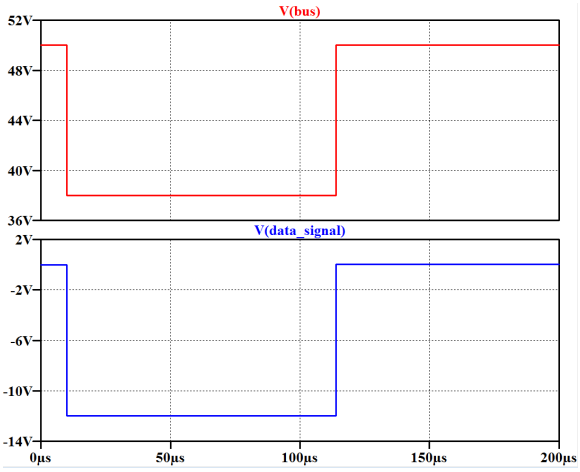



Figure 8: DC cancelation using a passive 1st order high pass filter with $f_c$ = 1 Hz at 9,600 Baud.

cutoff frequency is $f_c$ = 1 Hz, and the simulated baud rate is 300 Baud, which is also the lowest data rate for the sniffer. The upper graph shows the input signal, the lower the output signal of the high pass filter. As can be seen, even though the levels might drop slightly over time, the edges are clear enough to perform the required signal analysis. With higher baud rates, e.g. 9,600 Baud, these effects will no longer be recognizable as can be seen in figure 8. Higher baud rates do have frequencies around the baseband like lower baud rates, but the amplitudes are not this high. So the waveform of the higher baud rates will be less affected than the waveforms at lower baud rates. This method is also used in the AC mode in oscilloscopes.

To remove the voltage offset, the M-Bus analyzer uses such a 1st order high pass filter. After the offset is removed, the data signal will be amplified to get a larger voltage swing from the S2M direction. Furthermore, a specified offset will be added to achieve positive voltage levels. This signal will be captured by a microcontroller (MCU) using an ADC, and finally, the signal analysis will be performed.

The M-Bus does not specify which of the two wires of the cable is used as ground and which one carries the signal. This fact requires the sniffer to allow a voltage inversion at the bus input. In case there is no function or no component to invert the voltage, or if the device is not connected correctly, the phase of the data signal will be inverted, and the data cannot be analyzed correctly. A rectifier would represent the default solution here, but because of its transverse current it would also increase the influence on the bus. Therefore, the ADC input can simply be inverted to achieve the same effect. When the sniffer is plugged in with an incorrect phase, the configuration can be changed to fix this problem. This phase inversion is a very simple process where only the thresholds of the program and the sample of the ACD will be inverted. Thereafter, the whole program will run properly.

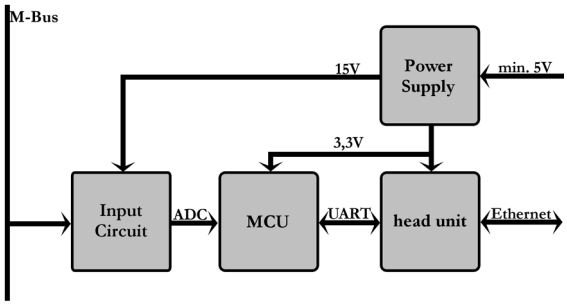



Figure 9: Architecture of the protocol analyzer.

Table 1: *CSPB* at 9,600 Baud.

| | *OSR* 32 | *OSR* 64 | *OSR* 128 |
|---|---|---|---|
| *CLK* | 39 | 19 | 9 |
| *CLK*/2 | 19 | 9 | 4 |
| *CLK*/4 | 9 | 4 | 2 |
| *CLK*/8 | 4 | 2 | 1 |

## IV. Architecture

### *A. General Architecture*

Figure 9 shows the protocol analyzer with its four functional blocks. The analyzer must be supplied with at least 5 V, with a maximum current of 500 mA. The power supply transforms the input voltage into two different output voltages.

First, the input circuit requires a voltage of 15 V to accommodate the M2S voltage swing. This swing is defined by a minimum peak to peak voltage of 12 V, but it is also possible to have a higher swing. To prevent the limits of the circuit to be reached, it works with 15 V. The rest of the circuit only needs a voltage of 3.3 V. The remaining blocks are mainly responsible for the actual signal capturing and analysis.

### *B. Analog to Digital Conversion*

The selected MCU is a derivate of the Texas Instruments MSP430 family including a delta-sigma converter (DSC). An advantage is the high resolution, compared to an ADC with a successive approximation register (SAR) conversion method. The high resolution is needed to recognize the S2M signal without an additional amplifier. A disadvantage of a DSC is the long conversion time, compared to a SAR ADC. In theory, there is only one sample per bit needed to recognize the logical level. However, if the signals are too short, or if there is no separate synchronization signal available, single bits might be missing. Since S2M communication implies small signals on the M-Bus an asynchronous interface is used.

For a single bit conversion, there are at least two or more samples needed. To fulfil this requirement, the

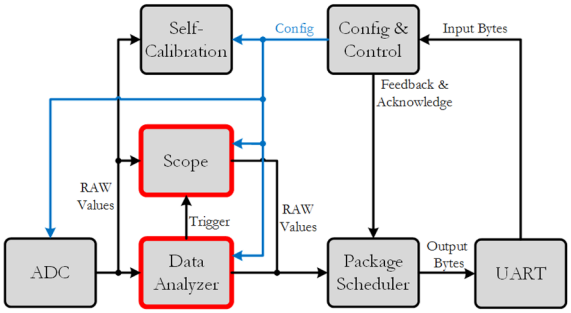



Figure 10: Software block diagram.

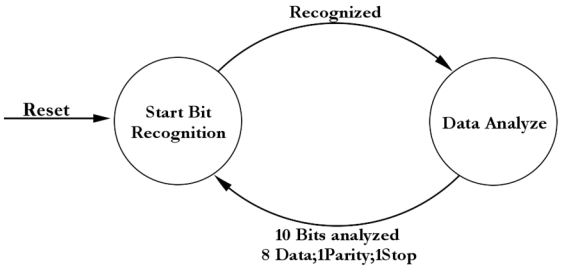



Figure 11: Finite state machine for data recognition.

conversion time of the ADC must match the configured baud rate. Therefore, a conversion table, as shown in table 1, can be used, which shows the count of samples per bit *CSPB* for a baud rate of 9,600. The conversion time depends on the selected clock frequency $CLK_{max}$ and the oversampling rate *OSR*. Both parameters are relevant to calculate the *CSPB*, as shown in equation 1.

$$CSPB = \left\lfloor \frac{t_{bit} \cdot CLK}{OSR} \right\rfloor \quad (1)$$

The parameter $t_{bit}$ describes the time, which is needed to transmit one single bit, e.g. $t_{bit} = 104$ µs for 9,600 Baud. The selected configuration for 9,600 Baud is $CLK = 3$ MHz and $OSR = 64$, thus $CSPB = 4$. The baud rates, which can be recorded, are between 300 and 9,600 Baud; otherwise, *CSPB* is too low to recognize a bit properly. This limitation at 9,600 Baud gives enough points to realize a scope function. The algorithms were designed with Matlab [6]. The microcontroller sends the ADC samples via its UART interface to a workstation. Thus, the algorithm could be written in Matlab and later be implemented in the controller source code.

## V. Software

The overall architecture of the MCU software is shown in figure 10, including its two main functionalities, the data analysis and the scope function. Furthermore, the different software parts can be controlled and configured via the UART interface using a packet-based communication protocol.

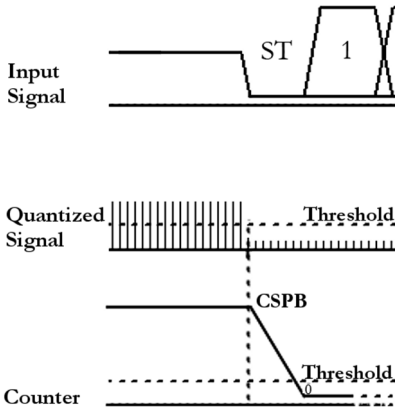



Figure 12: Recognition of the start bit.

In addition, there is also a mode for self-calibration. A self-calibration is required since the signal analysis uses level thresholds to regain the information. The values of the calibration are stored at the flash segment of the MCU, so a recalibration is only needed if there are significant errors within the conversion. The calibrated values are automatically reloaded during the initialization of the controller. The footprint of the software currently requires 5.6 kB for the code segment in flash and 860 Bytes of RAM memory.

### *A. Signal Analysis*

According to the UART specification, an incoming byte always begins with a start bit. However, since the interface is asynchronous, the timing of the start bit is unknown. Therefore, the internal timing needs to be synchronized to the bus transmission timing. If the timing is not synchronized, the following data bits will not be analyzed correctly and bit errors might occur. To avoid this, the data recognition state machine, as shown in figure 11, consists of two different states to analyze the data. After a reset, the software assumes that the next incoming bit is a start bit. If start bit comes in, the state machine will switch to the data analysis state.

#### *1) Start Bit Recognition*

The recognition of the start bit uses the time discrete functionality of an ADC. Each ADC sample will be passed to a state machine that uses an averaging method and thresholds to detect the start bit. Once the start bit is detected, the main state machine switches to the data analyze function which is described in the following chapter.

Averaging multiple values is very simple from a mathematical point of view. All values need to be summed up and divided by the number of values. Some controllers, like the used MSP430, do not have special hardware units for divisions. This will cause a timing problem, since a division needs many more machine cycles in that case. To avoid this, the state machine uses

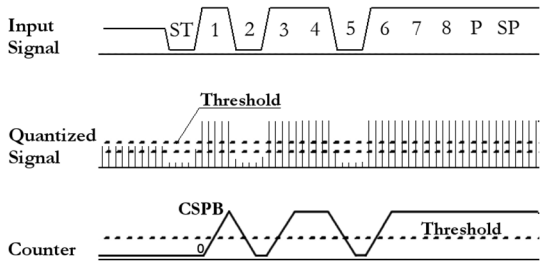



Figure 13: Signal analysis via ADC.

a counter mechanism. If the sample becomes lower than a predefined threshold, the counter will be decremented; if the detected value is higher, the counter will be incremented. The maximum value of the counter is equal to the CSPB. For example, in case the ADC uses eight samples per bit, the upper limit for the counter will be eight as well. The lower limit will always be zero. This method needs only a few machine cycles. Figure 12 shows the steps of how to recognize the start bit. The incoming signal is quantized and the counter's value is changed accordingly. In case the value of the counter runs below the threshold of the counter, the start bit is detected and the data analysis is triggered.

*2) Data Analysis*

In case of a successful start bit recognition, the following ADC values are understood as data bits. Like the start bit recognition, the data analysis also uses a state machine, which is driven by the ADC values. The main difference compared to the start bit recognition is that the state machine decides for the value of the recorded bit after each *CSPB* instead of after each sample.

The data analysis state machine uses three thresholds instead of two, if it is compared to the start bit recognition state machine. Two of the three thresholds will decide, how a counter value will changed, depending on the current ADC sample. The third threshold is to regain the logical information from the counter value. Figure 13 shows an example of the transmitted byte 0xED and how the data will be analyzed with this algorithm.

An additional functionality of the data analysis state machine is the parity filter, allowing the parity to be checked before the data is copied into the input buffer. If the parity calculation is not equal to the transmitted parity bit, the byte will not be discarded. Since parity errors are an important indicator for the bus analysis, this function can be enabled or disabled by the user.

*B. Scope Function*

The scope function is another important feature of the sniffer. Recording of the data by use of an ADC allows for sending the samples directly to the head unit. The head unit can then use these samples to draw a time and

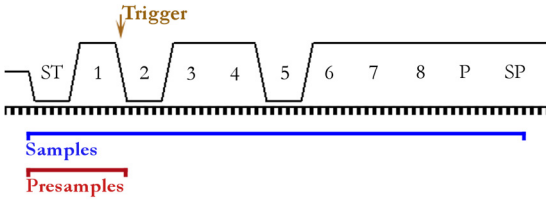



Figure 14: Example for scope record.

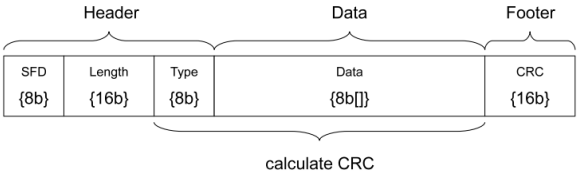



Figure 15: Packet design [7].

voltage discrete graph of signal edges and voltage levels on the bus. For the scope function common oscilloscope controls have been adopted. The scope has the ability to trigger at the end of each bit of the communication, as well as to operate in a free-running mode. A free running-mode prompts the data without waiting for any trigger. When using triggers, the scope offers two run modes. The first run mode is called single shot and will only record one interval and then freeze. The second mode is the continuous mode, where the record will be retriggered until the user stops the recording.

The scope function also supports presamples. For example, when using triggers, it would not be possible to see the start bit since the record can only be triggered after the bit was recognized. A memory or presample functionality helps in this situation. Figure 14 shows an example where the record is triggered by a first data bit. The presample functionality is implemented with a ring buffer using two pointers and a size of the ringbuffer equal to the maximum sample size as indicated in figure 14. The two pointers are used to get access to the buffer. One pointer is used to store the data and the other to pick the data. The position difference of both pointers is used for the presamples. If there are no presamples selected, both pointers are pointing to the same data segment, otherwise the pick pointer runs after the store pointer. The maximum possible number of presamples is the number of samples – 1.

The maximum possible sample frequency of the scope is limited by the communication interface between head unit and driver and the size of the ADC sample. Because of the 16 bit ADC resolution, one sample has the size of two bytes. The interface between the head unit and the driver is a UART communication with 115200 Baud. The maximum sample frequency can be estimated by use of equation 2.

$$f_{sample} < \frac{Baudrate}{n_{ADCBytes}} \qquad (2)$$

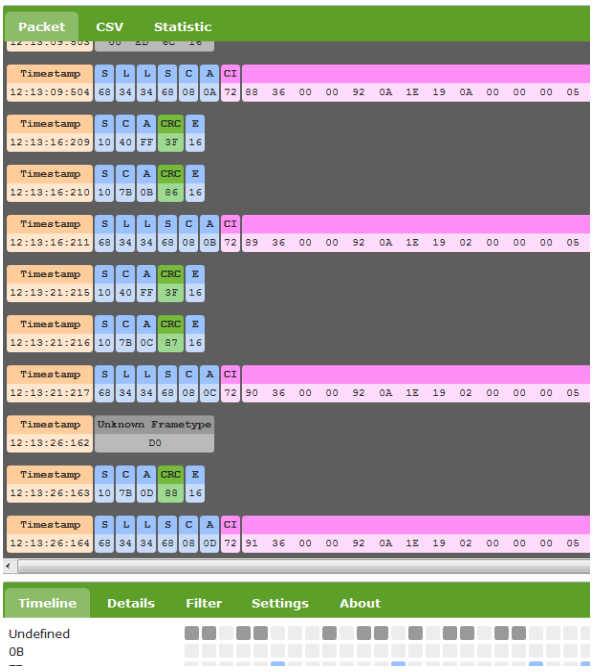



Figure 16: capt2web web interface.

Taking into account the data of table 1, the maximum sampling frequency is limited to 46 kHz.

### *C. Head Unit Interface*

The main platform for the analyzer is the Wireless M-Bus analyzer capt2web [3] that uses a packet-based protocol to communicate with a driver module. This protocol had to be implemented to finally integrate the wired M-Bus driver into the existing sniffer backend. As mentioned, figure 15 shows the structure of such a packet. The packets begin with a start field with a constant value, which is called start frame delimiter (SFD). Each packet has a length field describing the size of the following data field. The content of the data field is described by the type field at the beginning of the data field. At the end of each package, a cyclic redundancy check (CRC) over the complete data field is appended. For this, the calculated CRC has to be updated immediately while the ADC continues its sampling to save time.

The MCU uses an array of packets to send the data. This is required because a packet cannot be filled and transmitted at the same time. If there is only one packet to fill and send, there would be some access problems if some new data comes in. The handling of these packages will be done by a scheduler. So the data analysis or the scope function will just pass the data to the scheduler, which handles the rest.

## VI. Head Unit

The head unit uses a web server to process the recorded data, which can be displayed with any ordinary web browser like firefox or internet explorer. The usage of a web server as frontend increases the flexibility of the analyzer since it is possible to use any operation system with a web browser and there is no need to install additional tools. The languages which are used for the interface are Php and Javascript. The recorded data is stored in an SQL database.

The information is separated into the frame types of the M-Bus [1], which increases the comprehension of the data. The data is normally sent with one of the two M-Bus frames. These frames will be dissected into its fields. Also, the CRC will be checked and highlighted. Figure 16 shows a screenshot from the capt2web interface with some packages. Each information is displayed with a time stamp which is always at the beginning. The time stamp is generated by the head unit and shows the time when the packet with the information was received by the head unit. The parts of the data link layer are displayed in blue and the application layer in pink. If a data segment is not a part of a frame, the data will be in grey. The data can also be exported from the web interface; for this, there is a csv-export page available where the complete data displayed is available as semicolon separated textual fields. A future feature of the interface will be the implementation of the application layer. So, e.g., the values or mean values of meters will be shown.

## VII. Summary and Outlook

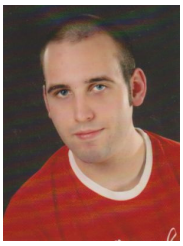
This project presents a bus analyzer for the M-Bus, which is a big step forward for finding network problems and errors since the detection of errors has always been a big problem until now. The analyzer is a flexible, easy, and low-cost tool. Furthermore, it is designed with a low-cost input circuit, which allows the user to connect the analyzer to any point of the M-Bus. The information itself will be regained by the software and the analog output value of the input circuit. Through this ability, it will be easy to implement other UART-based protocols into the analyzer, e.g. RS232.

The advantage of the web-based interface is that it does not depend on a specific operation system or additional software tools, which increases the flexibility. Also the Ethernet interface of the head unit improves the flexibility, since the analyzer runs as a stand-alone unit with access over a local area network. All plugs and jacks of the sniffer are removable, so it is easy to swap the tool to another test circuit.

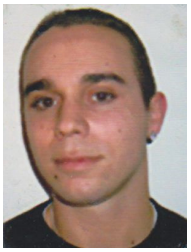
The future will bring some interesting features to the user. For example, the data will not only be presented as raw data, but the application layer data will be decomposed. This will allow the user to read everything in normal textual form without a second tool. Another future option is to unite the wireless and wired sniffer into one single unit to obtain a one-and-all solution for the M-Bus.

## REFERENCES

[1] M-Bus Documentation Rev 4.8, http://www.m-bus.com/, 11.09.2015.
[2] Open Metering Systems work group 4, http://oms-group.org/en/oms-group/working-groups/, 11.09.2015.
[3] SSV-Teleservice Gateway, http://www.ssv-comm.de/produkte/mgw865.php, 11.09.2015.
[4] SSV-DNP/9265, http://www.dilnetpc.com/dnp0096.htm, 11.09.2015.
[5] M-Bus Slave Transceiver, Texas Instruments TSS721a, http://www.ti.com/lit/ds/symlink/tss721a.pdf, 11.09.2015.
[6] Matlab Homepage, http://de.mathworks.com/products/matlab/, 11.09.2015.
[7] Serial Reference Manual, Steinbeis Transfer Center, page 21, 2014.



Thomas Matt studied Electrical and Information Engineering (EI) at the University of Applied Science Offenburg and received his B.Eng. degree in February 2015. Since that time he has been working as an engineer for the lab “Embedded Systems and Communication Electronics” at University of Applied Sciences Offenburg.



Manuel Schappacher studied Computer Engineering at the University of Applied Sciences, Furtwangen and received his Dipl.-Inform. degree in April 2009. After that, he continued to work as project engineer at Steinbeis Innovation Center Embedded Design and Networking (sizedn) mainly in the field of embedded wireless and wired communication, including simulation of networking protocols. Since 2014 he is with the laboratory of Embedded Systems and Communication Electronics at University of Applied Sciences Offenburg.

Prof. Dr.-Ing. Axel Sikora holds a diploma of Electrical Engineering and a diploma of Business Ad-ministration, both from Aachen Technical University. He has done a Ph.D. in Electrical Engineering at the Fraunhofer Institute of Microelectronics Circuits and Systems, Duisburg, with a thesis on SOI-technologies. 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 field of efficient, energy-aware, autonomous, and value-added algorithms and protocols for wired and wireless embedded communication. Dr. Sikora is author, co-author, editor and co-editor of several textbooks and numerous papers in the field of embedded design and wireless and wired networking, and head and member of numerous steering and program committees of international scientific conferences.

**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
rieger@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. Schick, (07531) 206-657
cschick@htwg-konstanz.de

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

**Hochschule Offenburg**
Prof. Dr. Sikora, (0781) 205-416
axel.sikora@hs-offenburg.de

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

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

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

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


**www.mpc.belwue.de**