MPC

MULTI PROJEKT CHIP GRUPPE
BADEN-WÜRTTEMBERG

Herausgeber: Hochschule Karlsruhe **Ausgabe:** 55 **ISSN** 1868-9221 **Workshop:** Karlsruhe Februar 2016

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.

Herausgeber:
Marc Ihle, Hochschule Karlsruhe, Moltkestr. 30, D-76133 Karlsruhe

Mitherausgeber (Peer Reviewer):
Jürgen Giehl, Hochschule Mannheim, Paul-Wittsack-Straße 10, D-68163 Mannheim

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

hochschule mannheim

Design und Verifikation eines On-Chip Tiefpass-Filters 3. Ordnung mit Operationsverstärkern in einer 0.35µm CMOS-Technologie

Ninja Koetsier, Frauke Bohinsky, Albrecht Zwick, Marcio Camoleze de Andrade, Bernd Vettermann, Jürgen Giehl

***Zusammenfassung*—Filterverstärkerschaltungen höherer Ordnung mithilfe von Operationsverstärkern wurden entwickelt. Ein solches Filter wurde in 3. Ordnung auf einem IC in 0.35 µm CMOS-Technologie realisiert und verifiziert. Es wurde die Möglichkeit zur Justierung der Filterfrequenz über die externen Biasströme der verwendeten Operationsverstärker vorgesehen und verifiziert.**
Das IC entstand in mehreren Diplom-, Master- und Bachelor-Arbeiten ([1]-[4]). Für den Entwurf und die Simulationen wurde das MENTOR IC Studio / Pyxis Layout verwendet. Zunächst wurde das Filter mit idealisierten Operationsverstärkern prinzipiell simuliert. Anschließend wurde ein entsprechender OP mit geringer Grenzfrequenz (1 kHz) entwickelt. Das Design wurde in ein Layout für den AMS 0.35 µm CMOS Prozess umgesetzt. Die Chipherstellung erfolgte über Europractice bei der Firma Austriamicrosystems (AMS). Zur Verifikation der gewonnenen Simulationswerte wurde eine vergleichende Messung der Schaltung durchgeführt. Hierbei galt es, das Verhalten des Operationsverstärkers und des Filters 3. Ordnung zu verifizieren.

Schlüsselwörter—CMOS, Operationsverstärker, OP, OP-Amp, Filter, Miller-Kompensation, Simulation, Messung.

I. Einleitung

Biosignale sind Körpersignale elektrischer, mechanischer, magnetischer oder chemischer Natur. Beispielsweise wird bei einem Elektrokardiogramm (EKG) die Herzmuskelaktivität gemessen. Die Signale sind nur einige Millivolt groß und bewegen sich in einem Frequenzbereich von 0 bis maximal 100 Hz [1].

Ein großes Problem bei der Erfassung dieser Signale sind Störungen aus der Umgebung. Lösung dafür ist, die schwachen Nutzsignale durch analoge Vorverarbeitung vom Millivolt- in den Voltbereich zu verstärken und Störsignale herauszufiltern [1].

Ninja Koetsier, Frauke Bohinsky, Albrecht Zwick, a.zwick@hs-mannheim.de, Marcio Camoleze de Andrade, m.camolezedeandrade@hs-mannheim.de, Bernd Vettermann, b.vettermann@hs-mannheim.de, Jürgen Giehl, j.giehl@hs-mannheim.de, Hochschule Mannheim, Paul-Wittsack-Str. 10, 68163 Mannheim

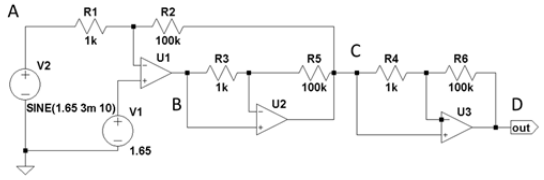


Abbildung 1: Schaltplan Filter 3. Ordnung mit Rückkopplungswiderständen.

Es wurde dafür eine aktive Filterschaltung entwickelt, welche die Tiefpass-Charakteristik von Operationsverstärkern nutzt. Damit lassen sich aktive Filter mit sehr niedriger Grenzfrequenz - welche üblicherweise große Filterkapazitäten benötigen - mit geringem Platzbedarf zum Beispiel auf einem IC herstellen. Mit Ausnahme der chipinternen Millerkapazität werden sonst keine weiteren externen Kondensatoren für den Schaltungsaufbau benötigt.

II. Aufgabenstellung

Es sollte ein Tiefpassfilter 3. Ordnung mit einer Grenzfrequenz von 10 Hz und einer Verstärkung von 100 (40 dB) entwickelt werden. Der Schaltplan des Filters 3. Ordnung ist in Abb. 1 gezeigt. Der Eingangsverstärker (U1) arbeitet in invertierender Konfiguration mit dem Verstärker (U2) in der Rückkopplung. Die Ausgangsstufe ist durch den Verstärker (U3) gegeben. Ziel dieses ersten Designs war es, die prinzipielle Funktion des Filters zu zeigen und Ansätze für Verbesserungen zu liefern. Der Einzel-OP sollte eine Lastkapazität von 50 pF treiben können und eine Transitfrequenz (Gain x Bandwidth) von 1 kHz aufweisen. Die Spezifikation des Filters ist in Tab. 1 zusammengefasst.

Aus dem erstellten Layoutdesign der Schaltung sollte ein Chip in einem QFN32 Gehäuse gefertigt und eine Testplatine erstellt werden [2].

Die Frequenzcharakteristik des einzelnen Operationsverstärkers und des Filters 3. Ordnung sollte ausgewertet werden und der Einfluss auf die Grenzfrequenz des Filters des von außen einstellbaren Biasstroms zur Versorgung der Komponenten der Schaltung sollte untersucht werden.

Tabelle 1: Spezifikation des Filters 3. Ordnung.

Parameter	Bezeichnung	Wert	Dim
DC-Verstärkung	A_{V0}	40	dB
Bandbreite	f_{-3dB}	10	Hz
Dämpfung	dA/df	-60	dB/Dekade
Max. kapazitive Last	$C_{Lastmax}$	50	pF

III. Filter 3. Ordnung

In diesem Abschnitt wird die Funktionsweise des Filters erläutert.

A. Funktionsprinzip

Ein Filter 3. Ordnung ließe sich prinzipiell durch Hintereinanderschaltung von 3 aktiven Tiefpässen erreichen, aber dann wäre der Übergangsbereich wesentlich flacher. Durch diese spezielle Anordnung wird dies vermieden. Die Übertragungsfunktion (Abb. 2) zeigt, dass der Amplituden-Frequenzgang horizontal verläuft und erst kurz vor der Grenzfrequenz scharf abknickt. Das Filter 3. Ordnung entspricht einem Butterworth-Filter.

Das Filter ist aus drei identischen Operationsverstärkern aufgebaut. Die Eingangsstufe ist als invertierende Verstärkerstufe realisiert. In der Rückkopplung der Eingangsstufe ist ein Verstärker eingefügt (Abb. 1), der die Übertragungsfunktion eines Tiefpasses aufweist. Die Übertragungsfunktion eines einzelnen invertierenden Verstärkers kann nach [8] berechnet werden und ist durch folgende Formel gegeben:

$$A_0 = \frac{R_2}{R_1} \cdot \frac{A_{op}}{A_{op} + \frac{R_1+R_2}{R_1}} \qquad (1)$$

Der einzelne Operationsverstärker hat dabei ein Tiefpassverhalten, das durch folgende Formel beschrieben werden kann:

$$A_{op}(f) = \frac{A_{V0}}{1 + jf/f_1} \qquad (2)$$

Hierbei ist $f_1 = f_T/A_{V0}$ die Grenzfrequenz.

Die Übertragungsfunktion der Eingangsstufe mit dem Verstärker in der Rückkopplung (Abb. 1 A-C) ist durch folgende Formel gegeben (mit R_1=R_3, R_2=R_5):

$$A_{AC} = \frac{R_2}{R_1} \cdot \frac{1}{1 + j\frac{f}{f_T(k_r)}\left(1 + j\frac{f}{f_T(k_r)}\right)} \; mit \; k_r = \frac{R_1}{R_1+R_2} \qquad (3)$$

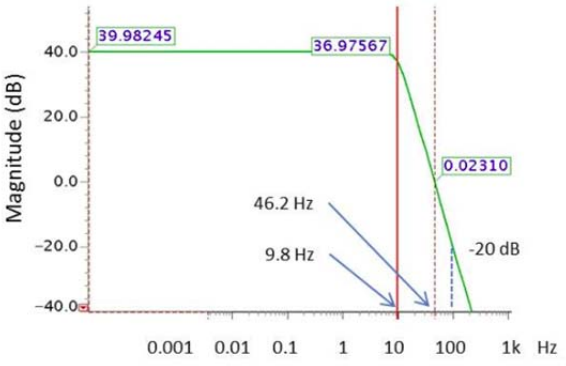


Abbildung 2:AC-Simulation des Filters 3. Ordnung mit idealen Bauteilen. Analyse von Grenz- und Transitfrequenz.

Die Übertragungsfunktion zwischen den Knoten C-D (gleich zu B-C) lautet (mit R_1=R_3=R_4, R_2=R_5=R_6):

$$A_{CD} = \frac{1}{1 + j\frac{f}{f_T}\frac{1}{(k_r)}} \qquad (4)$$

Die Hintereinanderschaltung der beiden Stufen ergibt dann als Gesamtübertragungsfunktion aus der Multiplikation der Formeln (3) und (4) für das Filter 3. Ordnung:

$$|A_{AD}| = |A| = \frac{R_2}{R_1} \frac{1}{\sqrt{1 + \left(\frac{f}{f_T(k_r)}\right)^6}} \qquad (5)$$

Die Tiefpasscharakteristik 3. Ordnung mit einer Dämpfung von -60dB/Dekade und einer -3 dB-Eckfrequenz von $f = f_T \cdot k_r$ ist daraus klar zu erkennen, ebenso wie die Butterworth-Charakteristik, welche konjugiert komplexe Pole bei $\pm f_T \cdot k_r$ zeigt [8].

B. Simulationen mit idealen Bauelementen

In diesen Simulationen wurde der ideale Operationsverstärker „opamp1“ aus der Mentor Pyxis Bibliothek „macrolib“ verwendet. Die Parameter der Verstärkung und der Bandbreite bestimmen das Tiefpass- und Verstärkungs-Verhalten (Gl. 2) des idealen Operationsverstärkers. Bei dieser Simulation wurde für die Verstärkung 100 000 (100 dB) und als Bandbreite 10 mHz gewählt. Daraus ergibt sich die Transitfrequenz von 1 kHz. Die minimale Zielspezifikation für den Operationsverstärker, welche sich aus Gleichung (2) ergibt, ist in Tabelle 2 aufgelistet.

Mit diesem Operationsverstärker wurde die Übertragungsfunktion des idealen Filters 3. Ordnung simuliert (Abb. 2). Es zeigt eine -3 dB-Grenzfrequenz bei 9.8 Hz und die Transitfrequenz bei 46 Hz. Die Verstärkung liegt bei knapp 40 dB und fällt mit -60 dB/Dekade.

Tabelle 2: Minimale Spezifikation des Operationsverstärkers.

Parameter	Bezeichnung	Wert	Dim
DC-Verstärkung	A_{V0}	100	dB
Transitfrequenz	f_T	1	kHz
Max. kapazitive Last	$C_{Lastmax}$	50	pF

IV. SCHALTUNGSREALISIERUNG

Die gesamte Schaltung setzt sich aus den Operationsverstärkern, Poly2-Widerständen und Kapazitäten zusammen. Bei letzteren wurde C_{stack} verwendet, bei der Gate-, Poly/Poly- und Metallkapazitäten platzsparend übereinander angeordnet sind („stacked“). Zur Versorgung der Operationsverstärker (Abb. 3) wird eine Strombank mit Stromquellen und Stromspiegeln mit MOS-Dioden am Eingang eingesetzt (Abb. 4).

A. OP

Der Operationsverstärker wurde als zweistufiger Verstärker mit kaskodierter PMOS-Stromquelle und –Differenzpaar in der Eingangsstufe und einem Eintransistorverstärker mit aktivem NMOS-Transistor und einer einfachen PMOS-Stromquelle in der Ausgangsstufe realisiert (Abb. 3) [1]. Die Parameter zur Handdimensionierung sind in Tabelle 3 zusammengefasst.

Die Differenzeingangsstufe wurde auf geringes g_m dimensioniert, damit die Verstärkung und somit die Transitfrequenz des Filters möglichst niedrig sind und eine kleinstmögliche Millerkapazität zur Einstellung der Transitfrequenz der OPs verwendet werden kann. Damit soll entsprechende Chipfläche gespart werden. Die Millerkapazität zur Kompensation liegt zwischen den Pins „vout“ und „Ckomp“ und ist in Abb. 3 nicht eingezeichnet. Um die Transitfrequenz so gering zu halten, musste dennoch eine Kompensationskapazität von 250 pF verwendet werden. Die Nullstelle der Miller-Kompensation [9]:

$$f_z = \frac{g_{m6}}{2\,\pi\,C_{Miller}} \qquad (6)$$

ergibt sich mit den Parametern aus Tabelle 3 zu 32 kHz und liegt somit ausreichend hoch in der Frequenz.

Der Tailstrom in der Eingangsstufe wurde auf 1 µA gesetzt. Mit der Dimensionierung der Transistoren wie in Abb. 3 gezeigt, liegt die effektive Gatespannung der Eingangstransistoren damit bei 0.5 V und die Steilheit g_m bei 2 µA/V. Die Transistoren wurden entsprechend

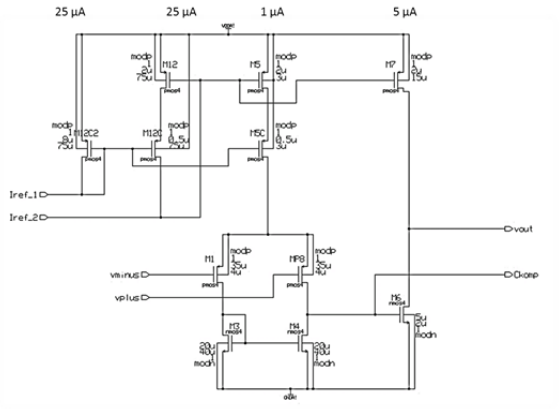


Abbildung 3:Innerer Aufbau des OPs mit dimensionierten Transistoren und Strömen.

Tabelle 3: Auszug Simulationsparameter der Transistoren [5].

Parameter	NMOS	PMOS	Dimension
Modell	modn	modp	
V_{sup}	3.3	3.3	V
V_{tsat} *(300 K)*	0.5	-0.7	V
K_p *(300 K)*	100	35	µA/V²
C_{ox}	4.5	4.5	fF/µm²
A_{VT}	9.5	14.5	mVµm
U_E *(l=1 µm)*	40	40	V

groß dimensioniert, damit deren Offset möglichst gering ausfällt. Dieser wird im Wesentlichen durch Schwellspannungsunterschiede hervorgerufen. Der Einfluss der anderen Parameter wurde durch Common Centroid Auslage minimiert. Der Offset vom Eingangsverstärker U1 erscheint mit der Verstärkung von 100 am Ausgang und würde die gesamte Schaltung in ihrem Betrieb stören. Mit der verwendeten Dimensionierung berechnet man mit dem 1-σ Wert der statistischen Schwellspannungsverteilung (A_{VT}) in Tabelle 3 einen 4-σ Offset-Wert von ±4.9 mV [9].

Die Ausgangsstufe wurde so dimensioniert, dass die Slew Rate, die durch den Tailstrom und die Millerkapazität eingangsseitig limitiert ist, ausgangsseitig nicht unterschritten wird – deshalb wurde ein Strom von 5µA für die Class-A-Stufe gewählt. Die Ausgangsstufe des 3. Verstärkers sollte zunächst nur den Tastkopf des Oszilloskops treiben können (R_{in} = *1 MΩ*). Die Dimensionierung der PMOS-Stromquelle und des NMOS-Transistors ergab sich aus der Forderung, dass die effektive Gatespannung 0.2 V sein sollte, was zu einem theoretischen Aussteuerbereich zwischen $0.2\ V \leq Vout \leq 3.1\ V$ bei $V_{DD} = 3.3\ V$ führt und zusammen mit dem zu erwartenden Offset Eingangssig-

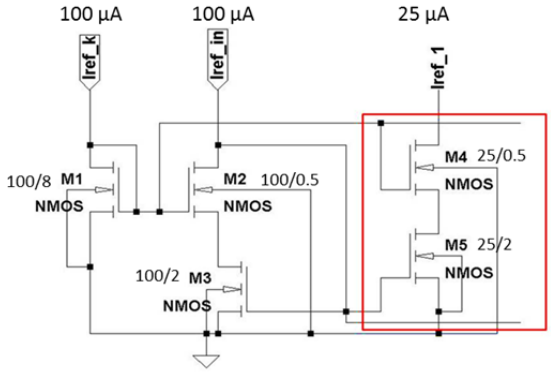


Abbildung 4:Aufbau Strombank Strömen und Dimensionierung der Transistoren.

nale bis zu 10 mV zulässt. Die Referenzspannung wird dazu auf $1.65\ V = (V_{DD}\text{-}V_{SS})/2$ gelegt.

Der Ausgangswiderstand in Rückkopplung sollte weniger als 10 % des geringsten Lastwiderstandes betragen, so dass nicht mehr als 1 dB an Verstärkung durch Spannungsteilung verloren geht. Dieser Punkt wird weiter unten noch diskutiert. Die Slew Rate sollte so groß sein, dass bis zur Transistfrequenz des Filters noch ein Sinus unverzerrt übertragen wird. Die angegebene Slew Rate beträgt dazu das 10-fache der benötigten Signaländerung.

B. Strombank

Die Strombank versorgt die Operationsverstärker mit den Biasströmen und ist in Abb. 4 gezeigt.

Jeder Operationsverstärker hat zwei Stromeingänge – einen für die Stromquellen (Iref_2) und einen für die Kaskodenspannung (Iref_1) (Abb. 3).

Versorgt wird die Strombank extern mit zwei 100 µA Strömen an den Eingängen Iref_k und Iref_in. Intern werden dann für jeden Operationsverstärker die benötigten 25 µA Biasstrom über Stromspiegel generiert. Je nachdem wie viele Operationsverstärker mit Biasstrom versorgt werden sollen, kann der rechts markierte Block der Stromquelle mehrfach parallel dazu geschaltet werden (Abb. 4). Im OP werden die 25 µA nochmals auf 1 µA Tailstrom für die Differenzstufe und 5 µA für die zweite Stufe gespiegelt. Die Strombank war bereits vorhanden und wurde bewusst nicht auf minimale Stromaufnahme optimiert, damit die externen Referenzströme präziser und leichter eingespeist werden können. Der einzelne OP hat in dieser Konfiguration eine Stromaufnahme von 6 µA bei externer Einspeisung von 100 µA. Die Stromaufnahme skaliert direkt mit dem externen Referenzstrom.

Die Dimensionierung der Strombank kann Abb. 4 entnommen werden. Diese wurde so ausgeführt, dass die effektive Gatespannung der Stromquellen 0.2 V beträgt, die der Kaskoden 0.1 V.

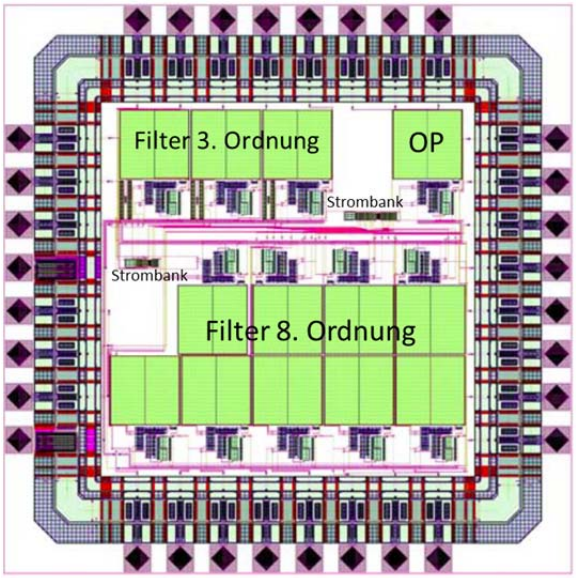


Abbildung 5: Finaler Entwurf mit Filter 3. Ordnung (oben links), Einzel-Operationsverstärker (oben rechts) und Filter 8. Ordnung.

C. Layout

Das Layout (Abb. 5) wurde über mehrere Arbeiten [1][1][3][3][4] entwickelt. Bei der Erstellung der Layouts wurden folgende Punkte beachtet:

Alle Bauteile sollten möglichst gleichartig in ihrem Verhalten sein. Das bedeutet, diese Bauteile sollten so angeordnet werden, dass sie alle gleichermaßen auf Veränderungen, zum Beispiel in der Temperatur, Prozessschwankungen oder mechanischen Stress reagieren. Diese Anordnung der Transistoren wird „Matching“ (Anpassen) genannt. Das Anpassen wird in der Common Centroid Anordnung realisiert [1][1].

In Abb. 5 ist das Layout des gesamten IC gezeigt. Die quadratischen großen Flächen sind die Kompensationskapazitäten, welche den größten Teil einnehmen. Links oben ist das Filter 3. Ordnung mit drei Operationsverstärkern erkennbar. Oben rechts befindet sich ein einzelner Operationsverstärker. Ein Filter 8. Ordnung, bestehend aus 8 Operationsverstärkern, ist ebenfalls auf dem IC untergebracht, wird aber hier nicht besprochen. Es befinden sich außerdem zwei Strombänke auf dem IC. Die gesamte Fläche des in Abb. 5 gezeigten ICs inklusiver der Pads beträgt ca. 2x2 mm². Das Filter 3. Ordnung nimmt dabei ca. 820x480 µm², der Operationsverstärker ca. 500x260 µm² und das Filter 8. Ordnung ca. 1250x780 µm² ein.

V. Messungen im Vergleich mit den Simulationen

Durch die Messungen am Chip wurde der Vergleich zu den simulierten Werten hergestellt. Dabei interes-

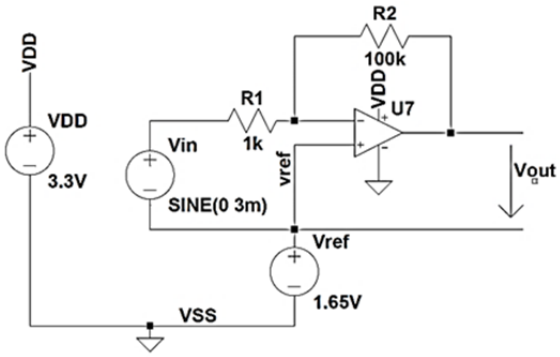


Abbildung 6: Testbench zum OP als invertierender Verstärker.

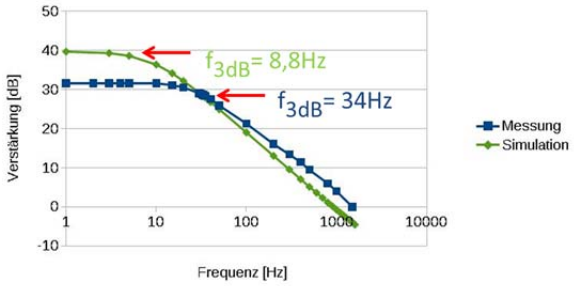


Abbildung 7: OP als invertierender Verstärker ohne Belastung im Vergleich.

sierte im Wesentlichen das Frequenzverhaltenen des Operationsverstärkers und des Filters 3. Ordnung.

A. OP

Der Einzeloperationsverstärker (Layout Abb. 5 oben rechts) wird mit externen Widerständen invertierend beschaltet (R_1=*1 kΩ,* R_2=*100 kΩ*). Die Verstärkung beträgt somit 100. Das Eingangssignal ist ein Sinussignal mit 3 mV Amplitude. Der Frequenzbereich wird von 1 Hz bis 2 kHz durchlaufen. Die Testbench für Simulation und Messung ist in Abb. 6 zu sehen. In der Messung wurde das Signal gegenüber der Referenzspannung von 1.65 V mit einem Tastkopf mit 1 MΩ Eingangswiderstand gemessen.

Die Messung (Abb. 7) zeigt das Tiefpassverhalten des Operationsverstärkers mit einer -3 dB-Grenzfrequenz bei 34 Hz und einer Transitfrequenz bei 1500 Hz. Die DC-Verstärkung beträgt 31 dB, pro Dekade fällt die Übertragungsfunktion mit -20 dB. Im Vergleich ist ein -3 dB Verstärkungsabfall in der Simulation schon bei 8.8 Hz erreicht und die Transitfrequenz bei 919 Hz. Die Änderung der Eckfrequenz lässt sich aus der verringerten Verstärkung erklären. Auffällig ist besonders die verringerte Verstärkung gegenüber der Simulation. Die Transitfrequenzen zeigen eine relativ gute Übereinstimmung.

B. Filter 3. Ordnung

1) Stromaufnahme

Zunächst wurde die Stromaufnahme an VDD festgestellt. Laut Messung werden 235 µA aufgenommen, im Vergleich zeigt die Simulation einen Wert von 220 µA. Messung und Simulation stimmen also in ihren Werten mit leichten Toleranzen überein.

2) Grenzfrequenz

Das Filter 3. Ordnung wurde in einer weiteren Messung auf seine Grenzfrequenzen genauer analysiert. Abb. 1 zeigt die verwendete Testbench. Als Eingangssignal lag ein Sinussignal mit einer Amplitude von 3mV an. Die Signalfrequenzen wurden von 1 Hz bis 100 Hz durchlaufen. Der Signaloffset lag bei 1.65 V und definierte damit die analoge Masse. Die Ausgangsspannung wurde gegenüber der analogen Masse gemessen.

In der Messung ist der -3 dB Verstärkungsabfall bei 19 Hz erreicht. Es zeigen sich zunächst ein schneller Abfall der Übertragungsfunktion und das Erreichen der Transitfrequenz bei 70 Hz. (Abb. 8). Es liegt eine DC-Verstärkung von 30 dB vor. Die Schaltung zeigt das Tiefpassverhalten eines Butterworth-Filters. Die Steigung beträgt in der Messung im Bereich 20 Hz bis ca. 50 Hz -60 dB/Dekade, flacht aber dann ab.

Die Simulation im Vergleich zeigt die Grenzfrequenz bei 9 Hz, die Transitfrequenz bei 45 Hz und eine Verstärkung von 40 dB (Abb. 8). Pro Dekade fällt der Frequenzgang um -60 dB.

3) Grenzfrequenz bei Stromvariation

Die Grenzfrequenz des Filters kann durch eine Variation des externen, in die Strombank eingespeisten Referenzstromes, verändert werden. Damit soll eine Abstimmung des Filters ermöglicht werden. Generell ist zu erwarten, dass bei Stromänderung in der Eingangsstufe (Tailstrom) sich das g_m im Sättigungsbetrieb der Transistoren und damit auch die Transitfrequenz des OPs und nach Gln. 2 und 5 respektive des Filters proportional zur Wurzel des Stromes ändert.

Durch eine Stromerhöhung auf 150 µA jeweils an den Eingängen Iref_k und Iref_in der Strombank zeigt die Messung eine Vergrößerung der Grenzfrequenz von 19 Hz auf 24 Hz an. Die Transitfrequenz steigt auf 80 Hz und die Verstärkung bleibt bei 30 dB (Abb. 9). In der Simulation liegt die Grenzfrequenz bei 11 Hz und die Transitfrequenz beträgt 57 Hz.

Die Messung zeigt deutlich eine Verkleinerung der Grenzfrequenz bei Stromsenkung. Bei 50 µA Refe-

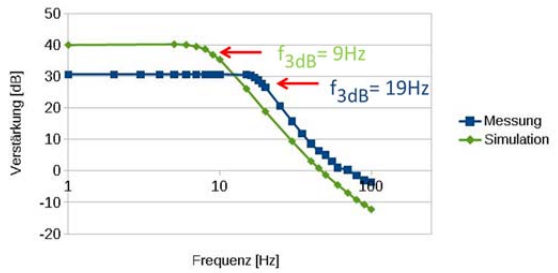


Abbildung 8: Filter 3. Ordnung mit 100 µA Stromeinspeisung. Vergleich Messung und Simulation.

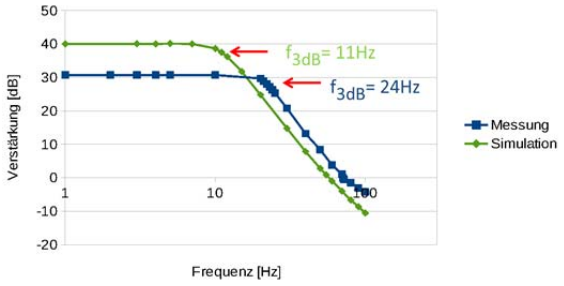


Abbildung 9: Filter 3. Ordnung mit 150 µA Stromeinspeisung. Vergleich Messung und Simulation.

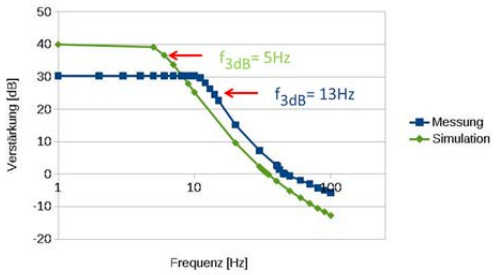


Abbildung 10: Filter 3. Ordnung mit 50 µA Stromeinspeisung. Vergleich Messung und Simulation.

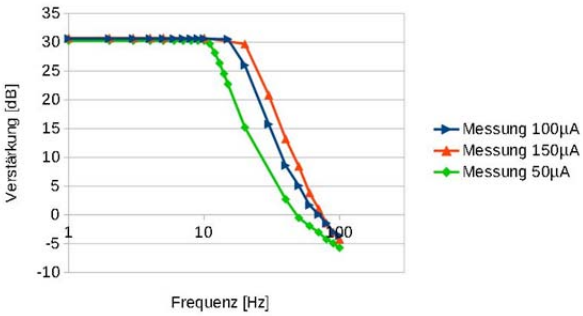


Abbildung 11: Filter 3. Ordnung . Zusammenfassung der Messergebnisse für verschiedene Biasströme.

renzstrom ist die Grenzfrequenz auf 13 Hz verringert und die Transitfrequenz liegt bei 60 Hz. (Abb. 10).

Im Vergleich ist die Grenzfrequenz in der Simulation bei 5 Hz erreicht und die Transitfrequenz beträgt 34 Hz. Die Verstärkung beträgt in der Messung 30 dB und in der Simulation 40 dB.

In Abbildung 11 sind die Messergebnisse für drei verschiedene Biasströme nochmals im Vergleich dargestellt.

In Tabelle 4 sind die Grenzfrequenzen der Messung und der Simulation im Vergleich bei drei verschiedenen Biasströmen angegeben.

Tabelle 4: Vergleich Grenzfrequenz Messung und Simulation Filter 3. Ordnung.

Strom Iref_in =Iref_k	-3dB-Grenzfrequenz	
	Messung	Simulation
150 µA	24 Hz	11 Hz
100 µA	19 Hz	9 Hz
50 µA	13 Hz	5 Hz

VI. Bewertung Messung und Simulation

Die Analyse des invertierend beschalteten unbelasteten Operationsverstärkers zeigt einen Unterschied in der -3dB-Grenzfrequenz von Messung zu Simulation in einer Größenordnung von Faktor 2. Allerdings stimmen die Transitfrequenz und damit der abfallende Teil der Übertragungsfunktion besser überein.

Erklärung für diese Frequenzunterschiede zwischen Messung und Simulation ist der Ausgangswiderstand des einzelnen rückgekoppelten Operationsverstärkers (OTA) der aus der Simulation mit 2.5 kΩ für Frequenzen kleiner 1 Hz ermittelt wurde. Der Ausgangswiderstand steigt dann im Bereich 1 Hz bis 100 Hz auf ca. 17 kΩ an. Die Belastung der Verstärker U1 und U2 (Abb. 1) mit 1 kΩ ist damit zu groß, so dass deren Ausgangsspannung heruntergeteilt wird. Damit ist die deutlich geringere Verstärkung in der Übertragungsfunktion erklärt. Unklar ist, warum die Simulation des kompletten Filters dies nicht korrekt widerspiegelt, obwohl der Ausgangswiderstand alleine realistisch simuliert werden kann.

Die -3 dB-Grenzfrequenz ist beim Filter 3. Ordnung in der Messung um den Faktor 2 größer als in der Simulation. Der Unterschied der Transitfrequenz beträgt von Messung zur Simulation nur ca. 20 Hz. Generell ist die Transitfrequenz genauer bestimmbar, da sie von der Transkonduktanz g_m der Transistoren der Differenzeingangsstufe und der Kompensationskapazität abhängt. Die Grenzfrequenz hängt deutlich von der Verstärkung ab und ist somit generell bei der vorliegenden geringeren Verstärkung zu höheren Frequenzen verschoben.

Tabelle 5: Spezifikation des realisierten Operationsverstärkers.

Parameter	Bezeichnung	Wert	Dim
DC-Verstärkung	A_{V0}	100	dB
Transitfrequenz	f_T	1	kHz
Max. kapazitive Last	$C_{Lastmax}$	50	pF
Tailstrom	I_{tail}	1	µA
Kapazität z. Millerkompensation	C_{Miller}	250	pF
Slew Rate	SR	4	V/ms
Max. Laststrom	$I_{Lastmax}$	5	µA
Nullstelle der Miller-Komp.	f_z	32	kHz
Ausgangswiderstand in Rückkop.	R_{out}	2.7	kΩ

Tabelle 6: Spezifikation des realisierten Filters.

Parameter	Bezeichnung	Wert	Dim
DC-Verstärkung	A_{V0}	31	dB
Transitfrequenz	f_T	90	Hz
Bandbreite	f_{-3dB}	10	Hz
Dämpfung	dA/df	-60	dB/Dekade
Rauschen (0.1 Hz - 10 kHz)	V_{rms}	320	µV

VII. Fazit

Die prinzipielle Funktionsfähigkeit des Filters konnte gezeigt werden.

In der Simulation des gesamten Filters wurde der Einfluss der zu großen Belastung der Verstärker nicht richtig wiedergegeben, obwohl der Ausgangswiderstand selbst realistisch simulierbar ist. Generell sind die Verstärker mit zu hohem Ausgangswiderstand ausgelegt, so dass die gewünschte Verstärkung mit der vorliegenden Auslegung der Widerstände nicht erreicht wird. Zusätzlich konnte bei den Messungen der Eingangspegel nicht weiter verringert werden, so dass die Ausgangsamplitude aufgrund der zu geringen Treiberfähigkeit der zweiten Stufe des OPs nicht erreicht werden kann.

Durch Variation der Referenzströme kann die Grenzfrequenz des Filters eingestellt werden. Bei Stromerhöhung steigt diese und bei Stromsenkung wird eine Verkleinerung der Grenzfrequenz erreicht.

Es zeigte sich bereits bei den Simulationen, dass das Filter 3. Ordnung mit eingebauter Verstärkung sehr stark durch Offsets der Verstärker beeinflusst wird. Die Eingangstransistoren wurden deshalb großflächig ausgelegt und zeigen in der Monte-Carlo-Simulation einen 4σ-Wert von ±4 mV. Ebenso könnte kritisch werden, dass die verwendeten Verstärker nur einen Eingangsspannungsbereich von 0 - 2.2 V aufweisen. Hier könnten Rail-to-Rail-Verstärker Abhilfe schaffen. Das Rauschen am Ausgang des Filters wurde aus der Simulation bestimmt und beträgt 320 µVrms in einem Frequenzbereich 0.1 Hz – 10 kHz. Den Hauptbeitrag liefern die Transistoren M3/4 (Stromspiegel in der Differenzeingangsstufe zur differential-to-single-ended Konversion) des Eingangsverstärkers U1 und die Ausgangstransistoren des Verstärkers U3. Bei der geringen Eckfrequenz des Filters hat das 1/f-Rauschen der MOSFETs einen großen Einfluss.

Danksagung

Die Autoren bedanken sich bei Irina Ohm. Im Rahmen ihrer Masterarbeit führte sie erste Untersuchungen und Designs des Filter-ICs durch. Ebenso bedanken sie sich bei Paul Kamou Fansi, er führte im Rahmen seiner Masterthesis das Schaltungsdesign und ein erstes Layout des verwendeten OPs durch. Weiterhin bedanken sich die Autoren bei Andreas Arnold, im Rahmen einer Studienarbeit erstellte er die Strombank des Filters.

Die Autoren bedanken sich insbesondere für die finanzielle Unterstützung der Chipherstellung bei der Multi Projekt Chip Gruppe.

LITERATURVERZEICHNIS

[1] P. Fansi, Masterarbeit, „Entwicklung eines Operationsverstärkers in 0.35µm CMOS-Technologie zur Anwendung in Filtern höherer Ordnung“, Hochschule Mannheim 2011. F. Bohinsky, Bachelorarbeit, „Layoutdesign und Simulation in 0.35µm CMOS-Technologie - On-Chip-Tiefpassfilter 3. und 8.Ordnung”, Hochschule Mannheim 2013.

[2] N. Koetsier; Masterarbeit, “Inbetriebnahme eines On-Chip-Tiefpass-Filters 3. und 8. Ordnung und Entwurf eines Bandpass-Filters in 0.35 µm CMOS-Technologie”, Hochschule Mannheim 2015

[3] I. Ohm, Masterarbeit „Design eines Filters 3.-8. Ordnung mit Operationsverstärkern in einer 0.35µm CMOS-Technologie”, Hochschule Mannheim 2010.

[4] A. Arnold, Studienarbeit „Chipentwurf eines Filter ICs”, Hochschule Mannheim 2012.

[5] N. Koetsier, Studienarbeit, „ Layout und Verifikation in 0.35µm CMOS Technologie mit dem Mentor Pyxis-System”, Hochschule Mannheim 2014.

[6] Jürgen Giehl, Bernd Vettermann, „Laboranleitung Simulation in EIS2 im Institut für Entwurf integrierter Schaltkreise”, Hochschule Mannheim 2014.

[7] U. Tietze, Ch. Schenk, „Halbleiterschaltungstechnik“, 11. Auflage, S. 479-488, Springer Berlin, 1999

[8] Sansen, Willy M.C., „Analog Design Essentials“, Springer Verlag, 2006.

Ninja Koetsier erhielt den akademischen Grad des Bachelor of Science von der Hochschule Mannheim im Jahr 2013. Aktuell arbeitet sie an ihrer Masterarbeit über die Verifikation eines on-Chip Filters 3. und 8. Ordnung und an einem Entwurf eines Bandpassfilters in 0.35µm CMOS-Technologie.

Frauke Bohinsky erhielt den akademischen Grad des Bachelor of Science von der Hochschule Mannheim 2013. Im Rahmen ihrer Bachelorthesis erstellte sie ein Toplevel-Layout mit parasitärer Extraktion und abschließenden Simulationen.

Albrecht Zwick lehrt seit 1974 als Professor an der Hochschule Mannheim und hält auch nach seiner Pensionierung 2011 noch Vorlesungen auf dem Gebiet der analogen Schaltungstechnik mit Spezialisierung auf rauscharme Schaltungen.

Marcio Camoleze de Andrade erhielt den akademischen Grad des Diplomingenieurs von der Universidade Federal de Fluminese in Brasilien im Jahr 2011 und den Master of Sience an der Hochschule Mannheim im Jahr 2014. Aktuell arbeitet er dort als wissenschaftlicher Mitarbeiter am Institut für Entwurf integrierter Schaltkreise im Rahmen eines BMBF-Projektes zur Realisierung von Mikroelektrodenarrays auf flexiblem Silizium.

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

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

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

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

Effiziente Embedded-Multicore-Programmierung

Automatische Parallelisierung von Scilab/MATLAB-Anwendungen

Oliver Oey, Michael Rückauer, Timo Stripf, emmtrix Technologies GmbH

***Zusammenfassung*— Durch immer weiter steigende Performanzanforderungen wird in immer mehr Bereichen anstelle von Einkernprozessoren auf Mehrkernprozessoren gesetzt. Dieser Wechsel ist im Bereich von Desktop-PCs oder Smartphones bereits vollzogen, im Bereich der eingebetteten Systeme ist der Umbruch jedoch noch im Gange. Durch die parallele Ausführung von Programmen kann sowohl die Performanz gesteigert als auch die Leistungsaufnahme reduziert werden. Bis heute verursacht die parallele Programmierung jedoch einen hohen Zeit- und Kostenaufwand und erfordert spezielles Wissen über die Zielsysteme. Innerhalb des ALMA-EU-Projekts hat ein Konsortium aus Forschung und Industrie eine Werkzeugkette entwickelt, die die parallele Programmierung erheblich vereinfacht. Mittels automatischer Parallelisierung wird sequentieller Scilab/MATLAB-Code für eingebettete Multicore-Prozessoren parallelisiert. Dadurch kann nicht nur die aufwändige manuelle Parallelisierung eingespart, sondern auch der Code auf verschiedenen Prozessoren wiederverwendet werden. Die Ergebnisse des Forschungsprojekts werden in dem EU-Projekt ARGO sowie im Startup emmtrix Technologies noch weiter vertieft.**

Schlüsselwörter—automatische Parallelisierung, MATLAB, heterogene Mehrkernarchitekturen, Echtzeit-Systeme,

I. Einleitung

Laut Studien [1] ist die Programmierung von eingebetteten Mehrkernsystemen 4,5 mal so teuer, dauert 25% länger und benötigt 3 mal so viele Software-Ingenieure wie die Programmierung von Einkernsystemen. Um diesen Mehraufwand der Entwicklung paralleler Software zu reduzieren, wurde im EU-Projekt ALMA eine automatische Parallelisierung und Codeerzeugung aus arraybasierten Programmiersprachen untersucht. Zu diesen gehören MATLAB und ähnliche Sprachen wie Scilab, die sehr nah an der rein mathematischen Beschreibung sind und standardmäßig auf Matrizen als Datentyp arbeiten. Aus diesem Grund eignen sie sich auch gut für Anwender, die keinen großen Programmierhintergrund haben. Der Programmierer kann sich auf die Entwicklung des Algorithmus konzentrieren, während die Anpassungen an die Zielhardware automatisch durch die Werkzeugkette erledigt werden. Der zusätzliche Aufwand für die manuelle Parallelisierung entfällt, wodurch Entwicklungskosten und -zeit eingespart werden.

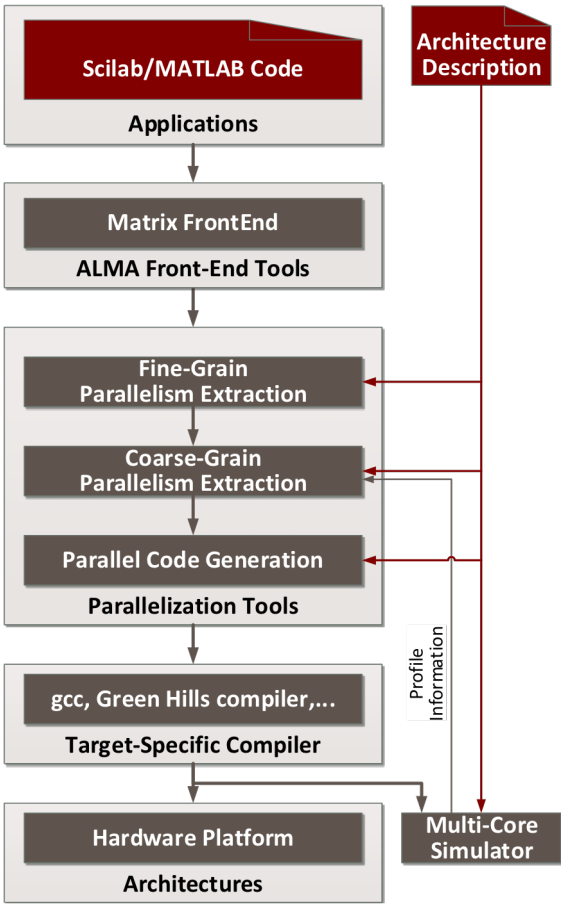


Abbildung 1: Übersicht über die ALMA-Werkzeugkette.

II. ALMA Projektübersicht

Die ALMA-Werkzeugkette, wie sie in [2] vorgestellt wurde, ist in mehrere Teilkomponenten unterteilt, die

Oliver Oey, oliver.oey@emmtrix.com, Michael Rückauer, michael.rueckauer@emmtrix.com, Timo Stripf, timo.stripf@emmtrix.com, emmtrix Technologies GmbH, Engesserstr. 5, 76131 Karlsruhe

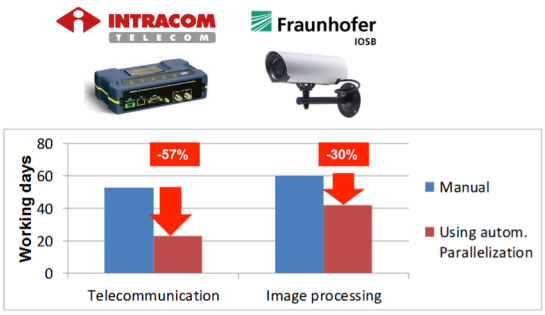


Abbildung 2: Ergebnisse des ALMA-Projekts.

in Abbildung 1 dargestellt werden. Als Eingangsdateien kommen die Anwendungen, die in Scilab/MATLAB geschrieben wurden, sowie eine abstrakte Beschreibung der Zielarchitektur zum Einsatz. Das Matrix FrontEnd wandelt den Eingangscode zunächst in einen sequentiellen C-Code um, der als Basis für die Parallelisierung verwendet wird. Die Parallelisierungswerkzeuge erzeugen optimierten Code für die Zielarchitekturen, indem der Code parallelisiert und an die Eigenheiten der Zielplattformen angepasst wird, wie sie in der Architekturbeschreibung hinterlegt sind. Das parallele Programm wird mit Hilfe eines Multicore-Simulators ausgeführt, der Informationen über die Laufzeit des Programms auf der Zielarchitektur ermittelt. Diese Informationen werden in weiteren Optimierungsschritten verwendet, um die Auslastung der Hardware zu verbessern und somit die Laufzeit des Programms zu reduzieren.

A. Evaluation

Zur Evaluation des Ansatzes kamen zwei Zielarchitekturen sowie zwei Testanwendungen zum Einsatz. Die Zielarchitekturen waren der wissenschaftliche Prozessor Kahrisma [3] sowie das X2014 System der Firma Recore Systems. Beide Plattformen setzen auf mehrere unabhängig arbeitende Kerne mit verteiltem Speicher. Die Zielanwendungen kamen zum einen aus der Bildverarbeitung vom Fraunhofer IOSB, bei der eine Objekterkennung mit Hilfe des SIFT-Algorithmus durchgeführt wurde, und zum anderen aus der Telekommunikation von Intracom Telecom, in dem Teile des WiMax Standards in Software ausgeführt werden.

B. ALMA Werkezugkette & Workflow

Scilab/MATLAB sind Skriptsprachen. Das bedeutet, dass die Befehle im Programm ohne vorherige Übersetzung von einem Interpreter ausgeführt werden. Dies ermöglicht es, dass Eigenschaften wie beispielsweise die Größe oder der Typ von Variablen erst zur Laufzeit festgelegt werden. Eine Umsetzung dieses Verhaltens in C ist zwar möglich, aber Aufgrund der Performanz und des Speicherverbrauchs nicht sinnvoll. Aus dem allgemeinen Matrix-Code wird zunächst C-Code erzeugt, der sich gut für die statische Analyse und die spätere Parallelisierung eignet. Dazu werden alle dynamischen Entscheidungen aufgelöst, so dass der statische Programmablauf besser analysiert werden kann. Im Falle von Variablen wird zunächst über die gesamte Laufzeit des Programms betrachtet, welcher Datentyp nötig ist, um alle Zahlen ohne Einschränkungen darstellen zu können. Im C-Code wird dieser Datentyp verwendet, um sowohl die dynamischen Entscheidungen zu reduzieren als auch den Speicherverbrauch zu minimieren. Ein weiterer Vorteil von Matrix-Code hinsichtlich der Parallelisierung ist das Fehlen von Pointern. Dies ermöglicht es, den Datenfluss in einem Programm eindeutig bestimmen zu können, was für die Analyse der nötigen Datentransfers zwischen den verschiedenen Kernen notwendig ist.

Die abstrakte Hardwarebeschreibung setzt auf eine im Projekt entwickelte Architektur-Beschreibungssprache (engl. architecture description language, ADL) [4]. Als Besonderheit wird eine Beschreibung auf verschiedenen Abstraktionsebenen unterstützt. So können Hardware-Module sowohl rein funktional als auch detailliert mit Angabe von zyklenakkuraten Instruktionen dargestellt werden. Auf diese Weise können sowohl Informationen, die zur Simulation der Hardware notwendig sind, als auch abstraktere Informationen, die für die Parallelisierungsentscheidung benötigt werden, dargestellt werden.

Um eine gute Performanz einer parallelisierten Anwendung zu erreichen, muss die Parallelisierung auf zwei unterschiedlichen Ebenen ansetzen: auf einer feinen und einer groben. Dabei zielt die feine auf eine Optimierung der Ausführung auf einem Kern ab und die grobe optimiert die gleichzeitige Ausführung auf mehreren Kernen. Durch diese Kombination kann die gegebene Hardware möglichst effizient ausgenutzt werden.

Die feingranulare Parallelisierungsextraktion (engl. fine-grain parallelism extraction) analysiert zunächst die benötigten Datentypen hinsichtlich eines Kompromisses aus effizienter Ausnutzung der Hardware und der benötigten Genauigkeit der Ergebnisse. Dies erlaubt es, die SIMD (engl. single instruction, multiple data) Einheiten der Architektur auszunutzen, indem beispielsweise vier 8-Bit-Additionen gleichzeitig ausgeführt werden anstelle einer 32-Bit-Addition. Wie in [5] dargestellt, wird außerdem ermittelt, ob Zahlen in einer Festkommadarstellung repräsentiert werden können. Der Einsatz erhöht die Performanz der Anwendung bei Ausführung auf einer Zielarchitektur, die keine Gleitkommaarithmetik unterstützt, hat aber Einfluss auf die Genauigkeit. Des Weiteren werden Schleifen transformiert, um Zugriffe auf Daten besser an die vorhandenen Caches anzupassen.

Die grobgranulare Parallelisierung, wie sie in [6] vorgestellt wurde, verteilt die Anwendung auf die einzelnen Kerne der Zielplattform. Ziel dieser Optimierung ist es, die Ausführungszeit des gesamten

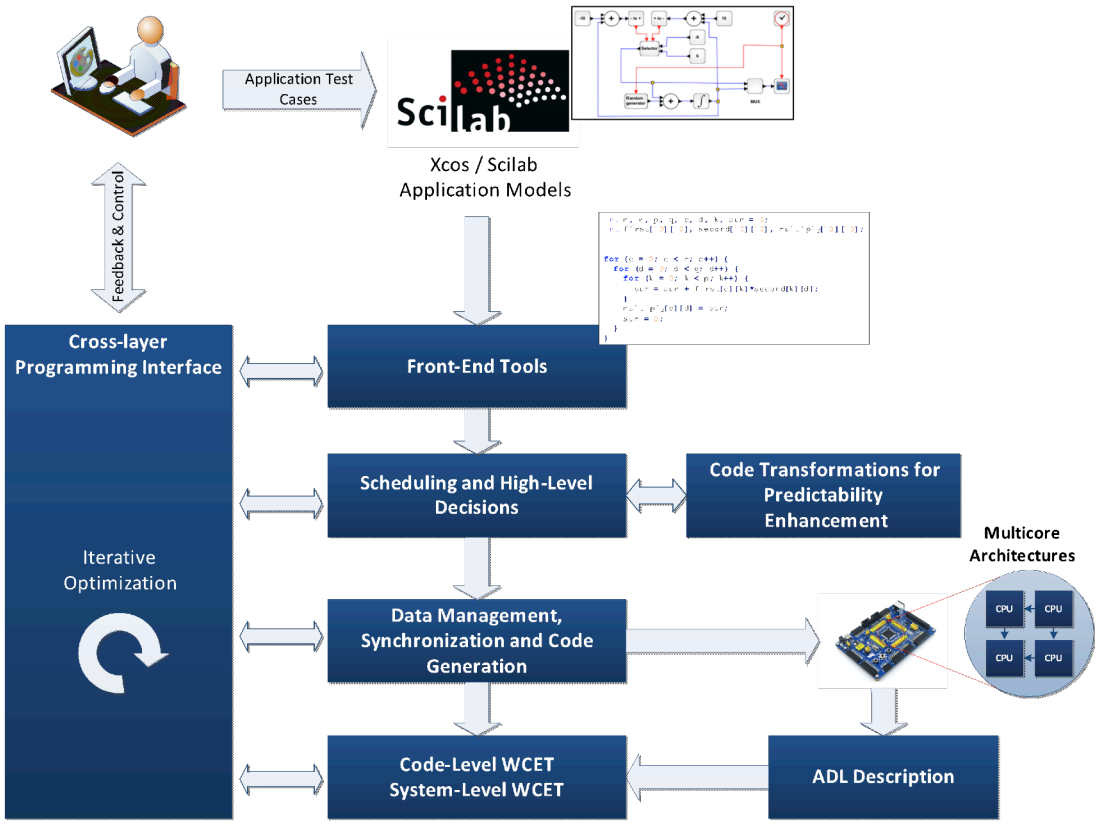


Abbildung 3 Übersicht über das ARGO Projekt

Programms zu reduzieren, indem möglichst viele parallele Recheneinheiten der Architektur gleichzeitig verwendet werden.

Die parallele Codegenerierung [6] erzeugt schließlich parallelen C-Code, der auf der Hardware oder den zugehörigen Simulatoren ausgeführt werden kann. Dazu erstellt die Codegenerierung separaten Quellcode für die einzelnen Kerne der Hardware-Plattform, indem Datenabhängigkeiten zwischen den einzelnen Prozessoren über Kommunikationsinstruktionen aufgelöst werden. Die Kommunikationssynthese achtet dabei auf eine Reduzierung der anfallenden Wartezeiten und ist auf Systeme mit verteiltem Speicher optimiert, aber nicht auf sie beschränkt. Für die Kommunikation können sowohl verbreitete Modelle wie MPI (message passing interface) als auch spezielle Funktionen der jeweiligen Plattformen ausgenutzt werden.

Der erzeugte Code kann automatisch instrumentiert werden, um Laufzeitinformationen mit Hilfe eines Simulators zu ermitteln. Diese Informationen werden verwendet, um die Parallelisierung und damit auch die Performanz iterativ zu verbessern [7]. Dazu fließen die Laufzeiten der einzelnen Codeteile und die Übertragungszeiten für Daten zurück in die grobgranulare Parallelisierung. Diese optimiert die Aufteilung mit dem Ziel einer besseren Auslastung der jeweiligen Zielarchitektur, um eine Verkürzung der Programmlaufzeit zu erreichen.

C. Ergebnisse

Die Werkzeugkette soll zwei Ziele erfüllen: zum einen soll die Leistungsfähigkeit von Mehrkernsystemen durch eine sinnvolle Parallelisierung der Anwendung ausgenutzt werden können. Zum anderen soll die Produktivität der Softwareentwicklung gesteigert werden, indem der Entwicklungsaufwand für parallele eingebettete Systeme reduziert wird.

In Abbildung 2 ist die im Projekt ermittelte Reduktion des Entwicklungsaufwands dargestellt. Verglichen wurde eine manuelle Umsetzung eines bestehenden Algorithmus in parallelen C-Code mit dem Einsatz des Parallelisierungswerkzeugs, das in ALMA entwickelt wurde. Bei den gewählten Beispielen konnten dabei Einsparungen von 30 bis 57 % in der Entwicklungszeit erreicht werden. Dabei konnte ein Speedup um den Faktor 2,8 beim Einsatz von vier Kernen erreicht werden.

Aus dem Projekt wurden zudem die folgenden Erkenntnisse gewonnen, die in weiteren Projekten berücksichtigt werden:

- Eine voll-automatische Parallelisierungs-Lösung ist nicht erstrebenswert, da der Benutzer keinen Einfluss auf das Ergebnis hat.

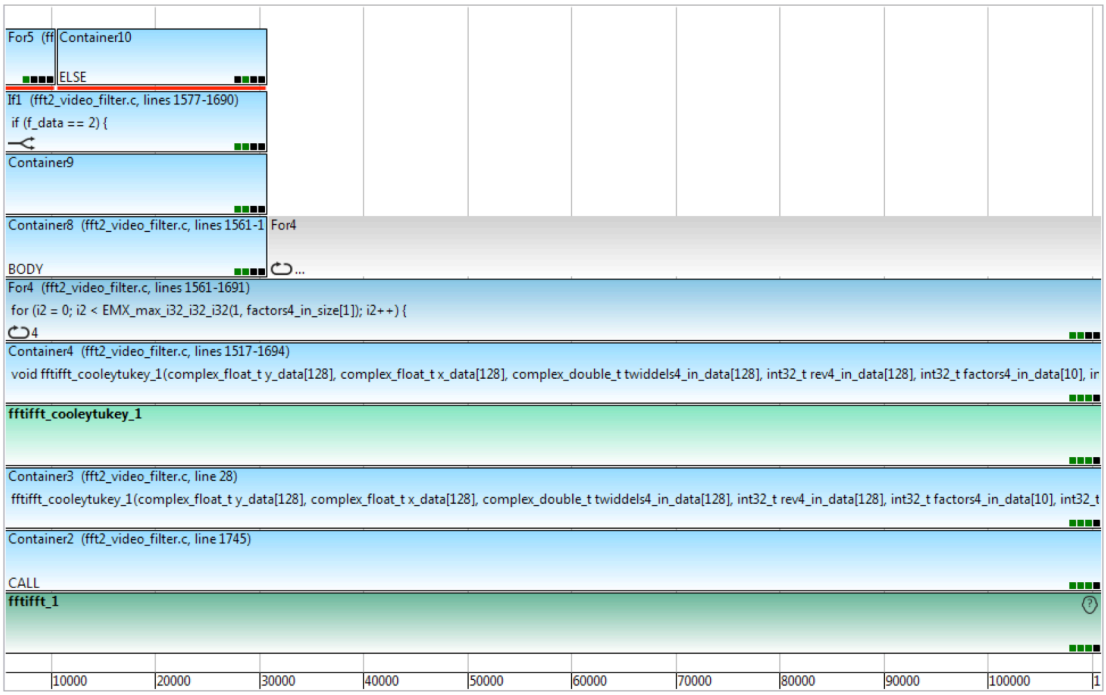


Abbildung 4 Hierarchische Darstellung einer Beispiel-Anwendung

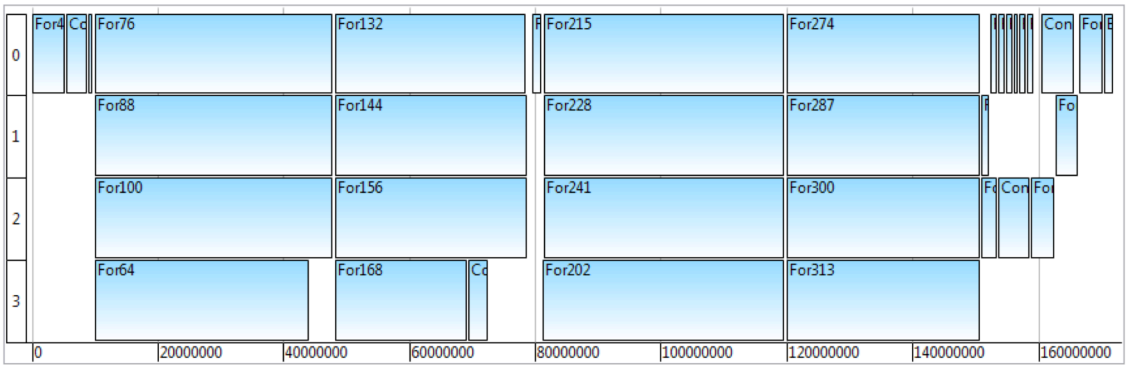


Abbildung 5: Scheduling Ansicht

- Der Nutzer muss das Ergebnis der Parallelisierung nachvollziehen können, d.h. welcher Teil des parallelen Programms welchem Teil des ursprünglichen Programms entspricht und welche Schritte die Toolchain vollzogen hat, um zum parallelen Programm zu kommen. Dazu gehört auch, dass die Codeerzeugung deterministisch arbeitet, d.h. dass sie für dasselbe Eingabeprogramm bei gleichen Parametern ein äquivalentes Ergebnis erzeugt.
- Der Benutzer muss die Kontrolle über den Parallelisierungsprozess haben, d.h. er muss selbst Parallelisierungsentscheidungen treffen können, etwa um anwendungsspezifisches Wissen in den Prozess einbeziehen zu können.

III. ARGO Projektübersicht

Das ARGO-Projekt ist ein neues EU-Projekt, das von Januar 2016 bis Dezember 2018 läuft und das die in ALMA erarbeiteten Konzepte noch weiterentwickelt. Ein Überblick über alle Teilgebiete des ARGO Projekts kann Abbildung 3 entnommen werden. Die Kernthemen sind:

- Benutzerfreundlichkeit: Den ALMA Parallelisierungsvorgang zu einer grafisch getriebenen Entwicklungsumgebung weiterzuentwickeln. Dies wird im Cross-Layer Programming Interface zusammengefasst.
- Echtzeit: Einen Hard- und Software übergreifenden Ansatz zur Berechnung der maximalen Laufzeit (engl. worst-case execution time, WCET) bereitzustellen, um die reaktive Natur von eingebetteten Anwendungen besser unter-

stützen zu können. Dies wird in ARGO durch Code-Transformationen, die die Vorhersagbarkeit erhöhen und durch WCET-Analysen auf Code- und System-Ebene realisiert.

- Heterogenität: Ausnutzung von heterogenen Ausführungseinheiten.
- Wiederverwendbarkeit: Produktivitätssteigerung durch Wiederverwendung von existierendem Code.
- Energieeffizienz: Steigerung der Energieeffizienz durch Ausnutzung von parallelen, heterogenen MPSoCs.

IV. EMMTRIX TECHNOLOGIES

Die in ALMA entwickelte Technologie wird in dem KIT-Spin-off „emmtrix Technologies“ weiter an die Bedürfnisse der Industrie angepasst. Im Vordergrund der Weiterentwicklung steht die Einführung einer interaktiven grafischen Oberfläche, um den Benutzer bei der Parallelisierung zu unterstützen. Dazu wird auf eine hierarchische Darstellung der Anwendung (Abbildung 4) sowie auf schnelle Rückmeldung über die Auswirkungen von möglichen und getroffenen Entscheidungen gesetzt. Beispielhaft ist in Abbildung 5 die Scheduling-Ansicht dargestellt, die einen schnellen Überblick über die Auslastung der einzelnen Prozessorkerne gibt. Die dargestellten Zeiten werden durch eine Mischung aus Profiling und statischer Analyse ermittelt, um möglichst schnell eine Rückmeldung liefern zu können, so dass der Benutzer viele Optionen in kurzer Zeit evaluieren kann.

Der erweiterte Workflow ist in Abbildung 6 dargestellt. In der ersten Phase wird der Eingangscode in sequentiellen C-Code übersetzt, der als Basis für Phase 2 - die Parallelisierung - verwendet wird. Der Code Generator setzt dabei auf das in ALMA entwickelte Matrix Frontend auf. Die Parallelisierung setzt zu großen Teilen auf die grob-granulare Parallelisierung. Die fein-granulare Parallelisierung kann abhängig von der Zielarchitektur eingesetzt werden. Der erzeugte Code kann an den verschiedenen Stellen sowohl in sequentieller als auch in paralleler Form auf dem Entwicklungssystem oder der Zielplattform getestet werden.

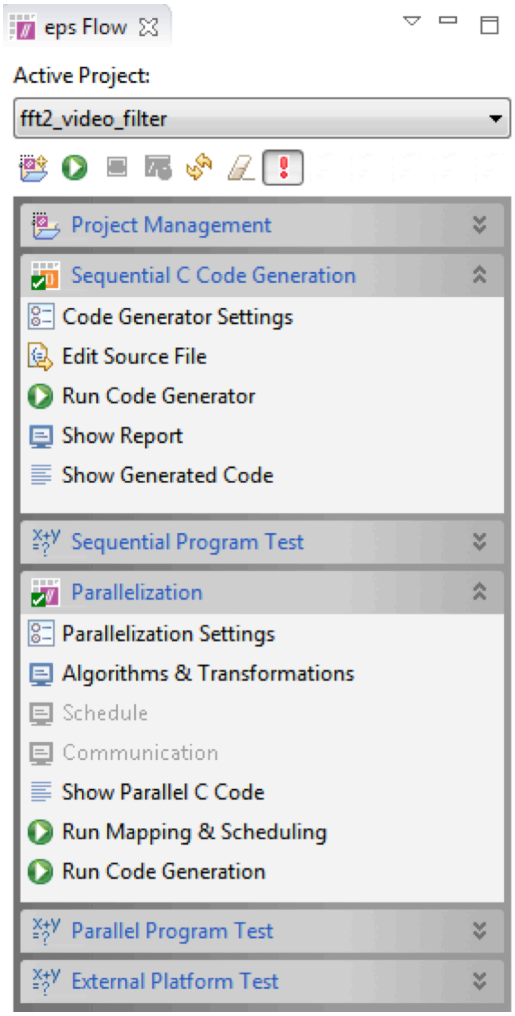


Abbildung 6 Workflow des emmtrix Parallel Studio

V. FAZIT

Die Programmierung von eingebetteten Mehrkernprozessoren ist ein Thema, das aufgrund steigender Anforderungen immer relevanter wird. Der im ALMA-Projekt erarbeitete Ansatz, die Programmierung in eine abstrakte Entwicklung des Algorithmus in einer arraybasierten Programmiersprache wie MATLAB oder Scilab und in eine hardwarenahe Parallelisierung aufzuteilen, hat sich als aussichtsreich herausgestellt. Um das Potential noch weiter auszunutzen, werden die Ergebnisse zum Einen im Projekt ARGO hinsichtlich ihrer Eignung für heterogene Echtzeitsysteme erweitert und zum Anderen vom KIT Spin-off emmtrix Technologies an die Anforderungen der Industrie angepasst.

DANKSAGUNG

ALMA: This work is co-funded by the European Union under the 7th Framework Programme under grant agreement ICT-287733.

Das Projekt emmtrix wird im Rahmen des EXIST-Forschungstransfers durch das Bundesministerium für Wirtschaft und Energie und den Europäischen Sozialfonds gefördert.

ARGO: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 688131.

LITERATURVERZEICHNIS

[1] „Next Generation Embedded Hardware Architectures: Driving Onset of Project Delays, Costs Overruns, and Software Development Challenges,“ VDC Research, September 2010.

[2] J. Becker, T. Stripf, O. Oey, M. Huebner, S. Derrien, D. Menard, O. Sentieys, G. Rauwerda, K. Sunesen, N. Kavvadias,

K. Masselos, G. Goulas, P. Alefragis, N. Voros, D. Kritharidis, N. Mitas and D. Goehringer, “From Scilab to High Performance Embedded Multicore Systems: The ALMA Approach,” in *Digital System Design (DSD), 2012 15th Euromicro Conference on*, 2012.

[3] R. Koenig, L. Bauer, T. Stripf, M. Shafique, W. Ahmed, J. Becker and J. Henkel, "KAHRISMA: A Novel Hypermorphic Reconfigurable-Instruction-Set Multi-grained-Array Architecture," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010, pp. 819-824.

[4] T. Bruckschloegl, O. Oey, M. Ruckauer, T. Stripf und J. Becker, „A Hierarchical Architecture Description for Flexible Multicore System Simulation,“ in *Parallel and Distributed Processing with Applications (ISPA), 2014 IEEE International Symposium on*, Milano, Italy, 2014.

[5] G. Deest, T. Yuki, O. Sentieys und S. Derrien, „Toward scalable source level accuracy analysis for floating-point to fixed-point conversion.,“ in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '14)*, Piscataway, NJ, USA, 2014.

[6] G. Goulas, C. Valouxis, P. Alefragis, N. Voros, O. Oey, T. Stripf, T. Bruckschloegl, J. Becker, C. Gogos, A. El Moussawi, M. Naullet und T. Yuki, „Coarse-Grain Optimization and Code Generation for Embedded Multicore Systems,“ in *Digital System Design (DSD), 2013 Euromicro Conference on* , 2013.

[7] J. Becker, T. Bruckschloegl, O. Oey, T. Stripf, G. Goulas, N. Raptis, C. Valouxis, P. Alefragis, N. Voros und C. Gogos, „Profile-Guided Compilation of Scilab Algorithms for Multiprocessor Systems,“ in *Reconfigurable Computing: Architectures, Tools, and Applications*, Springer International Publishing, 2014, pp. 330-336.

Dipl.-Ing. Oliver Oey hat ein Diplom in Elektrotechnik vom Karlsruher Institut für Technologie (KIT). Er ist ein Experte in Design und Programmierung von Networks-on-Chip für Multicore-Prozessoren sowie der parallelen Codegenerierung.

Dipl.-Inform. Michael Rückauer erhielt sein Diplom in Informatik im Jahr 2008 von der Universität Karlsruhe (TH). Er ist Experte für intuitives GUI-Design und den Entwurf und die Programmierung von Multicore-Prozessoren.

Dr.-Ing. Timo Stripf hat am KIT Informatik studiert und hat seinen Doktor in Elektrotechnik im Bereich der Compiler im Dezember 2013 bekommen. Seit Anfang 2012 hat er gemeinsam mit Prof. Jürgen Becker das ALMA EU-Projekt koordiniert.

hochschule aschaffenburg
university of applied sciences

Fusion von Fußgängererkennungen auf FPGAs

Sascha Hock, Michael Hahnle, Konrad Doll

***Zusammenfassung* — Viele Verfahren zur Erkennung von Objekten, beispielsweise Fußgängern, in Bildern benutzen das Sliding-Window-Prinzip. Die dabei entstehenden überlappenden Detektionen müssen anschließend zu einzelnen Objektdetektionen fusioniert werden. Des Weiteren müssen sporadisch auftauchende Falscherkennungen eliminiert werden. Diese Fusion von Detektionsergebnissen ist in der bildbasierten Objekterkennung ein wichtiger Verarbeitungsschritt und wird häufig auch als Non-Maximum Suppression (NMS) bezeichnet. Für praktische Anwendungen z.B. in Fahrerassistenz und Fußgängerschutzsystemen ist eine echtzeitfähige Implementierung erforderlich. In dieser Arbeit wird die erste, den Autoren bekannte, rein FPGA-basierte Implementierung vorgestellt. Sie ist in der Lage, die Erkennungsergebnisse mehrerer paralleler Skalenstufen in Echtzeit zu fusionieren. Auf einem Kintex-7 FPGA eines Xilinx Zynq SoCs ist damit eine Verarbeitung von mehr als 100 Bildern pro Sekunde mit einer Auflösung von 1280 x 720 Pixel möglich. Die Genauigkeit der FPGA-Implementierung ist dabei mit der einer auf *OpenCV* basierenden Softwareanwendung vergleichbar.**

***Schlüsselwörter* — Fußgängererkennung, Fusion, NMS, FPGA.**

I. Einleitung

A. Motivation

Die Bedeutung der bildbasierten Detektion von Personen, insbesondere von Fußgängern, ist in den letzten Jahren zunehmend größer geworden. Vor allem in modernen Überwachungs- und Fahrerassistenzsystemen wird diese immer häufiger benutzt. Letztere sollen nicht nur aufgrund der aktuellen Position eines Fußgängers warnend bzw. korrigierend eingreifen, sondern auch die Absicht von Fußgängern anhand ihrer aktueller Bewegungsrichtung und -geschwindigkeit miteinbeziehen. Damit kann die zukünftige Position einer Person prädiziert werden, i.d.R. repräsentiert durch den wahrscheinlichen Aufenthaltsbereich in den nächsten Sekunden.

Die in den genannten Systemen verwendeten Algorithmen sind sehr rechenintensiv. Außerdem werden zunehmend hochauflösende Kameras mit hohen Bildraten eingesetzt. Um die Echtzeitanforderungen einzuhalten, reichen auch die heutigen leistungsstärkeren **C**entral **P**rocessing **U**nits (CPUs) nicht immer aus. Abhilfe kann hier teilweise durch die Parallelisierung auf **G**raphics **P**rocessing **U**nits (GPUs) geschaffen werden. In einigen Fällen ist deren Geschwindigkeit jedoch ebenfalls nicht ausreichend. Daher werden auch **F**ield **P**rogrammable **G**ate **A**rrays (FPGAs) in der digitalen Bildverarbeitung verwendet. Neben dem in der Regel höheren Datendurchsatz besitzen diese zudem durch den geringeren Energie- und Platzbedarf den Vorteil, dass sie als eingebettete Systeme, z.B. in Fahrzeugen, eingesetzt werden können.

B. Stand der Technik

In den letzten Jahren wurden einige FPGA-Implementierungen zur Personenerkennung veröffentlicht (z.B. [1],[2],[3]). Diese verwenden meist den von Dalal und Triggs vorgestellten **H**istogram of **O**riented **G**radients (HOG) Deskriptor mit einer **S**upport **V**ector **M**achine (SVM) als Klassifikator [4],[5].

In diesen Veröffentlichungen wurden jedoch keine Angaben bezüglich der Fusion der Detektionen gemacht. Weiterhin sind den Autoren keine Veröffentlichungen zur FPGA-basierten Fusion von Fußgängererkennungen bekannt. Auch bei den zahlreichen CPU-Veröffentlichungen in der Objekterkennung wird meist nicht auf diese Thematik eingegangen oder ein heuristisches Verfahren benutzt [6]. Diese fusionieren in der Regel Detektionsfenster, welche sich mindestens zu einem bestimmten Anteil überlappen müssen. Darauf basiert auch die Standardfunktion zur Gruppierung von Rechtecken der Bildverarbeitungsbibliothek *OpenCV*, welche in vielen Softwareanwendungen zur Fusion von Objekterkennungen verwendet wird [7].

Ein komplexerer Ansatz, wie er z.B. auch in [5] eingesetzt wird, benutzt den Mean-Shift-Algorithmus. Dieser wurde bereits 1975 von Fukunaga und Hostetler veröffentlicht [8] und wird häufig zum Clustering und bei der Suche von Maxima benutzt. Beim Einsatz für die Fusion von Fußgängererkennungen werden die einzelnen Detektionen zunächst als Punkte in einen dreidimensionalen Raum projiziert und deren Verteilung wird durch eine Kernelfunktion abgebildet. Ausgehend von einem Startpunkt wird nacheinander ein Mean-

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

Abbildung 4: SVM-Klassifikationen mehrerer Skalenstufen

Abbildung 3: SVM-Klassifikationen mit fusionierten Detektionsergebnissen

Shift-Vektor berechnet, der iterativ die Zentren der einzelnen Cluster annähert. Diese stellen die fusionierten Detektionen dar. In *OpenCV* ist auch dieses Verfahren umgesetzt [7]. Durch den iterativen Ansatz ist es aber nicht für eine FPGA-Implementierung geeignet, da diese ihren Geschwindigkeitsvorteil durch Parallelisierung der Berechnungen erreichen.

C. Zielsetzung

Ausgehend von einer FPGA-basierten Fußgängererkennung [2], welche die Berechnung des HOG-Deskriptors mit nachgeschaltetem SVM-Klassifikator enthält, sollen die Detektionen mehrerer Skalenstufen (Multiskalenerkennung) geeignet fusioniert werden. Wie in Abbildung 1 beispielhaft zu sehen ist, unterscheiden sich Position und Größe der blauen Detektionen, die einer einzigen Person zugeordnet werden können, nur unwesentlich. Daher soll die hier vorgestellte Implementierung auf einem für eine FPGA-Umsetzung geeigneten Algorithmus basieren, der Fenster fusioniert, die eine hohe relative Überlappung besitzen.

Die Fusion soll jedoch nicht erfolgen, wenn sich die Fenster von benachbarten Fußgängern nur leicht überschneiden, wie es häufig bei Personengruppen auftritt. Zudem müssen sporadisch auftauchende Falscherkennungen eliminiert werden. Das Fusionsergebnis (grüne Rechtecke) in Abbildung 2 für das Beispielbild in Abbildung 1 entspricht diesen Anforderungen. Da Anhäufungen von Falscherkennungen (wie im rechten Bildteil) eine ähnliche Erscheinungsform wie positive Erkennungen haben, können diese nicht in der Fusion aussortiert werden und haben im Ergebnis eine Falscherkennung zur Folge (rotes Rechteck).

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
1	1	2	3	3	2	1	1
1	2	3	5	5	3	2	1
1	2	4	6	6	4	2	1
1	3	5	8	8	5	3	1
1	3	6	8	8	6	3	1
1	3	6	8	8	6	3	1
1	3	5	8	8	5	3	1
1	2	4	6	6	4	2	1
1	2	3	5	5	3	2	1
1	1	2	3	3	2	1	1
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Abbildung 2: Binäre (links) und Gauß-Gewichtung innerhalb des Detektionsfensters

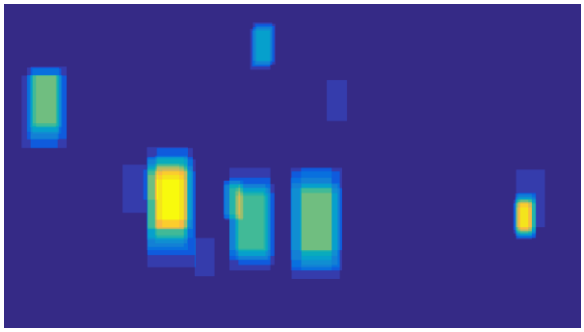


Abbildung 1: Gewichtete Detektionsfensterüberlappungen

Hinsichtlich der Echtzeitfähigkeit sollen mit dieser FPGA-Implementierung mindestens die Bilder einer automobilen Kamera verarbeitet werden können, welche eine Auflösung von 1280 x 720 Pixel besitzt und 30 Bilder pro Sekunde (fps) liefert.

Die Implementierung soll außerdem möglichst ressourcenschonend erfolgen. Dabei sollen etwa 20 Skalenstufen umgesetzt sein, da dies bei CPU- und GPU-Anwendungen üblich ist.

D. Überblick

Zunächst wird in Kapitel II der für die FPGA-Umsetzung entwickelte Fusionsalgorithmus vorgestellt. Dessen Implementierung und die darin enthaltenen Module werden anschließend in Kapitel III ausführlich beschrieben. Kapitel IV enthält die damit erzielten Ergebnisse, wobei auf Ressourcenbedarf, Echtzeitverhalten und Genauigkeit eingegangen wird. Abschließend folgt in Kapitel V eine Zusammenfassung.

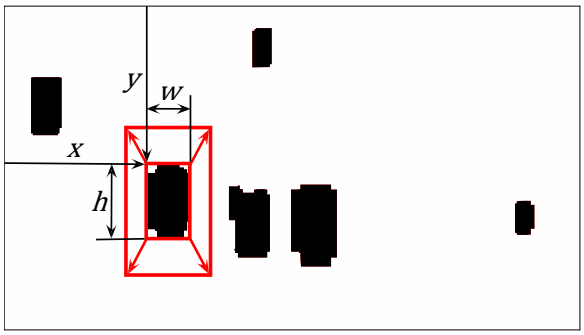


Abbildung 6: Berechnung der NMS-Ergebnisse aus der Position und Größe der binären Objekte

II. Algorithmus

Der konzipierte Algorithmus basiert auf der Auswertung der Überlappungen der Klassifikationsergebnisse (Abbildung 1). Deshalb werden zunächst die Überlappungen der Detektionsfenster pro Pixel für das gesamte Bild gezählt. Innerhalb von Fensteranhäufungen entstehen dabei höhere Werte. Damit die Anhäufungen von benachbarten Fußgängern, deren Detektionsfenster sich teilweise überlappen, besser voneinander getrennt werden können, werden die Pixel im Zentrum des Detektionsfensters stärker gewichtet. Eine binäre Gewichtung mit Kern und eine Gewichtung auf Basis der Gauß-Verteilung, wie sie in Abbildung 3 dargestellt sind, entsprechen dieser Anforderung. Abbildung 4 veranschaulicht die Verteilung der aufsummierten gewichteten Detektionsfensterüberlappungen für das Beispielbild, wobei hier die binäre Fenstergewichtung verwendet wird. Helle Bereiche kennzeichnen einen hohen Überlappungswert.

Im Anschluss wird über einen Vergleich mit einem Grenzwert ein binäres Bild erzeugt (Abbildung 5). Dadurch entstehen binäre Objekte, welche auch als **B**inary **L**arge **Ob**jects (BLOBs) bezeichnet werden. Mit Hilfe einer BLOB-Analyse wird die Position (x, y) sowie die Breite w und Höhe h des jeweils umschließenden Rechtecks ermittelt. Durch Vergrößerung wird abschließend das resultierende Fusionsergebnis erzeugt, was in Abbildung 5 beispielhaft für ein Objekt in rot dargestellt ist. Bei der hier eingesetzten binären Fenstergewichtung ist der Vergrößerungsfaktor 2, da der Kern der halben Fenstergröße entspricht (Abbildung 3).

Bei einem festen Vergleichswert für die Binarisierung hängt die Größe der erzeugten BLOBs von der Anzahl der Detektionsfenster ab, aus denen diese entstehen. Um die Größe unabhängig davon zu machen, wird der Vergleichswert $\tau_d(x, y)$ dynamisch für jede Bildposition bestimmt:

$$\tau_d(x, y) = \max(\lambda \cdot o_{max}(x, y), \tau_{min})$$

Der Vergleichswert wird vom mit λ gewichteten maximalen Überlappungswert $o_{max}(x, y)$ in der Umgebung von (x, y) abgeleitet, sobald dieser größer als ein

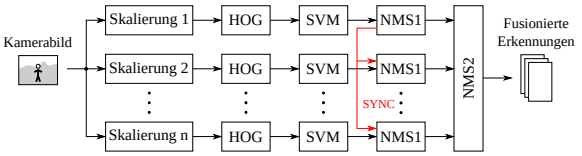


Abbildung 5: Gesamtsystem mit mehreren Skalenstufen und fusionierten Fußgängererkennungen

Minimalwert τ_{min} ist. Die Größe des Umgebungsfensters orientiert sich an den verwendeten Skalenstufen. In experimentellen Untersuchungen haben $\lambda = 0{,}75$ und $\tau_{min} = 3$ bei der Verwendung einer binären Fenstergewichtung zu guten Ergebnissen geführt.

III. Implementierung

Das hier vorgestellte FPGA-Design basiert auf der HOG- und SVM-Implementierung aus [2]. Für die Fußgängererkennung wird das Sliding-Window-Prinzip eingesetzt bei dem das Detektionsfenster mit der festen Größe von 64 x 128 Pixel von links oben nach rechts unten über das skalierte Bild verschoben wird. Der HOG-Deskriptor besitzt eine zellbasierte Arbeitsweise. Benachbarte Detektionsfenster sind um genau eine Zelle verschoben. Mit 8 x 8 Pixel wurde die in den meisten Anwendungen verwendete Zellgröße gewählt.

In diesem Kapitel wird zunächst auf die Architektur der gesamten Implementierung und den Aufbau der NMS eingegangen. Anschließend werden die Funktion und die Umsetzung der Hauptmodule der NMS genauer beschrieben.

A. Systemüberblick

1) Fußgängererkennung (Gesamtsystem)

Abbildung 6 zeigt den Aufbau des Gesamtsystems der Fußgängererkennung inklusive der NMS-Erweiterung. Durch den Multiskalenansatz können mehrere Skalenstufen parallel verarbeitet werden. In jedem Zweig befindet sich zunächst ein Skalierungsmodul mit dem das Eingangsbild auf eine bestimmte Auflösung herunterskaliert wird. Dadurch ist es möglich mit einer festen Fenstergröße für den HOG-Deskriptor unterschiedliche Skalenstufen zu klassifizieren. Nach der Deskriptorberechnung im HOG-Modul folgt die Klassifikation im SVM-Modul. Zum Schluss erfolgt die Fusion der Erkennungen in der NMS. Dabei enthält jede Skalenstufe den ersten Teil der NMS (Modul NMS1). Im Modul NMS2 folgt dann abschließend die eigentliche Fusion.

2) Fusion (NMS)

In Abbildung 7 sind die Hauptmodule der NMS enthalten. Die Module Zellüberlappungen und Pixelüberlappungen sind Teilmodule des NMS1-Moduls, Skalenfusion, Binarisierung und BLOB-Analyse gehören dem NMS2-Modul an.

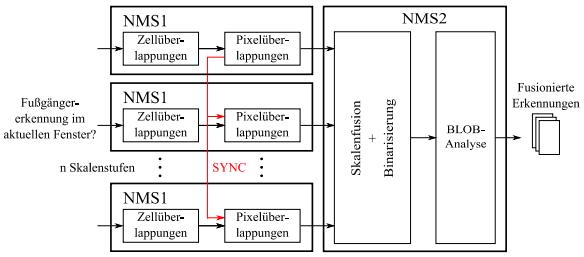


Abbildung 7: Aufbau des Fusionsmoduls (NMS)

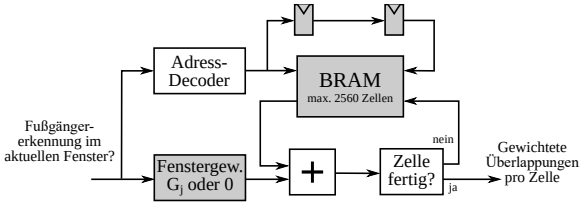


Abbildung 9: Aufbau des Moduls Zellüberlappungen

Der SVM-Klassifikator entscheidet, ob im aktuellen Fenster ein Fußgänger enthalten ist. Durch die zellbasierte Arbeitsweise ist das nächste Detektionsfenster, wie bereits erwähnt, um genau eine Zelle verschoben. Daher werden im Modul Zellüberlappungen zunächst die gewichteten Detektionsfensterüberlappungen pro Zelle berechnet. Im Anschluss erfolgt dann die Überführung in den Pixelbereich. Hier findet die Synchronisierung der Pixelüberlappungen der einzelnen Skalenstufen statt. Dabei fungiert die größte Skalenstufe, welche der Skalenstufe mit dem Eingangsbild entspricht, das am stärksten herunterskaliert ist, als Master. Die ausgegebenen und synchronisierten Pixelüberlappungen werden danach fusioniert und in das binäre Bild überführt. Die abschließende BLOB-Analyse wird in einem eigenen Modul durchgeführt.

B. Zellüberlappungen

Kern für die Berechnung der gewichteten Zellüberlappungen ist ein BRAM-Modul (Abbildung 8). Aufgrund des verwendeten Sliding-Window-Prinzips liegen in diesem die Zwischenwerte der gewichteten Zellüberlappungen für einen Bildbereich, welcher beim unskalierten Bild der Bildbreite (1280 Pixel) mal der Fensterhöhe entspricht (128 Pixel). Bei der hier gewählten Zellgröße von 8 x 8 Pixel ergibt sich die Anzahl der Zellen in diesem Bildabschnitt wie folgt:

$$\frac{1280}{8} \cdot \frac{128}{8} = 2560$$

Da die Bilder zusätzlich herunterskaliert werden, ist der Speicherbedarf, abhängig von der Skalenstufe, noch geringer.

Das Eingangssignal enthält die Information, ob im aktuellen Detektionsfenster eine Erkennung enthalten ist. Abhängig davon wird der zugehörige Bereich im BRAM aktualisiert. Um die Kontrolllogik einfach zu

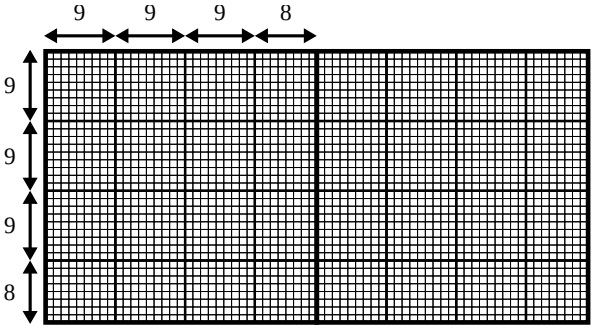


Abbildung 8: Interpolation von nicht ganzzahligen Zellgrößen

halten, wird bei einer positiven Erkennung für jede Zelle die entsprechende Zellgewichtung und im negativen Fall eine 0 addiert, da die Hardwareressourcen ohnehin benötigt werden. Die Zellgewichtungen sind dabei in einem ROM abgelegt. Dieses enthält für jede Zellposition im Detektionsfenster den passenden Wert. In Abbildung 3 sind die Zellgewichtungen schon in das 8 x 16 Zellen (64 x 128 Pixel) große Fenster eingeteilt.

Der Aktualisierungsvorgang wird für jede Zelle innerhalb des Detektionsfensters durchgeführt. Da im HOG- und SVM-Modul gemäß [2] ein Takt verwendet wird, welcher dem doppelten Pixeltakt entspricht, stehen zwischen zwei Detektionsfenstern immer die benötigten $8 \cdot 16 = 128$ Takte für die Aktualisierung zur Verfügung.

Liegt eine aktualisierte Zelle in keinem weiteren Detektionsfenster (linke obere Zelle im Fenster bzw. Zellen im Randbereich des Bildes), dann wird der gewichtete Überlappungswert ausgegeben. Andernfalls wird dieser in den BRAM zurückgeschrieben. Für den Speicherzugriff wurde ein Adress-Decoder implementiert, der für die aktuelle Zelle immer die korrekte Speicheradresse ausgibt.

C. Pixelüberlappungen

Nach der Berechnung der gewichteten Detektionsfensterüberlappungen pro Zelle erfolgt die Überführung in den Pixelbereich, da die Fusion der einzelnen Skalenstufen in der Auflösung des unskalierten Eingangsbildes (1280 x 720 Pixel in diesem Fall) erfolgt. Bezogen auf dieses Format ergeben sich größere Zellen. Wird z.B. das Eingangsbild auf die Hälfte herunterskaliert, so gehen die 8 x 8 Pixel großen Zellen in dem skalierten Bild aus 16 x 16 Pixel großen Bereichen im unskalierten Bild hervor. Entsprechende Werte ergeben sich bei anderen Skalenstufen.

Da die Pixel die Pipeline der FPGA-Implementierung seriell durchlaufen und eine Zellüberlappung nur zu einem bestimmten Takt anliegt, muss jeder Wert einer Zelle entsprechend der Zellbreite mehrfach in aufeinanderfolgenden Takten ausgegeben werden, um zu den Pixelüberlappungen zu gelangen. In Abbildung 9 wird daher der Wert der linken oberen 9 x 9 Pixel großen Zelle (schon in das unskalierte Bild überführt) zu Beginn des Bildes in neun aufeinanderfolgenden Takten

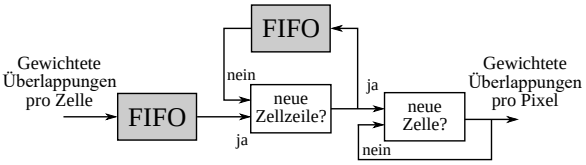


Abbildung 10: Aufbau des Moduls Pixelüberlappungen

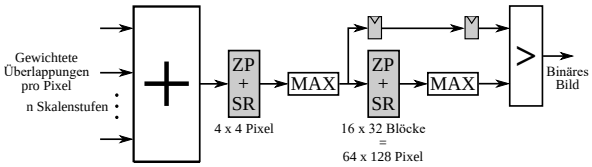


Abbildung 11: Aufbau des Fusions- und Binarisierungsmoduls

ausgegeben (horizontale Mehrfachausgabe). Mit den nächsten Zellen wird danach genauso verfahren bis die erste Pixelzeile vollständig ist. Da die Pixelzeilen innerhalb einer Zelle die gleichen sind, kann die erste Pixelzeile solange wiederholt werden, bis die nächste Zelle beginnt (vertikale Mehrfachausgabe). Jedes Pixel kann in Abbildung 9 demnach als Takt interpretiert werden, wobei zeilenweise von links oben nach rechts unten durchgegangen wird.

In der Regel sind die auf die Auflösung des Eingangsbildes zurückgerechneten Zellgrößen nicht ganzzahlig. Da für die Mehrfachausgabe nur mit diskreten Werten gearbeitet werden kann, wird ein Verfahren gewählt, welches auch bei Dual-Modulus-Zählern in Phasenregelschleifen eingesetzt wird. Allgemein kann die Zellengröße c in Pixeln wie folgt dargestellt werden:

$$c = N + \frac{A}{D} = \frac{A \cdot (N+1) + (D-A) \cdot N}{D}$$

Hierbei sind N, A und D ganzzahlig und positiv sowie $A \leq D$. Der rechte Teil dieser Gleichung bedeutet, dass A Zellen mit einer Breite von $(N+1)$ und anschließend $(D-A)$ Zellen mit einer Breite von N Pixeln im Durchschnitt den geforderten nicht ganzzahligen Wert für die Breite besitzen. Durch dieselbe Aufteilung in vertikaler Richtung können Einheitsblöcke gebildet werden, deren Zellen im Durchschnitt die geforderte Zellgröße besitzen, womit die Ungenauigkeit minimiert wird. Die Einheitsblöcke besitzen dabei eine Höhe und Breite von D Zellen. Abbildung 9 enthält beispielhaft zwei Einheitsblöcke für eine Zellgröße von $c = 8\frac{3}{4}$. Nach der Ausgabe von drei Zellen mit der Breite bzw. Höhe von 9 Pixeln folgt eine Zelle mit 8 Pixeln.

Für die Umsetzung dieses Konzepts wurde die Architektur in Abbildung 10 gewählt. Die Zellüberlappungen werden zunächst in einen Eingangs-FIFO-Speicher geschrieben. Dieser wird benötigt, da die unterschiedlichen Skalenstufen in diesem Modul synchronisiert werden (Abbildung 7).

Die Mehrfachausgabe ist durch zwei Schleifen dargestellt. Die horizontale entsteht durch die rechte Schleife. Hierfür wird der entsprechende Wert einer Zelle so lange ausgegeben bis die Zellbreite erreicht ist. Dies wird für jede Zelle wiederholt. Die vertikale Mehrfachausgabe entsteht in der Schleife in der Mitte. Für die Wiederholung der Pixelzeilen werden die Zellüberlappungen nach der Übergabe an die horizontale Ausgabeschleife zusätzlich in einem Speicher abgelegt (oberer FIFO). Dieser fungiert innerhalb einer Zellzeile als Eingang. Beginnt eine neue Zellzeile, dann werden

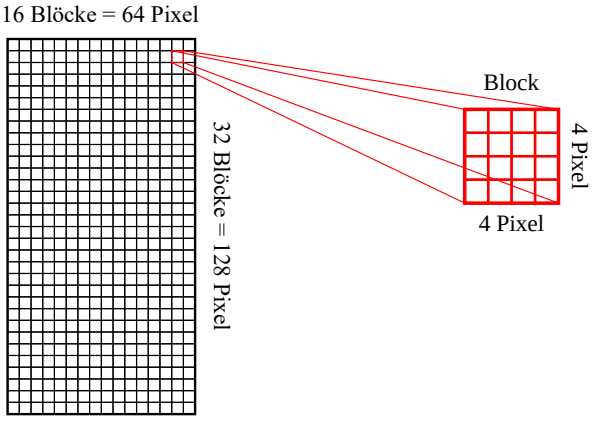


Abbildung 12: Umgebungsfensters für die Bildung des dynamischen Vergleichswerts

aus dem eigentlichen Eingangs-FIFO die Überlappungswerte gelesen. Die benötigen Zähler werden je nach Position innerhalb des Einheitsblockes zurückgesetzt.

D. Skalenfusion, Binarisierung und BLOB-Analyse

Die nun synchronisierten gewichteten Pixelüberlappungen aller Skalenstufen werden zunächst durch Addition gemäß Abbildung 11 fusioniert. Der benötigte dynamische Vergleichswert wird, wie bereits erwähnt, vom Maximum eines Umgebungsfensters (nicht zu verwechseln mit dem Fenster für die Klassifikation) abgeleitet. Die gewählte Umgebungsfenstergröße hängt dabei von den verwendeten Skalenstufen ab. Die besten Ergebnisse wurden hier mit einem Umgebungsfenster erzielt, dessen Maße einem durch mittlere Skalenstufen erzeugten BLOB entsprechen, was in dieser Implementierung in etwa 64 x 128 Pixel sind.

Bei naiver Implementierung müssten dabei aber 127 Zeilen und damit $127 \cdot 1280 = 162560$ Überlappungswerte gespeichert werden. Dies stellt für eine FPGA-Umsetzung einen unverhältnismäßig großen Speicheraufwand dar. Deshalb wird ein blockbasierter Ansatz gewählt, wie er in Abbildung 12 dargestellt ist. Jeder Block bekommt den maximalen Überlappungswert des Pixels zugeteilt, der sich in diesem befindet. Hierfür werden die fusionierten Pixelüberlappungen zuerst in einen Zeilenpuffer mit anschließendem Schieberegister (ZP + SR) geschrieben, damit zu jedem Takt der entsprechende zweidimensionale Pixelbereich vorliegt. Im Anschluss erfolgt die Bildung des Maximums. Die Blockgröße sollte dabei nicht zu groß gewählt werden, da die Genauigkeit sonst zu sehr abnimmt. Hier

Tabelle 1: Ressourcenbedarf der einzelnen Module der NMS bei 18 parallelen Skalenstufen

	NMS1				NMS2			
	Zellüber-lappungen	Pixelüber-lappungen	Gesamt		Binari-sierung	BLOB-Analyse	Gesamt	
Register	4.776	5213	10.478	2,4 %	16.069	3.881	20.778	4,8 %
LUTs	3.695	3.758	7.779	3,6 %	11.567	3.259	15.150	6,9 %
18k BRAMs	48	144	192	17,6 %	34	8	46	4,2 %
DSP-Zellen	0	0	0	0 %	0	0	0	0 %

Tabelle 2: Ressourcenbedarf für ein Xilinx Kintex®-7 FPGA (Zynq-SoC XC7Z045) bei 18 parallelen Skalenstufen

	Design	Verfügbar	Prozent
Register	172.294	437.200	39,4 %
LUTs	129.708	218.600	59,3 %
18k BRAMs	691	1090	63,4 %
DSP-Zellen	160	900	17,8 %

Tabelle 3: Ressourcenbedarf der einzelnen Module einer Skalenstufe der Fußgängererkennung

	Skalie-rung	HOG	SVM	NMS1
Register	1.484	3.742	3.229	655
LUTs	1.113	2.729	2.671	486
18k BRAMs	1	11	13	12
DSP-Zellen	4	4	2	0

wurde eine Größe von 4 x 4 Pixel gewählt, da dabei der Genauigkeitsverlust für diese Anwendung noch tragbar ist. Das Umgebungsfenster besteht demnach aus 16 x 32 Blöcken.

Mit einem weiteren Zeilenpuffer und Schieberegister für die Blockmaxima wird das Maximum des Umgebungsfensters bestimmt (Abbildung 11). Mit diesem werden die Blockmaxima verglichen und gemäß Kapitel II binarisiert.

Die hier eingesparten Speicherressourcen sind im Vergleich zu einer einfachen Implementierung erheblich. Für die Berechnung der Blockmaxima müssen $3 \cdot 1280 = 3840$ Überlappungswerte gespeichert werden. Nach Überführung in den Blockbereich besitzt das Bild eine Auflösung von 320 x 180 Blöcken. Daher müssen bei der Bildung des Umgebungsfensters nur weitere $31 \cdot 320 = 9920$ Überlappungswerte gespeichert werden. Der insgesamt benötigte Zeilenpuffer muss hier demnach nur für $3840 + 9920 = 13760$ anstatt für 162560 Werte ausgelegt sein (Einsparung von 91,5 %). Zudem werden weitere Ressourcen im Schieberegister und bei der Bildung der Maxima eingespart.

Für die abschließende BLOB-Analyse wird auf das Modul aus [9] zurückgegriffen.

IV. Ergebnisse

Im Folgenden werden die Ergebnisse dargestellt, die mit der hier beschriebenen Implementierung erzielt wurden.

A. Ressourcen

Zur Evaluierung wurden 18 parallele Skalenstufen auf dem Kintex®-7 FPGA, welches in das Zynq-**S**ystem-**o**n-**C**hip (Zynq-SoC) XC7Z045 von Xilinx integriert ist, implementiert. In Tabelle 1 ist der Ressourcenbedarf der Hauptmodule der NMS dieser Implementierung enthalten, wobei für das NMS1-Modul die gesamten Ressourcen aus allen Skalenstufen angegeben sind. Wie zu sehen ist, ist der BRAM im NMS1-Modul die Ressource mit der größten relativen Auslastung. Insbesondere das Modul für die Pixelüberlappungen ist hierfür verantwortlich. Jedoch ist der benötigte Anteil im Vergleich zu den insgesamt verfügbaren Ressourcen noch gering.

Im Modul NMS2 ist der relative Bedarf an den einzelnen Ressourcenarten nahezu ausgeglichen, wobei das Teilmodul für die Binarisierung hier den Hauptanteil der Ressourcen beansprucht. Durch den blockbasierten Ansatz werden jedoch schon erheblich Ressourcen eingespart. Auffällig ist weiterhin, dass in keinem Modul DSP-Zellen benötigt werden.

In Tabelle 2, welche den Ressourcenbedarf des gesamten Designs (inkl. HOG-Deskriptor und SVM-Klassifikator) enthält, ist zu erkennen, dass der BRAM das insgesamt begrenzende Element ist. Es können aber noch mehr als die hier implementierten 18 Skalenstufen umgesetzt werden, da noch Reserven vorhanden sind.

Tabelle 3 vergleicht den durchschnittlichen Ressourcenbedarf der einzelnen Module der Fußgängererkennung einer Skalenstufe (Abbildung 6). Es fällt auf, dass das NMS1-Modul jeweils deutlich weniger Ressourcen

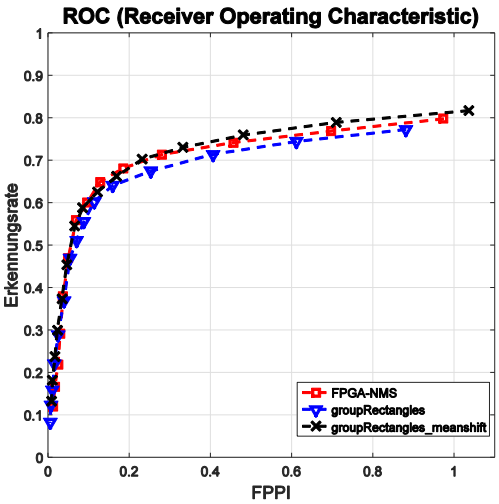


Abbildung 13: Vergleich der Erkennungsrate mit CPU-Anwendungen (basierend auf OpenCV)

benötigt als die anderen Hauptmodule der Fußgängererkennung. Nur der benötigte BRAM entspricht in etwa dem des HOG- und SVM-Moduls.

B. Echtzeitverhalten

Die Implementierung besitzt eine Schnittstelle zu einer automobilen Prototypenkamera, die Bilder mit einer Auflösung von 1280 x 720 Pixel bei einer Bildrate von 30 fps liefert. Hierfür wird ein Pixeltakt von mindestens 28 MHz benötigt. Das Design weist aber auch mit 100 MHz als Pixeltakt noch eine korrekte Funktionsfähigkeit auf. Das ergibt eine maximale Bildrate von 108 fps für die Auflösung der Kamera. Full-HD-Bilder (1920 x 1080 Pixel) können bei diesem Takt mit bis zu 48 fps verarbeitet werden.

C. Genauigkeit

Um die Anzahl der Falscherkennungen zu verringern, wurde gemäß [10] für jede Skalenstufe eine Region of Interest (ROI) innerhalb des Bildes festgelegt, mit der falsche SVM-Klassifikationen herausgefiltert werden (s. Falscherkennungen in Abbildung 2). Dabei wird angenommen, dass sich alle Fußgänger auf einer flachen und durch eine vorherige Kalibrierung bekannten Ebene (in der Regel die Straße) befinden. Durch diese Annahme können die möglichen Positionen der Fußgänger innerhalb des Bildes eingegrenzt werden.

Für die Beurteilung der Genauigkeit dieser Implementierung wurden die Bilder der *Daimler Pedestrian Path Prediction* Datenbank [11] verwendet. Als Vergleichswerte dienen die Ergebnisse, welche mit der *groupRectangles*- bzw. *groupRectangles_meanshift*-Funktion (Standardfunktion zum Gruppieren von Rechtecken bzw. Mean-Shift-Verfahren) in *OpenCV* erzielt wurden [7]. Um nur die Fusion der Fußgängererkennungen bewerten zu können, wurden für die CPU-Anwendungen die auf dem FPGA berechneten SVM-Klassifikationen als Eingangsdaten verwendet.

In Abbildung 13 ist zu sehen, dass für einen niedrigen Wert der Falscherkennungen pro Bild (**F**alse **P**ositives **p**er **I**mage: FPPI) die FPGA- und CPU-Anwendungen nahezu die gleiche Genauigkeit besitzen. Bei größeren FPPI-Werten ist die FPGA-Implementierung etwas besser im Vergleich zur *groupRectangles*-Funktion, wobei das Mean-Shift-Verfahren noch eine etwas höhere Genauigkeit besitzt (jeweils um ca. 2 %).

V. ZUSAMMENFASSUNG

Die hier vorgestellte Implementierung basiert auf einer Methode, welche gewichtete Detektionsfensterüberlappungen für die Fusion von Fußgängererkennungen (NMS) verwendet. Es ist die erste, den Autoren bekannte, rein FPGA-basierte Umsetzung. Dabei können die Erkennungen von mehreren parallelen Skalenstufen aus Bildern mit einer Auflösung von 1280 x 720 Pixel und einer Bildrate von über 100 fps fusioniert werden. Durch den blockbasierten Ansatz für den dynamischen Vergleichswert wurde die FPGA-Umsetzung erst möglich, da damit erhebliche Ressourcen eingespart werden.

Die Genauigkeit der Implementierung ist mit in der Bildverarbeitung gängigen Softwareanwendungen vergleichbar.

LITERATURVERZEICHNIS

[1] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, M. Yoshimoto, “Architectural Study of HOG Feature Extraction Processor for Real-Time Object Detection”, *2012 IEEE Workshop on Signal Processing Systems (SiPS)*, S. 197-202, Québec City, Okt. 2012.

[2] M. Hahnle, F. Saxen, K. Doll, "Erkennung von Fußgängern in Echtzeit auf FPGAs", *49. Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, S. 57-65, Mannheim, Feb. 2013.

[3] M. Hemmati, M. Biglari-Abhari, S. Berber, S. Niar, “HOG Feature Extractor Hardware Accelerator for Real-time Pedestrian Detection”, *2014 17th Euromicro Conference on Digital System Design (DSD)*, S. 543-550, Verona, Aug. 2014.

[4] N. Dalal, B. Triggs, „Histograms of Oriented Gradients for Human Detection“, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, S. 886-893, San Diego, Juni 2005.

[5] N. Dalal, „Finding People in Images and Videos“, *PhD Thesis*, Grenoble, Juli 2006.

[6] P. Viola, M. J. Jones, “Robust Real-Time Face Detection”, *International Journal of Computer Vision*, 57(2), S. 137-154, Kluwer Academic Publishers, Mai 2004.

[7] OpenCV. Open Source Computer Vision, “Object Detection”, 2015, *http://docs.opencv.org/master/d5/d54/group__objdetect.html*, Datum des Zugriffs: Jan. 2016.

[8] K. Fukunaka, L. D: Hostetler “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition”, *IEEE Transactions on Information Theory*, 21(1), S. 32-40, Jan. 1975.

[9] F. Fellhauer, M. Schmitt, K. Doll, "Echtzeit-BLOB-Analyse mit Lauflängenkodierung und –dekodierung auf einem FPGA", *47. Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg*, S. 35-41, Offenburg, Feb. 2012.

[10] W. Tian, M. Lauer, „Fast Cyclist Detection by Cascaded Detector and Geometric Constraint“, *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, S. 1286-1291, Las Palmas, Sept. 2015.

[11] N. Schneider, D. M. Gavrila, „Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Sudy“, *35th German Conference on Pattern Recognition (GCPR)*, S. 174-183, Saarbrücken, Sept. 2013.

Sascha Hock erhielt den akademischen Grad des B.Eng. in Mechatronik im Jahr 2014 von der Hochschule Aschaffenburg, wo er sein Masterstudium in Elektro- und Informationstechnik voraussichtlich im Jahr 2016 abschließen wird.

Michael Hahnle erhielt den akademischen Grad des B.Eng. in Elektro- und Informationstechnik im Jahr 2012 von der Hochschule Aschaffenburg, wo er sein Masterstudium in Elektro- und Informationstechnik voraussichtlich im Jahr 2016 abschließen wird.

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

Entwicklung eines GPS-gestützten Fahrdynamikmesssystems mit einem Dual-Core Bare-Metal Asymmetrischen Multiprozessorsystem und programmierbarer Logik auf einem Xilinx Zynq 7000

Alexander Riske, Manuel Schappacher, Axel Sikora.

***Zusammenfassung*—Die neueste Generation von programmierbaren Logikbausteinen verfügt neben den konfigurierbaren Logikzellen über einen oder mehrere leistungsfähige Mikroprozessoren. In dieser Arbeit wird gezeigt, wie ein bestehendes Zwei-Chip-System auf einen Xilinx Zynq 7000 mit zwei ARM A9-Cores migriert wird. Bei dem System handelt es sich um das „GPS-gestützte Kreiselsystem ADMA" des Unternehmens GeneSys. Die neue Lösung verbessert den Datenaustausch zwischen dem ersten Mikroprozessor zur digitalen Signalverarbeitung und dem zweiten Prozessor zur Ablaufsteuerung durch ein Shared Memory. Für die schnelle und echtzeitfähige Datenübertragung werden zahlreiche hochbitratige Schnittstellen genutzt.**

***Schlüsselwörter*— Systementwurf, Zynq-7000 SoC, Asymmetric Multiprocessing (AMP)**

I. Einleitung

Systeme mit einer hohen Rechengeschwindigkeit für die digitale Echtzeitsignalverarbeitung erfordern heutzutage oft den Einsatz eines ASICs oder eines FPGAs. Um einen langen Entwicklungs- und Fertigungsprozess eines ASICs einzusparen, werden mit Hilfe von FPGAs Konzepte getestet und anschließend in Hardware überprüft oder bei geringen Stückzahlen sogar als Produktivsystem eingesetzt. Diese programmierbaren Logikbausteine bieten mehrere tausend Logikgatter, vorgefertigte Funktionsblöcke und standardisierte Mikroprozessoren. Mit geringem Aufwand (Zeit und Kosten) können FPGAs neu konfiguriert werden und gewährleisten einen langen Einsatz des Systems.

Das Kreiselsystem ADMA3 (Automotive Dynamic Motion Analyzer) der Fa. GeneSys Elektronik GmbH [1] wird für Lokalisierungsanwendungen im Automotive Bereich eingesetzt. Es ist mit hochgenauen Inertialsensoren ausgerüstet. Ein integrierter DGPS-Receiver bzw. ein externes Geschwindigkeitsmesssystem verbessert die Systemperformance bei komplizierten und aufwändigen Testfahrten. Bewegungsdaten, wie Drehraten oder Beschleunigung in drei Achsen, von zwei oder mehreren Fahrzeugen können mit dem Kreiselsystem synchron erfasst und von zentralen Datenerfassungssystemen verrechnet werden [1].

Die aktuelle Version, das ADMA3-System, basiert auf einer klassischen Zweichip-Lösung, die einen rechenstarken Digitalen Signal Prozessor (DSP) mit einem programmierbaren FPGA kombiniert. Dadurch werden Rechenleistung und Flexibilität des Systems gleichermaßen gewährleistet. Neue FPGA-Familien, beispielsweise der Xilinx Zynq XC7020, kombinieren diese Vorteile, indem fest verdrahtete rechenstarke Mikrocontroller zusammen mit programmierbarer Logik in einem Chip integriert werden. Dadurch ergeben sich im Vergleich zum aktuellen ADMA3 System neue Möglichkeiten, wie z.B.:

- Reduzierung auf ein Single-Chip-System
- Einsparung und Vereinfachung beim Leiterplattendesign, da insbesondere die hochgetakteten Bussysteme zur Kopplung der ICs nicht mehr über die Platine geführt werden müssen,
- Senkung der Herstellungskosten,
- engere Kopplung zwischen programmierbarer Logik und Recheneinheit,
- hohe Rechenleistung durch fest verdrahtete Mikrocontroller.

Auf Basis eines solchen Zynq Systems soll die Performance und der Funktionsumfang der ADMA3 noch weiter gesteigert werden. Dieser Beitrag beschreibt die wichtigsten Schritte der Migration des ADMA3 Systems von einer Zweichip- zu einer Einchip-Architektur und geht dabei auf die Architektur und die Komponenten des Ausgangs- und des Zielsystems ein.

II. Aktuelles ADMA3 System

A. Funktionsweise

Das Kreiselsystem ADMA3 der Fa. GeneSys wird für Lokalisierungsanwendungen und Fahrdynamikmessungen im Automotive Bereich eingesetzt und besteht aus einem hochpräzisen Faserkreiselmesssystem mit DGPS. Es ermöglicht die hochgenaue Erfassung von Messdaten, welche für die weitere Datenverarbeitung über zahlreiche Schnittstellen ausgegeben und durch Dateneingänge unterstützt werden können.

Alexander Riske, alexander.riske@hs-offenburg.de und Manuel Schappacher, manuel.schappacher@hs-offenburg.de, arbeiten im Institut für verlässliche Embedded Systems (ivESK) an der Hochschule Offenburg, Badstr. 24 77652 Offenburg. Axel Sikora, axel.sikora@hs-offenburg.de, ist der wissenschaftliche Leiter des Instituts.

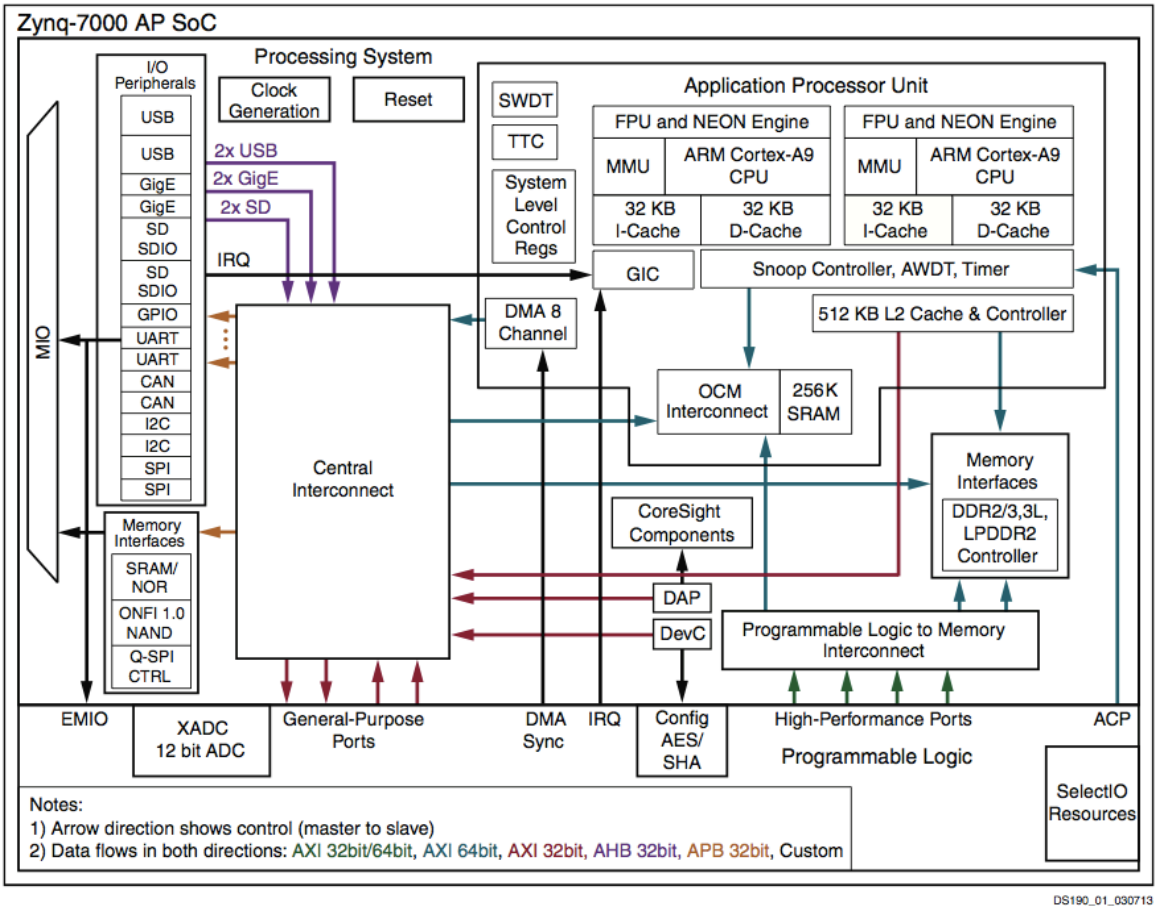


Abbildung 1: Übersicht der ZYNQ 7000er Architektur. [2]

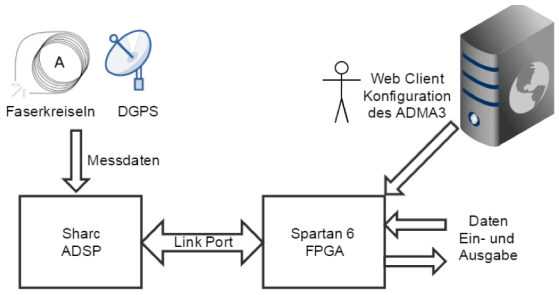


Abbildung 2: Blockschaltbild des ADMA3.

Hierzu zählen u.a.:3x Gigabit-Ethernet für die Messdatenausgabe, sowie zur Ausgabe von Steuerinformationen für Lenkroboter und für die Kommunikation mit andern ADMAs (über eine WLAN-Brücke),

- 5x CAN für die Messdatenausgabe und zur Steuerung von Messungen,
- 2x COM für die Messdatenausgabe und zur Konfiguration,
- Schnittstellen für die Anbindung von GPRS-Modems zum Empfang von DGPS Korrekturdaten,
- diverse Schnittstellen zur Anbindung von Peripheriegeräten, wie externe Barometer oder digitale Signal Ein-/Ausgänge.

Um die Funktionsweise, die Schnittstellen und die Datenausgabe konfigurieren zu können, steht dem Benutzer ein umfangreiches Webinterface zur Verfügung. Dort können sowohl die Basis-Konfigurationen der Schnittstellen (z.B. Geschwindigkeit) als auch die Datenausgabe (z.B. Paketwahl und Messfrequenz) konfiguriert werden. Dabei ist es prinzipiell möglich, alle Messdaten beliebig auf die zur Verfügung stehenden Schnittstellen zu verteilen.

B. Architektur des ADMA3

Das ADMA3 System (Abbildung 2) besteht aus einem 32-Bit Digitalen Signal Prozessor (Analog Devices SHARC) und einem Xilinx Spartan 6 FPGA, der neben Custom-Logic auch einen MicroBlaze Mikrocontroller [3] als Softcore IP enthält.

Der FPGA und der darin enthaltene MicroBlaze Prozessor haben zwei große Aufgabenbereiche. Zum einen wird über einen integrierten Webserver die Benutzerschnittstelle für die Konfiguration des Systems bereitgestellt. Zum anderen werden innerhalb des FPGAs die Schnittstellen für die Daten Ein- und Ausgabe in Form von IP-Cores instanziiert. Zur Systemeinbindung sind hier u.a. drei Gigabit-Ethernet und fünf CAN-Bus-Schnittstellen in programmierbarer Logik realisiert, die eine schnelle und echtzeitfähige Datenübertragung erlauben. Hinzu kommen diverse digitale, analoge und serielle Schnittstellen, z.B. für die Ein- und Ausgabe von digitalen/analogen Signalen oder zur Anbindung eines GPRS-Modems, welches Korrekturdaten für den GPS-Receiver liefert.

Der Datenaustausch und die damit verbundene Kommunikation zwischen den beiden Bausteinen findet über den sogenannten Link Port TigerSHARC (LP) [4], ein standardisiertes Transferprotokoll der Firma Analog Devices für den Datenaustausch zwischen DSPs, statt. Derzeit werden hier Datentransfermengen von bis zu 4kB pro Abtastzeitpunkt übertragen. Die Abtastfrequenz kann der Benutzer über das Webinterface im Bereich von 50Hz und 1kHz vorgeben, wodurch derzeit eine maximale Datentransferrate von 4MB/s erreicht wird.

Sowohl die Auswertung der Messdaten als auch die Berechnung der komplexen Navigationsparameter übernimmt der DSP. Mit Hilfe von Inertialsensorik berechnet der Signalprozessor Geschwindigkeit, Ort und Lage in einem 3-D-Raum zentimetergenau. Eventuelle Driftverhalten werden durch den Einsatz von GPS kompensiert. Die errechneten Parameter werden dann dem FPGA, welcher für das Senden der Daten zuständig ist, über den Link Port weitergeleitet.

III. Single-Chip Migration

A. Motivation

Zusätzlich zu den bereits genannten Vorteilen optimiert eine Single-Chip-Lösung zusätzlich die Kommunikation zwischen den beiden großen Aufgabenbereichen, die bisher von unterschiedlichen Bausteinen durchgeführt wurden. Der Datenaustausch muss nicht mehr über das Linkport-Transferprotokoll erfolgen, sondern kann durch einen gemeinsam genutzten Speicher (Shared Memory) stattfinden. Dadurch wird die Kommunikation vereinfacht und beschleunigt, da zum einen keine Rechenleistung z.B. für die Kommunikationssicherheit, Datenrahmengenerierung oder CRC Berechnung aufgebracht werden muss und zum anderen keine externe Schnittstelle erforderlich ist. Die Kommunikationsgeschwindigkeit des Systems erhöht sich zudem durch den gemeinsam genutzten Speicher, da sowohl Prozessor als auch das Shared Memory mit dem gleichen Takt und somit synchron betrieben werden. Die zeitliche Anpassung der Kommunikation an das Transferprotokoll und die Kompatibilität zu den unterschiedlichen Clock-Domänen wird somit minimiert. Des Weiteren können zusätzliche Flags oder Steuerbytes über das Shared Memory ohne viel Aufwand übermittelt werden. Dies führt zu einer flexiblen Steuerung des Gesamtsystems.

B. ZYNQ- 7000 SOC

Für die Migration auf eine Single-Chip Lösung wurde das Xilinx Zynq-7000 All Programmable SoC ZC702 Board als Grundlage verwendet. Xilinx bietet mit der Zynq-7000 Plattform (**Fehler! Verweisquelle konnte nicht gefunden werden.**) eine Möglichkeit einen Dual-Core ARM Cortex A9 Prozessor mit einem Systemtakt bis zu 1 GHz und mit konfigurierbaren, programmierbaren Logikblöcken in einem Chip zu verbinden [2].

Dabei unterteilen sich die Zynq 7000er FPGAs in zwei Bereiche: „Processing System“ und „Programmable Logic“. Zusätzlich zum ARM Dual-Core liegen im Prozessorsystem ein separater L1-Cache, gemeinsamer L2-Cache und ein 256kB On-Chip-Memory (OCM) als Hard IP Core vor. Zusätzlich werden die wichtigsten Peripherieschnittstellen unterstützt: GBit-Ethernet, CAN, USB2.0, Quad SPI Flash, General Purpose I/Os.

Mit Hilfe eines Bitstreams werden Logikblöcke konfiguriert, die dann entsprechende Algorithmen implementieren. Eine parallele Abarbeitung führt hierbei zu einer Leistungssteigerung des Systems, da verschiedene Verarbeitungsoperationen nicht auf die gleichen Ressourcen angewiesen sind. Diese Umsetzung wird durch Logikblöcke, BRAM oder vorkonfektionierte IP Cores unterstützt.

C. Asymmetrisches Multiprozessorsystem

Der Dual-Core Prozessor, der ZYNQ- 7000er Serie, erlaubt die Benutzung diverser Betriebssysteme, unter anderem Peta-Linux oder FreeRTOS. Zusätzlich können auf den einzelnen Prozessoren unterschiedliche Betriebssysteme unabhängig voneinander laufen. Dieser Systemaufbau wird als „Asymmetrisches Multiprozessorsystem“ bezeichnet.

Das vorgestellte Projekt baut auf eine Bare-Metal Konfiguration auf. Hierbei wird bei beiden Prozessorkernen komplett auf ein Betriebssystem verzichtet. Die einzelnen Prozessoren übernehmen einen eigenständigen Aufgabenteil des Gesamtsystems.

Der erste Prozessor (Core 0) wird im weiteren Verlauf als „Interface Core“ bezeichnet und übernimmt die Tätigkeit des bisherigen MicroBlaze Microcontrollers. Hierfür durchläuft der Interface Core in der ersten Phase das Boot-ROM und legt den Core 1, wird im weiteren Verlauf als „Navigation Core“ bezeichnet, schlafen. Die eigentliche Ausführung wird auf dem primären Kern (Core 0) fortgesetzt. Das Boot-ROM initialisiert das System, lädt in der zweiten Phase den First Stage Boot Loader (FSBL) und programmiert den FPGA mit seiner Netzliste. Sobald dieser Vorgang

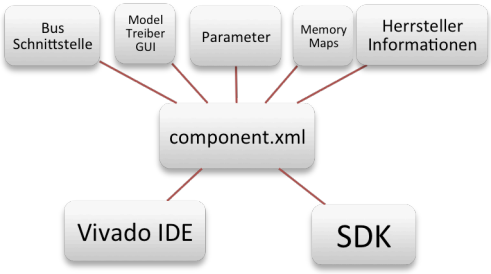


Abbildung 3: Generierung eines Custom IP Cores mit Hilfe des Vivado IP Packagers.

abgeschlossen ist, wird die Firmware aus einem vorher definierten Speicher geladen. Diese wird nach dem Advanced Encryption Standard (AES-128) Verschlüsselungsverfahren entschlüsselt, auf Veränderungen oder Fehler geprüft und ausgeführt. Der Interface Core initialisiert die Schnittstellen und startet sowohl den Webserver als auch den Navigation Core. Nach der Konfiguration der Interfaces über den Webserver fungiert der Interface Core als Datenbroker.

Der zweite Core (Core 1) übernimmt die Tätigkeit des Signalprozessors, dessen Aufgaben vom Industriepartner GeneSys gepflegt werden.

D. Shared Memory

Die beiden Bare-Metal Systeme besitzen jeweils einen eigenen nicht überlappenden Speicherbereich im externen DDR3-SDRAM Speicher. Für die Kommunikation zwischen den beiden Prozessorkernen ist im Processing System ein On-Chip-Memory (OCM) mit 256kB RAM und 128kB ROM (Boot-ROM) vorgesehen. Der RAM-Speicherbereich lässt sich in 4 x 64kB Seiten unterteilen [2] . Der Datenaustausch wird derzeit über eine Page in Form einer FIFO Struktur realisiert, welche die Datenfunktionalität, Datenziel und den Datenstrom beinhaltet.

Ein Mutex verwaltet den wechselseitigen Ausschluss des gemeinsam genutzten Speichers. Wenn ein Prozessorkern Zugriff auf einen kritischen Bereich benötigt, sperrt dieser mit der Prozedur *mutex_lock* die kritischen Daten. Der Datenzugriff wird für den zweiten Prozessorkern solange blockiert bis der komplette Schreib- bzw. Lesevorgang abgeschlossen und mit einem *mutex_unlock* freigegeben wird.

Um Inkonsistenzen im gemeinsamen Speicherbereich durch die L1-Caches der jeweiligen Prozessoren zu vermeiden, werden diese für den kompletten gemeinsam genutzten Speicherbereich deaktiviert.

IV. Custom Logic

A. Design-Suite Vivado

Ab der FPGA-Serie 7 hat sich Xilinx entschieden eine komplett neue Produktlinie von FPGAs einzuführen, welche nur mit der Design-Suite „Vivado“ pro

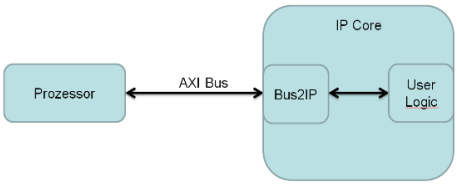


Abbildung 4: Anbindung der Custom IP Cores.

grammiert werden soll. Die Vivado-Werkzeuge beschleunigen im Vergleich zur Vorgängerlösung ISE nicht nur die Entwicklung von programmierbarer Logik und I/O, sondern vereinfachen auch die Systemintegration und -implementierung. Vivado ist eine integrierte Entwicklungsumgebung (IDE) welche auf Industriestandards, beispielsweise AMBA4-AXI4-BUS, IP-Packaging-Metadaten oder der Tool-Command-Language (Tcl) basiert [5].

Mit Hilfe des Vivado-IP-Packagers können eigene Custom IP-Cores generiert werden [6]. Eine VHDL-Rahmendatei (Schnittstellenbeschreibung des Funktionsblocks), die Quelldateien und die Treiber des IP Cores beschreiben den kompletten Funktionsblock in Form einer XML-Datei (Abbildung 3). Eine starre Pfadzuweisung der einzelnen Elemente wird hierfür nicht benötigt. Zusätzlich kann die IP-Beschreibung des Funktionsblockes mit weiteren Informationen, wie Dokumentationen, Simulationen oder Anwendungsbeispielen erweitert werden. Bei der Einbindung solcher IP-Cores in ein FPGA-Design können die Vivado Werkzeuge anhand dieser Datei die einzelnen Informationen auslesen und für weitere Vorgänge, beispielsweise Synthese oder Erstellung eines Software Projektes benutzen.

In wird die genauere Struktur eines Custom IP-Cores dargestellt. Ein AXI4-Lite IP Interface (IPIF) [7] vereinfacht die Schnittstelle zwischen dem ARM AXI und den Custom-IP-Core. Diese Schnittstelle ist speziell für Slave Anwendungen gedacht und übersetzt das AXI Protokoll zeitlich genau in IP interconnect (IPIC). Anhand der AXI Protokolladresse generiert der Adressen-Decoder die Steuersignale chip select, read- oder write enable, die von der sogenannten User-Logic (top sheet des IP Cores) zur Steuerung und Kommunikation mit den AXI-Teilnehmern verwendet werden.

Abbildung 5 zeigt die Ergebnisse der Hardware-Implementierung. Der aktuelle Systementwurf verbraucht ca. 85% der Lookup-Tabellen (LUT) und ca. 50% der Register. Die Ressourcen des FPGAs sind somit fast ausgeschöpft. Deshalb wird für weitere Entwicklungen zu einem größeren FPGA der gleichen Familie tendiert.

B. Multi-CAN IP-Core

Einer der am häufigsten genutzten Schnittstellen zur Messdatenausgabe ist die CAN-Schnittstelle. Die Datenübertragungsrate eines Highspeed-CAN-Buses beträgt maximal 1 Mbit/s und erlaubt eine Datenüber-

Ressourcen	Verbrauch	
FF	56689	(54%)
LUT	45180	(85%)
Memory LUT	2518	(15%)
I/O	99	(50%)
BRAM	35	(25%)

Abbildung 5: Ressourcenverbrauch nach der Implementierung.

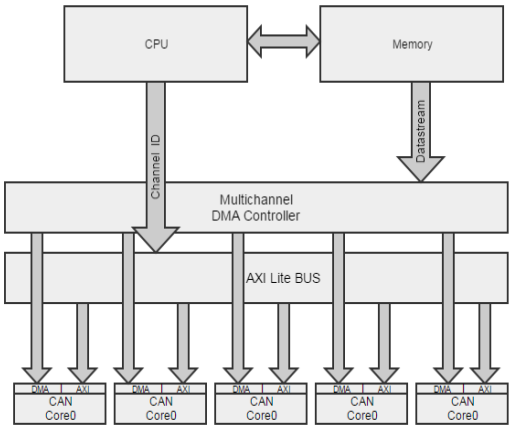


Abbildung 6: Aufbau des CAN Multichannel IP-Core.

tragung von bis zu 8Bytes bzw. 64bit pro Datenframe [8]. Laut dem Standard CAN 2.0A wird ein Overhead von 44bits für die Identifizierung der Nachricht benötigt. Der Benutzer hat z.Zt. die Auswahl von mittlerweile 40 einzelnen Datenpaketen mit jeweils 8Byte, wodurch eine maximale Übertragung von 9 Paketen mit der maximalen einstellbaren Samplingrate von 1kHz erlaubt wird.

$$anz.Pakete = \frac{1\text{Mbit/s}}{(64\text{bit} + overhead) * 1\text{kHz}}$$

Aus Performancegründen bzw. um alle möglichen errechneten Datenpakete schnell und echtzeitfähig übertragen zu können, muss die Anzahl der CAN Schnittstellen erweitert werden. Eine parallele Ausgabe über fünf CAN-Schnittstellen (Abbildung 6) erhöht den Datendurchsatz und bietet eine Flexibilität der externen Analysegeräte.

Die CAN-Kommunikation kann mit Hilfe des Analysetools DEWESoft [9] getestet und dargestellt werden. Abbildung 7 zeigt den Datenverkehr zwischen zwei parallelen CAN-Interfaces mit einer Baudrate von 1Mbit/s.

Eine fehlerfreie Kommunikation bei einem Transfer von 16Bytes konnte mit einer Frequenz von ca. 7,5kHz wiederholt werden. Das Timing wird im weiteren Projektfortschritt vom Navigation-Core durch Vorgaben des Sendezeitpunktes bzw. der Sendedaten vorgegeben.

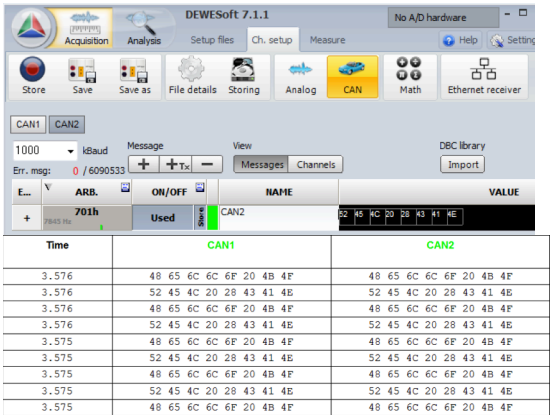


Abbildung 7: Überprüfung der CAN Kommunikation.

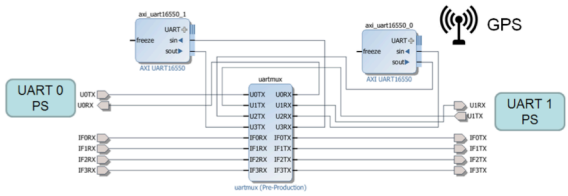


Abbildung 8: Realisierung des UART MUX IP-Core.

C. UART MUX IP-Core

Um genauere GPS-Informationen zu erhalten, kann das System über ein Modem mit Korrekturdaten versorgt werden. Zudem ist es möglich, die Rohdaten des GPS-Empfängers für weitere Analysezwecke auszugeben. Derzeit werden GPS Roh- und Korrekturdaten direkt zwischen dem GPS-Modul und dem GPRS-Modem, über welches die Korrekturdaten empfangen werden, über die UART-Schnittstelle ausgetauscht. Um den UART-Kanal abzugreifen und weiterleiten zu können, kommt ein *UART-Multiplexer* zum Einsatz. Dieser ermöglicht es nahezu beliebige Verbindungen zwischen den im System vorhandenen UARTs und den physikalischen Schnittstellen herzustellen. Zudem können die GPS Roh- und Korrekturdaten dem Navigation-Core für die weitere Verarbeitung zur Verfügung gestellt werden. Die Konfiguration des Multiplexers kann der Benutzter über den Webserver einstellen.

Ein AUX Eingang soll als eine alternative Stützquelle zur Ortung in geschlossen Räumen bzw. Hallen dienen, um z.B. Lokalisierungsdaten als Ersatz für ein fehlendes oder schlechtes GPS Signals zu liefern.

Im Gesamtsystem sind insgesamt vier UART Interfaces eingeplant. Zwei Schnittstellen sind für die GPS-Kommunikation zuständig. Die weiteren zwei Schnittstellen können für die externe Datenausgabe genutzt werden.

FAZIT

Für das Projekt wurde die Zynq-7000 Plattform mit dem Zynq Dual-ARM A9 Prozessor als Bare-Metal / Bare-Metal Asymmetric Multiprocessing (AMP) System konfiguriert. Dabei wurden die Aufgaben der Datenverarbeitung und der Datenausgabe auf die einzelnen Prozessorkerne verteilt. Der Einsatz eines gemeinsam genutzten Speichers erleichtert die Kommunikation zwischen den beiden Prozessorkernen. Zusätzlich sind drei Ethernet- und fünf CAN-Bus-Schnittstellen in programmierbarer Logik realisiert, die eine schnelle und echtzeitfähige Datenübertragung erlauben.

Durch den Gebrauch eines großen FPGAs können weitere funktionale Erweiterungen in Software implementiert oder in programmierbarer Logik umgesetzt werden. In Zukunft sind hier Funktionalitäten wie erweiterte Relativabstandsberechnungen oder Car-2-Car Kommunikation vorgesehen.

DANKSAGUNG

Die Autoren bedanken sich bei der GeneSys Elektronik GmbH, Offenburg, für die Mitarbeit an diesem Projekt, auf der diese Veröffentlichung beruht.

Die Arbeiten wurden u.a. durchgeführt im Rahmen des ZIM Projekts "Kooperatives Lokalisierungssystem zur Relativabstandsberechnung von mehreren Fahrzeugen zum Einsatz bei der Entwicklung von komplexen Fahrerassistenzsystemen (KoRel)" Förderkennzeichen KF2471319ED4, gefördert durch das Bundesministerium für Wirtschaft und Energie.

LITERATURVERZEICHNIS

[1] http://www.genesys-offenburg.de/.

[2] Xilinx, “Zynq-7000 All Programmable SoC Technical Reference Manual”, UG585 (v1.10), Feb. 23, 2015.

[3] http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/mb_ref_guide.pdf.

[4] http://www.analog.com/media/en/technical-documentation/data-sheets/ADSP-21467_21469.pdf.

[5] http://www.xilinx.com/products/design-tools/vivado.html.

[6] Xilinx, “Creating and Packaging Custom IP”, UG1119 (v2015.1), Apr. 01, 2015.

[7] Xilinx, “LogiCORE IP AXI4-Lite IPIF”, PG155 (2.0), Dez. 18, 2013.

[8] https://de.wikipedia.org/wiki/Controller_Area_Network.

[9] http://www.dewesoft.com/.

Alexander Riske erhielt den akademischen Grad des Dipl.-Ing. in Elektro- und Informationstechnik im Jahr 2009 von der Hochschule Offenburg und arbeitet derzeit im Institut für verlässliche Embedded Systems und Kommunikationselektronik (IvESK) an der Hochschule Offenburg.

Manuel Schappacher erhielt den akademischen Grad des Dipl.-Inform. in Technische Informatik an der Fachhochschule Furtwangen im April 2009. Danach arbeitete er als Projektingenieur im Steinbeis Innovation Zentrum für Embedded Design und Networking (SIZEDN) auf dem Gebiet der drahtgebundenen und drahtlosen Kommunikation.

Derzeit arbeitet er im Institut für verlässliche Embedded Systems und Kommunikationselektronik (IvESK) an der Hochschule Offenburg.

Prof. Dr.-Ing. Axel Sikora studierte Elektrotechnik (Dipl.-Ing.) Wirtschaftsingenieurwesen (Dipl. Wirt.-Ing.) an der RWTH Aachen. Er promovierte zum Dr.-Ing. am Fraunhofer-Institut für Mikroelektronische Schaltungen und Systeme in Duisburg mit einer Arbeit über SOI-Technologien.

Nach verschiedenen Positionen in der Telekommunikations- und Halbleiterindustrie wurde er 1999 an die Duale Hochschule Baden-Württemberg Lörrach berufen. Seit 2011 ist er Professor für Embedded Systems und Kommunikationselektronik in Offenburg. Sein Hauptinteresse liegt auf dem Gebiet der effizienten, energiebewussten, autonomen Algorithmen und Protokolle für drahtgebundene und drahtlose eingebettete Kommunikation.

Prof. Dr.-Ing. Sikora ist Autor, Co-Autor, Herausgeber und Mitherausgeber mehrerer Lehrbücher und mehr als 150 Fachartikel auf dem Gebiet der Embedded- Designs für sichere drahtlose und drahtgebundene Netze.

Seit 2015 ist er wissenschaftlicher Leiter des Instituts für verlässliche Embedded Systems und Kommunikationselektronik. Seit 2016 ist es ebenfalls Bereichsleiter „Software Solutions“ und stellv. Institutsleiter bei der Hahn-Schickard-Gesellschaft für Angewandte Forschung e.V.

Compact Modeling of Inkjet Printed, High Mobility, Electrolyte-Gated Transistors

Gabriel C. Marques, Suresh K. Garlapati, Simone Dehm, Subho Dasgupta, Jasmin Aghassi and Mehdi B. Tahoori

Abstract—High mobility, electrolyte-gated transistors (EGTs) show high DC performance at low voltages (< 2 V). To model those EGTs, we have used different models for the below and the above threshold regime with appropriate interpolation to ensure continuity and smoothness over all regimes. This empirical model matches very well with our measured results obtained by the electrical characterization of EGTs.

Index Terms—Electrolyte-gated field-effect transistors, Semiconductor Oxide based devices, Printed electronics

I. Introduction

The field of Printed Electronics (PE) is a promising, young research area where the devices or electronic circuits are typically printed on flexible substrates. The flexible substrates lead to interesting applications, for instance in photovoltaics [1], radio frequency identification (RFID) tags [2], smart sensors [3], wearables [4], etc. In future, printed devices, like transistors, can play a key role within the *Internet of Things*, where computers and sensors are linked together into personal, everyday items [5]. In PE, usually field-effect transistors based on organic materials (OFETs) are used for the circuit design. However, OFETs exhibit low field-effect mobility values and a short lifetime under ambient conditions [6], [7], which finally results in low performable and lifetime-limited circuits.

Electrolyte-gated field-effect transistors (EGTs) based on inorganic semiconductor oxides present themselves as an alternative to the above mentioned OFETs, especially in the low voltage regime. The low temperature required for the process [8], enables the usage of flexible plastic foils as substrates. Furthermore, EGTs with oxide semiconductors exhibit higher field-effect mobility values as well as thermal and long term stabilities even at lower voltage ranges [8] - [11].

A model, which might be used in the design of circuits based on EGTs, must capture the device characteristics accurately. A further requirement is that the modeled slopes must be continuous and smooth over all regimes in order to be integrated into circuit simulators, e.g. SPICE. For this purpose, several groups [12] - [15] have used a universal model [16] for amorphous thin film transistors (a-Si:H TFTs). However, the universal model does not reproduce the behavior of our EGTs and only considers voltage ranges above the threshold voltage V_{TH}. The high number of input parameters, that are necessary for the universal model, are also disadvantageous.

In this context, we use different models for the below and the above threshold regime [17]. The below threshold regime is modeled with the subthreshold swing model [16] and for the linear as well as the saturation regimes, an altered form of the Curtice model, originally developed for gallium arsenide field-effect transistors (GaAs FETs) [17] is employed. Regarding device and circuit simulations, continuity and smoothness are important criterions of a model. To ensure continuity and smoothness over all the regimes, an interpolation scheme is applied between the sub- and superthreshold regions. Before interpolating, the respective input parameters must be extracted from experimental data (output and transfer curves). The parameter extraction is done on a device with a W (channel Width) / L (channel length)-ratio of 1. For other W/L-ratios, the output current must just be multiplied with the W/L-ratio of interest. In case of input parameters change by replacing the materials for instance, the extraction routine must be repeated from beginning. Applying the methodology to our EGTs, the results show very reasonable agreements between measured and modeled curves.

Gabriel Cadilha Marques, gabriel.marques@kit.edu and Mehdi Baradaran Tahoori, mehdi.tahoori@kit.edu are with the Chair of Dependable Nano Computing (CDNC) at the Karlsruhe Institute of Technology (KIT), Haid-und-Neu-Str. 7, 76131 Karlsruhe

Suresh Kumar Garlapati, suresh.garlapati@kit.edu, Simone Dehm, simone.dehm@kit.edu and Subho Dasgupta, subho.dasgupta@kit.edu, are with the Institute of Nanotechnology (INT) at the Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen

Jasmin Aghassi, jasmin.aghassi@kit.edu is with the Institute of Nanotechnology (INT) at the Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen and the Hochschule Offenburg, Badstraße 24, 77652 Offenburg

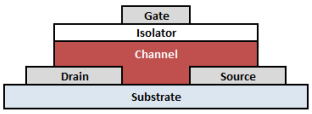


Figure 1: Cross-section view of an organic field-effect transistor

The organization of this paper is as follows. In Section 2, the technology used for our EGTs and the device characteristics are presented. Section 3 describes the modeling methodology. In Section 4 the accuracy of the model is shown on a modeling example. Finally, Section 5 concludes the paper.

II. Technology

Organic p-type transistors are frequent but n-type organic transistors are still rare in Printed Electronics. A popular architecture of an OFET is known as the top-gate bottom-contact configuration (Figure 1). In the top-gate bottom-contact configuration an organic material, e.g. poly(3-hexylthiophen-2.5diyl)(P3HT), acts as channel between two metal electrodes (source and drain). On top of the channel follows an organic or inorganic insulator, which is sandwiched between the channel and a further metal electrode (gate). While the field-effect mobility is improved from $< 10^{-3}$ $cm^2(Vs)^{-1}$ to 1 $cm^2(Vs)^{-1}$, since the first reported OFET in 1989, the typical supply voltages are still quite high (>> 20 V) .

To overcome the low field-effect mobility and the high voltage, which is a requirement for OFETs to perform well, an electrolyte-gated transistor based on an inorganic semiconductor oxide was proposed in [11]. Instead of a p-conducting channel, the EGT has an n-conducting channel, which is based on indium oxide (In_2O_3). Comparing EGTs with OFETs, the field-effect mobility of the semiconductor oxide based EGT (126 $cm^2(Vs)^{-1}$ in [11]) exceeds those of OFETs by orders of magnitude. The voltage range required for the EGTs ($\leq$ 2 V) is also smaller than the voltage range of a typical OFET, which makes it suitable for low-voltage applications.

Therefore, the transistor channel is based on an inorganic semiconductor and a solid polymer electrolyte forms the gate dielectric. The electrolyte-gating allows operating the devices at lower voltage ranges. By applying a positive potential on the gate, anions from the electrolyte are attracted to the gate-electrolyte interface. At the same time, cations migrate to the electrolyte-semiconductor interface, which causes accumulation of electrons inside the channel at the electrolyte-semiconductor interface. The accumulation of charges and ions at each interface leads to two electrical double layers (EDLs), known as the Helmholtz double layer (Figure 2) [20]. The formation of a

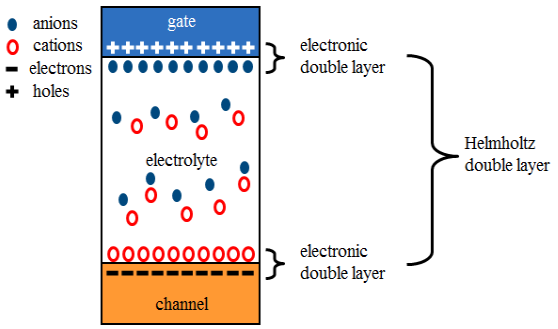


Figure 2: Formation of a Helmholtz double layer by applying a positive voltage to the gate

Helmholtz double layer results in a high gate capacitance (4.33 µF/cm^2 in [11]). The high gate capacitance in combination with high mobility channel materials results in high DC-performable devices [10], [11], [20]-[22].

Besides the high DC performance of EGTs, the device fabrication is very straightforward. First, the passive structures (source, drain and gate) are structured by lithography or laser ablation on a substrate. After structuring the electrodes, the channel is printed between source and drain with an inkjet printer. Then, the electrolyte is printed in a way that the channel and partially the gate electrode are covered by the electrolyte. In the last step (PEDOT:PSS), a conducting polymer, is printed on top of the electrolyte as top-gate.

III. Compact Modeling

For the silicon technology, a broad spectrum of transistor models are available. Some of the models, developed for silicon based transistors, can be adopted to model the printed transistors [12], [14], [23]. Within the overall range of transistor models, a universal model (AIM-SPICE level 15 model [24] or HSpice Level 61 model [25]) for amorphous thin film transistors (a-Si:H TFTs) [16] has gained a lot of attention. The popularity of the model is related to the fact that printed transistors have some similarities with a-Si:H TFTs [14], e.g. both are accumulation based devices. However, the AIM-SPICE level 15 model has a high number of input parameters, which first need to be extracted. In [12], [14], the unified model and parameter extraction method (UMEM) [26] is utilized to extract the parameters for the AIM-SPICE level 15 model, applied on an OFET. When we have employed UMEM to our inorganic oxide based EGTs, an accurate match between the modeled and the measured results is not possible. Therefore our objective has been to model EGTs with an accurate model, which has at the same time the lowest number of possible input parameters. Additionally, to avoid convergence problems in circuit simulation, the model behavior must be continuous and smooth.

Table 1: Summary of extracted input parameters needed for the electrolyte-gated transistor model

Parameters	Below Threshold	Near Threshold	Above Threshold
$Dn\ [cm^2s^{-1}]$	$33.16e^{-3}$	-	-
$I_0\ [A]$	$1.5e^{-10}$	-	-
$n_0\ [cm^3]$	$1.10e^{9}$	-	-
a	-	$213.42e^{-6}$	-
b	-	$-185.88e^{-6}$	-
c	-	$55.32e^{-6}$	-
d	-	$-5.58e^{-6}$	-
$\alpha\ [V^{-1}]$	4	4	4
$\lambda\ [V^{-1}]$	-	-	$50e^{-3}$
$\beta\ [A/V^2]$	-	-	$265e^{-6}$
$V_{TH}\ [V]$	-	-	0.45

To overcome the issues of UMEM, we used a model proposed in [17]. The model is based on the subthreshold swing model [16] for the below threshold regime and a modified version of the Curtice model [17] for the above threshold regime. A cubic interpolation in-between the subthreshold swing and the Curtice model ensure continuity and smoothness over all regimes.

Independent from the *W/L*-ratio of interest, it is recommended to start to model a device with a $W/L = 1$ and afterwards to multiply the output current with the respective *W/L*-ratio.

From [17], the subthreshold regime is modeled with the subthreshold swing model [16]:

$$I_{DS} = \frac{W}{L} q \cdot D_N \cdot n_0 \cdot e^{\frac{V_{GS}}{D_n}} \cdot \tanh(\alpha V_{DS}) + I_0,$$

where W is the channel width, L is the channel length, q is the elementary charge of an electron, D_n is the diffusion coefficient, n_0 is the electron density, α is the fitting parameter describing the knee region, V_{DS} is the drain-source voltage and V_{GS} is the gate-source voltage.

Next, the above threshold regime is modeled with the Curtice model:

$$I_{DS} = \beta(V_{GS} - V_{TH})^2 \cdot \tanh(\alpha V_{DS}) \cdot (1 + \lambda V_{DS}),$$

$$\text{with } \beta = \frac{W}{L} \cdot \mu_{FET} \cdot C_{Gate},$$

where V_{TH} is the threshold voltage, λ is the channel length modulation parameter, μ_{FET} is the field-effect mobility and C_{Gate} is the gate capacitance.

As final step, a cubic interpolation between the below and the above threshold regions ensures continuity and smoothness over all regimes:

$$I_{DS} = (aV_{GS}^3 + bV_{GS}^2 + cV_{GS} + d) \cdot \tanh(\alpha V_{DS}) \cdot (1 + \lambda V_{DS})$$

IV. Results and Validation with Measurements

The experimental data is based on an electrolyte-gated transistor with a printed indium oxide In_2O_3 channel and a *W/L*-ratio of 1. The electrolytic insulator and the PEDOT:PSS (top gate) have also been printed. The EGTs have been electrically characterized with a probe station and a parameter analyzer. In a second step we extracted the model parameters with the proposed methodology in [17] and summarized the results in Table 1. The simulated IV curves and transfer characteristics show a very good agreement with our measurement results (Figure 3). As the model behavior is continuous and smooth, it is possible to implement the model in a hardware description language (HDL), e.g. Verilog-A, and to use it for circuit simulation. To verify the usability of the model for circuit simulation, an inverter consisting of an EGT and a resistor is simulated at 1 V supply voltage (V_{DD}) (Figure 4). The EGT in the pull down network has a *W/L*-ratio of 1 and the resistor in the pull up network has a resistance of 200 kΩ. In Figure 4 the input voltage V_{in} of the inverter is swept from 0 V to 1 V and the output voltage V_{out} is also reaches from 0 V to 1 V.

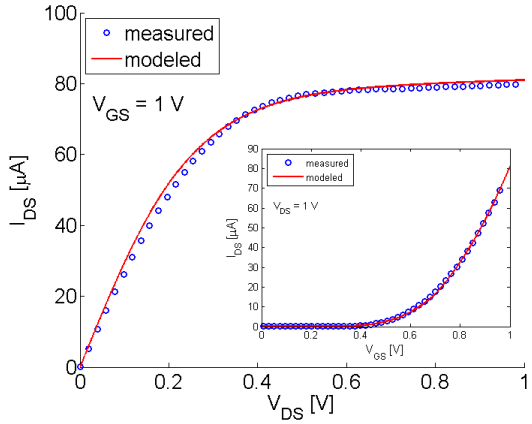


Figure 3: Measured drain-source current versus modeled drain-source current in dependence from the drain-source voltage and the gate-source voltage

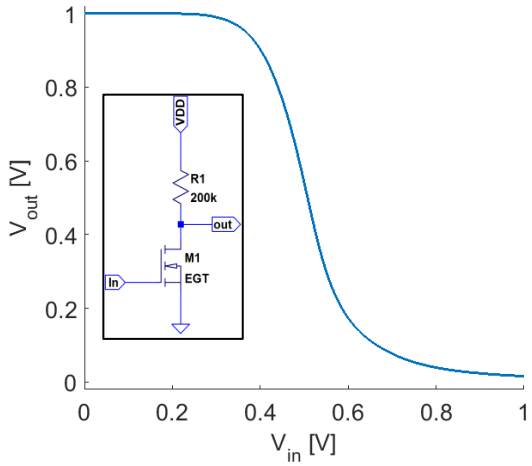


Figure 4: Simulation of an inverter with a resistor in the pull-up network and EGT in the pull-down network

Electrolyte-gated field-effect transistors based on inorganic semiconductor oxides show high DC performance compared with their organic counterparts. In this work, we modeled the DC behavior of such EGTs with different models corresponding to the transistor regimes. The model is implemented with Verilog-A and shows a very good agreement with experimental data. The usefulness of the model for circuit design is shown on the simulation of an inverter.

ACKNOWLEDGEMENT

This work is financially supported by the Helmholtz Gemeinschaft in kind of the Helmholtz Virtual Institute VI530.

REFERENCES

[1] H. Youn, H. J. Park, and L. J. Guo, “Printed nanostructures for organic photovoltaic cells and solution-processed polymer light emitting diodes”, Energy Technology, vol. 3(4), pp. 340-350, 2015.

[2] S. Zhang, S. Li, S. Cheng, J. Ma, and H. Chang, “Research on smart sensing rfid tags under flexible substrates in printed electronics,” in 16th International Conference on Electronic Packaging Technology (ICEPT), 2015, pp. 1006–1009.

[3] H. Marien, M. Steyaert, and P. Heremans, Analog Organic Electronics - Building Blocks for Organic Smart Sensor Systems on Foil. Springer New York Heidelberg Dordrecht London, 2012.

[4] H. F. Castro, E. Sowade, J. G. Rocha, P. Alpuim, A. V. Machado, R. R. Baumann, and S. Lanceros-Mèndez, “Degradation of all-inkjet printed organic thin-film transistors with tips-pentacene under process applied in textile manufacturing,” Organic Electronics, vol. 22, pp. 12 – 19, 2015.

[5] P. Rosa, A. Câmara, and C. Gouveia, “The potential of printed electronics and personal fabrication in driving the internet of things,” Open Journal of Internet of Things, vol. 1(1), pp. 16 – 36, 2015.

[6] Z. Bao and J. Locklin, Organic Field-Effect Transistors. CRC Press, Boca Raton, FL 2007.

[7] S. K. Garlapati, T. T. Baby, S. Dehm, M. Hammad, V. S. K. Chakravadhanula, R. Kruk, H. Hahn, and S. Dasgupta, “Inkjet printed cmos electronics from oxide semiconductors,” Small, vol. 9(3), pp. 3075 – 3083, 2015.

[8] T. T. Baby, S. K. Garlapati, S. Dehm, M. Häming, R. Kruk, H. Hahn, and S. Dasgupta, “A general route toward complete room temperature processing of printed and high performance oxide electronics,” ACS Nano, vol. 11(29), pp. 3591 – 3596, 2015.

[9] B. Nasr, D. Wang, R. Kruk, H. R ̈osner, H. Hahn, and S. Dasgupta, “High-speed, low-voltage, and environmentally stable operation of electrochemically gated zinc oxide nanowire field-effect transistors,” Advanced Functional Materials, vol. 23(14), pp. 1750 – 1758, 2013.

[10] S. Dasgupta, G. Stoesser, N. Schweikert, R. Hahn, S. Dehm, R. Kruk, and H. Hahn, “Printed and electrochemically gated, high-mobility, inorganic oxide nanoparticle fets and their suitability for high-frequency applications,” Advanced Functional Materials, vol. 22(23), pp. 4909 – 4919, 2012.

[11] S. Garlapati, N. Mishra, S. Dehm, R. Hahn, R. Kruk, H. Hahn, and S. Dasgupta, “Electrolyte-gated, high mobility inorganic oxide transistors from printed metal halides,” Applied Materials & Interfaces, vol. 5(22), pp. 11 498 – 11 502, 2013.

[12] M. Estrada, A. Cerdeira, J. Puigdollers, L. Reséndiz, J. Pallares, L. F. Marsal, C. Voz, and B. Iñiguez, “Accurate modeling and parameter extraction method for organic tfts,” Solid-State Electronics, vol. 49(6), pp. 1009 – 1016, 2005.

[13] L. Reséndiz, B. Iñiguez, M. Estrada, and A. Cerdeira, “Modification of amorphous level 15 aim spice model to include new subthreshold model,” in 24th International Conference on Microelectronics (MIEL), vol. 1, 2004, pp. 291–294.

[14] O. Yaghmazadeh, Y. Bonnassieux, and A. Saboundji, “A spice-like dc model for organic thin-film transistors,” Journal of the Korean Physical Society, vol. 54(1), pp. 523 – 526, 2009.

[15] X. Zhang, T. Ge, and J. S. Chang, “Fully-additive printed electronics: Transistor model, process variation and fundamental circuit designs,” Organic Electronics, vol. 26, pp. 371 – 379, 2015.

[16] K. Lee, M. Shur, T. Fjeldly, and T. Ytterdal, Semiconductor Device Modeling for VLSI. New Jersey: Prentice Hall, 1993.

[17] G. C. Marques, S. K. Garlapati, D. Chatterjee, S.Dehm, S. Dasgupta, J. Aghassi and M. B. Tahoori, submitted

[18] W. R. Curtice, “A mesfet model for use in the design of gaas integrated circuits,” IEEE Transactions on Microwave Theory and Techniques, vol. 28(5), pp. 448 – 456, 1980

[19] H. Sirringhaus, “25th anniversary article: Organic field-effect transistors: The path beyond amorphous silicon,” Advanced Materials, vol. 26(9), pp. 1319 – 1335, 2014.

[20] S. H. Kim, K. Hong, W. Xie, K. H. Lee, S. Zhang, T. P. Lodge, and C. D. Frisbie, “Electrolyte-gated transistors for organic and printed electronics,” Advanced Materials, vol. 25, pp. 1822 – 1846, 2013.

[21] M. J. Panzer, C. R. Newman, and C. D. Frisbie, “Low-voltage operation of a pentacene field-effect transistor with a polymer electrolyte gate dielectric,” Applied Physics & Letters, vol. 86(10), pp. 1 – 3, 2012.

[22] K. Hong, S. H. Kim, K. H. Lee, and C. D. Frisbie, “Printed, sub-2v zno electrolyte gated transistors and inverters on plastic,” Advanced Materials, vol. 25(25), pp. 3413 – 3418, 2013.

[23] J. Krumm, “Circuit analysis methodology for organic transistors,” Ph.D. Thesis, Universität Erlangen-Nürnberg, 2008.

[24] Aim-spice, simulator for device and circuit simulation. [Online]. Available: http://aimspice.com

[25] Hspice, simulator for device and circuit simulation. [Online]. Available: http://synopsys.com

[26] A. Cerdeira, M. Estrada, R. García, A. Ortiz-Conde, and F. G. Sánchez, “New procedure for the extraction of basic a-si:h tft model parameters in the linear and saturation regions,” Solid-State Electronics, vol. 45, pp. 1077 – 1080, 2001.

Gabriel Cadilha Marques received the B.Eng. degree in Electrical Engineering in 2012 and the M.Eng. degree in Electronic and Mechatronic Systems in 2014 from the Technische Hochschule Nürnberg Georg Simon Ohm. He is currently a PhD student working on the field of printed electronics at the Chair of Dependable Nano Computing (CDNC) in cooperation with the Institute of Nanotechnology (INT) both located at the Karlsruher Institute of Technology (KIT).

Suresh Kumar Garlapati received his Bachelor of Technology (B.Tech) degree in Metallurgical Engineering in JNTU Hyedrabad, India, in 2009. Then, he joined the Indian Institute of Technology (IIT) Madras and received his M.Tech degree in Metallurgical & Materials Engineering in 2011. After that, he is pursuing the PhD degree in Materials Science in Technical University of Darmstadt and Karlsruhe Institute of Technology. His research interests include printed electronics, oxide semiconductors, electrolyte gating, metal-insulator transition studies.

Simone Dehm finished her apprenticeship as lab assistant in physics at the Institute for Mechanical Process Engineering and Mechanics (KIT) in 1989. From 1989 until 2003 she worked as a Lab Technician at the Integrate Optics Technologies – JDS Uniphase Photonics. Since 2006 she is working as a lab technician in the research unit Charge transport and light-matter interaction in carbon nanosystems of Prof. R. Krupke at KIT in the Institute of Nanotechnology (INT).

Subho Dasgupta received his PhD degree in Material Science from Technical University of Darmstadt (TUD), Germany in 2009. Presently, he is working as a group leader at Karlsruhe Institute of Technology (KIT), Germany. In 2013-14 he was a visiting scientist at Lawrence Livermore National Laboratory, USA. His research interests include printed oxide electronics, nanomagnetism, electrochemistry, Li-ion batteries, electronic transport in nanoporous and bulk carbon species etc. He has authored/co-authored 20 research papers, and filed 8 US/European patent applications which are either granted or under review.

Jasmin Aghassi studied physics at the RWTH Aachen and received her PhD from Karlsruhe University (today KIT) in nano-electronics. Afterwards she joined Infineon Technologies AG in Munich as a research and development engineer focusing on CMOS low power development for wireless applications. In this function she worked as a short time delegate in the IBM Semiconductor Alliance ISDA) in East Fishkill, USA, as a modeling and device expert. In 2011 she moved on to Intel Mobile Communications as a Technology Platform Manager. She was hired as executive director of the R&D network NanoMat at KIT in 2012 and was additionally appointed in 2013 as a full Professor in Electrical Engineering at the University of Applied Sciences Offenburg. Her research interests are low power electronics, novel nano devices and printed electronics.

Mehdi Tahoori is professor and Chair of Dependable Nano-Computing (CDNC) at Karlsruhe Institute of Technology (KIT) in Germany since 2009. Before that he was an associate professor of ECE at Northeastern University, Boston, USA. He received his Ph.D. and M.S. in Electrical Engineering from Stanford University in 2003 and 2002, respectively, and B.S. in Computer Engineering from Sharif University, Iran, in 2000. He has published more than 200 conference and journal papers and holds several patents on various aspects of emerging technologies for computing and resilient system design. He is on the organizing and technical program committee of various design automation, test, and reliability conferences and workshops such as ITC, VTS, ETS, DATE, DAC, ICCAD, ASP-DAC, IOLTS, GLSVLSI, VLSID, SELSE. He is an associate editor of ACM Journal of Emerging Technologies for Computing and IEEE Design and Test Magazine. He was the recipient of National Science Foundation CAREER Award. He has received a number of best paper nominations and awards at various conferences.

A fully integrated ASIC for energy harvesting from glucose biofuel cell

Narayanan Seetharaman, Harald Richter and Joachim N. Burghartz

***Abstract*—Current glucose biofuel cells are able to generate an output power density in the range of 200 µWcm^{-2}. While this power density meets the requirements of common ASICs, unfortunately the voltage level is not sufficient. Unlike solar cells, the glucose biofuel cells cannot be connected in series to generate a higher output voltage level, as they share the same glucose medium. In this paper, a fully integrated low power energy harvesting ASIC without any external components is proposed, which overcomes this limitation by up-converting the low input voltage (0.4 V - 0.7 V) to a usable output voltage of more than 1.8 V and powers on-chip ring oscillator based temperature and voltage sensors. The proposed ASIC is implemented in 500 nm IMS gate array standard CMOS technology, with devices having threshold voltages higher than the input voltage. For up-conversion, an inductor-free, 2-stage integrated charge pump based DC/DC up-converter is used. An outlook to possible design improvements using a special low-power technology is given.**

***Keywords*—Glucose biofuel cell, IMS chips, DC/DC converter**

I. Introduction

Recent developments in smart sensors and Internet of Things (IoT) devices have been fast and diverse. But yet, they have not found application as body implants. The major obstacle for this is an old issue pertaining from the early days of the first implanted pacemaker device in 1960, i.e. the need for an everlasting and reliable source of electrical energy within a human body. Various sources and techniques of energy harvesting have been developed towards resolving this issue, such as energy harvesting from temperature difference, vibration, heartbeat, muscle movements and others [1,2]. One such energy harvesting technique exploits the body glucose, using Glucose Bio Fuel Cell (GBFC), where electrical energy is generated as a result of the following chemical reactions.

Narayanan Seetharaman, st111675@stud.uni-stuttgart.de and Joachim N. Burghartz, burgh@ims-chips.de, Institut für Nano- und Mikroelektronische Systeme (INES), Universität Stuttgart.

Harald Richter, richter@ims-chips.de, Institut für Mikroelektronik Stuttgart – IMS Chips, Allmandring 30a, 70569 Stuttgart.

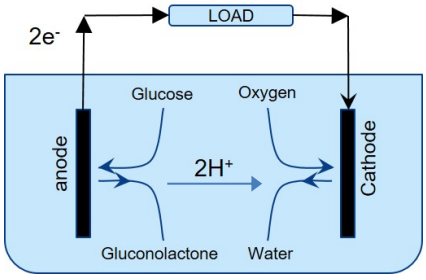


Figure 1: Glucose Bio Fuel Cell

Glucose Oxidation : $C_6H_{12}O_6 + H_2O \rightarrow C_6H_{12}O_7 + 2H^+ + 2e^-$
Oxygen Reduction : $O_2 + 2H^+ + 2e^- \rightarrow H_2O$

As shown in Figure 1, in a glucose & oxygen rich solution, the glucose oxidation takes place at the anode coated with glucose oxidase and oxygen reduction happens at the cathode coated with bilirubin oxidase. In this process two electrons are absorbed at the anode and discharged at the cathode. When a conduction path is established externally, continuous reaction is achieved with constant consumption of glucose and oxygen. As glucose and oxygen are constantly replenished in the human body and the byproducts gluconolactone & water are harmless to the human body, the GBFC provides a promising solution as a continuous and reliable source of energy. A successful implementation of this technique was carried out on a rat, where a maximum generated power density of 192 µWcm^{-2} is reported [2], which is more than a pacemaker's power requirement (~10 µW).

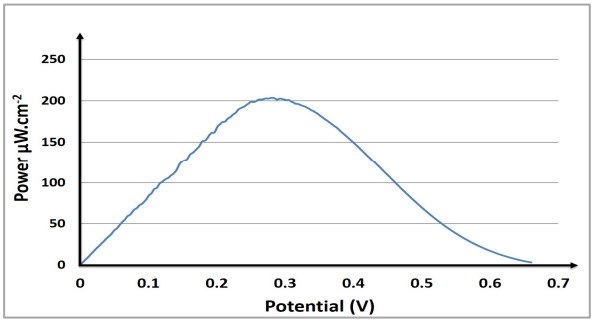


Figure 2: GBFC output power vs output voltage

In another experiment, an attempt to integrate the GBFC's electrodes directly on the surface of a Si (100) wafer was carried out at Max-Planck-Institut für Intelligente Systeme, Stuttgart. The power output versus potential for this experiment is shown in Figure 2. As shown, the output power level is satisfactory, but the voltage range is not useful for directly powering an application in a standard CMOS technology. Hence, a DC/DC converter is needed to up-convert it to a more usable voltage level.

This paper describes the design implementation of a fully integrated ASIC, to demonstrate a standalone bare die based implant using a GBFC as the power source. As such, GBFC's electrodes, a DC/DC up-converter and temperature & voltage sensors as sample applications, are all incorporated in the designed ASIC.

The structure of this paper is the following. Section II provides an overview of available DC/DC up-converter architectures and their feasibility with below threshold voltage input sources. In Section III, the design implementation of the proposed ASIC in IMS 500 nm technology is presented. And in Section IV the simulation results of the design are summarized. Finally, Section V draws conclusion of feasible design enhancements via a comparative study of design implementations in XFAB 180 nm and customized IMS 500 nm twin well technologies.

II. Design Architectures

Since this ASIC has to be used as an implanted device, the usage of external components is not feasible. Also on-chip inductors would require huge area and additional metal layers to achieve high values and efficiencies. Therefore, the available techniques for voltage up conversion are limited to switched capacitive converters, which can be implemented using on-chip capacitors.

A. Dickson charge pump

Dickson first proposed a voltage up conversion design [3], in which the charge is pumped from one node to the next, with the assistance of non-overlapping clock signals (CLK1 and CLK2). Such that, with each stage a voltage gain of ΔV is achieved. When diode connected NMOS are used to realize the diodes, as shown in the Figure 3, the gain (ΔV) is given by

$$\Delta V = V\text{clk} - V\text{th} - V\text{c} \qquad (1a)$$

$$\text{where } V\text{c} = \frac{\text{Iload}}{f\text{clk}(C + Cs)} \qquad (1b)$$

This design suffers from two major limitations. First, due to the body bias effect, the maximum number of stages that can be connected in cascade to achieve higher conversion ratio is limited. Second, the per stage gain (ΔV) depends on Vth.

For any upconversion to happen, ΔV should be positive. Meaning, $V\text{clk} - V\text{th} - V\text{c} > 0$. Considering a no load scenario ($V\text{c} = 0$) with clock signals powered by the input source ($V\text{clk} = V\text{in}$), the per stage gain (ΔV) is given as $\Delta V = V\text{in} - V\text{th}$. This places a lower limit on the acceptable input voltage value, i.e. $V\text{in} - V\text{th} > 0$ or $V\text{in} > V\text{th}$. Various design techniques have been proposed to eliminate this limitation, by making ΔV independent of $V\text{th}$. Two such designs are NCP2 and cross-connected devices.

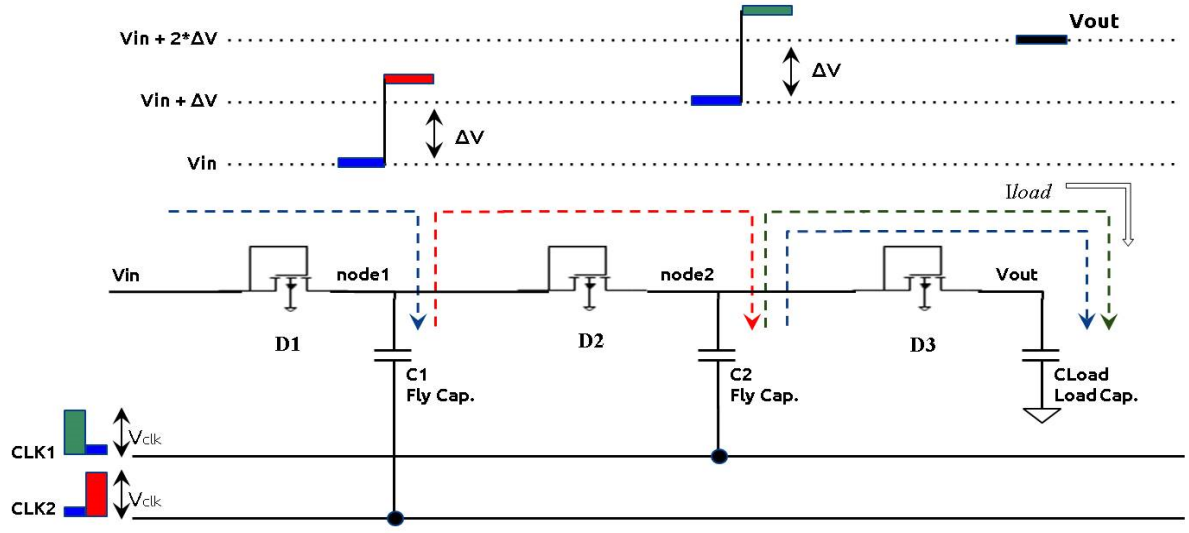


Figure 3: Switched Capacitor charge pump DC/DC converter

B. NCP2

In the New Charge Pump 2 (NCP2) [3] design, a dynamic Charge Transfer Switch (CTS) is used in place of diode connected NMOS. This switch remains active throughout the charging phase, irrespective of the device threshold voltage. This is achieved using backward control of the CTS with successive stage's higher voltage. As shown in Figure 4, here the current from the input has two paths to charge the fly capacitor. The first one is through the diode connected NMOS MN1, which remains turned 'ON' only till the output rises up to $Vin - Vth$. Whereas the second path is via MN2, which remains turned 'ON' throughout the charging phase of clock. Hence, the per stage gain (ΔV) is independent of Vth, i.e. $\Delta V = Vclk - Vc$. Consider the functioning of the nth stage. During the charging phase, the input node, the output node and the inverter power supply are Vn, Vn and $Vn + 2\Delta V$, respectively. I.e. MIN is turned 'OFF' while MN2 & MIP are turned 'ON' with $Vgs = 2\Delta V$.

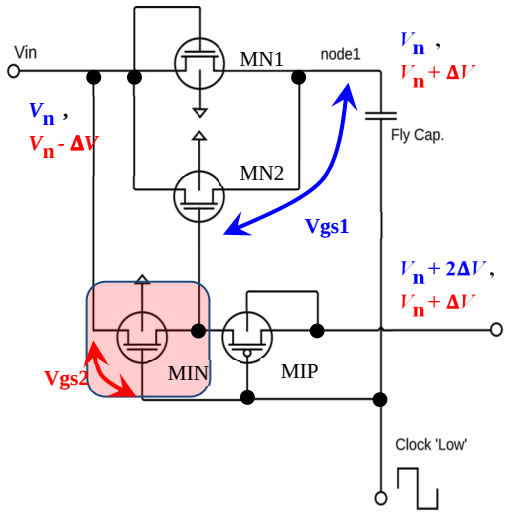


Figure 4: Dynamic charge transfer switch

While discharging, these nodes are $Vn - \Delta V$, $Vn + \Delta V$ and $Vn + \Delta V$, respectively. I.e. MIN is turned 'ON' with $Vgs = 2\Delta V$ while MN2 & MIP are turned 'OFF'. For efficient functioning of this CTS, $2\Delta V > Vth$ has to be fulfilled. As a special case, consider the input stage, where the input node remains fixed at Vin. Hence during the discharge phase, NMOS MIN has an $Vgs = \Delta V$ only to turn 'ON' and in turn set MN2 to 'OFF'. I.e. ΔV should be higher than the Vth to avoid backward charge sharing. This introduces a lower limit on the acceptable input voltage value. Such that, for a no load scenario ($Vc = 0$) with clock signals powered by the input source ($Vclk = Vin$), the per stage gain $\Delta V = Vclk = Vin$ should be greater than Vth.

C. Cross connected devices

Another class of charge pump specifically suited for input sources with voltage level below Vth is based on stages with cross connected devices [5], as shown in the Figure 5. This design removes Vth limitation from both the per stage gain ΔV as well as the minimum input voltage. Hence ΔV is given by $Vclk - Vc$ and the acceptable minimum $Vin \geq 0$. Consider MN1 and MN2 as shown in Figure 5. During charging phase of the fly capacitor C1, a maximum Vgs of $Vin + \Delta V$ is available across MN1. Due to this, C1 starts charging up and effectively reducing the Vgs to ΔV. During this clock phase MN2 is completely 'OFF', i.e. no backward charge sharing occurs.

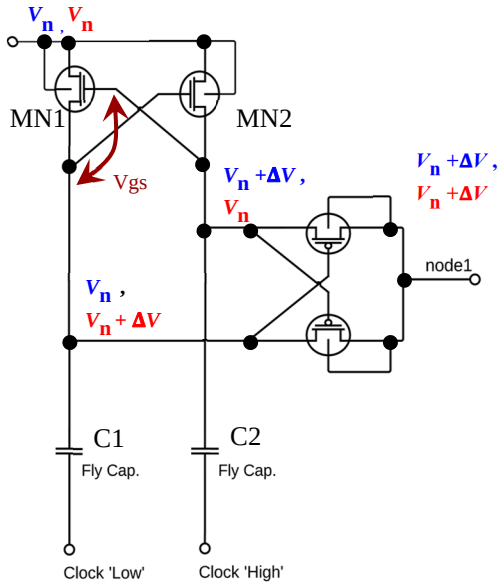


Figure 5: Cross connected device stage

During the next clock phase MN2 is turned 'ON' & MN1 is 'OFF' and C2 is charged to $Vin + \Delta V$. Hence for alternate clock phases, C1 & C2 are pumped up to $Vn + \Delta V$. The cross connected PMOS passes the pumped up $Vn + \Delta V$ to the output using the same working principle. Since the pumped up voltage is available in both the clock phases it can be directly used as input for the next stage and higher conversion ratios can be obtained with successive cascading.

As compared to Dickson's charge pump, where the minimum Vgs across diode connected NMOS at near full charge of fly capacitor C1 is zero, in this design the minimum Vgs across NMOS MN1 is ΔV. Hence for this design, the subthreshold conduction current (Isub) as given by (2), is comparatively higher than that of Dickson's charge pump. This advantage enables this design for application with inputs below device's threshold voltage.

$$Isub = \left(\frac{W}{L}\right)\mu_e C_{ox}\left[\frac{KT}{q}\right]^2 (n-1) * e^{\frac{q(V_{gs}-V_{th})}{nKT}}\left[1-e^{-\frac{qV_{ds}}{KT}}\right] \quad (2)$$

III. IMPLEMENTATION

The proposed ASIC consists of a DC/DC converter to up-convert the GBFC's low input voltage range of 0.5 V to 0.7 V to a more usable voltage range of 1.5 V to 2.1 V.

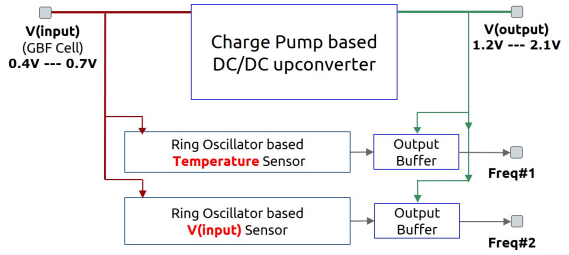


Figure 6: Designed ASIC's top level block diagram

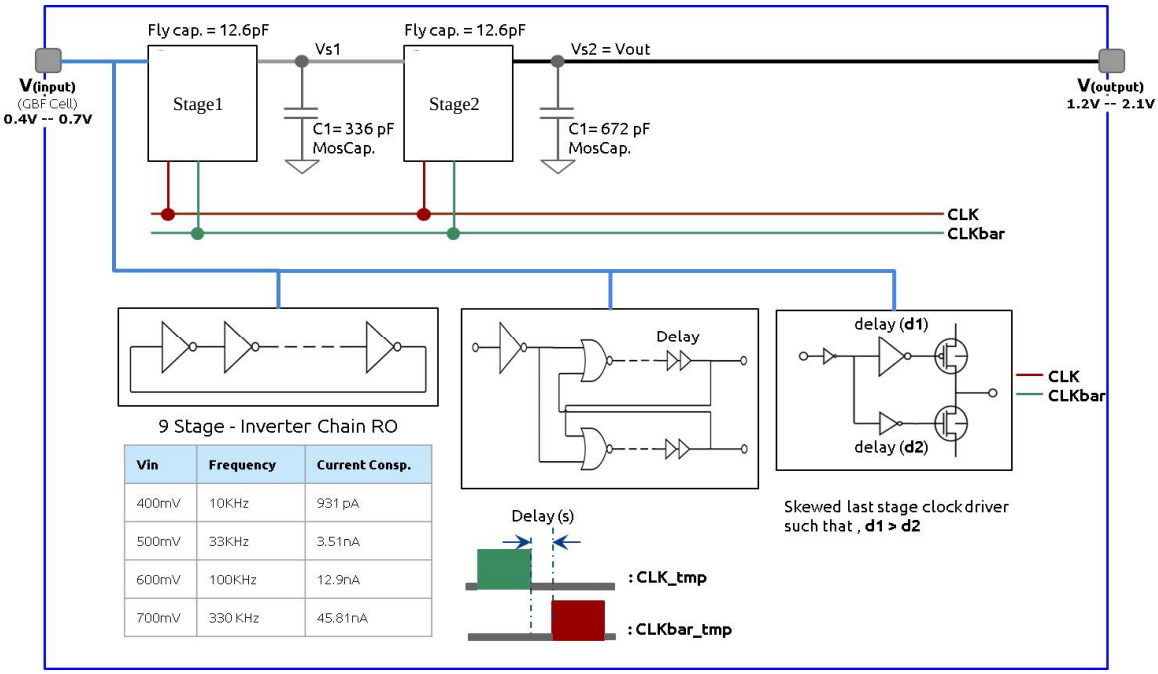
Since the on-chip temperature and voltage sensors are directly powered by the input voltage, their output signal voltage levels are too weak for off chip usage. Hence the internal DC/DC converter's output voltage is used to level shift & drive them to the output pads as shown in Figure 6. This design is customized for 500 nm IMS gate array technology.

A. Electrode

The GBFC's electrodes are layouted with an additional metal layer (Au) with an area of 0.8 x 2.4 mm^2, each and are directly connected to DC/DC converter's input.

B. DC/DC Converter

Since the input voltage range is lower than the threshold voltage of the available devices in the IMS technology, a two stage DC/DC converter, based on cross connected device architecture is implemented. Where, fly capacitors of value 12.6 pF are used along with additional load capacitances of 336 pF and 672 pF. These additional load capacitors are formed using the unused MOSs and are added at the interstage outputs to help reduce the output voltage ripple. The clock signal for DC/DC converter's operation is internally generated using a 9 stage inverter chain Ring Oscillator (RO). The design of RO is optimized such that, the subthreshold current of cross connected devices in DC/DC converter's stages get sufficient time to fully charge the fly capacitors. For high DC/DC converter efficiency, the clock phases CLK and CLKbar need to be perfectly non-overlapping. This is achieved using a NOR based bi-stable latch, as shown in Figure 7. Since the clock signals have to drive 25.2 pF capacitance each, within the oscillation time period, they are driven by an output buffer. To avoid power loss due to transition currents, the output buffer is implemented using skewed CMOS inverters, where NMOS is turned off first and then PMOS is turned on.



Vin	Frequency	Current Consp.
400mV	10KHz	931 pA
500mV	33KHz	3.51nA
600mV	100KHz	12.9nA
700mV	330 KHz	45.81nA

Figure 7: Block diagram of the designed DC/DC core

C. Temperature Sensor

Consider MOS pair MP and MN, as shown in Figure 8, where NMOS MN is permanently switched 'OFF'. Let the input of MP initially undergo a transition from 'Low' to 'High'. Therefore MP will initially be 'ON' and output node Vout will be fully charged up to supply voltage Vin. When the input turns 'High', MP is turned 'OFF'. And both MN and MP are turned 'OFF', i.e. subthreshold leakage currents become effective [6]. Since Isubn continuously tries to discharge the output node and Isubp tries to recharge it back to the supply voltage, the output node will stabilize to a voltage Vtran, where both currents are equal. If by design optimization it is ensured that Isubn >> Isubp, so that Vtran is near to GND, it can be said that the output node undergoes a transition from 'High' to 'Low'. Hence from the input node to the output node an effective inverter is realized, whose delay is proportional to the Isubn. Also during discharge, the Vds across NMOS is initially equal to power supply (Vin) and gradually decreases to Vtran. Since both Vin and Vtran are normally larger than $4KT/q$, Isubn given by (3) becomes effectively independent of the supply voltage.

$$I_{\text{sub\_leakage}} = \left(\frac{W}{L}\right)\mu_e C_{ox}\left[\frac{KT}{q}\right]^2 (n-1)e^{\frac{-qV_{th}}{nKT}} \quad (3)$$

If such an inverter is used to form a RO, then the frequency of oscillation will in turn be proportional to Isubn. Since Isubn strongly depends on the temperature and is independent of Vin, it can be said that, the frequency of oscillation depends on the temperature and is independent of the supply voltage.

Freq#1 = function{Temperature} (4)

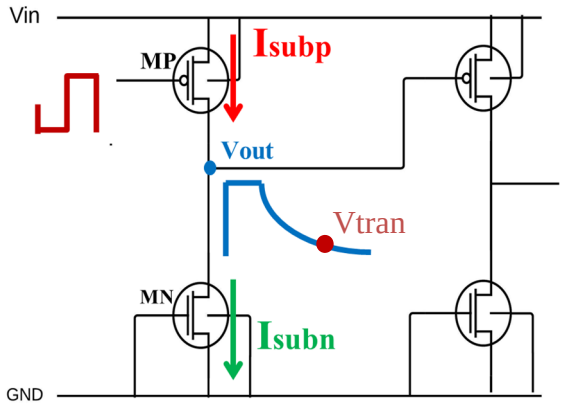


Figure 8: Schematic of ring oscillation based temperature sensor

D. Voltage Sensor

For a current starved ring oscillator as shown in Figure 9, the oscillation frequency is directly proportional to the current through the inverter branches, which are mirrored from Ibias of the bias circuit. In the bias circuit the current Ibias is given by the equation:

$$\text{Ibias} = \left(\frac{W}{L}\right)\mu_e C_{ox}\left[\frac{KT}{q}\right]^2 (n-1) * exp^{\frac{(Vin - Vth - Ibias*R)}{nKT}} \quad (5)$$

I.e. Ibias is a function of both temperature and supply voltage (Vin). Since the RO's frequency directly depends on the branch current, it can be defined as:

Freq#2 = function{Temperature, Voltage} (6)

By utilizing the temperature information from the temperature sensor mentioned in Section III-C, this sensor can be effectively used to evaluate the supply voltage Vin.

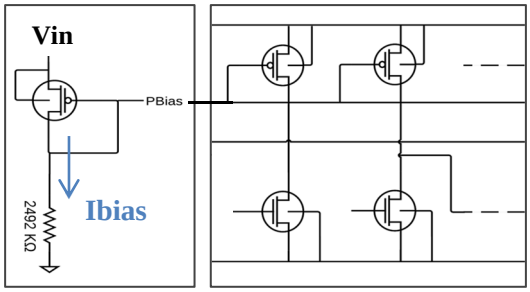


Figure 9: Schematic of bias circuit and ring oscillator based voltage sensor

E. Signal Driver

The signal output of these sensors have the same voltage range as the supply voltage. Hence for low Vin values, it is too weak to be connected directly to the output pads. Therefore the output signals are driven through an output driver, comprising a level shifter followed by an output buffer. This output driver is powered by the DC/DC converter's up-converted output voltage. In order to lightly load the DC/DC converter, the signal frequency and duty cycle is drastically reduced before feeding them into the output driver. For the voltage sensor's output signal, which has the frequency of oscillation (Freq#2) in the range of 120 kHz, it is first divided by a factor of 256 using a frequency divider. Then a short pulse of width equal to 4*(1/Freq#2) is generated using a 6 input NAND gate, i.e. the output signal has a frequency equal to (Freq#2)/256 and duty cycle of 4/256. For the temperature sensor's output signal (Freq#1), as the frequency is already in the range of 300 Hz, only its duty cycle is reduced. The duty cycle value is set using the delay of 5 inverters with capacitive load, which are optimized across temperature and process corners.

IV. SIMULATION RESULTS

The transient response of the DC/DC converters with no load (i.e. on-chip sensors are disabled) is shown in Figure 10. Ideal conversion ratio of 3 is observed across the input voltage range. As given by (1), for a given $f$clk and fly capacitance C, Iload reduces the per stage gain (ΔV) and hence the overall conversion ratio, as shown in Figure 11. The frequency variation w.r.t temperature across various input voltages for the temperature and the voltage sensor, is respectively shown in Figure 12 & 13. As estimated with (4) & (6), the temperature sensor is voltage independent, whereas the voltage sensor depends on voltage as well as temperature. The overall time response of the ASIC with sensors enabled at 10ms is shown in Figure 14. For Vin = 0.6 V the DC/DC converter provides an output of 1.5 V and consumes an average power of 200 nW.

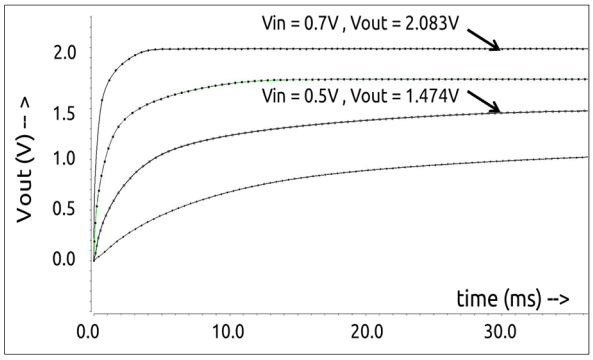


Figure 10: Transient response of DC/DC converter with sensors disabled and zero output load

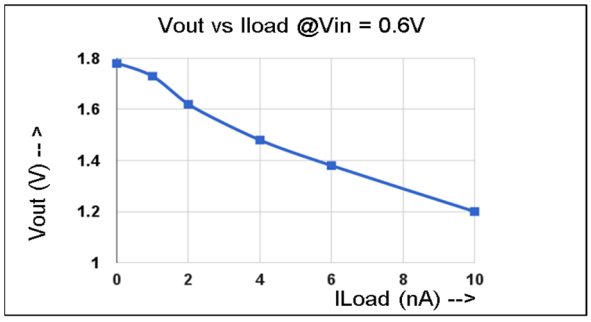


Figure 11: Current loading on DC/DC converter's output voltage

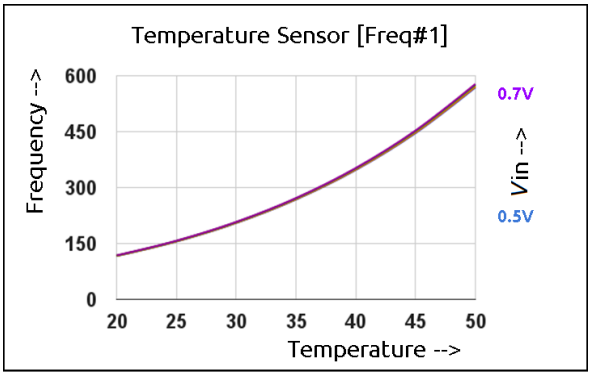


Figure 12: Temperature sensor's output characteristic

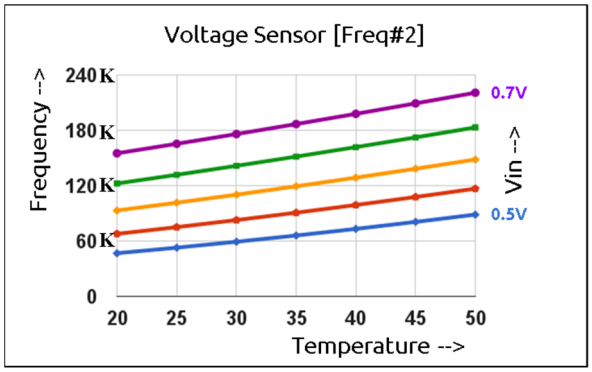


Figure 13: Voltage sensor's output frequency vs temperature

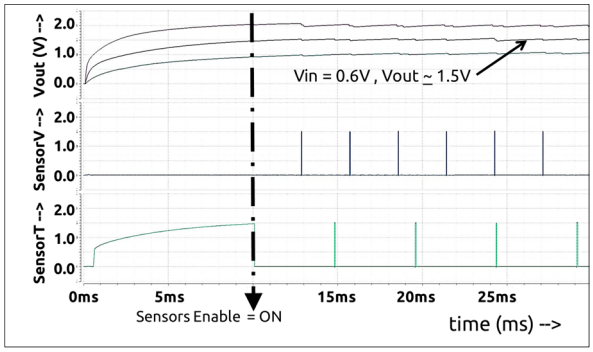


Figure 14: Transient response of the complete ASIC with on-chip sensors enabled at 10ms

V. FEASIBLE ENHANCEMENT

The three parameters that can enhance the performance of the implemented design are:

a) Usage of low Vth devices
b) Availability of separate P_well for NMOS devices
c) Higher capacitance density

To study their effects, the same DC/DC converter's design with ideal clocks and fly capacitors was implemented in IMS 500 nm twin well and XFAB 180 nm technologies. The IMS 500 nm twin well technology is a customized PDK with separate P_wells for the NMOS. The XFAB 180 nm technology is a twin well process with low Vth devices. The comparison of the technologies is provided in Table I.

Table I : Different technologies used for comparison

	IMS 500 nm	**IMS 500 nm**	**XFAB 180 nm**
Vth (mV)	700	700	350
Twin Well	No	Yes	Yes
Capacitance Density(fF/μm^2)	0.6	0.6	6.6

As explained in Section II, for low inputs, the conduction in cross connected NMOS happens by sub threshold current. As per (2), Isub depends on both Vth and the body bias (Vsb). The effect of Vth can be observed in Figure 15 & 16. For low input voltage, the Isub in IMS technologies becomes so low that for a given $f$clk and fly capacitance C, the $I_{\text{input}} << I_{\text{output}}$ and the output voltage drops. With increasing number of cascaded DC/DC converter stages the body bias effect becomes more and more relevant in non-P_well technology, such as IMS 500 nm technology. As a result the Isub current in later stages decreases exponentially, such that it becomes negligible after a few stages and the conversion ratio saturates, as shown in Figure 17.

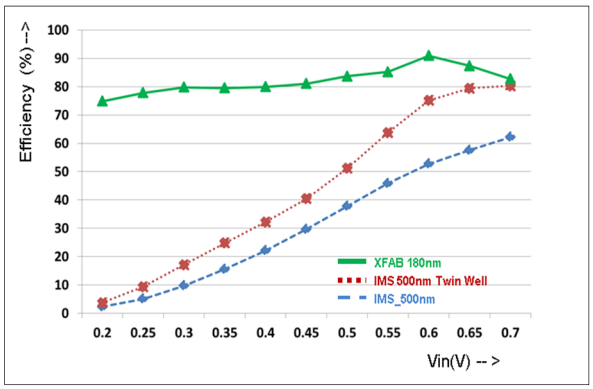


Figure 15: Efficiency of DC/DC up-converter with 180MΩ load

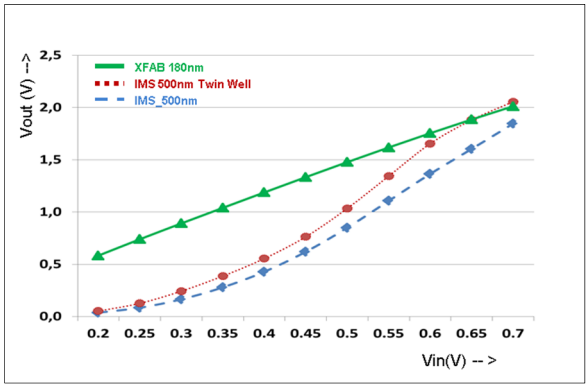


Figure 16: Output of DC/DC up-converter with 180MΩ load

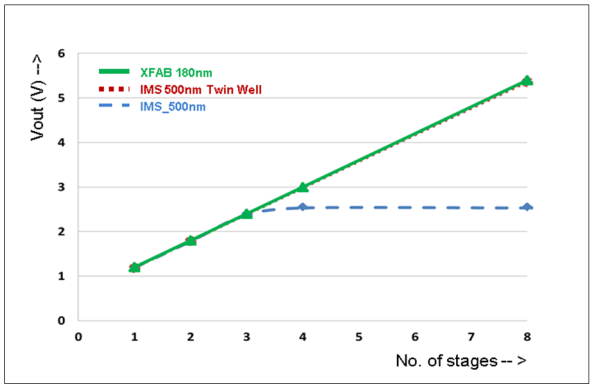


Figure 17: Output voltage of DC/DC up-converter for Vin = 0.6V with no load

VI. CONCLUSION

A fully integrated ASIC to harvest energy from GBFC was designed in IMS 500 nm technology with total on-chip capacitance of 52 pF. It harvests energy through on-chip GBFC's electrodes and up-converts the input voltage in the range of 0.4 V to 0.7 V by a factor of 3, using a charge pump based DC/DC converter and powers the on-chip temperature & voltage sensors. It consumes an average power of 200 nW, while providing an output voltage of 1.5 V for an input of 0.6 V. The implemented temperature & voltage sensors provide an output frequency sensitivity of 5% per ºC and 0.2% per mV, respectively. Finally a comparative study using a low power technology and modified IMS technology is presented for feasible design performance enhancements.

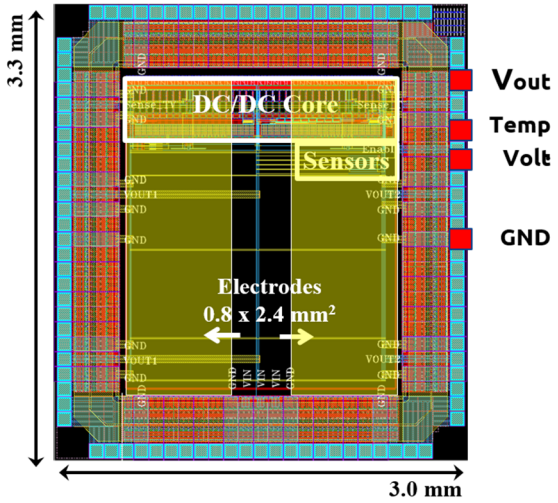


Figure 18: Layout of the designed ASIC

REFRERENCE

[1] Haeberlin, Andreas, Adrian Zurbuchen, Sébastien Walpen, Jakob Schaerer, Thomas Niederhauser, Christoph Huber, Hildegard Tanner, Helge Servatius, Jens Seiler, Heinrich Haeberlin, Juerg Fuhrer, and Rolf Vogel. "The First Batteryless, Solar-powered Cardiac Pacemaker." Heart Rhythm 12, no. 6 (06 2015): 1317-323.

[2] Zebda, A., S. Cosnier, J.-P. Alcaraz, M. Holzinger, A. Le Goff, C. Gondran, F. Boucher, F. Giroud, K. Gorgy, H. Lamraoui, and P. Cinquin. "Single Glucose Biofuel Cells Implanted in Rats Power Electronic Devices." Sci. Rep. Scientific Reports 3 (03, 2013).

[3] Dickson, J.f. "On-chip High-voltage Generation in MNOS Integrated Circuits Using an Improved Voltage Multiplier Technique." IEEE J. Solid-State Circuits IEEE Journal of Solid-State Circuits 11, no. 3 (06 1976): 374-78.

[4] Wu, Jieh-Tsorng, and Kuen-Long Chang. "MOS Charge Pumps for Low-voltage Operation." IEEE J. Solid-State Circuits IEEE Journal of Solid-State Circuits 33, no. 4 (04 1998): 592-97.

[5] Nakagome, Y., H. Tanaka, K. Takeuchi, E. Kume, Y. Watanabe, T. Kaga, Y. Kawamoto, F. Murai, R. Izawa, D. Hisamoto, T. Kisu, T. Nishida, E. Takeda, and K. Itoh. "An Experimental 1.5-V 64-Mb DRAM." IEEE J. Solid-State

Circuits IEEE Journal of Solid-State Circuits 26, no. 4 (04 1991): 465-72.

[6] Ituero, Pablo, José L. Ayala, and Marisa Lopez-Vallejo. "A Nanowatt Smart Temperature Sensor for Dynamic Thermal Management." IEEE Sensors J. IEEE Sensors Journal 8, no. 12 (12 2008): 2036-043.

Narayanan Seeetharaman received his B.Tech.degree in electronics and telecommunication engineering from India in 2010. Currently he is pursuing his Master's degree at University of Stuttgart, Stuttgart, Germany. He worked with Sanaklp Semiconductors as design engineer from 2010 until 2013.

Harald Richter received the Ph.D. degree in physics from the University of Stuttgart, Stuttgart, Germany, in 1983. He worked with Xerox Research Center Webster, NY, USA, from 1983 to 1984 and with Max-Planck Institut for Semiconductor Research, Stuttgart, Germany, from 1984 to 1986. He joined IMS, Stuttgart, Germany, in 1986, and held positions with Technology and Systems Division before being nominated the Head of System Division in 2005.

Joachim N. Burghartz (M'87–SM'92–F'01) was born in Aachen, Germany, in 1956. He received the Dipl.-Ing. degree from RWTH Aachen, Germany, in 1982, and the Ph.D. degree from the University of Stuttgart, Stuttgart, Germany, in 1987, both in electrical engineering. He is the Director of the Institute for Microelectronics Stuttgart (IMS CHIPS), Stuttgart, Germany, and a Full Professor with the University of Stuttgart, Stuttgart, Germany.

Hochschule
Ravensburg-Weingarten
Technik | Wirtschaft | Sozialwesen

Implementierung eines synchronen digitalen Taktteilers

Roman Martynenko, Christopher Bonenberger, Andreas Siggelkow

***Zusammenfassung*— Thema dieser Arbeit ist die Implementierung eines digitalen Taktgenerators. Dieser Taktgenerator wurde mittles digitalem VHDL-Entwurf realisiert, um einen Eingangstakt einstellbar auf niedrigere Frequenzen herunter zu teilen. Der Taktteiler hat vier Ausgänge, ist auf beliebige geradzahlige Teilerverhältnisse einstellbar, es kann mit automatischen Methoden ein Scanpfad erzeugt werden, die *High*-Phase wird nicht angeschnitten und das Teilerverhältnis wird aus dem gespeicherten Wert des Eingangstaktes und vier speicherbaren Zielfrequenzen errechnet.**

***Schlüsselwörter*—Taktgenerator, synchron, testbar, skalierbar, glitch-free, frei programmierbar**

I. Einleitung

Dieser Frequenzteiler passt das Zeitverhalten zwischen dem 8051-Core von OREGANO und der Speicherperipherie an. Die Speicherperipherie soll wie bei einem klassischen 8051 über die Ports P0 und P2 betrieben werden.

Dieser Taktgenerator ist *scan*-testbar, *glitch*-frei und frei programmierbar entworfen. Die erzeugten Taktnetze sollen während des Betriebs einzeln aktiviert und deaktiviert werden können. Die Vorgaben von [1] bezüglich abschaltbarer Takte sind eingehalten.

Alle erzeugten Takte müssen den Taktausgängen frei zuzuordnen sein, wobei dies auch während des Betriebs möglich sein soll. Weiterhin sind die generierten Takte vom Design her auf minimale Verzögerung der steigenden Flanken gegeneinander ausgelegt. Somit werden *race*-Bedingungen minimiert.

Die besonderen Anforderungen an den Taktgenerator ergeben sich also in erster Linie aus der Problematik der Metastabilität und der Notwendigkeit der Testbarkeit des Systems.

Das Design folgt den Vorgaben: asynchroner Reset, synchroner Takt.

A. Übersicht

Zunächst werden in Abschnitt II die Anforderungen an das System spezifiziert. Anschließend wird in Abschnitt III auf die Konzeptionierung und Implementierung des Teilers eingegangen und die genaue Struktur näher beschrieben.

R. Martynenko, rm-152424@hs-weingarten.de, und C. Bonenberger, cb-152428@hs-weingarten.de, sind Studenten an der Hochschule Ravensburg-Weingarten. A. Siggelkow ist Mitglied der Hochschule Ravensburg-Weingarten, Doggenriedstraße, 88250 Weingarten.

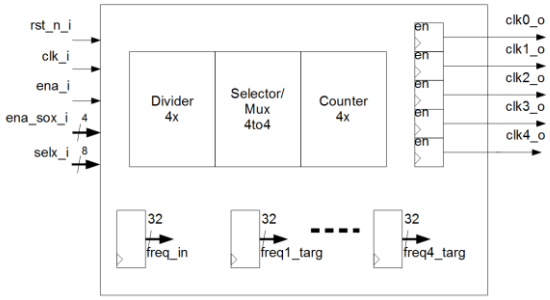


Abbildung 1 Top-Level Diagramm

II. Anforderungen

Der Taktteiler soll über Register vier Werte für die Taktfrequenzen der vier freien Ausgänge einlesen und aus diesen die Grenzwerte für die Teilerblöcke berechnen. Die Frequenzteiler werden als Zähler implementiert, sollen synchron zueinander arbeiten und abschaltbar sein. Die Ausgangstakte sollen ein Tastverhältnis von 50 Prozent haben. Die Deaktivierung einzelner oder aller Zähler darf nicht zum Anschneiden der *high*-Phasen führen. Die Deaktivierung aller Takte (globales Freigabe-Signal) soll synchron zum langsamsten Takt erfolgen. Wird ein einzelner Takt während des Betriebs deaktiviert und nach einer unbestimmten Zeit wieder aktiviert, so muss dieser Takt synchron zu sich selbst fortgeführt werden. Dies garantiert die kontinuierliche Synchronizität beziehungsweise Phasengleichheit der Taktnetze.

Im Folgenden werden wichtigsten Punkte näher beschrieben.

A. Clock-Gating

Zur Steuerung der einzelnen Taktnetze werden die verschiedenen Takte mittels eines asynchronen Freigabe-Signals geschaltet. Eine kombinatorische Logik zur Umsetzung kann zu Störimpulsen (*Glitches*) führen [1]. Die Taktausgänge müssen also über Flip-Flops oder *Latches* geschaltet werden, um zu gewährleisten, dass keine *high*-Phasen angeschnitten werden.

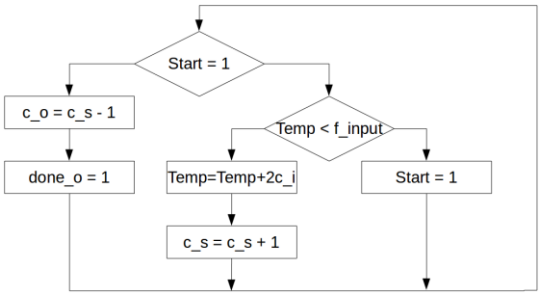


Abbildung 2 Flussdiagramm des Frequenzteilers

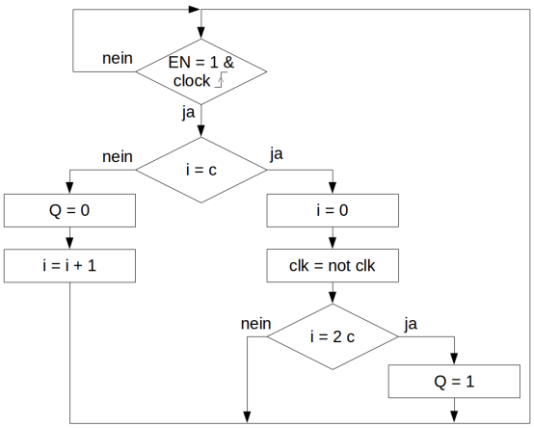


Abbildung 3 Zählereinheit

B. Metastabilität

Werden die *setup*- und *hold*-Zeiten an einem Flip-Flop nicht eingehalten, so kann sich am Ausgang ein metastabiler (meta, *griech*: zwischen) Zustand einstellen. Kennzeichnend für den Zustand der Metastabilität ist die fehlende Eindeutigkeit des Pegels am Ausgang eines Flip-Flops. Dies zeigt sich für eine unbestimmte Zeitdauer als ein Pendeln der Spannung am Ausgang zwischen den entsprechenden Spannungspegeln für eine ‚1' und eine ‚0'. Solche Effekte erweisen sich jedoch nur bei sehr hohen Taktfrequenzen als problematisch, denn *setup*- und *hold*-Zeiten öffnen nur ein kurzes Zeitfenster und zudem dauert ein metastabiler Zustand meist nur wenige ns an.

Dennoch kann eine ungünstige Auslegung des Taktnetzes, auch bei langsamen Taktfrequenzen, zu einer Verletzung der *setup*- und *hold*-Zeiten führen und so metastabile Zuständen verursachen. Neben Laufzeit-bedingten Störungen, beispielsweise durch unterschiedliche Leitungslängen, kann auch der funktionale Entwurf des Taktnetzes Ursache von asynchronem Verhalten mit der Folge von metastabilen Ausgängen sein. Wird zum Beispiel ein Signal ohne Synchronisation in zwei Taktnetzen verwendet, kann dies bei asynchronen Taktnetzen zu Metastabilität führen.

Aus diesem Grund muss der Taktgenerator eine Struktur aufweisen, bei welcher alle erzeugten Takte eine ähnliche Logik durchlaufen womit für jeden Takt dieselbe Phasenverschiebung erzeugt wird. Auf diese Weise ergibt sich funktional kein Taktversatz und die erzeugten Takte am Ausgang bleiben synchron beziehungsweise phasengleich.

Bei heterochronen Takten kann die Problematik allerdings nicht durch Berücksichtigung im Taktgenerator umgangen werden, sondern nur durch Einsynchronisieren der entsprechenden Signale.

C. Testbarkeit

Um die Testbarkeit des Systems zu gewährleisten muss der System-Entwurf die *DFT* (*Design for Testability*) Anforderungen erfüllen. Das bedeutet, dass die gesamte Logik des Taktgenerators von demselben Takt gesteuert werden muss. Nur so können die entsprechenden Prüfpfade eingefügt werden, welche für die Anwendung der Prüfmuster erforderlich sind und die Testbarkeit gewährleisten.

Weiterhin ist ein globaler asynchroner Reset erforderlich um das System in einen definierten Zustand zu versetzen.

III. KONZEPT

Die grundlegende Struktur des Taktteilers ist in Abbildung 1 dargestellt. Aus den entsprechenden Registern werden die Werte der gewünschten Frequenzen ausgelesen und im Frequenzteiler verrechnet. Die aus diesen Berechnungen resultierenden Grenzwerte werden von den Zählern eingelesen. Die Taktausgänge der Zähler werden über einen Multiplexer den verschiedenen Taktgenerator-Ausgängen zugewiesen. Die Steuerlogik stellt die Synchronizität sicher und verhindert das Auftreten von Störimpulsen bei Verwendung der globalen Freigabe.

Der in Abbildung 1, 5 und 6 zusätzlich dargestellte Taktausgang clk0_o dient lediglich Testzwecken.

Die Funktionsweise der einzelnen Komponenten wird im Folgenden näher beschrieben.

A. Frequenzteiler

Mit der gegebenen Eingangsfrequenz (freq_in) und den gewünschten Ausgangsfrequenzen (freq1_targ bis freq4_targ) werden im Frequenzteiler die benötigten Grenzwerte c_x für die Zähler berechnet, wobei gilt:

$$c_x = \left\lfloor \frac{f_{clk\_i}}{2f_{out,x}} - 1 \right\rfloor.$$

Hierbei bezeichnen f_{clk_i} die gegebene Eingangsfrequenz, $f_{out,x}$ die erforderliche Ausgangsfrequenz des x-ten Zählers und der Ausdruck $\lfloor z \rfloor$, mit $z \in \mathbb{R}$, die Zahl $i \in \mathbb{Z}$ mit $i \leq z < i + 1$. Bei ausreichend hoher Eingangsfrequenz bleibt der Rundungsfehler gegebenenfalls also vernachlässigbar.

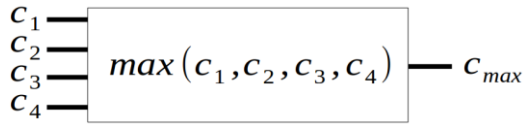


Abbildung 4 Selektor

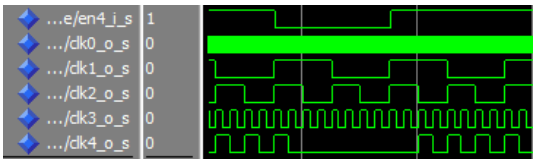


Abbildung 5 Zeitlicher Verlauf bei individueller Freigabe

Erst nach Abschluss der Berechnung aller Grenzwerte c_x werden die Zähler aktiviert.

Ein detaillierter Überblick ist in Abbildung 2 gegeben.

Zur Berechnung der Grenzwerte wird der Wert der gewünschten Eingangsfrequenz in Hardware (freqx_targ) aufaddiert bis der Wert der gegebenen Eingangsfrequenz (freq_in) erreicht ist.

B. Zähler

Die Funktion der Zähler ist in Abbildung 3 schematisch dargestellt.

Die Periodendauer T_x der ausgegebenen Frequenz berechnet sich wie folgt:

$$T_x = 2 \sum_{i=0}^{c_x} T_{clk\_i},$$

wobei T_{clk_i} die Periodendauer der Eingangsfrequenz clk_i bezeichnet.

Jeder Zähler verfügt über zwei Freigabe-Signale. Ein globales Freigabe-Signal ist mit dem Freigabe-Signal der Frequenzteiler gekoppelt. Einzelne Freigabe Signale für jeden Zähler ermöglichen die individuelle Aktivierung und Deaktivierung der verschiedenen Zähler. Das globale Freigabe-Signal (ena_i) ist an den langsamsten Takt gekoppelt, sodass alle Takte bis zum Ende der *low*-Phase des langsamsten Taktes aktiviert bleiben und bei Freigabe alle Takte gleichzeitig starten.

Die individuellen Freigabe-Signale (ena_so) werden erst zum Ende der *low*-Phase des entsprechenden Zählers aktiv, wodurch das ursprünglich womöglich asynchrone Signal auf den entsprechenden Takt synchronisiert wird.

Eventuell auftretende Verschiebungen der erzeugten Takte zueinander, welche sich durch unterschiedliche Zähler-Grenzwerte ergeben können, werden durch Puffer am Ausgang kompensiert.

Die entsprechenden zeitlichen Verläufe sind exemplarisch in Abbildung 5 und Abbildung 6 dargestellt.

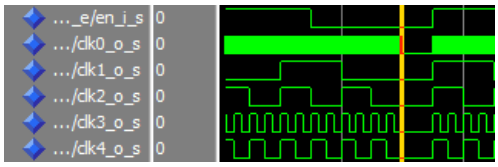


Abbildung 6 Zeitlicher Verlauf bei globaler Freigabe

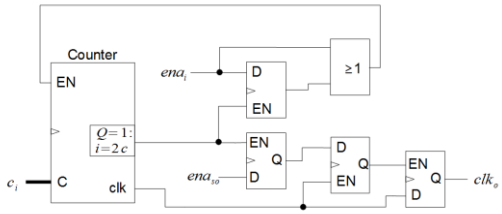


Abbildung 7 Clock Gating

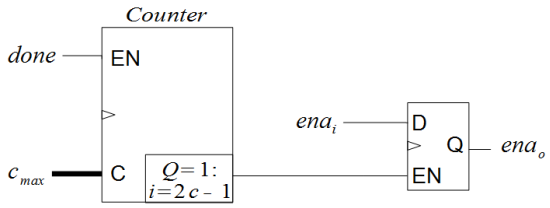


Abbildung 8 Synchronisierung der globalen Freigabe

C. Selektor

Um den langsamsten der gegebenen Takte zu identifizieren wird ein Selektor benötigt, welcher die entsprechenden Vergleichsoperationen durchführt. An die Steuerung des Taktgenerators wird dann der vom Frequenzteiler berechnete Grenzwert c_{max} des langsamsten Taktes ausgegeben (siehe Abbildung 4).

D. Steuerung

Ein in der Steuerungseinheit (Abbildung 8) implementierter Zähler mit dem Grenzwert c_{max} gibt das globale Freigabe-Signal nach Ablauf der *low*-Phase an die Taktgenerator-Ausgänge weiter. Dieser Zähler ist synchron zum erzeugten Takt der entsprechenden (langsamsten) Frequenz (siehe Abbildung 9).

Weiterhin ist in der Steuereinheit ein Multiplexer zur Auswahl der Frequenzen an den Ausgängen implementiert. Über die Steuereingänge des Taktgenerators (selx_i) werden die Takte den Ausgängen (clk1_o bis clk4_o) zugeordnet. Erfolgt während des Betriebs ein Umschalten des Taktes so wird dieser Umschaltvorgang durchgeführt ohne das die *high*-Phase des entsprechenden Ausgangs angeschnitten wird.

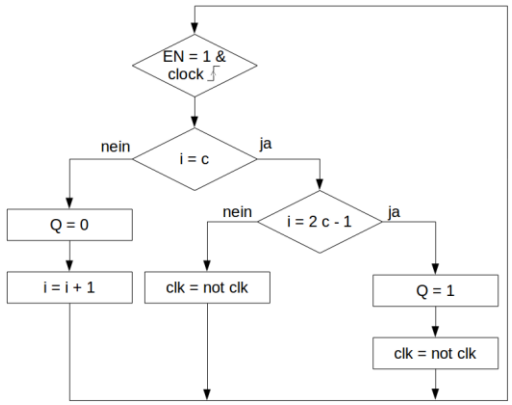


Abbildung 9 Zählereinheit der Steuerung

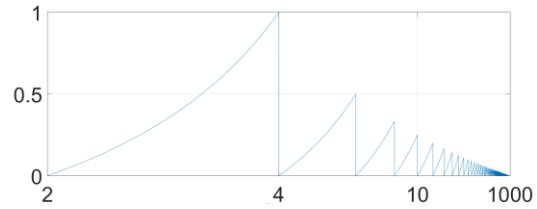


Abbildung 10 Relativer Fehler in Abhängigkeit des Teilerfaktors

E. Taktausgänge

Obwohl das globale Freigabe-Signal ena_i und auch die Freigabe-Signale der einzelnen Ausgänge (ena_so) auf die entsprechenden Takte synchronisiert werden, kann eine kombinatorische Logik am Ausgang aufgrund von Laufzeitunterschieden zu *Glitches* führen.

Um dieser Problematik des *Clock-Gatings* vorzubeugen, werden die Taktausgänge über Flip-Flops geschaltet. Die Realisierung des *Clock-Gatings* erfolgt mittels Flip-Flops, die synchron zum Systemtakt clk_i (siehe Abbildung 7) arbeiten. Der in Abbildung 7 dargestellte Zähler ist Abbildung 3 entsprechend implementiert.

IV. Ergebnis

Es konnte ein Taktteiler implementiert werden, welcher die Anforderungen erfüllt. Eine Einschätzung des relativen Fehlers der erzeugten Ausgangsfrequenzen in Abhängigkeit des Teilerfaktors kann anhand von Abbildung 10 vorgenommen werden.

Literaturverzeichnis

[1] M. Arora, „The Art of Hardware Architecture,“ *Springer*, New York, 2012.

[2] F. Kesel, R. Bartholomä, „Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs,“ *Oldenbourg Verlag*, München 2009.

Roman Martynenko erhielt den akademischen Grad des B.-Eng. in Elektro- und Informationstechnik im Jahr 2015 von der Hochschule Ravensburg-Weingarten. Seit 2015 studiert er an der Hochschule Ravensburg-Weingarten im Master Electrical Engineering.f

Christopher Bonenberger erhielt den akademischen Grad des B.-Eng. in Elektro- und Informationstechnik im Jahr 2014 von der Hochschule Ravensburg-Weingarten. Seit 2015 studiert er an der Hochschule Ravensburg-Weingarten im Master Electrical Engineering.

Andreas Siggelkow erhielt den akademischen Grad Dipl.-Ing. im Jahr 1988 von der Universität Karlsruhe. Im Jahr 1996 promovierte er an der Universität Stuttgart zum Dr.-Ing. Seit dem Jahr 2007 ist er Professor für ASIC-Design und Rechnerarchitektur an der Hochschule Ravensburg-Weingarten.

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. Ihle, (0721) 925-1571
marc.ihle@hs-karlsruhe.de

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

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

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

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

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

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

Hochschule Ulm
Prof. Dr. Terzis, (0731) 50-28341
terzis@hs-ulm.de

www.mpc.belwue.de