

MPC

MULTI PROJECT CHIP GRUPPE
BADEN - WÜRTTEMBERG

Herausgeber: MPC-Gruppe

Ausgabe: 67/68

ISSN: 1868-9221

Publiziert: Juni 2025

Workshop: Karlsruhe Januar 2024

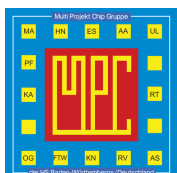
Workshop: Düsseldorf Januar 2025

Band 67 Workshop: Karlsruhe Januar 2024

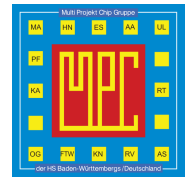
- 1 Analoges MPP-Tracking mit Spitzenwertdetektion**
Patrick Mader, Sascha Eckert, Rainer Merz, Hochschule Karlsruhe
- 9 Materialcharakterisierung basierend auf einem 120 GHz FMCW-Radar**
Marius Patzer, Nicolas Treier, Marlene Harter, Serdal Ayhan, Herman Jalli Ng, Hochschule Karlsruhe
- 15 Realisierung von elektronischen und sensorischen Komponenten mittels FDM-3D-Druck-Verfahren**
Jannik Söll, Patrick Hog, Elke Mackensen, Hochschule für Technik, Wirtschaft und Medien Offenburg
- 23 Low Cost Serial DAC Simulation, Realization, Error Correction and Characterization**
Jörg Vollrath, Hochschule Kempten
- 31 Improving Hardware Security through Encryption of JTAG TAP Ports: An Analytical Comparison of Algorithms, with a Focus on RSA**
Soham Sanjay Dekhane, Andreas Siggelkow, Hochschule Ravensburg-Weingarten

Band 68 Workshop: Düsseldorf Januar 2025

- 37 Coupling between traces and power planes in PCBs. How is it influenced and what are its consequences?**
Justus Gottwald, Michael Engelbrecht, Bernhard M. Rieß, Hochschule Düsseldorf
- 47 Evaluation of a new AI driven SoC Design Flow Approach**
Thierry Kersten, Thomas Haase, Bernhard M. Rieß, Hochschule Düsseldorf
- 59 Eine Debug- und Testumgebung für AXI-Stream basierte Ethernet-Frame-Verarbeitung in FPGAs**
Sebastian Haberkorn, Marcus Wür, Stefan Eiermann, Hochschule Esslingen
- 65 Entwicklung und Implementierung eines multifunktionalen PWM Modulators**
Maik Greschuck, Clemens Elskamp, Bernhard M. Rieß, Hochschule Düsseldorf



Cooperating Organisation
Solid-State Circuit Society Chapter
IEEE German Section



Inhaltsverzeichnis

Band 67 Workshop: Karlsruhe Januar 2024

Analoges MPP-Tracking mit Spitzenwertdetektion	1
Patrick Mader, Sascha Eckter, Rainer Merz, Hochschule Karlsruhe	
Materialcharakterisierung basierend auf einem 120 GHz FMCW-Radar	9
Marius Patzer, Nicolas Treier, Marlene Harter, Serdal Ayhan, Herman Jalli Ng, Hochschule Karlsruhe	
Realisierung von elektronischen und sensorischen Komponenten mittels FDM-3D-Druck-Verfahren	15
Jannik Söll, Patrick Hog, Elke Mackensen, Hochschule für Technik, Wirtschaft und Medien Offenburg	
Low Cost Serial DAC Simulation, Realization, Error Correction and Characterization	23
Jörg Vollrath, Hochschule Kempten	
Improving Hardware Security through Encryption of JTAG TAP Ports: An Analytical Comparison of Algorithms, with a Focus on RSA	31
Soham Sanjay Dekhane, Andreas Siggelkow, Hochschule Ravensburg-Weingarten	

Band 68 Workshop: Düsseldorf Januar 2025

Coupling between traces and power planes in PCBs. How is it influenced and what are its consequences?	37
Justus Gottwald, Michael Engelbrecht, Bernhard M. Rieß, Hochschule Düsseldorf	
Evaluation of a new AI driven SoC Design Flow Approach	47
Thierry Kersten, Thomas Haase, Bernhard M. Rieß, Hochschule Düsseldorf	
Eine Debug- und Testumgebung für AXI-Stream basierte Ethernet-Frame-Verarbeitung in FPGAs	59
Sebastian Haberkern, Marcus Wür, Stefan Eiermann, Hochschule Esslingen	
Entwicklung und Implementierung eines multifunktionalen PWM Modulators	65
Maik Greschuck, Clemens Elskamp, Bernhard M. Rieß, Hochschule Düsseldorf	

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

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

Herausgeber:

Lothar Schmidt, MPC-Gruppe, Albert-Einstein-Allee 53, D-89081 Ulm

Mitherausgeber (Peer Reviewer):

Joachim Gerlach, Hochschule Albstadt-Sigmaringen, Gartenstraße 15, D-72458 Albstadt

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

Eckhard Hennig, Hochschule Reutlingen, Alteburgstraße 150, 72762 Reutlingen

Bernhard Jungk, Hochschule Albstadt-Sigmaringen, Gartenstraße 15, D-72458 Albstadt

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

Andreas Siggelkow, Hochschule Ravensburg-Weingarten, Doggenriedstraße, 88250 Weingarten

Alle Rechte vorbehalten

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

Analogenes MPP-Tracking mit Spitzenwertdetektion

Patrick Mader, Sascha Eckerter, Rainer Merz

Zusammenfassung—Moduloptimierer erhöhen die Leistungsentnahme von Photovoltaik-(PV-)Anlagen bei Verschattungen, indem sie jedes Modul im Punkt maximaler Leistung (MPP) betreiben, unabhängig von der jeweiligen Einstrahlung. Dafür muss jedes PV-Modul mit einem Moduloptimierer ausgerüstet sein, was die Anschaffungskosten der Anlage erhöht. Deshalb ist eine besonders kostengünstige Realisierung von Moduloptimierern wichtig. Dies lässt sich durch Erhöhen der Taktfrequenz der Leistungselektronik und damit einem Verkleinern der passiven Bauelemente erreichen. Da das MPP-Tracking aber in der Regel auf einem Mikrocontroller erfolgt, führt das zu steigenden Kosten für einen schnellen Mikrocontroller. Deshalb stellt dieses Paper einen analogen MPP-Tracker vor, der ein kostengünstiges Erhöhen der Taktfrequenz ermöglicht und gleichzeitig durch Integration von eigener Spannungsversorgung und Treiberschaltung weitere Kosten reduziert. Der MPP-Tracker basiert auf einer Spitzenwertdetektion und wurde in einer analogen anwendungsspezifischen integrierten Schaltung (ASIC) integriert. Messung des ASICs an einer Leistungselektronik zeigen die Funktionsfähigkeit im erforderlichen Arbeitsbereich.

Schlüsselwörter—MPP-Tracking, Analogenes MPP-Tracking, Moduloptimierer, ASIC, Photovoltaik, Peak-Detector, Watchdog-Timer, Perturb-And-Observe

I. EINLEITUNG

Eine PV-Anlage besteht typischerweise aus mehreren in Serie geschalteten PV-Modulen, die einen Strang bilden. Dadurch ist der Strom I durch jedes Modul gleich und die Modulspannungen addieren sich zur Gesamtspannung. Antiparallel zu jedem Modul befinden sich Bypass-Dioden, diese schützen die Module vor thermischer Überlastung. Denn wie in Abbildung 1 zu sehen, ist der Strom I_1 eines PV-Moduls abhängig von der Einstrahlung E und die Spannung U_1 abhängig von der Temperatur ϑ . Dadurch ist auch der MPP abhängig von Einstrahlung E und Temperatur ϑ . Kommt es z.B. aufgrund einer Verschattung an einem Modul zu einer kleineren Einstrahlung E , dann sinkt der lieferbare Strom des Moduls ab. Liegt nun der Strangstrom, der sich durch die anderen Module ergibt über dem Kurzschlussstrom des verschatteten Moduls, dann verschiebt sich der Arbeitspunkt in den Verbraucherbetrieb des PV-Moduls. Dadurch nimmt das Modul Leistung auf und die Spannung über dem Modul U_1

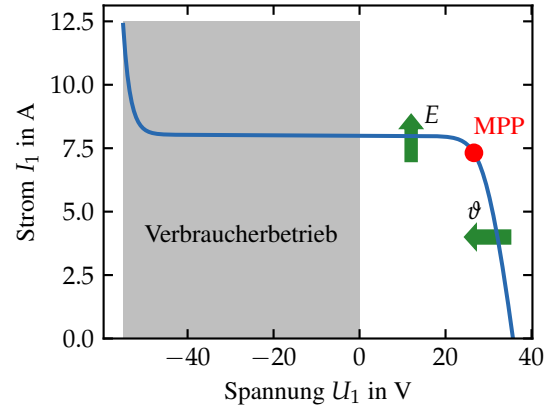


Abbildung 1. Strom-Spannungskennlinie eines PV-Moduls. Im MPP gibt das Modul die maximal mögliche Leistung ab. Eine steigende Temperatur ϑ verschiebt die Kennlinie zu kleineren Spannungen und eine steigende Einstrahlung verschiebt die Kennlinie zu höheren Strömen.

wird negativ. Durch die negative Spannung leitet die Bypassdiode und begrenzt die Verlustleistung des Moduls. Die Bypassdiode verhindert so eine thermische Zerstörung des Moduls. Allerdings liefert das verschattete Modul dann keine Leistung mehr, obwohl auch bei reduzierter Einstrahlung noch Leistung entnehmbar wäre. Eine Abhilfe schaffen hierbei sogenannte Moduloptimierer. Ein typischer Moduloptimierer, wie z.B. in [1] oder [8] vorgestellt, ist in Abbildung 2 zusehen und besteht aus einem Gleichspannungswandler und einem MPP-Tracker. Am Eingang des Moduloptimierers befindet sich das PV-Modul mit der Spannung U_1 und am Ausgang wird der restliche PV-Strang durch einen ohmschen Widerstand R_L mit der Spannung U_2 modelliert. Der vorgestellte Moduloptimierer verwendet als Gleichspannungswandler einen Tiefsetzsteller, der aus einer Halbbrücke mit zwei Transistoren T1, T2 und einem Ausgangsfilter L , C_2 besteht. Die Transistoren werden über komplementäre pulsweitenmodulierte Signale (PWM-Signale) angesteuert. Dabei ist eine wichtige Kenngröße für die PWM-Signale der Tastgrad

$$d := \frac{T_{T1, \text{ein}}}{T} = \frac{T_{T2, \text{aus}}}{T}, \quad (1)$$

der das Verhältnis von Einschaltdauer $T_{T1, \text{ein}}$ von T1 bzw. Ausschaltdauer $T_{T2, \text{aus}}$ von T2 bezogen auf die Periodenlänge T der PWM-Signale angibt. Die mit den PWM-Signalen angesteuerte Halbbrücke liefert

Patrick Mader, patrick.mader@h-ka.de, Sascha Eckerter, sascha.eckerter@h-ka.de, Rainer Merz, rainer.merz@h-ka.de, Hochschule Karlsruhe – University of Applied Sciences (HKA), Moltkestraße 30, 76133 Karlsruhe.

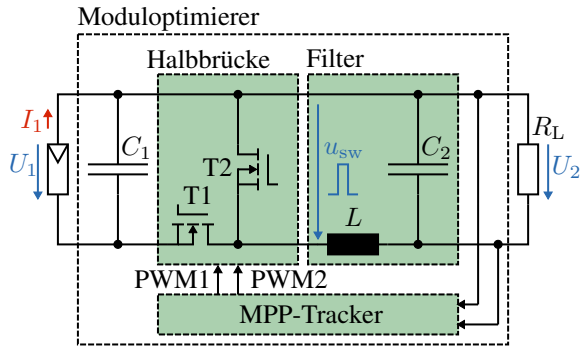


Abbildung 2. Typischer Aufbau eines Moduloptimierers. Halbrücke T1/T2 und Filter L/C_2 bilden einen Tiefsetzsteller. Durch Ansteuern des Tiefsetzstellers durch den MPP-Tracker arbeitet das PV-Modul mit der Spannung U_1 und dem Strom I_1 im MPP. Als Eingangsgröße verwendet der MPP-Tracker die Ausgangsspannung U_2 . Der Lastwiderstand R_L modelliert den restlichen Modulstrang.

dadurch am Ausgang eine Rechteckspannung u_{sw} mit dem Gleichanteil [9]

$$U_{sw} = \frac{1}{T} \int_0^T u_{sw} dt = d \cdot U_1 \quad (2)$$

und harmonische Oberschwingungen. Die Oberschwingungen werden durch den nachfolgenden Filter gedämpft und es ergibt sich der Zusammenhang

$$\frac{U_2}{U_1} = d, \quad (3)$$

des Verhältnisses von Ausgangsspannung U_2 zu Eingangsspannung U_1 und Tastgrad d . Diesen Zusammenhang nutzt der MPP-Tracker, um das angeschlossene PV-Modul im MPP zu betreiben, indem er durch Variieren des Tastgrades und Messen der Ausgangsleistung diesen MPP findet. Dabei verwendet der hier vorgestellte MPP-Tracker lediglich eine Spannungsmessung, um den MPP zu finden, da am ohmschen Widerstand der Zusammenhang

$$U_2 = \sqrt{P \cdot R_L} \quad (4)$$

gilt und demnach die Spannung proportional zur Leistung ist.

Da jedes Modul in einem PV-Strang mit einem Moduloptimierer ausgestattet wird, muss auf einen möglichst robusten und kostengünstigen Entwurf der Elektronik geachtet werden. Die meisten Kosten der Elektronik verursachen die passiven Bauelemente L , C_1 und C_2 aus Abbildung 2. Wie die Schaltung angesteuert werden muss, um möglichst niedrige Induktivitäts- und Kapazitätswerte verwenden zu können, wird klar, wenn die entsprechenden Dimensionierungsformeln herangezogen werden. Für die Eingangskapazität ergibt sich nach [5] der Zusammenhang

$$C_1 \propto \frac{1}{f}, \quad (5)$$

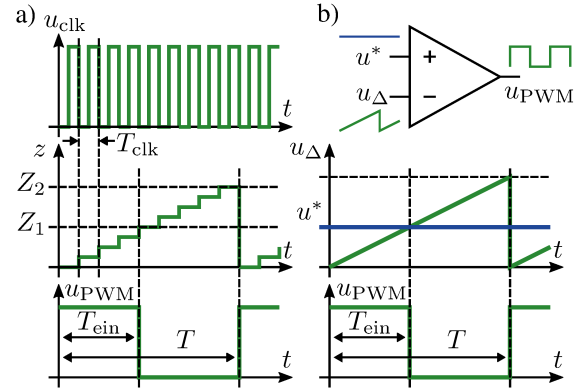


Abbildung 3. a) PWM-Erzeugung mit einem Mikrocontroller. Jeder Takt u_{clk} erhöht den Zähler z . Erreicht z das erste Vergleichsregister Z_1 schaltet u_{PWM} auf LOW und nach Erreichen von Z_2 ist die PWM-Periode beendet. b) Analoge PWM-Erzeugung mit Komparator durch Vergleich des Sollwerts u^* mit einem Trägersignal u_Δ . u^* bestimmt d und die Frequenz von u_Δ die PWM-Frequenz.

für die Ausgangskapazität liefert [2] den Zusammenhang

$$C_2 \propto \frac{1}{f} \quad (6)$$

und für die Induktivität resultiert nach [2] der gleiche Zusammenhang

$$L \propto \frac{1}{f}. \quad (7)$$

Damit lässt sich feststellen, dass das Erhöhen der PWM-Frequenz kleinere Werte für die passiven Bauteile des Tiefsetzstellers erlaubt. Jedoch führt das Erhöhen der PWM-Frequenz $f = 1/T$ bei einem Mikrocontroller gleichzeitig zu einer Verschlechterung der PWM-Auflösung d_{min} bei gleichbleibender Frequenz f_{clk} des Taktes u_{clk} , wie auch an Abbildung 3.a) zu erkennen ist. Denn um mit einem heutzutage standardmäßig verwendeten Mikrocontroller eine PWM zu erzeugen, wird ein Zähler $z \in \mathbb{N}$ verwendet, der nach jeder Periode T_{clk} des Taktes u_{clk} hochgezählt wird. Erreicht z das erste Vergleichsregister $Z_1 \in \mathbb{N}$, dann wird die PWM-Spannung u_{PWM} auf LOW geschaltet, somit bestimmt Z_1 die Einschaltzeit T_{ein} der PWM. Nach dem Erreichen des zweiten Vergleichsregisters $Z_2 = \{Z_2 \in \mathbb{N} | Z_2 \geq Z_1\}$ ist eine PWM-Periode beendet und z beginnt von vorne. Das Vergleichsregister Z_2 bestimmt somit die Periodendauer

$$T = (Z_2 + 1) \cdot T_{clk} \quad (8)$$

und neben Z_1 auch den Tastgrad $d = Z_1 / (Z_2 + 1)$. Weiter lässt sich die Auflösung, also die kleinste mögliche Änderung des Tastgrades mit

$$\Delta d_{min} = \frac{\partial d}{\partial Z_1} = \frac{1}{Z_2 + 1} \quad (9)$$

berechnen. Für ein effizientes MPP-Tracking mit hoher Taktfrequenz stehen die Gleichungen 8 und 9

allerdings im Widerspruch, da ein Vergrößern von Z_2 zwar die Auflösung verbessert, aber gleichzeitig die Taktfrequenz verkleinert und umgekehrt. Die naheliegende Möglichkeit die Taktfrequenz zu erhöhen, ist das Verwenden eines Mikrocontrollers mit hoher Clock-Frequenz, was allerdings zu einem teuren Mikrocontroller führt. Eine andere Möglichkeit ist die Verwendung eines analogen PWM-Generators, wie in Abbildung 3.b) zusehen. Analoge PWM-Generatoren arbeiten mit einem Komparator, an dem der Sollwert des Tastgrads u^* mit einem dreieckförmigen Trägersignal u_Δ verglichen wird. Damit ist die Auflösung

$$\Delta d_{\min} = \lim_{Z_2 \rightarrow \infty} \frac{1}{Z_2} = 0, \quad (10)$$

bei unendlich vielen Stufen Z_2 und damit theoretisch unendlich groß. Die PWM-Frequenz f kann dadurch kostengünstig mit der Frequenz des Dreiecksgenerators erhöht werden. Deshalb beschäftigt sich diese Arbeit mit der Entwicklung eines MPP-Tracking-Verfahrens für einen analogen ASIC. Die Verwendung eines ASICs führt zu zusätzlichen Kosteneinsparungen, da überflüssige Funktionen eines Mikrocontrollers entfallen und benötigte Funktionen, wie Spannungsversorgung und Gate-Treiber auch integriert werden können. Dadurch sinkt auch die Anzahl an externen Bauteilen und somit wird der MPP-Tracker durch die geringere Anzahl an Lötstellen robuster.

Zur Vorstellung des entwickelten ASICs beginnt diese Arbeit zunächst in Abschnitt II mit der Erläuterung des MPP-Tracking-Verfahrens anhand eines Blockschaltbilds. Anschließend werden die wichtigsten Teilschaltungen anhand von Schaltplänen gezeigt. Der Abschnitt III stellt den gefertigten ASIC und die Testschaltung vor und zeigt anschließend die Messergebnisse. Abschließend fasst Abschnitt IV die Arbeit zusammen.

II. ANALOGES MPP-TRACKING

Für die Auswahl des Verfahrens zum MPP-Tracking kommen verschiedene Algorithmen infrage, die in der Literatur [6] auch schon in digitale und analoge Implementation klassifiziert sind. Aufgrund der niedrigen Komplexität und der relativ hohen Effizienz wird der Perturb-And-Observe Algorithmus als Grundlage verwendet. Beim Perturb-And-Observe Verfahren wird der Tastgrad solange in die gleiche Richtung Schrittweise verändert, also entweder stetig vergrößert oder stetig verkleinert, bis die im aktuellen Schritt k gemessene Leistung $P[k]$ kleiner als die Leistung $P[k-1]$ aus dem vorherigen Schritt $k-1$ ist. Erst dann wird die Richtung umgeschaltet und der Tastgrad wieder solange in diese Richtung verändert, bis $P[k] < P[k-1]$ ist und so weiter. Ist der MPP gefunden, dann pendelt der Tastgrad um den Tastgrad im MPP. Beim klassischen Perturb-And-Observe Algorithmus muss demnach immer die Leistung aus dem vorherigen Schritt, z.B. mit

einem Abtasthalteglied gespeichert werden, um dies zu umgehen wird das vorgestellte Verfahren um einen Spitzenwertdetektor erweitert. Der Spitzenwertdetektor erkennt, wenn die aktuelle Leistung $P(t)$ kleiner ist als der Maximalwert

$$P_{\max} = \max_{t \in T_{r,k}} (P(t)) \quad (11)$$

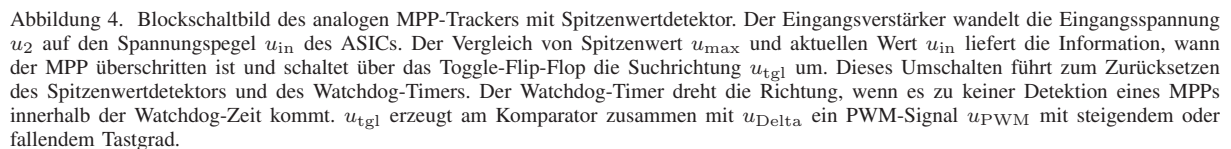
im Zeitabschnitt $T_{r,k} := [t_{k-1}, t_k]$ mit der aktuellen Richtung und schaltet dann die Richtung um. Die Umsetzung des MPP-Trackings mit Spitzenwertdetektion wird im Folgenden vorgestellt.

A. MPP-Tracking mit Spitzenwertdetektion

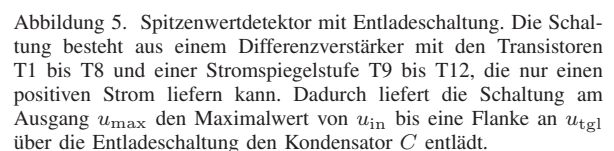
Das Blockschaltbild des MPP-Trackers ist in Abbildung 4 zusehen. Das PWM-Signal u_{PWM} mit steigendem oder fallenden Tastgrad d wird am Komparator durch Vergleich des Dreieckssignal u_Δ mit dem Signal aus dem Rampengenerator u_{Rampe} erzeugt. Die Eingangsgröße des Rampengenerators u_{tgl} speichert dabei die Information, ob der Tastgrad gerade vergrößert oder verkleinert wird. Die Eingangsgröße des MPP-Trackers ist die Ausgangsspannung des Tiefsetzstellers u_2 aus Abbildung 2, da diese Proportional zur Leistung P ist. Diese wird zunächst mit dem Eingangsverstärker auf den internen Spannungspegel u_{in} des ASICs gewandelt. Anschließend liegt u_{in} am Spitzenwertdetektor an. Dieser gibt an seinem Ausgang den im aktuellen Zeitabschnitt $T_{r,k} := [t_{k-1}, t_k]$ gespeicherten Maximalwert

$$u_{\max} = \max_{t \in T_{r,k}} (u_{\text{in}}) \quad (12)$$

von u_{in} aus. Der Zeitabschnitt $T_{r,k}$ wird von einem Reset-Signal beendet. Die beiden Spannungen u_{in} und $u_{\max}$ dienen zum Detektieren des MPP. Denn wenn $|u_{\max}| \leq |u_{\text{in}}|$ gilt, dann steigt u_{in} noch und damit auch die Leistung P und der MPP ist noch nicht überschritten. Gilt jedoch $|u_{\max}| \geq |u_{\text{in}}|$, dann fällt u_{in} und damit ist der MPP überschritten. Das wird durch die nachfolgende Differenzstelle mit Hystereseschaltung ausgewertet und führt zum Umschalten des Toggle-Flip-Flops. Das Toggle-Flip-Flop speichert mit seiner Ausgangsgröße u_{tgl} die Information in welche Richtung der Tastgrad des PWM-Signals geändert wird. Gleichzeitig setzt jede Flanke an u_{tgl} den Spitzenwertdetektor und den Watchdog-Timer zurück. Dadurch kann der Spitzenwertdetektor wieder den neuen Maximalwert $u_{\max}$ von u_{in} des neuen Zeitabschnitts $T_{r,k+1}$ speichern. Für den Fall, indem der MPP-Tracker in einem ungünstigen Betriebspunkt feststeckt und den MPP nicht findet, löst der Watchdog-Timer aus, da es zu keinem Reset durch u_{tgl} kommt. Ein ungünstiger Betriebspunkt kann entstehen, wenn das Rampensignal u_{Rampe} an der positiven oder negativen Spannungsgrenze ist und durch u_{tgl} weiter steigen oder fallen



1) *Spitzenwertdetektor*: Die Aufgabe des Spitzenwertdetektors ist es an seinem Ausgang $u_{\max}$ den aktuellen Maximalwert von u_{in} zu speichern bis ein Rücksetzbefehl an u_{tgl} kommt. Das Grundprinzip basiert dabei auf der Operationsverstärkerschaltung mit Diode am Ausgang und Kondensator mit Rückkopplung auf die Eingänge des Operationsverstärkers, wie sie in [10] vorgestellt ist. Allerdings zeigt sich bei dieser Schaltung, dass ein leichtes Überspringen der Kondensatorspannung sofort zu einer zu hohen Maximalspannung $u_{\max}$ führt, die wiederum aufgrund der Diode nicht mehr abgebaut werden kann. Deshalb wurde im Zuge der Arbeit ein neuer Spitzenwertdetektor mit einstellbarer Reglerverstärkung entwickelt, dieser ist in Abbildung 5 dargestellt. Der entwickelte Spitzenwertdetektor besteht am Eingang aus einem Differenzverstärker mit Sperrschicht-Feldeffekttransistoren (JFETs), da diese keine Gate-Ströme $I_G = 0$ besitzen und somit den Kondensator C nicht über diese laden/entladen. Die Stromspiegelkonfiguration bestehend aus T1, T3 und T5, T7 ermöglicht einen niedrigeren Eingangsspannungsbereich von



unter der Annahme, dass über der Stromquelle I mindestens $U_{\text{sq}} \approx 1 \text{ V}$ abfallen muss. Am Ausgang des Differenzverstärkers befinden sich ein NPN- und ein PNP-Stromspiegel, der aus den Transistorpaaren T9, T11 und T10, T12 besteht. Durch die Verwendung von unterschiedlichen Emittierflächen $n_{T,i}, i \in$

4

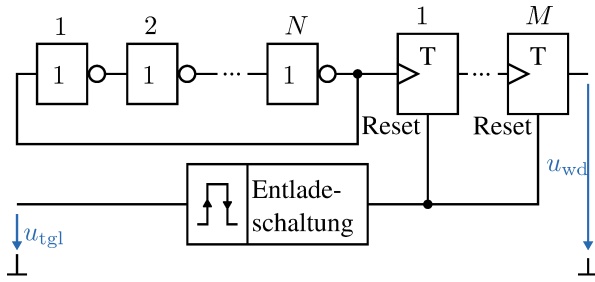


Abbildung 6. Watchdog-Timer mit Entladeschaltung. Der Ringoszillator bestehend aus Invertoren 1 bis N generiert den Takt. Toggle-Flip-Flops 1 bis M arbeiten als Frequenzteiler und passen so den Takt auf die erforderliche Watchdog-Zeit T_{wd} an. Nachdem die Watchdog-Zeit erreicht ist, schaltet Ausgang u_{wd} von LOW auf HIGH. Die Flanke an u_{tgl} setzt über die Entladeschaltung den Watchdog-Timer zurück.

{9, 10, 11, 12} lässt sich damit die Verstärkung der Stromspiegel [3] mit

$$k = \frac{n_{T11}}{n_{T9}} \frac{n_{T12}}{n_{T10}} \quad (14)$$

einstellen und so lässt sich ein Überspringen der Kondensatorspannung verhindern. Über die flankensensitive Entladeschaltung wird der Kondensator C entladen, sobald eine fallende oder steigende Flanke an u_{tgl} entsteht.

2) *Watchdog-Timer*: Der Watchdog-Timer erkennt, wenn für eine eingestellte Zeit T_{wd} kein MPP detektiert wird und schaltet dann seinen Ausgang u_{wd} von LOW auf HIGH. Der Aufbau der Schaltung ist in Abbildung 6 zusehen. Der Takt des Watchdog-Timers entsteht durch einen Ringoszillator [7], der mit den Invertoren 1 bis N aufgebaut ist. Durch eine ungerade Anzahl $N = 2n + 1, n \in \mathbb{N}$ an Invertoren ergibt sich durch die Kette eine Phasenverschiebung des Signals von 180° , was zu einer Oszillation führt. Die Frequenz des Oszillators ergibt sich aus der Summe der Verzögerungen τ der Inverter zu

$$f = \frac{1}{2N\tau} \quad (15)$$

Der anschließende Frequenzteiler, bestehend aus den Toggle-Flip-Flops 1 bis M verringert die Taktfrequenz auf die benötigte Auslösezeit

$$T_{wd} = N\tau M \quad (16)$$

Über eine flankensensitive Entladeschaltung werden die Toggle-Flip-Flops 1 bis M zurückgesetzt, sobald eine Signaländerung an u_{tgl} entsteht und der Watchdog beginnt von vorne.

3) *Rampengenerator*: Der Rampengenerator erzeugt je nachdem ob der Eingang u_{tgl} auf HIGH oder LOW ist eine steigende oder fallende Rampe und ist in Abbildung 7 dargestellt. Die Schaltung besteht aus einem Transkonduktanzverstärker (OTA), der ausschließlich im Großsignalbereich betrieben wird. Dafür ist die Ausgangsstufe des OTAs angepasst, um sehr

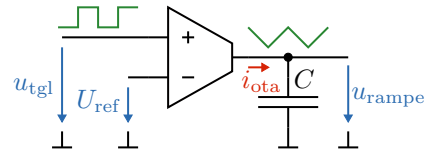


Abbildung 7. Rampengenerator bestehend aus OTA und Kondensator C . Je nachdem ob u_{tgl} HIGH oder LOW ist, entsteht am Ausgang des OTAs positiver oder negativer Ausgangsstrom i_{ota} , den der Kondensator zu rampenförmiger Spannung u_{rampe} integriert.

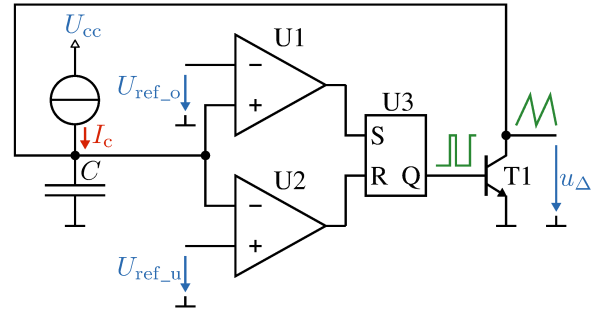


Abbildung 8. Dreieckgenerator bestehend aus Stromquelle I_c , die den Kondensator C lädt und einer Hystereseschaltung, bestehend aus U1 bis U3 und T1. Erreicht die Kondensatorspannung u_Δ die obere Schaltschwelle U_{ref_o} , dann entlädt die Hystereseschaltung den Kondensator C bis zur unteren Schaltschwelle U_{ref_u} . Dadurch entsteht eine dreieckförmige/sägezahnförmige Ausgangsspannung u_Δ .

kleine Ausgangsströme i_{ota} erzeugen zu können. Die rampenförmige Ausgangsspannung u_{rampe} ergibt sich aus der Spannungsdifferenz am Eingang $u_{tgl} - U_{ref}$ und der OTA-Verstärkung $g_{ota} := \partial i_{ota} / \partial (u_{tgl} - U_{ref})$ zu

$$u_{rampe} = \frac{1}{C} \int g_{ota} (u_{tgl} - U_{ref}) dt + U_0 \quad (17)$$

und ist je nachdem ob u_{tgl} HIGH oder LOW ist positiv oder negativ.

4) *Dreieckgenerator*: Der Dreieckgenerator liefert ein dreieck-/sägezahnförmiges Signal, das als Träger für die PWM dient. Die Ausführung des Dreieckgenerators ist in Abbildung 8 zusehen und an den RC-Kippschwinger aus [4] angelehnt.

Der Dreieckgenerator besteht aus einer Konstantstromquelle I_c die den Kondensator C lädt und einer Hystereseschaltung mit einer oberen Schaltschwelle U_{ref_o} und einer unteren Schaltschwelle U_{ref_u} . Durch die Konstantstromquelle I_c wird der Kondensator C geladen und die Spannung

$$u_\Delta = \frac{1}{C} \int_t^{t+T_\Delta} I_c dt + U_{ref_u} \quad (18)$$

steigt rampenförmig an, bis die obere Schaltschwelle U_{ref_o} erreicht ist. Das Erreichen der Schaltschwelle U_{ref_o} führt dazu, dass der Komparator U1 umschaltet und das SR-Flip-Flop U3 setzt. Dadurch schaltet der Transistor T1 und entlädt C in vernachlässigbarer Zeit bis die untere Schaltschwelle U_{ref_u} erreicht ist. Das Erreichen der unteren Schaltschwelle U_{ref_u} führt zum

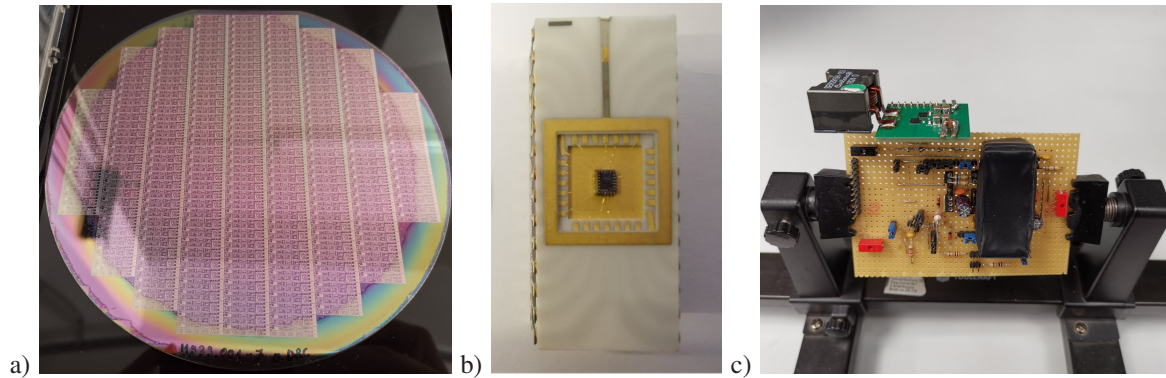


Abbildung 9. a) Produzierter Wafer der entwickelten Schaltung mit dem Smart-BCD Process von Prema Semiconductor. b) In Keramik-Gehäuse verpackter ASIC. c) Entwickelter Testaufbau, bestehend aus Lastpfadplatine mit Tiefsetzsteller und Regelungsplatine mit ASIC.

Umschalten von U_2 und damit zum Zurücksetzen von U_3 , die Schaltung beginnt nun wieder von vorne. Die Frequenz des Dreiecksgenerators

$$f_{\Delta} = \frac{I_c}{C(U_{\text{ref}_o} - U_{\text{ref}_u})} \quad (19)$$

ergibt sich durch Einsetzen von $u_{\Delta} = U_{\text{ref}_o}$ in Gleichung 18 und anschließendem Auflösen.

III. ERGEBNISSE UND DISKUSSION

Der nachfolgende Abschnitt zeigt den realisierten ASIC und stellt die Messergebnisse vor.

A. ASIC und Testschaltung

Die vorgestellte Schaltung wurde bei der Prema Semiconductor GmbH im Smart-BCD Process auf einem 6 Zoll Wafer hergestellt. Der Wafer ist in Abbildung 9.a) zusehen. Der Smart-BCD Process ermöglicht NPN- und PNP-Transistoren mit Kollektor-Strömen von bis zu 1 A und Spannungen bis zu 80 V [11]. Anschließend wurde der „Die“ des ASICs in einem Dual-In-Line (DIL) Keramik-Gehäuse verpackt (Abbildung 9.b)). Um den ASIC in Betrieb zunehmen, wurde der Hardwareaufbau aus Abbildung 9.c) entwickelt. Der Aufbau besteht aus dem Lastpfad, der über eine Klemmleiste auf die Regelungsplatine aufgesteckt werden kann. Auf der Regelungsplatine befindet sich der entwickelte ASIC, der durch eine schwarze Box gegen eintreffende Photonen geschützt ist, die einen Fehlerstrom verursachen würden. Der Lastpfad besteht aus einem Tiefsetzsteller wie in Abbildung 2 vorge stellt.

B. Messergebnisse

Zur Auswertung des entwickelten ASICs wurde an die Testschaltung ein PV-Modul an den Eingang des Tiefsetzstellers und an den Ausgang ein Lastwiderstand R_L angeschlossen. Dabei wurde der Lastwiderstand variiert, um den Arbeitsbereich des ASICs aufzunehmen. Abbildung 10 stellt die Ergebnisse der

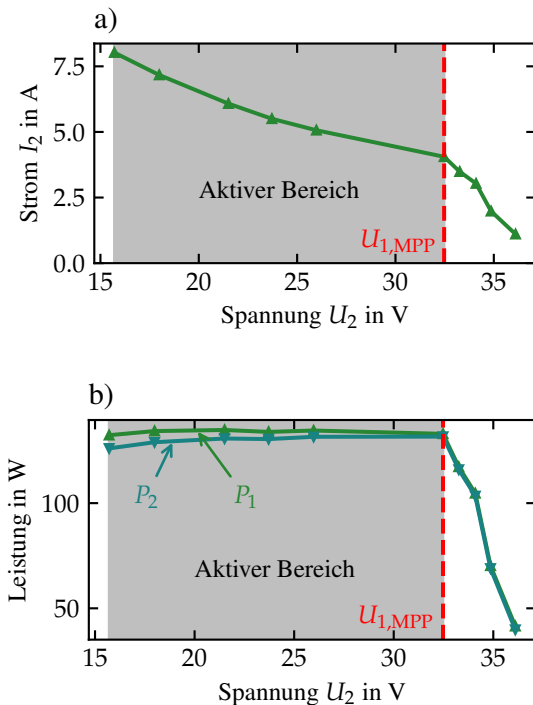


Abbildung 10. Messung der Testschaltung mit ASIC am PV-Modul und Variation des Lastwiderstands R_L . a) Ausgangsstrom I_2 über der Ausgangsspannung U_2 des Tiefsetzstellers. b) Eingangsleistung P_1 und Ausgangsleistung P_2 über der Ausgangsspannung U_2 des Tiefsetzstellers. Leistungen P_1 entspricht im aktiven Bereich der MPP-Leistung.

Messung vor. Dabei zeigt Abbildung 10.a) den Strom-Spannungsverlauf am Ausgang des Tiefsetzstellers. Der aktive Bereich beginnt ab der MPP-Spannung des PV-Moduls, dann kann der Tiefsetzsteller die Spannung U_2 nach dem Zusammenhang aus Gleichung 3 kleiner machen. Im aktiven Bereich ist dann die Eingangsleistung konstant und entspricht der MPP-Leistung, was auch in Abbildung 10.b) zusehen ist. Die Leistung am Ausgang des Tiefsetzstellers P_2 ist aufgrund von Verlusten der Leistungselektronik etwas

kleiner als P_1 . Aufgrund der konstanten Leistung P_2 und dem Zusammenhang am ohmschen Widerstand ergibt sich für den Strom im aktiven Bereich der hyperbolische Verlauf

$$I_2 = \frac{P_2}{U_2}. \quad (20)$$

Aus diesem Zusammenhang resultiert die untere Spannungsgrenze des aktiven Bereichs, da der Strom I_2 umso größer wird, je kleiner die Spannung U_2 ist und damit der maximale Strom der Induktivität und der Transistoren erreicht wird. Die Spannungsgrenze wird extern über den Strom I_2 eingestellt. Die Messergebnisse zeigen also das gewünschte Verhalten des MPP-Trackers, das PV-Modul wird im aktiven Bereich im MPP gehalten.

IV. ZUSAMMENFASSUNG

Um Moduloptimierer möglichst kostengünstig durch das Erhöhen der Taktfrequenz zu realisieren wurde in diesem Paper ein analoger MPP-Tracker entwickelt, der zur Integration in einen ASIC geeignet ist. Das Verfahren des MPP-Trackers basiert auf dem Perturb-And-Observe Algorithmus, der jedoch um einen Spitzenwertdetektor erweitert wurde. Der Spitzenwertdetektor wurde für diese Arbeit neu entwickelt und basiert auf einem Differenzverstärker mit Stromspiegel-Ausgangsstufe und Kondensator. Anschließend erzeugt ein Rampengenerator zusammen mit einem Dreiecksgenerator an einem Komparator das PWM-Signal für die Transistoren. Weiter besitzt die Schaltung einen Watchdog-Timer, um ein Feststecken in einem ungünstigen Arbeitspunkt zu verhindern und sorgt so für einen robusten Betrieb. Die Funktion des ASIC wurde anhand einer Testhardware gezeigt. Dafür wurde der verpackte ASIC auf einer Regelungsplatine zum Ansteuern einer Leistungselektronik verwendet. Die Messergebnisse beweisen die Funktion des ASICs und zeigen, dass das angeschlossene PV-Modul im geforderten Arbeitsbereich der Schaltung im MPP gehalten wird.

DANKSAGUNG

Die Autoren bedanken sich bei der BRC Solar GmbH und der Prema Semiconductor GmbH für die Förderung dieser Arbeit und ganz besonders bei Jonas Bargon, Richard Brace und Timm Czarnecki für die fachliche Unterstützung. Weiterhin gilt ein Dank dem Bundesministerium für Wirtschaft und Klima für die Finanzierung der Weiterentwicklung des analogen MPPT im Rahmen des Bundesprojekts Solarpark 2.0 mit der Nummer FKZ: 03EE1135C.

LITERATURVERZEICHNIS

- [1] Richard Brace, Angelika Neumann und Rainer Merz. „Kostengünstige Ertragssteigerung teilverschatteter PV-Stränge“. In: *Photovoltaische Solarenergie Bad Staffelstein*. Bd. 33. Apr. 2018.
- [2] Robert W. Erickson und Dragan Maksimovic. *Fundamentals of Power Electronics*. 3. Auflage. Springer-Verlag, 2020. ISBN: 9783030438791.
- [3] Paul R. Gray, Paul J. Hurst, Stephen H. Lewis und Robert G. Meyer. *Analysis and Design of Analog Integrated Circuits*. 5. Edition. Wiley, 2010. ISBN: 978-0470398777.
- [4] A. Grebene. *Bipolar and MOS Analog Integrated Circuit Design*. 1. Auflage. John Wiley & Sons, Inc., 2002.
- [5] *How to select input capacitors for a buck converter*. Texas Instruments. 2016. URL: https://www.ti.com/lit/an/slyt670/slyt670.pdf?ts=1671014830265&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [6] H. El-Khozondar, Rifa El-Khozondar, Khaled Matter und Teuvo Suntio. „A review study of photovoltaic array maximum power tracking algorithms“. In: *Renewables: Wind, Water, and Solar* 3 (Feb. 2016). DOI: 10.1186/s40807-016-0022-8.
- [7] Mrinal Mandal und Bishnu Charan Sarkar. „Ring oscillators: Characteristics and applications“. In: *Indian Journal of Pure and Applied Physics* 48 (Feb. 2010), S. 136–145.
- [8] Angelika Neumann, T Czarnecki und R. Merz. „>99% Spitzenwirkungsgrad mit Low-Cost-MPPT statt Poweroptimizer“. In: *Photovoltaische Solarenergie Bad Staffelstein*. Bd. 32. März 2017.
- [9] Dierk Schröder und Rainer Marquardt. *Leistungselektronische Schaltungen*. 4. Auflage. Springer Vieweg, 2019. ISBN: 978-3662553244.
- [10] Ulrich Tietze, Christoph Schenk und Eberhard Gamm. *Halbleiter-Schaltungstechnik*. 16. Auflage. Springer Vieweg, 2019. ISBN: 978-3662485538.
- [11] *Triple Well Smart-BCD Process with only 11 Mask Layers - A Process of Unsurpassed Simplicity and Modularity*. PREMA Semiconductor. URL: https://www.prema.com/index.php/de/know-how/prema-prozess/item/download/47_01894e25558788835b2d2467ef8f6390.



Patrick Mader erhielt den akademischen Grad B.Eng. in Elektrotechnik mit der Fachrichtung Energietechnik und Erneuerbare Energien von der Hochschule Karlsruhe im Jahr 2019. Anschließend erhielt er im Jahr 2023 den M.Sc. in Elektrotechnik mit der Fachrichtung elektrische Antriebe und Leistungselektronik vom Karlsruher Institut für Technologie. Derzeit arbeitet er an der Hochschule Karlsruhe im Bereich Leistungselektronik für Erneuerbare Energien an seiner Promotion.



Sascha Eckerter erhielt den akademischen Grad des Bachelor of Engineering in Mechatronik im Jahr 2020 von der Hochschule Karlsruhe (HKA). Dort erhielt er ebenfalls den akademischen Grad des Master of Science in Elektrotechnik im Jahr 2022 mit Spezialisierung im Bereich Erneuerbaren Energien und Energietechnik. Bis dato arbeitet er an der HKA im Bereich der Regenerativen Energien an seiner Promotion und entwickelt im Zuge seiner Dissertation

eine Simulationsumgebung für den Jahresertrag von Solarparks mit neuer Solartechnik.



Rainer Merz studierte Elektrotechnik, Fachrichtung Opto- und Mikroelektronik an der Universität Stuttgart. Im Anschluss wurde er am Institut für Photovoltaik der Universität Stuttgart promoviert. Von 2011 bis 2014 leitete Rainer Merz die Abteilung Insel und Speichersysteme beim Wechselrichterhersteller REFUsol. Im Herbst 2014 wurde er als Professor für Regenerative Energiesysteme an die Hochschule Karlsruhe berufen.

Materialcharakterisierung basierend auf einem 120 GHz FMCW-Radar

Marius Patzer, Nicolas Treier, Marlene Harter, Serdal Ayhan, Herman Jalli Ng

Zusammenfassung—Dieses Paper präsentiert eine innovative Methode zur Materialcharakterisierung unter Verwendung von Millimeterwellen. Ein selbstentwickelter 120 GHz Transceiver-Chip dient als Frequency-Modulated-Continuous-Wave-(FMCW)Radarsensor, um anhand der gemessenen Laufzeitunterschiede eine präzise Unterscheidung verschiedener Materialien zu ermöglichen. Das ausgeklügelte Messkonzept verursacht deutlich geringere Kosten als die Messung mit einem Vektor-Netzwerkanalysator (VNA). Durch die Anwendung eines hochgenauen Signalverarbeitungsalgorithmus erfolgt eine detaillierte Auswertung der aufgezeichneten Daten. Diese innovative Methode eröffnet vielversprechende Perspektiven für weiterführende Forschungen und Entwicklungen.

Schlüsselwörter—Radar, Materialcharakterisierung, relative Permittivität, FMCW, Millimeterwellen

I. EINLEITUNG

In der modernen Materialwissenschaft und im Ingenieurwesen ist die präzise Charakterisierung von Materialien von entscheidender Bedeutung für die Entwicklung innovativer Produkte und Technologien. In den letzten Jahren wurden deshalb viele Forschungsarbeiten im Bereich der Materialcharakterisierung getätigt [1][2][3].

Mit dem Einsatz von Frequency-Modulated-Continuous-Wave-(FMCW)Radarsensoren lassen sich detaillierte Analysen verschiedenster Materialien durchführen. Insbesondere der Einsatz von Millimeterwellen ermöglicht es, Materialien mit hoher Auflösung und Empfindlichkeit zu durchdringen sowie eine ausgezeichnete Oberflächenauflösung zu bieten. Durch diese Technologien können Strukturen und Eigenschaften eines Materials charakterisiert werden, ohne die Integrität zu beeinträchtigen [4].

Des Weiteren spielen FMCW-Radare eine wichtige Rolle in der Medizintechnik, insbesondere bei der Bildgebung und Diagnose, indem sie es ermöglichen, Gewebe und biologische Materialien präzise zu analysieren, ohne invasive Eingriffe vornehmen zu müssen. FMCW-Radare bieten außerdem mehrere Vorteile

Marius Patzer maleopa@gmx.de, Serdal Ayhan serdal.ayhan@h-ka.de, Herman Jalli Ng herman-jalli.ng@h-ka.de, Fakultät Elektro- und Informationstechnik, Hochschule Karlsruhe, 76133 Karlsruhe, Deutschland.

Nicolas Treier nicolas.treier@hs-offenburg.de, Marlene Harter marlene.harter@hs-offenburg.de, Fakultät Elektrotechnik, Medizintechnik und Informatik, Hochschule Offenburg, 77652 Offenburg, Deutschland.

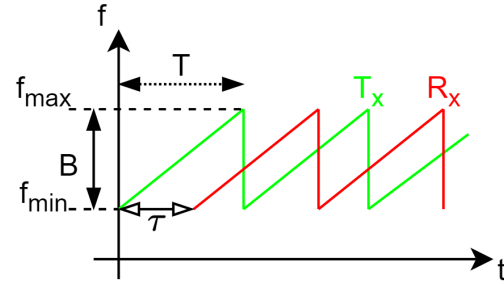


Abbildung 1. Prinzipdarstellung FMCW-Radar.

gegenüber Vektor-Netzwerkanalysatoren (VNAs), wie beispielsweise die Erzielung einer sehr niedrigen Zwischenfrequenz (ZF)-Bandbreite und die Reduzierung der Komplexität und der Kosten der Analog-Digital (AD)-Wandler [5].

II. KONZEPT / PRINZIP

A. FMCW-Prinzip

Bei einem FMCW-Radarsensor wird die Frequenz des Sendesignals über die Zeit verändert. Es entsteht eine Frequenzrampe mit einer Bandbreite B und einer Anstiegszeit T (vgl. Abbildung 1).

Das Empfangssignal R_x wird zur Auswertung mit dem Sendesignal T_x gemischt. Das daraus resultierende Signal s_{ZF} besitzt die Frequenz f_{ZF} , welche als Zwischenfrequenz bezeichnet wird.

$$s_{ZF}(t) = \cos(2\pi \cdot f_{ZF} \cdot t + \varphi_{ZF}) \quad (1)$$

$$f_{ZF} = f_{T_x} - f_{R_x} = k \cdot \tau = \frac{B}{T} \cdot \frac{2R}{c_0} \quad (2)$$

Die Zwischenfrequenz f_{ZF} ist deutlich niedriger als die ausgesendete Frequenzrampe des Sendesignals. Das resultierende Signal s_{ZF} kann somit direkt von einem Analog-Digital-Wandler abgetastet werden, ohne vorher heruntergemischt werden zu müssen. Die Variable k gibt die Steigung der Frequenzrampe an. Diese besteht aus dem Verhältnis zwischen der Bandbreite B und der Anstiegszeit T . Proportional mit der Höhe der Zwischenfrequenz f_{ZF} verändert sich auch die Distanz R zum Ziel (Abbildung 2).

Wenn sich das Ziel im Detektionsbereich des Radarsensors befindet, kann die Laufzeit τ des Sende-

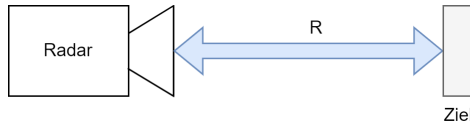


Abbildung 2. Darstellung der Signallaufzeit.

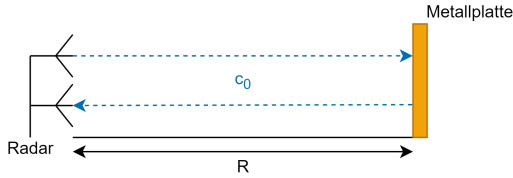


Abbildung 3. Laufzeitmessung zur Metallplatte.

und Empfangssignals aus der gemessenen Zwischenfrequenz f_{ZF} wie in (3) berechnet werden:

$$\tau = \frac{f_{ZF}}{\frac{B}{T}} = \frac{2R}{c_0} \quad (3)$$

Die Ausbreitungsgeschwindigkeit der elektromagnetischen Welle in der Luft wird dabei als die Lichtgeschwindigkeit c_0 angenommen.

B. Messung von Laufzeitunterschieden

Um ein Material charakterisieren zu können, muss die Ausbreitungsgeschwindigkeit c_1 der elektromagnetischen Welle in diesem Medium bestimmt werden. Dazu wird in einer definierten Distanz R eine Metallplatte platziert (vgl. Abbildung 3) und die Laufzeit des Signals über die gemessene Zwischenfrequenz f_{ZF_0} ermittelt (siehe (4)).

$$\tau_0 = \frac{f_{ZF_0}}{\left(\frac{B}{T}\right)} = \frac{2R}{c_0} \quad (4)$$

Diese Messung wird im weiteren Verlauf als Referenzmessung bezeichnet. Im nächsten Schritt wird ein Material mit einer bekannten Dicke D vor die Metallplatte gestellt. Die Distanz zwischen der Metallplatte und dem Radarsensor wird dabei nicht verändert (vgl. Abbildung 4).

Das empfangene Signal enthält zwei verschiedene Frequenzen f_{ZF_0} und f_{ZF_1} . f_{ZF_0} entspricht der Distanz zwischen der vorderen Kante des zu charakterisierenden Materials und dem Radarsensor, also $R - D$. Die Frequenz f_{ZF_1} entspricht wie in Abbildung 3 dargestellt der Distanz zwischen der Metallplatte und dem Radarsensor. Dadurch lässt sich folgende Gleichung für die Laufzeit τ_1 aufstellen:

$$\tau_1 = \frac{f_{ZF_1}}{\left(\frac{B}{T}\right)} = \frac{2R}{c_0} - \frac{2D}{c_0} + \frac{2D}{c_1} \quad (5)$$

Die Laufzeit τ_1 ist größer als τ_0 , da die Ausbreitungsgeschwindigkeit der elektromagnetischen Welle

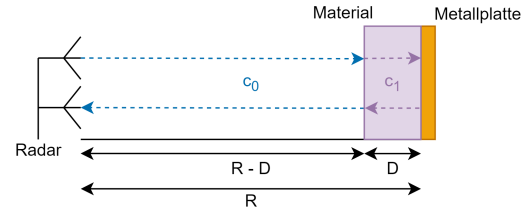


Abbildung 4. Laufzeitmessung zu Material und Metallplatte.

im Medium gedämpft wird. Daraus ergibt sich die Differenz der Laufzeiten aus (4) und (5). Diese lassen sich wie folgt nach der Ausbreitungsgeschwindigkeit c_1 auflösen:

$$\tau_1 - \tau_0 = -\frac{2D}{c_0} + \frac{2D}{c_1} \quad (6)$$

$$c_1 = \frac{c_0}{\sqrt{\epsilon_r \mu_r}} = \frac{2D}{\tau_1 - \tau_0 + \frac{2D}{c_0}} \quad (7)$$

Da die relative Permeabilität μ_r in den meisten Materialien nahe bei 1 liegt, kann diese für die weitere Betrachtung vernachlässigt werden. Die relative Permittivität ϵ_r eines Materials bestimmt sich daher wie folgt:

$$\epsilon_r = \left(c_0 \cdot \frac{\tau_1 - \tau_0 + \frac{2D}{c_0}}{2D} \right)^2 \quad (8)$$

III. HARDWARE

Die für die Messung verwendeten Komponenten lassen sich durch drei wesentliche Funktionsblöcke darstellen (siehe Abbildung 5). Diese Funktionsblöcke werden in den nachfolgenden Abschnitten erläutert.

A. PLL

Um den Frequenz-Sweep für das Sendesignal zu erzeugen, wird eine Phasenregelschleife (phase-locked loop (PLL)) eingesetzt. Über diese wird sichergestellt, dass die erzeugten Frequenzen des Sweeps stabil sind und nicht driften.

Der PLL-Chip *ADF4159* befindet sich auf einem Entwicklungsboard von *Analog Devices*. Mithilfe einer USB-Schnittstelle können die Register über ein Computerprogramm passend für den Anwendungsfall konfiguriert werden.

B. Sensor

Als Radarsensor wird ein miniaturisierter 120 GHz 2-Kanal-Radarsensor im System-on-Chip-(SoC) Design für Single Input Multiple Output-(SIMO) und Multiple Input Multiple Output-(MIMO) Radaranwendungen verwendet. Dieser Chip ist ausgestattet mit Binary Phase-Shift Keying-(BPSK) Modulatoren und I/Q-Empfänger sowie einem 30 GHz Voltage-Controlled Oscillator (VCO) und einem Frequenzvervierfacher. Die On-Chip gefalteten Dipolantennen

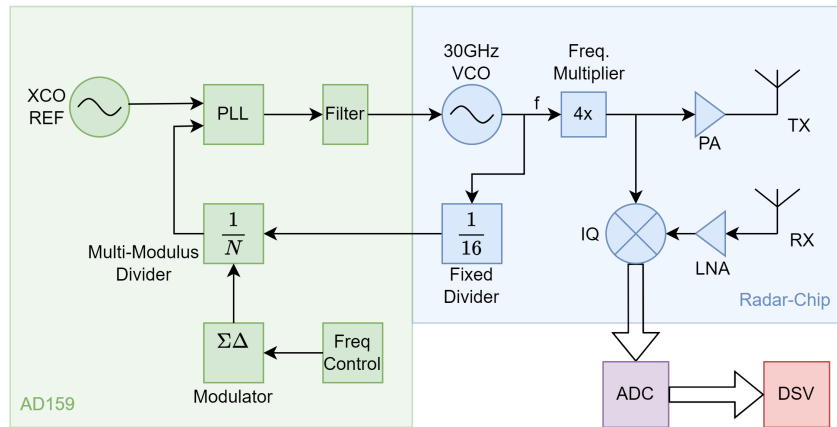


Abbildung 5. Schematische Darstellung der Transceiverarchitektur.

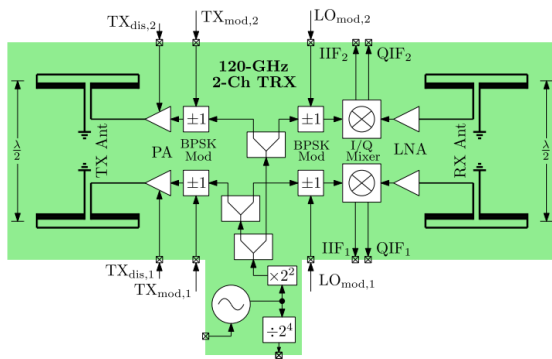


Abbildung 6. Blockdiagramm des 120-GHz Radarsensors [6].

sind mit einer Rückseitenätztechnik aufgetragen [6]. In Abbildung 6 ist das Blockschaltbild des Sensors dargestellt.

Der Radarchip befindet sich auf einer Entwicklungsplatine, auf der die Anschlüsse über Stiftleisten herausgeführt sind. Mit BNC-Kabeln wird ein I/Q Empfangskanal des Radarsensors über ein USB-Oszilloskop ausgelesen.

C. Analog/Digital-Wandlung

Die Abtastung des resultierenden Signals s_{ZF} erfolgt mittels eines USB-Oszilloskop, dem *Analog Discovery 2 (AD2)* von *Digilent*. Von dem PLL-Chip aus Kapitel III-A wird ein „Trigger“-Signal gesendet, sobald ein neuer Frequenz-Sweep erfolgt. Dieses Signal startet die Aufzeichnung des Analog/Digital-Wandlers.

D. Digitale Signalverarbeitung

Ein Python-Programm übernimmt sowohl die Steuerung des USB-Oszilloskops sowie die Durchführung der digitalen Signalverarbeitung (DSV). Mithilfe der mitgelieferten Dynamic Link Library (DLL) werden die aufgezeichneten Daten aus dem Speicher des USB-Oszilloskops ausgelesen und in einer CSV-Datei abgelegt. Die aufgezeichneten Daten werden anschließend

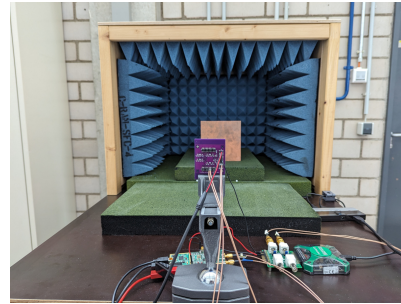


Abbildung 7. Messaufbau mit selbst gebauter Messkammer.

gemittelt, um das Rauschen zu minimieren. Weiterhin wird eine Fast Fourier Transformation (FFT) durchgeführt, um die Lage der Frequenzpeaks zu lokalisieren. Diese können in einem weiteren Schritt mittels einer Zoom FFT genauer analysiert werden. Aus den ermittelten Zwischenfrequenzen und den berechneten Laufzeiten wird nun wie in (8) beschrieben, die relative Permittivität bestimmt.

IV. MESSAUFBAU

Der Messaufbau erfolgt nach dem vorgestellten Prinzip aus Kapitel II-B. In einem Abstand von 60 cm wird eine Metallplatte vor dem Radarsensor platziert (vgl. Abbildung 7).

Die Metallplatte befindet sich in einer selbstgebauten Messkammer, die mit Pyramidenabsorber ausgekleidet ist, um die Umgebungseinflüsse zu minimieren. Mithilfe einer Linse auf dem Radarsensor wird sichergestellt, dass nur der Bereich, in dem sich die Metallplatte befindet, erfasst wird.

Nach der Referenzmessung werden verschiedene Dicken von Kunststoffblöcken vor die Metallplatte platziert und die Laufzeit wie in Abbildung 4 ermittelt.

Die verwendeten Kunststoffblöcke sind in Abbildung 8 zu sehen. Tabelle I listet weitere Materialeigenschaften auf.

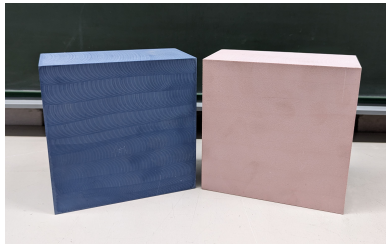


Abbildung 8. Kunststoff blau und Kunststoff braun.

Tabelle I
MATERIALEIGENSCHAFTEN

Material	Eigenschaften	Verwendete Größen
Metallplatte	Kupfer Dichte $8,92 \frac{\text{g}}{\text{cm}^3}$	Blockgröße: $150 \text{ mm} \times 150 \text{ mm}$ Dicke: 2 mm
Kunststoff braun	SikaBlock® M600 N Polyurethan, hellbraun Dichte $0,6 \frac{\text{g}}{\text{cm}^3}$	Blockgröße: $150 \text{ mm} \times 150 \text{ mm}$ Dicken: 25 mm, 50 mm, 75 mm, 100 mm
Kunststoff blau	SikaBlock® M960 Polyurethan, blau Dichte $1,2 \frac{\text{g}}{\text{cm}^3}$	

V. MESSERGEBNISSE

A. Auswertung der FFT Daten

Die aufgezeichneten Daten werden wie in Kapitel III-D beschrieben ausgewertet. In Abbildung 9 ist die FFT der Referenzmessung nach dem Messprinzip aus Abbildung 3 dargestellt. Dabei zeigt sich ein signifikanter Peak bei $f_{ZF_0} = 30,64 \text{ kHz}$, der von der Metallplatte verursacht wird.

Abbildung 10 zeigt die FFT einer Messung mit Kunststoffblock und Metallplatte. Man erkennt zwei signifikante Frequenzpeaks bei $f_{ZF_0} = 29,19 \text{ kHz}$ und $f_{ZF_1} = 33,18 \text{ kHz}$. Die erste Frequenz f_{ZF_0} entspricht der Distanz zwischen der vorderen Kante des Kunststoffblocks und dem Radarsensor. Die Distanz zwischen Metallplatte und Radarsensor wird durch die zweite Frequenz f_{ZF_1} repräsentiert (vgl. Kapitel II-B).

Unter Annahme einer identischen Ausbreitungsgeschwindigkeit erscheint die Metallplatte weiter entfernt, obwohl die Positionierung der Metallplatte bei beiden Messungen identisch ist. Die Frequenzverschiebung und damit die erhöhte Laufzeit τ_1 des Signals im Vergleich zur Referenzmessung τ_0 resultiert jedoch aus der verringerten Ausbreitungsgeschwindigkeit der elektromagnetischen Welle im Kunststoffblock. Durch die bekannte Dicke des Kunststoffblocks und damit der Entfernung zum Radarsensor kann der Messaufbau über die Frequenz f_{ZF_0} verifiziert werden.

B. Auswertung der relativen Permittivität

Wie die Messungen gezeigt haben, können für die Charakterisierung der unterschiedlichen Materialien nicht alle Materialstärken zur Auswertung herangezogen werden. In der FFT des braunen Kunststoffblocks

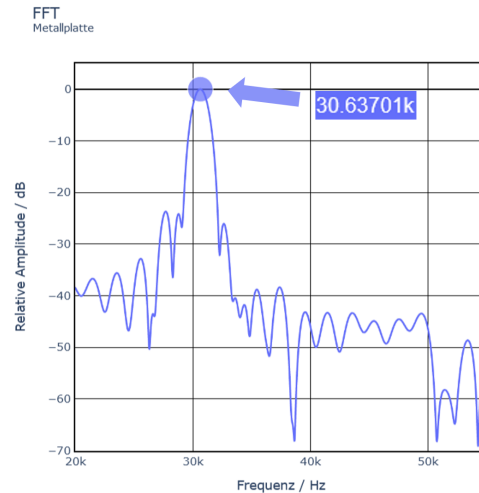


Abbildung 9. FFT-Plot der Messung mit Metallplatte.

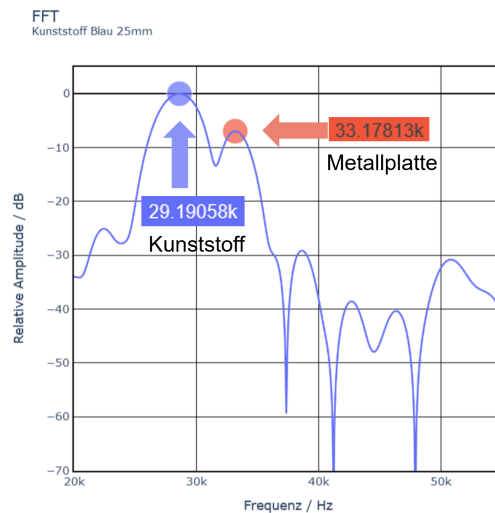


Abbildung 10. FFT-Plot der Messung mit Kunststoffblock und Metallplatte.

mit der Dicke 25 mm sind die beiden Frequenzpeaks überlagert und ermöglichen somit keine eindeutige Bestimmung der Laufzeit.

Die blauen Kunststoffblöcke ab einer Dicke von 50 mm absorbieren das ausgesendete Signal so weit, dass der Frequenzpeak der Metallplatte im Rauschen des Signals verschwindet. Das gleiche Phänomen tritt bei dem braunen Kunststoffblock mit einer Dicke von 100 mm auf.

Aus diesem Grund werden bei dem blauen Kunststoff nur die Dicke 25 mm und bei dem braunen Kunststoff nur die Dicken 50 mm und 75 mm für die Ermittlung der relativen Permittivität herangezogen. Die berechneten Werte des braunen Kunststoffs werden anschließend gemittelt. In Tabelle II sind die Ergebnisse aufgelistet:

Es liegen keine Referenzwerte für die verwendeten Materialien bei 120 GHz vor. Daher können die er-

Tabelle II
 RELATIVE PERMITTIVITÄTEN DER VERWENDETEN KUNSTSTOFFE

Kunststoff	Relative Permittivität
Braun	2,05
Blau	3,22

mittelten Ergebnisse nur als Anhaltspunkte für weitere Messungen verwendet werden. Durch ein Referenzmaterial mit bekannter relativer Permittivität ϵ_r bei 120 GHz kann das Messprinzip verifiziert werden.

VI. ZUSAMMENFASSUNG & AUSBLICK

Die Ergebnisse der Messungen zeigen, dass grundsätzlich eine Bestimmung der relativen Permittivität für unbekannte Materialien möglich ist. Für zukünftige Messungen wird ein Referenzmaterial bei 120 GHz für die Einordnung der Messwerte benötigt. Statistisch signifikante Aussagen sind über mehrere geeignete Materialstärken zu erreichen. Über eine genauere Positionierung der Materialproben kann der Messaufbau optimiert und die Wiederholungsgenauigkeit der Messungen verbessert werden.

Mit dem Radarsensor soll es künftig möglich sein, feine Materialunterschiede und Strukturen zu erkennen.

LITERATURVERZEICHNIS

- [1] J. Jebramcik, I. Rolfes and J. Barowski, „A Novel Calibration Technique for FMCW Radar Systems Enabling Material Characterization in Variable Distances,“ 2019 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), Bochum, Germany, 2019, pp. 106-108, doi: 10.1109/IMWS-AMP.2019.8880110.
- [2] S. Battistutta, M. Bressan, M. Bozzi and L. Perregriani, „Material characterization through a full-wave approach based on the BI-RME method,“ 2017 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), Pavia, Italy, 2017, pp. 1-3, doi: 10.1109/IMWS-AMP.2017.8247348.
- [3] G. L. Friedsam and E. M. Biebl, „Precision free-space measurements of complex permittivity of polymers in the W-band,“ 1997 IEEE MTT-S International Microwave Symposium Digest, Denver, CO, USA, 1997, pp. 1351-1354 vol.3, doi: 10.1109/MWSYM.1997.596579.
- [4] M. S. Hilario et al., „W -Band Complex Permittivity Measurements at High Temperature Using Free-Space Methods,“ in IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 9, no. 6, pp. 1011-1019, June 2019, doi: 10.1109/TCPMT.2019.2912837.
- [5] S. Abouzaid, T. Jaeschke, J. Barowski and N. Pohl, „FMCW Radar-Based Material Characterization Using Convolutional Neural Network and K-Means Clustering,“ 2022 24th International Microwave and Radar Conference (MIKON), Gdansk, Poland, 2022, pp. 1-4, doi: 10.23919/MIKON54314.2022.9924681.
- [6] H. J. Ng and D. Kissinger, „Highly Miniaturized 120-GHz SIMO and MIMO Radar Sensor With On-Chip Folded Dipole Antennas for Range and Angular Measurements,“ in IEEE Trans. on Microwave Theory and Techniques, vol. 66, no. 6, pp. 2592-2603, June 2018.



Marius Patzer wurde 1998 in Karlsruhe geboren. Nach seinem Abitur 2018 absolvierte er ein Bachelorstudium im Bereich Nachrichtentechnik an der Dualen Hochschule Baden-Württemberg (DHBW) Karlsruhe, das er im September 2021 erfolgreich abschloss. Im Anschluss war er ein Jahr lang bei seinem ehemaligen dualen Partner Physik Instrumente (PI) GmbH & Co. KG im Bereich der Sensor- und Firmware-Entwicklung tätig. Seit Oktober

2022 absolviert Marius Patzer den Masterstudiengang im Bereich Informationstechnik an der Hochschule Karlsruhe.



Nicolas Treier schloss sein Bachelorstudium in Elektrotechnik und Informationstechnik an der Hochschule Offenburg (HSO) erfolgreich im Jahr 2021 ab. Im Jahr 2023 beendete er sein Masterstudium an der HSO mit Auszeichnung. Seit Mai 2023 arbeitet er am Institute for Unmanned Aerial Systems als Mitarbeiter mit einem Fokus auf Millimeterwellen-Anwendungen in der Medizin.



Marlene Harter schloss ihr Studium der Elektro- und Informationstechnik zum Dipl.-Ing. an der Universität Karlsruhe (TH) im Jahr 2008 ab. Nach ihrem Studium arbeitete sie als wissenschaftliche Mitarbeiterin an der École Polytechnique de Lausanne und anschließend bei der Siemens AG. Im Jahr 2014 promovierte Marlene Harter in Zusammenarbeit mit der Siemens AG am Karlsruher Institut für Technologie (KIT). Während ihrer Promo-

tion forschte sie über ein dreidimensional bildgebendes Radarsystem mit digitaler Strahlformung für industrielle Anwendungen. Von 2014 bis 2018 arbeitete sie bei der Robert Bosch GmbH in Leonberg im Bereich der radarbasierten Fahrerassistenzsysteme. In dieser Zeit leitete sie mehrere Kundenprojekte und war Projektleiterin für das Seitenradar der 5. Generation für PKWs und LKWs. Im Oktober 2018 wurde Marlene Harter als Professorin an die Hochschule Offenburg im Bereich der Hochfrequenztechnik und Elektronik berufen. Dort ist sie seit 2022 Leiterin des Institute for Unmanned Aerial Systems. Sie erhielt 2008 den Werner von Siemens Excellence Award für ihre Diplomarbeit und den Best Paper Award auf der IEEE CIE International Conference on Radar in 2011. Im Jahr 2023 war sie Technical Program Chair der 20th European Radar Konferenz (EuRAD) in Berlin.



Serdal Ayhan schloss im Jahr 2009 sein Diplomstudium der Nachrichtentechnik/Elektronik und im Jahr 2010 das nachfolgende Masterstudium Informationstechnik an der Hochschule Mannheim ab. Im Jahr 2015 promovierte er am Institut für Hochfrequenztechnik und Elektronik des Karlsruher Instituts für Technologie (KIT) mit Schwerpunkt industrielle Radar-Sensorik für hochgenaue, radarbasierte Abstandsmessungen im Mikrometer-Bereich.

Nach seiner Promotion hat er bei der Valeo Schalter und Sensoren GmbH als Systemingenieur in der Vorausentwicklung für 79 GHz Radarsensoren in Fahrzeugen gearbeitet. In den Jahren 2016-2020 arbeitete er bei der SEW-Eurodrive GmbH & Co KG im Fachkreis Funk & Navigation an der Integration von Radarsensoren in smarte, autonom fahrende Förderfahrzeuge. Seit 2020 ist er an der Fakultät für Elektro- und Informationstechnik der Hochschule Karlsruhe Professor für Hochfrequenz- und Radartechnik.



Herman Jalli Ng schloss im Jahr 2005 sein Studium als Dipl.-Ing.(FH) an der Hochschule Karlsruhe in Deutschland ab und erwarb 2014 den Dokortitel in Mechatronik an der Johannes Kepler Universität Linz in Österreich. Von 2005 bis 2009 war er als ASIC-Entwickler bei der Robert Bosch GmbH in Reutlingen tätig, bevor er von 2009 bis 2015 als wissenschaftlicher Mitarbeiter an der Johannes Kepler Universität Linz arbeitete. In den Jahren

2015 bis 2020 leitete er erfolgreich die mm-Wave Wireless Forschungsgruppe am IHP Microelectronics. Seit 2020 ist er Professor an der Hochschule Karlsruhe. Er hat über 100 Zeitschriften- und Konferenzartikel verfasst, sowohl als Autor als auch als Co-Autor. Für seine wegweisenden Arbeiten im Bereich integrierter skalierbarer mm-Wellen-Radar-Transceiver in SiGe-BiCMOS-Technologien wurde ihm der VDE ITG-Preis 2018 und der APMC Best Paper Award 2019 verliehen. Zudem war er Mitpreisträger des IEEE Microwave Prize Award 2023 der IEEE Microwave Theory and Technology Society (MTT-S), des VDE/VDI GMM-Preises 2017 und des GAAS Association Student Fellowship 2019.

Realisierung von elektronischen und sensorischen Komponenten mittels FDM-3D-Druck-Verfahren

Jannik Söll, Patrick Hog, Elke Mackensen

Zusammenfassung—Im Rahmen dieses Papers wird auf die Realisierung von elektronischen und sensorischen Komponenten mittels FDM-3D-Druck-Verfahren unter Einsatz von leitenden Filamenten eingegangen. Präsentiert werden Untersuchungsergebnisse leitfähiger Filamente, daraus abgeleitete Designrichtlinien und die Ergebnisse aus der Realisierung beispielhafter Komponenten.

Schlüsselwörter—3D-Druck, gedruckte Schaltungen, gedruckte Elektronik

I. EINLEITUNG

Gedruckte Elektronik wird zu diesem Zeitpunkt üblicherweise durch zweidimensionale Druckverfahren hergestellt. Etablierte Verfahren zweidimensionalen Drucks sind klassische Verfahren wie die PCB-Herstellung mittels Ätztechniken und der Druck funktionaler Tinten oder leitender Pasten auf flache Oberflächen. Beispiele hierfür sind der Tiefdruck, der Siebdruck oder der Tintenstrahldruck. Auf Grund der zunehmenden Verfügbarkeit zuverlässiger, kostengünstiger 3D-Drucker und leitfähiger Filamente gewinnt die Herstellung von leitfähigen, dreidimensionalen Strukturen zunehmend an Bedeutung. Die Möglichkeit mittels leitfähiger Materialien eine zusätzliche PCB-Fertigung einzusparen und leitfähige Pfade sowie elektronische oder sogar sensorische Komponenten direkt in ein Produkt einzubetten, sind nur einige Beispiele für die Motivation sich mit dieser Thematik zu beschäftigen.

Nach einem kurzen Überblick zum Stand der Technik wird im Rahmen des Papers auf die elektrischen Eigenschaften experimentell untersuchter leitfähiger Filamente eingegangen. Es wird aufgezeigt, welcher Einfluss der 3D-Druck auf die finalen Eigenschaften von 3D-gedruckten elektronischen und sensorischen Komponenten hat, welche Herausforderungen hierbei bestehen und welche generellen Designrichtlinien sich für den 3D-Druck elektronischer Systeme ableiten lassen. Zudem werden die Messergebnisse aus dem Druck beispielhafter Komponenten wie Widerständen, Kondensatoren, einem Füllstandssensor und einem kleinen kompletten Produkt vorgestellt.

Jannik Söll, jsoell@stud.hs-offenburg.de, Patrick Hog, patrick.hog@hs-offenburg.de, Elke Mackensen, elke.mackensen@hs-offenburg.de. Hochschule für Technik, Wirtschaft und Medien Offenburg, Badstraße 24, 77652 Offenburg.

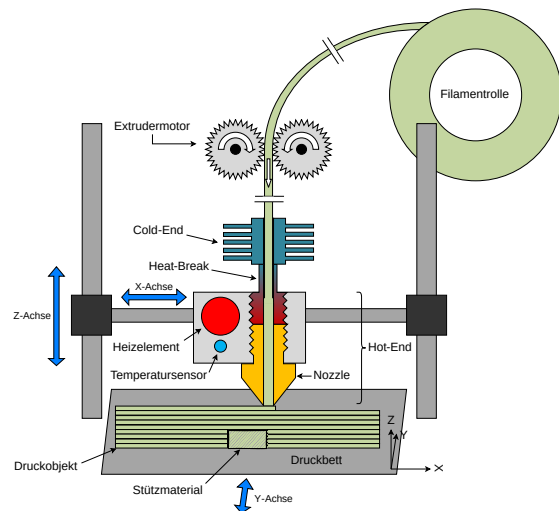


Abbildung 1. Aufbau eines 3D-Druckers in Gantry-Bauweise.

II. STAND DER TECHNIK

Beim 3D-Druck handelt es sich um ein additives Fertigungsverfahren, bei dem Material, z.B. Kunststoff, Metall oder Keramik, mittels eines 3D-Druckers Schicht für Schicht aufgetragen wird, um ein dreidimensionales Objekt zu erzeugen. Ausgangsbasis ist ein digitales 3D-Modell des Objektes, das gefertigt werden soll. Das digitale 3D-Modell wird üblicherweise durch eine spezielle Software in eine Vielzahl von dünnen Schichten zerlegt und dann als G-Code-Format exportiert. Dieser Code dient zur Ansteuerung des 3D-Druckers. Jede Schicht entspricht der exakten 2D-Form eines Abschnitts oder einer Scheibe des Objekts [1].

Es gibt eine Vielzahl an 3D-Drucktechnologien. Die bekanntesten Verfahren in diesem Bereich sind die Filament-Extrusion (engl. „Fused Filament Fabrication“ bzw. „Fused Deposition Modeling“, kurz FFF bzw. FDM) und die Vat-Photopolymerisation (kurz VPP, engl. „vat“ = Tank) mit den Unterverfahren Stereolithografie (SLA), „Digital Light Processing“ (DLP) sowie „Masked Stereolithography“ (MSLA).

Für die Untersuchungen, deren Ergebnisse im Rahmen dieses Beitrags vorgestellt werden, wurden FDM-Drucker verwendet, um 3D-gedruckte elektronische und sensorische Komponenten zu fertigen. FDM-

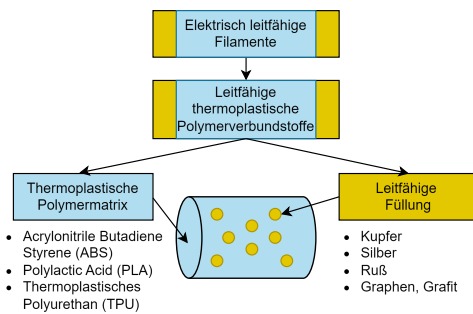


Abbildung 2. Schematischer Aufbau leitfähiger Filamente.

Drucker ermöglichen kostengünstige und schnelle Prototypenfertigung. Abbildung 1 zeigt den Aufbau eines FDM-Druckers, hier in Gantry-Bauweise [2]. Das FDM-Verfahren verwendet zur additiven Fertigung ein auf Rollen aufgewickeltes Kunststoff-Filament. Das Filament wird in einem Extruder aufgeschmolzen und durch eine Düse mit definiertem Durchmesser (= Nozzle) gepresst bzw. extrudiert. Das aufgeschmolzene Material wird dann durch Bewegung des Druckkopfes schichtweise auf dem Druckbett aufgetragen. Der Extruder selbst besteht aus mehreren Teilen. Das Filament wird vom Extrudermotor in Richtung Nozzle bewegt. Die restlichen Teile auf dem Weg des Filaments bis zum Verlassen der Nozzle werden als Cold-End, Heatbreak und Hot-End bezeichnet. Beim Cold-End handelt es sich um einen Kühlkörper, der die vom Hot-End und der Düse aufsteigende Wärme ableitet, um ein zu frühes Erweichen des Filaments zu verhindern. Die Heatbreak erhöht durch einen geringen Querschnitt den Wärmewiderstand und begünstigt so zum einen die Kühlung durch das Cold-End, und zum anderen das Aufheizen und Halten der Temperatur des Hot-End. Das Hot-End besteht aus dem Heizblock mit Heizelement und Temperatursensor sowie der Druckdüse. Innerhalb des Hot-End wird das Filament aufgeschmolzen und durch das vom Extrudermotor nachgeführte Filament aus der Düse gepresst. Die Druckdüsen gibt es in verschiedenen Größen, Öffnungsgrößen und Materialien. Weit verbreitet sind Messingdüsen mit einem Öffnungsdurchmesser von 0,4 mm [2].

Die Herstellung von leitfähigen 3D-Strukturen wird durch leitfähige Filamente ermöglicht. Elektrisch leitfähige Filamente bestehen, wie in Abbildung 2 dargestellt, aus einem leitfähigen thermoplastischen Polymerverbundstoff. Dieser besteht aus einer thermoplastischen Polymermatrix als Basismaterial und einer leitfähigen Füllung, die in dem Polymermatrixmaterial eingebettet ist. Das Polymermatrixmaterial wird verwendet, um die leitfähige Füllung zu umhüllen und eine stabile Form zu erhalten. Bei den in Abbildung 2 angegebenen Stoffen handelt es sich um Polymermaterialien, die üblicherweise für den FDM-Druck verwendet werden. Die leitfähige Füllung kann aus

Materialien wie Kupfer, Silber oder auf Kohlenstoff basierenden Materialien wie Ruß, Graphen, Grafit, Kohlenstoffnanoröhren, Graphitfolie und anderen Kohlenstoffmaterialien bestehen und dient als Stromleiter im Filament. Die Verfügbarkeit verschiedener Verbundwerkstoffe erzeugt eine Vielzahl mechanischer, elektrischer und thermischer Eigenschaften [3, 4, 5].

Verschiedene Forschungsgruppen beschäftigen sich weltweit mit der Realisierung von elektronischen Komponenten basierend auf 3D-druckbaren leitfähigen thermoplastischen Filamenten. Jangid, Strong et al. haben den Druck von flexiblen Schaltkreisen auf Papier mittels leitfähiger Filamente untersucht. Dabei wurden jedoch nur die Leiterbahnen gedruckt [3]. Flowers et al. haben die Dual-Material-Fused-Filament-Fertigung für den 3D-Druck elektronischer Komponenten und Schaltungen genutzt. Dabei sind beispielsweise experimentell Leiter, Widerstände, Kondensatoren und Induktoren und andererseits auch komplexere Komponente wie Hochpassfilter und Empfängerspulen für die drahtlose Energieübertragung entstanden [6]. Kwok et al. haben neben einfachen eingebetteten Leiterbahnen unter anderem auch einen Temperatursensor realisiert [7]. Die in der Literatur vorzufindenden Ergebnisse haben alle gemeinsam, dass die Ergebnisse für die Realisierung elektronischer Komponenten weitestgehend experimentell erzielt wurden, aber nicht auf beschriebenen Designrules basieren. Im Rahmen dieses Papers werden anhand der Untersuchungsergebnisse elektrisch leitfähiger Filamente erste Designrichtlinien formuliert (siehe Abschnitt III). Aufbauend auf den ermittelten Eigenschaften und den formulierten Gestaltungsrichtlinien wurden gezielt beispielhafte elektronische und sensorische Komponenten entwickelt, die in Abschnitt IV vorgestellt werden.

III. ERGEBNISSE AUS DER UNTERSUCHUNG ELEKTRISCHER EIGENSCHAFTEN VON LEITFÄHIGEN FILAMENTEN

Es wurden unterschiedliche leitfähige Filamente untersucht. Hierbei handelt es sich um.

- Proto-Pasta (= PP) [8]
- AIMPLA FILI (= AF) [9]
- Multi3D Electrifi (= EL) [10]

Als FDM-Drucker kamen der Ultimaker S5 und der Prusa i3 MK3S für die Fertigung der Untersuchungsobjekte sowie der elektronischen/sensorischen Komponenten zum Einsatz. Vorgestellt werden die Ergebnisse, die mit dem Proto-Pasta-Filament erzielt wurden.

Die generelle Vorgehensweise bei den Untersuchungen war wie folgt: Zunächst wurde ein CAD-Modell für 3D-Proben aus dem leitenden Filament entworfen. Danach wurden die Proben gedruckt. Vor der Messung wurde die Oberfläche der Probe noch mit Leitsilber veredelt, um einen möglichst geringen Übergangswiderstands zwischen Mess-Spitze und Oberfläche der

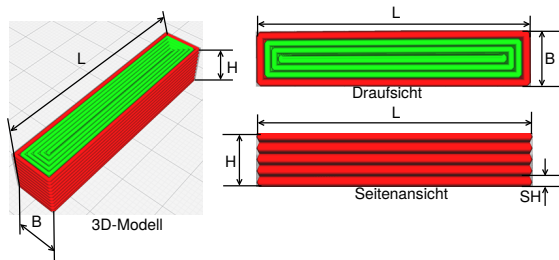


Abbildung 3. CAD-Modell der Messprobe in 3D-Ansicht, Draufsicht und Seitenansicht.

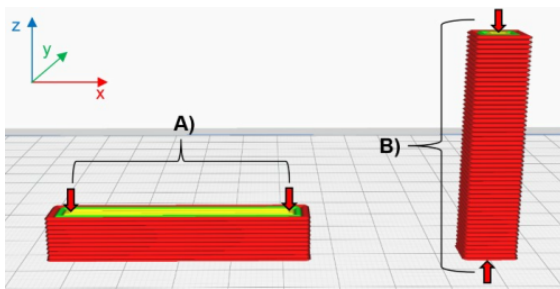


Abbildung 4. Messung des Widerstandes in X-Richtung (A) und in Z-Richtung (B).

Probe zu erhalten. Danach wurden die Messungen durchgeführt. Je nach Messung wurden die Proben variiert, z.B. in der Länge, Breite und Höhe.

A. Untersuchung des resistiven Verhaltens

In diesem Abschnitt werden Messergebnisse vorgestellt, die mit dem Filament Proto-Pasta (= PP) aufgenommen wurden. Die Messergebnisse können aber auch auf die anderen untersuchten Filament-Typen übertragen werden. Generell ist anzunehmen, dass sich der Ohmsche Widerstand R eines gedruckten Objektes nach Formel (1) verhält:

$$R = \rho \cdot \frac{L}{B \cdot H} \quad (1)$$

mit ρ = Spezifischer Widerstand des Materials
 L = Länge des Objektes
 B = Breite des Objektes
 H = Höhe des Objektes

Zunächst wurde der im Datenblatt angegebene spezifische Widerstand des Filaments messtechnisch überprüft. Das CAD-Modell der Material-Probe (ohne die Messanschlüsse), mit der die Messung durchgeführt wurde, ist in Abbildung 3 dargestellt. In der Seitenansicht sind deutlich die übereinander gedruckten Schichten zu erkennen, die eine Höhe SH aufweisen. In der Draufsicht ist zu erkennen, dass die einzelnen Schichten in Bahnen verlaufen, die abhängig vom Durchmesser der Druckdüse sind.

Dem Datenblatt des PP-Filaments kann entnommen werden, dass der spezifische Widerstand des Materials



Abbildung 5. Proben zur Messung des Widerstandes in X-Richtung.

abhängig von der Richtung des Stromverlaufs in der Probe (siehe Abbildung 4) ist. Im Fall A ist der Stromfluss in X-Richtung, im Fall B ist der Stromfluss in Z-Richtung. Den Datenblättern der anderen Filamente konnte hierzu keine Angabe entnommen werden, es ist jedoch anzunehmen, dass entsprechendes Verhalten vorzufinden ist, da eine Stromflussrichtung in Z-Richtung bedeutet, dass sich das Material beim Drucken zwischen den Schichten deutlich schlechter verbindet und inhomogener ist, als in X-Richtung.

Entsprechend Abbildung 4 wurden Proben gedruckt, um den Widerstand zu messen und damit den spezifischen Widerstand des Materials zu berechnen. Abbildung 5 zeigt die Proben zur Messung des Widerstandes R der Proben und zur Ermittlung des sich daraus ergebenden spezifischen Widerstand ρ . Abbildung 6 zeigt die Messung des Widerstandes für das Material PP bei einer Messung des Widerstandes in X-Richtung (siehe Abbildung 4A). Im Vergleich dazu ist der theoretische Widerstandsverlauf auf Grund der Datenblattangaben des Herstellers dargestellt, der den spezifischen Widerstand bei Messung in X-Richtung mit $\rho_x = 30 \Omega \text{ cm}$ angibt. Aus der Messung ergibt sich ein tatsächlicher spezifischer Widerstand von $\rho_{\text{ist},x} = 11,39 \Omega \text{ cm}$. Ähnlich Resultate wurden bei der Ermittlung des spezifischen Widerstandes durch Messung des PP-Materials in Z-Richtung (siehe Abbildung 4B) erzielt. Laut Datenblattangaben des Herstellers gibt dieser den spezifischen Widerstand bei Messung in Z-Richtung mit $\rho_z = 115 \Omega \text{ cm}$ an. Aus der Messung ergibt sich ein tatsächlicher spezifischer Widerstand von $\rho_{\text{ist},z} = 19,56 \Omega \text{ cm}$.

Bei beiden Messungen zeigte sich, dass der tatsächliche spezifische Widerstand ρ_{ist} stark von den Herstellerangaben abweicht. Beide Ergebnisse legen nahe, dass der Druckprozess einen starken Einfluss auf die resultierenden elektrischen Eigenschaften des Endprodukts hat. Aus diesem Grund wurde in einem nächsten Schritt jeweils die Längen, Breiten und Höhen der Proben geändert und die gleiche Probe auf zwei unterschiedlichen Druckern gedruckt (einem Prusa und einem Ultimaker). Bei den Messungen

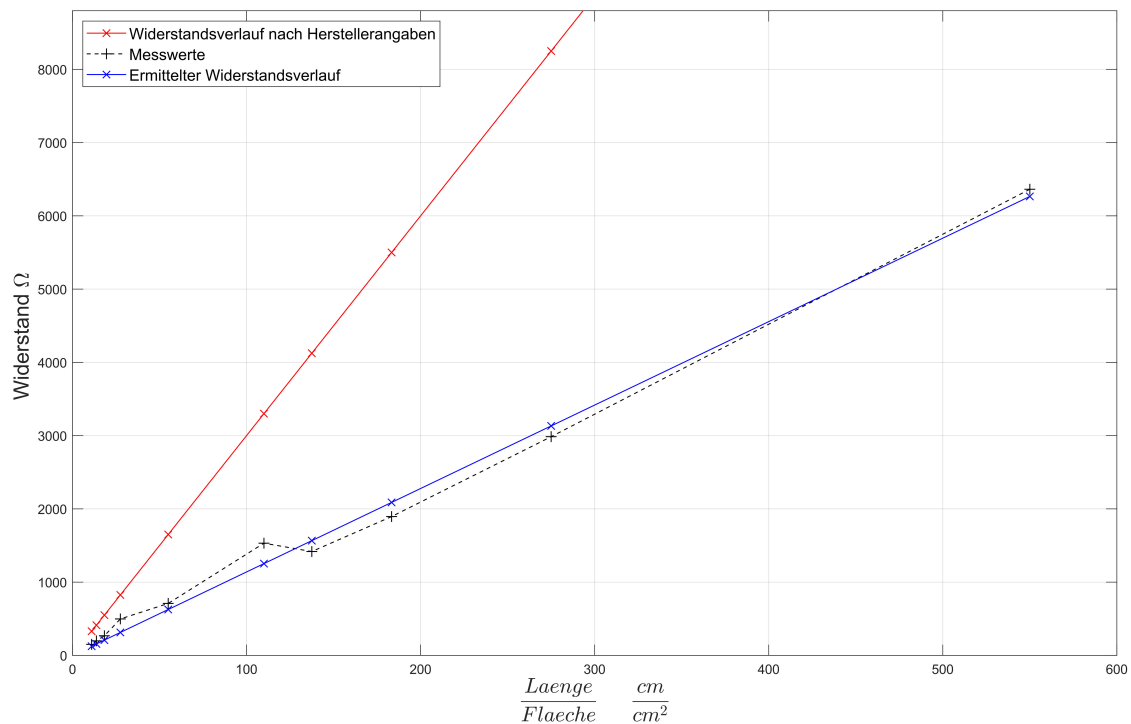


Abbildung 6. Messung des Widerstandes von PP-Proben in X-Richtung in Abhängigkeit der Länge pro Fläche.

Tabelle I
ERMITTELTES SPEZIFISCHES WIDERSTAND BEI MESSUNG DES
WIDERSTANDS DER PP-PROBEN IN X-RICHTUNG UND
VARIATION DER LÄNGEN, BREITEN UND HÖHEN DER PROBEN
UND DRUCK DER PROBEN AUF UNTERSCHIEDLICHEN
FDM-DRUCKERN

Varierte Größe	Drucker	$\rho_{\text{ist},x}$
Probenlänge	Prusa	12,80 Ω cm
Probenbreite	Prusa	12,76 Ω cm
Probenhöhe	Prusa	12,18 Ω cm
Probenlänge	Ultimaker	9,18 Ω cm
Probenbreite	Ultimaker	9,73 Ω cm
Probenhöhe	Ultimaker	9,18 Ω cm

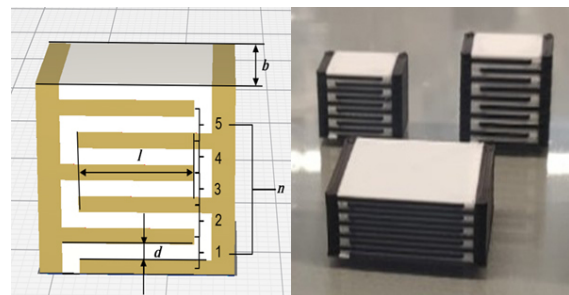


Abbildung 7. 3D-gedruckte Plattenkondensatoren.

zeigte sich, dass der Widerstand sich entsprechend der obigen Gleichung (1) verhält. Auch bei dieser Messung konnte festgestellt werden, dass der ermittelte spezifische Widerstand $\rho_{\text{ist},x}$ (siehe Tabelle I) von den Herstellerangaben abweicht, aber je nach verwendetem Drucker in der gleichen Region liegt (Ultimaker im Bereich 9 Ω bis 10 Ω , Prusa im Bereich von 12 Ω bis 13 Ω).

B. Untersuchung des kapazitiven Verhaltens

Zur Untersuchung des kapazitiven Verhaltens wurden Plattenkondensatoren entsprechend Abbildung 7 gedruckt, deren Kapazitäten sich nach Formel (2) berechnen lassen:

$$C = n \cdot \epsilon_r \cdot \epsilon_0 \cdot \frac{A}{d} \quad (2)$$

mit

- C = Kapazität des Kondensators
- n = Anzahl der Plattenpaare des Kondensators
- d = Abstand zwischen Platten des Kondensators
- A = Überlappungsfläche der Kondensatorplatten
- ϵ_0 = Dielektrizitätskonstante des Vakuums
- ϵ_r = relative dielektrische Permittivität des Dielektrikums: 2,4 - 2,8 bei einem Filament aus Polylactic Acid (= PLA)

Die Messergebnisse sind in Tabelle II dargestellt. Es zeigte sich ähnlich wie bei der Untersuchung des resistiven Verhaltens, dass sich die Kapazität gemäß Gleichung (2) verhält, auch dann wenn der Abstand der Platten und die Plattenfläche variiert wird. Allerdings weichen auch hier die tatsächlichen Messwerte immer von den erwarteten Kapazitätswerten ab.

Tabelle II
MESSERGEBNISSE AUS DEM DRUCK VON
PLATTENKONDENSATOREN BEI VARIATION DER ANZAHL DER
PLATTENPAARE

Proben Nr.	Platten- paare n	Erwartete Kapazitätswerte	Gemessene Kapazitätswerte
1	1	2,36 pF	6,84 pF
2	2	7,07 pF	15,68 pF
3	3	11,78 pF	25,78 pF
2	4	16,49 pF	34,06 pF
5	5	21,20 pF	44,32 pF

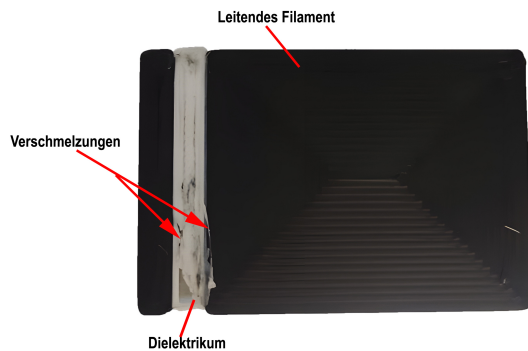


Abbildung 8. Verschmelzung der Filamente am Rande des Dielektrikums bei gedruckten Kondensatoren.

Auch bei diesen Messungen liegt es nahe, dass der Druckprozess einen Einfluss auf die elektrischen Eigenschaften einer Kapazität hat. Die Messungen an einem gedruckten Kondensator zeigten, dass der Plattenabstand d und die Fläche A der Platten den im CAD-Modell vorgegebenen Abmessungen entsprachen. Jedoch zeigte sich, dass bei der Erzeugung der Proben sich während des Druckprozesses die leitfähigen und isolierenden Schichten zu einem gewissen Grad verschmelzen (siehe Abbildung 8). Diese Verschmelzung beeinflusst die dielektrischen Eigenschaften des verwendeten isolierenden PLA-Materials.

C. Einfluss der Druckparametern auf die Eigenschaften elektrischer Komponenten

Am Beispiel des resistiven Verhaltens wurde detailliert untersucht, welchen Einfluss die folgenden Druckparameter auf die Eigenschaften elektrischer Komponenten haben:

- Druckdüsendurchmesser (nozzle size) des 3D-Druckers
- Höhe SH der gedruckten Schichten
- Drucktemperatur

Es wurden hierzu wieder Proben gemäß Abbildung 4 gedruckt. Dabei wurden jeweils die oben genannten Parameter geändert. Von diesen Proben wurde jeweils der Widerstand gemessen und dann wiederum der sich ergebende spezifische Widerstand errechnet. Zuvor wurden folgende Hypothesen aufgestellt:

Tabelle III
ERMITTELTEN SPECIFISCHER WIDERSTAND BEI MESSUNG DES
WIDERSTANDS VON PP-PROBEN IN X-RICHTUNG UNTER
VARIATION JEWEILS EINES DRUCKPARAMETERS

Druckparameters	Spezifischer Widerstand ρ
Nozzle Durchmesser: 0,4 mm	11,94 Ω cm
Nozzle Durchmesser: 0,6 mm	11,12 Ω cm
SH: 0,15 mm	12,65 Ω cm
SH: 0,30 mm	10,37 Ω cm
Drucktemperatur: 190 °C	14,01 Ω cm
Drucktemperatur: 230 °C	10,41 Ω cm

H1: Je größer der Druckdüsendurchmesser, desto geringer der spezifische Widerstand ρ . Grund dafür ist, dass für das Drucken einer Schicht weniger Bahnen gedruckt werden müssen (siehe Abbildung 3), also homogenere Schichten mit einer geringeren Anzahl an Grenzbereichen entstehen.

H2: Je höher die Schichtstärke SH beim Druckprozess, desto geringer der ermittelte spezifische Widerstand ρ . Hierdurch werden insgesamt weniger Schichten für das Objekt benötigt, was zu weniger Grenzbereichen im Objekt führt.

H3: Je höher die Drucktemperatur desto geringer der spezifische Widerstand ρ . Dies liegt daran, dass bei höherer Temperatur die einzelnen Bahnen und Schichten besser miteinander verschmelzen und dadurch das Material weniger Grenzbereiche aufweist, also auch hier homogener wird.

In Tabelle III sind die entsprechenden Messergebnisse aufgeführt. Alle oben aufgeführten Hypothesen konnten messtechnisch bestätigt werden. Um also kleine Widerstandswerte zu erzielen, sollte das Bauteil aus möglichst wenigen Bahnen und Schichten bestehen und bei einer hohen Temperatur gedruckt werden.

Auch für die gedruckten Kondensatoren wurde die Drucktemperatur variiert. Hier zeigte sich, dass die gemessene relative dielektrische Permittivität des Dielektrikums bei der Fertigung mit niedrigeren Drucktemperaturen geringe Abweichungen gegenüber der bekannten relativen dielektrischen Permittivität des PLA-Filaments aufzeigte, während hohe Drucktemperaturen zu großen Abweichungen führte.

IV. DESIGN UND REALISIERUNG BEISPIELHAFTER ELEKTRONISCHER UND SENSORISCHER KOMPONENTEN

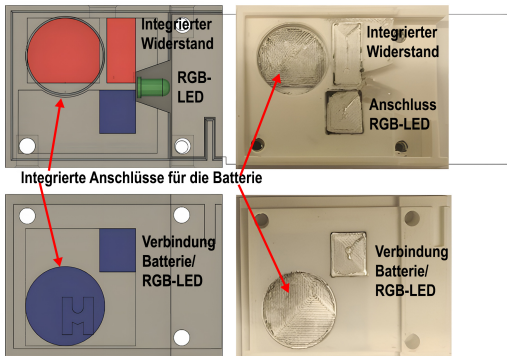
A. Realisierung von Widerständen

Mit den zuvor gewonnenen Erkenntnissen wurden verschiedene elektrische Widerstände mit vorgegebenen Werten aus Proto-Pasta-Filament ausgelegt und hergestellt. Die Druckparameter waren wie folgt:

- Schichthöhe: 0,2 mm
- Druckdüsendurchmesser: 0,6 mm
- Drucktemperatur: 219 °C



A) Alte Variante Namensschild mit Leiterkarte, Batteriehalter, Widerstand...



B) CAD-Modell und gedrucktes 3D-Objekt



C) Neue Variante Namensschild mit integrierten gedruckten Leitungen, Widerständen...

Abbildung 9. Realisierung eines beleuchteten Namensschildes im 3D-Druckverfahren mit integrierten leitenden Komponenten.

Der zuvor ermittelte tatsächliche spezifische Widerstand des Materials ist $\rho = 11 \Omega \text{ cm}$. Die Widerstände konnten erfolgreich realisiert werden. Es ergab sich eine Toleranz von ca. $\pm 15\%$ in der Genauigkeit der Auslegung der Widerstandswerte, und eine Toleranz von ca. $\pm 8\%$ in der Reproduzierbarkeit der Widerstandswerte.

B. Realisierung von Kondensatoren

Mit den zuvor gewonnenen Erkenntnissen wurden verschiedene Kondensatoren mit vorgegebenen Werten aus Proto-Pasta-Filament für die Kondensatorplatten und PLA-Filament für das Dielektrikum entworfen und hergestellt. Die Druckparameter waren wie folgt:

- Schichthöhe: 0,2 mm
- Druckdüsendurchmesser: 0,6 mm
- Drucktemperatur: 203 °C

Die zuvor ermittelte tatsächliche relative dielektrische Permittivität des Dielektrikums ist $\epsilon_r = 3,1$. Die Kondensatoren konnten erfolgreich realisiert werden. Dabei ergab sich eine Toleranz von ca. $\pm 12\%$ in der Genauigkeit der Auslegung der Kapazitätswerte.

C. Realisierung einer einfachen elektrischen Baugruppe

Anhand einer einfachen elektrischen Baugruppe sollte der mögliche Mehrwert aufgezeigt werden, der durch den 3D-Druck elektronischer Systeme entstehen kann. In diesem Fall handelt es sich um ein beleuchtetes Namensschild (siehe Abbildung 9), das in seiner alten Version (Abbildung 9A) aus einer Leiterkarte, einer RGB-LED, einer Batterie mit Batteriehalter, einem Vorwiderstand, einer Zenerdiode und einem 3D-gedruckten Gehäuse sowie einer gelaserten Plexiglasplatte bestanden hat. Ziel war es nun, mit dem 3D-Design die ursprünglichen Abmessungen des Namensschildes nicht zu überschreiten, die Gehäuseform also beizubehalten und final bis auf die RGB-LED und eine Batterie keine weiteren elektronischen Komponenten für das Namensschild zu benötigen.

Abbildung 9B links zeigt das CAD-Modell des neuen Gehäuses. Abbildung 9B rechts zeigt das gedruckte 3D-Gehäuse, gedruckt aus leitendem PP-Filament sowie PLA-Filament. Es ist zu erkennen, dass die Batteriehalterung direkt in das Gehäuse integriert wurde. Ebenso die Leiterbahnen und Anschlüsse, an welche die RGB-LED angelötet werden kann. Bei dem Vorwiderstand handelte es sich bisher um einen 33Ω -Widerstand. Aufgrund des hohen spezifischen Widerstand der Filamente kann der Widerstand als einfache leitende Fläche im Gehäuse umgesetzt werden. Die Auslegung des integrierten Widerstandes wurde so vorgenommen, dass eine Zener-Diode nicht mehr notwendig war.

Abbildung 9C zeigt die neue Variante des beleuchteten Namensschildes, bei dem, gegenüber dem alten Design, die Komponenten Leiterkarte, Batteriehalter, Vorwiderstand und Zenerdiode eingespart werden konnten.

D. Realisierung eines kapazitiven Füllstandssensors

Mit den gewonnenen Erkenntnissen wurden auch verschiedene sensorische Komponenten realisiert. Unter anderem ein Füllstandssensor, dessen Funktionsprinzip auf der Änderung der Kapazität eines Kondensators beruht. Wie in Abbildung 10 dargestellt, bilden eine isolierte Elektrode in Form einer Sonde und die Behälterwand, die als zweite Elektrode fungiert, einen Kondensator. Das Medium im Behälter fungiert als Dielektrikum. Die Veränderung der Füllstandshöhe verursacht eine Kapazitätsänderung. Ein leerer Behälter hat eine niedrige, ein gefüllter Behälter eine hohe Kapazität. Abbildung 11 stellt den realisierten Füllstandssensor dar, dessen Design typisch für eine kapazitive Füllstandsmessung in zylindrischen Messbehältern ist [11].

Abbildung 12 stellt die Messergebnisse dar, die mit dem Füllstandssensor und einer leitenden Flüssigkeit als Medium erzielt wurden. Im Vergleich dazu sind die theoretisch zu erwartenden (= berechneten) Messwerte dargestellt. Es ist zu erkennen, dass die tatsächlichen

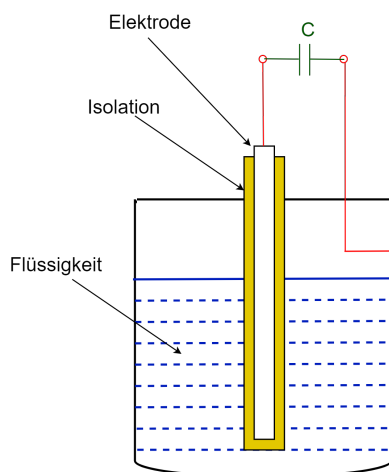


Abbildung 10. Messprinzip der kapazitiven Füllstandsmessung.



Abbildung 11. Realisierter Füllstandssensor im 3D-Druck. Die Messsonde entspricht Elektrode 1. Elektrode 2 ist in den zylindrischen Messbehälter integriert.

Messwerte von den theoretisch zu erwartenden Werten abweichen. Dies kann auch wieder auf den Druckprozess zurückzuführen sein. Prinzipiell ist aber zu erkennen, dass damit tatsächlich Füllstände detektiert werden können.

V. ZUSAMMENFASSUNG UND AUSBLICK

Im Rahmen dieses Papers wurde auf die Realisierung von elektronischen und sensorischen Komponenten mittels FDM-3D-Druck-Verfahren unter Verwendung von leitenden Filamenten eingegangen. Es hat sich gezeigt, dass sich elektronische und sensorische Komponenten mittels FDM-3D-Druck-Verfahren realisieren lassen. Die Untersuchungen des resistiven und kapazitiven Verhaltens leitfähiger Filamente haben gezeigt, dass die prinzipiellen formelmäßigen Gesetzmäßigkeiten für Widerstand und Kapazität gültig sind, um elektronische und sensorische Komponenten zu entwerfen. Ein weiteres Ergebnis ist, dass unter-

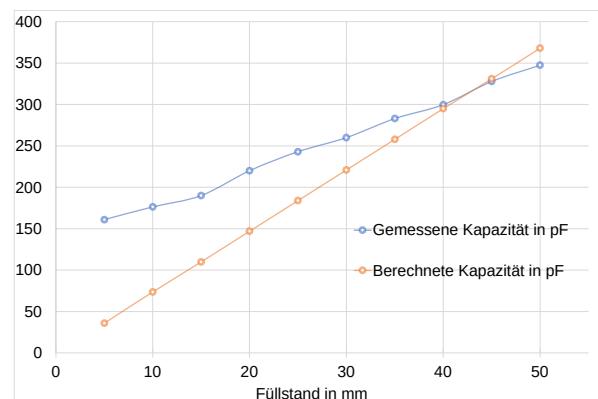


Abbildung 12. Messergebnisse einer Füllstandsmessung mit dem 3D-gedruckten Füllstandssensor.

schiedliche leitfähige Filamente am Markt verfügbar sind, jedoch nicht alle gleich gut für die Realisierung von elektronischen Komponenten geeignet sind. Es hat sich zudem gezeigt, dass der Druckprozess einen erheblichen Einfluss auf die finalen elektrischen Eigenschaften eines gedruckten Produkts hat. Generelle Designrichtlinien können daher nicht allgemein für unterschiedliche Drucker und unterschiedliche Filamente formuliert werden, sondern beziehen sich immer auf einen bestimmten Druckertyp und ein bestimmtes Filament. Die präsentierten Ergebnisse bestätigen das Potential der untersuchten Materialien und geben einen ersten Ausblick auf die Fähigkeiten und Möglichkeiten, die sich für die gedruckte Elektronik anhand der leitfähigen Filamente eröffnen. Ziel weiterer Arbeiten wird sein, noch weitere leitende Filamente unter Verwendung anderer 3D-Drucker zu untersuchen und dann für einen bestimmten 3D-Drucker Designrules für verschiedene leitfähige Filamente zu formulieren.

LITERATURVERZEICHNIS

- [1] Dassault Systems. *Was ist 3D-Druck?* 2025. URL: <https://www.3ds.com/de/make/guide/process/3d-printing> (besucht am 20.02.2025).
- [2] P. Hog. *Einführung in den 3D-Druck: Einstieg in den 3D-Druck im Edu FabLab der Hochschule Offenburg*. Vorlesungsscript. 2022.
- [3] Aditya R. Jangid, E. Brandon Strong, Jacqueline Chuang, Andres W. Martinez und Nathaniel W. Martinez. „Evaluation of commercially-available conductive filaments for 3D printing flexible circuits on paper“. In: *PeerJ Materials Science* 4 (). DOI: 10.7717/peerj-matsci.21. URL: <https://par.nsf.gov/biblio/10343141>.
- [4] Ziyad Aloqalaa. „Electrically Conductive Fused Deposition Modeling Filaments: Current Status and Medical Applications“. In: *Crystals* 12.8 (2022). ISSN: 2073-4352. DOI: 10.3390/cryst12081055. URL: <https://www.mdpi.com/2073-4352/12/8/1055>.
- [5] Dejana Pejak Simunec und Antonella Sola. „Emerging Research in Conductive Materials for Fused Filament Fabrication: A Critical Review“. In: *Advanced Engineering Materials* 24.7 (2022), S. 2101476. DOI: <https://doi.org/10.1002/adem.202101476>. URL: <https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/adem.202101476>.
- [6] Patrick F. Flowers, Christopher Reyes, Shengrong Ye, Myung Jun Kim und Benjamin J. Wiley. „3D printing electronic components and circuits with conductive thermoplastic filament“. In: *Additive Manufacturing* 18 (2017), S. 156–163. ISSN: 2214-8604. DOI: <https://doi.org/10.1016/j.addma.2017.10.002>. URL: <https://www.sciencedirect.com/science/article/pii/S2214860417301860>.
- [7] Sen Wai Kwok u.a. „Electrically conductive filament for 3D-printed circuits and sensors“. In: *Applied Materials Today* 9 (2017), S. 167–175. ISSN: 2352-9407. DOI: <https://doi.org/10.1016/j.apmt.2017.07.001>. URL: <https://www.sciencedirect.com/science/article/pii/S235294071730152X>.
- [8] Proto-Pasta. *Electrically Conductive Composite, Conductive 3D Printer PLA Filament*. URL: <https://www.protopasta.com/products/conductive-pla?variant=1265211476>. (besucht am 21.03.2023).
- [9] AIMPLAS - ASOCIACION DE INVESTIGACION DE MATERIALES PLASTICOS Y CONEXAS. *Flexible conductive filament Fili*. URL: <https://filament2print.com/gb/flexible-tpe-tpu/1726-flexible-conductive-filament-fili.html> (besucht am 21.03.2023).
- [10] Multi3D. *Electrifi Conductive Filament*. URL: <https://www.multi3d.com/product/electrifi/> (besucht am 21.03.2023).
- [11] S. Hesse und G. Schnell. *Sensoren für die Prozess- und Fabrikautomation: Funktion — Ausführung — Anwendung*. 6. Aufl. Wiesbaden: Springer Vieweg, 2014. ISBN: 9783658058661.



Jannik Söll absolvierte das Studium Mechatronik - Plus an der Hochschule Offenburg. Durch seine Mitarbeit im Edu FabLab der Hochschule Offenburg konnte er wertvolle praktische Erfahrungen sammeln und seine Kenntnisse in modernen Fertigungstechnologien vertiefen. Im Rahmen seiner gewählten Fachrichtung Mechatronik faszinierte ihn besonders die interdisziplinäre Verbindung aus fertigungstechnischen Prozessen wie der additiven

Fertigung, werkstofftechnischen Aspekten sowie elektrotechnischen Anwendungen, die seine Arbeit in diesem Bereich prägten. Derzeit befindet sich Jannik Söll im Referendariat zur Ausbildung als Lehrkraft an beruflichen Schulen, wo er sein technisches Fachwissen mit pädagogischen Kompetenzen verbindet, um zukünftige Fachkräfte praxisnah auszubilden.



Patrick Hog ist wissenschaftlicher Mitarbeiter an der Hochschule Offenburg und technischer Leiter des hochschuleigenen Makerspaces Edu FabLab. Nach einer Ausbildung und seinem Bachelor-Abschluss in Mechatronik absolvierte er ein Masterstudium in Bereich Elektro- und Informationstechnik an der Hochschule Offenburg. Sein Schwerpunkt liegt in additiven und digitalen Fertigungsverfahren, insbesondere im 3D-Druck. Seit 2019 leitet er den

hochschuleigenen Makerspace, den er maßgeblich aufgebaut hat. Dort unterstützt er Studierende bei der Umsetzung technischer Projekte und engagiert sich in der praxisnahen Ausbildung zukünftiger Ingenieurinnen und Ingenieure.



Elke Mackensen studierte Nachrichtentechnik an der Fachhochschule Offenburg. Nach dem Diplomabschluss in 1993 und dreijähriger Industrietätigkeit begann sie 1996 ihre wissenschaftliche Laufbahn an der Universität in Freiburg. Zunächst am Institut für Informatik wo sie ein ASIC-Design-Labor aufbaute, danach am Institut für Mikrosystemtechnik (IMTEK), wo sie 2006 über autarke drahtlose Mikrosysteme promovierte. Von 2006 bis 2010 leitete

sie bei der Firma NewTec die Entwicklungsgruppe Smart Embedded Systems. Seit 2010 ist sie Professorin an der Hochschule in Offenburg. Ihre Professur beschäftigt sich mit der digitalen Schaltungstechnik und dem mikroelektronischen Systementwurf. Forschungsschwerpunkte sind auf Energy-Harvesting basierte elektronische Systeme und gedruckte Elektronik/Sensorik mit Lowcost-Druckmethoden. Sie ist die wissenschaftliche Leiterin des hochschuleigenen Makerspaces Edu FabLab.

Low Cost Serial DAC Simulation, Realization, Error Correction and Characterization

Jörg Vollrath

Abstract—A serial C DAC has a minimum number of components. This structure makes it easy to simulate, realize, characterize and to study error correction. This paper presents a discrete 12 bit serial C DAC with digital error correction. Theory, high level (JavaScript), low level (LTSPICE) simulations, a real circuit and low cost measurements and error correction using an Arduino board are presented. Comparing theory, simulation and measurement gives deeper understanding of DAC leading to better circuits and improved characterization methods. The measurements show INL, DNL below 4 and FFT with 46.5 dB SNQR which gives 6.5 ENOB at 33 Hz sampling frequency of a 12 bit DAC. Hardware costs are below 100 Euro without an oscilloscope for detailed signal measurements.

Index Terms—serial DAC, circuit simulation, characterization, INL, DNL, SNR, error correction

I. INTRODUCTION

Data converters are very important elements connecting physical world to electrical world allowing digital signal processing. The quality of digital processing depends on the performance of the data converters. Therefore understanding and measurement of the quality of data converters allow to build high performance digital systems.

The resolution of data converters is limited by offset error, gain error, INL, DNL and SNR. The IEEE standards 1057 and 1241 are applicable for data converters. For high resolution data converters a lot of effort is required to calculate and measure these numbers. Unfortunately the relationship between the circuit configuration, faults and changes in these parameters is not easily determined. In literature there are only limited data connecting circuit faults and errors to patterns in INL, DNL and SNR. A direct link between INL and spectrum is mathematically difficult. There are limited examples showing improvements of digital error correction [1, 2, 3]. Therefore publicly available tools to study these effects are very interesting.

Test and characterization engineers can use these tools to be able to prepare very early in the design process a proper test and characterization method. Simulations can be directly transferred to tests. Additional tests can be simulated before hardware is available.

Jörg Vollrath, Joerg.vollrath@hs-kempten.de, University of Applied Science, Bahnhofstraße 61, 87435 Kempten, Germany.

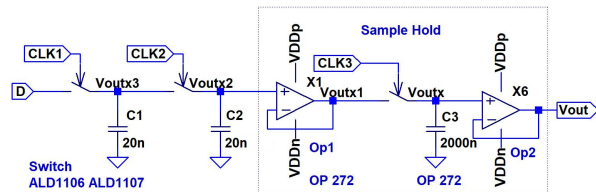


Figure 1. A serial DAC with sample and hold.

Test data analysis can be prepared using available simulation data. Direct comparison of simulated and measured data leads to verification of the device and the test environment.

Simulations are valuable tools to understand theory by implementing equations and exploring variations. For circuit simulation SPICE variants (LTSPICE, PSPICE, Multisim) are used. Due to the large number of bits low level circuit simulation of data converters is very slow and requires a lot of memory and computing power. High level simulations are used to estimate data converter performance and understand conversion limitations. For high level simulation tools like MatLab Simulink or a programming language like C are used. These tools are expensive and it is difficult to document simulation setup and results. This paper presents a low cost serial charge sharing DAC: schematic, high level simulation, practical realization, measurement and characterization.

A serial DAC is a simple, low component count easy to understand DAC [4]. Figure 1 shows a serial DAC (C1, C2, CLK1, CLK2) with sample and hold (X1, X5, C3, CLK3).

Data is serially latched with CLK1 in C1 (LSB first). Charge is shared between C1 and C2 with CLK2 activated. This cycle is repeated for each bit (serial). At the beginning CLK1 and CLK2 are active to put 0 V on both capacitors. The sample and hold circuit (preamplifier operational amplifier X1, switch CLK3 X5, operational amplifier X6) stores the output voltage. This circuit in principle could have infinite resolution for infinite number of charge sharing cycles. To calculate the final output voltage a charge computation is done [cf. equations (1) – (6)].

$$Q = C \cdot V \quad (1)$$

$$C2 \cdot Vx2(n) + C1 \cdot D(n) = Vx2(n+1) \cdot (C1 + C2) \quad (2)$$

$$Vx2(n+1) = (Vx2(n) \cdot C2 + C1 \cdot D(n)) / (C1 + C2) \quad (3)$$

with $C1 = C2$

$$Vx2(n+1) = 0.5 \cdot Vx2(n) + 0.5 \cdot D(n) \quad (4)$$

$$Vx2(n+1) = 0.25 \cdot Vx2(n-1) + 0.25 \cdot D(n-1) + 0.5 \cdot D(n) \quad (5)$$

$$Vout = \sum_{k=0}^{nBit-1} 2^{nBit+k} D(k) \quad (6)$$

The equations show the binary weight of the data inputs. This is a binary weighted DAC having a very low component count. Practical limits of this circuit are the precision of the capacitances $C1$ and $C2$, the leakage of the switches (back bias of transistors) and the leakage, offset and non linear gain of the operational amplifier. The precision of the matching of the capacitances have to be below 1 LSB. For a 16-bit DAC this would be $1/65536 = 152\text{ppm}$. Thermal noise of the R of the switches and operational amplifiers noise are also limiting the maximum resolution.

In the following sections first the LTSPICE simulation circuit with input signal generation for ramp and sine signals is presented. Open access internet pages are used to read and evaluate LTSPICE data [5]. The code for these pages is freely available at no cost. Due to the slow LTSPICE simulation and to model DAC errors properly an internet based JavaScript high level simulation is presented. All internet pages can also be run locally on a computer and source code can be modified to study other effects and architectures. Next the circuit is realized using an operational amplifier, NFET and PFET arrays and an Arduino board. Realization can prove the match between theory, simulation and real circuit and can highlight missing model assumptions. Then measurements of ramp and sine signals give INL, DNL, FFT, SNR and ENOB. Errors showed additional challenges of a real circuit and lead to error correction to improve ENOB. Finally results are discussed.

II. LTSPICE SIMULATION

For simulation with LTSPICE (no cost program) serial digital data for a ramp and sine signal have to be generated. An ideal model of a 4 bit pipeline ADC and a switch matrix is used to generate the appropriate input signals easily. Figure 2 shows the 8-bit serial DAC circuit. An ideal DAC is omitted on the picture, but implemented to have an ideal output voltage for comparison. The resolution is limited to 8-bit to get reasonable simulation times. Real transistor

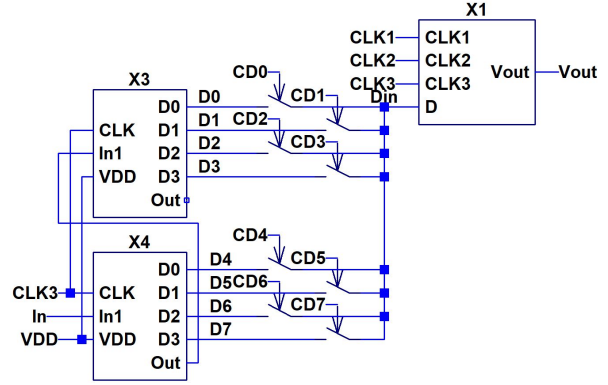


Figure 2. LTSPICE test circuit.

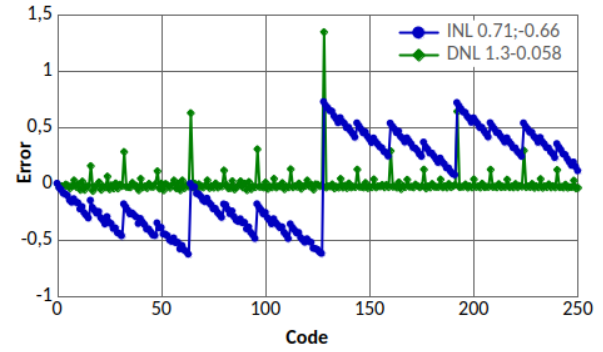


Figure 3. INL and DNL of LTSPICE ramp simulation.

models and operational amplifier models are used. Low frequency clock signals ($t_{CK} = 60\mu s$) are used since the practical circuit is also operated at low speed.

Care has to be taken to limit the raw output data file size with a save statement and to control precision with `.option plotwinsize=0`.

First an up/down ramp is used to have minimum settling time for accurate output values. The DAC output comes one cycle later than the input data. Care has to be taken that all codes are exercised and generate an output value. The circuit for download and simulation is available on a website [5].

A. Ramp simulation for INL and DNL

Figure 3 shows the INL, DNL of LTSPICE simulation (10 MB file size) after processing with JavaScript [5] tool in the browser.

LTSPICE writes data with varying step size. Output data has to be filtered with sampling times. From this raw ramp data minimum and maximum is extracted. Real step size LSB is calculated in (7).

$$LSB_{real} = (\text{Maximum} - \text{Minimum}) / NSteps \quad (7)$$

An ideal curve compensating for offset (Minimum) and gain error is generated (8):

$$OUT_{ideal}(i) = \text{Minimum} + i \cdot LSB_{real} \quad (8)$$

and INL and DNL are computed ((9) and (10)):

$$\text{INL}(i) = (\text{OUT}(i) - \text{OUT}_{\text{ideal}}(i)) / \text{LSB}_{\text{real}} \quad (9)$$

$$\text{DNL}(i) = (\text{OUT}(i) - \text{OUT}(i-1) - \text{LSB}_{\text{real}}) / \text{LSB}_{\text{real}} \quad (10)$$

All these steps are automatically done by JavaScript at client side in the browser. A graph is generated as displayed in Figure 3. Maximum DNL of 1.5 and INL minimum of -0.8 can be seen reducing the effective number of bits. The program works up to a few million data points.

The circuit has a big error step at MSB switching limiting the resolution of the DAC. The effective number of bits is reduced.

The pattern of INL and DNL is typical for binary weighted data converters [6]. The DNL has a maximum at the center and can be up to 2 times the INL maximum error. DNL peaks can be seen only at multiples of power of 2 bit switching operations. Higher order bits contribute more to DNL error. A mismatch between the capacitances C1 and C2 or added capacitance of the CMOS transistor switches can lead to this pattern.

B. Sine simulation for FFT and SNR

An 8-bit DAC needs at least $256 \cdot 4$ points for FFT due to changing slope of a sine signal and an odd or prime integer number of periods to exercise all codes.

Having an integer number prevents FFT bleeding and windowing can be omitted. The simulation time t_{meas} is calculated as given in (11).

$$t_{\text{meas}} = 1024 \cdot 480 \mu\text{s} = 491.52 \text{ ms} \quad (11)$$

Having $N_{\text{period}} = 13$ periods gives the signal frequency calculated in (12).

$$f_{\text{signal}} = N_{\text{period}} / t_{\text{meas}} = 26.44856771 \text{ Hz} \quad (12)$$

The raw output data is evaluated with the same web page as ramp data. Start time 0 s, stop time 491.7 ms and time step $480 \mu\text{s}$. Figure 4 shows the resulting FFT mapping the DC content to a frequency of 0.1 Hz. Numbers for signal and noise level are shown at the top of the graph. Total noise level is shown as a line. The number of FFT points can be easily extracted from the maximum frequency as $2 \cdot 512 = 1024$. To be able to assess an FFT the number of points determine the total noise, since all noise bins have to be summed up for total noise.

The signal can be seen at frequency 13 Hz with -3 dB magnitude, total noise is 50.14 dB and biggest harmonic HD3 at -55.67 dB below an expected total noise level calculated in (13).

$$-3 \text{ dB} - 1.76 \text{ dB} - 8 \cdot 6.02 \text{ dB} = -52.92 \text{ dB} \quad (13)$$

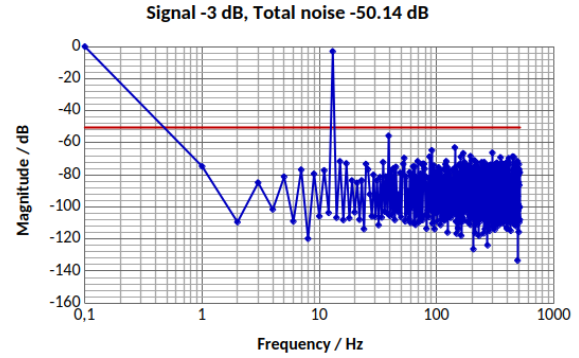


Figure 4. FFT of the simulated serial C DAC.

Table I
FFT MAGNITUDE RESULTS

Frequency Hz	Signal magnitude dB	Total noise magnitude dB
0	-0,04	-3,00
13	-3,00	-50,14
39	-55,67	-51,65
143	-63,56	-51,94
299	-65,98	-52,11
91	-66,28	-52,28

The total noise level can not be extracted easily from the FFT graph and needs some computation which is done by the web page in JavaScript at the client. Without the highest harmonic HD3 the total noise level is -51.65 dB. The -3 dB shows the rms value of the 1 V amplitude. The tool gives a table with the frequencies in order of signal magnitude (Table I).

This allows to look at the signal to noise level (47.29 dB) and the impact of harmonics. The effective number of bits (ENOB) can be calculated from Table I as given in (14) and (15).

$$\text{ENOB} = \frac{\text{signal lvl} - (-\text{total noise lvl}) - 1.76 \text{ dB}}{6.02 \text{ dB}} \quad (14)$$

$$\text{ENOB} = \frac{-3 \text{ dB} - (-50.29 \text{ dB}) - 1.76 \text{ dB}}{6.02 \text{ dB}} = 7.56 \quad (15)$$

The ideal sine signal with 3 V amplitude at the LT-SPIICE input with the ideal DAC, which was simulated as sanity check for comparison, shows no visible harmonic and gives the effective number of bits according to (16).

$$\text{ENOB} = \frac{0.51 \text{ dB} - (-49.1 \text{ dB}) - 1.76 \text{ dB}}{6.02 \text{ dB}} = 7.78 \quad (16)$$

The first highest harmonic for an ideal signal is at frequency 447 Hz with -65.33 dB. The simulation shows harmonics not limiting ENOB but the total

Table II
FFT MAGNITUDE OF BINARY DAC WITH SYSTEMATIC ERROR

Frequency Hz	Signal magnitude dB	Total noise magnitude dB
13	-9.04	-54.88
39	-59.57	-56.69
299	-68.10	-57.01
91	-69.44	-57.27

noise. LTSPICE ideal DAC accuracy is still not showing the full ENOB = 8.

Based on this LTSPICE model circuit improvements and characterization can be easily done at no cost. Unfortunately for larger number of bits the simulation time and data size becomes quite big. A 16 bit simulation stopped after 84 % of simulation after 10 h and 1.6 GB file size due to lack of virtual address space on a system with 16 GB main memory. A high level simulator is needed for faster analysis. A high level simulator shows also understanding of error sources by implementing the equations correctly.

III. JAVASCRIPT DAC SIMULATION

An Internet webpage [5] with a Javascript simulator was created to be able to study DAC errors faster (Figure 5). Unit or binary element DACs with different number of bits can be selected with random or systematic element error. Additionally a random noise measurement error can be added. Typical INL, DNL charts are generated. For FFT the number of points and number of periods of a test signal can be selected. Sine signal can have non integer and non prime number of periods. Effects on FFT with and without windowing can be studied. This can give a test engineer a feel for typical measurement errors and impact on results. Figure 5 shows the user interface and a result for an 8 bit binary element DAC with systematic error of -0.0065 . INL and DNL is provided for one random error run and absolute INL and DNL maxima for a user defined number of runs. The QR code allows to access the internet page directly. Also some error correction method is implemented as presented later.

The typical INL, DNL pattern of a binary element DAC can be seen again and the FFT showing a harmonic. The values were chosen to match the LTSPICE simulation. A table with FFT magnitude values is also generated (Table II).

The SNR of the simulation is $-9.04 \text{ dB} - (-54.88 \text{ dB}) = -45.84 \text{ dB}$ a little less than LTSPICE simulation. The first biggest visible harmonic (39 Hz) is still below total noise level. The next figures (Figure 6, Figure 7) show the results for a random error of 0.01 for unit element and binary element DACs. The INL graph shows also the maximum of INL and DNL of 16 different random runs giving a worst case scenario.

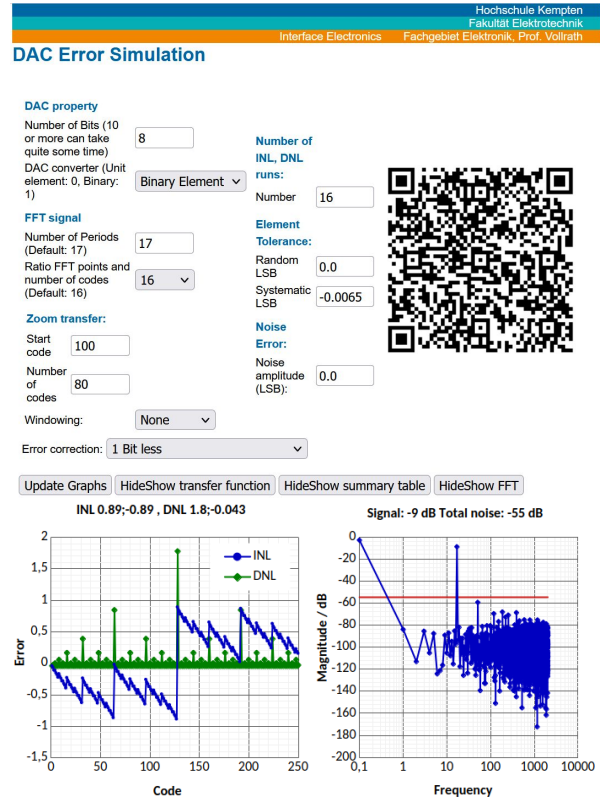


Figure 5. Javascript Internet Page with DAC Error Simulation.

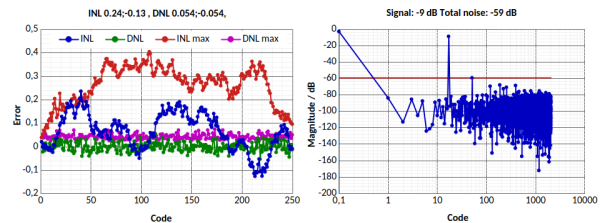


Figure 6. Random unit element DAC Error simulation.

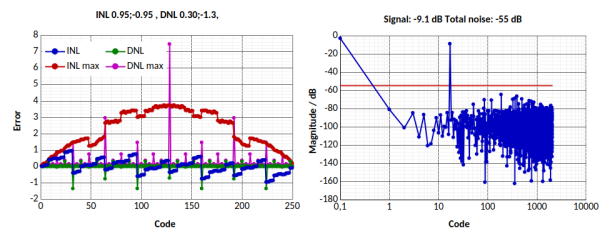


Figure 7. Random binary element DAC Error simulation.

The simulation results change each run time due to the randomness of errors. It is possible to see typical patterns due to the architecture. INL of the unit element DAC can have one or more bumps, DNL is much lower and more random.

DNL peaks are much bigger at the binary element DAC and INL shows a sawtooth pattern. FFT of the binary element DAC show a high harmonic due to the high DNL value. It is very easy to locate errors of the

Table III
SYSTEMATIC ERRORS, INL, DNL, SNR AND SDR

Type	Systematic Error	INL DNL max	Magnitude Harmonic dB	SNR w/o Harmonic dB
Unit	0.00001	0.08	HD2: -79	49.74
Unit	0.0001	0.8	HD2: -59	46.86
Binary	0.002	0.48	HD3: -76.24	49.50
Binary	0.01	2.2	HD3: -59.52	45.31

binary element DAC looking at the DNL and to correct them. Positive DNL determine the minimum step size, negative DNL causes code loss which can be corrected.

Only the binary element FFT shows visible harmonics. The bumps of the unit element DAC are too small to generate harmonics. It is interesting to vary the random error and look at many runs to observe the visibility of harmonics. The total noise will always be affected by INL, DNL error. The total noise of a binary element DAC is bigger than the unit element DAC with the same random error.

A systematic error can be applied to look at typical worst case INL, DNL and FFT patterns. Table III shows a summary.

Unit element DACs with systematic errors generate 2nd harmonics, binary element errors generate 3rd harmonics. Since there are more unit elements needed than binary elements smaller systematic errors can cause a harmonic.

IV. DAC CIRCUIT MEASUREMENT

To allow comparison between theory, simulations and measurement the circuit was built and measured. For cost reasons an Arduino board was used for control and data acquisition. It is very cheap, easy to program, has a 12 bit DAC and ADC for voltage acquisition and reference and enough digital pins to operate the switches and data input at 3.3 V. Measurement data is transferred via the serial interface to the PC. The sampling speed is very slow.

The serial DAC needs only a few components and is very simple. It was build with one opamp IC 272, one ALD1106 and one ALD1107 integrated circuit with 4 NFET and 4 PFET transistors each as switches and 3 10 pF capacitances. Beside the 3.3 V the operational amplifier needs extra voltages (5 V, 1.5 V) which were provided via battery or an Electronic Explorer power supply to amplify a signal in the 0 V to 3.3 V range. Figure 8 shows the final set up.

Data processing is done via another JavaScript tool. It is planned to have a local nodejs server controlling the Arduino via serial interface and providing a better web interface for the user. This would improve characterization capabilities. The Arduino program is also available on the web site.

The internal waveforms for circuit verification shown in Figure 9 were measured with an oscilloscope.

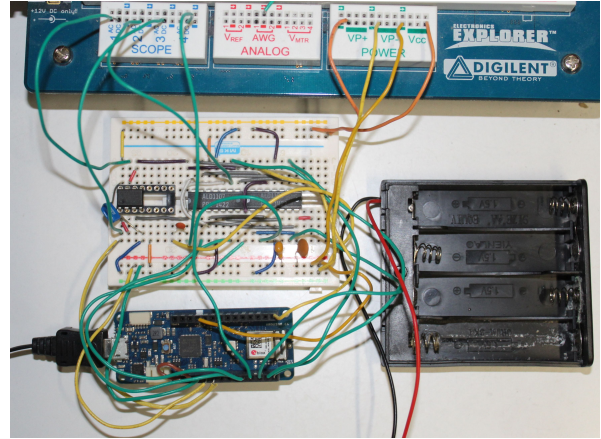


Figure 8. Serial DAC circuit realisation with Arduino and power supply.

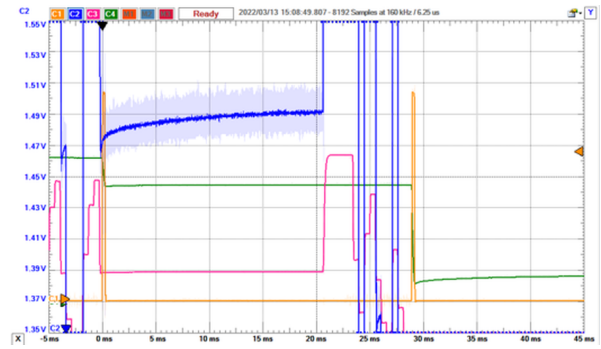


Figure 9. Internal signals of serial DAC.

C1 (yellow) shows the sampling every 29 ms. C3 (red) shows the internal node of the second capacitance C2 (blue). After a long initialization of 20 ms the data is transferred serially to the capacitance. C2 is a zoomed waveform of C3, which shows charging and discharging of this capacitance during the charge equalization times probably due to leakage. This is one reason for the steps. C4 (green) shows output voltage changing at the sampling with CLK3 and C1 signal.

The breadboard allows easy circuit modification and access to all signals. The Arduino environment provides easy change of CLK signal sequence, voltage levels and provides data transfer to a PC.

Since the basic functionality was confirmed, noise measurements were done and compared to averaging results (Table IV). Averaging of 16 values at code 2040 gave an average value of 2238 and shows a good compromise between accuracy and acquisition time. 64 would give a delta below 1, but was not used in this work due to time constraints. An automatic test sequence would allow application of higher averaging values. Theory states a gain of $10 \log(\text{OSR})$ in resolution for averaging. This can be seen in the reduction of standard deviation by a factor of 2 for an oversampling ratio of 4.

Table IV
AVERAGING OF SAMPLES AND DELTA FOR CODE 2040

	1	4	16	64	256
Min	2223.0	2234.0	2237.4	2238.0	2238.3
Max	2250.0	2241.8	2239.7	2238.9	2238.5
Delta	27.00	7.75	2.31	0.91	0.16
Npoints	1024	256	64	15	4
Standard Deviation	2.33	1.21	0.54	0.27	0.07

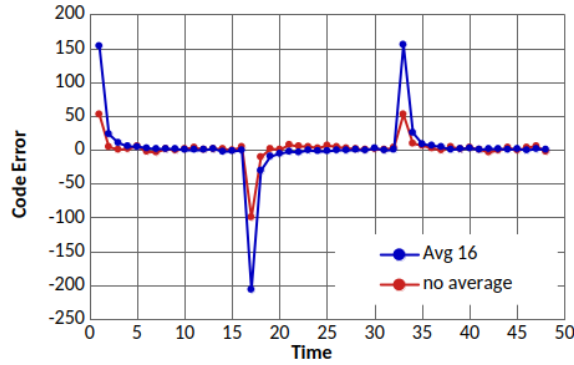


Figure 10. Code error for full scale settling time.

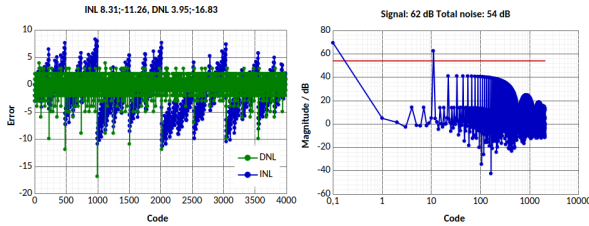


Figure 11. Raw measurement INL, DNL and SNR.

Next full scale and mid scale settling time was measured. For full scale settling time all bits are changing from 0 to 1 or vice versa. Figure 10 shows only the difference to final values of this measurement with and without averaging.

At least 2 samples are needed until the final value is reached. It is expected that large voltage changes will effect measured SNR of this set up. The circuit has some memory effect and should be improved. On the other hand this error can be measured and used in studying error correction.

Next ramp measurements were done and offset (20 codes) and gain errors observed. The output started with a code of 20 and reached 4095 for a input code of 4040. The full resolution of 12 bit could not be reached in this set up with these values since codes 20+55 are missing.

First ramp measurement for 12 bits gave INL between 8.31 and -11.26 and DNL between 3.95 and -16.83. FFT gave 62dB signal and 54dB total noise as seen in Figure 11.

Table V
FFT MAGNITUDE OF BINARY DAC WITH SYSTEMATIC ERROR

Min	Max	Signal Magnitude dB	Total Noise Magnitude dB	Cutoff
100	3899	62.56	-10.60	0
400	3500	61.72	40.17	300
200	3799	62.44	28.64	100
132	3868	62.54	17.13	32
116	3883	62.55	9.94	16
108	3891	62.56	2.79	8
104	3891	62.56	-3.77	4

Range matching is one important feature of DAC testing. Therefore it is interesting to see what happens if the sine signal is overshooting and top and bottom are cut off. Table V shows a typical evaluation. A cut off of 4 codes affects the signal to noise behaviour by 6 dB. During test only a slightly smaller range than minimum and maximum code should be tested.

V. ERROR CORRECTION

A calibration or error correction can be done based on the ramp INL, DNL data. Here 2 schemes are discussed. First only 31 code changes of multiples of 128 for the 5 MSB bits are corrected. Secondly a lookup table is constructed. Error correction always reduces the number of possible codes. Therefore the sine range and offset have to be adjusted.

Due to the pattern of the binary DAC transition voltages can be defined where an offset has to be added to the DAC code value. In this example of a 12-bit DAC 7 correction values were added for the 3 MSBs.

The correction values $C(i)$ can be extracted from the ramp measurement and added $C_{sum}(i)$. A comparator compares the input code with the transition voltages $V_t(i)$ and applies the correction values $C(i)$. For $V_t(i)$ the sum of the correction code has to be subtracted.

In this example the 3 MSB give the transition voltages:

$$V_t = 511, 1023, 1535, 2047, 2559, 3071, 3582$$

From the ramp measurement the following correction factors were taken:

$$C = 6, 10, 5, 12, 4, 8, 5$$

At each transition voltage the correction factors are added up:

$$C_{sum} = 6, 16, 21, 33, 37, 45, 50$$

Each transition voltage changes due to the $C_{sum}(i)$:

$$V_t = 511, 1017, 1519, 2014, 2522, 3026, 3532$$

In the control program the input code is compared with the transition voltages and correction $C_{sum}(i)$ is added. 3903 Codes remain, DNL and INL are improved (Figure 12).

Secondly lookup calibration can be done. A lookup table is created using sorted pivot operation on ramp

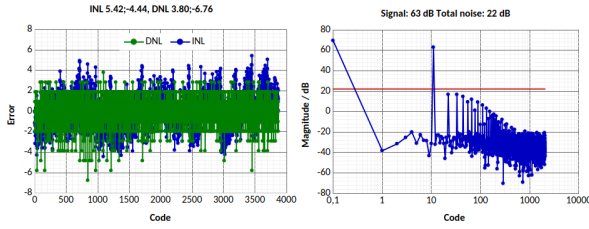


Figure 12. Improved custom calibration INL, DNL, and SNR.

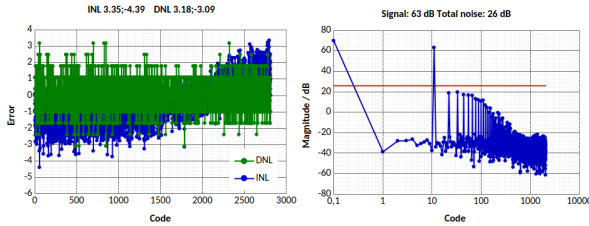


Figure 13. Improved lookup calibration INL, DNL and SNR.

data displaying for each output code the maximum input code. Then a lookup for each possible input code is done for the maximum input code. In this example 2840 unique codes remained (Figure 13).

Table VI shows the results for ENOB INL and DNL for an ideal 12-bit sine signal, the uncalibrated serial DAC sine signal, custom and lookup calibrated DAC sine signal.

It was also investigated how the number of FFT points NFFT is affecting signal to noise ratio (Table VII).

More FFT points decreased the frequency of the signal increasing the ENOB. This was confirmed measuring with different frequencies (Table VIII).

As expected from the full settling time measurement big changes in voltage present in high signal frequency measurements can cause this. This would also be a typical behaviour in the presence of jitter. Since the board is generating the signal synchronized to the ADC data acquisition there should be no jitter present. More research in this area to improve the circuit is needed.

VI. SUMMARY AND OUTLOOK

A binary element serial C DAC is presented with theory, low level, high level simulation, circuit realization, analysis tools and characterization. Typical characterization steps were done using online web based JavaScript internet tools for analysis. The circuit is very cheap to built and shows enough errors to be an interesting subject for study.

The acquisition of 1 k data points took 108 s, 60 ms and 30 s for LTSPICE simulation, JavaScript DAC simulation and measurement. With the number of bits the simulation times increase exponentially. A high level simulation is very versatile and fast, but can only model what is found via LTSPICE simulation or

Table VI
CALIBRATION SNR, INL, AND DNL

	Ideal Sine	No Calibration	Custom Calibration	Lookup Calibration
Periods	11	11	11	11
Signal Magnitude dB	63.01	62.39	62.92	63.01
Total Noise Magnitude dB	-10.90	54.41	22.27	25.74
ENOB	12.0	1.0	6.5	5.9
INL min	0	-12	-6	-6
INL max	0	8	5	3
DNL min	0	-16	-6	-3
DNL max	0	5	5	6
NFFT	4 k	4 k	4 k	4 k

Table VII
FREQUENCY, NUMBER OF FFT POINTS AND SNR

	Ideal Sine	Custom Calibration	Custom Calibration	Custom Calibration
Periods	11	11	11	11
Signal Magnitude dB	63.01	62.88	62.92	62.92
Total Noise Magnitude dB	-10.90	33.03	22.27	11.37
ENOB	12.0	4.7	6.5	8.3
NFFT	4 k	4 k	4 k	4 k

measurement. Correlating data confirm understanding of the DAC circuit.

JavaScript tools were also used to read and analyze LTSPICE data. INL, DNL from ramp data and FFT and SNR of sine data are automatically calculated and displayed.

Open access internet based tools make it very easy to analyze data and do simulation. The underlying JavaScript code can be locally copied and modified for different applications and failure modes. Even big data amounts can nowadays be analyzed in the browser in a short amount of time.

All characterization steps with pitfalls were presented: Noise (base code), averaging (increased number of bits), settling time (full scale and mid scale), ramp test (start, stop codes), sine test (offset, range, NFFT, odd or prime number of periods), FFT, SNR and error correction.

Two error correction procedures for binary element DACs were presented in detail and results discussed. The error correction showed a significant improvement in INL, DNL and SNDR. Since the correction procedures were also implemented in JavaScript the code can be studied in detail and improved. The simple circuit shows typical DAC errors and can be easily modified for experiments. This makes characterization of DACs easier and accessible.

Final measurement data showed INL, DNL below 6, SNR of 46.5 dB and ENOB with error correction.

Table VIII
FREQUENCY, PERIODS, AND SNR

Periods	Signal Magnitude dB	Total Noise Magnitude dB	ENOB	NFFT
11	62.00	32.81	4.6	1 k
43	61.68	41.30	3.1	1 k
44	61.68	41.37	3.1	1 k
179	59.42	44.51	2.2	1 k

Improvement of error correction was 32 dB or 5.5 ENOB.

Future work will improve the tool chain and enable more interactive web based DAC characterization. Variations of the circuit to operate from a battery supply and have higher voltage CLK levels are planned to study circuit limitations. More analysis of the circuit is needed to improve the circuit, decrease the error, increase the resolution and speed up the sampling frequency.



Joerg E. Vollrath received 1989 his Dipl. Ing. and 1994 his Ph. D. in electrical engineering, semiconductor technology at the University of Darmstadt, Germany. Since then he worked for the memory division of Siemens and Infineon Technologies and Qimonda in various locations in the USA and Germany. He is now a Professor for Electronics at the University of Applied Sciences, Kempten, Germany. His expertise and interests lie in the field of design

of analog and digital circuits, programmable logic, test, characterization, yield, manufacturing and reliability. He has published 22 papers and holds 23 patents.

REFERENCES

- [1] M.P. Tiilikainen. "A 14-bit 1.8-V 20-mW 1-mm/sup 2/ CMOS DAC". In: *Solid-State Circuits, IEEE Journal of* 36 (Aug. 2001), pp. 1144–1147. DOI: 10.1109/4.933474.
- [2] Jen-Huan Tsai et al. "A 14-bit 200MS/s current-steering DAC achieving over 82dB SFDR with digitally-assisted calibration and dynamic matching techniques". In: *Proceedings of Technical Program of 2012 VLSI Design, Automation and Test*. 2012, pp. 1–4. DOI: 10.1109/VLSI-DAT.2012.6212594.
- [3] D.A. Mercer. "Low power approaches to high speed CMOS current steering DACs". In: Oct. 2006, pp. 153–160. DOI: 10.1109/CICC.2006.320868.
- [4] Jesper Steensgaard, U.-K. Moon, and Gabor Temes. "Mismatch-shaping serial digital-to-analog converter". In: vol. 2. Aug. 1999, 5–8 vol.2. ISBN: 0-7803-5471-0. DOI: 10.1109/ISCAS.1999.780472.
- [5] J. Vollrath. *Low cost serial DAC simulation, realization, error correction and characterization*. URL: <https://personalpages.hs-kempten.de/~vollratj/Projekte/CSerial.html> (visited on 10/07/2024).
- [6] Chi-Hung Lin and K. Bult. "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm/sup 2/". In: *Solid-State Circuits, IEEE Journal of* 33 (Jan. 1999), pp. 1948–1958. DOI: 10.1109/4.735535.

Improving Hardware Security through Encryption of JTAG TAP Ports: An Analytical Comparison of Algorithms, with a Focus on RSA

Soham Sanjay Dekhane, Andreas Siggelkow

Abstract—In today’s age it is vital to protect our devices from cyber threats. This research paper explores the field of hardware-based cybersecurity specifically focusing on securing Joint Test Action Group (JTAG) Test Access Ports (TAP). This research evaluates different encryption algorithms’ effectiveness in enhancing the security of these ports, which are one of the most widely used debug ports for embedded systems. This research focusses on the RSA (Rivest Shamir Adleman) encryption algorithm analyzing its strengths and weaknesses compared to other cryptographic methods. Through an analysis, the aim is to offer insights into selecting the most robust and efficient encryption techniques for reinforcing JTAG TAP port’s security. By analyzing and comparing encryption techniques this research provides insights, into ongoing efforts aimed at enhancing the security of embedded systems.

Index Terms—RSA, JTAG, TAP, VHDL, debug, cryptography

I. INTRODUCTION

Cyber threats present enormous risks to individuals, businesses, and even entire nations in today’s connected world. Hardware security and trust have become a pressing issue during the last two decades due to the globalization of the semi-conductor supply chain and ubiquitous network connection of computing devices. Strong key management and strong authentication are provided by cryptographic hardware modules.[7] Only authorized individuals can access sensitive information and vital systems because of these hardware-based cryptographic solutions. Individuals and organizations can reduce the danger of unauthorized access, data leakage, and other criminal activity by utilizing hardware cryptography. Encryption is used in almost everyday life right from opening garage doors to credit cards. For such encryptions or cryptographic operations, generation of unique, random and secure keys is very important. Debug ports can be extremely helpful in designing embedded systems but they present a huge vulnerability. From the perspective of the security chain’s resilience, random number generation is a vital

task in modern encryption and security applications. In reality, the encryption keys needed for ciphers are created using random numbers. Therefore, any flaw in the key generation procedure may allow information to be leaked that might be exploited to crack even the most robust cipher [4]. A hacker with malicious intent can get physical access of the system using this debug port can cause a huge damage. The obvious solution is securing the debug port by limiting the access to it by implementing some encryption algorithm. Hardware based cybersecurity; i.e. integrating cybersecurity measures in the processing unit of the embedded system itself can provide with various advantages; the biggest being that it is not at all vulnerable to cyber attacks unless the attacker has physical access to the hardware.

II. JTAG

JTAG, also known as the Joint Test Action Group is an widely-used technology, in the field of digital testing. Its primary purpose is to ensure that electronic systems operate correctly and to diagnose any faults they may have. JTAG consists of components that work together to make testing processes efficient and effective. At the core of this system is the Test Access Port (TAP) which acts as a communication interface between test equipment and the on chip test logic. The TAP includes registers like the Instruction Register (IR) and Data Register (DR) which allow for data exchange during testing procedures. Through a set of predefined instructions JTAG can perform tasks such, as scan testing, memory testing and providing debug support. The JTAG is a serial communication, four wire protocol. The four signals TDI, TDO, TMS and TCLK with TSRT being an extra optional signal are a part of the JTAG protocol which belongs to the IEEE 1149.1 standard as shown in the image 1. JTAG’s structure and functionalities provide a seamless flow in the testing process. The division of the TAP registers into the IR and DR, along with the availability of various test instructions, allows for a clear hierarchy and organization of operations. This ensures that testing procedures can be carried out methodically and precisely, leading to more accurate fault identification and diagnosis.

Soham Sanjay Dekhane, sohamdekhane@gmail.com, Andreas Siggelkow, andreas.siggelkow@rwu.de.Hochschule Ravensburg-Weingarten, Doggenreidstraße, 88250 Weingarten.

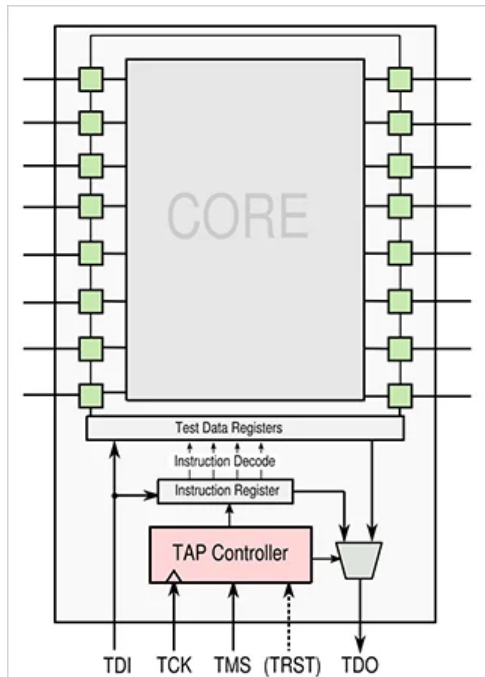


Figure 1. JTAG Registers [1].

Almost all digital systems have a debug interface [10] with different possibilities to attack the system [6]. This debug logic connects all sub-blocks in the system by means of a shadow bus system in order to test or debug it. This could act as a back door which is not secured. To equip this back door and all connection points of the debug bus with a lock, is the focus of the system introduced in the following. It is just the base system for different cipher/debug pairs, it is the base of a system evaluation. The lock can be a cipher system. Parallel to the debug problem is the update over the air possibility in such systems, especially modern cars and IoT. Also this back door can be secured by ciphering. An emulator of this kind has been presented in [5].

The back door itself is the well known JTAG port [8]. The element, which accesses all logic on chip is the JTAG port together with the test access port (TAP) controller. The TAP-Controller is implemented as a finite-state-machine (figure 2). The states of this FSM can be summarised as below.

- **Test-Logic-Reset:** This command resets JTAG circuits. It returns to this state whenever the TRST (optional) signal is asserted. Remember that if TMS is set to 1 for 5 consecutive TCK cycles, the TAP controller will return to this state regardless of the state it may be in. We can still reset the circuit, then, even without the TRST signal.
- **Run-Test/Idle:** The FSM is in this state as it waits for some test operations to finish.
- **Select-DR/Scan and Select-IR/Scan:** These are temporary states that enable the corresponding Register's test data sequence to be started (the

selected IR is in the Select-IR/Scan state, and the selected DR is in the Select-DR/Scan state).

- **Capture-DR and Capture-IR:** During this state, data can be loaded simultaneously into each Register.
- **Shift-DR and Shift-IR:** In these states, the necessary test data is serially loaded (or unloaded) into (or from) the appropriate Register. If you look at Figure 2, the TAP controller will remain in this state as long as TMS is zero. One data bit is shifted through TDI (or TDO) and into (or out of) the chosen Register for each clock cycle.
- **Exit1-DR and Exit1-IR:** All parallel-loaded (from the Capture-DR and Capture-IR state) or serial-loaded (from the Shift-DR and Shift IR state) data are held in the Register in this state.
- **Pause-DR and Pause-IR:** The FSM pauses here in order to await an external operation.
- **Exit2-DR and Exit2-IR:** These states signify the conclusion of the Pause-DR or Pause-IR operation and allow the TAP controller to return to the Shift-DR or Shift-IR state for additional data to be shifted in (or shifted out).
- **Update-DR and Update-IR:** The test data stored in the first flop of the Register (typically all Registers have two flops for each bit; we'll talk about that later) is loaded to the second flop in this state.

The signal timing is defined as follows: The **test mode select (TMS)** will be captured with every rising edge of **test clock (TCK)**. Also **test data in (TDI)** will be taken with the rising edge of TCK. Contrary to this, **test data out (TDO)** will be driven with the falling edge of TCK. So, the wiring to a second chip, which receives the output of the actual SoC, could be allowed a delay of one half of the period of TCK.

III. ENCRYPTION VS HASHING

For handling the data during the exchange via JTAG, there are 2 possible options. The data can either be sent as a plain text or it can be sent in a form which is completely unrecognisable. Sending the data as a plain text is obviously very risky as a hacker with malicious intent can easily get access to this data and it would not remain a secret anymore. If the data is to be sent as completely unrecognisable, there are again 2 options; the data needs to be either encrypted or hashed.

A. Encryption

Encryption involves converting data into a format using an algorithm and a key. The encrypted data, also known as ciphertext can only be deciphered by someone who possesses the corresponding decryption key. Encryption is commonly used to protect data during transmission (data, in transit). When stored on devices or servers (data at rest). This means that the encryption process is a reversible one and the original text/data can be obtained by decrypting it.

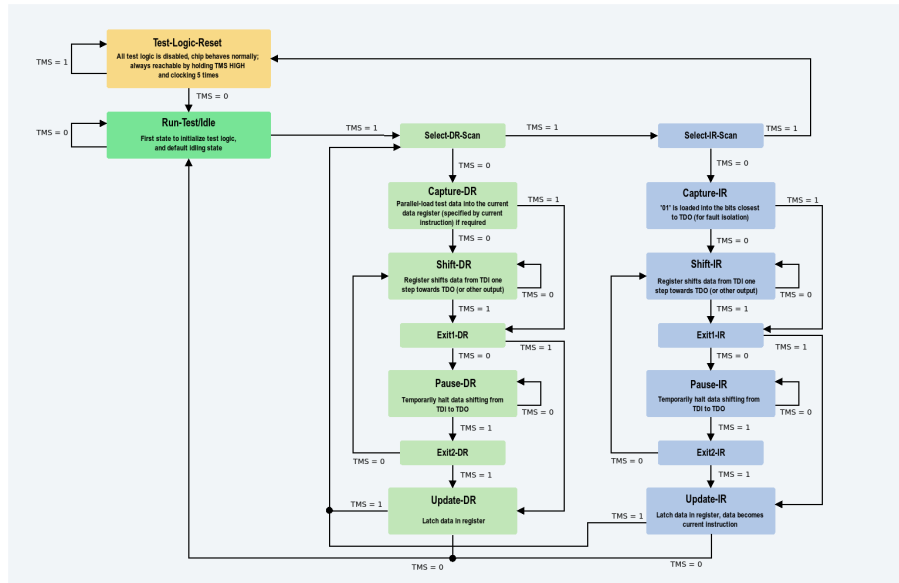


Figure 2. TAP-FSM [1].

B. Hashing

Hashing, unlike encryption functions is a one way process where data is transformed into a string of characters called a hash value. This hash value acts like a fingerprint ensuring the originality and authenticity of the data. Carefully designed hashing algorithms create collision hash values meaning it is computationally difficult to find two inputs that produce the same hash value. The property of collision resistance guarantees that if an attacker modifies the data the resulting hash value will change. This change serves as an indication that the data has been altered, notifying the recipient about information compromise. Hashing is particularly useful, for verifying file integrity when stored or transmitted over networks. By comparing the received files hash value with its counterpart one can confirm if any tampering has occurred.

When comparing Encryption and Hashing for the goal of this research work, it is clear that encryption has to be used to fulfill the aim. This is because the process of hashing, though secure, is an irreversible process and the original data can never be obtained once it is hashed. When exchanging data through JTAG, it would be computationally very heavy to generate the hashes for every permutation of the message data and compare it with the target hash. Considering an example of 16 bit hashed data being transmitted over JTAG, the worst possible case scenario would be generating 65535 different hashes and comparing them with the transmitted hash in order to authenticate the data.

IV. COMPARING VARIOUS CRYPTOGRAPHY ALGORITHMS

The debug interfaces opens a back door which can be easily accessed by a hacker with a malicious intent.

To secure the debug interface, some type of cybersecurity algorithm is required. But the question arises which one. A software based algorithm or a hardware based. When it comes to software-based security, it's all about using programs and applications to protect a system from malware, viruses, and other cyber threats. Many businesses and organizations rely heavily on software-based security because it's affordable and readily available. Examples of software-based security include antivirus programs, encryption software, and firewalls. While these can be effective in stopping many types of cyber threats, they still have their limitations. One major disadvantage of software-based security is that it's vulnerable to attacks from hackers who know how to exploit software vulnerabilities. No matter how sophisticated the software is, it will always be prone to vulnerabilities. Another disadvantage of software-based security is that it may not protect against physical attacks. If a hacker gains physical access to a computer or other device, software-based security will not be able to stop them. While software-based security can be effective in certain situations, it's not always the best solution. Meanwhile, hardware-based security involves using physical devices to safeguard networks, systems, and data. It offers a more robust defense mechanism against cyber threats, as compared to software-based security. Hardware-based security includes devices such as smart cards, token-based authentication, and biometric authentication. These devices provide an extra layer of security that cannot be duplicated by software. Hardware-based security is more secure than software-based security. This is because hardware devices are specifically designed to perform security functions, whereas software-based security can be compromised by vulnerabilities in

the software. Additionally, hardware-based security devices can protect against physical attacks and are inherently more difficult to hack. Before choosing a perfect cybersecurity algorithm, it has to be made sure that the correct algorithm is chosen by weighing the pros and the cons. The first decision would be whether to select a symmetric cryptography algorithm or an asymmetric one. Symmetric Cryptography employs a single key for both the encryption and decryption processes. It is just like a lock that can be unlocked using the same key used to lock it. Similarly, in symmetric cryptography, the sender and receiver share the same secret key, which is utilized to encrypt and decrypt the data. As a consequence, this method is highly efficient and suitable for securing large volumes of data. However, the main challenge lies in the secure distribution of the shared key among the communicating parties. Without proper measures, the secrecy may be compromised, leading to vulnerabilities. On the other hand, asymmetric cryptography, also known as public-key cryptography, operates with a pair of distinct yet mathematically linked keys. This method involves a public key, accessible to anyone, and a private key, tightly held by its owner. To borrow an analogy, picture a locked mailbox where anyone can drop a message using the public key, but only the owner possesses the private key to open and access the contents within. With this mechanism, data can be encrypted with the recipient's public key and decrypted with their private key. One notable advantage of asymmetric cryptography is that it eliminates the need for secure key exchange. Nevertheless, it is computationally more intensive, making it less suitable for encrypting large amounts of data compared to symmetric cryptography. Since our priority is to make the encrypt the data as securely as possible, and no large amount of data to be encrypted, a public key cryptography will be perfect for this algorithm. An algorithm is said to have a "security level of n bit" if the best known attack requires 2^n steps.[14] The table I compares the security level in bits provided by various cryptography algorithms.

By analysing the above table, it can be concluded that the RSA algorithm is perfect for the objective of this thesis. The RSA algorithm is a asymmetric cryptography algorithm which gives it an edge over the AES and the 3DES algorithms. Even though the RSA algorithm requires sizeably longer keys than the Elliptic Curve Algorithms and the same length of keys as the Discrete Logarithm Algorithms; the complexity of the algorithms is what makes the difference. The other algorithms are way too complex and require far more computational power when compared to the RSA algorithm. Such complexity is not required for this application.

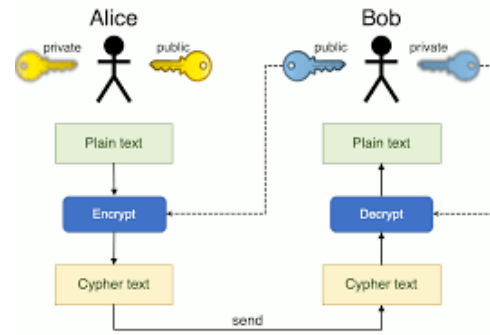


Figure 3. RSA example [2].

V. THE RSA ALGORITHM

RSA or Rivest-Shamir-Adleman, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, is a widely adopted cryptographic algorithm. The RSA Algorithm is a cryptographic algorithm that encrypts and decrypts messages through the use of public and private keys. It is based on the mathematical problem of factoring large integers into prime factors, which is believed to be difficult for classical computers. In RSA, each user has two different keys - a public key and a private key. The public key is shared with others, whereas the private key is kept secret. Considering the widely used Alice and Bob example; When Alice wants to send a message to Bob, she will encrypt it using Bob's public key. Bob can then decrypt the message using his private key. This way, even if someone intercepts the message, they won't be able to read it since they don't have Bob's private key. This is demonstrated in the figure 3. Similarly, Bob can send an encrypted message to Alice using her public key. Alice can then decrypt the message using her private key. This method ensures that only Bob and Alice can read the messages they send to each other. RSA Algorithm is based on the difficulty of factorizing large numbers into two prime numbers. This makes RSA highly secure and strong against most of the attacks. It has a security strength of 2048 bits and above, which makes it practically impossible to break using conventional computing methods. However, with the advancement in technology, even 2048 bits security may not be enough to withstand attacks from quantum computers. In such cases, RSA will have to be replaced with newer and stronger cryptographic algorithms. Breaking RSA is considered to be a daunting task due to the immense computational power and time required. RSA encryption can only be broken by factoring the large composite number into two prime numbers. This means that if the key used is long enough, it may take billions of years to break the encryption.[14] Currently, it is believed that it will be possible to factor 1024-bit values within the next 10 to 15 years with the help of quantum computing. To minimize the risk of such an attack,

Table I
DIFFERENT SECURITY LEVELS FOR CRYPTOGRAPHY ALGORITHMS [14].

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer Factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete Logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic Curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric Key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

it is recommended to choose the RSA parameters of 2048-4096 bits. Even though RSA is considered to be highly secure, there have been instances where security breaches have occurred. One such attack was the Bleichenbacher's attack, which exploited a vulnerability in the RSA encryption implementation. This resulted in the attacker gaining access to secure information, which highlights the importance of proper implementation of RSA. Another challenge is the susceptibility of RSA to side-channel attacks. These attacks can occur when an attacker measures the physical characteristics of the system, such as power consumption, to gain information about the key. Therefore, securing the implementation of RSA is essential to prevent such attacks. While the RSA algorithm offers robust security, it has some challenges that make its implementation complex. One of the biggest challenges is computational complexity. Generating large prime numbers and performing complex operations can be time-consuming and resource-intensive. Additionally, key management can be challenging, especially when dealing with a large number of users. Safe key storage and distribution are necessary to prevent unauthorized access. The following steps illustrate the generation of the key for the RSA algorithm:

- 1) Select two distinct prime numbers; Assume they are p and q .
- 2) Compute their product " n " such that $n = p \cdot q$.
- 3) Calculate the Euler's totient function $\varphi(n) = (p - 1) * (q - 1)$.
- 4) Select an " e " such that $0 < e < [\varphi(n)]$ and e & $\varphi(n)$ are coprime i.e $\gcd(e, \varphi(n)) = 1$.
- 5) Calculate a " d " such that $d \cdot e \equiv 1 \text{ mod } \varphi(n)$
- 6) (n, e) is the Public key and is used for encryption while (n, d) is the Private key and is used for decryption.

Once the public and private keys are calculated, the messages can be encrypted and decrypted as follows: Let x be the data and y be the encrypted data. Then, the encryption is done as $y = x^e \text{ mod } n$ while the decryption is done as $x = y^d \text{ mod } n$.

VI. DISCUSSION AND FUTURE WORK

Even though the RSA algorithm seems the best option for securing the JTAG debug port, there are some limitations when implementing it using VHDL.

As discussed in the previous section, a proper RSA implementation would require keys that are atleast 1024 bits long. But, the integers in VHDL 2008 are capped at 32 bits while the integers in VHDL 2019 are capped at 64 bits. This obviously poses an issue as the key can not be declared in VHDL. For this, there would be a need to create new VHDL libraries which support such large integers. The implementation of the RSA algorithm also involves many complex mathematical operations, e.g. it involves the exponential operator, which is cannot be synthesized by an FPGA. Hence this needs to be worked around when implementing the RSA algorithm using VHDL. Another thing that comes into one's mind while implementing such an asymmetric key algorithm is securing the keys by means of a certificate. But one can argue that this is not required with the RSA algorithm. The keys of a proper implementation of RSA algorithm are atleast 1024 bits long and factorizing the public key in order to obtain the private key is almost impossible. Hence, securing the public key with a certificate would just add unnecessary computational effort. One of the future works in order to implement this, as discussed before, is creating new libraries in VHDL. Creation of a server to pre-compute and store the keys could also be done. Although the keys for a specific implementation would be stored in the implementation itself, creating such a server could reduce the effort of computing the keys every time just before the implementation. Hence, here it has been discussed how can the encryption of the JTAG TAP Port can be done through hardware based encryption. Various methods have been discussed, compared and the importance of the RSA algorithm for this application has been discussed.

REFERENCES

- [1] Accessed on October 21, 2023. URL: <https://www.allaboutcircuits.com/technical-articles/jtag-test-access-port-tap-state-machine/>.
- [2] Accessed on October 6, 2023. URL: <https://www.coengodegebure.com/surviving-an-infosec-job-interview-cryptography/>.
- [3] Accessed on October 1, 2023. URL: <https://www.xjtag.com/about-jtag/what-is-jtag/>.
- [4] Luca Baldanzi u.a. *Cryptographically Secure Pseudo-Random Number Generator IP-Core Based on SHA2 Algorithm*. Applications in Electronics Pervading Industry, Environment, Society – Sensing Systems und Pervasive Intelligence), 2020. ISBN: <https://doi.org/10.3390/s20071869>.
- [5] Gregor Benz und Andreas Siggelkow. *Implementation of a GPS and GSM module into a Zynq Z7 SoC based emulator tracking system*. Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg, 2020.
- [6] Swarup Bhunia, Sandip Ray und Susmita Sur-Kolay (Editors). *Fundamentals of IP and SoC Security: Design, Verification, and Debug*. Springer, Cham, Switzerland, 2017. ISBN: 978-3-319-50055-3.
- [7] Wei Hu u.a. *An Overview of Hardware Security and Trust: Threats, Countermeasures and Design Tools*. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS und SYSTEMS, VOL. X, NO. X, 2021.
- [8] *IEEE Standard for Test Access Port and Boundary-Scan Architecture*. IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001), vol., no., pp.1-444, 2013. ISBN: doi: 10.1109/IEEESTD.2013.6515989.
- [9] Zhengato Jiang, Yang Zhan, Dan Chen und Yumin Wang. *Two methods of directly constructing probabilistic public-key encryption primitives based on third-order LFSR sequences*. <https://doi.org/10.1016/j.amc.2005.01.097>: Applied Mathematics and Computation, 2005.
- [10] C.F. Kao und H.M. Chen. *Hardware-Software Approaches to In-Circuit Emulation for Embedded Processors*. IEEE Design und Test, 25 (5): 462 - 477, 2008.
- [11] Barış Bülent Kirlar und Melek Çil. *On the k-th order lfsr sequence with public key cryptosystems*. <https://doi.org/10.1515/ms-2016-0294>: De Gruyter, 2017.
- [12] Paul Kocher u.a. *Security as a new dimension in embedded system design*. IEEE Proceedings. 41st Design Automation Conference, 2004. ISBN: 1-51183-828-8.
- [13] Kyungroul Lee, Yeunsu Lee, Hyeji Lee und Kangbin Yim. *A Brief Review on JTAG Security*. DOI: 10.1109/IMIS.2016.102: 10th International Conference on Innovative Mobile und Internet Services in Ubiquitous Computing (IMIS), 2016. ISBN: 978-1-5090-0984-8.
- [14] Christof Paar und Jan Pelzl. *Understanding Cryptography*. Springer, 2010. ISBN: 978-3-642-04101-3.
- [15] Keun-Young Park, Sang-Guun Yoo und Juho Kim. *Debug Port Protection Mechanism for Secure Embedded Devices*. DOI: 10.5573/JSTS.2012.12.2.240: Journal of Semiconductor Technology und Science, 2012.
- [16] Sang Guun Yoo, KEUN-YOUNG PARK und Josphine Kim. *Software Architecture of JTAG Security System*. WSEAS Transactions on Systems, 2012.



Soham Sanjay Dekhane received his B.Tech. degree in Electronics and Telecommunication Engineering from Symbiosis International (Deemed) University, India in July 2021. Since September 2021, he is pursuing his Master's degree in Electrical Engineering and Embedded Systems at Hochschule Ravensburg-Weingarten.



Andreas Siggelkow received the academic degree Dipl. -Ing. degree in 1988 from the University of Karlsruhe. In 1996, he obtained his doctorate at the University of Stuttgart for Dr. -Ing. From 1996 to 2007 he worked for Infineon on specifications for base-band processor ASICs. Since 2007, he is a professor for ASIC-Design und Computer Architecture at the Hochschule Ravensburg-Weingarten.

Coupling between traces and power planes in PCBs. How is it influenced and what are its consequences?

Justus Gottwald, Michael Engelbrecht, Bernhard M. Rieß

Abstract—This work analyzes coupling between transmission lines and power planes and to what extent power plane resonances influence the behavior. In addition, it examines how resonances can be reduced or prevented. Initially, some hypotheses on signal coupling on PCBs, the effect of stitching and decoupling capacitors, are put forward. Next, a new verification board is developed to measure the suspected effects. Finally, measurements were executed to either proof or reject the initial hypotheses. Recommendations on future PCB design strategies summarize this work.

Index Terms—Printed Circuit Board Design, Transmission Lines, Power Planes, Coupling, Capacitors, Interference, Resonances, S-Parameters

I. INTRODUCTION

Modern electronic systems demand high data rates, necessitating increased symbol rates per lane. However, higher edge rates introduce broad-spectrum energy, leading to potential interference with other signals and power delivery networks (PDNs). Modulation techniques like pulse-amplitude modulation (PAM) help improve data rates but increase susceptibility to interference, reducing the signal-to-noise ratio (SNR) [1][2]. Since power planes are integral to PCB designs, they can either be affected by or contribute to signal interference, impacting both digital and analog circuits.

PCB designers must address these challenges through effective layout strategies, including the use of decoupling capacitors and ground planes to minimize crosstalk and electromagnetic compatibility (EMC) issues [3]. While signal integrity (SI) is generally well-understood in transmission line theory [4], power plane resonances remain a critical factor affecting overall system performance. Standing waves between plane pairs can cause unwanted coupling, impacting high-speed signals and leading to EMC concerns [5][6].

This study examines these interactions by formulating hypotheses on coupling behavior, designing a verification board for experimental testing, and analyzing measurement results. The findings will help refine PCB design strategies to mitigate interference and enhance performance.

Justus Gottwald, justus.gottwald.xw@renesas.com, Michael Engelbrecht, michael.engelbrecht@renesas.com, Renesas Electronics Europe GmbH, Düsseldorf, Germany.

Bernhard M. Rieß, bernhard.riess@hs-duesseldorf.de, University of Applied Sciences Düsseldorf, Germany.

II. HYPOTHESES

This section gives a brief overview on all hypotheses to be proven or disproven in this work:

A. Coupling

- H1:** Coupling between planes and traces is highest at plane resonance frequencies.
- H2:** Coupling to planes is higher the closer and longer the trace is to the apex of standing waves.
- H3:** A plane between power plane and trace highly prevents coupling.
- H4:** Coupling to planes is much higher in common mode than in differential mode.

B. PCB

- H5:** Effects of power plane resonances are location-dependent.
- H6:** Frequency-dependent dielectric constant increases frequency difference between resonances of multiples of a mode.
- H7:** Smaller distance between overlying planes dampens resonance peaks.
- H8:** S-parameters of microstrips are nearly identical for same dielectric.

C. Capacitors

- H9:** Stitching capacitors prevent a sudden change of the characteristic impedance and therefore reduce emissions from traces.
- H10:** Well-selected decoupling capacitors dampen resonances and therefore reduce coupling.

III. FUNDAMENTALS

This section summarizes the theory required to evaluate measurements with regard to the hypotheses.

A. Traces

PCB design involves routing signals through various trace structures, each characterized by a specific impedance that must remain consistent to prevent signal reflections and degradation [7]. The equivalent circuit model of a transmission line is defined by its characteristic impedance [4]:

$$Z_l = \sqrt{\frac{R' + j\omega L'}{G' + j\omega C'}} \quad (1)$$

Assuming negligible resistance R' and conductance G' , the equation simplifies to [4] [7]:

$$Z_l \approx \sqrt{\frac{L'}{C'}} \quad (2)$$

where L' and C' depend on trace geometry and the dielectric medium.

1) Coated Microstrip:

A coated microstrip consists of a surface trace with a reference ground plane beneath it. The characteristic impedance is approximated as:

$$Z_N = \frac{60 \Omega}{\sqrt{\epsilon_{eff}}} \cdot \ln \left(\frac{6.8 \cdot h}{0.8 \cdot w + t_{cu}} \right) \quad (3)$$

where ϵ_{eff} is the effective dielectric constant, h is the substrate thickness, w is trace width, and t_{cu} is copper thickness. The formula assumes a rectangular cross-section, though real traces may have minor etching deviations [8].

2) Edge-coupled Coated Microstrip:

For differential signaling, an edge-coupled coated microstrip consists of two adjacent traces with a shared reference plane. Its differential impedance is approximated by:

$$Z_{diff} = 2Z_N \left(1 - 0.48 \cdot e^{-0.96 \frac{G}{h}} \right) \quad (4)$$

where G is the trace gap and h is the dielectric thickness. A high $\frac{G}{h}$ ratio results in Z_{diff} approaching twice the single-ended impedance.

Single-ended transmission is prone to interference, especially at high frequencies [4]. In contrast, differential transmission enhances signal integrity by reducing external noise through common-mode rejection. This is achieved as external interference affects both traces equally, allowing the receiver to cancel out noise by subtracting signals [9]. This leads to hypothesis **H4**.

B. Stack-up

A PCB always consists of a stack-up of different materials. We focus on the structure and materials of a flame retardant (FR-4) PCB. Prepregs and cores consist of a resin-filled fiberglass fabric. A core consists of several but a prepreg of one fabric layer only.

The core forms the inner layers, which are pressed together with copper foil using two prepregs on each side. This stack-up is used for the VBs as presented in subsection IV-A.

C. Capacitors

Capacitors are frequently used components in PCB design. The impedance is given in Equation 5 [10].

$$Z_C = \sqrt{R_S^2 + \left(\omega L - \frac{1}{\omega C_0} \right)^2} \quad (5)$$

Equation 5 implies that a real capacitor behaves inductively for frequencies greater than the resonance frequency. Multi-layer ceramic capacitors (MLCC) are mostly used today and are applied this work.

Decoupling capacitors serve as a small energy buffer to stabilize the supply voltage and thus avoid current peaks on the power path. Their task is to reduce the impedance of the supply to filter out noise. Depending on frequency and intensity of the noise, a suitable decoupling capacitor must be selected [11] [12].

In some cases, it is not possible to route an impedance-controlled trace completely via just one reference layer but it is necessary to route via slots. This will deteriorate the signal and may cause electromagnetic compatibility (EMC) problems. One solution may be a stitching capacitor being used for coupling instead of decoupling. A correctly selected stitching capacitor creates a low-impedance path for the HF alternating return current [13] [14]. This leads to hypothesis **H9**.

D. Crosstalk/Coupling

Crosstalk or coupling generally describes how electromagnetic energy is transferred from one part of the circuit to another. There are various subtypes of coupling: capacitive, inductive, and galvanic. Capacitive coupling occurs when a stray capacitance is present. A change in the voltage at this stray capacitor causes a displacement current through it. Inductive coupling arises analogously through mutual inductance.

The coupling of a differential trace to a plane is inductive and capacitive. One of the easiest ways to prevent and reduce crosstalk/coupling is to increase the distance between aggressor the victim. Since both, the mutual inductance and the stray capacitance decrease with increasing distance, crosstalk can be prevented this way. Increasing the layer distance leads to a wider trace if all other parameters remain the same.

The purpose of shielding is to deflect or weaken electromagnetic fields mostly performed by a ground plane [16]. Magnetic fields behave differently from electric fields. Their field lines are continuous. Therefore, they can only be deflected or canceled by other magnetic fields. The strength of the initial field is reduced resulting in a shielding effect [17] [18].

E. Power plane resonances

In the layout of a PCB, the PDN is often realized via planes. Power and ground planes are arranged in parallel, resulting in a plate capacitor. The geometric dimensions of this embedded capacitor generate power plane resonances and a standing wave is formed [19]. Equation 6 [6] describes possible resonance frequencies for a rectangular power plane pair:

$$f_{mnp} = \frac{c}{2\sqrt{\epsilon_r \mu_r}} \sqrt{\left(\frac{m}{a} \right)^2 + \left(\frac{n}{b} \right)^2 + \left(\frac{p}{h} \right)^2} \quad (6)$$

The geometry (length a , width b , and distance h) primarily determines the resonance frequencies. The factor in front of the root can be regarded as constant as the dielectric constant is only slightly frequency-dependent, while $m, n, p \in \mathbb{N}$. The p -modes can be neglected, as the distance of the power plane pair is typically much smaller than its length and width [5].

These resonances can be stimulated by coupling (see subsection III-D), but existing resonances can also disturb other parts of the circuit by the same mechanisms. Obviously, at the resonance frequencies of the plane, the coupling of traces is high. Together with the fact that coupling is always the same in both directions hypotheses **H1** is formulated.

One option to prevent or dampen resonances is to reduce the distance between the planes. However, depending on application and stack-up, a reduced distance between the planes may not be an option. The idea is taken up again in subsection IV-A and leads to hypothesis **H7**. The other dimensions cannot necessarily be reduced either in order to push the resonances into a higher frequency range. Stitching vias to other planes of the same potential splits the plane into multiple cavities with smaller dimensions, which shifts these resonances upwards [5] [19]. Decoupling capacitors can significantly reduce the impedance locally, especially for certain frequencies [12]. This leads to hypothesis **H10**.

F. Scattering parameters

Scattering parameters (S-parameters) are used in high-frequency (HF) technology to describe networks in the frequency domain. They are of particular relevance here as the following measurements will be carried out in this form of representation. The S-parameters for a 2-port network are defined as:

- S_{11} is the input reflection coefficient
- S_{12} is the reverse transmission coefficient
- S_{21} is the forward transmission coefficient
- S_{22} is the output reflection coefficient

S-parameters can be measured using a vector network analyzer (VNA). The VNA stimulates the input port with a specific frequency and amplitude. Simultaneously, the same port receives the reflected wave. The transmission wave is received via a receiver at the output port.

IV. VERIFICATION BOARD

Figure 1 shows the top view of both verification boards that were designed and manufactured.

A. Stack-up and Traces

Both VBs only differ in the core thickness, which is 1.55 mm in “VB 212” and 0.41 mm in “VB 098”. The VB names indicate their total thickness. Copper thicknesses and prepreps are identical. Identical prepreps lead to the assumption that the S-parameters

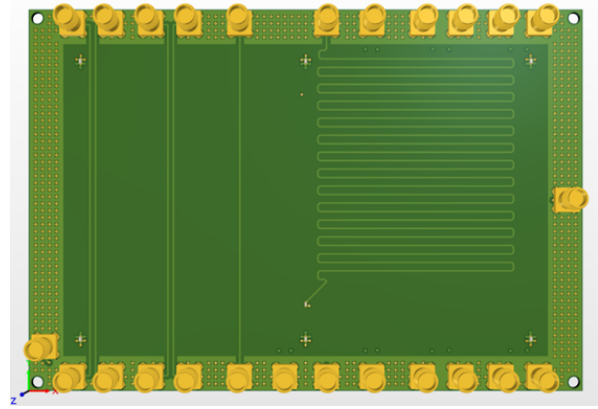


Figure 1. Verification board top view perspective.

Table I
ISOLA DE104 DIELECTRIC CONSTANTS AND DISSIPATION
FACTORS FOR VARIOUS FREQUENCIES [20]

Construction	Resin Content	Dielectric Constant/Dissipation Factor				
		100 MHz	500 MHz	1 GHz	2 GHz	5 GHz
2 x 7628 (WE-CO-0009)	44%	4.58 0.015	4.56 0.016	4.53 0.017	4.51 0.018	4.46 0.018
9 x 7628 (WE-CO-0021)	39%	4.72 0.013	4.68 0.015	4.67 0.017	4.65 0.017	4.56 0.017
2116 (WE-PP-0010)	50%	4.50 0.015	4.46 0.017	4.45 0.019	4.42 0.02	4.36 0.2

of microstrips are quite similar for both stack-ups according to Equation 3. This leads to hypothesis **H8**.

Table I provides the dielectric constant and the dissipation factor of the used “DE104” product series from the dielectric provider Isola.

The first two rows of Table I correspond to the cores while the last row corresponds to the prepreg.

As the power plane resonances depend on the dielectric constant (Equation 6), the resonance frequencies of “VB 098” and “VB 212” should be slightly different for the same mode and identical plane dimensions. With a frequency independent dielectric constant, the distances between neighboring resonances (e.g. f_{100} and f_{200}) should be equal. Since, the dielectric constant is reduced with increasing frequency, the frequency distance between neighboring resonances is expected to increase as expressed in hypothesis **H6**.

A smaller distance between the planes increases dielectric losses. Since the cores of both VBs have similar dissipation factors while having different thickness, hypothesis **H7** be verified with both VBs.

The advantage of both VBs having the same prepreg is that this leads to identical trace dimensions. This allows to compare the coupling behavior on both PCBs.

B. Placement and routing

The VBs have a “flipped” structure i.e. the traces and the decoupling capacitors on the bottom side are mirrored on the top side of the board. Stitching capacitors are only present on the bottom side, as there

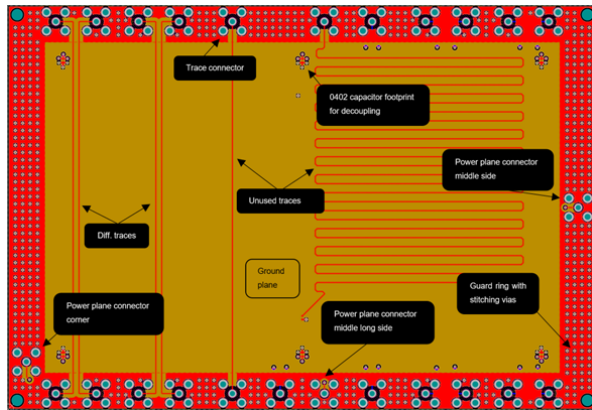


Figure 2. Top view of the VBs in 2D view.

is no reference layer change on the top side. Figure 2 shows the top 2D view of both VBs.

The upper outer layer is shown in red, while the first inner - grounded - layer is given in beige. The outer layer only consists of traces and the guard ring located identically on all other layers. It is connected to ground and improves the ground connection of the surrounding SubMiniature version A (SMA) connectors.

The variety of positions of the differential traces allows to determine the dependence of coupling to the power plane on the location. A trace that runs close to the theoretical location of the peak of a power plane resonance mode for a long distance is expected to couple more highly at this frequency than a trace that runs less long or not at all at the location of the peak as summarized in hypothesis **H2**.

The ground plane on layer 2 should prevent coupling between traces on top layer and the power plane on layer 3 as formulated in hypothesis **H3**.

Three SMA connectors were attached to the power plane to check the location dependence of power plane resonances: One in the middle of the short side (called "middle side"), one in the middle of the long side ("middle long side"), and one at the corner ("corner"). A peak is expected at the "corner" and the "middle side", while neglectable coupling is expected at the "middle long side", as the zero crossing of the resonance is located there described in hypothesis **H5**.

Figure 3 shows the power plane on layer 3 in cyan and the attached SMA connectors. The $140 \times 90 \text{ mm}$ power plane is surrounded by a ground ring and, together with the continuous ground plane (beige), forms the cavity causing power plane resonances.

To prove or disprove hypothesis **H10** a total of eight capacitor footprints are on the VBs. These are designed according to the Electronic Industries Alliance (EIA) 0402 standard to accommodate a variety of capacitors of different capacity.

A section of the bottom side of the PCB is shown in Figure 4. Layer 3 is given in cyan, while blue represents the bottom layer (layer 4). It is a mirror

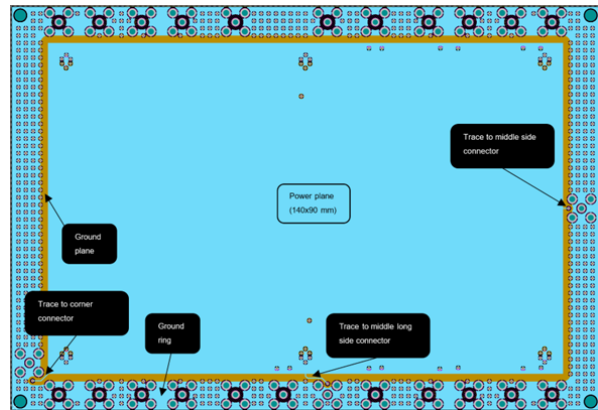


Figure 3. Inner view of VBs in 2D.

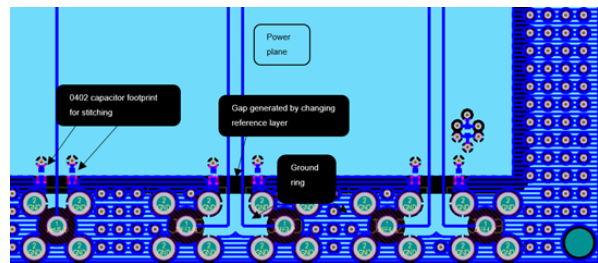


Figure 4. Section bottom view of VBs in 2D.

image of the top side and there again are EIA 0402 footprints for stitching capacitors. They are placed at the transition is between the power plane on the inside and the ground plane on the outside.

V. MEASUREMENT SETUP

Figure 5 shows our general S-parameters measurement setup. In the center a graphical representation of the VBs containing all important traces and connectors is shown.

The following S-parameter measurements were carried out for both VBs without decoupling or stitching capacitors:

- Inner differential pair on top
- Inner differential pair on bottom
- Outer differential pair on top
- Outer differential pair on bottom
- Single-ended traces on top
- Single-ended traces on bottom
- Outer single-ended trace on both sides
- Power plane connectors

Next, the selection of decoupling capacitors contained in Table II were then soldered in and the following S-parameter measurements were repeated:

- Single-ended traces on bottom
- Power plane connectors

Then, an S-parameter measurement was carried out exclusively with stitching capacitors:

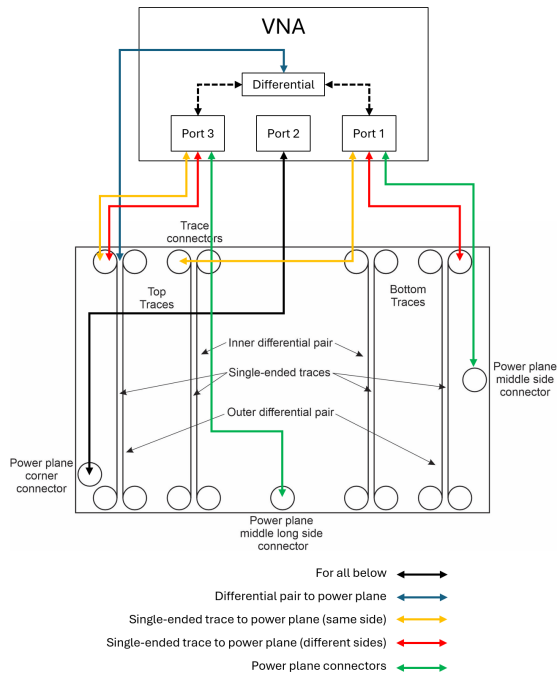


Figure 5. S-parameter measurement setup.

- Inner differential pair on bottom side

It was carried out for the bottom side traces only where the stitching capacitors are used. Only the inner differential pair was selected here, as hypothesis **H9** with regard to the stitching capacitors does not contain any position dependency of the traces.

We applied a Rohde & Schwarz ZVB 8 VNA in the range from 1 MHz to 6 GHz. The “open” ends of the measured traces as well as unused traces on the same side are terminated with $50\ \Omega$. Due to the “mirrored” structure, the measurements are comparable.

Colors in Figure 5 indicate the different measurements performed. As shown in black, port 2 of the VNA is always connected to the power plane corner connector (PPCC) on the bottom left.

A. Differential pair to power plane

The measurement setup for the outer differential pair on the top side to the power plane at the corner is shown in blue in Figure 5. The VNA has been configured so that ports 1 and 3 are combined into one differential port. Measurements are taken for the inner and outer differential pair for both the top and bottom. The unused differential pair on the same side is terminated with $50\ \Omega$. All other traces remain open.

B. Single-ended trace to power plane

The measurement setup for single-ended traces on the same side of the VB is shown in yellow in Figure 5.

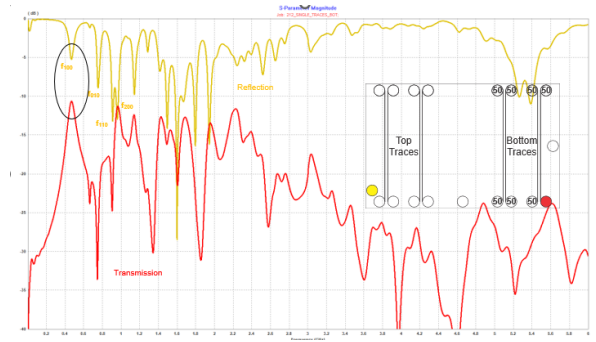


Figure 6. S-parameter plotting of outer single-ended trace to PPCC for “VB 212”.

Ports 1 and 3 of the VNA are both single-ended and are each connected to the outer trace of each differential pair on the same side.

An additional setup for a direct comparison between bottom and top trace can be seen in Figure 5 in red. Here only the outer single-ended traces on both sides are connected and terminated. The rest of the setup is identical to the previous measurement.

C. Power plane connectors

Similar to the measurements explained in subsection V-B, this again is a single-ended setup. The entire measurement arrangement can be seen in green in Figure 5. Port 1 of the VNA is connected to the power plane middle side connector, port 2 to the power plane corner connector, and port 3 to the power plane middle long side connector. This measurement is performed once for each configuration.

VI. MEASUREMENT RESULTS

The S-parameters are plotted using the “S-Parameter Utility” program from Ansys containing a graphical representation of the VB with the corresponding measurement ports for orientation.

A. Coupling

H1: Figure 6 shows the reflection parameter of the PPCC in yellow together with the coupling/transmission parameter in red with respect to the outer single-ended traces on the bottom side.

Coupling reaches a local maximum at the f_{100} resonance. This does not apply to all resonances, which suggests a dependence on the location of the trace (see **H2** below) and the measuring point on the power plane (see **H5**). We conclude that the coupling is highest at the resonances. Thus, hypothesis **H1** is considered to be **proven**.

H2: The two single-ended traces on the bottom side are compared to each other, as they are relatively far apart. The corresponding S-parameter measurement results are shown in Figure 7.

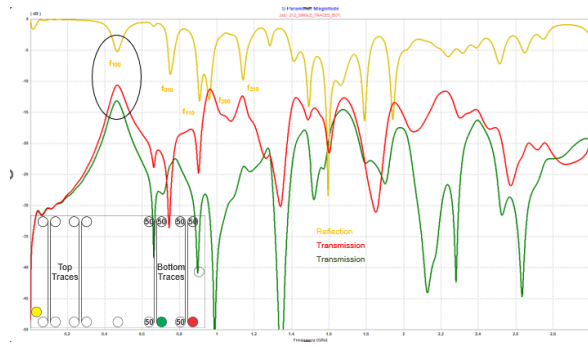
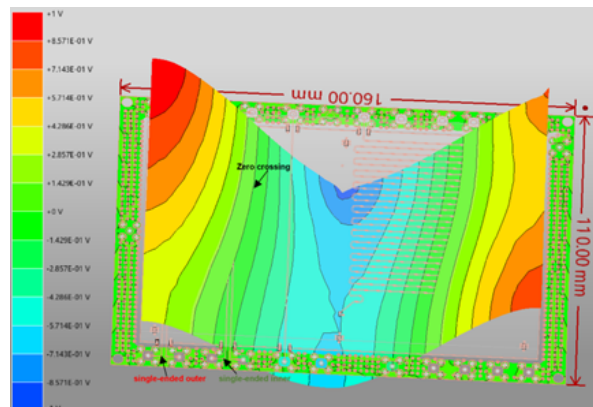


Figure 7. S-parameter plotting of single-ended traces to PPCC.


Figure 8. f_{200} resonance simulation on "VB 212".

The proximity of the trace correlates to the apex of the standing wave and the coupling. The outer trace shown in red is closer to the maximum than the inner trace shown in green indicating that the coupling is more intense. This effect proves to be even stronger on f_{200} whose simulation result can be seen in Figure 8. The standing wave has two zero crossings, the outer trace is again exposed to the maximum, while the inner trace now is almost exposed to zero. The result is a strong coupling of the outer trace and a very weak coupling of the inner trace as shown in Figure 7.

The f_{100} wave is orthogonal to f_{010} . All traces run parallel to the standing wave f_{100} at the same time. The coupling for f_{100} is larger for both single-ended traces than for f_{010} . However, the f_{200} resonance has a weaker coupling for the inner single-ended trace compared to f_{010} , although f_{200} also runs in parallel. Therefore, the coupling depends on how close and how long the trace runs to the apex of the standing wave. Hypothesis **H2** is **confirmed**.

To check **H3** the coupling of a trace to the power plane is compared with and without an intermediate ground plane. In Figure 9, the coupling of an outer single-ended trace on the top of both VBs is compared to the coupling of a trace on the bottom.

Both VBs show a difference of approximately over 50 dB between the bottom and top trace at the f_{100}

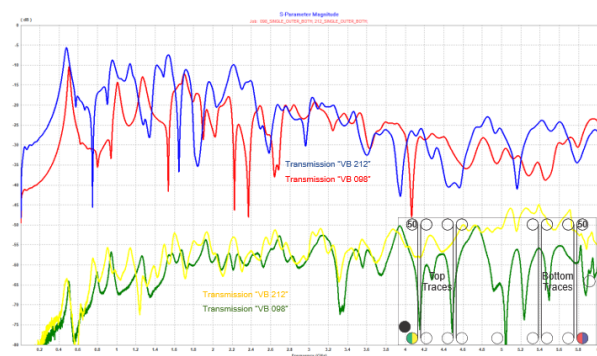


Figure 9. S-parameter plotting of outer single-ended trace to PPCC.

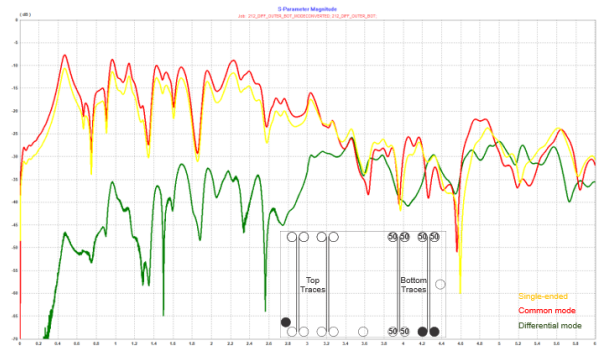


Figure 10. Coupling bottom outer differential pair to PPCC of "VB 212".

resonance. The difference of the f_{100} resonance frequency between the two VBs can be attributed to the different dielectric constants of the cores. In summary, the top traces couple significantly less to the power plane than the bottom traces. **H3** is **confirmed**.

To verify **H4** the coupling of the differential mode is compared to the common mode. Figure 10 shows the outer differential pair on the **bottom** side coupled to PPCC and single-ended coupling.

In the range from 0 to 3 GHz the weaker coupling of the differential mode compared to the common mode can be explained by the electric field. While the electric field penetrates the power plane in common mode, the traces act as a sink for each other in differential mode. The electric field propagates between the two traces.

Since both traces are equally stimulated in common mode, this also corresponds to a doubling of the coupling to the power plane. In logarithmic representation, this corresponds to a gap of 3 dB between common and differential mode coupling.

The range from 3 to 6 GHz reflects a different situation: Differential mode and common mode have a similar coupling. In some cases, the coupling in differential mode is even stronger than the common mode. The 3 dB gap is no longer present.

However, the coupling for the inner differential pair is slightly different as shown in Figure 11.

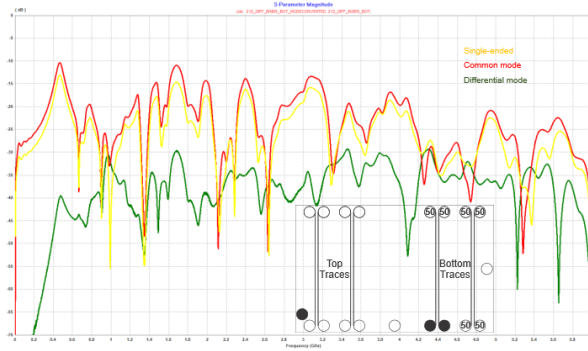


Figure 11. Coupling bottom inner differential pair to PPCC of “VB 212”.

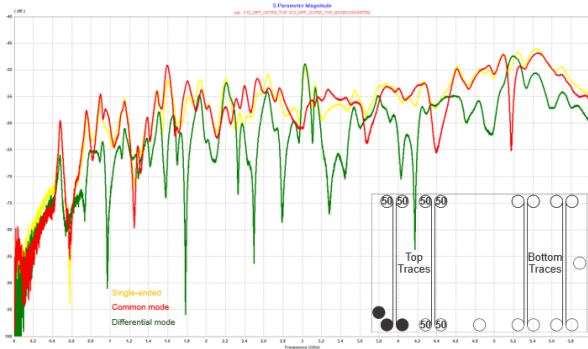


Figure 12. Coupling top outer differential pair to PPCC of “VB 212”.

Again, common mode and differential mode converge for higher frequencies. In general, common mode coupling for the bottom traces is stronger than differential mode coupling.

Measurement results with respect to coupling between the outer differential pair and the PPCC for the **top** traces are shown in Figure 12.

All three curves differ slightly. The 3 dB gap between common mode and single-ended is occasionally present. Since the ground plane absorbs the electric field completely, this coupling is solely due to the magnetic field. There appears to be only a small attenuation due to the differential mode. The reason for this is the large separation of the traces resulting in a low interference of each magnetic field.

In conclusion, common mode coupling is not necessarily stronger than differential mode coupling over the whole frequency range especially for top traces. Hypothesis **H4** is **rejected**.

B. PCB

The hypotheses in this section relate to the stack-up and geometry of the PCB and the applied materials.

Obviously, not only coupling is location-dependent, but also the effects on components. For example, a circuit whose supply is connected to the power plane

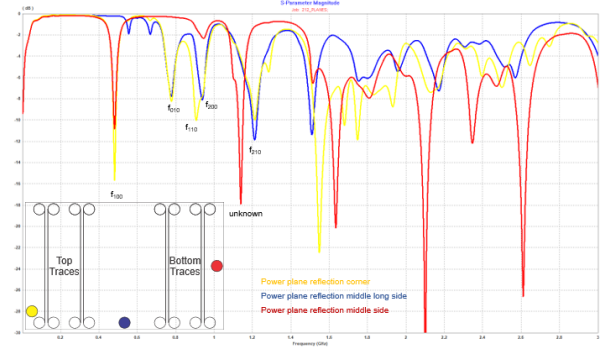


Figure 13. S-parameter plotting of power plane connectors on “VB 212”.

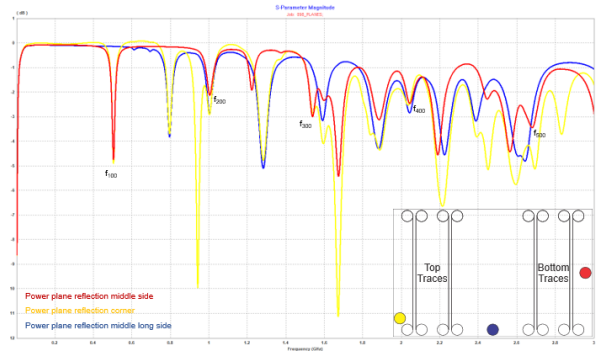


Figure 14. S-parameter plotting of power plane connectors on “VB 098”.

near a zero crossing of the f_{200} resonance, will not be exposed to the standing wave at all. To verify **H5**, the reflection parameters of the three individual power plane connectors are compared in Figure 13. It is immediately apparent that some resonances can be measured more or less clearly depending on the location of the connector.

Interestingly, some resonances cannot be directly assigned to a mode. **H5** is **confirmed**.

To check **H6**, reflection parameters of the power plane connectors of “VB 098” are shown in Figure 14.

For even “ m ”, all three plane connectors must display a deflection, while for odd “ m ” only the corner and middle side connectors must display a deflection. Ideally, the distance between the individual f_{m00} modes is equal to the base frequency f_{100} . Figure 15 clearly shows that the distance between the modes depends on frequency.

However, no direct correlation between the frequency-dependent dielectric constant and the distances can be found. Hypothesis **H6** is **rejected**.

To verify **H7** the S-Parameters of “VB 212” and “VB 098” are shown in Figure 16 in blue and red, respectively. Obviously, a smaller distance between power and ground plane damps the resonances.

The f_{100} resonance is attenuated by an estimated 6 dB by reducing the core distance from 1.55 mm

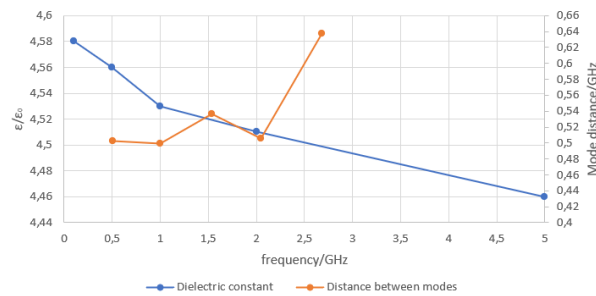


Figure 15. Mode distance and dielectric constant.

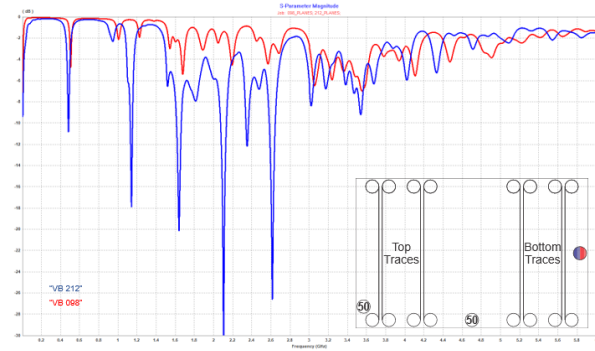


Figure 16. S-parameter plotting of power plane middle connector.

to 0.41 mm. Higher frequencies are attenuated even significantly more. Since the resonances above 3 GHz are generally already strongly attenuated, the difference here is less significant. Thus, hypothesis **H7** is **confirmed**.

To verify hypothesis **H8** the reflection parameters of the outer single-ended traces of both VBs are measured and compared in Figure 17.

The curves of the two top traces shown in blue and green as well as those of the two bottom traces given in yellow and red differ from each other. Although the curves are similar in general, the damping differs, and the resonance frequencies are shifted. Hypothesis **H8** must therefore be **rejected** concluding that a change in the core distance leads to a significant change in the S-parameters of the traces on the outer layers. This can be attributed to the change in the coupling of the plane pair, which forms the references for the traces.

C. Capacitors

Next, the influence of stitching capacitors on the characteristic impedance and the emissions is examined. In addition, the influence on coupling to the plane and the influence on the power plane itself is verified. For this examination 1 nF stitching capacitors are connected between the plane pair. Figure 18 shows the single-ended S-parameters of the inner differential pair in blue and green and the PPCC in yellow and red with and without stitching capacitors respectively on "VB 212".

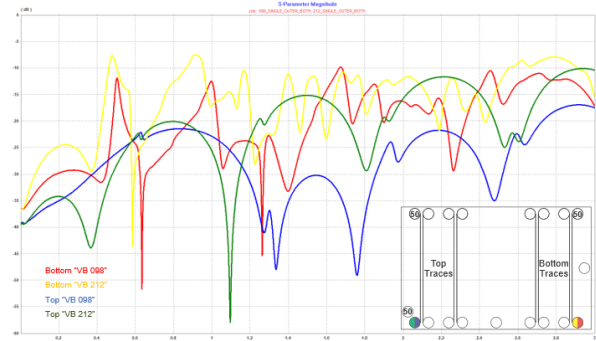


Figure 17. S-parameter plotting of outer single-ended traces.

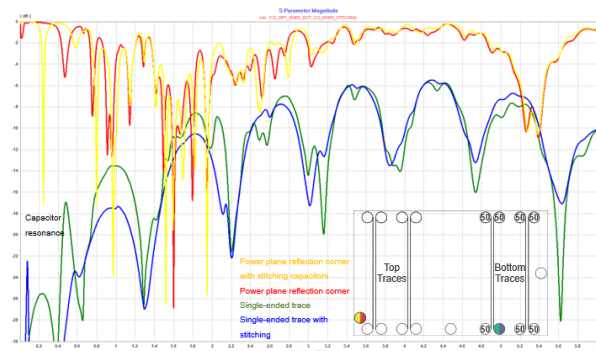


Figure 18. Reflection S-parameter of inner trace and PPCC of "VB 212".

The selected stitching capacitors alter the reflection parameters of the trace being significantly more homogeneous. The stitching capacitors also change the resonances on the power plane.

The mixed-mode S-parameters show a different situation: In differential mode - as the traces reference each other - stitching capacitors do not lead to any significant improvement in the S-parameters. In common mode stitching capacitors have a positive influence.

Another interesting point is the change in the coupling to the power plane due to the stitching capacitors. Figure 19 shows the mixed-mode transmission parameters of the inner differential pair with and without stitching capacitors for "VB 212".

"VB 212" is a good example of how the stitching capacitors also have a high influence on coupling to the power plane. Common mode coupling shown in red is reduced by a few dB up to 3 GHz by the use of stitching capacitors as shown in yellow. Differential mode coupling shown in blue is even reduced by 5 to 10 dB by the stitching capacitors shown in green. In conclusion, hypothesis **H9** is **valid**.

To proof **H10** decoupling capacitors are selected that have a low impedance for certain power plane resonances to dampen the resonances. These capacitors are placed near the apex of the standing wave.

Table II shows various footprint locations, which standing wave mode is intended to be prevented and

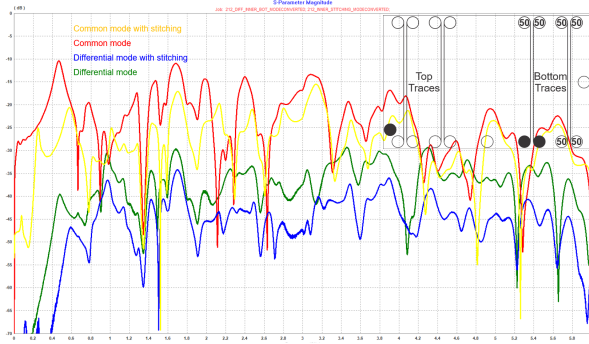


Figure 19. Mixed-mode S-parameter coupling inner differential pair to PPCC of "VB 212".

Table II
DECOUPLING CAPACITOR SELECTION

Footprint location	Wave mode	Cap.	P/N
Top upper left	f_{100} (500 MHz)	220 pF	GRM1555C1H221JA01
Top upper middle	f_{010} (800 MHz)	100 pF	GRM1555C1H101JZ01
Top upper right	f_{100} (500 MHz)	220 pF	GRM1555C1H221JA01
Top lower left	f_{100} (500 MHz)	220 pF	GRM1555C1H221JA01
Top lower middle	f_{010} (800 MHz)	100 pF	GRM1555C1H101JZ01
Top lower right	f_{100} (500 MHz)	220 pF	GRM1555C1H221JA01
Bottom upper left	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01
Bottom upper middle	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01
Bottom upper right	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01
Bottom lower left	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01
Bottom lower middle	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01
Bottom lower right	f_{210} (1250 MHz)	47 pF	GRM1555C1H470JZ01

which capacitor is applied. Capacitors were selected from the "GRM15" capacitor series with the lowest impedance for various wave modes. While the f_{100} (500 MHz) mode shows its maxima on the left and right side of the VB, this is the case with the f_{010} (800 MHz) mode at the upper and lower side of the VB. The f_{210} (1250 MHz) mode has several maxima on each edge.

Figure 20 shows the reflection parameters at the PPCC without decoupling capacitors in blue and green and with decoupling capacitors in yellow and red.

It is easy to see that the selected resonances at the corner connector are largely reduced. However, the capacitors generate additional resonances at around 200 MHz. The 220 pF capacitor generates the resonance at about 180 MHz, while the 100 pF capacitor generates resonances at about 230 and 280 MHz. Various parasitic inductances cause the shift between the two VBs. The thicker stack-up has a higher inductance, which shifts the resonances to lower frequencies compared to "VB 098".

To investigate whether coupling is also reduced Figure 21 shows the coupling between the single-ended traces and the PPCC.

Similar to Figure 20, the capacitors reduce the coupling, especially for f_{100} and f_{010} modes. The f_{210} mode shows a reduction of a few dB only, for the outer trace. For the inner trace the coupling even slightly increases. However, the capacitors increase

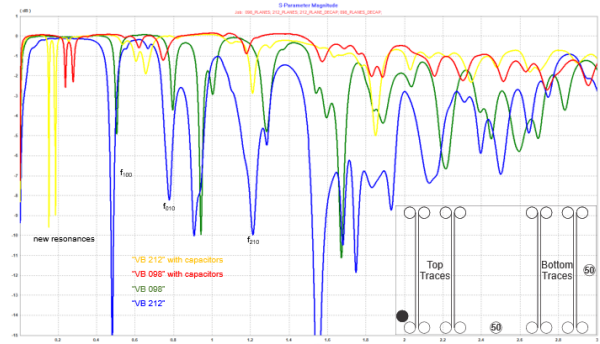


Figure 20. S-parameter PPCC with(out) capacitors.

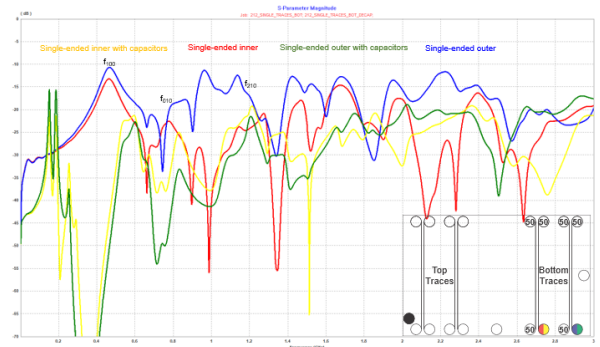


Figure 21. S-parameter bottom single-ended traces to PPCC of "VB 212".

coupling around 200 MHz. In summary, well-selected decoupling capacitors can locally damp resonances and reduce coupling. However, this only applies to a small frequency range and may cause new resonances.

Therefore, hypothesis **H10 cannot be proven** in general. A more accurate statement is:

Well-selected decoupling capacitors dampen certain local resonances and reduce coupling locally.

VII. CONCLUSION

Resonances on the power plane are unavoidable but can be strongly damped locally by using decoupling capacitors with the disadvantage that they may generate new resonances. Power planes should be kept relatively small in order to shift the resonances into a higher frequency range. A smaller distance between the power plane and the ground plane results in stronger damping and increases the capacitance. Differential signaling can significantly reduce coupling to the power plane in the range of a few GHz. Traces should always be routed via a low-impedance ground plane in order to minimize coupling to other copper structures and keeping the characteristic impedance constant. In addition, routing near the edge of the power plane should be avoided, as these are the locations of the maxima of the standing waves. Stitching capacitors can be used to reduce but not fully prevent a sudden

impedance change in case of a reference layer change or a gap in the reference layer. Similar to decoupling capacitors, they reduce coupling to a power plane.

ACKNOWLEDGEMENTS

The authors would thank Rudolf Romes and all members of the Renesas Hardware Section 1 Team for their continuous engagement and support.

REFERENCES

- [1] Eseye, "High Data Low Latency IoT Connectivity", <https://www.eseye.com/resources/blogs/high-data-low-latency-iot-connectivity/#:~:text=Typical%20use%20cases%20for%20high,as%20with%20an%20autonomous%20vehicle.> [Accessed 8 October 2024].
- [2] I-PEX, "What is PAMx (x=2,3,4...) Signaling Technology?", <https://www.i-pex.com/library/article/what-is-pam.> [Accessed 16 October 2024].
- [3] P. Jella, "Solving Signal & Power Issues in High-Speed Data Centers", Synopsys, <https://www.synopsys.com/blogs/chip-design/signal-and-power-integrity-tools.html>. [Accessed 25 September 2024].
- [4] H. Heuermann, "Hochfrequenztechnik", Springer Vieweg Wiesbaden, 2023.
- [5] Z. Peterson, "When Do PCB Power Plane Resonances Occur?", Altium, 11 January 2022. <https://resources.altium.com/p/when-do-pcb-power-plane-resonances-occur.> [Accessed 22 July 2024].
- [6] Cadence, "Power Plane Resonance Analysis in Your PCB", <https://resources.system-analysis.cadence.com/blog/msa2021-power-plane-resonance-analysis-in-your-pcb.> [Accessed 22 July 2024].
- [7] R. Thüringer, "AVT: HighTech-HighSpeed Baugruppen", https://homepages.thm.de/~hg7313/lehre/avt/skript/avt_hightechspeed_bg_kap3.pdf. [Accessed 16 July 2024].
- [8] Z-zero, "Etch Effects Explained", 18 February 2021, <https://www.z-zero.com/blog-post/etch-effects-explained/>. [Accessed 2024 July 2024].
- [9] Altium, "Differential Pair Routing", <https://www.altium.com/documentation/altium-designer/interactively-routing-differential-pairs-pcb.> [Accessed 16 July 2024].
- [10] L. Stiny, "Passive elektronische Bauelemente", Springer Vieweg Wiesbaden, 2019.
- [11] B. Deutschmann, N. Juch and G. Winkler, "Procedure for the Selection of Decoupling Capacitors", 2023, Joint Asia-Pacific International Symposium on Electromagnetic Compatibility and International Conference on ElectroMagnetic Interference & Compatibility (APEMC/INCEMIC), Bengaluru, 2023.
- [12] J. N. Tripathi, J. Mukherjee, P. R. Apte, N. K. Chhabra, R. K. Nagpal and R. Malik, "Selection and placement of decoupling capacitors in high speed systems", IEEE Electromagnetic Compatibility Magazine, vol. 2, pp. 72-78, 2013.
- [13] Texas Instruments Incorporated, "High-Speed Layout Guidelines for Signal Conditioners and USB Hubs", 2018.
- [14] C. K. Lun, Y. T. Guan, Y. C. Kheong, F. K. W. Lee, H. Yong and G. Vetharatnam, "Platform stitching capacitors impact to high-speed differential links on non-ideal return path", in 1st Asia Symposium on Quality Electronic Design, Kuala Lumpur, 2009.
- [15] L. Ritchey, "Crosstalk or Coupling", Altium, <https://resources.altium.com/p/crosstalk-or-coupling>. [Accessed 18 July 2024].
- [16] H. W. Ott, "Electromagnetic Compatibility Engineering", Wiley & Sons, 2009.
- [17] LearnEMC, "Practical EM Shielding", <https://learnemc.com/practical-em-shielding.> [Accessed 18 July 2024].
- [18] A. Payne, "SHIELDING OF MAGNETIC FIELDS BY EDDY CURRENTS", 2016. https://www.researchgate.net/profile/Alan-Payne-3/publication/343772602_SHIELDING_OF_MAGNETIC_FIELDS_BY_EDDY_CURRENTS/links/6050b20292851cd8ce4810ac/SHIELDING-OF-MAGNETIC-FIELDS-BY-EDDY-CURRENTS.pdf?__cf_chl_tk=NAF0WmyyQsrdZC5KLCmimiVHzWB_

mYZ9kZIF_oefvA-1734338467-1.0.1.1-zT6F0I9p3LlVl47igR5fUYMrKglGU9wnOmuU1SBTU28. [Accessed 16 December 2024].

- [19] B. Olney, "Plane Cavity Resonance", The PCB Design Magazine, pp. 46-50, September 2017.
- [20] Isola DE104, "Dielectric Constant(Dk)/Dissipation factor (Df) Table" 30 March 2021, https://www.isola-group.com/wp-content/uploads/data-sheets/de104-laminate-and-prepreg_Dk_Df_Tables.pdf?t=2079665717. [Accessed 24 July 2023].



degree in 2024 he started working full time in this field.



early prototyping systems using FPGAs in large scales. Besides designing, he also has the focus on verification and simulation, electrically and functionally. For completed designs he implements automated tests and accompanies manufacturing processes.



sign and electronic design automation of integrated digital circuits.

Justus Gottwald was born in Lübeck, Germany in 1999. In 2018 he graduated from high school and then started his dual studies bachelor at Renesas Electronics Europe GmbH in cooperation with University of Applied Sciences in Düsseldorf. After finishing the bachelor's degree in Microelectronics, he continued with the master's degree in 2022. In this time, he started working part time as a hardware engineer at Renesas. After completing his master's degree in 2024 he started working full time in this field.

Michael Engelbrecht was born in Nashua, NH, USA, in 1976 living in Germany since shortly after. He received the Dipl.-Ing. degree in electrical engineering and information technology from the Technical University of Darmstadt, Germany, in 2001. In 2001, he joined NEC Electronics in Düsseldorf, Germany merging into Renesas Electronics in 2010. He started designing digital circuits for MCUs. Later he also modeled circuits and developed

Bernhard Rieß was born in Würzburg, Germany in 1966. He received the Dipl.-Ing. degree in Electrical Engineering and Information Technology from the Technical University of Munich, Germany in 1992 and his Dr.-Ing. degree in 1996. In 1997, he joined Infineon Technologies in Munich, Germany. Since 2012, he has been a Professor of Microelectronics at the University of Applied Sciences Düsseldorf, Germany. His research interests include design and electronic design automation of integrated digital circuits.

Evaluation of a new AI driven SoC Design Flow Approach

Thierry Kersten, Thomas Haase, Bernhard M. Riess

Abstract—Cadence Design Systems recently introduced a tool, called “Cerebrus”, that utilizes ML algorithms to find good P&R solutions with significant less manual iterations. The physical design flow was set up for nine benchmark designs with various design sizes and instance counts up to 2.6 M to evaluate Cerebrus’ feasibility and benefits with respect to floorplan shrinking, macro placement, reduction of DRC violations, and power optimization. Floorplan shrinking reduced the chip area up to 23%. The automatic macro placer was able to create a macro placement with implementation characteristics comparable to the manual macro placement. Routing design rule failures were reduced up to 96%. Power optimization resulted in improvements up to 2.3%. Overall, the tool outperforms the state-of-the-art implementation flow by exploring specific issues in less engineering time which results in improved engineers’ efficiency and/or reduced time to market.

Index Terms—Artificial Intelligence, Machine Learning, SoC Design, Place&Route, Area, Performance, Power

I. MOTIVATION

The development of semiconductor devices is a highly complex and challenging task consisting of many design steps involving various engineering teams. Over time, the complexity of semiconductor systems increased so much, that manual development is no longer feasible. Automated design tools were developed to assist engineers during the design process. During this process, it is mandatory to follow many specifications and requirements to achieve high final design quality. Primary targets are power, performance and area (PPA) that must be monitored and optimized in the implementation process [1]. With the growing demand for more efficient, faster, and smaller devices, optimizing PPA has become more critical than ever. In physical implementation many iterations are needed to get good PPA, as automated design tools are only capable of automating individual steps. However, the expertise in only one single development step does not guarantee good results in later physical implementation steps and a high final design quality. Bad PPA results in final flow steps often require fundamental changes

in previous steps resulting in many time consuming iterations.

Recent advances in artificial intelligence (AI) and machine learning (ML) simplify complex tasks and assist in various areas of technology. ML algorithms can analyze large datasets to identify patterns and relationships. Consequently, ML algorithms were utilized to support physical implementation [2] [3]. Cadence is a major electronic design automation (EDA) vendor who recently introduced a tool, called “Cerebrus”. It combines physical implementation tools with ML algorithms with the aim of identifying good PPA solutions without many manual iterations. Target of this work is to evaluate whether this promise is kept.

A brief introduction into semiconductor integrated circuits and fundamentals on the development flow is given in section II. Basic information on Cadence Cerebrus is provided in section III. The used benchmark designs and the applied cost function are shown in section IV. Finally, the abilities of Cerebrus are evaluated and the obtained results are presented and discussed in section V.

II. FUNDAMENTALS

A. Design Flow Goals

The goal of the design flow is to get an implemented design that meets all functional requirements defined in the specification and all physical requirements mandatory for production. Every design step is crucial and can influence characteristics and quality of the design. A crucial parameter to visualize the design’s characteristics is the PPA metric. PPA are three parameters that are strongly connected and can indicate whether the design meets the criteria or not [1]. Although minimum area is a primary design target, if the area of the design is too small, the power density and thus the heat flux may be too high. Consequently, either the package of the design must be adjusted for better heat resistance and better thermal management, or the power density must be lowered by reducing power dissipation [4]. This can be accomplished by changing the type of standard cells to lower their power consumption, which may lead to higher cell delays and thus, lower performance [1]. If the design needs to meet certain power density requirements but must be high performant, the area of the design may need to be increased. A bigger chip area results in

Thierry Kersten, thierry.kersten.yw@renesas.com, Thomas Haase, thomas.haase@renesas.com, Renesas Electronics Europe GmbH, Düsseldorf, Germany.

Bernhard M. Rieß, bernhard.riess@hs-duesseldorf.de, University of Applied Sciences Düsseldorf, Germany.

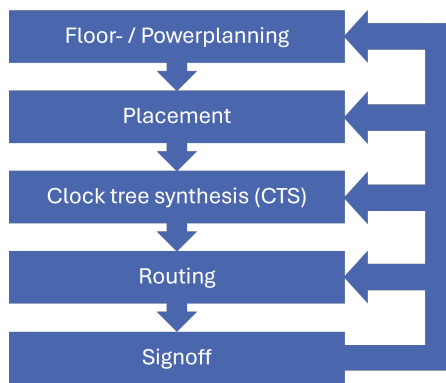


Figure 1. Physical Design Flow.

higher production cost and lowers the yield which will increase the overall cost [1] [4]. This shows that it is often not possible to optimize all PPA parameters at the same time.

B. Physical Layout Flow Stages

The individual steps of physical design flow are shown in Figure 1 [5] [6] [7] [8] and briefly summarized in the following.

1) *Floor- and Powerplanning*: Floorplanning defines the basic setup for the chip implementation task: The die size is specified, locations of external pads/ports are assigned, and IO (Input/Output) related macros are preplaced. For hierarchical designs (typically SoCs), design planning (size, shape, and location of sub-partitions) is also performed during floorplanning [6].

A die typically falls in one of the two following categories: “pad-limited” which describes a floorplan limitation by its IO pads, or “core-limited” which means the die size depends on the core/logic area [1].

IO placements are normally determined by the targeted PCB and package design. IO related macros must be preplaced in the floorplan. Macros are pre-existing crucial blocks like intellectual properties (IPs) or memory blocks. Connections between the IOs and its macros sometimes lead to conflicts either in the timing or the physical domain which may only get solved by reconsideration of the IO planning increasing development time until the floorplan gets settled [1].

The circuit must be analyzed for setup and hold timing, as well as power across various process corners and functional modes. This is typically called “Multi-Mode-Multi-Corner” (MMMC) setup [9], where various analysis views hold information on corner (voltage, temperature, process) and timing mode.

2) *Placement | Prects*: In the placement or prects (pre clock tree synthesis) flow stage, the optimal location for all standard cells of the netlist is determined and the netlist is further optimized to the physical requirements to be able to meet timing and power requirements [8]. Placement is traditionally split in

global placement that searches for global optimal cell locations and detailed placement that fine-tunes the cell locations to legal positions [5].

3) *Clock tree synthesis*: After placement is finished, the clock tree must be built. I.e. a clock distribution network must be generated in order to connect and balance all sequential design elements such as flipflops and clock gating cells to their particular clock source. Clock skew is an important metric and describes the clock delay difference between a launching element and a corresponding capture element [12]. A minimum skew is mandatory to avoid excessive hold violations and hold fixing, but also helps for setup timing closure.

4) *Routing*: Routing is the process of connecting the signal pins of the standard cells and the predefined macros according to the netlist by wire segments through available metal layers. Since this is a very complex step with various algorithms involved, it is divided into global routing and detailed routing. In higher complex designs a third timing-driven routing step may be performed [5].

5) *Signoff*: To check whether the design meets all timing requirements a static timing analysis (STA) is performed. Setup and hold describe the timing requirements to be fulfilled for a signal to be captured by a flipflop. This signal needs to be stable for a specific “setup” time before and a specific “hold” time after the clock edge arrives at the capture flipflop [1].

This analysis will focus primarily on the steps placement, cts, and routing which comprise the main implementation tasks.

III. ARTIFICIAL INTELLIGENCE IN ELECTRONIC DESIGN AUTOMATION

A. Artificial Intelligence and Machine Learning

AI is a development style of computers to mimic human capabilities [10]. These can emulate human thoughts to perform tasks in which learning is crucial. AI describes a general meaning of how computers perform and has many subfields. One of them is machine learning (ML). It describes algorithms to let the system adapt to big data and predict outcomes. ML has rapidly developed over the last two decades and has a main standing in today’s data processing [11].

ML algorithms can improve and speed-up VLSI design [2] and EDA applications [3] by offering many design prediction and optimization techniques. For quality of results (QoR) improvement, a model should predict the outcome of a flow step, choose the best candidate out of several runs, estimate and guide the flow. Design prediction can improve efficiency without the need to start multiple parallel runs by hand. For QoR prediction many different ML models can be used. Those models are Graph Neural Networks (GNNs), Long Short-Term Memory (LSTM) networks, or Reinforcement Learning (RL). GNNs work well on data that can be represented as a graph such as circuit

netlists. LSTMs have the ability to memorize data of recurrent neural networks and thus address long-term dependencies. RL models work on trial-and-error basis. An agent uses the outcome of previous runs, models the experience and then can predict the outcome of following runs.

B. Cadence Cerebrus

Cerebrus has been recently introduced by Cadence with the intent to improve P&R flows applying Machine Learning (ML) algorithms [12] [13] [14]. It comes with several sub-applications to improve several steps of the design flow claiming to boost productivity and to reduce time to market. It uses model-based RL, in which several parallel runs are started. Their results/metrics are compared to the metrics of a base run. Once a model is learned for a specific design, this model can be reused for subsequent tool runs to improve turn around time (TAT).

For each design, several PPA goals can be defined. Cerebrus searches optimization strategies via defined primitives. These primitives hold tool setting modifications, which get assigned to each scenario. A scenario is a new run started by Cerebrus. Modifying these settings in each scenario, Cerebrus tries to find design suited primitive input values to teach the model. Scenarios with better results compared to the base run are considered good values for this primitive. There are many product-defined primitives, whose contents are not visible to the user and therefore cannot be modified manually, but only reused within the Cerebrus environment.

Cerebrus comes with a few applications helping the user to improve specific design goals, by activating application-specific primitive functions:

The **floorplan optimizer** has two major features: **Floorplan resizing** and **macro placement**. It can be used upfront or standalone within the full Cerebrus flow. It can be applied to find alternative, perhaps better suited, floorplans by resizing, modifying the area or changing the aspect ratio. Updated floorplan dimensions require to adapt the macro placement. Therefore, macro placer capabilities are integrated. It is also possible to apply the macro placer only, by disabling floorplan resizing.

Standard cells are typically available in a lot of different flavors like voltage threshold (VT) types and driver strengths. VT has a big impact on delay and power consumption of the standard cells. Higher VTs result in slower signal delays but reduced power consumption. The **VT optimizer** explores various VT settings throughout the design to reduce power consumption.

Cerebrus will be evaluated with various existing designs explained in section IV. The focus will be on floorplan resizing, macro placement, and PPA improvements with the full P&R flow. For power optimizations

Table I
BENCHMARK DESIGNS (*DIGITAL **ANALOG RING)

Design	#Inst	RAMs Macros	Size (S/M/L)	TAT	Type	Process Node
1	18 k	0	S	41 min	D*	180 nm
2	17 k	0	S	76 min	D*	180 nm
3	22 k	0	S	73 min	D*	180 nm
4	1.77 M	256	L	55 h	D*	7 nm
5	676 k	2	M	10 h	D* w/ AR**	55 nm
6	642 k	2	M	9.5 h	D* w/ AR**	55 nm
7	431 k	2	M	6.6 h	D* w/ AR**	55 nm
8	2.66 M	233	L	36 h	Full SoC	40 nm
9	10 k	0	S	1 h	D*	180 nm

the VT optimizer will be tested. It will be evaluated whether these applications can improve the efficiency of finding new and better P&R solutions.

IV. CEREBRUS FLOW SETUP

This chapter presents benchmark designs used to evaluate the relevant Cerebrus features. For that, the complete flow environment and scripting must be set up. Furthermore, first testing on a few designs is made to adjust scripting and optimize tool settings.

A. Benchmark Designs

Nine designs in four process technologies are used to evaluate Cerebrus. All designs, despite one, were already implemented previously and the obtained results are used as a reference for evaluation. Not all tool features could be evaluated on all designs. Instead, for each design, one or several evaluation purposes are defined. For example, macro placement of the floorplan optimizer can only be used on designs that contain macros. Floorplan shrinking only makes sense on designs having no IO cell limitation. Table I gives an overview on all benchmark designs.

Blocks 1, 2, and 3 are very small, have short run times and thus are good candidates to evaluate floorplan shrinking and the full flow.

Block 4 is a big block with 256 macros contained suffering from long runtimes and high resource requirements with respect to memory and CPU. It is used for automatic macro placement and power optimization techniques. The base run utilizes a manually created macro placement.

Blocks 5, 6, and 7 have a very similar structure. Their digital core logic is surrounded by an analog padding. Thus, floorplan shrinking is not an option. The original implementations had to deal with routing problems due to high density in some parts of the core. These blocks will be used for full flow Cerebrus evaluation with focus on routing improvements while maintaining PPA. Also, VT optimization will be tested on block 7.

Block 8 is a 40 nm full SoC. It features an IO ring and several fixed macro locations. It will be used

Table II
BENCHMARK DESIGNS AND EVALUATED FEATURES

Design	Floorplan Shrink	Macro Placer	Full Flow DRC Reduction	Power (VT) optimization
1	X	-	X	-
2	X	-	X	-
3	X	-	X	-
4	-	X	-	X
5	-	-	X	-
6	-	-	X	-
7	-	-	X	X
8	-	X	-	-
9	-	-	X	-

Table III
DEFAULT COST FUNCTION FROM THE CEREBRUS USER GUIDE [12]

Constraint	ideal_value	ideal_base_percent	normalize_base_percent
timing.setup.wns.cerebrus	0	-	0.1
timing.setup.tns.cerebrus	0	-	0.05
power.switching	-	0.8	0.01
power.leakage	-	0.5	0.05
power.internal	-	0.8	0.01
design.congestion.hotspot.total	500	0.5	0.1
design.density	-	0.9	0.005

to evaluate automatic macro placement, while a few macros must be fixed and are not allowed to be moved.

Block 9 features a challenging floorplan with high cell density and routability issues.

Each design is associated to one ore more applications or features of Cerebrus to be tested according to Table II.

B. Initial Cost Function

An initial test of the Cerebrus flow is executed with block 3. The default cost function as recommended in the Cerebrus user guide described in Table III is used.

Each metric or constraint that is considered in the cost function gets an “ideal_value” or an “ideal_base_percent” value. Once the metric is lower (or above, depending on the metric) the ideal_value, its cost is considered to be zero. The ideal_base_percent value transforms the metric result of the base run per multiplication into an ideal_value. For example, if the power.leakage value of the base run is 100 mW, the ideal_value of this metric would be 50 mW if the ideal_base_percent is set to 0.5. If both, ideal_value and ideal_base_percent are defined (as in design.congestion.hotspot.total), the higher ideal_value will be used. The normalize_base_percent value specifies a normalization factor. Any change would be considered significant if it is above this factor. E.g. if the design.congestion.hotspot.total value is 1000 in the base run, a change of less than 100 in the generated scenario will be considered as insignificant by Cerebrus. The total cost value of these metrics is calculated by a hidden formula within Cerebrus and is therefore incomprehensible.

QoR Summary

Filter Metrics

Run	base	scenario_1	scenario_46	scenario_71	scenario_73	scenario_3	scenario_11	scenario_26	scenario_29
Cerebrus Cost	16	8	14	7	7	8	15	9	10
WNS.1 (ns)	0.000	-1.006	-0.373	-0.986	-0.696	-1.149	-0.656	-1.275	-1.682
WNS.100 (ns)	0.000	-0.225	-0.241	-0.224	-0.217	-0.330	-0.244	-0.346	-0.505
TNS (ns)	0.000	-18.258	-1.687	-4.027	-7.175	-14.501	-4.396	-11.085	-20.690
Hold TNS (ns)	-0.314	-0.067	-0.003	0.000	0.000	-0.463	-0.141	-0.148	-0.248
Setup WNS (ns)	-2.299	-3.360	-2.890	-3.144	-2.939	-3.128	-2.700	-2.965	-2.690
TNS (ns)	-4	-32	-12	-15	-16	-21	-9	-17	-26
FEPS	5	112	41	48	54	56	24	42	54
Hold WNS (ns)	-0.314	-0.059	-0.034	-0.230	-0.025	-0.157	-0.138	-0.145	-0.168
TNS (ns)	-0	-0	-0	-0	-0	-0	-0	-0	-0
FEPS	1	16	3	4	2	28	8	5	18
Power Leakage (mW)	1.14	1.11	1.16	1.11	1.11	1.06	1.08	1.08	1.08
Total (mW)	18.19	17.45	17.63	17.67	17.69	17.38	17.47	17.42	17.45
Congestion Cerebrus Total Hotspot	3925.11	291.00	3801.00	405.00	383.00	476.11	3929.67	410.44	479.67
Design Core Area	1181864	1063165	1063165	1063165	1063165	1152702	1152702	1152702	1152702
Runtime Wall (s)	1:31:52	1:58:24	2:13:38	1:59:10	1:54:40	2:00:11	1:50:57	1:48:29	2:03:15
Cpu (s)	0:00:22	0:00:22	0:00:23	0:00:23	0:00:23	0:00:22	0:00:22	0:00:23	0:00:23

Figure 2. QoR summary of block 3.

The post routing QoR summary of block 3, after a floorplan shrinking run in Cerebrus can be found in Figure 2. The third column shown in yellow gives the results of the base run, while the columns to the right of column three give the results of alternative scenarios generated by Cerebrus.

The cost values are relatively small and several scenarios have significant lower cost compared to the base run despite having a much worse setup timing. From Figure 2 remains unclear, why the TNS (total negative slack) value in the category “Cerebrus” is much lower than the TNS value in the “Setup” category. The reason is that pathgroups must be specified. The Cerebrus TNS and WNS (worst negative slack) values consider only the pathgroups defined in the config file, although the default TNS/WNS in setup considers all pathgroups. Cerebrus uses the average WNS timing of the first worst 100 violated paths in the design, whereas WNS.1 uses the “real” WNS (the first violated path). Cadence argues, that a single WNS path can be “too noisy and does not quite often represent the state of the design” [12].

C. Cost Function Adjustment

To overcome misleading cost values reported by Cerebrus we modified the cost function: The Cerebrus metric values are replaced by the standard Innovus metric values. Innovus is the standard Cadence P&R tool invoked by Cerebrus. All Cerebrus built-in metrics get disabled by setting their weight to 0. Exceptions may be Cerebrus application (floorplan optimizer or VT optimizer) specific metrics. The weights of all specified costs are adapted according to Table IV. The normalize_base_percent value for setup and hold are, in contrast to leakage and density, adapted to the default of 0.1 to use a weight of 1.

Table IV
COST WEIGHT FUNCTION TEMPLATE

Metric	place	cts	route
timing.setup.wns	1	1	1
timing.setup.tns	1	1	1
timing.hold.wns	-	1	1
timing.hold.tns	-	1	1
power.leakage	0.5	0.5	0.5
design.congestion.hotspot.total	1	1	1
design.congestion.hotspot.max	1	1	1
route.drc	-	1	1
route.short.cerebrus	-	0	0
route.overflow.horizontal	1	1	-
route.overflow.vertical	1	1	-
design.density	0.5	0.5	0.5

Table V
INITIAL FLOORPLAN DATA FOR BLOCKS 1 - 3

Design	Shape	Width	Height	Area
1	L-Shape	1800 μm	810 μm	11.327 mm ²
2	L-Shape	1850 μm	860 μm	12.654 mm ²
3	Rectangle	1750 μm	680 μm	11.818 mm ²

V. EVALUATION AND EXPERIMENTAL RESULTS

The setup elaborated in section IV was adapted to each application and design and the results generated by Cerebrus are evaluated and analyzed. For each block and evaluation the resource requirements are noted. The Cerebrus manager jobs are similar across all designs and use a maximum of ~ 1.5 GB.

A. Floorplan shrinker

To evaluate Cerebrus quality to shrink a floorplan blocks 1, 2, and 3 are used. The initial floorplan data of these blocks is given in Table V.

An L-Shape describes a layout that is arranged in the shape of the letter “L”. Floorplan shrinking is allowed in both x- and y-direction. Allowed floorplan shrinking percentages are in the range from 2.5% to 30%. The cost weight function is set equally. Therefore, all weights for the metrics explained in subsection IV-C are set to 1. The routing DRCs get a cost weight of 2, since routing resources are more crucial while shrinking the design. However, the routing overflow values (used during prects and cts) cannot be improved by decreasing the floorplan area and will thus artificially increase the cost values of shrunken scenarios. To evaluate whether the weight of these two metrics should be kept or removed from the function, two sequential Cerebrus runs are executed. The first run is executed with the settings mentioned above, the second run with disabled routing overflow cost weight.

Block 1:

In the first run, Cerebrus created a total of 135 scenarios. Promising scenarios in a particular flow step are pushed down to the next flow step. There, one scenario can be the base for one or even more scenarios. Scenarios with results that are not promising are deleted by

Table VI
FIRST FLOORPLAN SHRINK RUN FOR BLOCK 1

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
base	69	-15.7 ns	0 ns	-6.7 ns	0 ns	0	-
124	47	-3.4 ns	0 ns	-3.8 ns	0 ns	0	5.8
129	48	-4.2 ns	0 ns	-4.6 ns	0 ns	0	12

Table VII
SECOND FLOORPLAN SHRINK RUN FOR BLOCK 1

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
120	42	-4.2 ns	-52 ps	-2.6 ns	0 ns	0	23
70	43	-3.7 ns	-5 ps	-2.1 ns	0 ns	0	23

Cerebrus. 66 placement scenarios were executed from which 24 were deleted from the model. The cts step had 52 scenarios, 18 were deleted. 55 routing databases were created. To evaluate which floorplan resizings are feasible, only the results of the routing step are considered. The relevant metric results for the base run, the best cost scenario and the best core area scenario are shown in Table VI.

The lowest cost was generated in scenario 124 with 5.8% shrinking. The lowest core area shrink gained is 12%. Although the overall results of scenario 129 are slightly worse than the best result, it can be considered as reasonable.

The second run again explored 55 routing scenarios out of a total of 141 scenarios. The best two scenarios are shown in Table VII.

Without considering the routing overflow values in the cost function, Cerebrus explored higher shrinking values. The best scenario (120) offers a floorplan shrink of 23%. Several additional good scenarios (e.g. 70) with 23% shrinking were found. This shows that the cost function must be adapted to consider the route overflow for every design. Thus, we follow this procedure and perform two subsequent Cerebrus runs for all following blocks. The overall average resource usage for placement, cts and routing is 5.7 GB, 33 min on 4 CPUs for a total of 174 scenarios. Executing all scenarios sequentially the total runtime would be 30 hours. On a compute farm, 12 parallel jobs could be utilized and a total runtime of 4.5 hours was achieved.

Block 2:

The relevant metric values for block 2 can be found in Table VIII.

Cerebrus generated 146 scenarios and pushed 54 of them to the routing stage. The run with route overflow involved in the cost function did not achieve any shrinking in the best cost scenario (42). The maximum shrinking explored by Cerebrus is 12% (scenario 28). The metrics of these scenarios show no challenges in setup timing and routing resources. Nevertheless, there is one very high IO hold timing violation of -925 ps.

Table VIII
FIRST FLOORPLAN SHRINK RUN FOR BLOCK 2

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
base	73	0 ns	0 ns	-77 ps	0 ns	2	-
42	49	0 ns	0 ns	0 ns	0 ns	1	0
28	185	-1 ps	0 ns	-925 ps	0 ns	0	12

Table IX
SECOND FLOORPLAN SHRINK RUN FOR BLOCK 2

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
98	49	0 ns	0 ns	0 ns	0 ns	1	0
44	185	0 ns	0 ns	-894 ps	0 ns	0	14

Since IO hold timing is fully clean in the best run with no shrinking, it needs to be manually analyzed, whether this scenario is feasible.

The second run generated 137 scenarios, 55 of them were pushed to the routing stage. The comparison can be found in Table IX with scenario 98 being the best cost scenario and scenario 44 being the best scenario with highest shrinking.

Again, this run did not yield any shrinking in scenario 98. The highest shrinking is 14% in scenario 44 again resulting in one high IO hold timing violation (-894 ps) while setup timing is clean and no routing violations occur. Shrinking to this percentage or maybe even more is only possible once the violations are understood and fixed within scripts or constraints. This block also resulted in higher shrinking values once route overflow was removed from the cost function. The overall average resource usage for placement, cts and routing is 4.6 GB, 70 min on 4 CPUs for a total of 184 scenarios. Executing all scenarios sequentially the total runtime would be 3 days. On a compute farm, 12 parallel jobs could be utilized and a total runtime of ~ 8 hours was achieved.

Block 3:

The first run explored several resize values with a total amount of 153 scenarios from which 55 got pushed down to routing. The results for the base scenario, the best cost scenario 145 and the scenario with highest shrinking (94) can be found in Table X.

The best scenario (145) gained no shrinking, but the second-best scenario (94) offered a moderate shrink of 2.5%. The base run and both scenarios have a high IO hold violation, which needs to be investigated and clarified manually.

The second run with removed routing overflow from the cost function confirms the behavior observed before: It pushed various scenarios with higher resize values into routing stage. The relevant results of the cost best scenario (128) and several reduced area scenarios (12, 140, 8) are shown in Table XI.

Table X
FIRST FLOORPLAN SHRINK RUN FOR BLOCK 3

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
base	100	-4.2 ns	0 ns	-0.9 ns	-1 ps	2	-
145	73	-3.5 ns	0 ns	-1.4 ns	-24 ps	0	0
94	79	-6.3 ns	0 ns	-1.5 ns	-27 ps	0	2.5

Table XI
SECOND FLOORPLAN SHRINK RUN FOR BLOCK 3

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC	Shrink %
128	84	-5.7 ns	0 ns	-1.3 ns	0 ns	1	0
12	177849	-52.8 ns	0 ns	-2.6 ns	-9.9 ns	$> 10^4$	27.5
140	617	-11.6 ns	0 ns	-3.8 ns	-23 ps	30	22.5
8	328	-20.5 ns	0 ns	-0.9 ns	-8 ps	13	17.5

The highest shrink gained is 27.5%. As can be seen in Table XI, this high shrink results in a very high cost induced by a very high route DRC count. Moreover, high timing violations arise. These resize values can not be utilized, despite being considered by Cerebrus. The lowest-cost scenario did not yield any shrinking. The lowest non-zero shrink value is 17.5% and the corresponding scenario 8 has the forth lowest cost among all others. Although the routing DRC count is reasonable small, the IO setup timing got much worse (-4.2 ns TNS to -20.5 ns TNS). It seems that IO timing is a crucial factor of this design and whether it is limiting resizing or not it must be reviewed. In scenario 140 a floorplan was found with 22.5% shrinking. This scenario has smaller setup IO violations (-11.6 ns TNS) but increased IO hold violations (-0.9 ps TNS to -3.8 ns TNS). It must be clarified whether these violations can be fixed to allow this high shrinking. The overall average resource usage for placement, cts and routing is 4.6 GB, 68 min on 4 CPUs for a total of 184 scenarios. Executing all scenarios sequentially the total runtime would be 3 days. On a compute farm, 12 parallel jobs could be utilized and a total runtime of ~ 11 hours was achieved.

B. Automatic Macro placer

The macro placer feature is included in the floorplan optimizer, since a rearrangement of the macros is necessary once the floorplan has been changed in size. It is also possible to skip shrinking and only explore the macro placement feature by setting the shrink target to zero.

Block 4:

The manually created macro placement for block 4 is used as reference for comparison and can be found in Figure 3. The engineering effort to figure out this macro placement was two to three weeks.

Cerebrus is applied to explore macro placements automatically. The cost function was updated to include

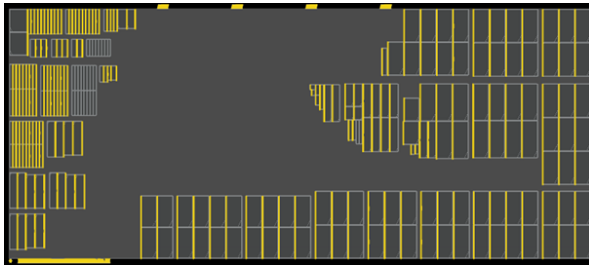


Figure 3. Manual macro placement of block 4.

Table XII
FIRST MACRO PLACER RUN FOR BLOCK 4

Scenario	Cost	Setup TNS	Hold TNS	Route DRC
base	102	-19 ns	-217 ns	31
47	83	-6 ns	-309 ns	7
40	86	-7 ns	-313 ns	9

the power consumption of the macro instances and not only standard cell leakage. The first run found several scenarios with various macro placements and good cost values but increased hold time violations. The results of the best two scenarios are shown in Table XII.

The flow steps were executed without additional hold fixing optimization step. Since hold fixing adds more buffer cells to the design, the local cell density may increase and the explored macro placements can be insufficient. Therefore, this run is aborted after exploring 111 scenarios and rerun with activated additional hold time fixing. To reduce run time for promising scenarios, the RL model of the first run is reused to prevent the new exploration of primitive values. Once restarted with this RL model, Cerebrus explored 37 routing scenarios, 16 of them referring to one particular macro placement. This macro placement was the only placement found by Cerebrus, that offered good results and thus was further explored for better PPA. The base run with improved results due to hold time fixing and the best three scenarios are given in Table XIII. Obviously, the implemented hold fix was successful in reducing the hold violations significantly.

All three best scenarios shown in Table XIII were generated based on the same macro placement. All other identified macro placements resulted in higher routing cost. In comparison to the base run, this macro placement is a good initial start for implementation saving weeks of manual trials. Figure 4 shows the automatically generated macro placement.

Manual optimizations can further optimize the generated macro placement to be abutted to free up area for standard cells. The good results of this macro placement are surprising, since the state of the art procedure in manual macro placement is to arrange the macros near the chip boundary to gain maximum space in the center of the floorplan for standard cells.

Table XIII
SECOND MACRO PLACER RUN FOR BLOCK 4

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC
base	88	-15.8 ns	-0.4 ns	0 ns	-37.3 ns	41
75	69	-11.5 ns	-0.7 ns	0 ns	-26.9 ns	2
78	76	-26.7 ns	-0.1 ns	0 ns	-27.4 ns	3
28	81	-24.7 ns	-0.3 ns	0 ns	-27.8 ns	9

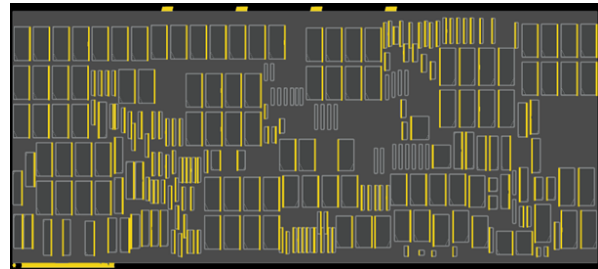


Figure 4. Automatically generated macro placement for block 4.

The first – aborted – run lasted for almost 20 days with six jobs in parallel. The overall average resource usage for the second run for placement, cts and routing is 105 GB, 54 hours on 16 CPUs for a total of 124 scenarios. Executing all scenarios sequentially the total runtime would be ~90 days. On a compute farm, 6 parallel jobs could be utilized and a total runtime of ~20 hours was achieved.

All exploration runs in total exceed the required two to three weeks exploration by hand. With the given RL model, the 20 days for the first run even exceed the time for manual exploration. The best scenario was generated after additional 14 days for the second run summing up to a total exploration time of 34 days. Still, the automatic exploration can save weeks of engineering resources.

Block 8:

The automatic macro placement exploration of block 8 faced various challenges. Since the RL model compares the scenarios to the base run, the base run needs to have reasonably good metrics. For block 8 it was only possible to get acceptable timing results with placement guides for crucial instance groups like the phased-locked loop (PLL), CPU, or instances that are connected to the IO ring. Several macros are not allowed to be moved and therefore need to be fixed. The user guide [12] recommends to set the placement status of those macros to “fixed” in the macro constraint file. Once the macro placement is generated, a new power mesh must be generated.

Starting from 26 scenarios pushed to routing Cerebrus finally identified only 3 routing scenarios with alternative macro placements. The best of them had mediocre quality and was continued to cts and routing, the other two were not promising and thus, deleted by Cerebrus. That shows that macro placement ex-

Table XIV
FIRST MACRO PLACER RUN FOR BLOCK 8

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC
base	88	-10.8 ns	-0.6 ns	-266.6 ns	-36.0 ns	163
89	60	-5.8 ns	-0.5 ns	-226.2 ns	-33.6 ns	146
52	61	-6.7 ns	-0.5 ns	-246.4 ns	-33.0 ns	162
103	172	-11.5 ns	-0.5 ns	-191.4 ns	-33.5 ns	663
125	180	-11.5 ns	-0.5 ns	-191.4 ns	-33.5 ns	700

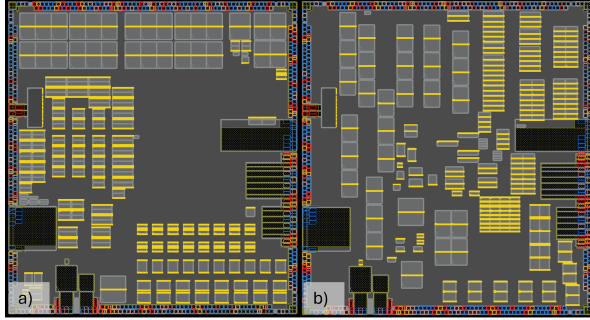


Figure 5. Comparison of the a) base and b) scenario 103/125 macro placement for block 8.

ploration is a particular difficult task. The best two scenarios (89 and 52) given in Table XIV base on the macro placement manually developed for the base run. Two additional scenarios (103 and 125) use the best macro placement automatically identified by Cerebrus.

The scenarios 103 and 125 have a significant higher cost, caused by significant higher route DRC values. It shows that the original manual macro placement and placement guidings from weeks of manual explorations cannot be outperformed by Cerebrus for this complex SoC. The comparison of the base macro placement with the best Cerebrus generated placement is shown in Figure 5.

The overall average resource usage for placement, cts and routing is 46 GB, 67 hours on 8 CPUs for a total of 126 scenarios. Executing all scenarios sequentially the total runtime would be ~148 days. On a compute farm, 12 parallel jobs could be utilized and a total runtime of ~18 days was achieved.

C. Full-Flow DRC Reduction

Primary intention of Cerebrus is to improve PPA over the full P&R flow. Moreover, this offers to focus on particular metric improvements.

The blocks 5, 6, and 7 have fix-sized digital core areas with high densities and thus routability issues. A high count of DRC violations needed to be fixed manually in the design phase of these blocks via several ECOs. Therefore, these blocks are perfect candidates to evaluate the general PPA improvement potential with focus on routing DRC reduction.

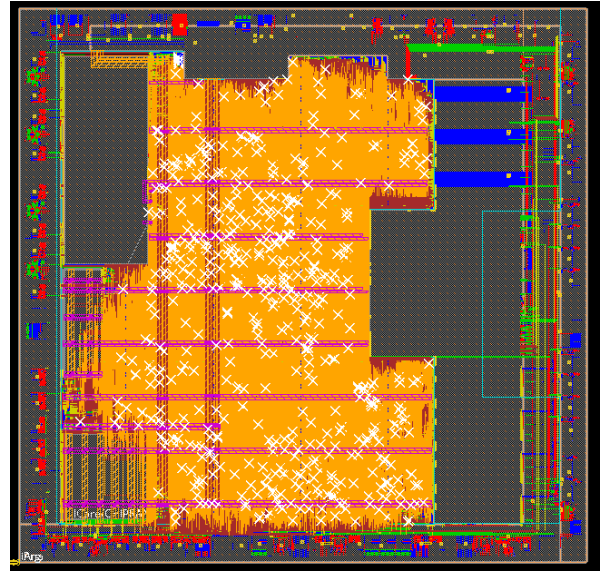


Figure 6. Block 7 with DRC markers in the digital core area.

Table XV
ROUTING EXPLORATION FOR BLOCK 7

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC
base	95	-0.3 ns	-64.7 ns	-1.7 ns	-0.4 ns	717
9	29	0	-67.3 ns	-160 ps	-97 ps	25
13	28	0	-63.5 ns	-161 ps	0 ns	32

Block 7:

The reference P&R result of this block is shown in Figure 6. It contains 717 routing DRCs and is not timing clean. Focus will be to reduce the number of DRC violations.

Cerebrus generated 104 scenarios, 55 were pushed down to route. The two best runs can be seen in Table XV.

Several scenarios were found with a significant reduction in routing violations down to 25 (scenario 9). The scenario (13) with the lowest cost (28) has better timing result as the base run with a minor increase in routing violations (32) compared to scenario 9.

A comparison of the placement areas of the instance groups in the digital core area of the base run and scenario 9 is shown in Figure 7.

The main instance groups, shown in various colors, were substantially rearranged in the core area. Although the specific flow adaptations by Cerebrus are not transparent, it emphasizes the possible effect of automatically evaluating many tool settings and their combinations which is hard to manage manually.

The overall average resource usage for placement, cts and routing is 17 GB, 6 hours on 8 CPUs for a total of 167 scenarios. Executing all scenarios sequentially the total runtime would be ~13 days. On a compute

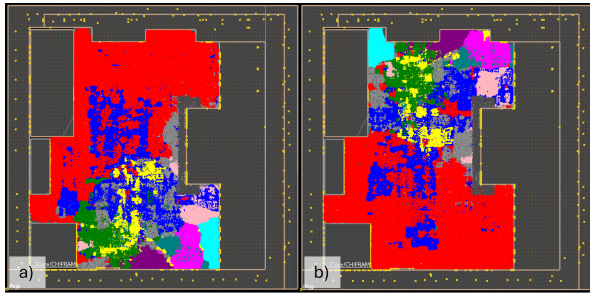


Figure 7. Comparison of instance group placement of a) base run and b) scenario 9.

Table XVI
ROUTING EXPLORATION FOR BLOCK 5

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC
base	93	-6.6 ns	-705 ns	-5.8 ns	0 ns	1459
82	46	-7 ns	-393 ns	-4.8 ns	0 ns	526
43	50	-7.2 ns	-194 ns	-5.1 ns	0 ns	601
51	51	-7 ns	-115 ns	-5.9 ns	0 ns	595

farm, 12 parallel jobs could be utilized and a total runtime of ~ 1.5 days was achieved.

Block 5:

The exploration of routing improvements for block 5 also showed major improvements – however, not that significant compared to block 7. The three best scenarios are displayed in Table XVI.

An issue with this design is the initial high internal setup TNS. It is not clear how this affects the routing. However, scenarios are found with significant routing improvements (up to 64%) additionally providing a significant reduction of the internal setup TNS.

The overall average resource usage for placement, cts and routing is 22 GB, 10.5 hours on 8 CPUs for a total of 166 scenarios. Executing all scenarios sequentially the total runtime would be ~ 26 days. On a compute farm, 12 parallel jobs could be utilized and a total runtime of ~ 3 days was achieved.

Block 6:

In block 6, similar to blocks 5 and 7, high routing improvements are observed. The summary including the best cost scenario (46) and the best DRC scenario (101) is displayed in Table XVII.

They show a maximum reduction from 755 down to 121 DRCs (84%) with the drawback of some added internal hold violations. Another scenario (81) with an improvement to 166 routing DRCs, but without significant internal setup and IO hold timing drawbacks could also be found. In comparison to block 7, this again shows how design dependent the results can be.

Table XVII
ROUTING EXPLORATION FOR BLOCK 6

Scenario	Cost	Setup TNS IO	Setup TNS Internal	Hold TNS IO	Hold TNS Internal	Route DRC
base	92	-27.7 ns	0 ns	0 ns	-21 ps	755
46	26	-13.5 ns	-0.5 ns	0 ns	0 ns	221
101	391	-11.8 ns	-0.2 ns	-9 ps	-0.8 ns	121
81	68	-14.6 ns	-78 ps	-23 ps	0 ns	166

The overall average resource usage for placement, cts and routing is 19 GB, 12 hours on 8 CPUs for a total of 164 scenarios. Executing all scenarios sequentially the total runtime would be ~ 29 days. On a compute farm, 12 parallel jobs could be utilized and a total runtime of ~ 3 days was achieved.

Block 9:

Similar to block 7, this block has a fixed floorplan for the digital logic and shows high routing densities. The floorplan size is crucial to the success of the design and thus it is important to make the design routable. The initial Cerebrus run provided a high reduction of the route DRC issues. Manual investigation revealed several large maximum transition violations and a missing optimization step after routing. After fixing this, a new Cerebrus run reduced the DRC violation count from 3402 down to 209. After an additional re-baselining run of this scenario, a further reduction down to 131 DRC violations was possible which results in a total reduction of $\sim 96\%$.

D. VT-Optimizer

The VT optimizer uses a two-run approach to optimize the power consumption of the design. First, only one primitive is activated. Cerebrus changes the VT distribution across the design by radically changing the VT of many standard cells. That changes the power consumption and timing results due to slower or faster cells being inferred. Once a scenario with improved leakage power consumption is found, this scenario is used in a second run for re-baselining and thus a second Cerebrus run.

This application was implemented on blocks 4 and 7. The cost function was updated to only include leakage power, timing and density. The density can change due to added or resized standard cells.

Block 7:

The QoR summary of block 7 can be found in Figure 8 and its VT distribution in Figure 9.

Cerebrus generated only six scenarios aiming to reduce leakage power consumption. The four best scenarios had a leakage improvement of 2.3% with slightly worse timing and constant density. Thus, the obtained results for this block show no significant power reduction. Additionally, a buggy behavior of the VT distribution table was found. Here, visible in

Run	base	scenario_5	scenario_1	scenario_2	scenario_3	scenario_6	scenario_4
Cerebrus Cost	7	7	7	7	7	7	10
Setup							
WNS (ns)	-6.779	-6.512	-6.512	-8.713	-8.713	-6.779	-7.036
TNS (ns)	-862	-11394	-11394	-557	-557	-860	-1824
FEPS	4372	24042	24042	3722	3722	4345	7009
Hold							
WNS (ns)	-0.323	-0.395	-0.395	-0.358	-0.358	-0.323	-0.371
TNS (ns)	-2	-3	-3	-2	-2	-2	-3
FEPS	31	36	36	22	22	23	26
Power							
Leakage (mW)	7.77	7.59	7.59	7.59	7.59	7.78	8.62
Total (mW)	83.70	82.52	82.52	82.66	82.66	83.69	84.77
Congestion Cerebrus Total Hotspot	3794.56	3783.28	3783.28	3782.68	3782.68	3780.16	3801.58
Density (%)	77.76	77.74	77.74	77.74	77.74	77.76	78.55
Runtime							
Wall (s)	8:02:44	8:16:44	7:47:01	8:31:30	8:02:01	7:51:37	8:12:24
Cpu (s)	0:00:25	0:00:25	0:00:24	0:00:25	0:00:26	0:00:26	0:00:25

Figure 8. QoR summary, VT optimization run for block 7, sorted by leakage values.

Inv (um^2)	base	scenario_2	scenario_3	scenario_5	scenario_1	scenario_6	scenario_4
Buf (um^2)							
Macro (um^2)							
Std Cell (um^2)	1233493	1233172	1233172	1233091	1233091	1233410	1245940
hvt (%)	95.9	95.9	95.9	95.9	95.9	95.9	92.2
lvt (%)	0.1	0.0	0.0	0.0	0.0	0.1	0.0
svt (%)	4.0	4.1	4.1	4.1	4.1	4.0	7.8

Figure 9. VT distribution in VT optimization run for block 7.

Figure 9 the lowering of lvt cells (which will lead to a lower power consumption) is marked red, despite showing a better result.

Block 4:

The VT optimization results for block 4 are shown in Figure 10. Cerebrus identified just one scenario which reduced leakage by 0.4%, with slightly worse timing and higher density.

VI. CONCLUSION

To overcome the highly iterative process of physical implementation of digital semiconductor devices, an AI-based solution from Cadence, “Cerebrus”, was evaluated on 9 industry designs. It allows the integration of ML algorithms, reinforcement learning, to adapt the flow and optimization techniques on designs individually without the need for time-consuming manual iterations. Cerebrus offers several applications to allow various kinds of explorations. To evaluate their functions and benefits, an independent scripting environment was established, and several benchmark designs were chosen. First, the floorplan optimizer with its shrinking function and macro placer function was evaluated. The floorplan shrinker was successful in shrinking floorplans up to 23%, the macro placer offered good initial macro placements for further investigation. Both functions also showed that the success of this application is highly design dependent. Especially the macro placer requires several technology specific settings and prerequisites. Next, routing improvements

Run	base	scenario_6	scenario_12	scenario_9	scenario_4	scenario_2	scenario_1	scenario_3	scenario_14	scenario_13	scenario_10	scenario_11	scenario_7	scenario_5	scenario_8
Cerebrus Cost	100	101	105	102	88	105	109	85	107	102	116	105	99	101	98
Setup															
WNS (ns)	-0.093	-0.122	-0.117	-0.086	-0.058	-0.103	-0.119	-0.031	-0.098	-0.104	-0.101	-0.092	-0.084	-0.091	-0.048
TNS (ns)	-19	-18	-20	-22	-13	-21	-21	-4	-22	-16	-32	-19	-34	-11	-14
FEPS	1751	1962	2104	2026	1783	2011	2184	1123	2261	1557	1846	2552	1586	754	2196
Hold															
WNS (ns)	-0.442	-0.405	-0.438	-0.442	-0.397	-0.438	-0.442	-0.442	-0.448	-0.448	-0.448	-0.447	-0.438	-0.456	-0.476
TNS (ns)	-217	-194	-209	-218	-204	-217	-208	-218	-248	-219	-253	-253	-237	-257	-246
FEPS	1661	1608	1732	1697	1449	1630	1635	1673	1461	1213	1444	1440	1176	1245	1607
Power															
Total (mW)	984.42	981.32	983.82	979.95	988.37	981.52	984.30	987.72	984.04	983.15	984.25	984.37	984.44	996.80	994.01
Switching (mW)	438.30	436.63	438.43	437.13	438.27	438.01	438.92	439.07	437.95	437.67	438.04	438.07	438.11	440.94	435.68
Leakage (mW)	80.79	80.46	80.81	80.94	81.36	81.47	81.57	81.68	82.61	82.62	83.34	83.38	83.54	83.79	81.14
Internal (mW)	465.34	464.23	464.57	461.88	468.75	462.04	463.72	466.97	463.48	462.86	462.87	462.92	462.78	472.08	467.20
Congestion															
Max Hot Spot	0.17	0.17	0.17	0.17	0.00	0.00	0.00	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
Total Hot Spot	0.17	0.17	0.17	0.17	0.00	0.00	0.00	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
Route Overflow Y (D)															
Route Overflow X (D)															
Route															
Total Route DRC	31	29	31	32	34	31	30	26	36	31	36	40	47	47	27
Total Route DRC cerebrus	46	39	32	34	36	23	41	46	43	39	46	43	123	54	32
Total Route Shorts	39	31	28	30	33	13	35	29	26	22	30	26	26	31	19
Density (D)	65.26	65.31	65.27	65.29	65.30	65.28	65.33	65.26	65.30	65.26	65.31	65.30	65.31	65.67	64.62
Design															
Core Area (um^2)	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565	1565
dx	672	672	672	672	672	672	672	672	672	672	672	672	672	672	672
dy	672	672	672	672	672	672	672	672	672	672	672	672	672	672	672
Runtime															
Wall (s)	34:27:55	35:38:36	40:31:19	40:40:03	35:52:13	36:49:34	38:29:22	34:21:18	37:47:56	39:20:20	36:26:47	37:54:52	38:52:13	35:58:08	34:43:52
Cpu (s)	0:00:27	0:00:30	0:00:28	0:00:28	0:00:28	0:00:27	0:00:27	0:00:27	0:00:27	0:00:28	0:00:28	0:00:28	0:00:28	0:00:27	0:00:29

Figure 10. QoR summary, VT optimization run of block 4, sorted by leakage values.

with the general PPA flow were evaluated. Those showed significant (up to 96%) reduction in routing DRCs across all tested designs. Furthermore, a VT optimizer was tested, showing minor improvements in leakage (2.3% and 0.4%) at a slightly higher overall cost. A possible drawback for a wide adoption of the evaluated approach may be the amount and cost of necessary computing resources. Small digital designs may be handled easily, whereas large designs or complex SoCs may end up blocking several compute machines for weeks. Overall, Cerebrus offers a good addition to the implementation flow by exploring specific issues in less time improving engineering efficiency.

REFERENCES

- [1] N. H. E. Weste and D. M. Harris, "CMOS VLSI Design a circuits and systems perspective", Boston: Addison Wesley, 2011. 4. ed., 2011.
- [2] I. M. Elfadel, D. S. Boning, and X. Li, "Machine Learning in VLSI Computer-Aided Design", Cham: Springer International Publishing, Imprint: Springer, 2019. <https://doi.org/10.1007/978-3-030-04666-8> (accessed Feb. 16, 2025)
- [3] H. Ren and J. Hu, "Machine Learning Applications in Electronic Design Automation", 1st ed. 2022. Cham: Springer International Publishing, Imprint: Springer, 2022. <https://doi.org/10.1007/978-3-031-13074-8> (accessed Feb. 16, 2025)
- [4] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, "Digital integrated circuits: a design perspective", 2. ed. Upper Saddle River, NJ: Prentice Hall, 2003.
- [5] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, "VLSI Physical Design: From Graph Partitioning to Timing Closure", 2nd ed. 2022. Cham: Springer International Publishing, Imprint: Springer, 2022. Available: <https://doi.org/10.1007/978-3-030-96415-3> (accessed Feb. 16, 2025)
- [6] V. S. Chakravarthi and S. R. Koteswarar, "SoC Physical Design: A Comprehensive Guide", 1st ed. 2022. Cham: Springer International Publishing, Imprint: Springer, 2022. Available: <https://doi.org/10.1007/978-3-030-98112-9> (accessed Feb. 16, 2025)
- [7] G. Herrmann and D. Müller, "ASIC - Entwurf und Test: mit 17 Tabellen", München: Fachbuchverl. Leipzig im Carl-Hanser-Verl, 2004.
- [8] D. Das, "Physical Design—Floorplanning, Placement, and Routing" in VLSI design. Oxford Univ. Press, 2015.
- [9] K. Golshan, "Multi-mode Multi-corner Analysis" in "The Art of Timing Closure", Springer, 2020. https://doi.org/10.1007/978-3-030-49636-4_2 (accessed Feb. 16, 2025)
- [10] "Artificial Intelligence (AI) vs. Machine Learning", Columbia engineering. <https://ai.engineering.columbia.edu/ai-vs-machine-learning/> (accessed Sep. 04, 2024)
- [11] J. P. S. Rosa, D. J. D. Guerra, N. C. G. Horta, R. M. F. Martins, and N. C. C. Lourenço, "Using Artificial Neural Networks for Analog Integrated Circuit Design Automation", 1st ed. 2020. Cham: Springer International Publishing, Imprint: Springer, 2020. <https://doi.org/10.1007/978-3-030-35743-6> (accessed Feb. 16, 2025)
- [12] Cadence, "Cerebrus User Guide", Product Version 23.11, April 2024
- [13] Cadence, "AI in Chip Design", https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/cerebrus-intelligent-chip-explorer.html (accessed Sep. 10, 2024)
- [14] Cadence, "Cerebrus, intelligent chip explorer", https://www.cadence.com/en_US/home/explore/ai-chip-design.html (accessed Sep. 10, 2024)



master's degree in 2024

Thierry Kersten was born in Heerlen, the Netherlands in 1999. In 2017 he graduated from high school. He started his dual studies bachelor at Renesas Electronics Germany GmbH in cooperation with University of Applied Sciences in Düsseldorf in 2018. After finishing the bachelor's degree in Microelectronics, he continued with the master's degree in 2022. In this time, he worked on physical implementation as an intern at Renesas. After completing his



Thomas Haase was born in Zwickau, Germany in 1966. He received his Dipl.-Ing. degree in Information Technology from the Technical University of Dresden, Germany in 1993. In 1995, he joined NEC in Düsseldorf, Germany. Since 2010, he has been managing a Design Service Department at Renesas Electronics in Düsseldorf, Germany. He is responsible for overseeing all aspects of design implementation for complex integrated digital circuits.



sign and electronic design automation of integrated digital circuits.

Bernhard Riess was born in Würzburg, Germany in 1966. He received the Dipl.-Ing. degree in Electrical Engineering and Information Technology from the Technical University of Munich, Germany in 1992 and his Dr.-Ing. degree in 1996. In 1997, he joined Infineon Technologies in Munich, Germany. Since 2012, he has been a Professor of Microelectronics at the University of Applied Sciences Düsseldorf, Germany. His research interests include design and electronic design automation of integrated digital circuits.

Eine Debug- und Testumgebung für AXI-Stream basierte Ethernet-Frame-Verarbeitung in FPGAs

Sebastian Haberkern, Marcus Würß, Stefan Eiermann

Zusammenfassung—Ein mögliches Interface zur Weiterleitung von Ethernet-Frames in FPGAs ist das AXI-Stream Interface. Es ermöglicht den Transport eines Ethernet-Frames als Byte-Stream auf einer Punkt-zu-Punkt Verbindung zwischen zwei Komponenten.

Das Testen und Analysieren solcher einzelnen Ethernet-Frame verarbeitenden Komponenten sowie von Teilen eines Gesamtsystems erfordert oft eine Interaktion mit einem oder mehreren weiteren Systemen. Dies steigert die Komplexität und den zeitlichen Aufwand für die Vorbereitung und Durchführung eines einzelnen Tests. In diesem Beitrag wird eine Möglichkeit vorgestellt, wie in einem speziellen Anwendungsfall, der Entwicklung eines Aggregations-IPs für OPC UA PubSub UADP Telegramme, Teile des Gesamtsystems im FPGA selbst emuliert werden könnten.

Hierzu wird im Blockdesign, welches Teil des Vivado® IP Integrator Tools der Vivado® Design Suite von Xilinx® ist, ein weiterer IP eingefügt, der anschließend zur Laufzeit über die Debug-Schnittstelle gesteuert werden kann. Dabei können sowohl ankommende Ethernet-Frames simuliert als auch ausgehende Ethernet-Frames analysiert werden. Neben einfachen Hardware-in-the-Loop-Tests ermöglicht dies außerdem, die Komponenten gezielt mit in der Realität nur aufwändig zu reproduzierenden Grenzfällen zu testen. Durch die Anbindung an die Debug-Schnittstelle können mit demselben Bitstream viele verschiedene Testfälle abgedeckt werden.

Schlüsselwörter—FPGA, MPSoC, Ethernet, AXI-Stream, Testing

I. MOTIVATION

Bei einer PubSub basierten Kommunikation sendet der Publisher die Nachrichten an eine Message Oriented Middleware (MOM), wobei er nicht weiß, wie viele Subscriber es für diese Daten gibt. Der Subscriber hingegen kennt den Publisher nicht, sondern teilt lediglich der MOM mit, welche Daten er gerne erhalten möchte. [3]

Erfolgt die Kommunikation beispielsweise über IPv4 Multicast-Pakete, so handelt es sich um eine Brokerless MOM die lediglich aus der Netzwerk-Infrastruktur besteht und die Nachrichten der Publisher an die Subscriber weiterleitet. Bei mehreren Publishern wird

Sebastian Haberkern, sehait02@hs-esslingen.de, Hochschule Esslingen, Flandernstraße 101, 73732 Esslingen am Neckar. Marcus Würß, marcus.wuerr@se.com, Stefan Eiermann, stefan.eiermann@se.com, Schneider Electric, Schneiderplatz 1, 97828 Marktheidenfeld.

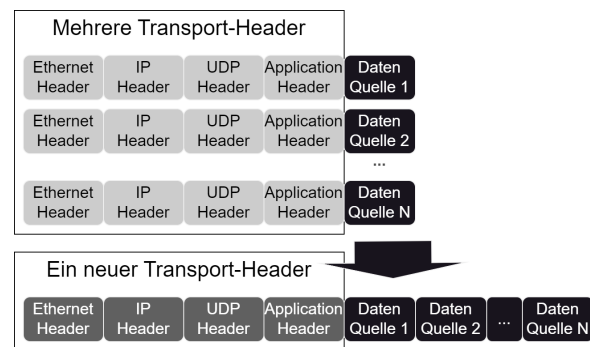


Abbildung 1. Aggregation von Applikations-Nutzdaten in Ethernet-Frames.

somit von jedem eine einzelne Nachricht an den Subscriber gesendet.

Wenn den Daten der Publisher eine eindeutige ID zugeordnet ist, müssen diese nicht zwingend in einzelnen Nachrichten an den Subscriber übertragen werden. Dies wäre auch gemeinsam in einer Nachricht möglich. Hierdurch könnte eine Entlastung des Subscribers erfolgen, da dieser für die Daten aller Publisher innerhalb eines Zyklus lediglich eine Nachricht der MOM verarbeiten muss.

Dies soll mit einem FPGA im Pfad zwischen MOM und Subscriber erfolgen, das die Nachrichten zunächst zwischenspeichern und anschließend gesammelt in einer neu erstellten Nachricht an den Subscriber weiterleiten soll. Dieser Ablauf ist schematisch in Abbildung 1 dargestellt.

Als Grundlage liegt eine Test-Umgebung vor, in der mehrere Publisher zyklisch Daten an einen Subscriber senden. Auf dem Kommunikations-Pfad ist ein FPGA eingebunden, in dessen Ethernet-Empfangs-Pfad die Ethernet-Frames an einem AXI-Stream Interface bereit liegen. Da es sich bei diesem Aufbau um ein komplexes System mit mehreren externen Komponenten (Publisher-Geräte, die Netzwerk-Infrastruktur, u.ä.) handelt, soll nun eine zweite, virtuelle Test-Umgebung erstellt werden.

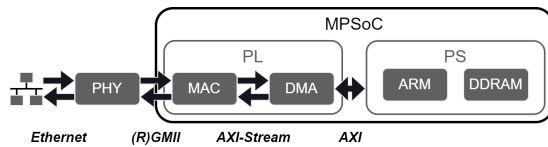


Abbildung 2. MPSoC mit Ethernet-Schnittstelle.

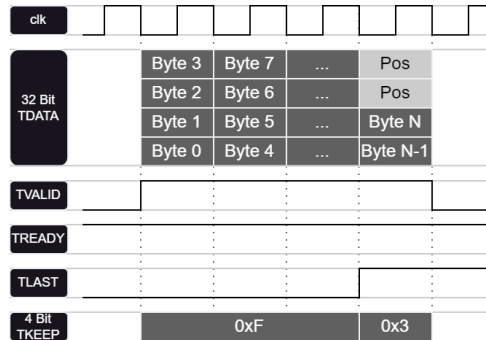


Abbildung 3. AXI-Stream Übertragung eines Pakets.

II. GRUNDLAGEN

A. Ethernet MPSoC

Um ein Multi-Prozessor-System-on-Chip (MPSoC) mit einer Ethernet-Schnittstelle zu verbinden gibt es unterschiedliche Möglichkeiten. Bei einem Xilinx® Zynq™ UltraScale+™, das neben dem Processing System (PS) auch eine Programmable Logic (PL) besitzt, kann das bspw. wie in Abbildung 2 dargestellt erfolgen.

Die Applikation und der Netzwerk-Stack werden auf dem PS ausgeführt, während die Ethernet-Frame-Verarbeitung in der PL erfolgt. Beide sind über eine AXI-Schnittstelle miteinander verbunden. Der PHY, der die Ethernet-Schnittstelle bereitstellt, ist über das Media-Independent-Interface (MII) mit dem MAC verbunden. Die Datenkommunikation zwischen MAC und PS erfolgt mit Hilfe eines DMA, wobei hier das AXI-Stream Protokoll als Verbindung zwischen MAC und DMA zum Einsatz kommt.

B. AXI-Stream Protokoll

Das AXI-Stream Protokoll ist für den Datenaustausch auf einer gerichteten Punkt-zu-Punkt-Verbindung vorgesehen. Die Steuerung der Datenübertragung erfolgt mit Hilfe der beiden Signale TVALID und TREADY. Darüber können der Sender und der Empfänger einen Handshake ausführen, wodurch die Übertragung eines Datenworts stattfindet. Mehrere Datenwörter können durch das TLAST Signal zu einem zusammengehörigen Paket gruppiert werden. [1] Eine Beispiel-Übertragung eines N+1 Byte großen Pakets ist in Abbildung 3 dargestellt.

Die Übertragung der Datenwörter erfolgt über den Datenbus TDATA, dessen Wortbreite ein Vielfaches von 8 Bit beträgt. Um eine beliebige Anzahl an Bytes in einem Paket zu übertragen, können am Ende

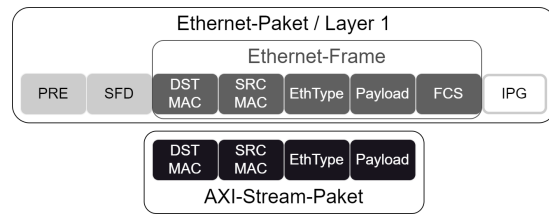


Abbildung 4. Ethernet-Paket, Ethernet-Frame und AXI-Stream.

Positions-Bytes zum Auffüllen angefügt werden. Diese werden mit Hilfe des TKEEP Signals angezeigt. [1]

C. Ethernet-Frames in AXI-Streams

Ein Ethernet-Frame besteht aus drei Teilen: Dem Header, der die Ziel-Adresse (DST MAC), die Quell-Adresse (SRC MAC) sowie den Typ der enthaltenen Daten (EthType) enthält, gefolgt von den Nutzdaten (Payload) sowie einer abschließenden Prüfsumme, genannt Frame-Check-Sequence (FCS). [2]

Die Übertragung auf dem Layer 1 beginnt mit der Präambel (PRE) und dem Start Frame Delimiter (SFD), bevor der eigentliche Ethernet-Frame folgt. Nach dem Ethernet-Frame muss eine Pausenzeit eingehalten werden, bevor die nächste Übertragung eines Ethernet-Pakets mit der PRE beginnen darf. Diese Pause wird Inter Package Gap (IPG) genannt. [2]

Diese Aufteilung zwischen Ethernet-Frame und -Paket ist in der oberen Hälfte der Abbildung 4 dargestellt.

Soll ein Ethernet-Paket nun mit dem AXI-Stream Protokoll übertragen werden, so ist es ausreichend den eigentlichen Ethernet-Frame zu übertragen. Die restlichen Teile des Ethernet-Pakets dienen lediglich der Übertragung auf dem Layer 1. [5]

Des Weiteren wird auch die FCS nicht mehr benötigt, wenn diese vom MAC bereits überprüft wurde, so dass nur gültige Ethernet-Pakete mit dem AXI-Stream weitertransportiert werden. Der Zusammenhang zwischen dem Layer 1 Ethernet-Paket und dem AXI-Stream Ethernet-Frame ist auch in Abbildung 4 dargestellt.

Die Datenrate des AXI-Streams sollte dabei mindestens so groß gewählt werden, dass ein auf Layer 1 ununterbrochener Datenstrom an Ethernet-Paketen auch auf dem AXI-Stream weitergeleitet werden kann. Ist die Datenrate des AXI-Streams zusätzlich größer, so ergeben sich selbst bei Maximal-Auslastung des Layer 1 zwischen den Paketen auf dem AXI-Stream Pausen. Dies ist in Abbildung 5 schematisch dargestellt, wobei im Beispiel die Datenrate des AXI-Streams ungefähr doppelt so groß der des Layer 1 ist.

III. KONZEPT

Zur Erstellung der virtuellen Testumgebung werden Komponenten benötigt, die AXI-Stream Pakete wiedergeben sowie aufzeichnen können. Die wiedergebende Komponente, im Folgenden als

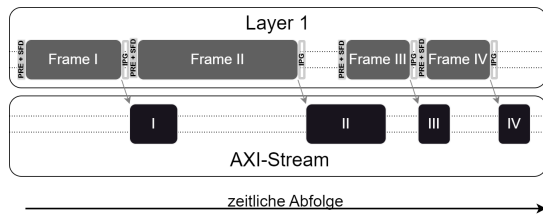


Abbildung 5. Unterschiedliche Datenraten zwischen Ethernet und AXI-Stream.

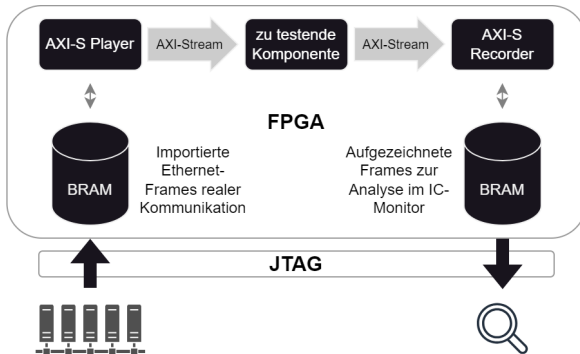


Abbildung 6. Hardware-in-the-Loop Test.

AXI-S Player bezeichnet, simuliert am System eintreffende Ethernet-Pakete, die als Stimuli der zu testenden Komponente innerhalb des FPGAs verwendet werden können.

Um den Ausgang der zu testenden Komponente verifizieren zu können, wird dieser mit der als AXI-S Recorder bezeichneten Komponente aufgezeichnet und anschließend mit einer Dekodier- und Analyse Software für industrielle Kommunikations-Protokolle, in diesem Fall dem IC-Monitor [4], offline analysiert. Dieser Ablauf eines Hardware-in-the-Loop (HIL) - Tests ist in Abbildung 6 schematisch dargestellt.

Die wiedergegebenen Pakete können je nach Testfall sowohl aus einer Aufzeichnung einer realen Ethernet-Kommunikation stammen oder auch virtuell generiert werden.

IV. REALISIERUNG

A. Datenformat

Um die Frames im FPGA zwischenspeichern zu können wird ein Datenformat benötigt, welches sowohl die Daten eines Ethernet-Frames als auch eine Pause zwischen zwei AXI-Stream Paketen abbilden kann. Zusätzlich soll es außerdem das Ansteuern von vier unabhängigen Ausgangs-Bits unterstützen, um weitere Komponenten auf den Ablauf synchronisieren zu können. Dazu wurde das in Abbildung 7 dargestellte Datenformat festgelegt.

Es besteht aus einem Start Marker, einem Info-Block, den eigentlichen Nutzdaten sowie einem End Marker.

Der Start Marker besteht aus einem konstanten Wert

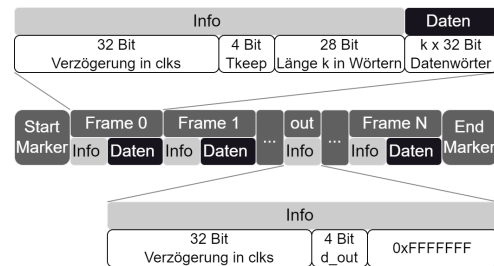


Abbildung 7. Datenformat.

und zeigt lediglich an, dass es sich bei den folgenden Datenwörtern um Daten für den AXI-S Player handelt. Die Daten selbst beginnen immer mit einem Info-Block. Dieser kann zwei Formen annehmen:

Im ersten Fall zeigt er an, dass nun ein Ethernet-Frame folgt. Hierbei beinhaltet er die Anzahl an Idle-Takten vor dem Frame, dem Wert des TKEEP Signals des letzten Datenworts sowie die Anzahl der Datenwörter, die unmittelbar folgen.

Alternativ kann der Info-Block auch dafür verwendet werden, um die vier Ausgangs-Bits anzusteuern. In diesem Fall beginnt er erneut mit der Anzahl an Idle-Takten, enthält dann jedoch zuerst das Datenwort der Ausgangs-Bits und abschließend einen konstanten Wert, der zur Unterscheidung diesem zu einem Frame-Info-Block dient.

B. Python Skript Player

Um das Generieren von Daten im zuvor erläuterten Format zu vereinfachen, wurde ein Python-Skript erstellt. Dieses kann aus Wireshark exportierte oder mit der Python Library Scapy erzeugte Frames als Eingangsdaten verarbeiten. Neben der Angabe von konkreten Pause-Zeiten zwischen zwei Frames gibt es außerdem die Möglichkeit den Wert so berechnen zu lassen, dass die Idle-Takte und die zur Übertragung auf dem AXI-Stream notwendigen Takte der Zeit eines äquivalenten Ethernet-Pakets auf dem Layer 1 entspricht. Hierdurch lässt sich eine 100% Auslastung der Ethernet-Schnittstelle emulieren.

Ein Beispielcode ist in der Abbildung 8 dargestellt.

Das Skript bietet zwei Möglichkeiten zur Ausgabe der Daten: Das Erzeugen von Verilog-Code zur direkten Initialisierung des Player-IPs sowie die Ausgabe einer tcl-Datei, die den Player zur Laufzeit über die Debug-Schnittstelle mit Daten beschreiben kann. Für letzteres muss der Anwender einen BRAM-Controller sowie einen JTAG-AXI-Master im Blocks-Design entsprechend konfigurieren - der Player selbst stellt lediglich ein BRAM-Interface zur Verfügung.

C. Verilog IP: AXI-S Player

Der AXI-S Player besteht aus einem 36 kBit BRAM-Block sowie einem Zustandsautomaten und

```

1 from AxisEthFrameTestEnv import *
2
3 AXIS_CLK = 125e6
4 myGen = AxisPlayerBramGenerator(AXIS_CLK, ETHERNET_CLK = 1e9)
5 myGen.addHeader()
6
7 # burst of 2 frames
8 from scapy.all import Ether, ARP
9 myGen.addFrame(PAUSE_AXIS_H2T, bytes(Ether()/ARP()).hex())
10 myGen.addFrame(PAUSE_ETH_H2T, "FFFF[..]0000")
11
12 # wait 1us and generate 2us pulse
13 PAUSE_1us = 1e-6 * AXIS_CLK
14 myGen.addDout(PAUSE_1us, 0b0001)
15 myGen.addDout(2 * PAUSE_1us, 0b0000)
16
17 myGen.addTrailer()
18 myGen.generateInit("initPlayer.v")
19 myGen.generateTcl("loadPlayer.tcl")

```

Abbildung 8. Beispiel Python-Code zur Datenerzeugung für den AXI-S Player.

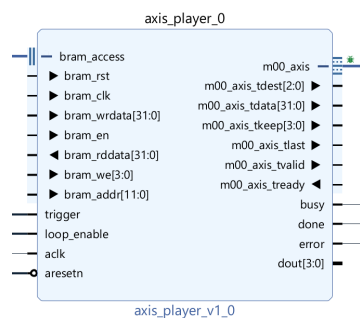


Abbildung 9. AXI-S Player IP.

kann anhand vorliegender Daten, in dem zuvor erläuterten Datenformat, einen AXI-Stream erzeugen. Die Daten des BRAMs können zur Laufzeit über das BRAM-Interface „bram_access“ modifiziert werden. Zur Steuerung stehen mehrere Signale zur Verfügung. Der IP-Block mit allen Ein- und Ausgängen ist in Abbildung 9 abgebildet.

Mit den beiden Eingangs-Signalen „trigger“ sowie „loop_enable“ kann das Erzeugen des AXI-Streams gesteuert werden. Eine steigende Flanke am trigger-Eingang startet das Erzeugen. Ist der Player am End Marker angekommen, so springt er automatisch wieder zum Start Marker und wiederholt dieselbe Sequenz erneut, wenn das loop_enable-Signal gesetzt ist. Das vier Bit breite Ausgangs-Signal „d_out“ ist zur Synchronisation weiterer Komponenten auf den erzeugten AXI-S vorgesehen. Dessen Ansteuerung kann in den Ablauf integriert werden. Damit die mit Hilfe des Python-Skripts konfigurierten Idle-Zeiten zwischen den Frames korrekt eingehalten werden können, darf der Ausgangs-Stream des AXI-S Players nicht blockiert sein: Das TREADY-Signal muss über die gesamte Zeit logisch gesetzt sein. Dies kann beispielsweise mit einem AXI-Stream FiFo direkt am Ausgang des AXI-S Players sichergestellt werden.

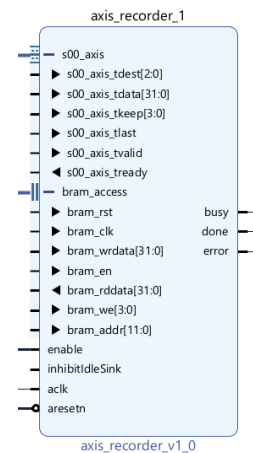


Abbildung 10. AXI-S Recorder IP.

D. Verilog IP: AXI-S Recorder

Wie der AXIS-S Player besteht auch der AXI-S Recorder aus einem 36 kBit BRAM-Block mit zugehörigem Interface sowie einem ähnlichen Zustandsautomaten. In der nachfolgenden Abbildung 10 ist der IP-Block mit allen Ein- und Ausgängen abgebildet.

Das Aktivieren einer Aufnahme erfolgt mit dem enable-Signal. Im inaktiven Zustand akzeptiert der IP-Block standardmäßig keinen AXI-Stream. Dies kann über einen Parameter geändert werden, so dass Transfers auch im inaktiven Zustand auf dem AXI-Stream angenommen, die Daten jedoch verworfen werden.

E. Python Skript Recorder

Nach einer Aufzeichnung liegen die Ethernet-Frames im vorgestellten Datenformat im BRAM des Recorders vor.

Um diese Daten zur Weiterverarbeitung in das pcap Format zu konvertieren, gibt es ein weiteres Python-Skript. Dieses erstellt zunächst eine tcl-Datei, die den Inhalt des AXI-S Recorder BRAMs, analog wie beim AXI-S Player zuvor beschrieben, über die JTAG-Schnittstelle ausliest. Diese Daten können anschließend in das Python-Skript eingefügt werden, welches dann daraus eine pcap-Datei mit den aufgezeichneten Ethernet-Frames erstellt.

Diese Datei kann nun zur weiteren Analyse in gängige Software importiert werden.

V. FAZIT

Das ursprünglich vorgestellte Ziel, das Erstellen einer virtuellen Testumgebung für einen OPC UA UADP- Telegramm Aggregations-IP-Block, konnte mit dem hier vorgestellten Konzept umgesetzt werden.

A. Beispiel: Aggregator-IP

In Abbildung 11 ist ein Screenshot eines einzelnen Testfalls dieses Aggregations-IP-Blocks zu sehen.

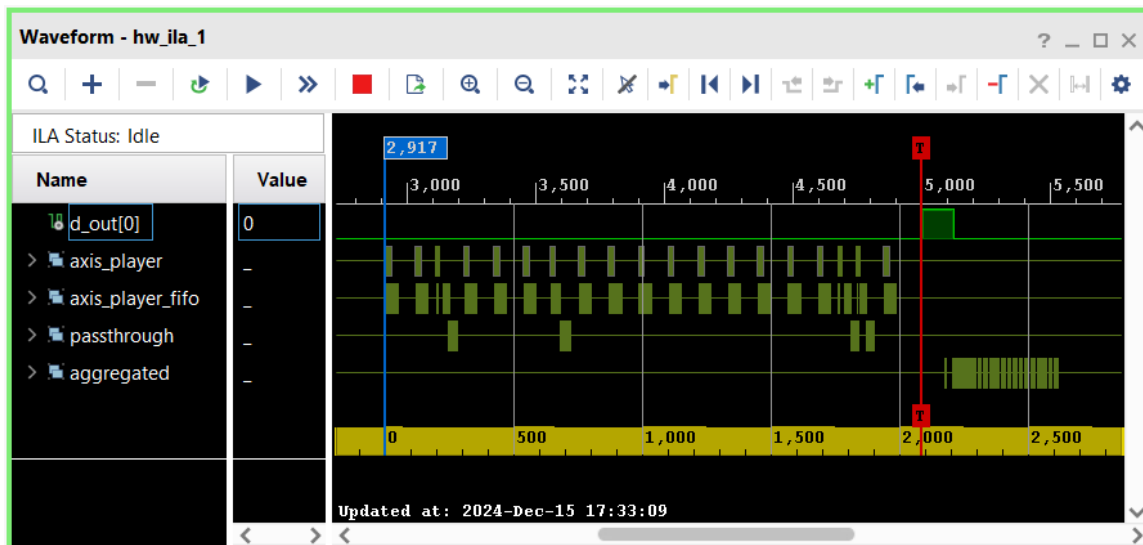


Abbildung 11. Vivado® Screenshot: Aufzeichnung des System Integrated Logic Analyzer (System ILA) am Aggregator-IP Beispiel.

Dessen Ablauf bestand aus den folgenden Frames, die alle mit einer effektiven Übertragungsrate von 1 Gbit/s abgespielt wurden:

Zwei UADP-Frames (ID: 14, 3), einem ARP-Frame, drei UADP-Frames (10, 1, 2), einem PING-Frame, neun UADP-Frames (8, 7, 5, 4, 15, 11, 6, 9, 13), zwei ARP-Frames, einem UADP-Frame (12) und eine steigende Flanke des d_out-Signals als Trigger für das Erzeugen des aggregierten Frames.

Zusätzlich zu dem d_out Signal (erste Zeile der Abbildung 11) sowie den erzeugten Frames des AXI-S Players (zweite bzw. dritte Zeile) sind die beiden Ausgänge des Aggregations-IP-Blocks zu sehen: Der Passthrough-Ausgang (Zeile vier) und der Ausgang für die neu erstellten Frames mit den aggregierten Daten (fünfte Zeile).

In der Abbildung 11 ist außerdem zu sehen, dass die drei ARP-Frames sowie der PING-Frame mit einer kleinen Verzögerung am Passthrough-Interface ausgegeben werden. Die UADP-Frames hingegen werden verarbeitet und deren Nutzdaten gesammelt in einem neuen Frame nach der steigenden Flanke des d_out-Signals ausgegeben. Der Frame mit den aggregierten Daten konnten anschließend in der Software IC-Monitor analysiert und verifiziert werden.

B. Datenpfade

Neben dem in Abbildung 6 dargestellten Ablauf lassen sich weitere Pfade des Systems testen.

Diese sind in der Abbildung 12 schematisch dargestellt. Der Pfad des vorgestellten HIL-Tests, beginnend mit dem Aufzeichnen der realen Kommunikation bis zum Auswerten der erzeugten Frames des Aggregator-IPs, ist in orange eingezeichnet.

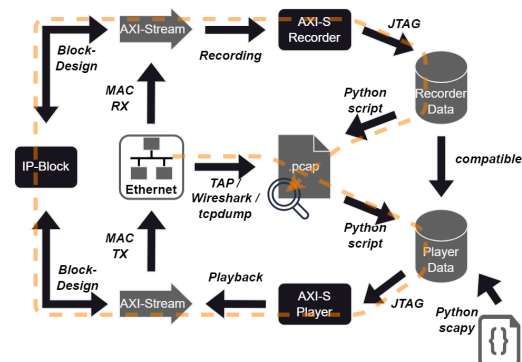


Abbildung 12. Mögliche Datenpfade für einen Testfall.

C. Open-Source Repository

Die beiden vorgestellten Verilog IPs AXI-S Player und AXI-S Recorder sowie die zugehörigen Python-Skripte zur BRAM-Daten-Verarbeitung werden in Kürze in einem Github-Repository unter der MIT Lizenz veröffentlicht:

<https://github.com/schneider-electric/AxiStreamEthernetFrameTestEnv>

Die beiden IPs wurden bereits auf einem Zynq™ UltraScale+™ XCZU9EG-2FFVB1156 MPSoC sowie einem AMD Kria K26 SoM verwendet und sollten allgemein mit allen SoCs der Zynq™ UltraScale+™ Serie kompatibel sein. Die tcl-Skripte für den Zugriff auf den BRAM über die Debug-Schnittstelle wurden für Vivado® 2024.1 erstellt.

Da der Verilog-Code keine speziellen Hardware-Anforderungen benötigt, sollten die IPs jedoch auch ohne großen Aufwand auf anderen Architekturen implementierbar sein.

LITERATURVERZEICHNIS

- [1] Arm Limited (or its affiliates). *AMBA AXI Stream Protocol Specification*. Zugriff am 28 Oktober 2024. 2024. URL: <https://developer.arm.com/documentation/ih0051/latest>.
- [2] Wolfgang Babel. *Systemintegration in Industrie 4.0 und IoT*. Springer Vieweg Wiesbaden, 2024. ISBN: 978-3-658-42987-4.
- [3] OPC Foundation. *UA Part 14/ PubSub Annex B (informative)Client Server vs Publish Subscribe*. Zugriff am 6 November 2024. 2024. URL: <https://reference.opcfoundation.org/Core/Part14/v104/docs/B>.
- [4] Steinbeis Embedded Systems Technologies GmbH. *Industrial Communication Protocol Analysis - IC-Monitor*. Zugriff am 26 November 2024. 2023. URL: <https://www.ic-monitor.com>.
- [5] Inc. Xilinx. *ds759_axi_ethernet*. Zugriff am 27 November 2024. 2012. URL: https://docs.amd.com/v/u/en-US/ds759_axi_ethernet.



Sebastian Haberkern erhielt den akademischen Grad B.Eng. in Technischer Informatik mit Schwerpunkt Cyber-physische Systeme von der Hochschule Esslingen im Jahr 2021. Anschließend war er als Entwicklungsingenieur bei Steinbeis EST (Embedded Systems Technologies) GmbH in Esslingen am Neckar angestellt. Seit Ende 2023 studiert er Angewandte Informatik im Master an der Hochschule Esslingen und macht aktuell seine Abschlussarbeit bei der Firma Schneider Electric in Marktheidenfeld.



Marcus Würß erhielt den akademischen Grad Diplom in Technischer Informatik mit Schwerpunkt Automatisierungstechnik von der FH Würzburg im Jahr 2005. Anschließend arbeitete er als Software Developer, Abteilungsleiter und seit 2022 als Senior Principal Network Architect bei der Firma Schneider Electric in Marktheidenfeld.



Stefan Eiermann erhielt den akademischen Grad Diplom Ingenieur (FH) in Elektronik und Informationstechnik von der Hochschule Heilbronn im Schwerpunkt Kommunikationstechnik im Jahr 2002. Anschließend war er als Freiberufler bei Diehl Luftfahrt Elektronik tätig. Seit 2004 ist bei der Firma Schneider Electric in Marktheidenfeld angestellt, wo er sich mit Schaltungsentwicklung für funktionale Sicherheit in der Antriebstechnik und ab 2012 mit FPGA Entwicklung für PLCs betätigt.

Entwicklung und Implementierung eines multifunktionalen PWM Modulators

Maik Greschuck, Clemens Elskamp, Bernhard M. Rieß.

Zusammenfassung—In dieser Arbeit wird ein flexibles und vielseitiges Steuergerät vorgestellt. Dieses dient dem Einsatz in verschiedenen Anwendungen der Leistungselektronik, insbesondere zur Ansteuerung von Testobjekten wie z.B. Wechselrichtern. Dazu wurde eine Leiterplatte entwickelt, die die notwendigen Komponenten, insbesondere einen Microcontroller, zur Ansteuerung der Testobjekte beinhaltet. Darüber hinaus wurde die benötigte Software für den Microcontroller entwickelt und getestet. Der multifunktionale PWM-Modulator bietet die Möglichkeit, während des Betriebs Parameter wie Modulationsgrad und Ausgangsfrequenz frei zu konfigurieren. Damit stellt er eine flexible und kostengünstige Alternative zu kommerziellen Steuergeräten dar.

Schlüsselwörter—Leistungselektronik, Test, Wechselrichter, IGBT, Mikrocontroller, Pulsweitenmodulation

I. EINLEITUNG

Die Leistungselektronik spielt eine zentrale Rolle in modernen technischen und industriellen Anwendungen. Sie ermöglicht die effiziente Steuerung elektrischer Energie, etwa in Antriebssystemen und für die Telekommunikation, und ermöglicht die Umwandlung und Einspeisung erneuerbarer Energien. Wechselrichter beeinflussen dabei entscheidend die Leistung und Zuverlässigkeit dieser Systeme.

A. Motivation

Für die Entwicklung und Optimierung von Leistungshalbleitern wie IGBTs und MOSFETs sind präzise Tests erforderlich. Kommerzielle Steuergeräte [1] bzw. PCI-Pulserkarten [2] existieren, doch sind diese teuer und oft auf spezifische Anwendungen begrenzt. Benötigt werden vielmehr anpassbare Steuergeräte, die flexibel verschiedene Testszenarien abdecken können. Verschiedene Arbeiten [3] [4] [5] [6] haben gezeigt, dass das XMC4500 Relax Kit von Infineon Technologies zur flexiblen Ansteuerung von Testständen geeignet ist.

B. Ziele der Arbeit

Diese Arbeit verfolgt das Ziel, einen kostengünstigen, flexiblen und multifunktionalen PWM-Modulator zu entwickeln, der verschiedene Modulationsverfahren

unterstützt. Dadurch sollen praxisnahe Testszenarien realisiert und Anpassungen an das Testszenario ohne Hardwareänderungen ermöglicht werden mit der Möglichkeit, den Leistungsfaktor $\cos(\varphi)$ zur gezielten Belastung der Halbleiterbauelemente einstellen zu können. Dazu wurde eine Leiterplatte und die Software für das XMC4500 Relax Kit entwickelt, die Wechselrichter flexibel ansteuern kann. Die wichtigsten Anforderungen sind:

- Modulationsgrad und Frequenz sind im laufenden Betrieb einstellbar.
- Synchronisation auf externe periodische Signale.
- Zuverlässige Datenübertragung via Ethernet.
- Speicherung von Parametern, um Testeinstellungen nach Neustart zu erhalten.

Abschnitt II gibt einen Überblick über den Stand der Technik und beschreibt die für die hier vorgesehenen applikationsnahen Halbleitertests relevanten Wechselrichter-Topologien und Modulationsverfahren. Anschließend wird die verwendete Hard- und Software in Abschnitt III sowie die Generierung der Modulationssignale und der Versuchsaufbau in Abschnitt IV erläutert. Abschnitt V erklärt den Programmablauf der entwickelten Software und präsentiert die erzielten Ergebnisse. Den Abschluss bildet eine Zusammenfassung.

II. STAND DER TECHNIK, WECHSELRICHTER UND MODULATIONSVERFAHREN

In der Leistungselektronik sind verschiedene Methoden und Topologien bekannt, um Gleich- und Wechselspannung ineinander umzuformen [7]. Relevant für diese Arbeit sind lediglich grundlegende 2-Level-Topologien.

A. Einphasige Wechselrichter

Einphasige Wechselrichter sind elektronische Schaltungen, mit denen Gleichspannung durch gezieltes Schalten der Leistungshalbleiter in Wechselspannung umgewandelt wird [8] [9]. Einphasige 2-Level-Wechselrichter bzw. H-Brücken-Wechselrichter bestehen aus zwei Halbbrücken mit jeweils zwei Transistoren (hier IGBTs) und deren antiparallelen Freilaufdioden. Die in der Regel induktive Last wird zwischen die Mittelpunkte der beiden Halbbrücken geschaltet. Die Transistoren und Dioden in den jeweiligen Halbbrücken können als Schalter betrachtet werden, von

Maik Greschuck, m.greschuck@gmx.de, Clemens Elskamp, clemens.elskamp@infineon.com

Bernhard Rieß, bernhard.riess@hs-duesseldorf.de, Hochschule Düsseldorf, Münsterstraße 156, 40476 Düsseldorf.

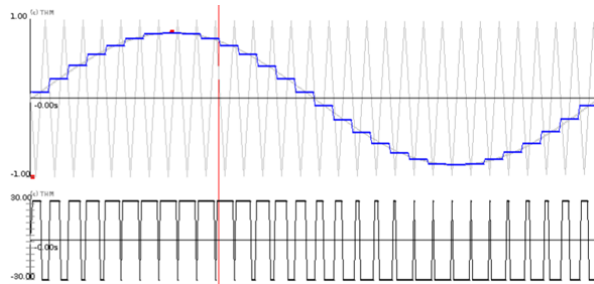


Abbildung 1. Sinusförmige Treppenfunktion (blau) mit 30 Stufen und Dreiecksfunktion (grau) [11].

denen im Betrieb immer ein Schalter offen und der andere Schalter geschlossen ist.

B. Pulsweitenmodulation

Um mit den beiden beschriebenen Schalterstellungen am Ausgang nicht nur dauerhaft die volle Spannung der Spannungsquelle anliegen zu haben, wird Pulsweitenmodulation (PWM) eingesetzt. Dabei wird der Mittelwert einer Spannung durch zeitlich unterschiedlich langes Verweilen in den Schaltpositionen beeinflusst [7]. Ein PWM-Signal wird über folgende Parameter beschrieben [10]:

- 1) Die maximale Spannung (HIGH) sowie die minimale Spannung (LOW), die eine PWM-Spannung annehmen kann.
- 2) Die PWM-Frequenz, welche angibt, wie schnell Pulse aufeinander folgen.
- 3) Der Duty Cycle gibt das Verhältnis zwischen dem HIGH-Anteil des Pulses und der gesamten Periodendauer an.

Eine Möglichkeit, eine PWM zu erzeugen, ist die Dreieckmodulation [7]. Dabei wird ein Referenzsignal, welches die Form des zu modulierenden Signals hat, mit einem Trägersignal verglichen. Das Ergebnis dieses Vergleichs wird dazu verwendet, die Schaltzeitpunkte und damit den Duty Cycle zu bestimmen. Bei der Dreieckmodulation wird als Trägersignal eine Dreiecksfunktion benutzt.

Eine übliche Art und Weise, um in einer H-Brückenschaltung eine sinusförmige Ausgangsspannung zu generieren, ist die sinusförmige Pulsweitenmodulation (SPWM). Dabei wird eine Abfolge diskreter Steuerspannungswerte in Form einer Treppenfunktion erzeugt. Wird dieses Verfahren auf eine Vielzahl von aufeinander folgenden Treppenstufen angewendet, so entsteht eine Ausgangsspannung, die ebenfalls einen treppenförmigen Verlauf von Mittelwerten annimmt. Dieses Verfahren stellt das einfachste Modulationsverfahren für ein- und dreiphasige Wechselrichter dar [12]. In Abbildung 1 wird ein Sinussignal mit einer Treppenfunktion angenähert, die durch ein PWM-Signal generiert wurde.

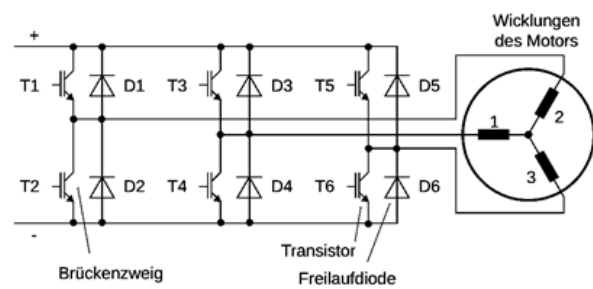


Abbildung 2. Dreiphasiger Wechselrichter bestehend aus 3 Halbbrücken (links) mit Motor (rechts)[15].

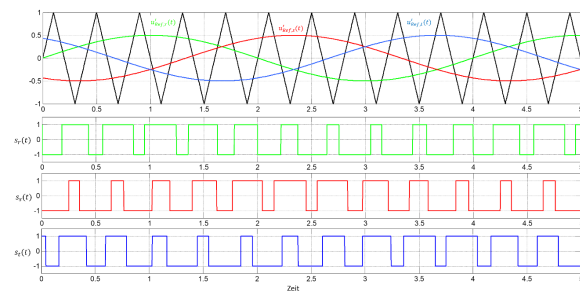


Abbildung 3. Erzeugung dreier um jeweils 120° zueinander versetzter SPWM-Signale [7].

Das Verhältnis zwischen der Amplitude des sinusförmigen Modulationssignals und der Dreiecksfunktion wird als Modulationsgrad m bezeichnet [13]:

$$m = \frac{(\text{Amplitude des modulierten Signals})}{(\text{Amplitude des Dreiecksignals})} \quad (1)$$

C. Dreiphasige Wechselrichter

Ein dreiphasiger 2-Level-Wechselrichter besteht aus drei Halbbrücken, also in Summe sechs Transistoren und sechs Dioden, wobei die Dioden antiparallel zu den Transistoren geschaltet werden wie in Abbildung 2 dargestellt [14].

D. Dreiphasige sinusförmige Pulsweitenmodulation

Das in Abschnitt II-B beschriebene Prinzip der Dreieckmodulation kann auf dreiphasige Anwendungen erweitert werden, wodurch auch dreiphasige Wechselrichter angesteuert werden können. Dazu müssen zwei weitere sinusförmige Referenzsignale, die um jeweils 120° phasenverschoben sind, hinzugefügt werden [7]. Die Abbildung 3 zeigt drei sinusförmige Referenzsignale, die zueinander um jeweils 120° versetzt sind.

III. CONTROLLER UND SOFTWAREUMGEBUNG

A. XMC4500 Relax Kit

In dieser Arbeit wird das XMC4500 Relax Kit von Infineon Technologies verwendet, welches in Abbildung 4 dargestellt ist. Es besteht aus einem Evaluation

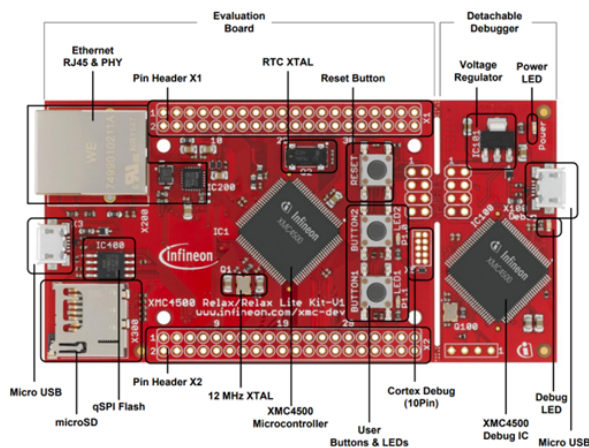


Abbildung 4. XMC4500 Relax Kit [16].

Board mit dem zu nutzenden Controller samt Peripherie (links) und dem zur Programmierung nötigen Programmer und Debugger (rechts), der bei Bedarf abgetrennt werden kann. Auf dem Evaluation Board befindet sich ein Ethernet-Port, der für die Kommunikation mit dem Steuer-PC genutzt wird.

B. Entwicklungsumgebung „Digital Application Virtual Engineer“

„Digital Application Virtual Engineer“ (DAVE) ist eine von Infineon Technologies zur Verfügung gestellte Entwicklungsumgebung, die speziell für die Programmierung und Konfiguration von Mikrocontrollern der XMC-Serie konzipiert wurde. In DAVE ist es möglich, als „Apps“ bereitgestellte Funktionsblöcke auszuwählen, mit einer graphischen Oberfläche zu parametrieren und den C-Code generieren zu lassen. Zudem können die initialen Einstellungen der jeweiligen Komponenten in den Apps vorkonfiguriert werden, was der Fehlervermeidung dient. In dieser Arbeit werden die folgenden DAVE-Applikationen verwendet:

ADC_Measurement: Die im XMC4500 Relax Kit eingebauten AD-Wandler werden genutzt um angelegte Spannungswerte an den Analogeingängen in digitale Werte umzuwandeln, aus denen sich im Betrieb die Parameter für die zu generierenden Pulse berechnen lassen.

PWM_CCU8 – Capture and Compare Unit: Die CCU8 wird verwendet, um die PWM-Signale zu erzeugen. Dabei kommen bis zu drei CCU8-Slices zum Einsatz. Jedes Slice steuert die Treiber einer Halbbrücke und erzeugt das normale und das dazu invertierte PWM-Signal, um die High-Side- und Low-Side-Schalter abwechselnd ein- und auszuschalten. Werden mehrere CCU8-Slices eingesetzt, können diese synchron betrieben werden. Es können Frequenz, Duty Cycle und die Totzeiten der PWM-Signale eingestellt werden.

ETH_LWIP - lightweight Internet Protocol: Diese Applikation ermöglicht den Zugang zu verschiedenen Protokollen für die Datenübertragung via Ethernet. In dieser Anwendung wird das Transmission Control Protocol (TCP) verwendet, welches Fehlerkorrekturmechanismen mit sich bringt und korrekte Datenübertragung gewährleistet.

DAC_LUT (Digital to Analog Converter - Lookup table): Die DAC_LUT-App verwendet eine Look-Up Table (LUT) um vorab berechnete oder definierte Werte aus dem Speicher abzurufen und in analoge Signale umzuwandeln. Dabei kann zwischen verschiedenen „Standard-Signalen“ wie z.B. einem Sinus oder einem Dreieck ausgewählt werden.

(Pin-)Interrupt: Interrupts sind grundlegende Mechanismen in Mikrocontrollersystemen, die es ermöglichen, auf externe Ereignisse oder interne Zustandsänderungen zu reagieren.

Digital_IO: Die Digital_IO-APP ermöglicht die Konfiguration und Verwaltung digitaler Ein- und Ausgänge (I/O). Die Software verwendet die digitalen I/Os, um sowohl Signale an externe Geräte zu übermitteln, als auch von Extern zu empfangen.

EEPROM (Electrically Erasable Programmable Read-Only Memory): Diese App stellt einen persistenten Speicher bereit. Da der XMC4500 Mikrocontroller keinen echten EEPROM-Speicher besitzt, wird der Flash-Speicher zur Speicherung der eingestellten Betriebsmodi und Parameter genutzt. Dadurch bleiben die Parametereinstellungen nach einem Neustart vorhanden und es ist ein mobiler Einsatz des Systems im Feld möglich.

IV. ERZEUGUNG DREIPHASIGER STEUERSIGNALE UND VERSUCHSAUFBAU

A. Implementierung der SPWM

Zur Generierung sinusförmiger PWM-Signale werden die Pulsbreiten mit Werten aus der LUT, die von der DAC_LUT-App generiert wurde, berechnet. Die LUT enthält 1000 Einträge, die jeweils einen Spannungspegel innerhalb eines Wertebereichs von -2048 bis +2047 repräsentieren. Durchläuft man die LUT, wird ein Sinusverlauf mit 1000 Stufen nachgebildet. Zur Erzeugung der PWM-Signale wird die LUT an der Stelle ausgelesen, wo sich der jeweilige Sinus als nächstes befinden soll. Diese Werte werden über eine lineare Funktion in einen Duty Cycle umgewandelt. Der LUT-Wert -2048 wird dabei zum Duty Cycle 0 % und der LUT-Wert +2047 wird zum Duty Cycle 100 %. Diese Werte werden an die CCU8 übergeben und von dieser als PWM-Signale ausgegeben. Abbildung 5 zeigt ein auf diese Weise generiertes PWM-Signal (rot) und das vom Oszilloskop demodulierte Sinussignal (grün).

B. Validierung des SPWM-Ansatzes

Um den Ansatz der SPWM-Generierung zu validieren, wurde ein dreiphasiger Wechselrichter, be-

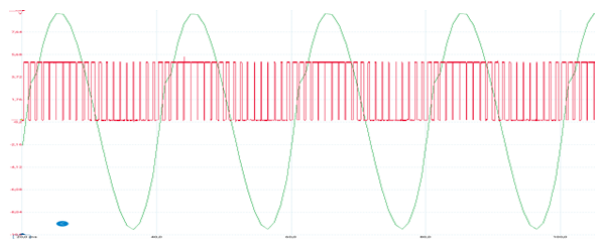


Abbildung 5. Generiertes PWM-Signal (rot) und demoduliertes Sinussignal (grün).

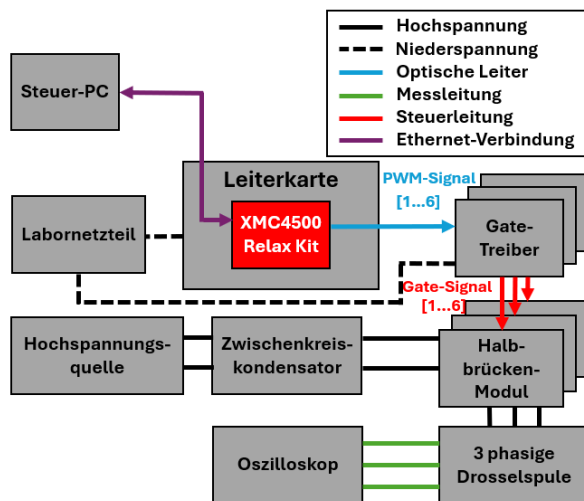


Abbildung 6. Blockschaltbild des Versuchsaufbaus zur Validierung des dreiphasigen SPWM-Ansatzes.

stehend aus drei Halbbrückenmodulen, angesteuert. Dazu wurden zwei weitere CCU8-Slices hinzugefügt, die ebenfalls ihren Duty Cycle mit den Werten aus der LUT berechnen. Es wird die Position, wo die LUT ausgelesen wird, für die anderen zwei CCU8 Slices um 1/3 bzw. 2/3 der Periodenlänge nach hinten verschoben, und damit eine Phasenverschiebung von 120° zwischen den Phasen sichergestellt. Abbildung 6 zeigt das Blockschaltbild des Versuchsaufbaus zur Validierung des dreiphasigen SPWM-Ansatzes mit folgenden Komponenten:

Steuer-PC: Zur Ansteuerung des Versuchsaufbaus wird ein Laptop verwendet, welcher via Ethernet-LAN und TCP mit dem XMC4500 Relax KIT zur Parametrierung des Controllers verbunden ist.

Leiterkarte: Im Zuge dieses Projekts wurde eine Leiterkarte entwickelt, auf der sich das XMC4500 Relax Kit befindet und mit der ein robuster Aufbau und übersichtliche Verbindungen zwischen dem Kit und allen weiteren Komponenten des Versuchsaufbaus hergestellt werden kann. In Abbildung 7 ist die bestückte Leiterkarte mit aufgestecktem XMC4500 Relax Kit dargestellt. Diese stellt folgende Ein- und Ausgänge zur Verfügung:

Eingänge: Auf der Leiterkarte befinden sich rechts Eingänge für elektrische und optische Synchronsignale

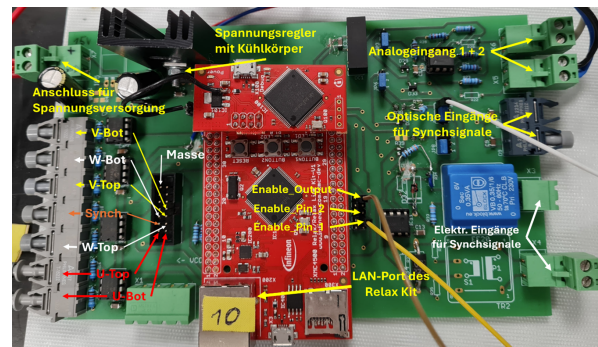


Abbildung 7. Leiterkarte (grün) mit aufgestecktem XMC4500 Relax Kit (rot).

und für Analogsignale, mit denen der Controller im Betrieb gesteuert werden kann. Zwei Eingänge rechts unten dienen über jeweils einen Transformator entkoppelt der möglichen Synchronisation auf Sinusspannungen bis zu 230 V. Rechts mittig stehen zwei optische Empfänger als Eingang für Synchronsignale zur Verfügung. An die Eingänge rechts oben können die Analogsignale angelegt werden, mit denen während des Betriebs sich verändernde Parameter an den Controller übergeben werden können.

Ausgänge: Die Aufgabe des Controllers liegt in der Generierung von Signalen, die an die Gate-Treiber übertragen werden sollen. Um die Leiterkarte mit dem darauf befindlichen Controller und den Messaufbau elektrisch zu entkoppeln, werden Lichtwellenleiter verwendet. An der linken unteren Kante der Leiterkarte befinden sich optische Transmitter zum Übertragen der SPWM-Signale. Zusätzlich werden die SPWM-Signale elektrisch auf freiliegende Pins auf der Leiterkarte ausgegeben, über die die Treiber galvanisch gekoppelt mit Flachbandkabel angesteuert oder z.B. Messgeräte wie ein Oszilloskop angeschlossen werden können. Diese befinden sich in Abbildung 7 unmittelbar links vom XMC4500 Relax Kit. Links oben wird die Versorgungsspannung angeschlossen. Die Kosten für die bestückte Leiterkarte liegen unter 200 €, die für das XMC4500 Relax Kit bei ca. 35 €. Damit betragen die gesamten Hardwarekosten der entwickelten Lösung weniger als 235 €.

Gate-Treiber: Infineon EiceDRIVER: Der Controller ist mit Lichtwellenleitern mit drei Infineon EiceDRIVER Gate-Treibern (2ED300C17-S auf Eval-Board 2ED300E17-SFO) verbunden. Sie steuern drei Halbbrückenmodule, besitzen also je zwei optische Empfänger, je einen für die High- und einen für die Low-Side.

Halbbrückenmodul: Im Versuchsaufbau werden drei FF900R12ME7-Module der Fa. Infineon eingesetzt. Diese bestehen aus je zwei IGBT-Systemen und antiparallelen Dioden mit einer Nennstromstärke von 900 A und einer Spannungsfestigkeit von 1200 V.

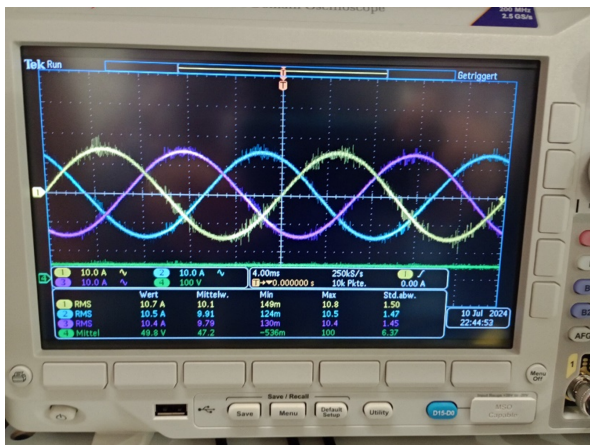


Abbildung 8. Messung dreier, um 120° versetzter Ausgangsströme.

Hochspannungsquelle: Der EA-PSB 12000-40 4U der Fa. EA Elektro-Automatik versorgt die Zwischenkreiskondensatoren in diesem Funktionstest mit bis zu 70 V. Alle Messungen wurden mit Sicherheits-Kleinspannung durchgeführt, wenngleich diese Quelle eine Hochspannungsquelle ist.

Zwischenkreiskondensatoren: Der Zwischenkreis besteht aus neun parallelgeschalteten Kondensatoren mit jeweils einer Kapazität von 400 µF, in Summe also 3.6 mF und einer Nenngleichspannung von 1200 V.

Labornetzteil – Voltcraft PS 2403 Pro: Dieses bietet zwei galvanisch getrennte Ausgangsspannungen bis 24 V und einen Ausgangsstrom bis 3 A. Im Versuchsaufbau versorgt es die Leiterkarte mit 7 V und die Gate-Treiber mit 15 V.

Dreiphasige Drosselspule: Verschaltet im Stern auf einem Sammelkern mit einer Induktivität von jeweils 10 mH.

Abbildung 8 zeigt exemplarisch die mit einem Oszilloskop aufgezeichneten Ströme im Testaufbau. Zu sehen sind die erwarteten drei sinusförmigen Lastströme mit der Frequenz von 50 Hz, der Amplitude von 15 A und einer Phasenverschiebung von je 120° zueinander.

V. PROGRAMMABLAUF UND ERGEBNISSE

A. Initialisierung und Parameterauswahl

Nach der Initialisierung des Programms werden vom Nutzer bestimmte Programmparameter erfragt:

Controller-Modus: Es stehen vier Modi zur Verfügung, auf deren Ablauf und Ergebnisse im Folgenden eingegangen wird:

Modus 1: Einkanalige PWM

Es wird nur ein Slice der CCU8 gestartet, so dass ein PWM-Signal und das dazu invertierte Signal ausgegeben wird.

Modus 2: Dreiphasige SPWM ohne externe Synchronisation

Drei CCU8-Slices erzeugen drei um 120° versetzte SPWM-Signale und deren invertierte Signale.

Modus 3: Dreiphasiger SPWM Slave

Der Controller empfängt ein periodisches Signal und synchronisiert sich darauf. Es werden drei SPWM-Signale und die dazu invertierten Signale ausgegeben.

Modus 4: Dreiphasiger SPWM Master

Es wird lediglich ein Synchronsignal einstellbarer Frequenz ausgegeben, auf das sich andere Controller synchronisieren können. Eine PWM-Modulation findet in diesem Modus nicht statt.

PWM-Frequenz: In diesem Menüpunkt wird die PWM-Frequenz, also die Pulsfrequenz eingestellt.

Modulationsgrad: Einstellung der unteren und oberen Modulationsgrenze. Die Modulationsgrenzen legen fest, in welchem Bereich die Modulation über den Analogeingang gesteuert werden kann.

Grenzen der Sinusfrequenz: Einstellung der je nach Modus über die Analogeingänge einstellbaren unteren und oberen Grenze der Sinusfrequenz.

Grenzen der Phasenverschiebung: Einstellung der je nach Modus analog einstellbaren unteren und oberen Grenze der Phasenverschiebung.

Referenzwinkel: Der Referenzwinkel bestimmt in Kombination mit der Phasenverschiebung den Winkel, auf den das Programm bei Erkennung eines eingehenden Synchronsignals springt. Dies dient der Kompensation möglicher Verzögerungen durch unterschiedliche Signallaufzeiten bei zu synchronisierenden Controllern.

DeadTimeRising und -Falling: Die DeadTimeRising definiert die Verzögerungszeit zwischen dem Abschalten des High-Side-Transistors und dem Einschalten des Low-Side-Transistors. Die DeadTimeFalling hingegen definiert die Verzögerungszeit zwischen dem Abschalten des Low-Side-Transistors und dem Einschalten des High-Side-Transistors.

Pulsanzahl: In diesem Menüpunkt wird die Anzahl der Pulse eingestellt, die der Controller ausgibt, bevor er automatisch stoppt.

Duty Cycle: Hier wird im Programm die untere und die obere Grenze des Bereichs eingestellt, in dem im entsprechenden Modus der Duty Cycle der PWM mittels Analogsignal eingestellt werden kann.

B. Ablauf der verschiedenen Controller-Modi und Demonstration der Ergebnisse

1) Modus 1: Einkanalige PWM: Der einkanalige Modus generiert ein PWM-Signal mit einstellbarem Duty Cycle sowie das dazu invertierte Signal. Ein mögliches Anwendungsszenario ist ein Pulstest mit einer Halbbrücke und einer Spule welches in Abbildung 9 dargestellt ist. Anstelle der drei in Abbildung 6 verwendeten Halbbrückenmodule wird hier nur ein Halbbrückenmodul eingesetzt.

Der Gate-Treiber steuert die Halbbrücke. Er besitzt zwei optische Empfänger, je einen für die High-Side und für die Low-Side. Die Gates der IGBTs werden

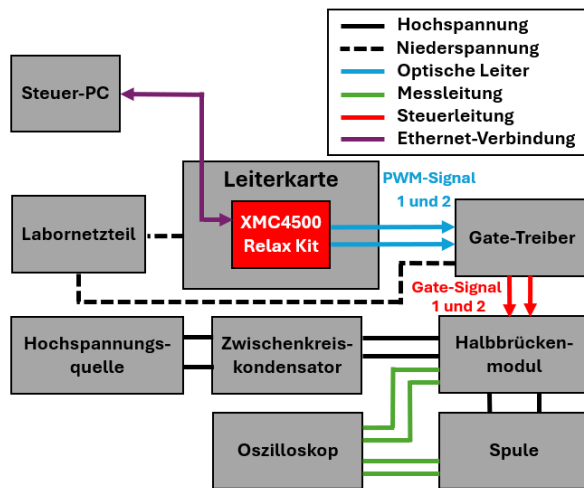


Abbildung 9. Blockschaltbild eines Versuchsaufbaus für Pulstests.

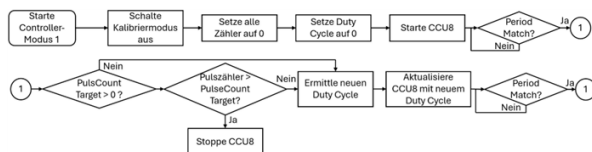


Abbildung 10. Ablaufdiagramm in Modus 1: Einkanalige PWM.

mit +15 V oder -15 V angesteuert, je nachdem, ob die IGBTs ein- oder ausgeschaltet sein sollen.

Abbildung 10 zeigt den Programmablauf in diesem Modus. Ein CCU8-Slice gibt ein PWM-Signal und das dazu invertierte Signal aus. Zunächst wird die PWM-Frequenz, sowie der minimale und maximale Duty Cycle eingestellt. Zudem kann der Anwender die Anzahl der auszugebenden Pulse einstellen. Dann wird der Kalibriermodus ausgeschaltet und alle Zähler werden auf null gesetzt. Anschließend startet die CCU8. In der ersten Periode ist der Duty Cycle 0, d.h. während der gesamten Periode bleibt der Ausgang im LOW-Zustand. In Abbildung 10 ist erkennbar, dass das Programm keine Funktionen ausführt, bis ein „Period Match“-Interrupt aufgerufen wird. Dieser Period-Match des hochgenauen Timers der CCU8 triggert die Aktualisierung des PWM-Duty-Cycles und das Auslösen eines neuen Pulses. So kann die PWM-Signalgenerierung in Echtzeit angepasst werden. Bei Erreichen der maximalen Pulsanzahl wird ein Stopp-Signal an die CCU8 gesendet und das Pulsen beendet.

Abbildung 11 zeigt die in diesem Modus generierten Signale beispielhaft bei einem Duty Cycle von 80 %, einer gewünschten Pulsanzahl von „5“ und einer PWM-Frequenz von 10 kHz. Die CCU8 wurde gestartet und in der ersten Periode wurde kein Signal geschaltet. Anschließend wurden fünf Pulse mit einem Duty Cycle von 80 % ausgegeben (blau). Eine Periode hat eine Breite von 100 μ s, was einer PWM-Frequenz von 10 kHz entspricht. In rot ist das invertierte Signal

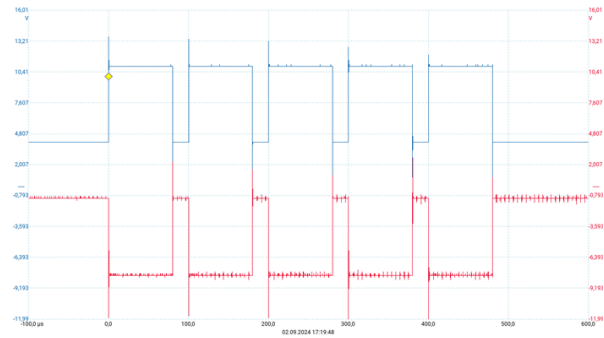


Abbildung 11. Fünf PWM-Pulse mit 80 % Duty Cycle.

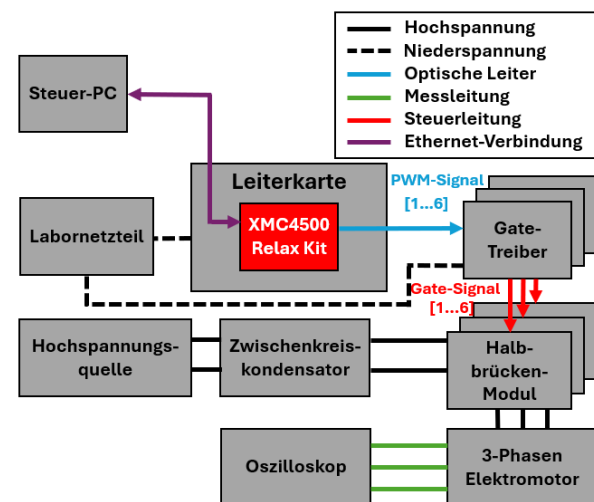


Abbildung 12. Blockschaltbild einer Open-Loop-Steuerung eines Elektromotors.

mit dem entsprechenden Duty Cycle von 20 % zu sehen.

2) *Modus 2: Dreiphasige SPWM ohne externe Synchronisation:* Dieser Modus dient der Erzeugung dreiphasiger SPWM-Signale, mit denen dreiphasige Wechselrichter angesteuert werden können. Ein möglicher Anwendungsfall wäre der Betrieb eines dreiphasigen Elektromotors in einer Open-Loop-Steuerung. Das bedeutet, dass die Signale an den Motor ausgegeben werden, ohne dass Messdaten wie Drehzahl oder Position zum Controller zurückgeführt werden. Abbildung 12 zeigt das entsprechende Blockschaltbild, welches Abbildung 6 entspricht, außer dass der Versuchsaufbau einen drei-Phasen Elektromotor anstatt der dreiphasigen Drosselspule enthält. Die drei Halbbrückenmodule bilden zusammen einen dreiphasigen Wechselrichter, der, gesteuert von den Signalen des XMC4500 Relax Kits über drei Gate-Treiber, einen Drei-Phasen-Elektromotor ansteuert.

Abbildung 13 zeigt den Programmablauf in diesem Modus: Drei CCU8-Slices steuern die drei Gate-Treiber und damit die drei Halbbrückenmodule über die optischen Ausgänge U-Top, U-Bot, V-Top, V-

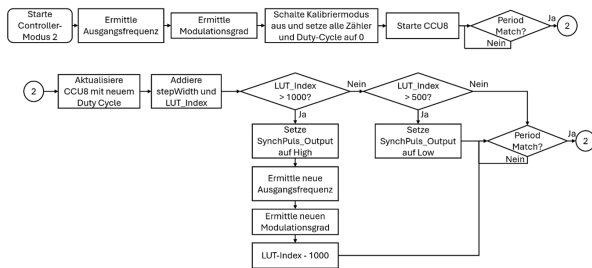


Abbildung 13. Ablaufdiagramm in Modus 2: Dreiphasige SPWM ohne externe Synchronisation.

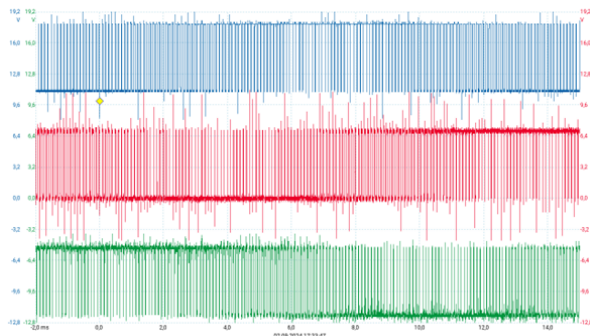


Abbildung 14. Messung von drei um 120° versetzten SPWM-Signalen.

Bot, W-Top und W-Bot (siehe Abbildung 7). Zunächst werden die PWM-Frequenz, die Falling und Rising DeadTime und die Bereiche, in denen der Modulationsgrad und Grenzen der Sinusfrequenz eingestellt werden können, an den Controller übergeben. Wird dem Controller ein Start-Befehl übermittelt, berechnet der Controller die Frequenz des modulierten Signals und den Modulationsgrad. Anschließend starten die drei CCU8-Slices, wobei während der ersten Periode, wie in Modus 1, keine Signale ausgegeben werden.

Es werden nach jeder PWM-Periode timergesteuert im Period-Match-Interrupt die neuen PWM-Duty-Cycle-Werte aus der LUT ausgelesen und in den CCU8-Slices aktualisiert. Zudem werden die Positionen in der LUT gezählt, um darüber ein rechteckiges Synchronisationssignal mit Ausgangs- bzw. Modulationsfrequenz zu generieren und auszugeben, auf das sich weitere externe Baugruppen synchronisieren können. Bei Erreichen des Endes der LUT werden die eventuell zwischenzeitlich aktualisierten Parameter für die Ausgangsfrequenz und den Modulationsgrad ausgewertet und eine neue LUT berechnet.

Abbildung 14 zeigt beispielhaft die in diesem Modus generierten Signale bei ein Modulationsgrad von $m = 90\%$, einer Sinusfrequenz von 50 Hz und einer PWM-Frequenz von 10 kHz. Man erkennt, in welchen Bereichen ein SPWM-Signal einen hohen oder einen niedrigen Duty Cycle hat.

Mit Hilfe der „Positive-Duty-Cycle“-Funktion im Oszilloskop wurden die PWM-Signale aus Abbil-

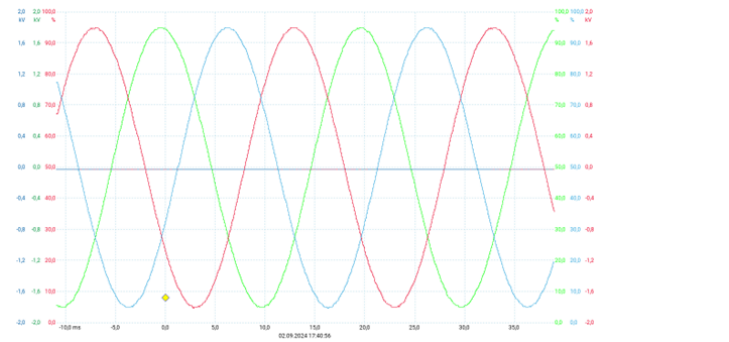


Abbildung 15. Positive Duty Cycle der generierten SPWM-Signale.

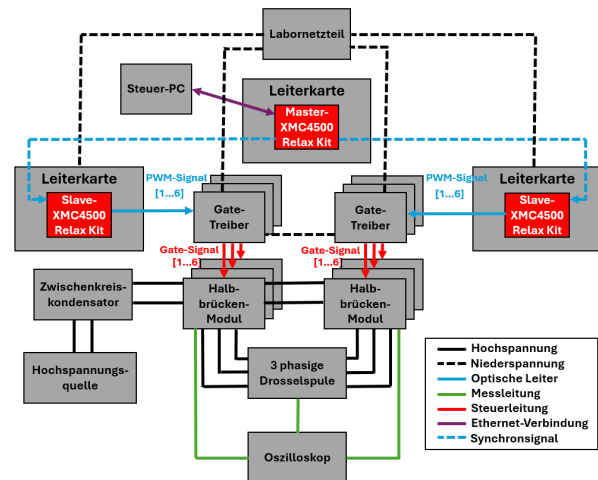


Abbildung 16. Blockschaftbild in Modus 3 und 4: Dreiphasige SPWM.

dung 14 demoduliert und in Abbildung 15 dargestellt. Es sind drei um 120° versetzte Sinusverläufe mit gleicher Amplitude zu erkennen. Jede Sinusperiode ist 20 ms lang, was der eingestellten Frequenz von 50 Hz entspricht.

3) *Modus 3: Dreiphasiger SPWM Slave:* In diesem Modus werden, wie in Abbildung 16 gezeigt, zwei XMC4500 Relax Kits im Slave-Modus durch ein externes Synchron-Signal angesteuert, das von einem XMC4500 Relax Kit im Master-Modus generiert wird. Wie bereits in dem in Abbildung 6 dargestellten Szenario steuern je drei CCU8-Slices eines Slave-Controllers über ihre optischen Ausgänge U-Top, U-Bot, V-Top, V-Bot, W-Top und W-Bot die drei Gate-Treiber des dreiphasigen Wechselrichters bestehend aus drei Halbbrückenmodulen an. Im Folgenden wird der Slave-Betrieb der beiden XMC4500 Relax Kits beschrieben, die Beschreibung des Master-Modus erfolgt in Abschnitt V-B4.

Zwei Slave-Controller befinden sich im synchronen Betrieb, wodurch zwei dreiphasige Wechselrichter synchron angesteuert werden können. Wenn beide Wechselrichter identische Sinusverläufe an die dreiphasige Drosselspule anlegen, fließt lediglich ein kleiner Rip-

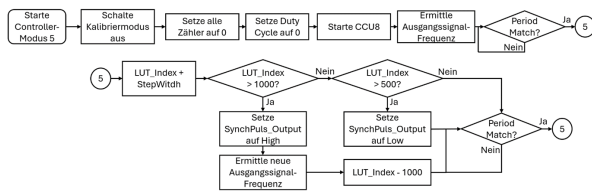


Abbildung 19. Ablaufdiagramm in Modus 4: Dreiphasige SPWM Master.

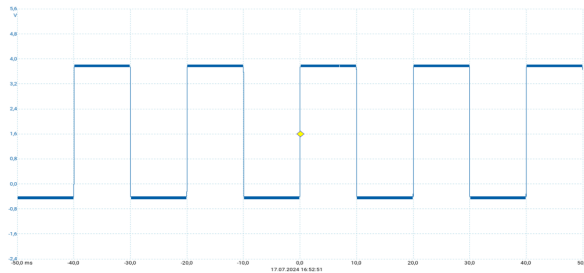


Abbildung 20. Vom Master-Controller ausgegebenes 50 Hz Synchronsignal.

4) *Modus 4: Dreiphasiger SPWM Master:* Dieser Modus wird als reiner Frequenzgeber verwendet, auf den sich andere Controller oder sonstige externe Baugruppen synchronisieren können. Das Blockschaltbild aus Abbildung 16 zeigt oben ein XMC4500 Relax Kit im Master Modus und Abbildung 19 den zugehörigen Programmablauf. Die beiden Synchronisationssignale für die beiden Slave-Controller aus Abbildung 16 (links und rechts) werden nun von dem XMC4500 Relax Kit oben, das sich im Master Modus befindet, erzeugt.

Bei einem eingehenden Start-Befehl wird ein CCU8-Slice gestartet. In diesem Modus wird nurmehr ein Synchronsignal generiert, welches am optischen Ausgang „Synch“ ausgegeben wird. Die Frequenz des Synchronsignals wird mit der am Analogeingang_2 anliegenden Spannung unter Berücksichtigung der eingegebenen Grenzen der Sinusfrequenz berechnet. Dieser Modus ähnelt stark dem Modus 2, jedoch erzeugt der Controller selbst nur das Synchronisationssignal und keine PWM-Ausgangssignale. Er könnte als Master-Controller zukünftig zusätzlich zu seiner Synchronisationsfunktion für Slave-Controller für verschiedenste weitere Aufgaben erweitert werden, seien es sicherheitsrelevante Features oder kontinuierliche Messaufgaben im Testaufbau.

Ein Beispiel wie das von diesem Modus ausgegebene Signal aussehen kann, ist in Abbildung 20 dargestellt. Es handelt es sich um ein Rechtecksignal mit einem konstanten Duty Cycle von 50 % und einer Periodendauer von 20 ms, was einer Frequenz von 50 Hz entspricht, wie dem Controller vorgegeben.

VI. ZUSAMMENFASSUNG

In dieser Arbeit wurde ein multifunktionaler PWM-Modulator basierend auf dem XMC4500 Relax Kit von Infineon Technologies entwickelt und getestet, mit dem Ziel, eine kostengünstige synchrone Modulation von mehreren 2-Level-Umrichtern zum Zweck des gezielten Test von Leistungshalbleitern bei einstellbarem $\cos(\varphi)$ zu ermöglichen (Back-To-Back-Betrieb). Auf dem Controller des XMC4500 Relax Kit wurde ein sinusförmiges Pulsweitenmodulationsverfahren implementiert. Dabei wird auf eine Look-Up-Table (LUT) zugegriffen, in der die Einträge mit Spannungswerten beschrieben sind, die einen Sinus modulieren. Die SPWM-Implementierung wurde mit der Ansteuerung einer H-Brückenschaltung und eines dreiphasigen 2-Level-Wechselrichters erfolgreich validiert und es wurden in ausführlichen Tests ein- und dreiphasige Sinusströme generiert. Die Controller können mit derselben Software aufgrund verschiedener Betriebsmodi sowohl als Master zur Generierung von Synchronisationssignalen, als auch als Slave zur Ansteuerung der Wechselrichter verwendet werden. Dies wegen der Analogeingänge auch ohne bestehende Verbindung zu einem Rechner, was im Versuchsaufbau validiert wurde. Durch die Flexibilität der Software und des Controllers können viele praxisnahe Szenarien eingestellt werden. Die Hardwarekosten der vorgestellten Lösung liegen bei ca. 235 € und sind damit um den Faktor 6 - 20 niedriger als bei Nutzung kommerzieller Generatoren oder Pulserkarten. Damit stellt der entwickelte multifunktionale PWM-Modulator eine weiter ausbaufähige, flexible und kostengünstige Lösung für unterschiedliche applikationsnahe Tests von Leistungshalbleitern dar.

DANKSAGUNGEN

Besonderer Dank gilt Herrn Dipl.-Ing. Dirk Brieke und Herrn Andreas Schmal für die Unterstützung bei dieser Arbeit.

LITERATURVERZEICHNIS

- [1] Plexim, „Plexim Real Time Box“, https://www.plexim.com/de/products/rt_box [abgerufen am: 25.02.2025].
- [2] National Instruments Corporation, „NI PCIe-6612, Counter/Timer-Modul“, <https://www.ni.com/de-de/shop/model/pcie-6612.html> [abgerufen am: 25.02.2025].
- [3] M. Greschuck, „Entwicklung und Implementierung einer frei programmierbaren mikrocontollerbasierenden Ansteuerung für Dauerschaltversuche an Leistungshalbleitern“, Bachelorarbeit, HS Düsseldorf, Düsseldorf, 2022. [abgerufen am: 01.10.2024].
- [4] B. Bagaber, „Build of Automated Test System for Repetitive Pulse Test“, Internship Report, RWTH Aachen, Aachen, 2016. [abgerufen am: 01.10.2024].
- [5] U. Bormes, „Entwicklung und Implementierung eines Teststandes mit variablen Belastungszyklen zur Untersuchung von Leistungshalbleitermodulen“, Masterarbeit, University of Applied Sciences Münster, Münster, 2017. [abgerufen am: 01.10.2024].
- [6] J. Pfeiffer, „Lifetime predictions for power modules under overload conditions for different applications“, Masterarbeit, Fachhochschule Südwestfalen, Südwestfalen, 2018. [abgerufen am: 01.10.2024].
- [7] M. Pischtschan, „Kapitel 9 Umrichtertechnologie | Skript Antriebsregelungstechnik“, https://bookdown.org/martin_pischtschan/skript_atrt1/umrichtertechnologie.html [abgerufen am: 27.09.2024].
- [8] U. Probst, „Leistungselektronik für Bachelors: Grundlagen und praktische Anwendungen : mit 83 Beispielen und 75 Übungen“, 3rd ed. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2015.
- [9] U. Tietze and C. Schenk, „Halbleiter-Schaltungstechnik: [neuer Teil: nachrichtentechnische Schaltungen]“, 12th ed. Berlin, Heidelberg: Springer, 2002. <https://swbplus.bsz-bw.de/bsz098304070err.htm>
- [10] H. Glasmachers, „Lerninhalte und Abschlussarbeiten - PWM“, https://www.professorglasmachers.de/le_pwm/ [abgerufen am: 04.10.2024].
- [11] A. Kleimaier, „Leistungselektronik“, https://www.alexander-kleimaier.de/vorlesung/VorlesungLE_Kapitel6.pdf, Hochschule Landshut [abgerufen am: 24.01.2025].
- [12] H. Selent, U. Probst „Sinusbewertete Pulsweitenmodulation“, <https://homepages.thm.de/~hg13555/Datenbank/lei/index.php/de/?catid=0&id=39> [abgerufen am: 29.09.2024].
- [13] H.-G. Eckel, „Einführung Leistungselektronik für Windenergieanlagen Kapitel 4 bis 6“, https://www.uni-rostock.de/storages/uni-rostock/UniHome/Weiterbildung/KOSMOS/E-Technik/Material_Ergebnisse_E-Technik/Lehrmaterial_Listungselektronik_fuer_Windenergieanlagen_Kapitel_4-6.pdf, Universität Rostock, [abgerufen am: 24.01.2025].
- [14] Z. Han, „Was ist ein Dreiphasen-Wechselrichter und wie funktioniert er?“, PowMr, 01 May., 2024. <https://powmr.com/de/blogs/news/what-is-three-phase-inverter> [abgerufen am: 29.09.2024].
- [15] J. Weidauer, „Elektrische Antriebstechnik: Grundlagen - Auslegung - Anwendungen - Lösungen“, 2nd ed. Erlangen: Publicis Publ, 2011.
- [16] „BoardUsersManualXMC4500RelaxKit-V1R12releasedpdf“, https://www.infineon.com/dgdl/Board_Users_Manual_XMC4500_Relax_Kit-V1_R1.2_released.pdf?fileId=db3a30433acf32c9013adf6b97b112f9 [abgerufen am: 24.01.2025].



ar 2025 arbeitet er als Firmware-Entwickler für FPGA-Entwicklung bei der NewTec GmbH.

Maik Greschuck schloss sein Bachelorstudium der Elektro- und Informationstechnik mit der Vertiefungsrichtung Mikroelektronik an der Hochschule Düsseldorf im Jahr 2022 ab. Sein Masterstudium mit der Vertiefungsrichtung Mikroelektronik an derselben Hochschule beendete er im Jahr 2024. Parallel zu seinem Masterstudium war er von 2022 bis 2024 als Werkstudent bei der Infineon Technologies AG im Bereich Leistungselektronik tätig. Seit Febru-



Clemens Elskamp schloss 2008 sein Diplomstudium der Informations- und Kommunikationstechnik Fachrichtung Mobilkommunikationstechnik an der Fachhochschule Dortmund und bei der Nokia GmbH Bochum ab. Im gleichen Jahr nahm er seine Tätigkeit bei der Infineon Technologies AG in Warstein als Testingenieur für Leistungshalbleitermodule auf. Seit 2012 ist er dort als Entwicklungsingenieur tätig.



seldorf als Professor für das Lehrgebiet Mikroelektronik berufen.

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

MULTI PROJEKT CHIP GRUPPE

Hochschule Aalen

Prof. Dr. Bürkle, (07361) 576-2103
heinz-peter.buerkle@htw-aalen.de

Hochschule Albstadt-Sigmaringen

Prof. Dr. Gerlach, (07571) 732-9155
gerlach@hs-albsig.de

Hochschule Esslingen

Prof. Dr. Lindermeir, (0711) 397-4221
walter.lindermeir@hs-esslingen.de

Hochschule Furtwangen

Prof. Dr. Benyoucef, (07723) 920-2342
bed@hs-furtwangen.de

Hochschule Heilbronn

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

Hochschule Karlsruhe

Prof. Dr. Ng, (0721) 925-1520
Herman-Jalli.Ng@hs-karlsruhe.de

Hochschule Konstanz

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

Technische 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. Hennig, (7121) 271-7101
eckhard.hennig@reutlingen-university.de

Technische Hochschule Ulm

Prof. Dr. Terzis, (0731) 96537-627
anestis.terzis@thu.de

www.mpc-gruppe.de