

# Sachbericht zum Verwendungsnachweis

## Teil 1 - Kurzbericht

**Verbundprojekt:** Simulationsbasiertes Entwurfswerkzeug zur automatisierten Überprüfung von analog-digitalen Elektroniksystemen - AUTOASSERT -

**Teilvorhaben:** Automatisierte Simulationsbasierte Überprüfung von heterogener virtueller Hardware/Software-Plattformen

| Zuwendungsempfänger: COSEDA Technologies GmbH |  |
|---|--|
| Vorhabenbezeichnung:                          | AUTOASSERT   |
| Förderkennzeichen:                            | 16ME0116K  |
| Projektlaufzeit:                              | 01.04.2020 bis 31.03.2023  |
| Erstelldatum:                                 | 08.06.2023   |
| Autor:  | Dr. Ing. Thilo Vörtler<br>Karsten Einwich  |
| Ansprechpartner:                              | Dr. Ing. Thilo Vörtler<br>Phone: +49 351 321490-15<br><a href="mailto:thilo.voertler@cosedatech.com">thilo.voertler@cosedatech.com</a> |

# Kurzbericht

## Aufgabenstellung

Mikroelektronische Hard- und Softwaresysteme sind die Schlüsselkomponente für die Mehrzahl der Innovationen in aktuellen und künftigen Systemen. Aufgrund des bisher noch nicht absehbaren Endes der Gültigkeit des Mooreschen Gesetzes, welches eine regelmäßige Verdopplung des Integrationsgrades prognostiziert, sind immer komplexere Systeme integrierbar und extrem kostengünstig zu produzieren. Diese sind der Enabler für viele aktuelle Technologien von Industrie 4.0, IoT bis hin zu autonomem Fahren.

Diese Systeme bestehen aus einer Vielzahl von Hardware- und Softwarekomponenten. Durch die ständig ansteigende Komplexität moderner Systeme muss der Entwurfseinstieg zunehmend auf hohen Abstraktionsebenen begonnen werden. Es werden sogenannte *virtuelle Prototypen* (engl. Virtual Prototypes, VPs) erstellt. Diese ermöglichen es, die Hardware- und Softwareentwicklung zu parallelisieren, da die Software für das Zielsystem bereits mit Hilfe des VPs, d.h. ohne, dass die reale Hardware zu Verfügung steht, programmiert und getestet werden kann.

Allerdings stellt auf Grund der enormen Komplexität heutiger bzw. zukünftiger Systeme die Verifikation den Flaschenhals dar. Die Aufgabe der Verifikation ist es zu überprüfen, ob das System seine Spezifikation erfüllt. Für die Verifikation wird der virtuelle Prototyp simuliert. Dafür sind (1) geeignete Eingaben an den virtuellen Prototyp (sogenannte Stimuli) zu beschreiben und (2) das erwartete Ausgabeverhalten zu prüfen.

Moderne digitale Verifikationsmethoden beruhen darauf, dass sie anstatt von gerichteten Tests, welche genau ein Szenario beschreiben und testen, zu Constraint Randomized und Assertion based Tests übergehen. Dabei wird zur Stimulierung ein Stimuliraum (z.B. für die Datenübertragung auf dem Bus müssen Adressen immer an der Wortgrenzen ausgerichtet sein und valide Daten Pakete enthalten unterschiedliche Quell- und Zieladressen) beschrieben und zur Überprüfung Regeln (Assertions) formuliert (z.B. auf ein Request muss immer ein Acknowledge folgen oder ein Abbruch innerhalb von 4 Taktzyklen). Damit kann der Simulator automatisch Szenarien erzeugen, simulieren und die Ergebnisse kontinuierlich überprüfen. Somit steigt mit der Simulationszeit die Testabdeckung ohne, dass weiterer Implementierungsaufwand erforderlich ist.

## Projektziele und Projektlauf

Ziel des Projektes AUTOASSERT war es **Assertion-basierte Verifikationsmethoden für den Systementwurf verfügbar und anwendbar zu machen**. Dazu sollte die in der rein digitalen Welt sehr erfolgreiche Assertion-basierte Verifikation so weiterentwickelt werden, dass diese für eine Gesamtsystembetrachtung auf der Systemebene nutzbar wird. Zusätzlich sollte die Methodik zur Verifikation von analogen zeitkontinuierlichen und mixed-signal Komponenten erweitert werden. Die Hauptziele des Projektes waren dabei die:

- Entwicklung einer Assertion-Bibliothek für heterogene Systeme,
- Entwicklung neuer Methoden für Gesamtsystem-Assertions, d.h. Überprüfung von analogem und digitalem Verhalten in Kombination mit eingebetteter (Steuerungs-) Software,
- Entwicklung von Debugging Methoden von fehlgeschlagenen Assertions,
- Entwicklung einer Methodik zur sprach- und Ebenen übergreifenden Verifikation.

Innerhalb des Projektes arbeiteten die **COSEDA Technologies GmbH** und das **Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI)** zusammen. Die COSEDA Technologies GmbH ist ein Startup, welches mit der Entwurfsumgebung COSIDE® eine Entwicklungsumgebung für die Entwicklung von virtuellen Prototypen, basierend auf SystemC AMS Modellen entwickelt und vermarktet.

Das Projekt wurde in 4 Arbeitspakete unterteilt, welche gemeinsam von COSEDA und dem DFKI bearbeitet wurden:

- AP1: Entwicklung einer systemübergreifenden Assertion-Methodik für heterogene Systeme
- AP2: Implementierung einer Assertion-Bibliothek für SystemC AMS
- AP3: Mixed Language Verification und Tooling
- AP4: Evaluation der Ergebnisse

In AP1 wurden die methodischen Grundlagen und die Anforderungen für die Entwicklung der Assertion-Bibliothek entwickelt. Dabei wurden insbesondere die industriellen Anforderungen der Kunden der COSEDA Technologies GmbH berücksichtigt.

AP2 diente der Implementierung der Assertion Bibliothek für SystemC AMS. Hierbei wurde einerseits ein Backend entwickelt, welches die Assertions innerhalb einer SystemC Simulation ausführt und auch heterogene virtuelle Prototypen unterstützt. Weiterhin wurde eine Nutzerfreundliche API für Assertions entwickelt, welche sich an SystemVerilog Assertions orientiert.

AP3 untersuchte, wie die Assertion Bibliothek für andere Implementierungsebenen z.B. SystemVerilog verwendet werden konnte. Dabei konnte die prototypische Integration in die Cadence Incisive Simulationsumgebung als auch die Anbindung des Verilator Tools gezeigt werden.

AP4 diente der Evaluation der Projektergebnisse und es wurden Demonstratoren erstellt. Hierbei wurde insbesondere darauf geachtet, dass die im Projekt erstellte Assertionbibliothek auch in Gremien wie Accellera standardisiert werden kann.

## **Projektergebnisse**

In AUTOASSERT wurde somit eine neue Technologie für System Level Assertions entwickelt, die es ermöglicht, komplexe Systeme zu verifizieren und zu debuggen. Die Technologie basiert auf einer SystemC basierten Eingabesprache für Assertions, die eine hohe Abstraktionsebene und eine einfache Integration in bestehende SystemC Modelle bietet. Die Assertion Sprache wurde prototypisch in einer Bibliothek implementiert und anhand von Demonstrator Systemen u.a. basierend auf einem ARM-Prozessor evaluiert. Die Demonstrator Systeme zeigten die Vorteile der Assertion Technologie für die Fehlererkennung und -lokalisierung sowie für die Leistungsanalyse.

Zusätzlich wurden für die Assertion-Technologie auch prototypische Debugging- und Auswertungs-Tools für Assertions entwickelt, die eine visuelle Darstellung der Assertions ermöglichen.

Insgesamt wurden innerhalb des Forschungsprojekts 9 Peer Reviewed Veröffentlichungen erstellt, welche auf internationalen Konferenzen vorgestellt wurden.

Zur Verwertung ist geplant, die erstellte Bibliothek mit der Assertionsprache und den zugehörigen Tools, den Kunden von COSEDA Technologies zugänglich zu machen. Dadurch wird es COSEDA möglich ein effizientes und zuverlässiges System Level Verifikations-Tool anzubieten. Darüber hinaus soll die Assertionbibliothek innerhalb der Accellera Systems Initiative standardisiert werden, um eine breite Akzeptanz und Kompatibilität in der Industrie zu erreichen.

# Sachbericht zum Verwendungsnachweis

## Teil 2 – Eingehende Darstellung

**Verbundprojekt:** Simulationsbasiertes Entwurfswerkzeug zur automatisierten Überprüfung von analog-digitalen Elektroniksystemen - AUTOASSERT -

**Teilvorhaben:** Automatisierte Simulationsbasierte Überprüfung von heterogener virtueller Hardware/Software-Plattformen

| Zuwendungsempfänger: COSEDA Technologies GmbH |  |
|---|--|
| Vorhabenbezeichnung:                          | AUTOASSERT   |
| Förderkennzeichen:                            | 16ME0116K  |
| Projektlaufzeit:                              | 01.04.2020 bis 31.03.2023  |
| Erstelldatum:                                 | 08.07.2023   |
| Autoren:                                      | Dr. Ing. Thilo Vörtler<br>Karsten Einwich  |
| Ansprechpartner:                              | Dr. Ing. Thilo Vörtler<br>Phone: +49 351 321490-15<br><a href="mailto:thilo.voertler@cosedatech.com">thilo.voertler@cosedatech.com</a> |

# 1 Ziele und Aufgaben des Projekts

## 1.1 Ausgangssituation, unter denen das Vorhaben durchgeführt wurde

Mikroelektronische Hardware- und Softwaresysteme sind zentral für die meisten Innovationen in aktuellen und zukünftigen Systemen. Das anhaltende Mooresche Gesetz [1] ermöglicht immer komplexere und kostengünstigere Systeme, die entscheidend für Technologien wie Industrie 4.0, IoT und autonomes Fahren sind.

Mit der steigenden Integrationsdichte wächst auch die Systemkomplexität. Um zu verhindern, dass dies zu exponentiell wachsenden Entwurfsaufwänden und Kosten führt, ist es notwendig, fortlaufend neue Entwurfsmethoden zu erforschen und einzuführen. Hierbei spielt der modellbasierte Entwurf und die Verifikation eine entscheidende Rolle [9].

Aufgrund der zunehmenden Komplexität und erhöhter Sicherheitsanforderungen steigt der Aufwand für die Verifikation auf allen Abstraktionsebenen. Daher ist es unerlässlich, den Verifikationsprozess auf der Spezifikations- oder Systemebene zu starten und dabei die Software und die Anwendungsszenarien miteinzubeziehen.

Künftige Technologien erfordern eine intensivere Zusammenarbeit entlang der Wertschöpfungskette. Beispielsweise werden neue Lidar-Systeme von spezialisierten Systemhäusern konzipiert, während die tatsächliche Implementierung in herkömmlichen Designhäusern erfolgt. Effektive Zusammenarbeit ist daher für den Erfolg, insbesondere in der Elektronikentwicklung für autonomes Fahren, entscheidend.

COSEDA Technologies bietet Softwaretools, welche die Erstellung, Verifizierung und den Austausch von virtuellen Prototypen für analoge/digitale Hardware-/Softwaresysteme kostengünstig ermöglichen. Die Verifikation nimmt dabei einen immer größeren Stellenwert ein und stellt oft den Engpass auf Systemebene dar.

Zur Verbesserung der Effizienz und Qualität des Verifikationsprozesses existieren leistungsstarke Technologien. Moderne digitale Verifikationsmethoden setzen dabei auf constraint randomisierte [7],[8] und assertionbasierte Tests, die automatisch Szenarien erstellen und überprüfen, wodurch die Testabdeckung mit der Simulationsdauer steigt, ohne dass zusätzlicher Implementierungsaufwand erforderlich ist.

Die COSEDA Technologies GmbH konnte auf eine langjährige Erfahrung bei der Erstellung und Simulation von virtuellen Prototypen auf Systemebene zurückblicken und hatte mit der Entwurfsumgebung COSIDE® eine Entwicklungsumgebung für die Entwicklung von virtuellen Prototypen basierend auf SystemC AMS Modellen entwickelt und vermarktet.

Das Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI) verfügte als Forschungseinrichtung über langjährige Erfahrungen auf dem Gebiet der formalen und funktionalen Systemverifikation, insbesondere des eigenschaftsbasierten Testens (und konnte diese im Rahmen des Projekts AUTOASSERT erheblich ausbauen).

Diese komplementären Kompetenzen waren die Voraussetzung für den erfolgreichen Abschluss des Projekts.

## 1.2 Wissenschaftliche und technische Ziele des Vorhabens

Ziel des Projektes AUTOASSERT war es **Assertion-basierte Verifikationsmethoden für den Systementwurf verfügbar und anwendbar zu machen**. Dies beinhaltet, dass digitale Techniken zum Test von analogen zeitkontinuierlichen und mixed-signal Komponenten erweitert werden müssen. Da aktuelle Systeme sehr stark von eingebetteter Software gesteuert werden, muss es auch möglich sein, interne Zustände der Software mit in die Verifikation einzubeziehen.

Die Arbeitsziele des Projektes AUTOASSERT umfassen die angewandte Erforschung von Assertion-basierten Verifikationsmethoden für analog/digitale Hard- und Softwaresysteme sowie ihre prototypische Implementierung und Integration in die Entwurfsumgebung COSIDE® von COSEDA Technologies. Das DFKI plant die Verifikationskompetenz im Bereiche Assertions und Mixed Signal Verifikation auszubauen. Im Projekt AUTOASSERT sollen die folgenden Techniken erforscht und nutzbar gemacht werden:

- Entwicklung einer Assertion-Bibliothek für heterogene Systeme basierend auf SystemC [2] und SystemC AMS[3]
- Entwicklung neuer Methoden für Gesamtsystem-Assertions, d.h. Überprüfung von analogem und digitalem Verhalten in Kombination mit eingebetteter (Steuerungs-) Software
- Entwicklung von Debugging Methoden von fehlgeschlagenen Assertions
- Entwicklung einer Methodik zur sprach- und ebenenübergreifenden Verifikation

Insgesamt konzentriert sich das Projekt auf die Erstellung virtueller Prototypen auf Systemebene, wobei in verschiedenen Bereichen erhebliche Fortschritte erzielt werden. Es umfasst die Erstellung einer Assertionsbibliothek für heterogene Systeme und die Formulierung innovativer Techniken für Gesamtsystemassertionen. Dazu gehört die integrierte Verifikation von analogem, digitalem und eingebettetem Softwareverhalten. Darüber hinaus werden Methoden zum Debuggen fehlgeschlagener Assertions entwickelt, um die Stabilität des Systems zu verbessern. Schließlich wird im Rahmen des Projekts eine Methodik entwickelt, die eine sprach- und ebenenübergreifende Verifikation ermöglicht und so die Kompatibilität und Interoperabilität zwischen verschiedenen Plattformen verbessert.

## 1.3 Planung und Ablauf des Vorhabens

Für das Projekt AUTOASSERT wurde eine Laufzeit von drei Jahren geplant. Der Start erfolgte im April 2020 und endete im März 2023.

Aus den zuvor beschriebenen Zielen ergaben sich besondere Aspekte, die das Projekt beim Arbeitsplan berücksichtigt hatte.

| Arbeitspaket   | Beginn | Ende | PM  | COSEDA | DFKI |
|--|--------|------|-----|--------|------|
| AP1 - Entwicklung einer systemübergreifenden Assertion-Methodik für heterogene Systeme | 1      | 24   | 24  | 12     | 12   |
| AP2 - Implementierung einer Assertion-Bibliothek für SystemC AMS                       | 4      | 36   | 32  | 20     | 12   |
| AP3 - Mixed Language Verification und Tooling  | 13     | 33   | 28  | 24     | 4    |
| AP4 - Evaluation der Ergebnisse  | 7      | 36   | 24  | 16     | 8    |
|  | Summe  |      | 108 | 72     | 36   |

Die Arbeitspakete wurden von beiden Projektpartnern bearbeitet, wobei sich das DFKI vor allem auf die Entwicklung der Grundlagen, Algorithmen und die Bereitstellung von Basisbibliotheken und COSEDA Technologies auf deren Umsetzung und prototypische Integration konzentrierte.

Das Projekt verlief weitgehend wie geplant. Aufgrund der Aufmerksamkeit, die den Projektergebnissen zuteilwurde, war es möglich, mit der Diskussion über den Accellera-Standardisierungsprozess zu beginnen. Aufgrund der erhaltenen Rückmeldungen und der daraus resultierenden Diskussion kam es zu einigen Anpassungen in den Bibliotheken. Die dadurch entstehenden Arbeiten konnten jedoch in den Arbeitsplan integriert werden. Somit konnte das Projekt sowohl im Zeitplan als auch im Budget abgeschlossen werden.

## 2 Technische Ergebnisse

In diesem Abschnitt werden die Technischen Ergebnisse des AUTOASSERT-Projektes beschrieben. Dabei gehen wir auf die Arbeitspakete (AP) mit den entsprechend für COSEDA definierten Teilaufgaben (TA) ein.

### 2.1 AP1 Entwicklung einer systemübergreifenden Assertion-Methodik für heterogene Systeme

In diesem AP sollten die methodischen Grundlagen und die Anforderungen für die Entwicklung der Assertion-Bibliothek entwickelt werden. Dabei wurden insbesondere die industriellen Anforderungen der COSEDA Nutzer berücksichtigt.

Bei der Entwicklung von Mixed-Signal Systemen muss insbesondere das domänenübergreifende (Zeit-, Frequenz-) Systemverhalten berücksichtigt werden. Dazu war es notwendig die Datentypen auszuwählen, die das Zusammenspiel der analogen und gemischten Signalkomponenten vollständig darstellen können.

Darüber hinaus wurden Anforderungen aus unterschiedlichen Branchen wie Automobil, Internet of Things, Industrie 4.0 untersucht, und in die Assertions-Bibliothek integriert. Unter Berücksichtigung des IEEE SystemVerilog Standards[4] wurden die Prinzipien für die Assertion Bibliothek erarbeitet, so dass diese automatisiert während der Simulation des Systems geprüft werden können. Da analoge, digitale und Softwarekomponenten gleichzeitig interagieren, ist dabei eine für den Nutzer einfach zu verstehende aber gleichzeitig ausdrucksstarke Assertionssprache von entscheidender Bedeutung.

#### Teilaufgabe 1.1.1 Definition der Anforderungen der Assertion-Bibliothek für heterogene Gesamtsysteme

Um die Anforderungen zu ermitteln wurden Systeme untersucht, welche für die industriellen Anwender von COSEDA relevant sind. Dabei wurden u.a. folgende Anforderungen definiert:

- Assertions sollen direkt in SystemC/C++ ohne zusätzlichen Parser geschrieben werden.
- Syntax soll es erlauben einfach Assertions aus bestehen Sprachen (SystemVerilog Assertions-SVA) zu übertragen.
- Beschränkung auf simulationsbasierter Überprüfung (keine formale Verifikation).
- Unterstützung von Ereignissen zur Beschreibung analoger Systeme.
- Kompatibilität zu bestehenden IEEE SystemC und SystemC AMS Standards

Die finale Bibliothek sollte eine weitere Bibliothek sein welche auf die bestehenden Industriestandards, welche in der EDA-Branche üblich sind aufsetzen.

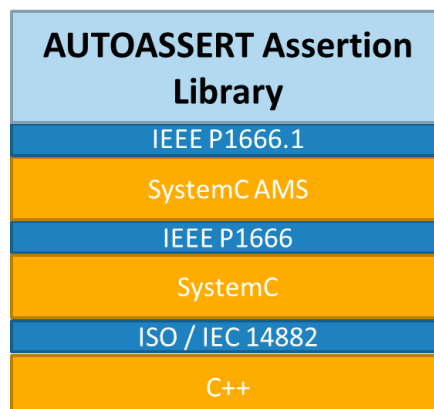


Abbildung 1: Assertion-Bibliothek aufsetzend auf bestehenden Industriestandards

### **Teilaufgabe 1.1.2 Untersuchung der zu unterstützenden Domänen (Zeit/Frequenzbereich, benötigte Datentypen)**

Es wurden Festlegung hinsichtlich der zu unterstützenden Domänen getroffen. Hierbei wurde festgelegt, dass die Assertion-Sprache insbesondere dynamische Simulationseignisse (Events) unterstützen muss. Dies erweitert bestehende Konzepte und ermöglicht es Assertions auch für analoge Systeme zu benutzen. Als Datentypen und Eingangssignale sollen alle von SystemC AMS unterstützen Datentypen verwendet werden. Dies betrifft insbesondere Fließkomma Datentypen.

### **Teilaufgabe 1.1.3 Ermittlung industrieller Anforderungen für die Assertion-Bibliothek**

Bei der Ermittlung industrieller Anforderungen wurde eng mit Kunden von COSEDA zusammengearbeitet. Hierbei hat sich u.a. gezeigt, dass eine einfache Einarbeitung der Nutzer in die Assertion Sprache nötig ist. Deswegen wird die Assertion Sprache sich an den Konzepten von SystemVerilog orientieren. Weiterhin wurde durch regelmäßige Teilnahme der Projektpartner an der Standardisierung von SystemC innerhalb von Accellera der Bedarf und Stand der Technik ermittelt.

Auch durch die Teilnahme an Industriekonferenzen wie der DVCon Europe wurde Feedback eingeholt. So wurde sichergestellt, dass die Entwicklung der Assertion Bibliothek den Anforderungen der Industrie entspricht.

Im Laufe des Projekts wurde eine von Intel vorgeschlagene Assertion-Bibliothek für digitale Systeme berücksichtigt. Dieser Vorschlag bezieht sich auf die Generierung von SystemVerilog Assertions für rein digitale Systeme bei der High Level Synthese digitaler Schaltungen [10].

## **2.2 AP2 Implementierung einer Assertion-Bibliothek für SystemC AMS**

In diesem AP wurde die Assertion-Bibliothek basierend auf der in AP1 in Zusammenarbeit mit dem DFKI entwickelten Assertion API implementiert. Herausforderungen dabei war die Berücksichtigung der industriellen Anforderungen von Kunden. Es wurden die grundlegenden Algorithmen zur Überprüfung von Assertions in SystemC implementiert, sowie entsprechende Unit Tests zur Qualitätssicherung implementiert.

### **Teilaufgabe 2.1.1 Festlegung industrieller Anforderungen für die Library Entwicklung**

Für die Library Entwicklung war es notwendig eine technische Basis zu nutzen, welchen Industriellen Anforderungen genügt. Als Compiler für die Demonstrator-Entwicklung wurde ein gcc 8.4 compiler sowie C++ 14 als Sprachstandard festgelegt. Die Entwicklung der Library sollte mit einem Continuous Integration basierten Regressionsuite erfolgen. Die Bibliothek soll sowohl für Windows als auch Linux einsetzbar sein.

Um die Assertion-Bibliothek kommerziell zu verwenden sind auch Lizenzrechtliche Anforderungen zu beachten, da der Assertion Code direkt in den Quellcode eines virtuellen Prototyps integriert wird. Somit dürfen keine Copyleft Lizenzen (u.a. GPL) verwendet werden. Die Bibliothek sollte deshalb keine externen Libraries verwenden welche nicht unter einer BSD oder Apache 2.0 Lizenz stehen.



## Teilaufgabe 2.1.2 Implementierung von temporalen Assertion Funktionen sowie für die Frequenzebene

Bei der Implementierung der Assertion Bibliothek wurde die Implementierung in Frontend und Backend unterteilt. Dabei hat sich COSEDA zuerst auf das Backend Konzentriert.

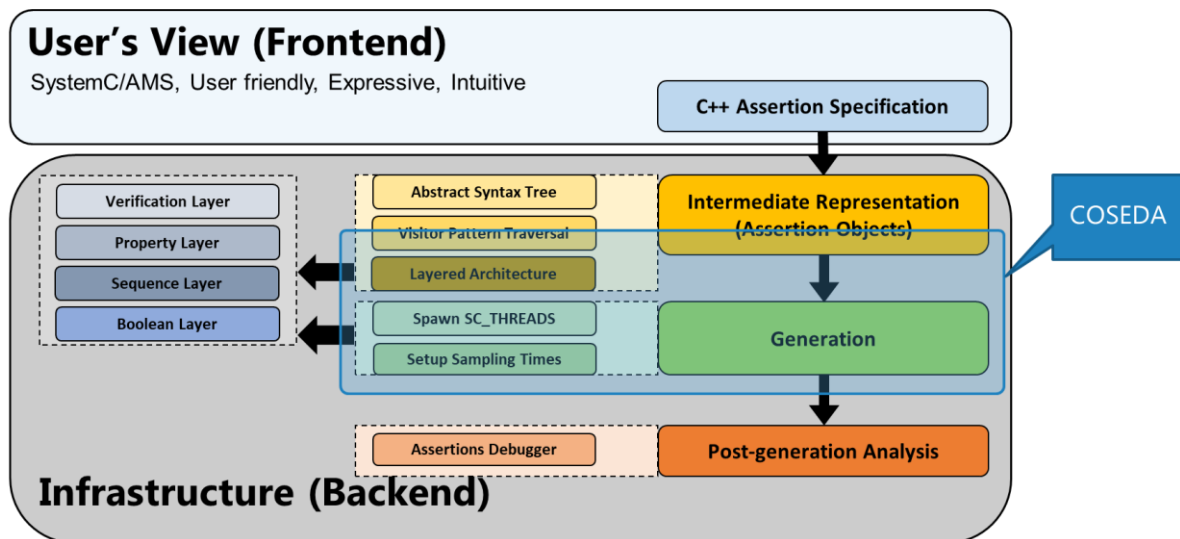


Abbildung 2: Übersicht der AUTOASSERT Assertion-Bibliothek

Zuerst wurden Algorithmen entwickelt die das temporale Verhalten, welches in Assertions umgesetzt wurde, auf den SystemC AMS Simulationskernel mappt. Dazu wurde zuerst eine Zwischensprache verwendet um die Algorithmen zur Überprüfung von Assertions zu entwickeln. Als Ausgangspunkt dienten dabei SystemVerilog Assertions. Die Ausführung der Assertions wurde durch das dynamische Erstellen von SystemC Prozessen umgesetzt.

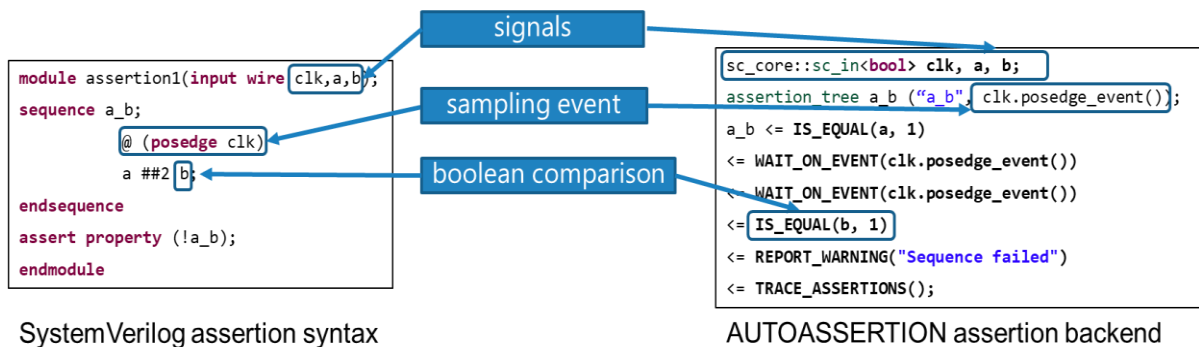


Abbildung 3: Mapping von SystemVerilog Assertions auf AUTOASSERT assertion backend

Zusätzlich zu rein digitalen Assertions unterstützt das Backend auch zeitkontinuierliches Verhalten, welches für AMS-Systeme benötigt wird.

Für den Zugriff auf andere Domänen (FFT-Analyse) wurde ein Konzept für den Zugriff auf Simulationsdaten via Lambda-Funktionen erarbeitet. Dies bildet die Grundlage für die Erweiterung der Assertion API für andere Domänen.

Nachdem die Grundlagen für die Verwendung von Assertions in SystemC AMS geschaffen wurden, lag der weitere Schwerpunkt nun auf der einfachen Verwendung der Assertions durch eine benutzerfreundliche API. Dazu wurden in der Zusammenarbeit mit dem DFKI (Teilaufgabe 2.2.1) Algorithmen entwickelt welche die Front-End Sprache in das Backend mit Hilfe eines Visitor Patterns übersetzen.

### Teilaufgabe 2.1.3 Bereitstellung für Zugriffsfunktionen für SystemC AMS Datentypen und Berechnungsmodelle

Durch die Erweiterung von digitalen Assertions auf System Assertions welche mit analogen Werten arbeiten werden entsprechende Zugriffsfunktionen benötigt, welche es erlauben auf Daten des SystemC Kernels zu beliebigen Zeitpunkten innerhalb von Assertions zuzugreifen.

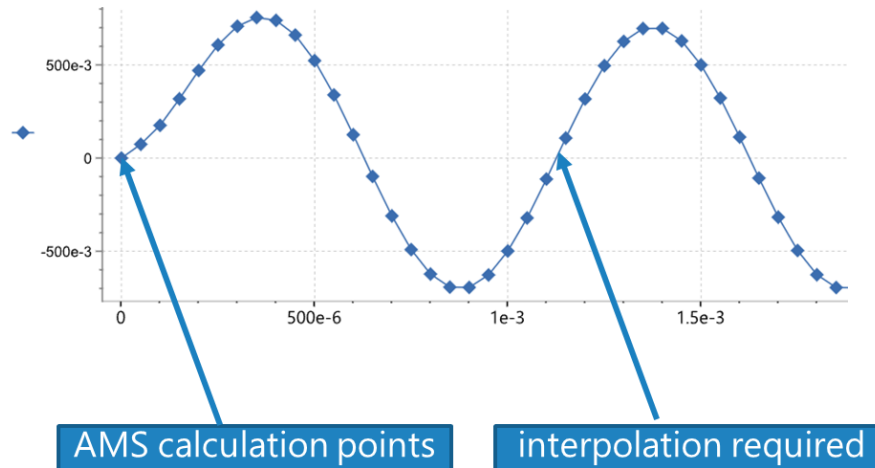


Abbildung 4: Zugriff auf Werte außerhalb von SystemC AMS Berechnungspunkten

So ist es z.B. möglich mit Hilfe des *delay* Operators innerhalb von Assertions Werte zu Zeitpunkten abzufragen, welche noch nicht berechnen wurden. Diese müssen entsprechend interpoliert werden. Es wurden Interfaces geschaffen welche es erlauben aus dem Assertion Backend auf solche Werte zuzugreifen.

### Teilaufgabe 2.1.4 Implementierung von Unit Tests

Während des Projekts wurde für die Überprüfung der Basisimplementierung und für die Zugriffsfunktionen ein C++ Unit Test Framework aufgesetzt. Dieses wurde während des Projekts kontinuierlich, mit fortlaufendem Stand der Implementierung, erweitert.

Dazu wurden unter anderem Unit Tests mit dem Framework GoogleTest implementiert, die auf einem Continuous Integration System ausgeführt werden können. Die Tests überprüfen insbesondere die API, die es ermöglicht, Assertions mit Hilfe von überladenen Operatoren auszudrücken. Zusätzlich wurden auch Integrationstests entwickelt, die das Zusammenspiel von komplexeren Assertions in einem SystemC AMS Modell untersuchen.

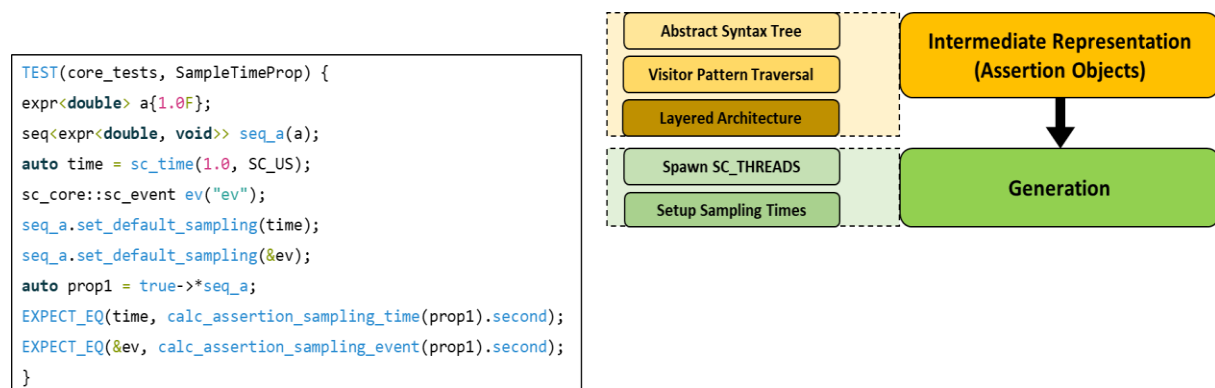


Abbildung 5: Beispiel für die Implementierung eines Unit-Tests zum Setzen der Samplingtime

## 2.3 AP3 Mixed Language Verification und Tooling

In diesem Arbeitspaket wurden Techniken entwickelt, welche es ermöglichen die Methodik auch auf anderen Implementierungsebenen (z.B. Register-Transferebene (RTL)) und den zugehörigen Sprachen wie SystemVerilog oder *specman e*[5] zu verwenden. Dazu müssen entsprechende Schnittstellen zu anderen Programmiersprachen entwickelt werden (T3.1.2). Diese soll es ermöglichen auf der Systemebene entwickelte Assertions auch für spätere Systemimplementierungsschritte wiederzuverwenden.

### Teilaufgabe 3.1.1 Evaluierung vorhandener Standards und Ansätze zum Multi language support von Verifikations IP

Die in AUTOASSERT entwickelte Assertion Bibliothek soll nicht nur auf der Systemebene, sondern auch der Implementierungsebene eingesetzt werden. Hierzu wurde untersucht wie die Assertions für die Verifikation einerseits auf RTL-Ebene (SystemVerilog/VHDL) eingesetzt werden können. Zusätzlich wurde die Anbindung an die Software-Domäne (u.a. mit ARM Fast Modellen, Abstrakte TLM-Modelle) untersucht. Dabei hat sich gezeigt, dass der Zugriff auf unterschiedliche Sprachen und Domänen machbar ist. Die Herausforderung liegt bei der Erstellung von Adaptern und Zugriffsfunktionen.

Aus der Sicht eines durchgängigen Entwurfsablaufs wurde der Hauptfokus auf die Weiterverwendung von Assertions auf der Implementierungsebene gelegt. Dadurch wird es möglich Assertions von der Systemebene auch in der Implementierungsebene (RTL) weiterzuverwenden.

### Teilaufgabe 3.1.2 Anbindung anderer Sprachen wie SystemVerilog

Zur Anbindung an SystemVerilog wurde ein Prototyp erstellt. Dazu wurden unterschiedliche Verilog Simulatoren (Verilator [6], Cadence Excellium, Xilinx Xsim) mit Hinblick auf die benötigten Adapter untersucht. Es wurde ein erster Prototyp für Cadence erstellt, der die Verwendung der AUTOASSERT Assertions im Zusammenspiel mit SystemVerilog zeigt. Dabei sind zwei Ansätze möglich:

- 1) Weiterverwendung von System-level-testbenches in der Implementierung. Hierbei sind die Assertions Teil einer SystemC basierten Testbench. Wenn im Laufe der Implementierung ein SystemC Modell durch eine Verilog-RTL Implementierung ersetzt wird können die Assertions verwendet werden um auch die RTL-Implementierung zu überprüfen.

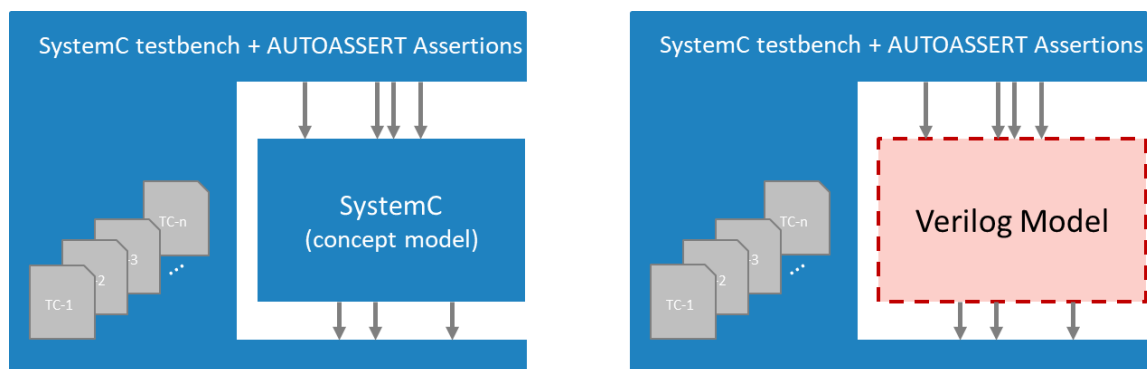


Abbildung 6: Weiterverwendung von Testbenches mit AUTOASSERT Assertions

- 2) Verifikation mit Hilfe von Sprachen wie „specman e“ oder SystemVerilog. In diesem Fall werden die Assertions in das SystemC Modell eingebunden. Dies ermöglicht es die AUTOASSERT Assertions mit bestehenden Verifikationsumgebungen zu verwenden.

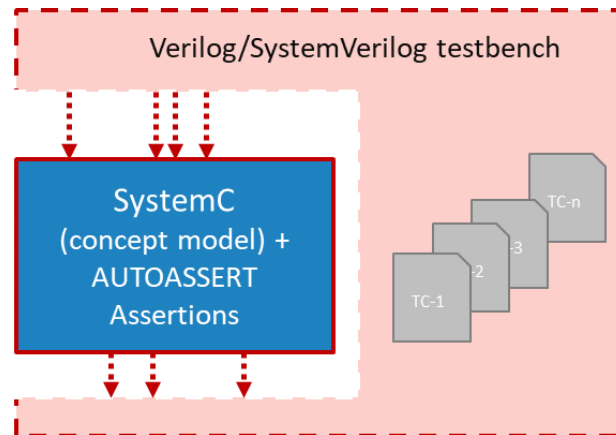


Abbildung 7: Verwendung von AUTOASSERT Assertions in anderen Verifikationsumgebungen

Zusätzlich zur Anbindung von Cadence-basierten Tools wurde eine Integration des Open-Source-Programms Verilator geschaffen. Dieses Programm ermöglicht die Simulation digitaler Verilog-Schaltungen auch innerhalb einer SystemC-Umgebung. Da Verilator jedoch noch keine Assertion-Sprache unterstützt, ermöglicht diese Integration die Nutzung der in AUTOASSERT entwickelten Assertionbibliothek zusammen mit Verilator. Dadurch wird der in Abbildung 6 gezeigte Ansatz möglich ohne einen weiteren digitalen Simulator zu koppeln.

### Teilaufgabe 3.1.3 Code Generierung für Wrapper Funktionen

Um die Cadence Tools und Verilator in SystemC AMS zu integrieren, wurden spezielle Wrapperfunktionen und Import Tools entwickelt. Für Cadence wurde eine Integration von SystemVerilog an SystemC geschaffen, wodurch es einfach wird, SystemVerilog-Designs mit SystemC-Designs zu mischen. Es wurde auch eine Code-Generierung für entsprechende Wrapper erstellt. Zusätzlich wurde ein Compile-flow erstellt, um eine gemischte Simulation von Cadence und SystemC Assertions zu ermöglichen. Ähnlich dazu wurde die Integration des Verilators durchgeführt. Hierfür wurde eine Import-Möglichkeit nach SystemC implementiert, um digitale Verilog-Schaltungen in SystemC-Umgebungen nutzen zu können.

### Teilaufgabe 3.1.4 Implementierung von Unit Tests für die Bibliotheken zur Anbindung anderer Sprachen als auch der Codegenerierung

Um sicherzustellen, dass die Integration von SystemVerilog Modellen gelingt, wurden Tests implementiert, welche die korrekte Generierung der Wrapper Funktionen überprüfen. Weiterhin wurden Integrationstests erstellt, welche überprüfen, dass mit dem Verilator importierte Designs sich auch innerhalb von SystemC AMS Systemen simulieren lassen.

## 2.4 AP4 Evaluation der Ergebnisse

Dieses Arbeitspaket diente der Evaluation der Projektergebnisse. Es wurden Demonstratoren erstellt, welche unter anderem für Standardisierungsgremien wie Accellera/IEEE verwendet werden. Zusätzlich wurde die Methodik Kunden von COSEDA vorgestellt und an Beispielen erprobt.

### Teilaufgabe 4.1.1 Aufbau von Demonstratoren zur Erprobung und Bewertung der Methodik

Die Erprobung der Assertion Methodik soll an einem möglichst realen kundenrelevanten System erfolgen. Dazu wurden mehrere Demonstrator-Beispiele aufgebaut. Diese dienten u.a. dem DFKI zur Analyse der Performance der Bibliothek in den Teilaufgaben 4.2.1 und 4.2.2.

Die Demonstratoren sind u.a.

- Ein RISC-V basiertes SPI welches über Register eines Prozessors angesprochen wird. Dieser Demonstrator wurde nach Cadence exportiert um die AUTOASSERT Assertions mit System-Verilog Assertions vergleichen zu können.
- ARM Fast Model basierter Demonstrator, welcher die Temperatursteuerung eines Hauses zeigt. Das System modelliert ein Temperaturregelungssystem, das mehrere Bereiche abdeckt, d.h. Software, digitale Hardware und analoges Verhalten. Das System ist in SystemC AMS modelliert und verwendet verschiedene Berechnungsmodelle, insbesondere Timed Data Flow (TDF), Electrical Linear Networks (ELN) und Transaction Level Modelling (TLM). Es besteht aus folgenden Komponenten:
  - eine ARM V8-basierte CPU mit ARM Fast Models, implementiert als SystemC TLM mit darauf laufendem Linux-Betriebssystem und SW,
  - vier ADT7420-Temperatursensoren, implementiert als SystemC AMS TDF und ereignisgesteuertem Berechnungsmodell,
  - einen AMBA-Bus (Advanced Microcontroller Bus Architecture), der als Brückengerät zur Verbindung von Temperatursensoren und ARM-Prozessor dient (erstellt in SystemC TLM) - (COS\_AMBA\_DEVICE in Abbildung 8),
  - ein Umgebungsmodell (Thermal\_Network), das 3 verbundene Räume und eine als Sinus (SIN\_SRC\_TDF) modellierte Umgebungstemperatur bildet, d. h. jeder Sensor erfasst eine andere Temperatur (Implementierung als SystemC/AMS ELN und ereignisgesteuertem Berechnungsmodell),
  - ein als SystemC AMS ELN implementiertes Heizungsmodell, das zur Steuerung der Temperatur verwendet werden kann.

Auf diesem System wurde eine Software ausgeführt, welche folgende Aufgaben übernimmt:

- Booten eines Linux-Betriebssystems auf dem ARM-Prozessor.
- Eine Kontroll-Software wird auf Linux ausgeführt. Diese überwacht kontinuierlich den Ausgang der Temperatursensoren.
- Wenn die Software feststellt, dass der Temperaturwert unter einen programmierten Schwellenwert fällt, schaltet sie das Heizgerät ein.
- Andernfalls, wenn die Temperatur einen bestimmten programmierten Schwellenwert überschreitet, wird das Heizgerät ausgeschaltet.

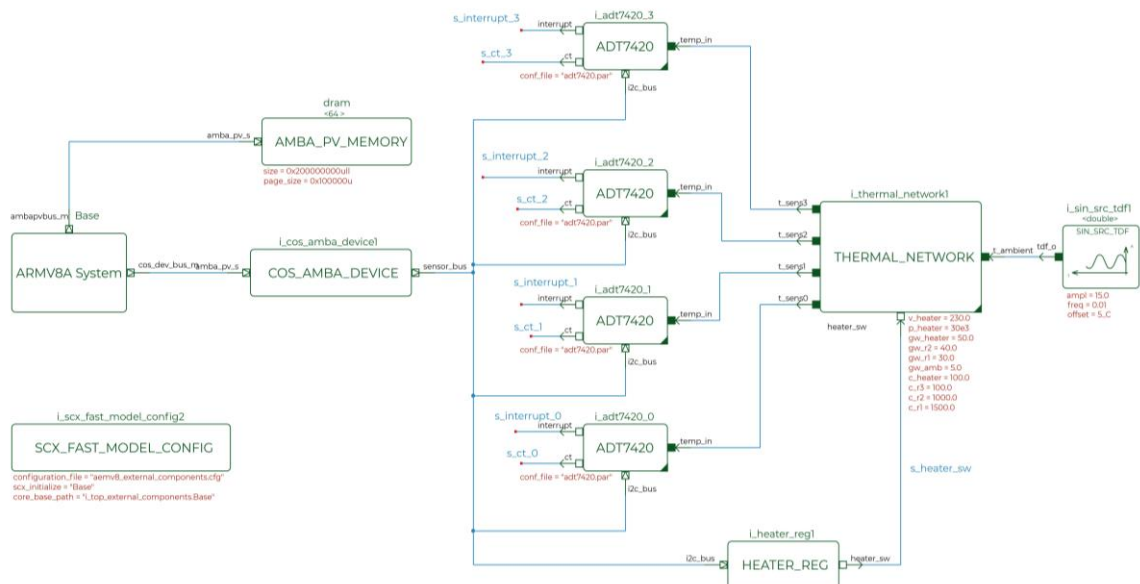


Abbildung 8: ARM FastModel basiertes Temperaturregelsystem [V05]

Dieser Demonstrator wurde vom DFKI zur Performanceanalyse in Teilaufgabe 4.2.2 verwendet.

### **Teilaufgabe 4.1.2 Evaluierung und Bewertung der Ergebnisse in Zusammenarbeit mit Continental**

Die entwickelten Bibliotheken wurden dokumentiert und es wurde Material zur Präsentation bei Kunden erstellt. Dieses wurde ersten Kunden vorgestellt und Feedback eingeholt. Weiterhin wurde die Bibliothek unter anderem auf dem COSEDA UserGroup Meeting 2021 den Kunden von COSEDA vorgestellt. Das direkte Feedback zur ersten prototypischen Verwendung der Bibliothek steht noch aus.

### **Teilaufgabe 4.1.3 Erprobung der Methodik Anhand von Kundenbeispielen**

Die Evaluierung mit Hilfe von Kundenbeispielen, welche Analoge Assertions verwenden wurde begonnen. Hierzu wurden Assertions entwickelt, welche die Stabilität eines DC-DC Konverters überprüfen. Ein weiteres Beispiel war die Anwendung der Assertion Bibliothek auf ein Modell eines hochgenauen Timing Prozessors welcher durch einen Kunden während des Projekts zur Verfügung gestellt wurde. Hierbei sind insbesondere die Mixed Signal Fähigkeiten der Bibliothek von Bedeutung.

### 3 Geplante Verwertung der Ergebnisse

Die Komplexität elektronischer Komponenten steigt auf Grund der Entwicklung der Elektromobilität, automatisiertem Fahren, als auch dem zunehmende Automatisierungsgrad in vielen Bereichen speziell auch im Zusammenhang mit der Nutzung von KI basierten Methoden, derzeit rasant an.

Dabei spielen auch Safety und Security (Sicherheit und Vertraulichkeit) eine immer größere Rolle. Die Anzahl der Anwendungsszenarien und Varianten steigt ebenfalls stark an.

Somit ist derzeit zu beobachten, dass die Anforderungen an die Verifikation von elektronischen Hard-Softwaresystemen stark steigen. Die im Projekt entwickelten Methoden und Verfahren zielen genau auf diesen Aspekt – sie werden die Automatisierung der Tests sowie deren Wieder- und Weiterverwendung als auch eine signifikante Effizienzsteigerung bei der initialen Erstellung der Tests ermöglichen. Somit werden sie einen signifikanten Beitrag leisten, dass aktuelle und künftige Systeme sicher und mit vertretbarem Aufwand entwickelt und verifiziert werden können. Sie tragen dazu bei, dass aktuelle und künftige Sicherheitsstandards erfüllt werden können.

Speziell leisten die Methoden auch einen Beitrag, die Verifikation entlang der Wertschöpfungskette, z.B. von Automobilen, zu ermöglichen. Durch den steigenden Vernetzungsgrad der Komponenten wie er z.B. durch künftige Elektronikarchitekturen wie z.B. Zonenkonzepte entsteht, werden diese Methoden essenziell und ermöglichen eine Vendorübergreifende Verifikation.

Somit werden die ersten Anwender der Methoden im Bereich der Automobilelektronik sein. Neben klassischen Halbleiterentwurfshäusern, haben verstärkt Automobilzulieferer Interesse und Bedarf bekundet.

Geplant ist, nach einem Jahr nach Projektende, eine erste Version der entwickelten Bibliotheken in die aktuelle Release des Entwurfswerkzeugs COSIDE® zu integrieren. Damit erwarten wir im darauffolgenden Jahr bei ersten Leitkunden erste Projekte, welche die Bibliotheken verwenden und damit perspektivisch bei diesen Kunden eine stärkere Nutzung unserer Werkzeuge und somit einen höheren Lizenzbedarf.

Durch die Integration der entwickelten Methoden in COSIDE® wird damit ein Jahr nach Projektende eine zusätzliche Umsatzsteigerung von 5-10% durch erweiterte Anwendungsszenarien bei Bestandskunden prognostiziert. In den folgenden Jahren werden die erweiterten Verifikationsmöglichkeiten die Akquise neuer Kunden ermöglichen und dadurch eine zusätzliche Umsatzsteigerung von anfangs ca. 5% und später 15% je Jahr ermöglichen.

Parallel zur Integration in COSIDE® werden wir innerhalb der Accellara Systems Initiative die Standardisierung der SystemC basierten Assertionsprache anstoßen. Ziel ist es diese als anerkannten Standard zu etablieren und damit als Standardaustauschmethode für Systemebenen Verifikationsszenarien zu etablieren. Wir erwarten damit den Zugang zu neuen Kunden und Kundengruppen zu erlangen, da ein Bedarf an Werkzeugen entstehen wird, welche den neuen Standard unterstützen.

Durch diese Aktivitäten, rechnen wir mit einer Amortisierung der gesamten Projektkosten nach ca. 5 Jahren und danach mit einer weiteren zusätzlichen jährlichen Umsatzsteigerung von ca. 15%.

## 4 Zusammenfassung

Das Projekt AUTOASSERT hat den Bereich der Systemverifikation erfolgreich weiterentwickelt, indem es Assertion-Methoden eingeführt hat, die speziell für die Untersuchung heterogener Systeme entwickelt wurden, welche sowohl analoge als auch digitale Hardware- und Softwarekomponenten umfassen. Diese Methoden haben sich aus den traditionellen Verifikationsstrategien für digitale Schaltungen auf der Implementierungsebene entwickelt und wurden auf die Systemebene ausgeweitet, insbesondere auf den Bereich der zeit- und wertkontinuierlichen Verhaltensweisen. Diese Entwicklung bedeutet einen Sprung von manuellen Testverfahren zu einer automatisierten Umgebung, in deren Mittelpunkt Assertion-basierte Tests zur Systemverifizierung stehen. Dadurch kann ein größeres Spektrum an Systemverhalten gleichzeitig automatisiert untersucht werden, was den Zeit- und Arbeitsaufwand im Vergleich zu herkömmlichen manuellen Verfahren erheblich reduziert.

Der Clou des Projekts liegt in der Technologie der Eigenschaftsprüfung. Sie ermöglicht die benutzerfreundliche Definition von Assertions ohne zusätzlichen Aufwand, was in der Vergangenheit eine große Herausforderung darstellte. Darüber hinaus erkennt sie auch die kontinuierlichen Zeit- und Wertecharakteristika verschiedener Systeme und kann diese effektiv darstellen, was mit den bisherigen Technologien nicht möglich war. Entscheidend ist, dass diese Technologie nun auch die Darstellung der Software/Hardware-Interaktion ermöglicht, was die Qualität der Verifikation erheblich verbessert, ohne die verfügbare Rechenleistung oder Rechenzeit zusätzlich zu belasten.

Das Hauptziel des AUTOASSERT-Projekts bestand darin, den Verifizierungsprozess zu rationalisieren, indem der erforderliche Aufwand reduziert und die Systemkomplexität effizient gehandhabt wird, um so einen kostengünstigen und überschaubaren Entwurf integrierter Systeme zu ermöglichen, die modernen Technologien wie IoT, autonomes Fahren und Industrie 4.0 gerecht werden.

Das Projekt hat seine Ziele erfolgreich erreicht. Die Highlights haben in der Community großes Interesse geweckt, wie eine Nominierung für den besten Beitrag auf der renommierten Design, Automation and Test in Europe Conference (DATE) zeigt. Darüber hinaus wurden die Projektergebnisse auf der Design and Verification Conference (DVCon) Europe vorgestellt, was weitere Anerkennung und Interesse hervorrief.

Die Aufmerksamkeit, die das Projekt erregt hat, hat dazu geführt, dass die Ergebnisse noch während des Projekts in der Verification Working Group der Accellera Systems Initiative mit der Absicht der Standardisierung diskutiert wurden. Es ist daher zu erwarten, dass die Projektergebnisse in den kommenden Jahren im Rahmen eines Accellera-Standards weiterverfolgt werden, öffentlich verfügbar werden und möglicherweise später als IEEE-Standard angenommen werden.

Auch auf Grund der hohen Aufmerksamkeit, welche die Projektergebnisse erzielt haben, gab es schon mehrere Gespräche mit Kunden über erste Pilotprojekte. Es gibt eine prototypische Integration, welche ersten Kunden zum Test und zur Erlangung von Feedback zur Verfügung gestellt wurde. Es ist geplant die vollständige Integration der im Projekt entwickelten Methoden innerhalb der nächsten zwei Jahre abzuschließen. Dies wird im ersten Schritt zu weiteren Nutzern und damit zusätzlich verkauften Lizenzen bei vorhandenen Kunden führen und im zweiten Schritt auch die Akquise neuer Kunden ermöglichen.

Das AUTOASSERT-Projekt befasste sich in erster Linie mit der Bewertung von Systemreaktionen durch einen automatisierten Ansatz, der auf eigenschaftsbasierten Assertions beruht. Eine große Herausforderung auf dem Gebiet der Simulation und Verifikation ist die Verringerung der Simulationszeit. Ein möglicher Ansatz für zukünftige Arbeiten ist die Verwendung des Prinzips des Checkpointing. Diese Ansätze sind jedoch aktuell noch oft ungeeignet für die heterogenen virtuellen Prototypen (VPs), die in der Automobilindustrie verwendet werden. Bei diesen VPs handelt es sich um komplexe Systeme, die nicht nur digitale Komponenten, sondern auch Software und analoge Hardware umfassen und auf die Bewältigung komplexer Szenarien ausgelegt sind. Außerdem sind sie nicht mit den Abläufen der kontinuierlichen Integration (CI) kompatibel, was ihre Anwendung weiter einschränkt.

In Anbetracht dieser Herausforderungen haben DFKI und COSEDA ihre zukünftige Arbeit darauf ausgerichtet, diese Methoden von der rein digitalen Domäne auf ein breiteres Spektrum zu übertragen, das sowohl analoge als auch digitale Hardware/Software-Systeme auf Systemebene umfasst. Ziel ist es, robuste, umfassendere Simulations- und Verifikationsansätze zu entwickeln, die für komplexe Systeme geeignet sind und aussagekräftige Erkenntnisse liefern können, ohne wichtige Details zu übersehen oder Fehler in das endgültige Design einfließen zu lassen.



## 5 Veröffentlichungen des Projekts

Die im AUTOASSERT-Projekt erzielte Ergebnisse wurden auf nationalen und internationalen Konferenzen und Workshops veröffentlicht. Dabei sind insgesamt 12 Publikationen entstanden.

[V01] Hassan, M., Große, D., & Drechsler, R. (2021, January). System-level verification of linear and non-linear behaviors of RF amplifiers using metamorphic relations. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference* (pp. 761-766).

[V02] Hassan, M., Große, D., & Drechsler, R. (2021, February). System level verification of phase-locked loop using metamorphic relations. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1378-1381). IEEE. **(Best Paper Candidate)**

[V03] Coşkun, K. Ç., Hassan, M., & Drechsler, R. (2022, April). Equivalence Checking of System-Level and SPICE-Level Models of Linear Analog Filters. In *2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)* (pp. 160-165). IEEE.

[V04] Coşkun, K. Ç., Hassan, M., & Drechsler, R. (2022). Equivalence Checking of System-Level and SPICE-Level Models of Linear Circuits. *Chips*, 1(1), 54-71.

[V05] Hassan, M., Vörtler, T., Einwich, K., Drechsler, R., & Große, D. A Cross-domain Heterogeneous ABV-Library for Mixed-signal Virtual Prototypes in SystemC/AMS. DVCon Europe 2022.

[V06] Hassan, M., Vörtler, T., Einwich, K., Drechsler, R., & Große, D. (2023, February). Toward System-Level Assertions for Heterogeneous Systems. In *Advanced Boolean Techniques: Selected Papers from the 15th International Workshop on Boolean Problems* (pp. 67-81). Cham: Springer International Publishing.

[V07] Hassan, M., & Drechsler, R. System Level Verification of Analog/Mixed-Signal Systems using Metamorphic Relations. *TUZ 2022*

[V08] Coskun, K. C., Hassan, M., & Drechsler, R. New Directions for Equivalence Checking of System-Level and SPICE-Level Models of Linear Circuits. *TUZ 2023*

[V09] Deb, A., Datta, K., Hassan, M., Shirinzadeh, S., & Drechsler, R. (2023). Automated Equivalence Checking Method for Majority based In-Memory Computing on ReRAM Crossbars.

[V10] Coşkun, K. Ç., Hassan, M., & Drechsler, R. (2023, April). Equivalence Checking of System-Level and SPICE-Level Models of Static Nonlinear Circuits. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1-6). IEEE.

[V11] Coşkun, E. N. D., Hassan, M., Goli, M., & Drechsler, R. (2023, April). VAST: Validation of VP-based Heterogeneous Systems against Availability Security Properties using Static Information Flow Tracking. In *2023 24th International Symposium on Quality Electronic Design (ISQED)* (pp. 1-8). IEEE.

[V12] Coskun, E. N. D., Hassan, M., & Drechsler, R. Security Validation of VP-based Heterogeneous Systems: A Completeness-driven Perspective. *MBMV 2023*.

## 6 Literatur

- [1] Heise Online, Chip-Auftragsfertiger TSMC: Moore's Law lebt!, <https://www.heise.de/newsticker/meldung/TSMC-Moore-s-Law-lebt-4501547.html>, 21. August 2019, abgerufen 19. September 2019
- [2] IEEE Std. 1666. IEEE Standard SystemC LRM, 2011
- [3] IEEE Std 1666.1-2016, IEEE Standard for Standard SystemC Analog/Mixed-Signal Extensions LRM, 2016
- [4] "IEEE Standard for SystemVerilog--Unified Hardware Design, Specification, and Verification Language," in IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012) , vol., no., pp.1-1315, 22 Feb. 2018, doi: 10.1109/IEEESTD.2018.8299595.
- [5] "IEEE Standard for the Functional Verification Language e," in *IEEE Std 1647-2011 (Revision of IEEE Std 1647-2008)* , vol., no., pp.1-495, 26 Aug. 2011, doi: 10.1109/IEEESTD.2011.6006495.
- [6] Snyder, Wilson. "Verilator and systemperl." *North American SystemC Users' Group, Design Automation Conference*. 2004.
- [7] A. Piziali, Functional verification coverage measurement and analysis. Springer Science & Business Media, 2007.
- [8] J. Yuan, C. Pixley, and A. Aziz, Constraint-based Verification. Springer, 2006.
- [9] Arbeitskreis Automotive, Working Group Virtual Platforms, "Automotive Virtual Prototyping Platform – White Paper", edacentrum e.V., November 2019
- [10] Mikhail Moiseev, Leonid Azarenkov, and Ilya Klotchkov. Temporal assertions in SystemC. In SystemC Evolution Day, October 29<sup>th</sup>, 2020