Ein Cloud-basierter Workflow für die effektive Fehlerdiagnose von Loop-Back-Strukturen

Matthias Gulbins, André Schneider, Steffen Rülke
Fraunhofer IIS/EAS Dresden
{matthias.gulbins|andre.schneider|steffen.ruelke}@eas.iis.fraunhofer.de

Kurzfassung

Eine hochkomplexe und zeitaufwändige Aufgabe beim Entwurf integrierter Mixed-Signal-Schaltkreise ist die Fehlerdiagnose. Der vorliegende Beitrag stellt einen auf Cloud-Technologien basierenden Lösungsansatz vor, der Fehler in für solche Schaltkreise typischen Strukturen aus Analog-Digital- und Digital-Analog-Wandlern lokalisiert. Das Diagnoseverfahren (Ergebnis des BMBF-Projektes *DIANA*) beruht auf dem sogenannten Loop-Back-Test, der zwar die Generierung von Testdaten vereinfacht, aber eine Vielzahl von Variantensimulationen mit verschiedenen Simulationsprinzipien und erheblichen Datenmengen erfordert. Diese sollen nunmehr problemangepasst und damit effizient in der Cloud realisiert werden. Für die entsprechende Informationsverarbeitung in der Cloud wurde das in dem Projekte *OptiNum-Grid* entwickelte Framework *GridWorker* adaptiert. Experimente mit ersten Anwendungsbeispielen bestätigen die Leistungsfähigkeit und Praktikabilität des Ansatzes für daten- und verarbeitungsintensive Schaltkreisentwurfsaufgaben.

1 Einleitung

Analog-Digital- und Digital-Analog-Wandler (ADC/DAC) sind typische Schaltungskomponenten vieler Mixed-Signal-Schaltkreise, z.B. im Bereich der Automobilelektronik. Der Test solcher Strukturen und die Diagnose der Fehlerursachen gestalten sich als überaus komplex und aufwendig. Eine Strategie zur Reduktion der Test- und Diagnosekosten ist ein kombiniertes Vorgehen aus strukturellen und simulations-methodischen Ansätzen. Neben einer Loop-Back-Struktur für den Test kommen hierarchische Ansätze und Cloud-Technologien für die Fehlerdiagnose zur Anwendung.

Bei einer Loop-Back-Struktur werden für den Test DAC und ADC in Reihe geschaltet, so dass das Ausgangssignal des ADCs mit dem Eingangssignal des DACs verglichen werden kann. Vorteile sind, dass

- der Test digital abläuft, was eine einfache Stimuligenerierung und eine reproduzierbare Fehlerdiagnose ermöglicht,
- die Fehler der Wandler direkt auf der elektrischen Netzwerkebene definiert werden und
- Schwankungen der Herstellungsparameter von Schaltkreisen unmittelbar beim Test berücksichtigt werden.

Für die Beherrschbarkeit der hochkomplexen und (entsprechend der Parametervariationen und der Vielfalt von möglichen Fehlerorten und -werten) sehr umfangreichen Fehlersimulationen ist folgendes vorgesehen:

- Einerseits ist ein hierarchisches Vorgehen erforderlich: Alle Fehlersimulationen der Wandler werden auf Netzwerkebene ausgeführt, die Ergebnisse in die funktionelle Ebene überführt und die Simulationen in der Loop-Back-Umgebung sowie die Fehlerdiagnose auf funktioneller Ebene durchgeführt.
- Andererseits werden alle Operationen (Simulationen und Konvertierungen auf Netzwerk- und Funktionsebene) in einer Cloud unter Verwendung geeigneter Virtualisierungstechnologien (*GridWorker*) ausgeführt.

Mit dem vorgestellten Lösungsansatz wird die Leistungsfähigkeit der Cloud-Technologien für die zunehmend daten- und verarbeitungsintensiven Schaltkreisentwurfsaufgaben demonstriert. Gerade aus der immer höheren Integrationsdichte und der zunehmenden Verarbeitungsbreite der Schaltungskomponenten erwachsen diesbezüglich erhebliche Anforderungen.

Weiterhin ist der beschriebene Lösungsansatz auch die Grundlage für künftige Dienstleistungsangebote zur Varianten- bzw. Fehlersimulation, Analyse und Diagnose beim Mixed-Signal-Schaltkreisentwurf. Dabei könnten auch die von den Autoren entwickelten Werkzeuge zur Informationsverarbeitung in der Cloud (*GridWorker*) und zur analogen Fehlersimulation (*aFSIM*) als *Software-as-a-Service* (SaaS) Verwendung finden.

Schließlich wird mit dem Lösungsansatz auch die Qualität der Fehlerdiagnose signifikant erhöht. Das wird möglich, weil dank der Cloud-Technologien anstelle von relativ groben Verhaltensmodellen nunmehr auch genauere Strukturmodelle auf der elektrischen Netzwerkebene verwendet werden können, die das auszuwertende Datenvolumen und Anzahl der erforderlichen Simulationsläufe drastisch ansteigen lassen.

Der weitere Beitrag ist wie folgt gegliedert: Abschnitt 2 erläutert das Fehlerdiagnoseverfahren von Loop-Back-Strukturen, Abschnitt 3 stellt den Cloud-basierten Workflow dafür vor und Abschnitt 4 die Adap-

tion des *Gridworker*s an diesen Workflow. Abschnitt 5 diskutiert die bei Experimenten erreichten Ergebnisse. Eine Zusammenfassung sowie der Ausblick auf weiterführende Arbeiten schließen den Beitrag ab.

2 Prinzip der Fehlerdiagnose für Loop-Back-Strukturen

Wichtige Schritte bei der Fertigung integrierter nanoelektronischer Systeme (System-on-Chip – SoC) sind Fertigungs- und On-Chip-Test sowie – im Fehlerfall – die Fehlerdiagnose. Insbesondere bei SoC mit Mixed-Signal-Komponenten ist dies meist sehr zeit- und kostenaufwendig. Ein Ansatz zur Aufwandsreduktion ist der Loop-Back-Test.

Grundliegende Idee des Loop-Back-Tests ist es, zwei funktional zueinander inverse Blöcke in Reihe zu schalten und den eigentlichen Test dann für diese Zusammenschaltung durchzuführen. Der Testaufwand kann erheblich reduziert werden, weil Ein- und Ausgangssignale der Zusammenschaltung weitgehend einander entsprechen und damit einfach verglichen werden können

Im Projekt DIANA wurde ein Zugang entwickelt, um den Loop-Back-Test auch für Mixed-Signal-Komponenten vom Typ ADC und DAC anzuwenden, wenn diese gemeinsam in einem System-on-Chip-Halbleiterbaustein integriert sind. Die Zusammenschaltung von DAC und ADC zu einer Loop-Back-Struktur mit digitalen Ein- und Ausgängen ermöglicht den Test auf digitaler Basis. Dabei erfolgt ein statisches Vorgehen, bei dem der ADC mit einer Rampenfunktion und der DAC mit einer Treppenfunktion stimuliert werden. Nach der Stimulierung des DACs mit einer digitalen Treppenfunktion wird das analoge Ausgangssignal des DACs auf den analogen Eingang des ADCs geschaltet. Ein zwischengeschalteter Tiefpass erzeugt analoge Zwischenwerte, um die Rampenfunktion besser nachzubilden und die erforderliche Testgenauigkeit zu erreichen (vergl. Abbildung 1). Die Ausgabe des ADCs ist eine Folge aus Digitalwerten.

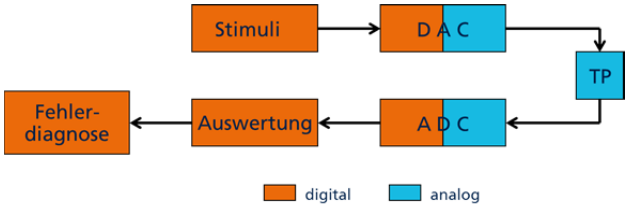

Abbildung 1: Loop-Back-Struktur aus Digital-Analog- und Analog-Digital-Wandler

Aufgrund der großen Datenmenge, werden aus der Ausgabe des ADCs, also der Loop-Back-Struktur, Kenngrößen berechnet, die es auch für jeden der beiden Wandler gibt. Statische Kenngrößen sind z. B. integrale Nichtlinearität (INL) und differentielle Nichtlinearität (DNL) [1]. Zur Verbesserung der Fehlerdiagnose können weitere Kenngrößen wie Monotonie oder fehlende Codes herangezogen werden. Alle Kenngrößen einer Folge aus Digitalwerten werden zu einer Signatur zusammengefasst, die diese Folge charakterisiert.

Im Unterschied zu den aus der Literatur bekannten Verfahren liegt der Schwerpunkt bei unserem Vorgehen auf der Fehlerdiagnose [2], [3]. Aus Stimulus und Ausgabe der Loop-Back-Schaltung soll nicht nur erkannt werden, ob ADC und DAC innerhalb einer vorgegebenen Genauigkeit korrekt funktionieren, sondern im Fehlerfall auch diagnostiziert werden, welcher der beiden Wandler fehlerbehaftet und wo der Fehler lokalisiert ist.

Dafür wird die fehlerfreie Loop-Back-Schaltung simuliert und aus den Simulationsergebnissen die Kenngrößen für die Signatur berechnet. Anschließend wird ein Fehler ausgewählt und in die Schaltung eingefügt, d. h., die Schaltung entsprechend diesem Fehler verändert. Die mit dieser Schaltung ausgeführte Simulation liefert eine Fehlersignatur. Die Signatur der fehlerfreien Schaltung und die Fehlersignaturen aller Fehler werden vor der eigentlichen Fehlerdiagnose berechnet und dann bei der Fehlerdiagnose mit der Signatur der konkreten zu testenden Schaltung verglichen.

Bei der Fehlerdiagnose wird die zu untersuchende Schaltung mit der Treppenfunktion stimuliert und aus der Ausgabefolge die Kenngrößen und damit die „gemessene“ Fehlersignatur berechnet. Diese „gemessene“ Fehlersignatur wird mit den berechneten Fehlersignaturen verglichen und aus dem Vergleich auf den vorliegenden Fehler geschlossen.

Um einen möglichst großen Hardware-Bezug zu erreichen, wurden die Wandler auf der elektrischen Netzwerkebene modelliert. Als Fehler wurden Kurzschlüsse und Unterbrechungen sowie parametrische Veränderungen der Widerstände und Kapazitäten betrachtet.

Da Simulationen auf elektrischer Netzwerkebene sehr zeitaufwändig sind, wurde ein hierarchisches Vorgehen gewählt. Dabei werden ADC und DAC unabhängig voneinander simuliert und die Simulationsergebnisse jeweils in Kennlinien umgewandelt. Die Simulation der Loop-Back-Zusammenschaltung erfolgt unter Verwendung der Kennlinien auf der Verhaltensebene z. B. mit MATLAB.

Bei der Durchführung der Fehlerdiagnose hat sich herausgestellt, dass sich viele Fehlersignaturen nur geringfügig unterscheiden, so dass viele Fehler als Fehlerkandidaten für eine gemessene Fehlersignatur in Frage kommen. Ein Abstandsmaß ist schwer zu definieren. Dabei stellt sich die Frage, wie genau die Kennlinien sind, denn diese werden von Parametern, z. B. Kennwerten von Bauelementen wie Widerständen und Transistoren oder von Zellen wie Operationsverstärker, beeinflusst. Infolge von kleinen Veränderungen im Produktionsprozess können diese Parameter variieren. Von ihnen seien jeweils Nominalwert und Varianz bekannt. Die Parametervariationen werden in Monte-Carlo-Simulationen nachgebildet und so Kennlinien erzeugt, die diese Parametervariationen berücksichtigen. Damit sind eine Quantifizierung der Fehlerabstände und eine Verbesserung der Fehlerdiagnose möglich.

Durch die Monte-Carlo-Simulationen erhöht sich der Simulationsaufwand drastisch. Um die damit gleichfalls wachsende Datenmenge zu reduzieren und für die Fehlerdiagnose anwendbar zu machen, wird neben den Nominalwerten nur die Datenverteilung in Form von Histogrammen sowie Min-Max-Intervallen verwendet.

Die Kenngrößen und die Min-Max-Intervalle einer Fehlersignatur bilden die Grundlage für die Fehlerdiagnose. Werden *n* Kenngrößen als Achsen eines kartesischen Koordinatensystems betrachtet, so liegen für jeden Fehler alle möglichen Kombinationen der *n* Kenngrößen innerhalb eines *n*-dimensionalen Quaders, der durch die Min-Max-Intervalle begrenzt wird. Zur Veranschaulichung werden die beiden Kenngrößen „Gain Error“ und „Offset Error“ betrachtet, wie in Abbildung 2 illustriert. Für den fehlerfreien Fall F0 und sechs mögliche Fehler F1, …, F6 wurden per Monte-Carlo-Simulation die Intervalle für Offset und Gain Error bestimmt, in denen diese Kenngrößen auftreten. Diese spannen unterschiedlich große Flächen auf. Aufgabe der Fehlerdiagnose ist es, einer gemessenen Fehlersignatur Ti Fehlerkandidaten zuzuordnen. Abbildung 2 zeigt, dass sich diese Aufgabe grafisch lösen lässt: Liegt die Fehlersignatur Ti innerhalb der aufgespannten Fläche bzw. im aufgespannten *n*-dimensionalen Quader eines Fehlers, so kommt dieser Fehler als Kandidat für die vorliegende Fehlersignatur in Frage. Gibt es nur einen Kandidaten, so ist die Fehlerdiagnose damit beendet. Im vorliegenden Beispiel gemäß Abbildung 2 wird bei den Fehlersignaturen T1 und T2 kein Fehler (F0); bei T3 Fehler F3; bei T4 F5 und bei T5 F4 klassifiziert. Gibt es mehrere Kandidaten, so wird zunächst der Fehlerort (Fehlerlokalisation) bestimmt.

Sind die Fehler F1 und F5 Fehler am gleichen Fehlerort mit unterschiedlichen Fehlerwerten, so ist der Fehlerort gefunden. Im Schritt der Fehleridentifikation muss der Fehlerwert, d. h. die Abweichung von einem Nominalwert, bestimmt werden.

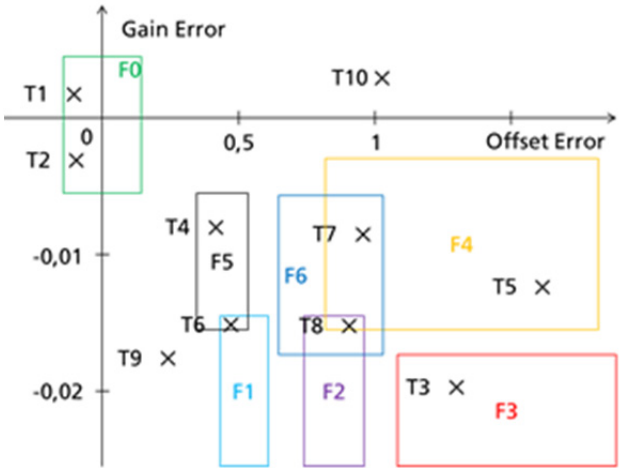

Abbildung 2: Fehlerintervalle Fi der Kenngrößen „Gain Error“ und „Offset Error“ für verschiedene Fehler

Haben die Fehlerkandidaten unterschiedliche Fehlerorte, so können weitere Kenngrößen in Simulation und Messung einbezogen werden, um so die Fehlerkandidaten voneinander zu unterscheiden.

Unabhängig davon ist eine Bewertung der Kandidaten unter Verwendung der berechneten Verteilungen möglich. Dabei werden Fehler gleichen Fehlerorts zusammengefasst.

In Abbildung 2 gibt es für die Fehlersignaturen T6 und T7 jeweils zwei, F1 und F5 bzw. F4 und F6, sowie für die Fehlersignatur T8 drei Kandidaten: F2, F4 und F6.

Falls einer gemessenen Fehlersignatur kein Fehler zugeordnet werden kann, gehört diese Signatur nicht zu den in den Simulationen untersuchten Fehlern. In Abbildung 2 gilt das für T9 und T10.

3 Workflow zur Erzeugung der Fehlersignaturen

Wie in Abschnitt 2 erläutert, erfordert die Detektion von Fehlern in den ADC/DAC-Komponenten eines konkreten gefertigten Mixed-Signal-Schaltkreises sogenannte Fehlersignaturen, die im Rahmen eines Loop-Back-Tests zu berechnen sind.

Der in diesem Abschnitt vorgestellte Workflow realisiert die algorithmische Umsetzung des Loop-Back-Tests durch Simulatoren und Skripte und liefert im Ergebnis die für die Fehlerdiagnose benötigten Fehlersignaturen. Für berechnungs- und datenintensiven Schritte werden dabei Cloud-Technologien eingesetzt.

Für die Arbeiten wurde zunächst der im Fraunhofer IIS/EAS entwickelte analoge Fehlersimulator *aFSIM* [4] in Verbindung mit dem Netzwerksimulator ngSPICE sowie Octave als Rahmenprogramm und Simulationsprogramm auf Verhaltensebene verwendet.

Der analoge Fehlersimulator *aFSIM* ist ein Programm zur analogen Fehlersimulation auf elektrischer Netzwerkebene, mit dem auf der Grundlage von symbolischen Fehlerbeschreibungen in Form von Pattern eine Fehlerliste erzeugt, die Fehler der Fehlerliste in die Schaltung injiziert, die Fehlersimulationen angestoßen und die Ergebnisse der Fehlersimulation ausgewertet werden können. Für unsere Zwecke werden nur die Funktionen Fehlerlistengenerierung und Fehlerinjektion verwendet. Die Simulationsaufrufe erfolgen vom *GridWorker* und die Auswertung der Simulationsergebnisse auf Verhaltensebene mit Hilfe von Octave, einer Open-Source-Alternative zu MATLAB.

Abbildung 3 zeigt den Workflow zur Erzeugung von erweiterten Fehlersignaturen für die Fehlerdiagnose. Um die große Anzahl der Dateien zu verdeutlichen, wird deren Anzahl in Abbildung 3 durch die Variablen x, y und m ausgedrückt. Dabei ist x die Anzahl der Fehler im ADC, y die der Fehler im DAC und m die Anzahl der Monte-Carlo-Simulationen. Für die 5-Bit-Varianten gilt: (x, y, m) = (9.137, 395, 1.000).

Die Kennlinien für den ADC und für den DAC werden unabhängig voneinander berechnet. Ausgangspunkt ist jeweils eine elektrische Netzliste und eine symbolische Beschreibung der Fehler. Im **Schritt 1** wird daraus

jeweils eine Fehlerliste erzeugt. Mit Hilfe eines Skripts werden die Fehler separiert, und es entsteht für jeden Fehler eine Einzelfehlerliste. Im folgenden **Schritt 2** wird für jeden Fehler aus elektrischer Netzliste und Einzelfehlerliste eine modifizierte Netzliste generiert. Die in dieser modifizierten Netzliste enthaltenen Widerstände und Kapazitäten werden per Skript einer Parametrisierung und damit der Monte-Carlo-Simulation zugänglich gemacht. Mit Hilfe des *GridWorkers* wird in **Schritt 3** jede der modifizierten Netzlisten mit ngSPICE einer Monte-Carlo-Simulation unterworfen. Die Dateien mit den SPICE-Simulationsergebnissen enthalten sehr viele Daten, während die Kennlinien selbst nur aus der Zuordnung analoger Eingangswert – digitaler Ausgangswert beim ADC und digitaler Eingangswert – analoger Ausgangswert beim DAC bestehen. In **Schritt 4** extrahiert deshalb ein Octave-Programm aus den SPICE-Simulationsergebnissen die Kennlinien.

Im Ergebnis erhält man für den fehlerfreien und jeden fehlerhaften Wandler ein Kennlinienbündel, bestehend aus der Nominalkennlinie und den Kennlinien der Monte-Carlo-Simulation. Die ersten vier Schritte beziehen sich überwiegend auf die elektrische Netzwerkebene und sind deshalb sehr zeitintensiv.

Sie liefern die Daten für **Schritt 5**, die Loop-Back-Simulation. Dabei kommt es darauf an, einen DAC und einen ADC, jeweils repräsentiert durch ein Kennlinienbündel, in der Loop-Back-Struktur zu simulieren. Aus Effektivitätsgründen wird die Simulation auf der Verhaltensebene durchgeführt. Die Kennlinien wurden schon entsprechend konvertiert.

Zunächst werden die Kenngrößen für den fehlerfreien Fall berechnet. Deren Nominalwerte ergeben sich bei Simulation der Nominalkennlinien der fehlerfreien Wandler. Durch Kombination von je einer Monte-Carlo-Kennlinie des einen mit einer Monte-Carlo-Kennlinie des anderen Wandlers erhält man eine entsprechende große Anzahl von variierten Werten für jede Kenngröße. Die große Anzahl von Werten wird in **Schritt 6** dadurch reduziert, dass nur die Nominalwerte der Kenngrößen aufgehoben und alle anderen in 11 Bins eingeordnet werden. So entsteht ein Histogramm für jede Kenngröße.

Die Fehlerdiagnose geht von einem Einzelfehlermodell aus. Das bedeutet, dass nur einer der beiden Wandler fehlerbehaftet und nur einer der Fehler wirksam ist. Bei der Fehlersimulation in der Loop-Back-Struktur muss deshalb jeder fehlerhafte Wandler mit dem fehlerfreien anderen Wandler kombiniert werden. Zur Berechnung der Nominalwerte der Kenngrößen eines Fehlers werden die Nominalkennlinien ausgewählt und in der Loop-Back-Struktur simuliert. Danach wird jede Monte-Carlo-Kennlinie des Fehlers mit einer zufällig ausgewählten Monte-Carlo-Kennlinie des fehlerfreien anderen Wandlers kombiniert. In Schritt 6 werden wie im fehlerfreien Fall die berechneten Kenngrößen der Monte-Carlo-Kennlinien in jeweils 11 Bins zu Histogrammen zusammengefasst. Auf diese Weise erhält man eine erweiterte Signatur für die fehlerfreien Wandler und erweiterte Fehlersignaturen für jeden Fehler des ADCs und jeden Fehler des DACs.

Abschließend erfolgt die in Abschnitt 2 beschriebene eigentliche Fehlerdiagnose, d. h., der Vergleich einer gemessenen Fehlersignatur mit den vorausberechneten Fehlersignaturen. Im beschriebenen Ansatz wird dieser Schritt lokal mit der Rechentechnik des Nutzers realisiert, ist also nicht an die Verfügbarkeit einer Cloud-Infrastruktur gebunden.

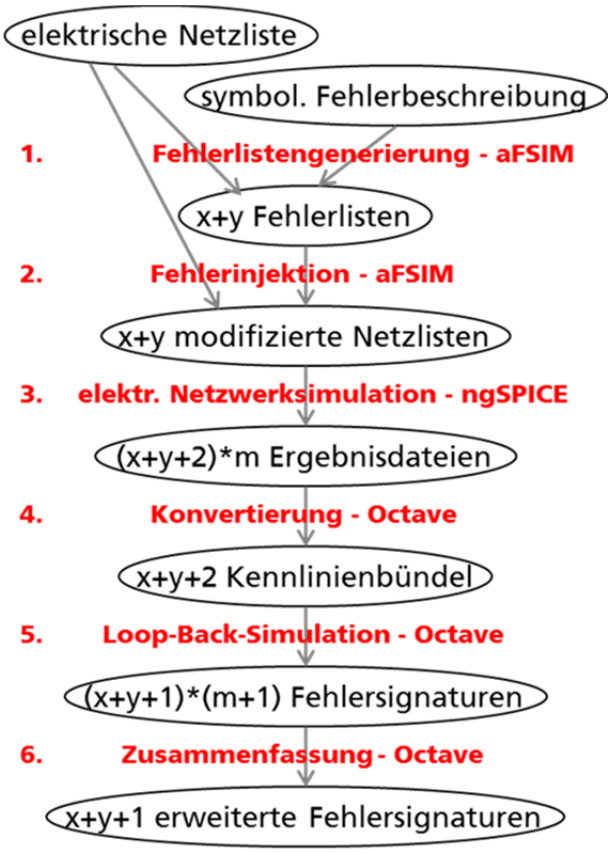

Abbildung 3: Workflow zur Erzeugung von erweiterten Fehlersignaturen. x Anzahl der ADC-, y Anzahl der DAC-Fehler, m Anzahl der Monte-Carlo-Simulationen

4 Adaption *GridWorker*

GridWorker ist ein am Fraunhofer IIS/EAS entwickeltes Framework [5], das auf Basis des Map/Reduce-Paradigmas [6] parallel ausführbare Bearbeitungseinheiten (Map-Phase) über angepasste Adapter auf Multi-Core-, Cluster-, Grid- und Cloud-Infrastrukturen verteilen und nach Beendigung die Ergebnisse wieder zusammenführen kann (Reduce-Phase). Ziel ist es dabei, die wachsenden Compute-Ressourcen insbesondere im Grid- und Cloud-Umfeld für Endanwender im Systementwurf zu erschließen. Diese haben in der Regel keine Möglichkeit, die vielfältigen Low-Level-Middleware-Schnittstellen wie gLite, UNICORE, Globus Toolkit, Grid Engine, PBS (Portable Batch System), OpenStack, OCCI (Open Cloud Computing Interface) etc. direkt aus ihren Applikationen

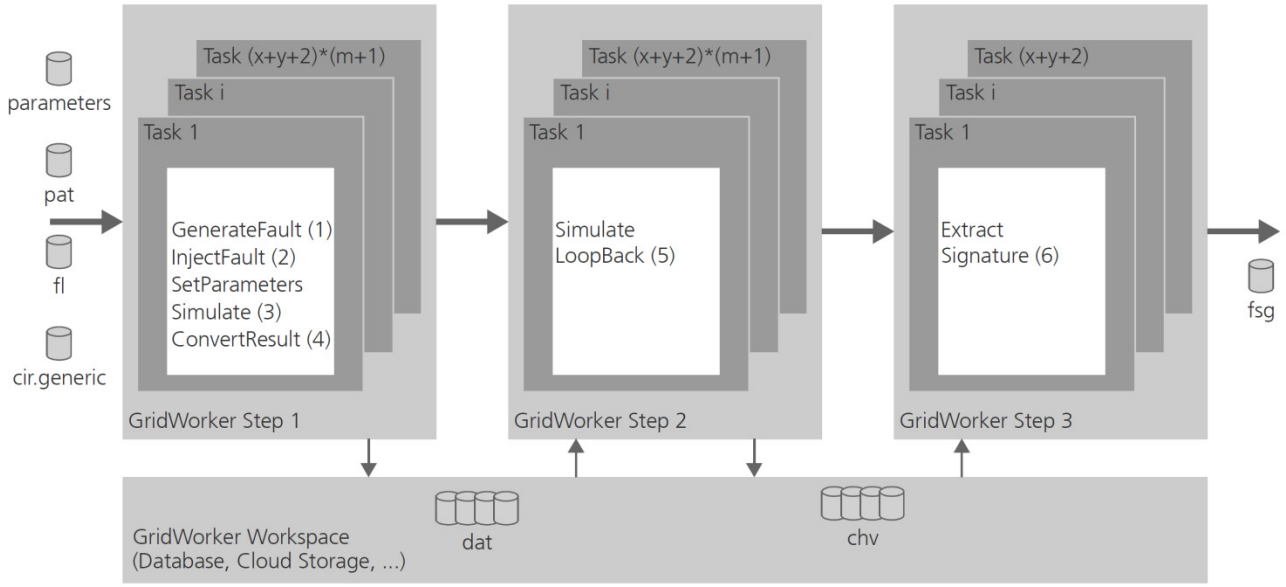

Abbildung 4: Gesamtablauf des Workflows aus *Gridworker*-Perspektive

heraus anzusprechen. *GridWorker* schließt die Lücke zwischen diesen Middleware-Diensten und den Entwurfsprogrammen und ermöglicht dem Systementwerfer das Konfigurieren von Simulations-, Optimierungs- oder Analyseexperimenten und deren Ausführung auf jeweils verfügbaren Backend-Ressourcen, wie sie beispielsweise beim Cloud-Computing von Providern zur Verfügung gestellt werden.

Bei der Fehlerdiagnose besteht die Aufgabe, für alle Fehler jeweils eine Monte-Carlo-Simulation durchzuführen. Da die Einzelsimulationen untereinander entkoppelt und damit unabhängig voneinander ausgeführt werden können, kann der Gesamtzeitbedarf drastisch reduziert werden, wenn Cluster-, Grid- oder Cloud-Ressourcen mit Hilfe von *GridWorker* eingesetzt und ein maximal möglicher Parallelisierungsgrad ausgenutzt werden.

Die Realisierung der Fehlerdiagnose mit *GridWorker* erfolgte in fünf Schritten:

1. Der gesamte Fehlerdiagnose-Workflow wurde in drei Phasen (Step 1 - 3) zerlegt, da es im gesamten Ablauf zwei Synchronisationspunkte gibt: Die Loop-Back-Simulationen (5) können erst ausgeführt werden, wenn die Ergebnisse aller Schaltungssimulationen vorliegen. Und die Fehlersignaturen können erst nach Vorliegen der einzelnen Signaturen zusammengefasst werden (6).

2. Alle pro Einzelsimulation erforderlichen Programmaufrufe wurden in einem Shellskript *map.sh* zusammengefasst. *GridWorker* ruft dieses Skript während der Map-Phase für einen konkreten Fehlerfall mit den jeweils durch den Monte-Carlo-Algorithmus bestimmten Parameterwerten auf, führt die Simulation durch und generiert als Ergebnis eine Kennliniendatei (Step 1).

3. Auch die für die Loop-Back-Simulation notwendigen Programmaufrufe (MATLAB oder Octave) wurden in einem Shellskript *map.sh* beschrieben und *GridWorker* für den Step 2 zur Verfügung gestellt. Analog wurde für Step 3 vorgegangen.

4. Da zwischen den einzelnen Phasen sehr viele Dateien (Tausend bis Millionen) übergeben werden müssen, wurde *GridWorker* um das Konzept der zentralisierten Speicherung erweitert. Dabei kann der Anwender bei Bedarf zwei Referenzen WORKSPACE und STORAGE konfigurieren und jeder *GridWorker*-Task die Möglichkeit geben, über diese Referenzen Daten zentralisiert abzulegen. Das Konzept erlaubt es, über die Standard-URL-Syntax Pfade in Dateisystemen, Datenbanken oder Referenzen innerhalb von Cloud ObjectStorages zu spezifizieren.

5. Schließlich wurden konkrete Anwendungsbeispiele für die Verwendung mit *GridWorker* aufbereitet. Dazu wird jeweils das Simulationsmodell für den fehlerfreien Fall und die zu untersuchenden Fehlermodelle in einem Verzeichnis *inputs/* bereitgestellt. Außerdem müssen die bei der Monte-Carlo-Simulation zu variierenden Parameter in einer Datei *parameters* spezifiziert werden. Die *GridWorker*-bezogenen Konfigurationen (Cloud-Adapter etc.) werden in einer Datei *configuration* zusammengefasst.

5 Anwendungsbeispiele und Experimente

Der Nachweis der Wirksamkeit des Workflows wurde zunächst anhand der 3-Bit-Varianten von Flash-ADC und R2R-DAC geführt (Blockschaltbilder s. Abbildung 5).

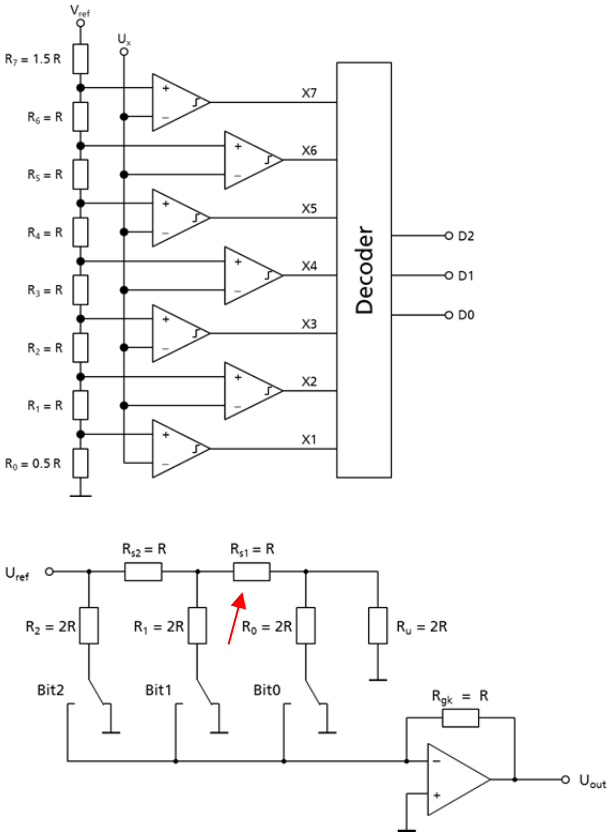

Abbildung 5: 3-Bit-Flash-ADC (oben) und 3-Bit-R2R-DAC (unten)

Ausgangspunkt für die Fehlerdiagnose ist die elektrische Netzliste, die den Aufbau jedes Wandlers aus Widerständen, Kondensatoren und Transistoren beschreibt. Die Netzliste des 3-Bit-Flash-ADCs enthält 85 Widerstände, sieben Kondensatoren und 161 Transistoren, die des 3-Bit-R2R-DACs 17 Widerstände, einen Kondensator und 21 Transistoren. Die symbolische Fehlerbeschreibung gibt an, wie sich diese Bauelemente im Fehlerfall verändern können. Bei den Widerständen wird deren Wert von Kurzschluss (sehr kleiner Wert) bis Unterbrechung (sehr großer Wert) variiert, wobei hier mit insgesamt acht verschiedenen Widerstandswerten gearbeitet wird. Das betrifft insbesondere die funktionsbestimmenden Widerstände, beim Flash-ADC die der Widerstandskette und beim R2R-DAC die des R2R-Netzwerkes. Bei den Kondensatoren und bei den Transistoren werden nur Kurzschluss- und Unterbrechungsfehler betrachtet.

Bei der Fehlerlistengenerierung werden die Fehlerdefinitionen der symbolischen Fehlerbeschreibung auf die elektrische Netzliste angewendet und auf diese Weise eine Fehlerliste erzeugt, auf die sich dann die Fehlerdiagnose bezieht. Da der Flash-ADC für jede Stelle im Thermometer-Code einen Widerstand und einen Komparator benötigt, nimmt die Anzahl der Fehler exponentiell mit der Stellenzahl zu. So hat der 3-Bit-Flash-ADC 1.197 Fehler, die 5-Bit-Variante hingegen 9.137 Fehler. Demgegenüber nimmt die Anzahl der Fehler beim R2R-DAC nur linear zu: 223 Fehlern bei der 3-Bit-Variante stehen 395 Fehler bei der 5-Bit-Variante gegenüber.

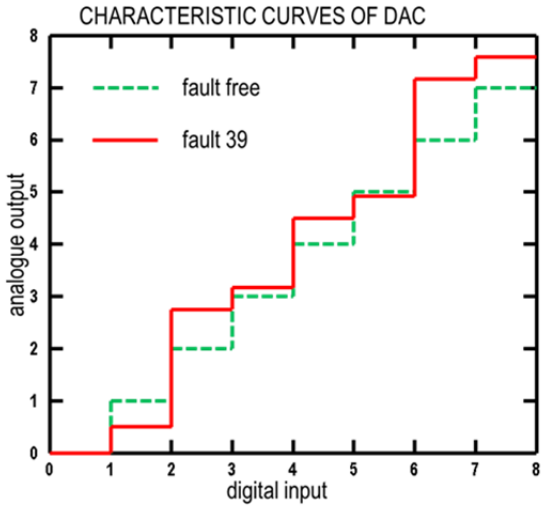

Abbildung 6: Kennlinie des DACs im fehlerfreien Fall (grün) und bei Fehler

Da alle Fehler der Fehlerliste in gleicher Weise behandelt werden, bietet sich eine Parallelisierung der Bearbeitung mit Cloud Computing geradezu an. Nach der Bildung von Einzelfehlerlisten werden modifizierte Netzlisten gebildet, in die einheitlich Parameter für die Monte-Carlo-Simulation eingefügt werden. Die Simulation mit ngSPICE und die Konvertierung in Kennlinienbündel werden in der Cloud durchgeführt.

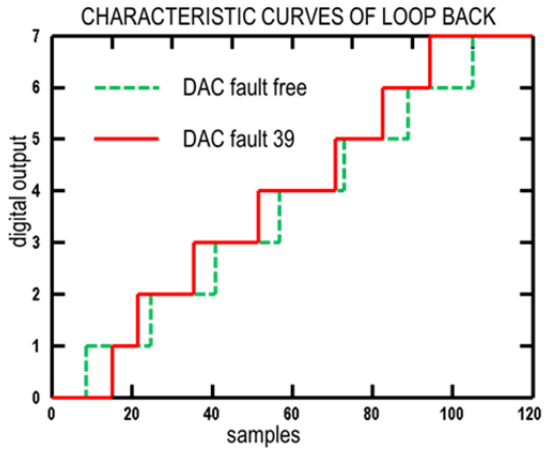

Abbildung 7: Ergebnis der Loop-Back-Simulation für DAC-Fehler 39

Für die Loop-Back-Simulation müssen alle Kennlinienbündel berechnet sein. Da die Loop-Back-Simulation besonders rechenintensiv ist, wird sie in der Cloud durchgeführt. Dabei können die Schritte 5 und 6 aus Abbildung 3 für jeden Fehler unmittelbar hintereinander

ausgeführt werden. Die Abbildungen 6 und 7 sowie die Tabellen 1-3 veranschaulichen die entstehenden Kennlinien, Kenngrößen und Fehlersignaturen.

In Abbildung 6 werden die mit ngSPICE simulierten Kennlinien für den fehlerfreien Fall und DAC-Fehler 39 gegenübergestellt. Bei Fehler 39 erhöht sich der in Abbildung 5 durch den Pfeil markierte Widerstand Rs1 auf den fünffachen Wert. Abbildung 7 zeigt das Ergebnis nach der Loop-Back-Simulation.

Daraus werden Kenngrößen, z.B. die in Tabelle 1 gezeigten berechnet. Diese Kenngrößen werden bei jeder Monte-Carlo-Simulation ermittelt. Zur Datenreduktion werden nur die Kenngrößen der Nominalparameter in der Fehlersignatur gespeichert. Alle anderen berechneten Kenngrößen werden zu Histogrammen zusammengefasst. Die Fehlersignatur für DAC-Fehler 39 wurde aus Platzgründen auf die Tabellen 2 und 3 aufgeteilt.

fnr	Δgain	offset	tue	dnl+	dnl-	inl+	inl-
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.05	0.03	0.69	0.67	-0.58	0.67	-0.61

Tabelle 1: Kenngrößen nach der Loop-Back-Simulation für fehlerfreien Fall und DAC-Fehler 39

fnr	char value	nom	min	max
39	Δgain	0.05	0.00	0.09
	offset	0.03	-0.22	0.23
	tue	0.69	0.33	1.59
	dnlmax	0.67	0.17	0.87
	dnlmin	-0.58	-0.59	-0.46
	inlmax	0.67	0.47	1.18
	inlmin	-0.61	-1.16	0.30

Tabelle 2: Fehlersignatur für DAC-Fehler 39 Teil 1, bestehend aus Nominal-, Maximal- und Minimalwert

fnr	char value	bin width	bins										
39	Δgain	0.01	1	2	5	11	18	25	19	12	6	2	0
	offset	0.05	1	3	7	14	20	22	17	10	5	1	0
	tue	0.18	0	0	0	1	8	61	15	5	4	4	2
	dnlmax	0.10	5	13	17	18	16	26	4	1	0	0	0
	dnlmin	0.02	0	0	0	0	0	74	0	0	18	0	8
	inlmax	0.10	0	0	0	6	20	27	31	12	2	1	1
	inlmin	0.19	0	0	1	3	10	42	9	11	13	9	2

Tabelle 3: Fehlersignatur für DAC-Fehler 39 Teil 2, bestehend aus den Histogrammen

Zur Validierung der Zeiteffektivität wurden die Rechnungen zunächst auf einem Cluster mit einem, fünf, zehn und 20 parallelen Jobs durchgeführt und dabei Zeitmessungen gemacht. Insgesamt wurden neben dem fehlerfreien Fall 1.197 Fehler im ADC und 223 Fehler im DAC berücksichtigt. Für jeden Fehler wurden zehn Monte-Carlo-Simulationen durchgeführt. Zusammen mit den Simulationen der Nominalwerte ergaben sich damit 15.642 Einzelsimulationen. In Abbildung 8 sind die Bearbeitungszeiten für verschiedene Zahlen von parallelen Jobs dargestellt. Für Step 1 und 2 ist die Trennung von Map- und Reduce-Phase markiert. Die Map-Phase (jeweils unterer Balkenabschnitt) ist nahezu ideal parallelisierbar. Bei der Reduce-Phase (jeweils obererer Balkenabschnitt) ist keine Parallelisierung möglich.

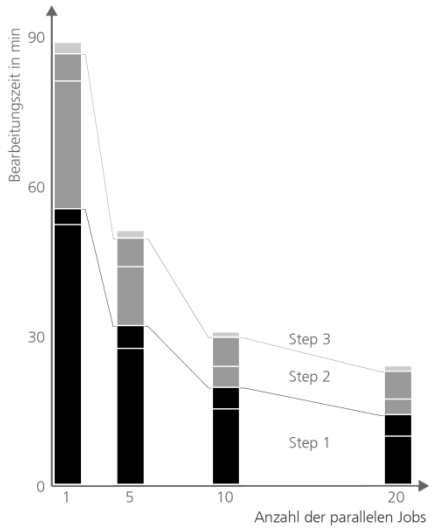

Abbildung 8: Dauer der Berechnungen für die verschiedenen Schritte: Step 1 (schwarz), Step 2 (grau) und Step 3 (hellgrau).

6 Zusammenfassung und Ausblick

Der im Beitrag vorgestellte Ansatz nutzt Cloud-Technologien für daten- und berechnungsintensive Aufgaben bei Entwurf und Test integrierter elektronischer Systeme. Für die Fehlerdiagnose von Mixed-Signal-Komponenten DAC und ADC konnte gezeigt werden, dass signifikante Verbesserungen der Diagnoseeffizienz in Form von Zeitgewinn, höherer Auflösung der Fehler (Netzwerk- anstelle Verhaltensebene) und Skalierbarkeit auf größere Verarbeitungsbreite der Komponenten erreichbar sind. Der Workflow zur Fehlerdiagnose basiert auf dem Ansatz, DAC und ADC in einer Loop-Back-Struktur zu analysieren und partitioniert die Diagnoseaufgabe in Teilschritte, die für eine Bearbeitung in der Cloud prädestiniert sind. Für die Organisation dieser Bearbeitung wurde das bei Fraunhofer entwickelte Werkzeug *GridWorker* weiterentwickelt. Als Software-Werkzeuge für die Bearbeitung der algorithmischen Teilaufgaben des eigentlichen Diagnoseprozesses kommen Public-Domain-Simulatoren und der bei Fraunhofer entwickelte analoge Fehlersimulator *aFSIM* zum Einsatz.

Ziel weiterführender Arbeiten ist es einerseits, die Skalierbarkeit unseres Ansatzes weiter zu verbessern und Voraussagen zur Laufzeit zu ermöglichen. Andererseits wird an einem nutzerfreundlichen Zugang gearbeitet, um bei Fraunhofer verfügbare Simulations-, Test- und Diagnoseverfahren (bzw. die darauf aufbauenden Werkzeuge) auch in der Cloud als Entwurfsdienstleistungen für externe Nutzer anbieten zu können.

Diese Arbeit wurde im Rahmen der BMBF-Projekte DIANA - Durchgängige Diagnosefähigkeit für Elektroniksysteme im Automobil (Kennzeichen 01M3188B) und OptiNum-Grid - Optimierung technischer Systeme und naturwissenschaftlicher Modelle mit Hilfe numerischer Simulationen im Grid (Kennzeichen 01IG0901D) gefördert.

7 Literatur

[1] *ADC and DAC Glossary*, MAXIM Application notes, Jul 22, 2012. http://www.maximintegrated.com/appnotes/index.mvp/id/641

[2] Gulbins, M., Vermeiren, V., Redlich, St.: *Fehlerdiagnose für AD- und DA-Wandler in einer Loop-Back-Struktur*. Dresdner Arbeitstagung Schaltungs- und Systementwurf (DASS 2012), Dresden, 3./4. Mai 2012

[3] Gulbins, M., Vermeiren, V., Redlich, St.: *Fehlerdiagnose für ADCs und DACs in einer Loop-Back-Struktur unter Einbeziehung von Parametervariationen*. Testmethoden und Zuverlässigkeit von Schaltungen und Systemen 25. GI/GMM/ITG-Workshop Dresden, 24. - 26. Februar 2013

[4] Straube, B.; Müller, B.; Vermeiren, W.; Hoffmann, C.; Sattler, S.: *Analogue fault simulation by aFSIM*. Design, Automation and Test in Europe Conference and Exhibition, DATE 2000 – User Forum, Paris, March 27-30, 2000, pp. 205-210.

[5] Schneider, A.: *Variantensimulation mit GridWorker*. ASIM-Workshop „Simulation technischer Systeme und Grundlagen und Methoden in Modellbildung und Simulation“, Krefeld, 24.-25. Februar 2011

[6] Dean, J.; Ghemawat, S.: *MapReduce*: *Simplified Data Processing on Large Clusters*. Sixth Symposium on Operating Systems Design and Implementation (OSDI '04), San Francisco, California, USA, December 6-8, 2004

[7] Schneider, A.: *Automatisierte Ressourcenbedarfsschätzung für Simulationsexperimente in der Cloud*. eingereicht zu Grid4Sys-Workshop „Grid-, Cloud- und Big-Data-Technologien für Systementwurf und -analyse“, Dresden, 27.-28. November 2013