

Design-Charakterisierung durch automatisierte Bewertung

Design characterization by automated assessment

Leif Müller¹, Roland Jancke¹ und Ronny Naumann²

¹Fraunhofer Institut Integrierte Schaltungen, Institutsteil Entwurfsautomatisierung, Dresden, Deutschland, {leif.mueller, roland.jancke}@eas.iis.fraunhofer.de

²DMOS GmbH, Dresden, Deutschland, ronny.naumann@dmos2002.de

Kurzfassung

Die Design-Charakterisierung durch eine automatisierte Bewertung von Schaltungseigenschaften stellt für Analog- und Mixed-Signal-Schaltungen aufgrund der Vielzahl von Bewertungskriterien, Betriebsbedingungen und technologisch bedingten Parameterschwankungen eine große Herausforderung dar. Verfahren zur Automatisierung der Schaltungsbeurteilung sind Gegenstand der Forschung. Im vorliegenden Beitrag wird ein Ansatz vorgestellt, bei dem für wiederkehrende Bewertungsaufgaben eine automatisierte Charakterisierung erfolgt. Voraussetzung dazu ist eine maschinenlesbare Spezifikation basierend auf dem Format XML. Damit lässt sich automatisch ein Bewertungsskript erzeugen und eine Beurteilung von Simulationsergebnissen bezüglich der spezifizierten Schaltungseigenschaften durchführen.

Abstract

The Variety of assessment criteria, operating conditions and technological corners of assessment tasks for integrated analog mixed signal circuits is a great challenge for the automated design characterization. In this paper we present an approach for automated characterization in periodic assessment tasks. A prerequisite to this is a machine-readable specification based on the XML format. This allows to automatically generate an executable script and to perform an assessment of simulation results with respect to the specified circuit properties.

1 Einleitung

Die Umsetzung einer Schaltung auf eine neue Technologie wird im Normalfall mehrere Lösungsvarianten zur Realisierung besitzen. Um die Einsetzbarkeit der Schaltungsvarianten prüfen zu können, werden Verfahren benötigt, die spezifizierte Randbedingungen berücksichtigen und auf diese Weise Varianten der portierten Schaltung aussortieren können [1].

Vor allem im Bereich von analogen und Mixed-Signal-Schaltungen sind die Varianten für eine Implementierung sehr vielfältig, wodurch der Test und die Bewertung heute noch immer viel Handarbeit erfordern. Umfangreiche Simulationsszenarien sind zu durchlaufen, ihre extrahierten Ergebnisse sind zu vergleichen und zu bewerten. Neben Ja/Nein-Aussagen sind im Allgemeinen auch Aussagen zur Größe der Abweichung, zur Einhaltung von Toleranzen und zu Tradeoffs zwischen verschiedenen Eigenschaften gefordert. Wenn dabei nicht nur skalare Kenngrößen, Schwellwerte und Zeitpunkte sondern analoge Signalverläufe zu vergleichen sind, wird es wesentlich aufwändiger, die Qualität einer entworfenen Schaltung bezüglich einer Spezifikation zu bestimmen.

Eine Methodik und adäquate Realisierungen, mit denen Schaltungsentwürfe automatisiert gegenüber einer Spezifikation bewertet werden können, stellen derzeit noch eine offene Herausforderung dar. Eine praktische Wiederverwendung skalierter Schaltungen ist aufgrund der fehlen-

den automatischen Verifikation kompliziert [2]. Ein dabei notwendiger Aspekt ist die eindeutige Dokumentation eingesetzter Bewertungsfunktionen. Für die Übergabe des Designs an einen anderen Simulator, einen anderen Entwerfer oder den Vergleich zwischen Messung und Simulation ist das Verständnis der zugrundeliegenden Messvorschrift von großer Bedeutung. Mit Blick auf den implementierten Algorithmus ist das für vorhandene Tools nicht immer gegeben. In [3] wird die Wichtigkeit von eindeutigen Algorithmen am Beispiel des Vergleichs zweier Waveforms angedeutet. Es wird gezeigt, dass es oft eine Vielzahl existierender Möglichkeiten gibt eine Bewertungsaufgabe mittels einer geeigneten Kenngröße zu repräsentieren.

Zur Bestimmung der Gütekriterien einer Schaltung existieren unterschiedliche Wege. Eine Möglichkeit ist die Verwendung von Extraktions-Anweisungen in der Simulations-Netzliste. Dabei wirkt sich nachteilig aus, dass Simulation und Bewertung nicht unabhängig voneinander ausgeführt werden können. Weiterhin besteht eine enge Bindung an die Formate eines Simulators und die Liste der Anweisungen wird gerade für komplexe Postprocessing-Aufgaben unübersichtlich. Eine zweite Möglichkeit stellt die skriptbasierte Automatisierung dar. Dadurch können für typische Schaltungsklassen und wiederkehrende Bewertungsaufgaben Postprocessing-Funktionen flexibel eingesetzt und mit eigenen Basisfunktionen erweitert werden. Die Postprocessings-Tools von

Cadence [4] und Mentor Graphics [5] bieten die Möglichkeit einer Skriptsteuerung.

Eine automatische Bestimmung der Gütekriterien ist somit für wiederkehrende Bewertungsaufgaben möglich und erfordert nun vor allem die korrekte und automatische Umsetzung der Spezifikationsangaben in ein ausführbares Skript für die betrachtete Schaltung. Eine in der Entwicklung befindliche Methode nutzt die formalisierte Erstellung einer Spezifikation basierend auf Extensible Markup Language (XML) [6]. Mit einer weitgehend automatisierten Skripterstellung werden zwei Ziele verfolgt. Einerseits lässt sich Zeitaufwand für die Erstellung der Verifikationsumgebung einsparen, zum anderen werden die Angaben aus der Spezifikation fehlerfrei an das Postprocessing übergeben.

Im Folgenden soll ein Flow vorgestellt werden, mit dem sich Bewertungsskripte automatisiert erstellen lassen.

2 Bewertungsflow

Ein vollständiger synthesesegerechter Analog-Designflow beinhaltet die nachfolgenden Punkte:

- Spezifikationserfassung,
- Schaltungsimplementierung,
- Schaltungssimulation und
- Schaltungsbewertung

Von diesem gesamten Designflow, der für den Entwurf analoger Schaltungen durchzuführen ist, betrachten wir hier nur die Schaltungsbewertung. Darunter verstehen wir eine vollständige Überprüfung von Schaltungsentwürfen gegen ihre Spezifikation. Dies ist ein wichtiger Teil innerhalb des Entwurfsablaufs, um eine Rückkopplung vom Ergebnis zum vorgegebenen Ausgangspunkt zu erreichen. Mit einer Automatisierung der Schaltungsbewertung wird das Ziel eines synthesesegerechten Designflows unterstützt.

Im Bild 1 ist der Gesamtablauf der Bewertung einer Schaltung gegenüber ihrer Spezifikation dargestellt.

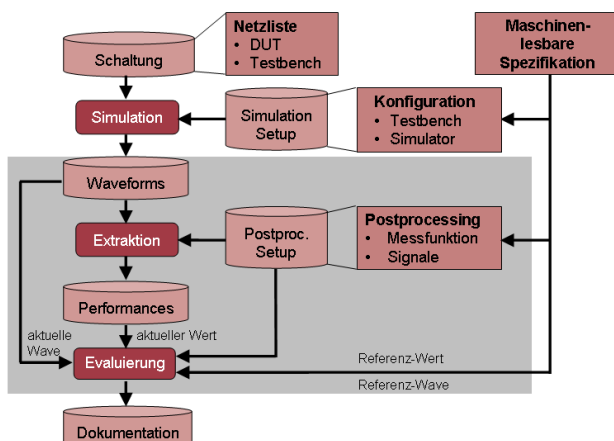


Bild 1 Bewertungsablauf der Schaltungsüberprüfung

Dabei wird der Weg von der Schaltung über Simulation, Parameterextraktion und Bewertung bis zur Dokumenta-

tion der Ergebnisse dargestellt. Die Eingangsdaten des Flows sind Netzlisten und Setups für Simulation und Extraktion. Die notwendigen Angaben für die Konfiguration von Simulation und Postprocessing stammen aus einer maschinenlesbaren Spezifikation [11]. Diese muss demzufolge Informationen dazu enthalten, welche Simulationen für die Bewertung einer bestimmten Eigenschaft durchzuführen sind und wie sich die charakteristische Kenngröße für diese Eigenschaft aus den Simulationsdaten extrahieren lässt.

Im Weiteren soll ausschließlich der in Bild 1 hinterlegte Teil des Flows betrachtet werden. Die Ergebnisse der Simulation und eine maschinenlesbare Spezifikation bilden dabei die Grundlage für Charakterisierung der Schaltung. Für die Extraktion der Schaltungskenngrößen werden Messfunktionen benötigt. Diese und die notwendigen Signale werden im Bewertungsskript definiert. Die Informationen zur Erstellung der Skripte kommen aus der maschinenlesbaren Spezifikation. Abschließend kann eine Beurteilung der extrahierten gegenüber den spezifizierten Schaltungseigenschaften erfolgen.

3 Erstellung der Skripte

Typische Schaltungsklassen können beim Analogentwurf häufig wiederverwendet werden und benötigen demnach auch immer wieder ähnliche Bewertungsaufgaben. Die Charakteristiken der Schaltungen müssen bei der Schaltungsoptimierung und der Verifikation immer wieder bestimmt und mit Spezifikationswerten verglichen werden. Es ist sinnvoll, wiederkehrende Bewertungsaufgaben durch eine skriptbasierte Arbeitsweise zu automatisieren. Zunächst wird jedoch die interaktive Arbeitsweise vorgestellt. Diese bietet sich an, wenn komplexe Bewertungsfunktionen erstellt und überprüft werden sollen. Der Vorteil ist hierbei die visuelle Überprüfbarkeit der korrekten Extraktionsbedingungen.

3.1 Interaktive Erstellung

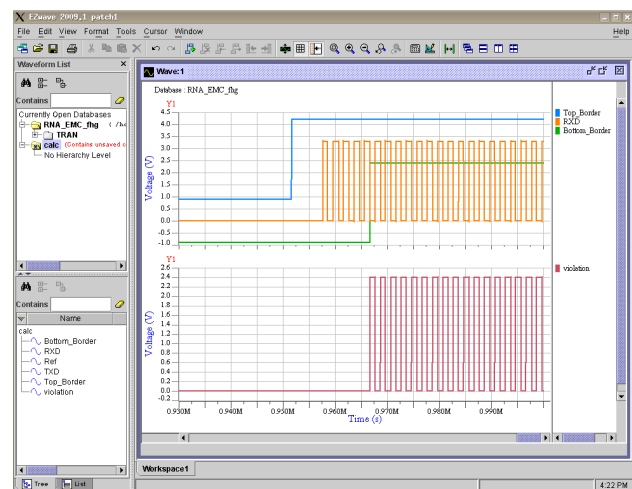


Bild 2 EZwaver mit Ergebnis der manuellen Beurteilung einer Toleranzschlauchverletzung

Im Bild 2 ist der Waveform-Viewer EZwave von Mentor Graphics [5] dargestellt. Mit Hilfe der Postprocessing-Funktionen des Calculators können Berechnungen auf den geladenen Signalen (z.B. RXD) durchgeführt werden.

Bewertungsszenarien werden schrittweise abgearbeitet und in mehreren Iterationsschleifen verfeinert. Die Historie der Bearbeitungsschritte kann als Datei abgespeichert und zu einem Skript angepasst werden. Bei der Bewertung einer Schaltung aus dieser Klasse kann das Skript im EZwave [5] gestartet werden. Wie in Bild 2 dargestellt, werden die Ergebnisse im dafür vorgesehenen Ordner organisiert abgelegt und, wenn erforderlich, zur Anzeige gebracht. Eine visuelle Kontrolle der Ergebnisse und der eingesetzten Triggerpunkte ist zeitnah möglich. Hierbei ist es von Vorteil, wenn eigens erzeugte Referenzsignale vor der Anwendung auf Korrektheit kontrolliert werden können.

Die manuelle Erstellung der Bewertungsskripte ist im Besonderen für einmalige Aufgaben eine sehr zeitintensive und fehleranfällige Vorgehensweise. Werden allerdings die interaktiv erstellten Skripte zur Beurteilung von unterschiedlichen Designs einer Schaltungsklasse genutzt so führt dies schon zu einer deutlichen Zeitersparnis. Weiterhin ist es nicht immer erforderlich, eine interaktive Unterstützung beim Postprocessing zu haben.

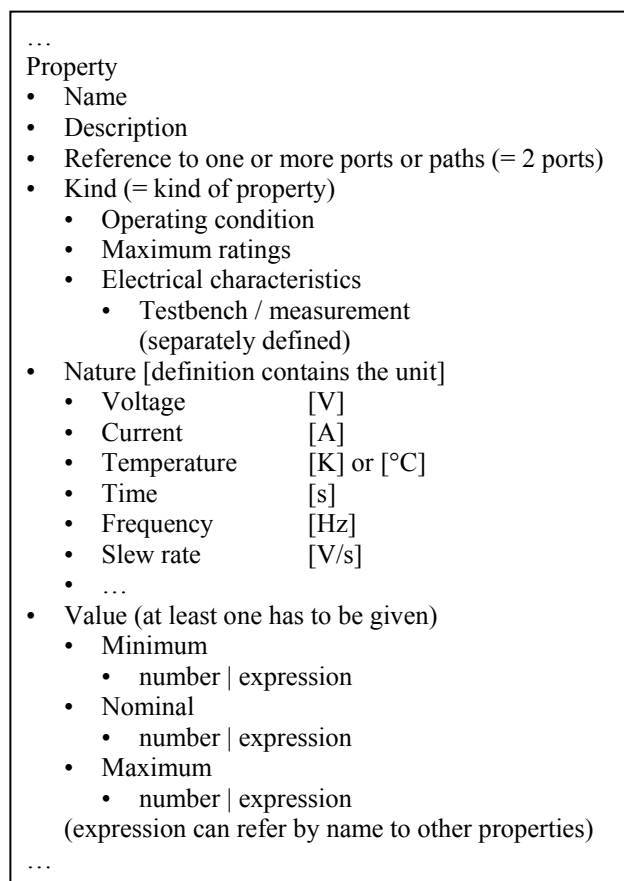


Bild 3 Strukturelle Beschreibung einer Schaltungseigenschaft als Teil der XML Spezifikationsbeschreibung

3.2 Skriptgenerierung aus XML-Spezifikation

Komfortabler ist eine automatisierte Erstellung der Bewertungsskripte. Das setzt zwingend eine formalisierte Beschreibung der Spezifikation voraus, aus der das Bewertungsskript abgeleitet werden kann. Allerdings wird die Spezifikation für eine Schaltung heute üblicherweise in Textform und mit Grafiken erstellt. Eine formalisierte Spezifikation, oder auch maschinenlesbare Spezifikation, kann man beispielsweise in XML codieren. Diese Markup-Language stellt ein verbreitetes Format zur Erfassung von strukturierten Beschreibungen dar [6, 7, 8].

Mit Hilfe von XML-Editoren und entsprechenden Eingabemasken können formalisierte Spezifikationen auf Basis von XML als Beschreibungssprache sehr einfach und komfortabel erstellt werden. Die maschinenlesbare Spezifikation der Schaltung ist unmittelbar für den Schaltungsentwurf und die Bewertung verwendbar. Ein Teil der Spezifikation können Vorschriften zur Bestimmung von geforderten Schaltungseigenschaften sein. In Bild 3 ist ein erster Vorschlag für die Beschreibung von Schaltungseigenschaften innerhalb der XML-Spezifikation dargestellt. Die strukturelle Beschreibung der gesamten Schaltungsspezifikation wird im Rahmen von SyEnA [9] stetig weiterentwickelt und ausgebaut [10].

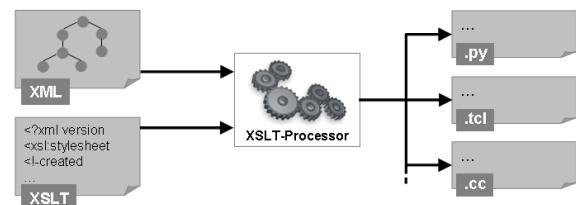


Bild 4 Skriptgenerierung mit XSLT-Transformation-Sheet

Mit Hilfe von XSLT-Transformation-Sheets (Extensible Stylesheet Language Transformations) können diese Vorschriften in Postprocessing-Skripte umgesetzt werden. In Bild 4 ist eine schematische Illustration zur Skriptgenerierung dargestellt. Dabei deutet die rechte Seite an, dass je nach vorhandenem Styles-Sheet ein entsprechendes Zielformat erzeugt werden kann. Die Transformation von XML-Dokument mittels XSLT-Transformation-Sheet wurde auf der Kommandozeile durchgeführt. Exemplarisch wurden ein Tcl- und ein Pythonskript generiert.

Die optionale Einbindung von Postprocessing-Aufgaben in die Spezifikation erfordert eine Erweiterung der XML-Beschreibung. Die Bewertungsfunktion einer spezifizierten Schaltungseigenschaft enthält dazu eine variable Anzahl von Informationen, welche zur eindeutigen Beschreibung dienen. Neben einer Funktionsbeschreibung werden vor allem die notwendigen Variablen und Parameter gegliedert und die zur Anwendung kommende Messfunktion beschrieben. Als ein verwendbares Format zur Beschreibung von mathematischen Ausdrücken eignet sich beispielsweise MathML [11], wie in Abschnitt 4 beschrieben. Eine mögliche Struktur eines Funktionsblo-

ckes zur Beschreibung einer Schaltungseigenschaft ist in Bild 5 dargestellt [10].

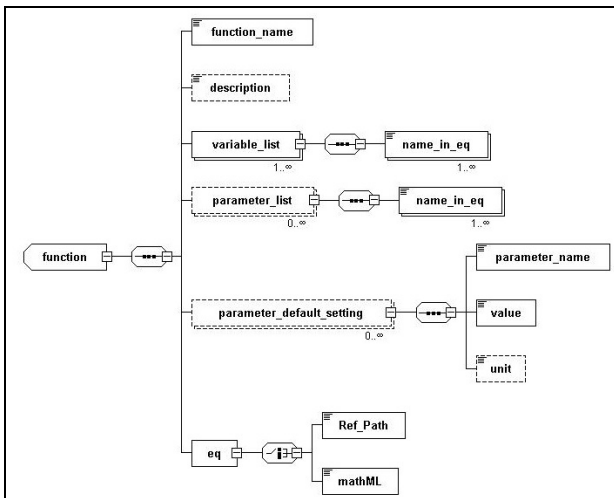


Bild 5 Struktur der XML Beschreibung für eine Postprocessing-Funktion

3.3 Skriptverarbeitung

Mit dem Umstieg auf eine skriptgesteuerte Bewertung von Simulationsergebnissen ist die Trennung von Simulation und Bewertung möglich. Eine Anwendung unterschiedlicher Skripte auf vorliegende Simulationsergebnisse ist damit unabhängig von der Simulation durchführbar. Die Bewertungsskripte können zum einen im Waveform-Viewer gestartet werden und die ermittelten Schaltungscharakteristika werden als skalarer Wert oder Waveform abgebildet. Andererseits ist es möglich, ohne interaktive Unterstützung eines Viewers eine Bewertung durchzuführen. Ein dafür erforderlicher Skriptaufruf erfolgt mittels eines eigens dafür in Java implementierten Werkzeugs. In dieser Betriebsart 'Batch-Mode', d.h. Postprocessing ohne grafische Visualisierung der Simulations- und Bewertungsergebnisse, begrenzt sich die Rückgabe ausschließlich auf skalare Werte.

4 Postprocessing-Bibliothek

Die Extraktionsalgorithmen in den Bewertungsskripten verwenden Basisfunktionen zur Ausführung mathematischer Algorithmen auf analogen Signalen. Diese Funktionen wurden standardkonform nach IEEE 181-2003 [12] in Java implementiert und als Bibliothek stetig weiterentwickelt. Das Wissen über den verwendeten Algorithmus und dessen Implementierung bietet dem Nutzer die Möglichkeit der direkten Einflussnahme. Dies wird vor allem deutlich vorteilhaft, wenn die Anzahl der betrachteten Applikationen einer Schaltungs-kategorie zunimmt und somit die Variantenvielfalt einer Bewertungsfunktion steigt. Die Komplexität einer differenzierten Beurteilung wird auch in [3] deutlich. Weiterhin stellt die Portierbarkeit von

Bewertungsfunktionen auf andere Design-Systeme einen großen Vorteil dar.

Für die Definition einer Bewertungsfunktion müssen eindeutige Festlegungen zu folgenden Fragen getroffen werden: was ist zu bewerten, wie ist zu bewerten und unter welchen Randbedingungen. Am Beispiel der Funktion Slew Rate wird das in den nachfolgenden Punkten verdeutlicht (Bild 6):

- Messfunktion
Slew Rate
- Messbeschreibung
 $V(2)-V(1)/(T2-T(1))$
[beschreibbar mit MathML]
- Messparameter
Edge – fallend/steigend,
Level – s_1 und s_2
Reference Level – 10% und 90%
Betrachtungszeitraum – start und Stopp

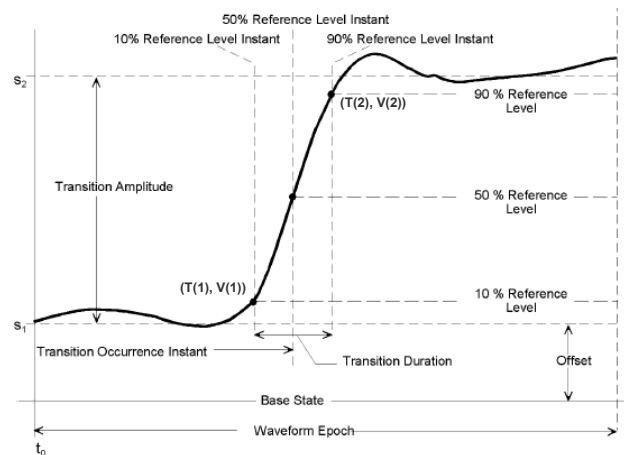


Bild 6 Definition wichtiger Größen am positiven Anstieg eines Signals [12]

Aufgrund der Vielzahl von möglichen Postprocessing-Funktionen sind diverse Einteilungen nach Kategorien möglich und sinnvoll. Innerhalb dieser Kategorien wird jeweils unterschieden nach Grundfunktionen und komplexen Funktionen, welche auf den Grundfunktionen aufbauen. Das Spektrum an Funktionen reicht von mathematischen Funktionen bis hin zu solchen, die auf spezielle Disziplinen des Schaltungsentwurfs abzielen. Die nachfolgende Auflistung zeigt denkbare Einteilungsvarianten:

- Einteilung nach den Grundkategorien der Argumente und Ergebnisse:
(z.B. Skalar zu Skalar, Wave zu Skalar, Wave zu Wave)
- Einteilung nach Anwendungsgebiet
(z.B. Analog/ Digital, HF/ NF, Signalverarbeitung)
- Einteilung nach Domänen
(z.B. Zeitbereich/ Frequenzbereich)

Für die vorliegende Bibliothek wurde eine Unterteilung nach den Domänen Zeit und Frequenz vorgenommen. Die Basiskategorie Wave enthält Funktionen (z.B. Base- und

Topline, Slope, xval, yval), welche in allen Subkategorien benötigt werden. Weiterhin erfolgte eine Unterteilung in die Subkategorien Timewave (z.B. mit den Funktionen Periode, Frequency, Slew Rate, Settlingtime) und Frequencywave (z.B. mit den Funktionen F3db, Transitfrequency, Bandwidth).

Für die Beschreibung mathematischer Ausdrücke von Bewertungsfunktionen bietet sich die XML-Erweiterung MathML [11] an. Damit können Algorithmen von Bewertungsfunktionen in einer Form notiert werden, die unabhängig von Programmiersprachen oder einer konkreten Implementierung ist. Von dieser Beschreibungsform aus kann später die Implementierung in eine beliebige Skript- oder Codesprache erfolgen.

Weiterhin ist es möglich, aus der MathML-Beschreibung des Algorithmus eine Dokumentation für die betreffende Bewertungsfunktion abzuleiten. Im einfachsten Fall handelt es sich um eine grafische Notation der implementierten Gleichungen aber auch andere algorithmische Beschreibungen sind denkbar. Damit kann bei der Erstellung der Spezifikation für die Bewertung bereits festgelegt und zur späteren Referenz dokumentiert werden, wie die einzusetzende Bewertungsfunktion anzuwenden ist.

Bei der Beschreibung mathematischer Algorithmen in MathML stellt sich allerdings das Problem, dass für beide umrissene Anwendungsfälle jeweils ein anderer Dialekt vorgeschlagen ist. In MathML sind zweierlei Klassen von Tags definiert, je nachdem ob die Beschreibung der mathematischen Formel für die Codegenerierung oder für deren grafische Darstellung verwendet werden soll. Die Verwendung zweier Dialekte ist ein Hindernis bei der Erstellung einer einheitlichen Datenbasis für die Bewertungsalgorithmen. Zukünftig sollte nach alternativen mathematischen Beschreibungsformen gesucht werden, die diese Unterscheidung nicht erfordern.

5 Ergebnisse

5.1 Interaktiv erstellte Skripte für die Schaltungsklasse Operationsverstärker

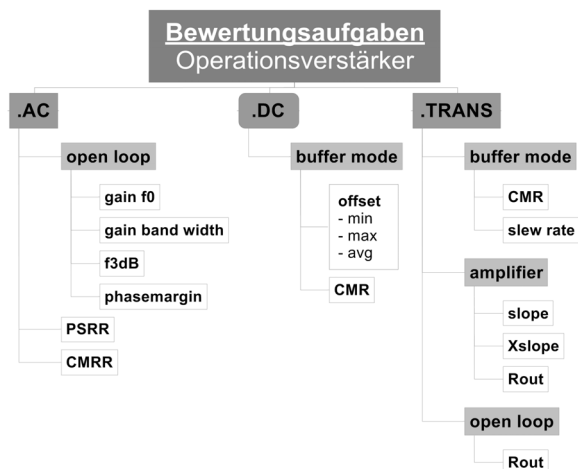


Bild 7 Übersicht Bewertungsaufgaben

In Bild 7 ist eine Übersicht der Bewertungsaufgaben für einen Operationsverstärker dargestellt. Am Beispiel der Funktion Slew Rate soll die interaktive Erstellung eines Bewertungsskripts skizziert werden. Für die Beurteilung der weiteren Schaltungseigenschaften existieren analog entsprechende Skripte.

Die Testbench in Bild 8 dient zur Bestimmung der Slew Rate. Der Ausgangsport "OUT" wird kapazitiv belastet und mittels Spannungsquelle V_{FB_INN} auf den Eingangsport "INN" zurückgeführt. Der Eingangsport "INP" wird durch eine PWL-Quelle V_{IN_CM} innerhalb von 100 ns zwischen der oberen und unteren Grenze des zulässigen Eingangsbereiches umgeschaltet.

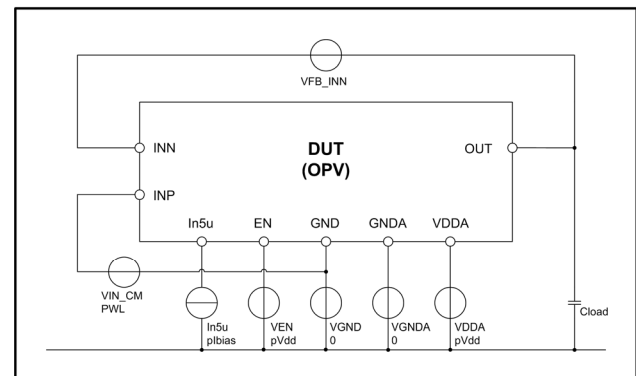


Bild 8 Testbench zur Extraktion der Schaltungseigenschaft Slew Rate

Die Bewertung der Simulationsergebnisse erfolgt in der Regel innerhalb eines Waveform-Viewers. Der hier verwendete EZwave [5] bietet viele Messfunktionen innerhalb des Waveform Calculators, um Schaltungseigenschaften zu bestimmen. Die Operationen können in ein Skript (Bild 9) für die automatisierte Bewertung exportiert werden.

```
# parameter
tStart = 100E-3
tCycle = 1E-3

# calculation
slewrise_Rise = slewrise( V(OUT), up = "90%",
low = "10%", x_start = tStart, x_end = 5E-3 )
slewrise_Fall = slewrise( V(OUT), up = "90%",
low = "10%", x_start = tStart+ tCycle, x_end = 5E-3 )

# output
toResultFile(abs(slewrise_Rise))
toResultFile(abs(slewrise_Fall))
```

Bild 9 Bewertungsskript zur Bestimmung der Slew Rate

Eine weitere hier untersuchte, vielversprechende Methode zur Generierung von Bewertungsskripten ist die Verwendung des EZwave Measurement Tools, welches grafisch durch die Messfunktionen und deren Parametrierung führt. Mit der anschließenden Konvertierung des Saved

Window Files in eine XML Beschreibung lassen sich auf komfortablem Weg syntaktisch richtige und durch grafische Markierung im Signaldiagramm semantisch überprüfte Messfunktionen generieren. Dann entsteht mit insgesamt geringem Aufwand und durch eine geringfügige Nachbearbeitung ein Bewertungsskript wie das in Bild 9 dargestellte zur Bestimmung der Slew Rate eines Operationsverstärkers. Die generierten Absolutwerte für die Zeitfenster wurden hier durch die Parameter tStart und tCycle ersetzt und die Ausgabe der Ergebnisse überarbeitet.

5.2 Automatische Skripterstellung und dessen Anwendung am Beispiel eines LIN-Transceivers

Am Beispiel eines LIN-Transceivers wird die Störeinstrahlungsfestigkeit beurteilt (Bild 10). Wie in Abschnitt 3.2 beschrieben wird das Bewertungsskript ausgehend von einer maschinenlesbaren Spezifikation mittels XSLT-Transformation-Sheet automatisiert erstellt. Das Skript wurde im Waveform-Calculator geladen und erhält als Eingangsgröße das Simulationsergebnis als Waveform.

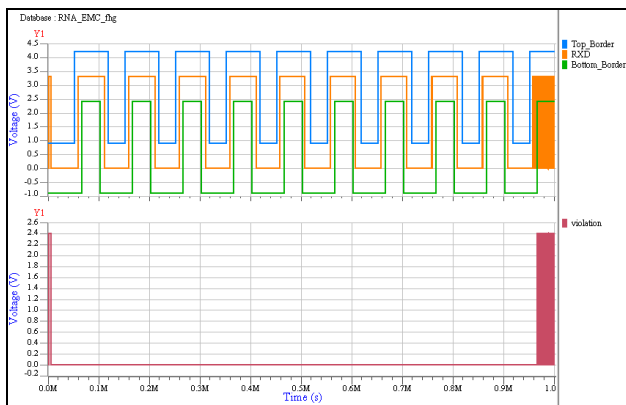


Bild 10 Simulationsergebnis für einen LIN-Transceiver bei der Bewertung der Festigkeit gegenüber Störeinstrahlung

Im Folgenden wird aus dem Transmitted eXchange Data (TXD, als goldene Referenz) ein Toleranzschlauch erzeugt, um damit das Received eXchange Data (RXD) zu bewerten. Toleranzschlauch und RXD-Signal sind im oberen Diagramm von Bild 10 dargestellt. Verletzungen des Toleranzschlauches durch das zu bewertende Signal RXD werden als neue Wave Violation im unteren Diagramm angezeigt.

Beim Start von EZwave auf der Kommandozeile werden folgende Parameter übergeben:

- Simulationsergebnis in Form eines wdb-Files
- Tcl-Skript einer selbstdefinierten Funktionen
- Tcl-Skript für die Bewertung

6 Zusammenfassung

In dieser Arbeit wurde, ausgehend von einer maschinenlesbaren Spezifikation basierend auf dem XML Format, die Generierungen von Bewertungsskripten vorgestellt. Es zeigte sich, dass für wiederkehrende Charakterisierungsaufgaben eine skriptbasierte Bewertungsmethode viele Vorteile bietet. Mit der automatisierten Skriptgenerierung aus der formalisierten Spezifikation können die Daten direkt an das Postprocessing übergeben werden. Wir haben damit den Analog-Design-Flow teilweise automatisiert und an ausgewählten Beispielen aus der Praxis getestet. Die Testschaltungen wurden von der DMOS GmbH bereitgestellt. Zukünftig soll das vorgestellte Verfahren weiter automatisiert und in den produktiven Design-Flow implementiert werden.

7 Danksagung

Dieser Beitrag ist im Projekt SyEnA entstanden. SyEnA wird unter dem Förderkennzeichen 01 M3086 im Förderprogramm IKT 2020 teilweise durch das BMBF gefördert.

8 Literatur

- [1] Sobe, U.; Graupner, A.; Böhme, E.; Ripp, A.; Pro-nath, M.: Conversion and Optimization Flow for Analog IP Porting, DAC, 2011
- [2] International Technology Roadmap for Semiconduc-tors, 2009 Edition
- [3] Ohlendorf, O., Steinhorst, S., Hedrich, L.: Comparing Two Analog Waveforms – A Trivial Task?, Sep-tember 2008, ZUE in Ingolstadt
- [4] <http://www.cadence.com>
- [5] <http://www.mentor.com>
- [6] IEEE Standard Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML, IEEE 1671 ATML
- [7] Ma, M.; Hedrich, L.; Sporrer, C.: An Attempt: Specification Descriptions of AMS-Systems using a XML-Format., November 2010, IP-XACT Work-shop Grenoble
- [8] IEEE Standard for Signal and Test Definition, IEEE Std 1641
- [9] <http://www.edacentrum.de/syena>
- [10] Ma, M.; Hedrich, L.; Sporrer, C.: A Machine-readable Specification of Analog Circuits for Inte-gration into a Validation Flow, September 2011, FDL in Oldenburg
- [11] <http://www.w3c.org/Math/>
- [12] IEEE Standard on Transitions, Pulse and Related Waveforms, IEEE Std 181-2003