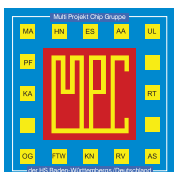


MPC

MULTI PROJEKT CHIP GRUPPE
BADEN - WÜRTTEMBERG

Herausgeber: Hochschule Ulm Ausgabe: 53 ISSN 1868-9221 Workshop: Esslingen Februar 2015

- 1 ASICs für medizinische Geräte und Implantate**
M. Nawito, H. Richter, C. Scherjon, J. N. Burghartz, IMS Chips Stuttgart
- 7 Low-Power-SC-Wandler mit hoher variabler Eingangsspannung**
D. Lutz, P. Renz, B. Wicht, Robert Bosch Zentrum für Leistungselektronik, Reutlingen
- 13 Entwicklung und Layoutentwurf eines Analog-Digital-Wandlers mit 12 Bit Auflösung in einer 180 nm-CMOS-Technologie**
B. Dusch, M. Bhattacharya, D. Jansen, HS Offenburg
- 23 Konzeptstudie eines durchgängig auf parametrisierten Modulgeneratoren basierenden Entwurfsflusses für Analogdesign**
M. Greif, D. Marolt, J. Scheible, Robert Bosch Zentrum für Leistungselektronik, Reutlingen
- 31 Hardware to Software Migration and Optimization of a Serial Peripheral Interface**
M. Freier, A. Wenzler, T. Mayer, Robert Bosch GmbH
K. Schulz, J. Gerlach, HS Albstadt
W. Rosenstiel, Uni Tübingen
- 39 A Flexible Approach on FPGA-based Digital Sensor Interfacing for Cyber Physical Systems**
I. Verdu, K. F. Ackermann, HS Mannheim



Cooperating Organisation
Solid-State Circuit Society Chapter
IEEE German Section

Tagungsband zum Workshop der Multiprojekt-Chip-Gruppe Baden-Württemberg

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie.

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

Herausgeber:

Gerhard Forster, Hochschule Ulm, Prittwitzstraße 10, D-89075 Ulm

Mitherausgeber (Peer Reviewer):

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

Frank Kesel, Hochschule Pforzheim, Tiefenbronner Straße 65, D-75175 Pforzheim

Axel Sikora, Hochschule Offenburg, Badstraße 24, D-77652 Offenburg

Alle Rechte vorbehalten

Diesen Workshopband und alle bisherigen Bände finden Sie im Internet unter:

<http://www.mpc.belwue.de>

Konzeptstudie eines durchgängig auf parametrisierten Modulgeneratoren basierenden Entwurfsflusses für Analogdesign

Matthias Greif, Daniel Marolt, Jürgen Scheible

Zusammenfassung—Im Bereich integrierter Schaltungen (ICs) für die Fahrzeugelektronik ist in den letzten Jahren ein Trend zum Einsatz komplexer Mixed-Signal-Komponenten erkennbar. Dies führt dazu, dass ein altes Problem zunehmend in den Fokus der EDA-Entwickler rückt: Während der digitale Entwurfsfluss hoch automatisiert ist, findet der Entwurf analoger Komponenten überwiegend in einem manuellen, zeitaufwändigen und interaktiven Entwurfsstil statt. Die folgende Arbeit beschreibt ein Konzept, diesen Mangel mit Hilfe eines durchgängigen analogen Entwurfsflusses unter Verwendung so genannter Modul-Generatoren zu mildern. Der vorgestellte Ansatz zur Erzeugung von Schaltkreis-Automatismen berücksichtigt die implizite Nutzung von Erfahrungswissen des Designers, bietet eine volle Topologie-Flexibilität und steigert die Wiederverwendung („re-use“) gängiger Schaltungstopologien. Die erreichten Zwischenergebnisse lassen einen erheblichen Nutzen erkennen und zeigen das Potenzial sogenannter „Parametrisierter Schaltkreise“ auf, den Automatisierungsgrad des analogen Schaltungsentwurfs zu steigern.

Schlüsselwörter—Electronic Design Automation EDA, Generatoren, IP, re-use, Expertenwissen, Analog Design, Analoger Entwurfsfluss, Constraints, Dimensionierung.

I. EINLEITUNG

Electronic Design Automation (EDA) ist im Bereich integrierter Schaltungen nicht weg zu denken. Seit mehreren Dekaden wird – parallel zu den Weiterentwicklungen im Bereich der Technologie – die Funktionalität der Hilfsmittel für den Schaltungsdesigner vorangetrieben. Hierbei hat sich ein erheblicher Unterschied zwischen den Automatisierungsgraden im Bereich von analoger und digitaler Welt ergeben. Mit der Einschränkung von Entwurfsfreiheitsgraden (z.B. Einsatz von Standardzellen und layerspezifische Vorzugs-

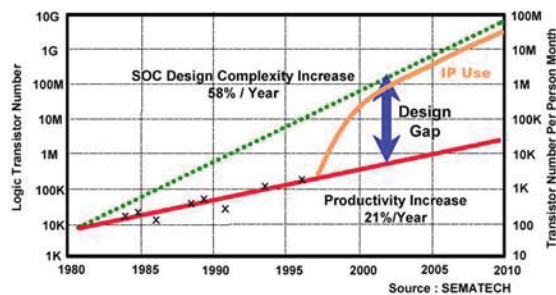


Abbildung 1: Design-Komplexität und -Produktivität über die letzten zwei Dekaden. (Quelle: http://www.eetimes.com/document.asp?doc_id=1279120).

richtungen der Verdrahtung) wurde der Automatisierungsgrad im Entwurfsfluss (engl. Design-Flow) von digitalen integrierten Schaltkreisen erheblich gesteigert. Durch die Fokussierung der EDA-Entwickler auf die digitale Welt ist dabei jedoch leider die Entwicklung eines automatisierten analogen Design-Flows in den Hintergrund gerückt. Bis heute ist der analoge IC-Entwurf größtenteils manuell geprägt. Sicher ist ein guter Handentwurf (manueller Entwurfsfluss) – ohne auf die Entwicklungszeit zu achten – einer algorithmisch entworfenen Schaltung vorzuziehen, jedoch spielen Forderungen nach möglichst schnellem Markteintritt („time-to-market“) und dadurch getrieben auch nach einer hohen Wiederverwendbarkeit („re-usability“) von Designs in Zeiten großen Preis- und Konkurrenzdrucks, denen industrielle IC-Unternehmen ausgesetzt sind, eine wichtige Rolle.

Abbildung 1 zeigt die Kluft zwischen der rasch ansteigenden Design-Komplexität und der nachhinkenden Design-Produktivität (Design Gap). Daraus lässt sich schlussfolgern, dass die Entwicklung von EDA-Software für analoge ICs deutlich hinter dem Bedarf der industriellen Anwender zurückbleibt. Nun ist es an der EDA-Community, die Herausforderung anzunehmen, diese Kluft mit Hilfe neuer innovativer Automationsansätze für die Design-Werkzeuge zu verringern. Die bisher verfolgten Ansätze im Analogbereich blieben meist im akademischen Stadium stecken. Als Beispiel kann hier das Optimierungstool OPTIMAN zusammen mit ISAAC, einem grafischen Simulator, genannt werden [1]. Hierbei übernimmt ISSAC die Modellie-

Matthias Greif, Daniel Marolt, Jürgen Scheible, {matthias.greif; daniel.marolt; juergen.scheible}@reutlingen-university.de, Robert Bosch Zentrum für Leistungselektronik, Hochschule Reutlingen, 72768 Reutlingen.

rung der Schaltung, anschließend wird die Dimensionierung des entwickelten Schaltungs-Modells in OPTIMAN mittels eines *simulated annealing*-Verfahrens optimiert. Weitere akademische Beispiele sind unter anderem STAIC [2], oder IDAC [3].

Auch kommerzielle Tools wie zum Beispiel ANALOG INSYDES, ISTONE von IPGEN oder ANACONDA von SAGANTEC werden auf dem Markt angeboten, stoßen aber auf wenig Akzeptanz beim Analog-Designer. Gründe hierfür sind von unterschiedlicher Natur, wie etwa das Misstrauen in Automatisierungsansätze eines fachfremden Programmierers, oder die Abneigung, die bewährte grafisch-manuelle Eingabemethode durch eine kryptische Konsoleneingabe zu ersetzen. Unser Ansatz unterscheidet sich hier von den bisherigen. Bis zum heutigen Zeitpunkt sind die meisten Ansätze zur Automatisierung des Analog-Flows algorithmisch und simulationstechnisch geprägte Optimierungsverfahren für den Designschritt der Dimensionierung einer Schaltung [4]. Andere Ansätze definieren abstrakte Beschreibungssprachen zur Eingabe von hierarchischen Schaltungsbeschreibungen, welche zusammen mit einem symbolisch/numerischen Gleichungssimulator zu „flachen“ Modell-Lösungen, entsprechend der durch den Anwender definierten Topologie- und Performance-Definition, führt [2]. Das hier beschriebene Konzept der parametergesteuerten Generatoren für den Schaltungsentwurf („Frontend“) auf Modulebene bietet dabei einen praktikablen und bisher nicht explizit verfolgten Ansatz, die Design-Effizienz zu erhöhen.

Die vorliegende Arbeit baut sich wie folgt auf: Abschnitt II zeigt den Stand der Technik parametergesteuerter Generatoren und gibt einen Einblick in deren Eigenschaften und Vorteile. Kapitel III gibt eine Übersicht zum Gesamtkonzept parametergesteuerter Schaltplan-Generatoren („Schematic Modul-PCells“) und stellt den Gesamtentwurfsfluss auf Modulebene dar. Hierbei unterscheiden wir zwischen dem Entwurf aus Sicht des PCell-Entwicklers und aus Sicht des späteren Anwenders der erstellten PCell. In IV wird der Entwurfsprozess eines Schaltplan-Modul-Generators aufgezeigt. Anschließend folgt ein Implementierungsbeispiel eines parametrisierten Schaltplans mit einem abschließenden Fazit.

II. PARAMETERGESTEUERTE GENERATOREN

Mit parametergesteuerten Generatoren lässt sich nicht nur die Design-Effizienz, sondern auch die Qualität des Design-Ergebnisses signifikant steigern. Bei den parametergesteuerten Generatoren handelt es sich um Ablaufskripte, welche auf Grundlage von Expertenwissen erstellt werden und das Vorgehen des menschlichen Designers nachahmen. Das bedeutet, ein Experte – hier auf dem Gebiet des Analog-Designs – nutzt seine Expertise, dafür einen Code-basierten Generator zu erstellen, welcher die Eigenschaft besitzt, das inkludierte Expertenwissen jederzeit und unabhängig von der Designerfahrung des Anwenders reproduzierbar

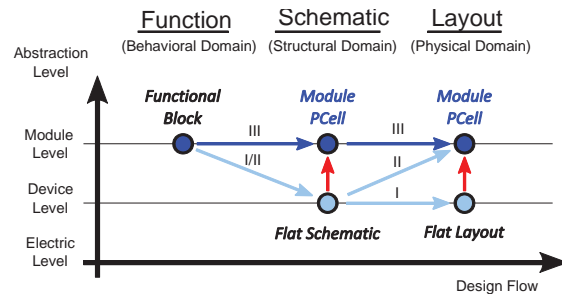


Abbildung 2: Schematic-Driven-Layout (SDL) Design Flow; (I) Konventioneller SDL-Flow; (II) Entwurfsfluss mit Layout-Modul-PCell; (III) Einheitlicher SDL-Flow auf Modulebene.

wiederzugeben. Parametergesteuerte Generatoren sind bereits heute gängiger Gebrauch in der Layout-Erstellung. Ein sehr weit verbreiteter Ansatz ist das sogenannte *PCell*-Konzept des US-amerikanischen EDA-Software-Anbieters CADENCE DESIGN SYSTEMS®, basierend auf der werkzeugspezifischen Skriptsprache SKILL. Andere EDA-Firmen setzen auf allgemein verfügbare Programmiersprachen, wie zum Beispiel SYNOPSIS®, deren sogenannte „PyCells“ in der Open-Source-Sprache Python implementiert werden. Aufgrund der marktbeherrschenden Stellung der Cadence-Werkzeuge im Bereich des analogen IC-Entwurfs hat sich in der Fachsprache der Name PCells als Synonym für parametergesteuerte Generatoren eingebürgert und wird deshalb im Folgenden auch hier verwendet.

Im Layout (häufig auch als „Backend“ bezeichnet) ist die Verwendung von PCells Stand der Technik, allerdings zielt die Anwendung meist auf einzelne parametrisierte Bauelemente ab. Weitere Vorteile des PCell-Konzepts sind:

- Erhöhter Automatisierungsgrad, somit Zeitersparnis bei der Layout-Erstellung
- Reproduzierbare, gute Layout-Ergebnisse
- Inkludiertes Expertenwissen, Ergebnis ist nicht von der Expertise des Benutzers (Anwenders) abhängig
- Ergebnis ist vergleichbar mit gutem Handentwurf (abhängig von der Expertise des PCell-Entwicklers)
- Implizite (verlustfreie) Umsetzung von Constraints (schaltungstechnische Randbedingungen).

Moderne Entwicklungen, wie der PCell Designer bieten dem Layout-Entwickler eine grafische Benutzeroberfläche, welche ihm die Entwicklung von Modul-PCells entscheidend erleichtert, da diese die grafikbasierte Arbeitsweise der Layouter unterstützt und damit ihrer Denkweise entgegenkommt. Als Folge daraus ergibt sich nun die Möglichkeit, das Konzept der parametergesteuerten Generatoren auch auf ganze Module, also komplette Schaltungen oder Schaltungsteile im Layout anzuwenden. Diese Module im Backend enthalten eine Vielzahl an wiederum parametrisierten Bauelementen und deren Verbindungen. Die Weiterentwicklung zur Modul-PCell im Design-Flow integrierter

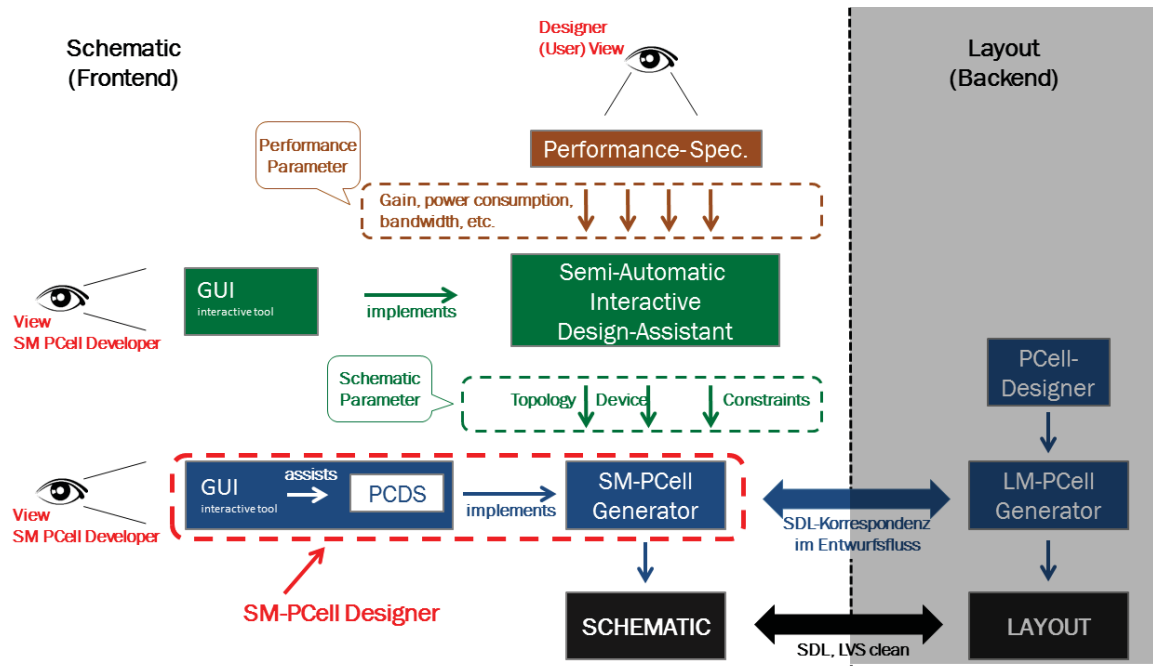


Abbildung 3: Gesamtkonzept Schematic-Modul-PCell aus Designer- (User) und SM-PCell-Entwickler-Sicht.

Schaltungen führt allerdings zu einem Nachteil. Es ergibt sich ein Hierarchiebruch im Designfluss. Nach dem Gajski-Diagramm [5] (auch als Y-Diagramm bekannt) ergeben sich die Abstraktionsebenen Verhalten, Struktur und Geometrie (in Abbildung 2 als *Behavioral*, *Structural* und *Physical Domain* beschrieben). In der Abbildung sind auf der x-Achse die Domänen („Sichten“) und auf der y-Achse die Hierarchiestufen abgetragen. Durch die Einführung einer Layout Modul-PCell (LM-PCell) resultiert ein Wechsel der Hierarchieebenen am Übergang zwischen den Domänen (Abbildung 2: Entwurfsfluss II). Hierarchiebrüche verhindern die Durchgängigkeit des Entwurfsflusses. Eine flache Schaltung auf Strukturebene verhindert die Korrespondenz des SDL-Flows, welche nur innerhalb einer Abstraktionsebene funktioniert. Ziel der Weiterentwicklung ist daher eine grundsätzliche Vermeidung von Hierarchiebrüchen mit dem Ziel eines durchgängigen SDL-Entwurfsflusses auf Modul-Ebene. Hierfür ist für die LM-PCell ein korrespondierendes Design-Element in der Strukturdomäne zu schaffen. Diese besteht in einem parametrisierten Schaltplan, den wir ebenfalls mit dem PCell-Konzept realisieren und daher als Schematic Modul-PCell („SM-PCell“) bezeichnen. Dieser Ansatz ist in Abbildung 2, Entwurfsfluss III dargestellt.

III. KONZEPT DER SM-PCELL

Ein wichtiger Punkt, die Akzeptanz des Designers gegenüber dem analogen Automationstool zu erhöhen, ist, dem Designer während des gesamten Design Flows die vollständige Kontrolle über die Schaltungsentwicklung zu gewährleisten. Dieser Punkt wurde bei vielen bisherigen Ansätzen nicht explizit berücksichtigt. Ein

nicht-algorithmischer Generator-Ansatz, beruhend auf Expertenwissen, zusammen mit der Einbindung eines Tools zur Generatorentwicklung in eine dem Designer bekannte Entwurfsumgebung führt unserer Meinung nach zu einer höheren Akzeptanz durch den Analog-Entwickler. Der im Folgenden beschriebene methodische Bottom-Up-Ansatz, welcher gleichzeitig eine Erweiterung des bewährten SDL-Flows darstellt, soll den Designer nicht zu einer Abkehr von Altbewährtem bewegen, sondern eine evolutionäre Weiterentwicklung der bisherigen analogen Design-Methodik darstellen.

Das SM-PCell Konzept soll dem Analog-Entwickler ein, seiner Entwurfsmethodik nachempfundenen, Werkzeug zur Schaltungsgenerierung bieten. Es geht dabei nicht um einen weiteren Versuch einer vollautomatischen Schaltungssynthese. Vielmehr soll ein Verfahren entwickelt werden, welches den Entwickler unter Einbeziehung von Expertenwissen unterstützt. Das Gesamtkonzept gliedert sich in zwei verschiedenen Entwurfsflüssen und sollte zwischen den Sichtweisen des PCell-Entwicklers und des PCell-Anwenders differenzieren. Abbildung 3 zeigt die beiden Entwurfsflüsse, inklusive des Backend-Entwurfsflusses mit dargestellter Layout Modul-PCell. (Auf den Layout-Entwurf wird im Rahmen dieses Artikels nicht weiter eingegangen). Horizontal dargestellt ist der Entwurfsfluss für den SM-PCell-Generator aus der Sicht des PCell-Entwicklers. Vertikal dargestellt ist der Entwurfsfluss für die Eingangsparameter der SM-PCell aus Anwendersicht. Beide Entwurfsflüsse enthalten Schaltungsklassen-spezifische Eigenschaften und müssen für jede Schaltungsklasse individuell erstellt werden. Das Konzept begründet zwar einen einmaligen Mehraufwand bei der Erzeugung der spezifischen Komponenten

(Aufgabe des PCell-Entwicklers), ergibt aber einen immensen Vorteil bei der Wiederverwendung (re-use) auf Grund des Generator-Charakters (PCell-Anwender). Im Folgenden sollen die zwei angedachten Entwurfsflüsse aus Anwender- und Entwicklersicht konkretisiert werden.

A. Die SM-PCell aus Anwendersicht

Wir betrachten zunächst den Entwurfsfluss aus Sicht des Anwenders der SM-PCell. Dieser nutzt die in Teil B beschriebenen unterstützenden Entwurfswerkzeuge. Allgemein besteht die Aufgabe darin, aus den für den Schaltungsentwickler relevanten Schaltungseigenschaften (im Folgenden als „Performance-Parameter“ bezeichnet und in Abbildung 3 oben dargestellt) die Schaltplanparameter („Schematic-Parameter“) zu generieren. Dies ist gleichbedeutend mit der Übertragung einer Spezifikation in einen elektrischen Schaltplan und stellt damit nichts anderes dar als die originäre Aufgabe der Schaltungsentwicklung selbst. Dabei soll dem Anwender ein interaktives und teilautomatisiertes Werkzeug zur Verfügung stehen (im weiteren Verlauf auch als Design-Assistent bezeichnet). Die aus diesem Entwurfsfluss ermittelten Schematic-Parameter werden an die SM-PCell weitergegeben, welche daraus den fertigen Schaltplan generiert.

Konkret soll der Entwurfsfluss wie folgt aussehen: Der Anwender definiert – auch auf Grundlage der Schaltungsspezifikation – die spezifischen Performance-Parameter, wie zum Beispiel *gain*, *power consumption*, *bandwidth*, uvm.. Diese werden dem Design-Assistenten (in Abbildung 3 als *Semi-Automatic interactive Design-Assistant* bezeichnet) als Eingangswerte übergeben. Der Design-Assistent soll anhand algorithmisch-optimierender Verfahren, unter Einbeziehung des enthaltenen (formalisierten und nicht-formalisierten) Expertenwissens und durch Interaktion mit dem Anwender, diesen dabei unterstützen, die entsprechenden Schaltplanparameter zu Topologie, Dimensionierung und Constraints zu ermitteln. Die Schaltplanparameter werden als Eingangsparameter an den Schaltungsgenerator (SM-PCell) weitergereicht. Der Generator entwickelt daraus den dimensionierten Schaltplan. Zusammengefasst erzeugt der Design-Assistent aus den Performance-Parametern die für die SM-PCell benötigten Schaltplanparametersätze zu Topologie, Bauteildimensionierung und Constraints.

B. Die SM-PCell aus Entwicklersicht

Bei der Beschreibung der Vorgehensweise aus Sicht des Anwenders wurden sowohl der Schaltungsgenerator (SM-PCell) als auch der Design-Assistent als gegeben betrachtet. Dies ist – zumindest bei der Erstimplementation der Schaltungsklasse – nicht der Fall. Daher soll im Folgenden der Entwurfsfluss aus Sicht des PCell-Entwicklers näher erläutert werden. Bei der Entwicklung des Schaltungsklassen-spezifischen Design-Assistenten soll das Erfahrungswissen eines Designers

effizient erfasst und verfügbar gemacht werden. Hierfür sollten dem Entwickler unterstützende Methoden angeboten werden, wie etwa interaktive Werkzeuge, deren spezifische Oberfläche der Arbeits- und Denkweise der Schaltungsentwickler so nahe wie möglich kommt (vgl. hierzu Abbildung 3). Je besser dies gelingt, umso eher sind die Designer bereit, mit angebotenen Werkzeugen zu arbeiten und umso umfangreicher wird das im Schaltplangenerator eingebaute Expertenwissen sein. Daneben sollten Möglichkeiten der Automatisierung in Teilschritten soweit als möglich genutzt werden, so dass im Ergebnis ein teilautomatisiertes interaktives Verfahren entsteht, das den Designer in seiner Entwicklungsarbeit unterstützt. Wichtig ist hierbei, den Grad der geplanten Generalisierung nicht zu groß zu wählen, um die Realisierbarkeit des Vorhabens nicht zu gefährden.

Wie bereits erwähnt, ist in Abbildung 3 horizontal der Entwurfsfluss für den Schaltplangenerator (SM-PCell) dargestellt. Der SM-PCell Designer soll in Anlehnung an den von CADENCE® entwickelten PCell Designer eine Designumgebung mit entsprechenden Werkzeugen zur Verfügung stellen, welche den SM-PCell-Entwickler bei der Erstellung der Schaltungsklassenspezifischen SM-PCell unterstützt. Wie in der Darstellung zu erkennen ist, besteht die Designumgebung aus einer grafischen Benutzeroberfläche (GUI) und einer „darunterliegenden“ code-basierten Beschreibungssprache (PCDS) zur Umsetzung der GUI-gestützten Benutzereingaben in den entsprechenden Generator-Code. Grundlage für die Erstellung des Generators bildet die dem Schaltungs-Designer bekannte Schaltplan-Entwurfsumgebung (schematic editor). Die in die GUI des zu entwickelnden SM-PCell Designers eingegebenen Informationen werden im schematic editor als (parametrisierter) Schaltplan wiedergegeben. Die bildliche Darstellung entspricht dem aktuellen Generator-Ergebnis.

Ziel des Konzeptes ist, für jede Schaltungsklasse einen spezifischen Generator bereitzustellen, der auf Basis der Eingangs-Parametersätze (Topologie, Bauteildimensionierung und Constraints) des Design-Assistenten die entsprechende Modul-Instanz (Architecture) ausgibt (vgl. hierzu Abbildung 4). Eine Modul-PCell kann mehrere Schaltungs-Architekturen enthalten, eine Schaltungs-Instanz ist aber nur einer Modul-PCell zugehörig.

Im weiteren Verlauf soll am Beispiel der interaktiven Entwurfsumgebung für den Schaltplangenerator (SM-PCell) eine Funktionalitätsbeschreibung („Use-Model“) vorgestellt werden. Die Entwurfsumgebung für den Design-Assistenten soll eine ähnliche Funktionalität bereitstellen, wird aber an dieser Stelle nicht weiter erläutert. Nach Abbildung 3 unterscheiden wir bei der SM-PCell zwischen den drei Bereichen der Eingangs-Parametersätze Topologie, Bauteildimensionierung und Constraints, welche im Folgenden nun näher beschrieben werden sollen.

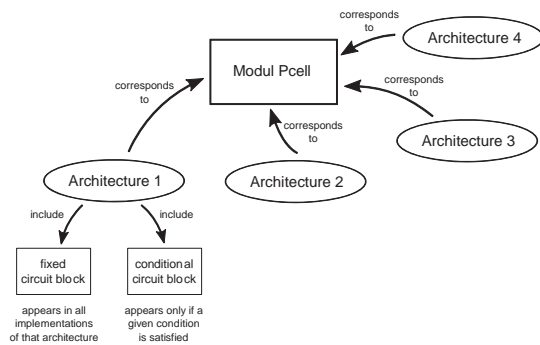


Abbildung 4: Modul-PCell und ihre Instanzen (Architectures).

1) Topologie

Jede Schaltungsklasse kann auf eine Grundtopologie zurückgeführt werden. Anhand dieser Grundstruktur entwickeln wir die Freiheitsgrade der PCell und definieren deren Parametersätze und den gültigen Parameterraum. Die Erweiterungen werden mit Hilfe einfacher if/then/else-Funktionen realisiert.

- Definition der Grundstruktur
- Relative Platzierung der Einzelkomponenten, dadurch keine Überlagerung
- Einfacher Wechsel zwischen nMOS und pMOS-Variante (generische Funktion: Auto-Flipping)
- Simple Erweiterbarkeit der Schaltung um z.B.
 - Kaskode- Stromspiegel
 - ENABLE-Funktion (Abbildung 5)
- Setzen von (Layout-)Constraints wie
 - Maching
 - Stromkritische Pfade
 - Leitungsabstände
- Trennung zwischen (Modul-)Symbol und Schaltplan(-Modul)

Als Hilfsmittel stehen neben einfachen Operationen wie das Platzieren einzelner Bauteile auch ganze Bauteil-Module wie etwa Eingangs-, Ausgangs-, oder Differenzstufen-Module, Stromspiegel-Module, etc. zur Verfügung. Die angegebenen Bauteil-Module können ohne Weiteres erweitert werden, Beispiele hierfür findet man unter anderem in [8]. Um eine umfangreiche Bauteil-Modul-Bibliothek übersichtlich zu halten, kann die Generator-Entwurfsumgebung um eine Bibliotheksverwaltung (Library Manager, Browser) inklusive Filter- und Suchfunktion ergänzt werden. Diese Erweiterbarkeit bietet die Möglichkeit, auch unternehmensspezifisches Knowhow in die Modul-Bibliothek des SM-PCell-Entwicklungstools einfließen zu lassen.

Grundsätzlich gilt, dass jeder Designer seinen eigenen Layout-Stil in Bezug auf die Anordnung der einzelnen Bauteilkomponenten eines Schaltplanes (schematic) entwickelt. Bei einer Automatisierung wird dem Anwender meist ein wenig seines Entscheidungsfreiraumes abgenommen. Aus der digitalen Schaltungssynthese sind zum Teil haarsträubende Schaltbilder (Schaltplan-Abbilder) durch Platzierungsalgorithmen

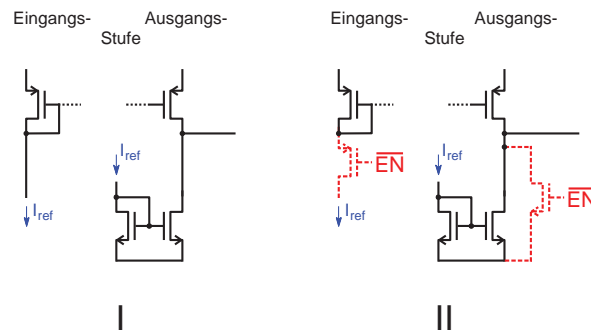


Abbildung 5: ENABLE-Funktionalität.

durchaus bekannt. Da in dem hier beschriebenen Ansatz die Platzierung der Bauelemente nicht mit Hilfe eines Algorithmus erfolgt, sondern von einem erfahrenen Designer vorgegeben wird, ist die Akzeptanz, ein „menschlich“ generiertes Ergebnis anzunehmen, deutlich höher als – im Gegensatz – ein algorithmisch-optimiertes Schaltungs-Design „hinzunehmen“.

2) Dimensionierung

Die Dimensionierung der Bauteile erfolgt ausschließlich über die Eingangsinformationen *Bauteile-Dimensionierung* aus dem semi-automatischen Generator (Design-Assistent) des Anwender-Entwurfsflusses (vgl. hierzu nochmals Abbildung 3). Die SM-PCell enthält lediglich Informationen zum gültigen Parameter-Raum, um eine falsche Dimensionierung mittels einer Fehlermeldung abzufangen, sofern diese außerhalb der zulässigen Grenzen liegt.

3) Constraint-Handling

Notwendig ist der Einbezug von Layout-Parasiten wie z.B. IR-Drop. Kritische Pfade können mit entsprechenden Leitungswiderständen versehen werden, damit die parasitären Einflüsse bei der Simulation berücksichtigt werden. Im Schaltplan selber sollen diese aber nicht sichtbar sein. Weiter wird der entsprechende Widerstandswert als Randbedingung, oder Constraint (Leitungslänge in DU) an das Layout übergeben und dort beim Ziehen der Leiterbahn interaktiv berücksichtigt. Sollte der Leitungswiderstand bei der Layout-Generierung nun auf Grund der Streckenlänge zu hoch sein, so muss entweder die Leitung verbreitert oder die Leitungslänge verkürzt werden.

Eine weitere nützliche Funktion ist eine Verbindung als Strom-kritische Leitung zu kennzeichnen. Somit kann bei der Simulation die Belastung der Leitung automatisch in die entsprechende Constraint (minimale Leitungsdicke in DU bei entsprechender Strombelastung) umgewandelt und der Leitung angeheftet werden. Auch hier kann bei der Layout-Generierung dieses Constraint explizit miteingebunden und eine Leitungsbreite unter dem gegebenen Wert über eine Fehlermel-

dung abgefangen werden. Die Constraints werden automatisch mit einbezogen und direkt an die entsprechende Layout-PCell weitergereicht. Diese berücksichtigt die entsprechenden Constraints letztlich bei der Layout-Erstellung.

Die vollständige Umsetzung des hier beschriebenen SM-PCell-Konzeptes erfolgt erst nach der Erstellung eines vollständigen Use-Model. Dennoch sollen hier nun beispielhaft einige Eigenschaften aufgelistet werden:

- Hoher Freiheitsgrad
- Mehrere Topologien in einer Modul-PCell
- Freie Transistorauswahl (nMOS, pMOS, NPN, PNP)
- Variable Bauteildimensionierung in einem vordefinierten Parameterraum
- Flatten-Funktion zum Ersetzen der PCell-Instanz durch die entsprechende interne Schaltung
- Differenzierung zwischen Symbol und darunterliegender Schaltung
- Technologieunabhängigkeit
- Einbringung von Expertenwissen
 - Implizit durch die Abbildung von Lösungsstrategien in Generatoren
 - Explizit durch Unterstützung formaler Constraints

IV. DESIGN FLOW EINER SM-PCELL

In Abbildung 6 ist schematisch der Entwurfsfluss (Design Flow) einer Schematic-Modul-PCell dargestellt. Wie bereits beschrieben ist die Generator-Umsetzung Schaltungsklassen-spezifisch, daher müssen die im Folgenden beschriebenen Punkte für jede Schaltungsklasse, welche neu als PCell entwickelt werden soll, durchgeführt werden. Somit erfolgt im ersten Schritt zunächst die Festlegung bzw. die Definition der Schaltungsklasse. Anschließend findet die Exploration der Schaltungsklasse statt, bei der alle charakterisierenden Kenngrößen (Performance-Parameter) ermittelt werden und der entsprechende Parameter Raum abgesteckt wird. Es können – praktisch gesehen – nicht alle bestehenden Schaltungstopologien mit Hilfe des hier vorgestellten Generator-Konzeptes abgebildet werden, da der Entwurfsaufwand einfach zu hoch wäre, oder anders gesagt eine Umsetzung aller erdenklichen Schaltungen unmöglich ist. So muss als Grundvoraussetzung eine Auswahl von Topologien erfolgen, welche eine ausreichend hohe Wiederverwendbarkeit (reuse-Faktor) aufweisen [9]. Diese lassen sich dann mittels Kenngrößen charakterisieren (Beispiele solcher Kenngrößen für OTA-Schaltungen finden sich in [5]). Damit sind die einzelnen Schaltungstopologien bekannt und können erstellt werden. Dies führt zum Aufziehen des gültigen Performance-Parameter-Raumes und zur Definition des Device-Parameter-Raums. Bei der anschließenden Exploration der separaten Schaltungstopolo-

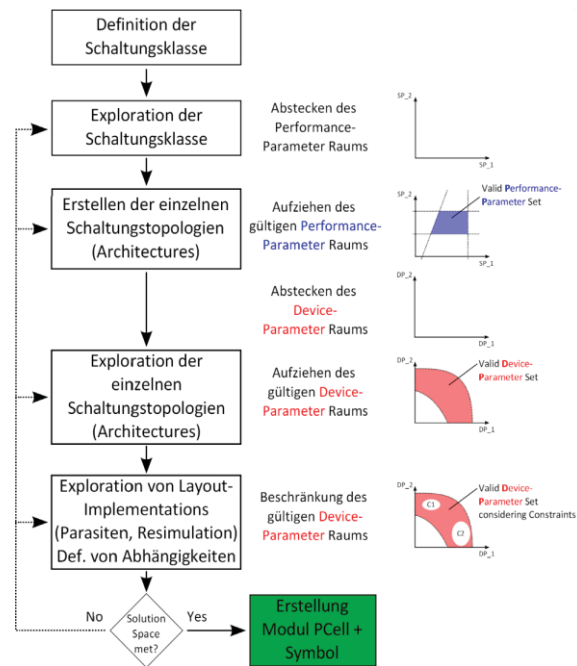


Abbildung 6: Design Flow einer SM-PCell.

gien wird dann der gültige Device-Parameter-Raum bestimmt. Dieser wird im letzten Schritt durch Randbedingungen (Constraints) und die Berücksichtigung von Layout-Parasiten weiter eingeschränkt. Entsprechen die nun gefundenen gültigen Performance- und Device-Parameter-Räume dem geforderten Lösungsraum, werden diese Informationen als Eingangswerte sowohl für den Entwurf des Design-Assistenten (Semi-Automatic interactive Design-Assistant), als auch für die SM-PCell verwendet.

V. BEISPIELE PARAMETRISierter SCHALTUNGEN

Das Konzept der SM-PCell wurde zwar noch nicht vollständig umgesetzt, jedoch bestehen bereits einige grundlegende Ansätze. Erste Entwicklungen, den Modul-Charakter auch auf das Frontend (Schaltplan, schematic) zu übertragen, waren z.B. eine automatische Strukturerkennung und ein parametergesteuerter Topologie-Generator, welcher für die OTA-Schaltungsklasse implementiert wurde [9].

Die automatische Strukturerkennung zielt darauf ab, die Teilschaltungen eines Schaltplanes zu gruppieren und von Device-Ebene auf Modul-Ebene zu bringen, ohne die bereits beschriebenen Module erzeugen zu müssen, sondern diese anhand eines Algorithmus eigen und automatisch erkennen zu lassen. Implementiert wurde hierbei eine Erkennung für Stromspiegel, welche die detektierten Stromspiegel mit einer entsprechenden Layout-PCell eines parametrisierten Stromspiegel-Layouts verknüpft.

VI. FAZIT

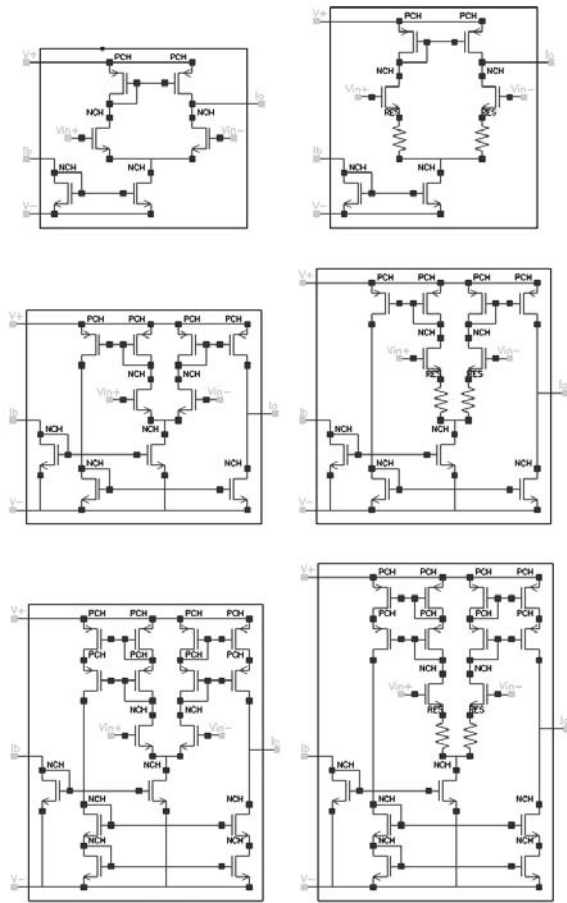


Abbildung 7: Verschiedene Architekturen (Topologien) einer Schematic Modul-PCell.

Abbildung 7 zeigt sechs verschiedene Varianten einer OTA-PCell mit nMOS Eingang, welche mit Hilfe des tools PCDS [11] erstellt wurde. PCDS ist eine Entwurfsumgebung, welche unter Verwendung einer Metasprache Hilfswerkzeuge zur Erstellung eines parametergesteuerten Schaltplan-Generators bietet. Dem Entwickler wird dabei eine grafische Benutzeroberfläche zur Parameterdefinition und Code-Eingabe zur Verfügung gestellt. Implementiert wurde dieses PCell-Beispiel in der von CADENCE® angebotenen Entwurfsumgebung *Design Framework II (DFII)*. Der Fokus wurde hier auf die Topologie-Auswahl gelegt, die Dimensionierung der Schaltung wurde dabei nicht berücksichtigt. Die Modul-PCell erzeugt – parametergesteuert – verschiedene Topologien (Instanzen), die Auswahl erfolgt über das Property-Menü. Die Implementierung der exemplarisch dargestellten PCell erfolgte mit knapp 80 Zeilen PCDS-Code, was übersetzt über 1800 Zeilen SKILL-Code ergibt. Die erhebliche Reduzierung der zu implementierenden Code-Zeilen zur Erstellung der PCell zeigt eindrucksvoll den Nutzen eines höheren Automationsgrades und die Notwendigkeit, den Analog-Flow auf höherer Abstraktionsebene zu realisieren.

In der vorliegenden Arbeit wurde ein Konzept eines durchgängig auf parametrisierten Modulgeneratoren basierenden Entwurfsflusses für Analogschaltungen vorgestellt. Hierbei ist es wichtig, die Sichtweise des PCell-Entwicklers und des PCell-Anwenders zu unterscheiden. Nach dem hier gezeigten Vorgehen sollen dem Entwickler bereits unterstützende, interaktive Entwurfswerkzeuge zur Verfügung stehen, mit Hilfe derer er einen Design-Assistenten und einen Schaltplangenerator für den Anwender implementiert. Design-Assistent und Schaltplangenerator ermöglichen dem Anwender einen Werkzeug-unterstützten Entwurfsfluss mit einfacher Wiederverwendung des durch den Entwickler eingebundenen Expertenwissens. Zwar begründet das Konzept einen einmaligen Mehraufwand bei der Erstellung der Schaltungsklassen-spezifischen Werkzeuge, der Generatorcharakter stellt jedoch bei der Wiederverwendung einen erheblichen Vorteil da. Weitere Vorteile sind:

- Hierarchische Konsistenz, d.h. einheitlicher SDL-Flow, ohne Hierarchiewechsel. Damit wird Informationsverlust vermieden.
- Reduzierter Design Aufwand durch einfaches Instanzieren einer PCell, ohne manuelle Schaltungserzeugung oder fehleranfälliges Copy-und-Paste.
- Einheitlicher „Layout-Stil“ in Bezug auf die Anordnung der Bauteile und deren Konnektivität. Dies erhöht die Lesbarkeit der Schaltungsdesigns und damit die Akzeptanz.
- Keine Abweichung von der bisherigen Entwurfsmethodik durch die Einbindung der tools in einer dem Designer vertrauten Entwurfs-Umgebung. Damit wird ebenfalls die Akzeptanz erhöht.

DANKSAGUNG

Diese Arbeit ist Teil eines Projektes, welches durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert ist.

LITERATURVERZEICHNIS

- [1] G. Gielen, H. Walscharts, and W. Sansen, “Analog circuit design optimization based on symbolic simulation and simulated annealing,” *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 707–713, Jun 1990.
- [2] J. Harvey, M. Elmasry, and B. Leung, “Staic: an interactive framework for synthesizing cmos and bimos analog circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 11, pp. 1402–1417, Nov 1992.
- [3] M. Degrauwe, O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. L. A. G. Goffart, E. Vittoz, S. Cserveny, C. Meixenberger, G. Van Der Stappen, and H. Oguey, “Idac: an interactive design tool for analog cmos circuits,” *IEEE Journal of Solid-State Circuits*, vol. 22, no. 6, pp. 1106–1116, Dec 1987.
- [4] G. G. E. Gielen and R. A. Rutenbar, “Computer-aided design of analog and mixed-signal integrated circuits,” *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1852, Dec. 2000.

- [5] D.D. Gajski, "The Structure of a Silicon Compiler", *Proceedings of IEEE ICCD*, 1987, pp. 272-276.
- [6] A. Gerlach, M. Junge, and J. Scheible, "Universal OTA Test Bench," *Workshop der ASIM/GI-Fachgruppe*, Robert Bosch Center for Powerelectronics, Reutlingen, Germany, Feb. 2014, pp. 83–87.
- [7] J. Crossley, A. Puggelli, H.-P. Le et al., "BAG: A designer-oriented integrated framework for the development of AMS circuit generators", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 74–81, 2013.
- [8] M. Eick, "Structure and signal path analysis for analog and digital circuits," *Ph.D. dissertation*, Technical University of Munich; Department of Electrical Engineering and Information Technology; Institute for Electronic Design Automation, 2013.
- [9] A. Gerlach, M. Junge, J. Scheible, T. Rosahl, "Optimierte, wiederverwendbare OTA-Schaltungen für moderne Power BiCMOS-Technologien", MPC14, 52. Fachtagung der Multi Projekt Chip Gruppe, Künzelsau, Germany, July 2014.
- [10] D. Marolt, J. Scheible, G. Jerke, "The Application of Layout Module Generators upon Circuit Structure Recognition", *CDNLive EMEA 2011*, Munich, Germany, 2011
- [11] D. Marolt, M. Greif, J. Scheible, and G. Jerke, "Peds: A new approach for the development of circuit generators in analog ic design," *22nd Austrian Workshop on Microelectronics (Austrochip)*, Oct 2014, pp. 1–6.



Matthias Greif erhielt den akademischen Grad des B. Eng. in Mechatronik im Jahr 2011 von der Hochschule Reutlingen und den Grad des M. Sc. in Leistungs- und Mikroelektronik im Jahr 2013 von der Hochschule Reutlingen. Er ist wissenschaftlicher Mitarbeiter am Robert Bosch Zentrum für Leistungselektronik (rbz) der Hochschule Reutlingen und befasst sich mit dem Thema Electronic Design Automation (EDA).



Daniel Marolt erhielt den akademischen Grad des B. Eng. in Mechatronik im Jahr 2008 von der Hochschule Reutlingen und den Grad des M. Sc. in Mechatronik im Jahr 2009 von der Hochschule Reutlingen. Er ist wissenschaftlicher Mitarbeiter am Robert Bosch Zentrum für Leistungselektronik (rbz) der Hochschule Reutlingen und befasst sich mit dem Thema Electronic Design Automation (EDA).



Jürgen Scheible erhielt den akademischen Grad des Dipl.-Ing. in Elektrotechnik im Jahr 1987 von der Universität (TH) Karlsruhe und den Grad des Dr.-Ing. in Elektrotechnik im Jahr 1991 von der Universität (TH) Karlsruhe. Er ist Professor für Electronic Design Automation (EDA) am Robert Bosch Zentrum für Leistungselektronik (rbz) an der Hochschule Reutlingen.