Siegfried Puga-Reichle

Ansätze einer High-Level-Synthese in der Electronic Design Automation

Diplomarbeit

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de/ abrufbar.

ISBN: 9783656998938

Siegfried Puga-Reichle

Ansätze einer High-Level-Synthese in der Electronic Design Automation

Examicus Verlag

Examicus - Verlag für akademische Texte

Der Examicus Verlag mit Sitz in München hat sich auf die Veröffentlichung akademischer Texte spezialisiert.

Die Verlagswebseite www.examicus.de ist für Studenten, Hochschullehrer und andere Akademiker die ideale Plattform, ihre Fachtexte, Studienarbeiten, Abschlussarbeiten oder Dissertationen einem breiten Publikum zu präsentieren.

Ansätze einer High-Level-Synthese in der Electronic Design Automation

Diplomarbeit

Zur Erlangung des akademischen Grades

Dipl.-Ing. (FH)

Vorgelegt an der
Fachhochschule Konstanz
Fachbereich Elektrotechnik und Informationstechnik

Diplomand:
Siegfried Puga-Reichle

Fachhochschule Konstanz
FB: Elektrotechnik und Informationstechnik

Erstellt in der Zeit
Von 01.03.05 bis 24.10.2005

Printed in Germany

Inhaltsverzeichnis

1 Einleitung / Überblick

Die Gesellschaft wandelt sich immer mehr zu einer Informations- und Kommunikationsgesellschaft. Die Schlüsseltechnologie in dieser Entwicklung stellt die Mikroelektronik dar. Die Mikroelektronik ist heute allgegenwärtig und aus unserer Gesellschaft nicht mehr weg zu denken und sie gewinnt immer noch mehr an Bedeutung in allen Lebenslagen.
Im Jahre 2010 werden über 5 Milliarden Transistoren auf einem einzigen Chip integrierbar sein und die Entwicklungszyklen werden aus Wettbewerbsgründen immer kürzer. Das Entwurfsteam muss trotz der Komplexitätsexplosion dem Kosten- und Zeitdruck entgegenwirken. Aufgrund dessen muss sich die Entwurfsproduktivität in jedem Jahr mehr als verdoppeln, will sie der Chipentwicklung folgen.

Der ungebrochene Technologiefortschritt hat dazu geführt, dass heute ganze Systeme aus mehreren Prozessoren und komplexen Verbindungsstrukturen auf einem einzelnen Chip gefertigt werden können (*SoC*). Um die Komplexität dieser Systeme und mögliche Anwendungen kontrollieren zu können, bedarf es einer Automatisierung des Entwurfs auch auf höheren Entwurfsebenen (High-Level-Synthese). Die Automatisierung des Entwurfs (Electronic Design Automation, EDA) stellt deshalb den Schlüssel zur Mikroelektronik und damit zu den Systemen der Zukunft dar.

Heutiger Standard des Schaltungsentwurfs ist die Hardwarebeschreibung durch Hardwarebeschreibungssprachen (*HDL*), die durch *CAE*-Werkzeuge (Computer Aided Engineering) zur Schaltungssimulation und –synthese benützt werden. Dabei dient die *Simulation* der Überprüfung der Funktion des Entwurfs und die *Synthese* der Umsetzung der Beschreibung in eine Netzliste für die Implementierung der Schaltung auf die gewählte Zieltechnologie wie ASICs oder FPGAs. Als Hardwarebeschreibungssprachen haben sich weltweit die beiden Sprachen Verilog und VHDL etabliert.

Die *Simulation* und *Verifikation* gewinnt zunehmend immer mehr an Bedeutung, je komplexer die Schaltungen werden. Es ist nicht mehr möglich Signale Takt für Takt auf ihre Richtigkeit zu überprüfen, sondern es müssen neue Verifikationsstrategien gefunden werden. Einer davon ist die Entwicklung von *HDVL-* Sprachen (Hardware Description and Verification Language).

Diese Arbeit beschäftigt sich daher mit den bestehenden Problemen im Systementwurf und behandelt neue Sprachen und Werkzeuge die eine High-Level-Synthese ermöglichen.
Im ersten Teil sollen die Problematik näher dargestellt werden und einen kleinen Background geschaffen werden. Nachfolgend sollen neue *HDL*-Sprachen vorgestellt werden, bevor ein Überblick über High-Level-Synthese-Tools gegeben werden soll.
Eine Sammlung derzeitiger Simulatoren und Back-End-Werkzeuge sollen die Tools abschließend ergänzen. Da die Herausforderung der Entwurfskomplexität nicht von einem einzelnen Unternehmen beherrschbar ist, wird auf weiterführende EDA-Organisationen und Quellen eingegangen.
Abschließend soll anhand des DSP-Builders von Altera untersucht werden wie eine High-Level-Synthese aus Matlab/Simulink nach VHDL funktioniert und wie Effizient sie arbeitet.

2 Hintergrund der Hardwarebeschreibungssprachen

Seit vielen Jahren besteht eine große Dynamik im Entwurf von Digitalschaltungen und die Entwurfsverfahren wandeln sich ständig. Früher wurden integrierte Schaltungen mit graphischen Rechner-Werkzeugen in Form von Schaltplänen manuell erstellt. Dazu mussten die Grundelemente der Schaltung - logische Gatter, die in einer Bibliothek zur Verfügung stehen - bzw. deren Symbole ausgewählt, auf einem Schaltplan platziert und anschließend miteinander verbunden werden. Auf diese Weise konnten zunächst einfache Module generiert und mit diesen wiederum komplexere Schaltungen zusammengesetzt werden, was einem sog. Bottom-Up Entwurf entspricht. Dieser Prozess konnte bei umfangreichen Schaltungen sehr viel Zeit in Anspruch nehmen. Darüber hinaus war ein derartiger Schaltungsentwurf wenig flexibel gegenüber Änderungen, die wiederum ein mühsames Umzeichnen der Pläne erforderlich machten.

2.1 Motivation: IC-Entwurfsverfahren

Die heutige Situation beim Entwurf elektronischer Systeme erfordert jedoch den Umgang mit immer komplexeren Aufgabenstellungen, deren Integration durch die gestiegenen Packungsdichten ermöglicht wird. Um dem wachsenden Konkurrenzdruck und den Anforderungen der Kunden genügen zu können, ist zusätzlich eine kurze Entwicklungszeit ein entscheidender Faktor. Auch wird versucht, einmal generierte Funktionsblöcke bzw. Module in neuen Systemen wieder zu verwenden, was eine technologieunabhängige Beschreibung erforderlich macht.

Durch diese Randbedingungen wird heute die Entwicklung von Digitalschaltungen streng methodisch als Top - Down Entwurf vorgenommen. Zum Einsatz kommen hierbei Hardware-Beschreibungssprachen (*HDL* - Sprachen), die eine Modellierung des Systems auf vielen Abstraktionsebenen ermöglichen. Bedingt durch die beim Top - Down Entwurf ständig durchgeführte, schrittweise Verfeinerung müssen, angefangen von der Systemspezifikation über eine Beschreibung der Algorithmen und deren Aufteilung in Funktionsblöcke bis hin zu einer Gatternetzliste, alle Ebenen mit einer solchen Sprache realisiert werden können. Wichtiger Punkt im heutigen Entwurfsprozess ist die Unterstützung durch Synthesewerkzeuge, die auf der Grundlage einer Verhaltensbeschreibung eine Gatternetzliste automatisch erstellen. Dies setzt jedoch eine normierte Beschreibungssprache voraus. Des Weiteren soll ein damit modelliertes System in verschiedenen Abstraktionsebenen simulierbar sein um schon in einem frühen Stadium der Entwicklung Systemfehler zu entdecken.

Es gibt viele verschiedene Ansätze all diese Forderungen in einer Sprache zu vereinen. Eine davon ist die Hardwarebeschreibungssprache VHDL. VHDL (**V**HSIC **H**ardware **D**escription **L**anguage; VHSIC (**V**ery **H**igh **S**peed **I**ntegrated **C**ircuit)) erfüllt all diese Anforderungen. Es verfügt über Konstrukte zur Beschreibung von nebenläufigem oder sequentiellem Verhalten von Digitalschaltungen mit oder ohne Timing auf einer Vielzahl von Abstraktionsebenen. Auch besteht die Möglichkeit, durch Verschalten von Submodulen einen hierarchischen Entwurf aufzubauen. Durch die Normierung als *IEEE* Standard unterstützen alle wichtigen Entwurfswerkzeuge diese Sprache weshalb sie sich auch gut als Format zum Datenaustausch eignet. [1]

2.2 Bestehende Probleme des heutigen Systementwurfs

Elektronikkomponenten und die daraus aufgebauten informationsverarbeitenden Komponenten sehen sich ständig wachsenden Anforderungen entgegen. Gleichermaßen bezieht sich dies auch auf die Leistungsfähigkeit ihrer Prozessschnittstellen, die Komplexität und Verarbeitungsgeschwindigkeit der realisierten Steuer-, Regler- und Signalverarbeitungsfunktionen sowie die Flexibilität einer integrierten Netz- oder Systemschnittstelle. Ursachen liegen darin, dass bei vielen technisch hochwertigen Produkten eine Verdrängung mechanischer, elektrischer oder fluidischer Lösungen durch elektronische und informationsverarbeitende Systeme erfolgt. Umgekehrt bedeutet dies, dass viele mechanische, elektrische oder fluidische Funktionen, die von ihren physikalischen Phänomenen meistens als zeit- und amplitudenkontinuierliche Größen zu beschreiben sind, in informationsverarbeitende Komponenten möglichst wirklichkeitsgetreu abgebildet werden müssen.[4]
Rapide zunehmende Komplexität, Funktionsvielfalt und Leistungsanforderungen von elektronischen Systemen finden sich nahezu in jedem Bereich. Nachfolgend soll ein kleiner Überblick über die gestiegene Komplexität von Systemen unterschiedlicher Bereiche aufgezeigt werden.

Bereich	**Anforderungs-spektrum**
Automobilelektronik: (Abb. 2-1)	• Echtzeit, • Ausfallsicherheit, • Flexibilität, • Nachrüstbarkeit, • Hohe Performanz, • Niedrige Kosten, …

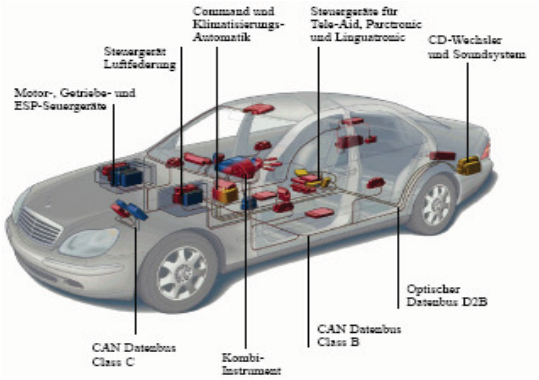

Abb. 2-1 30% der Wertschöpfung und 90% der Innovationen resultieren aus Elektronik und Software. Ca. 70 teilweise vernetzte Steuergeräte pro high-end Fahrzeug

Mobilkommunikation: (Abb. 2-2)	• Funktionsvielfalt • (PC-) Rechenleistung • GByte Speicherkapazität • Hohe Integrationsdichte • Geringe Verlustleistung

Abb. 2-2 Mobiltelefon, Megapixel-Kamera, www-Browser, MP3-Player, PDA, Diktiergerät, Computerkonsole, ...

Unterhaltungs-elektronik:
(Abb. 2-3)

- (Supercomputer-) Rechenleistung
- Vernetzung,
- „Plug & Play“
- Miniaturisierung,
- geringe Verlustleistung und Kosten

Abb. 2-3 Unterhaltungselektronik wird immer kleiner, Leistungsfähiger und billiger

Supercomputer on a Chip:
(Abb. 2-4)

- 2x64-bit *PPUs*
- 8x SPEs mit 32 Gops
- 4.6 GHz Takt, 234 M Transistoren,90 nm SOI *CMOS*
- 76 GBytes/s I/O
- 25 GBytes/s Speicherbandbreite

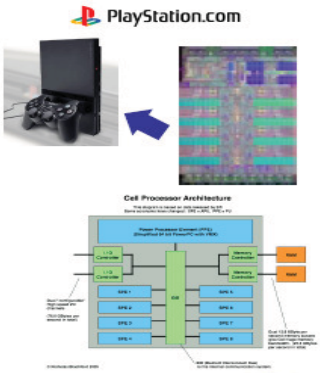

Abb. 2-4 Leistungsfähige Systeme werden immer kleiner

Industriesteuerungen:
(Abb. 2-5)

- hohe Ausfallsicherheit
- Echtzeitfähigkeit
- extreme Umweltbedingungen und mechanische Belastungen

Abb. 2-5 Anforderungen an die Elektronik steigen ständig an

Vernetzte Welten:
(Abb. 2-6)

- Breitband Netzzugangsdienste x-DSL, Cable Modem
- Internet: Terabit *IP* Router mit Dienstgüten und Datensicherheit
- „Jeder mit jedem“

Abb. 2-6 Die Informationstechnologie ist aus unserer Welt nicht mehr weg zu denken

Durch die zunehmende Komplexität und die Zahl der Aufgaben eines elektronischen Systems werden jedoch auch die Entwurfsaufgaben zunehmend anspruchsvoller. Zusätzlich wird eine sehr geringe Zeit vom Beginn der Entwicklung bis zur Marktreife (engl. *time-to-market*) gefordert, die ohne aufwendige nachträgliche Korrekturen am Endprodukt in ihrer Einsatzumgebung auskommen müssen. Die schnell wachsende Komplexität der Entwurfsaufgaben ist durch den hohen Anteil an Wechselwirkungen für den Menschen nur noch schwer nachzuvollziehen. Auch die Qualitätssicherung gestaltet sich als zunehmendes Problem. Um Entwurfsfehler in frühen Phasen zu vermeiden, werden verstärkt formale Hilfsmittel zur Spezifikation und zur abstrakten Modellierung verwendet. Mit den formalen Hilfsmitteln finden auch die *Simulation* und formale *Verifikation* der Modellierung Einzug in den Entwurf elektronischer Systeme. Diese Techniken sind in vielen Jahren gewachsen.
Deswegen existieren mehrere parallele Beschreibungsformen, mit denen dieselbe Funktionalität modelliert werden kann. Zusätzlich sind die Beschreibungsformen sowie *Simulation* und formale *Verifikation* oftmals auf einzelne Werkzeuge festgelegt, die keine oder eine eingeschränkte Interaktion mit weiteren Werkzeugen erlauben.

Diese parallele *Simulation* und *Verifikation* im Systementwurf stellt ein Problem dar, da keine Durchgängigkeit der einzelnen Ebenen gegeben ist. In jeder Systemebene kommen so beispielsweise unterschiedliche Beschreibungssprachen vor, deren Interaktion mittels weiterer Werkzeuge hergestellt werden kann, vgl. Abb. 2-7. Diese Problematik verkompliziert den gesamten Systementwurf unnötig. Ein neuer Ansatz, unter vielen, versucht mittels der *HDL*-Sprache SystemC eine Durchgängigkeit des gesamten Systementwurfs zu ermöglichen. Die verschiedenen Ansätze der Durchgängigkeit werden in Kapitel 4 Systembeschreibungssprachen (System-Level-Entwurf näher erläutert.

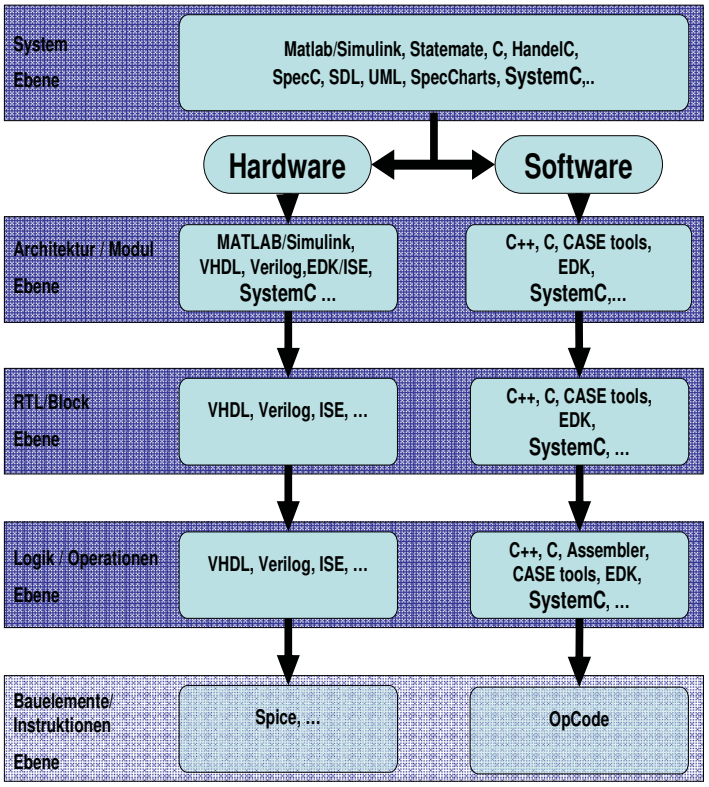

Abb. 2-7 Durchgängigkeitsproblem im (HW/SW) Systementwurf.

2.3 Entwurfssichten

Die Entwicklung elektronischer Systeme ist bei der heutigen Komplexität und den genannten Anforderungen nur durch eine strukturierte Vorgehensweise beherrschbar. Idealerweise wird, ausgehend von einer Spezifikation auf Systemebene, die Schaltungsfunktion partitionieren und die grundsätzlichen Funktionen den einzelnen Modulen zugeordnet. Schrittweise wird der Entwurf weiter strukturiert und zunehmend mit Details der Implementierung versehen, bis die für die Fertigung des elektronischen Systems notwendigen Daten vorliegen. Dies können Programmierdaten für Logikbausteine, Layouts für Leiterplatten oder Datensätze für die IC-Fertigung sein. Im Entwurf elektronischer Systeme wird üblicherweise in die drei Sichtweisen Verhalten, Struktur und Geometrie unterschieden. Diese Einteilung wird durch das sog. Y-Diagramm verdeutlicht (Abb. 2-8):

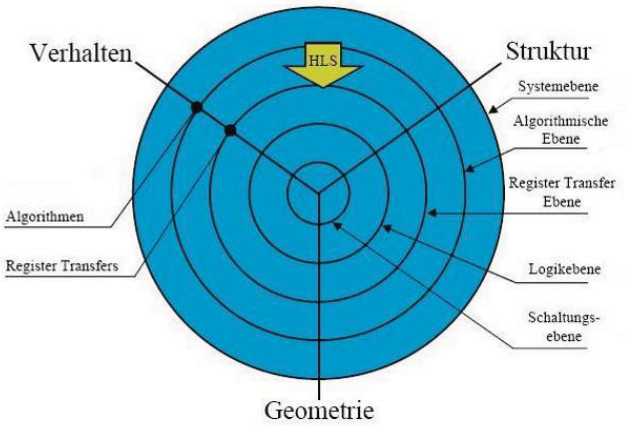

Abb. 2-8 Y-Diagramm nach Gajski-Walker und die einzelnen Abstraktionsebenen

Gleichzeitig zu den drei Sichtweisen, die durch die Äste im Y-Diagramm repräsentiert werden, sind auch die verschiedenen Abstraktionsebenen durch Kreise mit unterschiedlichen Radien dargestellt. Ein großer Radius bedeutet hohe Abstraktion. Es kann nun vereinfacht der Entwurf elektronischer Systeme als eine Reihe von Transformationen (Wechsel der Sichtweise auf einem Abstraktionskreis) und Verfeinerungen (Wechsel der Abstraktionsebene innerhalb einer Sichtweise) im Y-Diagramm dargestellt werden. Beginnend auf dem Verhaltensast in der Systemebene wird der Entwurf durch Verfeinerungs- und Syntheseschritte bis hin zum Layout auf dem geometrischen Ast durchgeführt. Ein "Place- and Route" - Werkzeug für Standardzellenentwürfe überführt beispielsweise eine strukturale Beschreibung in der Logikebene (Gatternetzliste) in eine geometrische Beschreibung in der Schaltkreisebene (IC-Layout). Das reine Top-Down- Vorgehen (Entwicklung in Richtung Kreismittelpunkt) kann dabei nicht immer konsequent beibehalten werden. Verifikationsschritte zwischen den einzelnen Ebenen zeigen Fehler beim Entwurf auf. Gegebenenfalls muss das jeweilige Entwurfsergebnis modifiziert, der Entwurfsschritt wiederholt oder sogar auf höherer Abstraktionsebene neu eingesetzt werden. Dies wird auch als "Jojo-Design" bezeichnet. [10]