

Verringerung des Parameterraums von IP unter Verwendung von Parameterabhängigkeiten

V. Jerinić and D. Müller

Technische Universität Chemnitz, Fakultät für Elektrotechnik und Informationstechnik, Reichenhainer Straße 70, 09126 Chemnitz, Germany

Zusammenfassung. Um eine Wiederverwendung in verschiedenen Anwendungen zu ermöglichen, verfügen Intellectual Properties (IP) häufig über einen umfangreichen Parametersatz. Auf der einen Seite erlaubt die extensive Verwendung von Parametern die Anpassung von IP an verschiedene Spezifikationen. Andererseits vergrößert jeder Parameter den Parameterraum, wodurch es praktisch unmöglich wird, die korrekte Funktion der IP für alle Parameterkombinationen zu verifizieren. Es wird eine Methode vorgestellt, die, basierend auf vorgegebenen Abhängigkeiten der Parameter untereinander, den Parameterraum in orthogonale Subräume, sogenannte Parameter-Domänen, unterteilt, wobei ungültige Parameterkombinationen entfernt werden. Durch diesen Schritt kann der Parameterraum zum Teil erheblich verkleinert werden. Durch die Verwendung von orthogonalen Subräumen werden Verifikationsumgebungen, wie z.B. *Specman EliteTM* von *Verisity*, die Stimuli und Parameterzuweisungen pseudozufällig erzeugen können, wirkungsvoll unterstützt, indem mehrfache Simulationen gleichwertiger Zuweisungen verhindert werden.

In order to be reused in different applications IP are usually parameterized. While a rich parameter set enables users to customize IP to their needs, verification complexity is increased by enlarging the parameter space with every additional parameter. In this work, a graph-based approach to splitting the parameter space into orthogonal subspaces on the basis of defined parameter interdependences is proposed. By utilizing so-called parameter domain graphs, invalid parameter configurations are removed from the parameter space. Verification environments with the capability to automatically generate pseudo-random parameter combinations, e.g. *Specman EliteTM* from *Verisity*, may create parameter combinations which are virtually equal. This can be avoided by using orthogonal subspaces.

1 Einleitung

Um eine Wiederverwendung in vielen verschiedenen Anwendungen zu ermöglichen, verfügen Intellectual Properties (IP) häufig über einen umfangreichen Parametersatz. Auf der einen Seite erlaubt die extensive Verwendung von Parametern die Anpassung von IP an verschiedene Spezifikationen. Andererseits vergrößert jeder Parameter den Parameterraum, wodurch es praktisch unmöglich wird, die korrekte Funktion der IP für alle Parameterkombinationen zu verifizieren. Die hier beschriebene Methode verwendet Abhängigkeiten von IP-Parametern untereinander, um die gültigen Konfigurationen zu bestimmen und alle anderen aus dem Parameterraum entfernen und damit diesen und den resultierenden Verifikationsaufwand zu minimieren.

2 Grundlagen

Viele Arbeiten auf dem Gebiet des Systementwurfs suchen nach optimalen Parameterkonfigurationen für parametrisierte *System-on-Chip (SoC)* in Abhängigkeit von anwendungsspezifischen Vorgaben für Chipfläche, Leistungsfähigkeit oder Energieverbrauch. Es existieren verschiedene Ansätze zur Ermittlung dieser Systemparameter auf verschiedenen Abstraktionsebenen, beispielsweise auf der Register-Transfer-Ebene (Raghunathan et al., 1996), auf der algorithmischen (Macii et al., 1997) oder Systemebene (Givargis et al., 2001a) oder unter Anwendung der Informationstheorie (Marculescu et al., 1996). In Verbindung mit diesen versuchen sogenannte erschöpfende Verfahren, möglichst viele verschiedene Konfigurationen zu simulieren und die resultierenden Systemparameter zu bestimmen.

Der Parameterraum von IP ist oft sehr groß, was diese Lösungsansätze auf Grund begrenzter zeitlicher, personeller und rechentechnischer Ressourcen unpraktikabel macht. Deshalb wurden Verfahren entwickelt, die neben den eigentlichen Parameterdefinitionen auch deren Abhängigkeiten untereinander in Betracht ziehen (Givargis and Vahid, 2000).

Darauf bauen verschiedene Methoden zur Bestimmung von Energie- und Leistungscharakteristika in Abhängigkeit der verwendeten Konfiguration anhand von Systemmodellen wie beispielsweise Givargis et al. (2001b) auf.

Alle diese Verfahren haben gemein, daß sie SoC-Parameter als eine Möglichkeit zur Anpassung von Leistung, Energie- und Flächenbedarf usw. einer Applikation verstehen, ohne dabei die grundlegende Funktionalität des Systems zu ändern. Im Gegensatz dazu, werden IP-Parameter dazu verwendet, IP für verschiedene Aufgaben zu konfigurieren, d.h. ihre Funktionalität je nach Applikation zu ändern. Dennoch können Teile der Theorie aus Givargis and Vahid (2000) auch auf IP-Parameter angewendet werden, weshalb eine ähnliche Terminologie verwendet werden soll.

3 Parameterraum

Auf Register-Transfer-Ebene werden zwei verschiedene Klassen von IP-Parametern unterschieden (Givargis and Vahid, 2000):

- *Statische* Parameter beeinflussen die Struktur des Designs und müssen deshalb vor der Synthese, z.B. auf der Register-Transfer-Ebene, zugewiesen werden. Es sind nur diskrete Zahlenwerte zulässig, da nur diese von den Synthesewerkzeugen unterstützt werden (Synopsys-2001.08).
- *Dynamische* Parameter werden in Konfigurationsregistern gespeichert oder verwenden Steuersignale um das Verhalten einer Schaltung zu ändern. Sie können deshalb auch auf niedrigeren Abstraktionsebenen bis hinab zur Implementierung zugewiesen werden.

In dieser Arbeit soll der Parameterraum als die Menge aller statischen und dynamischen Parameterzuweisungen verstanden werden (Givargis and Vahid, 2000). Ein Parameterraum mit den Parametern $p_1 \dots p_n$ kann mit

$$P = p_1 \times p_2 \times \dots \times p_n \quad (1)$$

beschrieben werden. Für die Parameter-Wertebereiche $R_1 \dots R_n$ ergibt sich seine Größe zu

$$|P| = \prod_{i=1}^n |R_i|, \quad (2)$$

wobei die Größe $|R|$ eines Wertebereichs als Anzahl der möglichen Zuweisungen bzw. Kardinalität der Menge R definiert ist. IP, die als synthetisierbare RTL-Beschreibung geliefert werden, führen zu unterschiedlichen Implementierungen bzw. Instanzen, je nachdem, welche Werte für die statischen Parameter gewählt werden. In diesem Fall muss der gesamte Parameterraum in die Betrachtungen einbezogen werden. Während der Verifikation von IP, die als Gatternetzliste oder Layout geliefert werden, können die statischen Parameter vernachlässigt werden, da diese bereits früher im Entwurfsprozess, i.allg. vor der Synthese, zugewiesen worden sind.

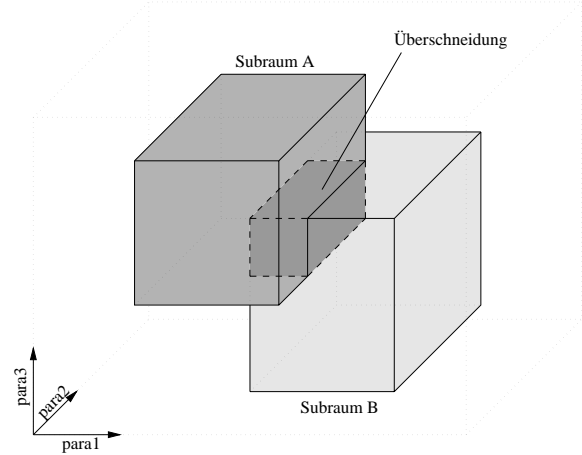


Abbildung 1. Subräume mit gemeinsamen Elementen.

Verschiedene Methoden verwenden eine Kombination aus zufällig erzeugten Parameterkonfigurationen und vorgegebenen Spezialfällen, sogenannten *corner cases*, um eine akzeptable Abdeckung des Parameterraums mit der gewünschten Verteilung zu erzielen (Verisity-2002). Die zufällige Erzeugung kann zum einen zu unzulässigen Konfigurationen führen. Zum anderen können Konfigurationen gewählt werden, die trotz verschiedener Parameterwerte ein und dasselbe Verhalten abbilden, z.B. verschiedene Positionen des Parity-bits in einem seriellen Datenstrom bei abgeschalteter Parity-erzeugung.

Der Parameterraum kann in Subräume unterteilt werden, indem einige der Parameter auf feste Werte bzw. Wertebereiche beschränkt werden, während andere jeden Wert im Rahmen ihres gültigen Wertebereichs annehmen dürfen. Werden Parameterkombinationen zufällig gewählt, kann das zur Verwendung von sich überschneidenden Subräumen für aufeinanderfolgende Simulationen führen (Abb. 1).

Die Auswahl unzulässiger Konfigurationen und die Verwendung sich überschneidender Subräume können vermieden werden, wenn zwei Voraussetzungen erfüllt werden:

- unzulässige Konfigurationen werden vollständig entfernt und
- es werden orthogonale Subräume verwendet.

4 Parameter-Domänen

Beide Voraussetzungen können durch Verwendung von Subräumen mit speziellen Eigenschaften, sogenannten Parameter-Domänen, erfüllt werden:

Definition: Eine Kombination von zulässigen Parameterzuweisungen, die einen Wertebereich oder Wert für jeden verfügbaren Parameter eindeutig beschreibt und eindeutig im gesamten Parameterraum ist, wird als Parameter-Domäne bezeichnet.

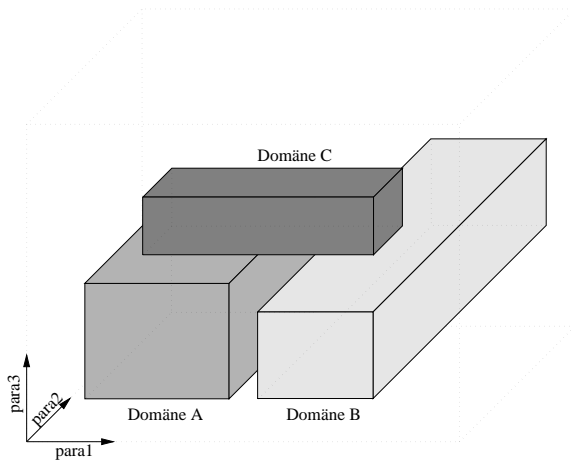


Abbildung 2. Parameter-Domänen

Wird die Auswahl von Parameterkonfigurationen auf Parameter-Domänen beschränkt, ist ihre Gültigkeit garantiert und die Verwendung von Subräumen mit gemeinsamen Elementen wird vermieden (Abb. 2).

Da keine gültige Konfiguration existiert, die nicht in einer der Domänen enthalten ist, wird der Parameterraum auf die Menge aller Parameterdomänen reduziert und damit i.allg. verringert. Sind alle IP-Parameter vollständig voneinander unabhängig, kann nur eine Parameter-Domäne, die identisch zum kompletten Parameterraum ist, gebildet werden. In diesem Fall kann die Größe des Parameterraums nicht durch Verwendung von Parameter-Domänen reduziert werden.

5 Abhängigkeitsmodell

Wie in Givargis and Vahid (2000) erläutert, wird ein Modell zur formalen Erfassung und Abbildung der Abhängigkeiten der Parameter untereinander benötigt. In dieser Arbeit soll der sogenannte *Parameter-Domänen-Graph (PDG)* verwendet werden, bei dem es sich im Grunde um eine spezielles Multi-valued Decision Diagram (Kam and Brayton, 1990, MDD) handelt. PDG verwenden Bäume mit Knoten und Kanten zur Abbildung von Parameter-Domänen. Parameter werden durch Knoten, die den Namen des Parameters tragen, repräsentiert. Jede Kante, die einen Knoten "verläßt", stellt eine zulässige Zuweisung in Form eines Einzelwertes oder Wertebereichs zum jeweiligen Parameters dar. Das Ende einer jeden Kante ist mit einem Knoten für einen weiteren Parameter verbunden, dessen Zuweisung nicht mit dem durch die übergeordnete Kante eingestellten Wert oder Wertebereich kollidiert (Abb. 3).

Eine Parameter-Domäne wird durch eine Anzahl von Blättern und alle Parameterzuweisungen auf dem Weg von dieser zur Wurzel des Baums repräsentiert.

Parameter-Domänen werden auf Basis von Parameter-abhängigkeiten gebildet. Ein Parameter wird als *abhängig* bezeichnet, wenn eine Änderung seines Wertes nur für den Fall, daß ein anderer Parameter einen bestimmten Wert trägt,

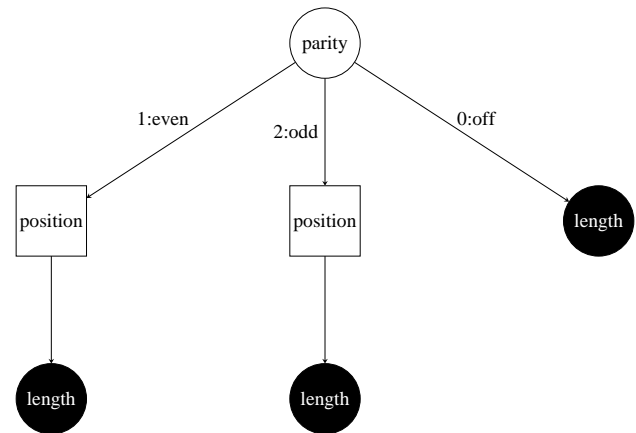


Abbildung 3. PDG mit 3 Parametern

Einfluß auf die Struktur bzw. das Verhalten der IP hat. Diese Definition weicht von jener in Givargis and Vahid (2000) ab, bei der auch Relationen wie $para1 > para2$ oder $para1 + para2 = 32$ berücksichtigt werden. Da sich diese Art von Bedingungen nicht eignet, um den Parameterraum in eindeutige Subräume zu unterteilen, wird sie während der Konstruktion der Parameter-Domänen vernachlässigt.

Gegeben sei eine IP zur seriellen Kommunikation mit den folgenden Parametern:

- $parity[off,even,odd]$ als Typ der Parität,
- $length[1...10]$ als Länge des Datenworts und
- $position[1...10]$ als Position des Paritätsbits innerhalb des Datenworts.

Die Anzahl möglicher Konfigurationen und damit die Größe des Parameterraums ergibt sich nach Gleichung 2 zu

$$\begin{aligned} |P| &= |R_{parity}| \cdot |R_{len}| \cdot |R_{pos}| \\ &= 3 \cdot 10 \cdot 10 \\ &= 300 \end{aligned} \quad (3)$$

Im Falle einer Zuweisung $parity \Rightarrow off$ hat Parameter *position* keinerlei Einfluß auf das Verhalten, d.h. Parameter *position* ist von der Einstellung des Parameters *parity* abhängig. Die vorliegende Art von Abhängigkeit stellt einen Konflikt dar, da das Abschalten der Paritätserzeugung die Wirkung von *position* eliminiert. Grundsätzlich sind zwei Arten von Abhängigkeiten möglich:

- Parameter x hat keinen Einfluß auf das Verhalten, wenn Parameter y einen bestimmten Wert trägt, d.h. es besteht ein *Konflikt* zwischen Parameter x und einer Einstellung von Parameter y ,
- Parameter x hat nur für den Fall, daß Parameter y einen bestimmten Wert trägt, Einfluß auf das Verhalten, d.h. eine Einstellung von Parameter y ist *Voraussetzung* für die Wirkung von Parameter x .

Abbildung 3 zeigt den PDG für die Beispiel-IP. Da der Parameter *position* nur dann Einfluß auf das Verhalten hat, wenn Parameter *parity* entweder den Wert *even* oder *odd* trägt, ist er nur in den entsprechenden Domänen gültig und damit im Teilbaum unterhalb der Wertzuweisung $parity \Rightarrow off$ nicht vorhanden. Parameter *length* hingegen hat keinerlei Abhängigkeiten und ist deshalb in allen Domänen bzw. Teilbäumen vertreten. Auf diese Weise werden drei Domänen gebildet:

- $parity[even], position[1...10], length[1...10]$,
- $parity[odd], position[1...10], length[1...10]$,
- $parity[off], length[1...10]$.

Das Einsparungspotential durch Verwendung von Parameterdomänen ist stark von der Anzahl der IP-Parameter und der definierten Konflikte und Voraussetzungen abhängig. Für das angegebene Beispiel reduziert sich die Größe des Parameterraums als Menge aller in Parameter-Domänen enthaltender Konfigurationen zu

$$\begin{aligned}
 |P| &= |D_{parity \Rightarrow even}| + |D_{parity \Rightarrow odd}| + |D_{parity \Rightarrow off}| \\
 &= 1 \cdot |R_{pos}| \cdot |R_{len}| + 1 \cdot |R_{pos}| \cdot |R_{len}| + |R_{len}| \\
 &= 1 \cdot 10 \cdot 10 + 1 \cdot 10 \cdot 10 + 1 \cdot 10 \\
 &= 210,
 \end{aligned}$$

d.h. der Parameterraum wird um 30% verringert (vgl. Gleichung 3). Für den Fall, daß diese Einsparung nicht dazu verwendet wird, um den Verifikationsaufwand zu senken, wird sie sich in einer besseren funktionalen Abdeckung des Parameterraums niederschlagen.

6 Zusammenfassung und Ausblick

In dieser Arbeit wird eine Methode zur Verringerung des Parameterraums von IP unter Verwendung von Abhängigkeiten der Parameter untereinander vorgestellt. Diese unterteilt den Parameterraum orthogonale Subräume und stellt diese mit Hilfe von sogenannten Parameter-Domänen-Graphen dar, wobei seine Größe z.T. erheblich reduziert werden kann.

Im weiteren Verlauf der Arbeiten soll die Implementierung der Methodik als Softwarepaket fertiggestellt werden, welches PDG auf Basis vorgegebener Parameterdefinitionen

generiert. Besonderer Wert soll dabei auf die automatische Erstellung von HDL-Komponenten gelegt werden, die innerhalb von Testbenches zur Implementierung einer Parameterprüfung verwendet werden können.

Desweiteren soll das Werkzeug den Export von Parameter-Domänen in Verifikationsumgebungen, beispielsweise *Specman EliteTM*, ermöglichen, was IP-Integratoren in die Lage versetzt, die Gültigkeit der verwendeten Konfigurationen nicht nur zu prüfen, sondern die pseudo-zufällige Generierung von vornherein auf diese zu begrenzen.

Acknowledgement. Diese Arbeit wird unter dem Förderkennzeichen <01M 0348 A> "Intellectual Property Qualifikation für effizientes Systemdesign (IPQ)" vom Bundesministerium für Bildung und Forschung unter dem Schwerpunkt "Entwurfsplattformen für komplexe angewandte Systeme und Schaltungen der Mikroelektronik" gefördert.

Literatur

- Givargis, T. and Vahid, F.: Parameterized System Design, IEEE/ACM International Symposium on System Synthesis (ISSS), pp. 163–169, 2000.
- Givargis, T., Vahid, F., and Henkel, J.: Evaluating Power Consumption of Parameterized Cache and Bus Architectures in System-on-a-Chip Designs, IEEE Transactions on VLSI, Vol 9, No. 4, pp. 500–508, 2001a.
- Givargis, T., Vahid, F., and Henkel, J.: System-level Exploration for Pareto-optimal Configurations in Parameterized Systems-on-a-chip, International Conference on Computer Aided Design, 2001b.
- Kam, T. and Brayton, R.: Multi-valued Decision Diagram, Tech. Rep. M90/125, U. C. Berkeley, UCB/ERL, 1990.
- Macii, E., Pedram, M., and Somenzi, F.: High-Level Power Modeling, Estimation and Optimization, Proceedings of DAC, pp. 504–511, 1997.
- Marculescu, D., Marculescu, R., and Pedram, M.: Information Theoretic Measures for Power Analysis, IEEE Trans. on CAD of Integrated Circuits, 15, 1996.
- Raghunathan, A., Dey, S., and Jha, N. K.: Register-transfer level Estimation Techniques for Switching Activity and Power Consumption, Proceedings of ICCAD, pp. 158–165, 1996.
- Synopsys-2001.08, Synopsys Online Documentation: HDL Compiler for VHDL Reference Manual, 2001.
- Verisity-2002, <http://www.verisity.com/prodcuts/specmanelite.html>, 2002.