

# 1 Testauswahl auf Basis abstrakter Schaltungsmodelle

Der Entwurfsaufwand informationsverarbeitender Systeme wird überwiegend von der Anzahl der im Entwurfsprozess entstehenden Fehler und ihrem Beseitigungsaufwand bestimmt. Der Untersuchungsgegenstand in diesem Projekt sind große digitale Schaltungen, die aus bis zu mehrere Millionen von Gattern bestehen sollen. Die Technik, Systeme dieser Größe zu entwerfen, in FPGAs zu programmieren und zu testen, stehen heute den Universitäten und jedem anderen Interessenten zu Verfügung.

Schwerpunktmäßig sollen zwei Aspekte und ihre Verknüpfung untersucht werden:

- abstrakte Funktionsmodelle, die helfen, die Entwurfsbeschreibung kleiner und übersichtlicher zu halten
- eine halbautomatische Erzeugung großer Mengen von Testbeispielen aus abstrakten Modellen.

Abstrakte Modellierung soll hier bedeuten, dass der Entwerfer nur die wesentlichen Verhaltensvorgaben beschreibt und der Rechner die übrigen benötigten Informationen über algorithmische Vorgaben ergänzt. Das ist ein bekanntes und bewährtes Konzept der Informatik, das bei der Entwicklung der Programmiersprachen von den Maschinensprachen über die Assemblersprachen, die Hochsprachen bis hin zu Sprachen und Entwicklungsumgebungen für spezielle Problemlassen wie Datenbanken immer eine tragende Rolle gespielt hat. Es ist das Grundkonzept, um mit demselben Arbeitsaufwand und der gleichen zu erwartenden Fehlerrate immer größere Systeme entwickeln zu können. Im Hardwarebereich hat es eine ähnliche Entwicklung gegeben vom Entwurf auf der geometrischen Ebenen über den Entwurf auf der Transistorebene, den Entwurf auf der Gatterebene etc.. Ein wesentlicher Meilenstein war die Einführung von VHDL, eine Sprache, die auch eine Beschreibung und Simulation einer Schaltung mit abstrakten Datentypen und Algorithmen erlaubt. Für ein bestimmtes Subset der Sprache gibt es die Möglichkeit der Synthese. Mit noch höheren Abstraktionsebenen wird seit Jahren experimentiert (Stichworte Silicon-Compiler [Quelle?], Hardware-Beschreibung mit UML [Quelle?]).

Die Orientierung auf eine halbautomatische Erzeugung großer Mengen von Testbeispielen geht auf eine eigene Studie in [eigens Buch] zurück, die wegen des Umfangs der verketteten Zusammenhänge in ein Lehrbuch eingebettet wurde. Es gibt keine Möglichkeit, in einem Entwurf Fehler auszuschließen, sondern nur die zu erwartende Häufigkeit ihres Auftretens ausreichend klein zu halten. Alle Entwurfskontrollen

- Syntaxkontrollen
- formalen Beweistechniken, z.B. um die Identität zweier logischer Funktionen nachzuweisen, oder, dass jeder zulässige Eingabewert auch auf einen zulässigen Ausgabewert abgebildet wird

- Simulation von Testbeispielen etc.

wirken wie Filter, die einen bestimmten Teil der Fehler herausfiltern und andere nicht bemerken. Eine Schlüsselstellung nimmt dabei der Test mit zufälligen Eingaben ein. Denn das ist einerseits die einzige Kontrolle auf Abwesenheit unvorhergesehener Fehlertypen, die es immer geben kann. Andererseits gibt es einen direkten statistischen Zusammenhang zwischen der Testdauer und der späteren Zuverlässigkeit des Systems. Zufallstests müssen immer sehr lang sein (Test- und Simulationszeiten von Tagen bis Wochen) und deshalb automatisiert ablaufen.

Tage und Wochen simulieren zu lassen. Außerdem enthält die Theorie einen recht großen Unsicherheitsfaktor, das Operationsprofil. Es geht um das jedem Praktiker bekannte Phänomen: Ein erfahrener Nutzer kann auch mit einem fehlerhaften System oft gut arbeiten, weil die Fehler, die ihn bei der Einarbeitung gestört haben, beseitigt sind, oder er geeigneten Input-Work-Arounds gefunden hat. Der neue Nutzer, der das System etwas anderes benutzt, kämpft erstmal für einige Zeit ständig mit Problemen, bis auch die Fehler, die ihn stören, beseitigt sind, oder er sich an die notwendigen Input-Work-Arounds gewöhnt hat. Dasselbe Problem gibt es bei Hardware, wie man leider immer wieder bemerkt, wenn man einen neuen hochintegrierten Schaltkreis, einen neuen Sensortyp etc. versucht zu benutzen.

In dem Forschungsprojekt geht es in einem gewissen Sinn um eine Verifikation und eine Weiterführung der Studie in [eigens Buch]. Wir wollen für eine Menge möglichst allgemeingehaltener Funktionsklassen abstrakte Entwurfstemplates zusammenstellen. Die Entwurfstemplates sollen dabei so geartet sein, dass:

- sie die manuelle Beschreibung der Funktion drastisch vereinfacht
- eine parametrisierte Beschreibung des Operationsprofils angehängt werden kann
- es einen Algorithmus für die Abbildung auf eine synthesefähige VHDL-Beschreibung gibt

Wie diese Templates aussehen werden, ist noch offen. Wir haben bisher mit Protokollautomaten, Bildverarbeitungs- und anderen Signalverarbeitungsfunktionen und Speicherschnittstellen (SDRAM, DDR-RAM) experimentiert. Wünschenswert wäre, wenn es gelingt, eine Beschreibung zu finden, in der die eigentliche Informationsverarbeitung vom zeitlichen Verarbeitungsfluss getrennt beschrieben wird. Denn das würde sowohl der menschlichen Denkweise bei der Spezifikation von Funktionen als auch der Beschreibung sinnvoller Operationsprofile für die Testauswahl sehr entgegen kommen.