

Diss. ETH No 12188

An Integrated Framework for Structured Circuit Design with Field-Programmable Gate Arrays

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZÜRICH
(ETH Zürich)

for the degree of
Doctor of Technical Sciences

presented by
Stephan Walter Gehring
Dipl. Informatik-Ing. ETH
born April 17, 1965
citizen of Zürich, Switzerland

accepted on the recommendation of
Prof. Dr. Niklaus Wirth, examiner
Prof. Dr. Hans Eberle, co-examiner

1997

Kurzfassung

Field-Programmable Gate Arrays (FPGAs) sind programmierbare Schaltungsbausteine, welche durch den Entwickler selbst programmiert werden können und sich daher als schnelle und einfache Möglichkeit zur Schaltungsrealisierung anbieten.

Die Werkzeuge zur Schaltungsentwicklung sollten den Entwicklungsprozess in Bezug auf Geschwindigkeit und Einfachheit komplementieren. Sie sollten dem Entwickler interaktive Kontrolle über die Schaltungsimplimentation geben und sollten durch die Werkzeugentwickler leicht an die noch immer wachsende Zahl neuer FPGA Architekturen angepasst werden können.

In dieser Dissertation wird ein System zur Schaltungsentwicklung präsentiert, das diese Vorgaben erfüllt. Das Front-End besteht aus einem *Programmgerüst von eng gekoppelten architekturunabhängigen Werkzeugen*. Die enge Kopplung wird durch Verwendung einer *universellen hierarchischen Schaltungsrepräsentation* erreicht. Das Programmgerüst umfasst einen *Übersetzer* für die Schaltungsbeschreibungssprache Lola, einen *Schemaeditor*, ein *Programmgerüst für Schaltungseeditoren* und einen *Schaltungsprüfer*, welcher zwei Schaltungsrepräsentationen auf funktionale Äquivalenz überprüft.

Das Programmgerüst wird mit Back-Ends erweitert, welche architektur-spezifische Werkzeuge umfassen. Die Brauchbarkeit dieses Ansatzes wird durch Werkzeuge für das Xilinx XC6200 FPGA aufgezeigt. Diese umfassen einen *Schaltungseeditor* und einen *Schaltungsextrahierer*. Letzterer wird benötigt, um Verbindungen zwischen den mittels des Schaltungseeditors erzeugten Schaltungselementen herzustellen.

Messungen zeigen, dass beträchtliche Teile (40–50%) von Schaltungsentwicklungssystemen architekturunabhängig, d.h. wiederverwendbar, sind. Indem diese Funktionalität in ein gemeinsam benutztes Programmgerüst gekapselt wird, reduziert sich der Aufwand zur Entwicklung von neuen Back-Ends substantiell. Die vorliegende Arbeit unterstreicht ferner die Wichtigkeit hierarchischer Datenstrukturen, nicht nur um Schaltungen zu repräsentieren, sondern auch um die Ausführungszeit von Werkzeugen stark zu reduzieren. Um davon zu profitieren, müssen jedoch alle beteiligten Werkzeuge diese hierarchische Repräsentation unterhalten.

Die enge Kopplung besitzt Vorteile sowohl für den Anwender, als auch für den Programmierer. Der Benutzer profitiert von kurzen Antwortzeiten und schnellem Umschalten zwischen den im Entwicklungsprozess beteiligten Werkzeugen. Für den Programmierer erleichtert die hohe Integration die Werkzeugentwicklung, da komplexe Prozeduren zur Kommunikation zwischen den Werkzeugen und Datenstrukturkonversionen entfallen.

Abstract

In recent times, Field-Programmable Gate Arrays (FPGAs) have become a viable alternative to Application-Specific Integrated Circuits for implementing digital circuits. FPGAs are programmable logic devices which can be programmed entirely by the customer and therefore offer a quick and simple way to realize circuits.

The software tools used to implement circuits with FPGAs should complement the design process in terms of speed and ease of use: they should give the designer interactive control over the circuit implementation and should be easily adaptable to the still growing number of FPGA architectures.

In this thesis, we present a system for circuit design which addresses these issues. The front-end consists of a *framework of tightly integrated architecture-independent tools*. Tight integration is achieved by using a *universal hierarchical circuit representation* throughout the system. The framework consists of a *compiler* for the hardware description language Lola, a *schematic editor*, a *(sub-)framework for layout editors*, and a *circuit checker* which allows to check two circuit representations for functional equivalence.

The framework is extended with back-ends comprising tools tailored to specific FPGA architectures. We demonstrate the viability of our approach by means of tools for the Xilinx XC6200 FPGA comprising a *layout editor* and a *circuit extractor*. The latter serves to create links between the circuit elements created using the layout editor.

Our measurements show that large parts (40–50%) of a circuit design system are architecture-independent, i.e. reusable. Encapsulating this functionality in a commonly used framework thus substantially reduces the effort to develop back-ends for new architectures. This work also stresses the importance of hierarchical data structures, not only to represent structured circuits, but also to significantly cut down the execution times of design tools. To be effective, the hierarchical information must be maintained by all design tools.

The tight coupling of the circuit design tools has benefits for both the user and the programmer of the system. For the user, the tight coupling results in a highly responsive system and fast transitions between the various tools used in the design process. For the programmer, the tight integration simplifies tool development by avoiding complex inter-tool communication procedures and conversions between different circuit representations.