

Rapid Prototyping mit FAUmaschine am Beispiel einer VHDL-PCI-Soundkarte

Stefan Potyra, Matthias Sand, Volkmar Sieh, Dietmar Fey
Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl für Rechnerarchitektur
Martensstr. 3, 91058 Erlangen

Email: {Stefan.Potyra, Matthias.Sand, Volkmar.Sieh, Dietmar.Fey}@informatik.uni-erlangen.de

Zusammenfassung—Das Koppeln der virtuellen Maschine FAUmaschine mit einem VHDL-Interface ermöglicht eine hoch-performante Kosimulation. Dies unterstützt den Systementwurf mittels Rapid Prototyping, z.B. für die Entwicklung von kompletten Baugruppen für einen PC. Das vorgestellte Verfahren bietet durch Abstraktion eine hohe Simulationsperformanz, ist jedoch dennoch detailliert genug, um konkrete Hardware-Modelle zu verifizieren.

I. ÜBERSICHT FAUMACHINE

FAUmaschine [1], [2] ist eine quelloffene Implementierung einer virtuellen Maschine, welche gebräuchliche PC-Hardware simulieren kann. FAUmaschine kann eine Vielzahl unterschiedlicher Betriebssysteme in unmodifizierter Form ablaufen lassen. Im Gegensatz zu anderen virtuellen Maschinen, wie QEMU [3], VirtualBox [4] oder VMware [5], sind die Komponenten von FAUmaschine stark an der zugrundeliegenden, realen Hardware orientiert. FAUmaschine kann interaktiv benutzt werden, bietet aber ebenfalls die Möglichkeit, automatisierte Systemtests durchzuführen [6]. Die Möglichkeit, Fehler zu injizieren [7], unterstützt das Entwickeln von fehlertoleranten Softwaresystemen.

II. KOPPLUNG DER VHDL-SIMULATION MIT EINER VIRTUELLEN MASCHINE

Die Hardware-Komponenten, die FAUmaschine simulieren kann, sind in C modelliert. Die Struktur des zu simulierenden Systems wird in VHDL beschrieben. Der VHDL-Compiler von FAUmaschine übersetzt diese Beschreibung in einen schnell interpretierbaren Zwischencode. Der VHDL-Compiler ermittelt, wie Signale genutzt werden. Eine Vermittlungsschicht gibt eine VHDL-Signaländerung wenn nötig an C Komponenten weiter und umgekehrt. Dies ermöglicht, dass eine VHDL-Beschreibung in transparenter Weise mit den in C modellierten Komponenten von FAUmaschine interagieren kann.

III. METHODE ZUR SCHNELLEN SIMULATION

FAUmaschine basiert auf der Methode der diskreten, ereignisorientierten Simulation. Signale oder Signalbündel verbinden Komponenten. Die Änderung eines Signals in einer Komponente hat das Aufrufen einer Callback-Funktion in jeder verbundenen Komponente zur Folge.

Die C-Implementierung von FAUmaschine kennt nicht nur Signale für einfache Typen (bspw. `boolean` oder `std_logic`), sondern auch Signale, welche komplette Bus-Transaktionen modellieren. Dies führt bereits zu einem enormen Geschwindigkeitsgewinn. Zusätzlich besitzen diese Bus-Signale einen Mechanismus, um die Zieladressen von angesprochenen Geräten zwischenspeichern. Somit kann der CPU-Simulator direkt auf die angesprochene Komponente zugreifen, ohne dass die Callbacks der anderen verbundenen Komponenten dazu aufgerufen werden müssen.

Bei der Anbindung einer VHDL-Komponente an den PCI-Bus stellt die hohe Taktfrequenz ein Problem dar. Folgende Verfahren gewährleisten dennoch eine hohe Performanz: Zum einen erfolgt die Abbildung von Bus-Zyklen nicht mit exakter Zeitauflösung. Im Gegenteil: Die Simulationszeit schreitet während eines Bus-Zyklus gar nicht voran. Lediglich ein Anstoßen des Interpreters nach Änderung des Taktsignals gewährleistet zeitliche Kausalität. Dies stellt jedoch keine Einschränkung bei der Simulation von synthetisierbarem VHDL dar.

Zum anderen wird das Taktsignal lediglich *während* eines Bus-Zugriffs simuliert. Dies steigert die Simulationsgeschwindigkeit gewaltig, führt jedoch zu folgender Einschränkung: Die Zustandsänderungen in der Hardwarebeschreibung müssen in direkter Abhängigkeit von Bus-Zugriffen erfolgen. Ein Zähler, welcher als Zählimpuls das Taktsignal verwendet, würde folglich

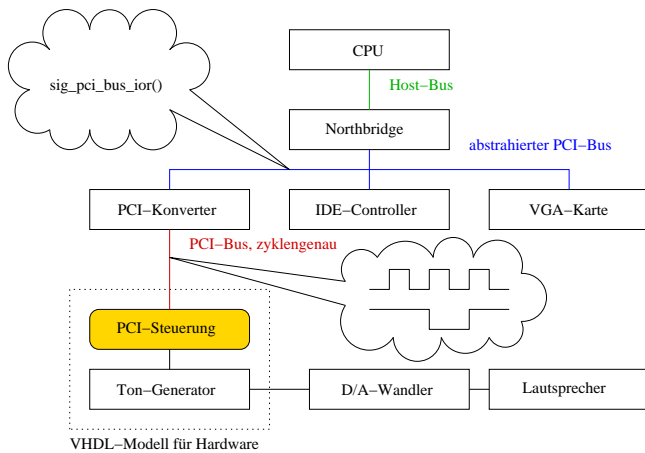


Abbildung 1. Simuliertes Modell

nicht korrekt simuliert.

Kein anderes Verfahren existiert, welches eine HDL-Simulation im Kontext eines vollständigen Systems mit ähnlicher Performanz simulieren kann.

IV. PROOF-OF-CONCEPT

A. Hardware-Modell

Um die Brauchbarkeit der vorgestellten Methode zu zeigen, wurde eine PCI-Soundkarte in VHDL modelliert. Die Karte besteht zum einen aus einer PCI-Steuerung, die die Signale des PCI-Busses dekodiert. Darüberhinaus besitzt die PCI-Steuerung die zur Konfiguration notwendigen Register. Ebenfalls sind Register zur Ansteuerung von 16 IO-Ports vorhanden. Die IO-Ports der PCI-Steuerung sind mit einem Ton-Generator verbunden. Dieser kann die Frequenzen für unterschiedliche Noten erzeugen. An den Ton-Generator ist ein Digital-Analog-Wandler angeschlossen. Der Ausgang dieses Wandlers kann mit handelsüblichen Lautsprecherboxen verbunden werden. Sowohl die PCI-Steuerung als auch der Ton-Generator wurden in VHDL modelliert. Das entwickelte Modell wurde auf einer PCI-Prototyp Karte mit einem CPLD des Typs *iM4A3-512/160* der Firma Lattice Semiconductors auf Funktionsfähigkeit verifiziert.

B. Simulation im Kontext von FAUmachine

Abbildung 1 zeigt das für FAUmachine verwendete Modell: Innerhalb der virtuellen Maschine werden abstrahierte Signale für den Host- und den PCI-Bus verwendet. Der PCI-Konverter wandelt diese in zyklengenaue Signale des PCI-Busses um. Auf diese Signale kann von VHDL aus zugegriffen werden.

Das entwickelte Hardware-Modell wurde in PCI-Steuerung und Ton-Generator aufgeteilt: Für die PCI-

Steuerung wurde **exakt** das zuvor entwickelte VHDL-Modell verwendet. Um zu zeigen, dass auch C-Komponenten aus dem Kontext von VHDL erzeugt und angesteuert werden können, wurde der Tongenerator in C modelliert. Das Interface des Ton-Generators stimmt hierbei mit dem der VHDL-Beschreibung überein. Zum automatisierten Testen wurde ein Test-Skript erstellt, das einen Selbsttest der PCI-Steuerung durchführt und anschließend ein Lied abspielt.

Durch die vorgestellte Methode, das PCI-Taktsignal nicht kontinuierlich zu verändern, wird die Performanz durch die VHDL-Simulation nur unwesentlich beeinträchtigt. Lediglich bei Zugriffen auf die VHDL-Karte, muss der abstrahierte Zugriff zyklengau abgebildet werden. Die VHDL-Simulation erzeugt einen durchschnittlichen Overhead von 4.08 % im Vergleich zu einer Simulation ohne PCI-Soundkarte. Interaktive Benutzung der virtuellen Maschine ist somit weiterhin möglich.

V. AUSBLICK

Die vorgestellte Verfahrensweise ist in der Lage, VHDL-Hardware-Modelle im Kontext einer virtuellen Maschine mit äußerst hoher Performanz zu simulieren. Dieser Ansatz ist jedoch im Hinblick auf das Taktsignal ungenau, und kann somit beispielsweise keine taktabhängigen Zähler korrekt nachahmen. Forschungsbedarf besteht darin, eine Methode zu finden, die auch in dem gerade genannten Fall sowohl hohe Performanz als auch korrekte Simulation gewährleistet.

LITERATUR

- [1] FAUmachine Team, "FAUmachine," URL: <http://www.FAUmachine.org/>, 2003–2009.
- [2] M. Sand, S. Potyra, and V. Sieh, "Deterministic high-speed simulation of complex systems including fault-injection," in *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2009.
- [3] F. Bellard, "QEMU, a fast and portable dynamic translator," in *ATEC'05: Proceedings of the USENIX Annual Technical Conference 2005 on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2005, pp. 41–46.
- [4] innotek GmbH, "VirtualBox," URL: <http://www.virtualbox.org/>, 2008.
- [5] VMware Inc., "VMware," URL: <http://www.vmware.com/>, 2001.
- [6] S. Potyra, V. Sieh, and M. Dal Cin, "Evaluating fault-tolerant system designs using FAUmachine," in *EFTS '07: Proceedings of the 2007 workshop on Engineering fault tolerant systems*. New York, NY, USA: ACM, 2007, p. 9.
- [7] H. Höxer, M. Waitz, and V. Sieh, "Fast simulation of stuck-at and coupling memory faults using FAUmachine," in *Supplement to Proc. HASE 2005: International Symposium on High Assurance Systems Engineering (Ninth IEEE International Symposium on High Assurance Systems Engineering Heidelberg, Germany 12-14 October 2005)*, Oct. 2005, pp. 1–2.