

Evaluation von OpenCL für FPGAs

Matthias Roth*, Peter Väterlein, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2015/2016

Field Programmable Gate Arrays (FPGAs) werden häufig als programmierbare Hardware beschrieben. Durch ihre Anpassungsfähigkeit an Probleme zeichnen sie sich in verschiedenen Anwendungen durch hohe Performanz bei guter Energieeffizienz im Vergleich zu CPU- und GPU-Lösungen aus. Bei der Geschwindigkeit und Energieeffizienz werden FPGAs jedoch meist von den nicht wiederprogrammierbaren anwendungsspezifischen Schaltungen (ASICs) übertroffen. Doch auch moderne FPGAs beinhalten immer mehr vorgefertigte Funktions-Blöcke, wie zum Beispiel arithmetische Recheneinheiten. Dies führt dazu, dass die Leistungsdifferenz zwischen FPGAs und ASICs kleiner wird, wodurch FPGA-Lösungen wirtschaftlich sinnvoller werden und ihr Einsatz in immer mehr Anwendungsgebieten attraktiv erscheint.

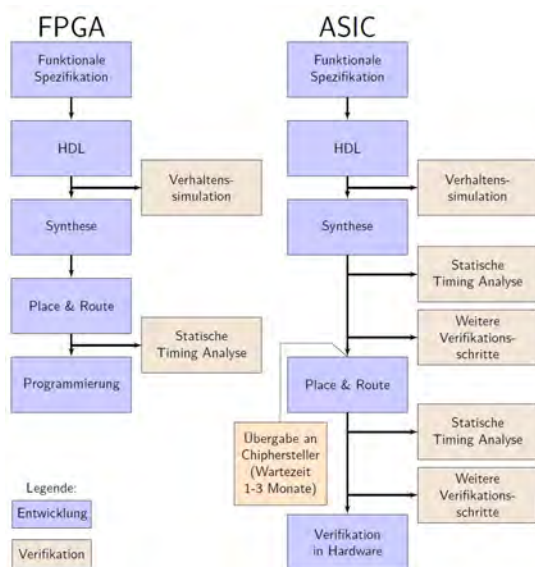


Abbildung 1: Entwicklung von FPGA- und ASIC-Designs im Vergleich

Die Anwendungsentwicklung für digitale Hardwarekomponenten wie FPGAs ist sehr komplex. Ein Algorithmus muss erst in einzelne Logikbausteine zerlegt werden, wozu detaillierte Kenntnisse der Hardware nötig sind und beachtet werden muss, dass die Ressourcen eines FPGAs begrenzt sind. Wie diese

Bausteine verwendet werden und welche Beziehungen zwischen diesen herrschen, wird in einer Hardwarebeschreibungssprache wie beispielsweise VHDL festgehalten. Daraufhin wird die beschriebene Logik in der Synthese an die Hardware des FPGAs angepasst und anschließend muss das Ergebnis der Synthese noch auf die vorhandene Hardware gelegt (Place) und verdrahtet (Route) werden. Nach diesem Schritt müssen bei zeitkritischen Anwendungen unter Umständen noch die Timings von einzelnen Verdrahtungen überprüft werden. ASICs sollten in der Entwicklung zusätzlich in mehreren Schritten verifiziert werden, da nachträgliche Änderungen mit hohen Mehrkosten verbunden sind, dadurch erhöht sich der Aufwand in der Entwicklung wie Abbildung 1 anschaulich zeigt. Der Einsatz von digitalen Hardwarekomponenten führt somit zu einer vergleichsweise hohen Entwicklungszeit, wodurch sie nur bei hohen Stückzahlen wirtschaftlich konkurrenzfähig zu CPU- und GPU-Lösungen sind.

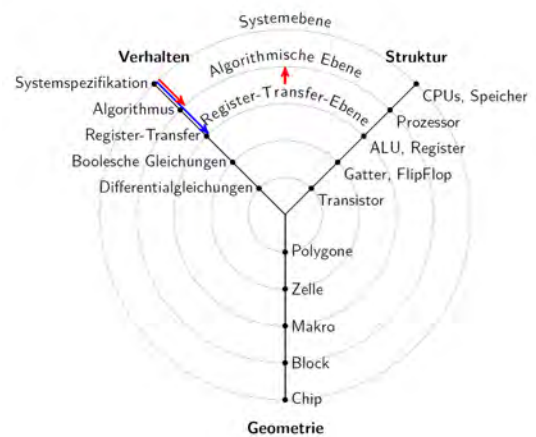


Abbildung 2: Y-Diagramm nach Gajski

In den 1980er Jahren wurde das Y-Diagramm (s. Abbildung 2) von Gajski und Kuhn entworfen, welches die verschiedenen Sichtweisen auf die Hardwareentwicklung beschreibt. Der Entwurf einer Anwendung erfordert bei der Verwendung von Hardwarebeschreibungssprachen ein Vorgehen, bei welchem Konzepte für alle Ebenen von der Sys-

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Renningen

temebene bis zur Register-Transfer-Ebene erstellt werden müssen (blauer Pfeil). Beispielsweise im Bereich der Bildverarbeitung tritt nun das Problem auf, dass digitale Hardwarekomponenten die Algorithmen stark beschleunigen, diese sich aber schnell ändern oder schnell ausgetauscht werden, sodass der Einsatz von ASICs oder FPGAs den Fortschritt einschränkt, da der Fokus auf die Hardwareentwicklung gerichtet wird und nicht auf das Problem. Es gibt bereits verschiedene Ansätze zur Abstraktion der Entwicklung nach Vorbild des Y-Diagramms (rote Pfeile). Meist werden Programmiersprachen verwendet, welche auf die sequentielle Verarbeitung von Befehlen ausgelegt sind. Die Parallelität der FPGAs lässt sich mit diesen Sprachen oft nur ineffizient nutzen, weshalb sich deren praxistaugliche Verwendung nur auf spezielle Anwendungsgebiete beschränkt.

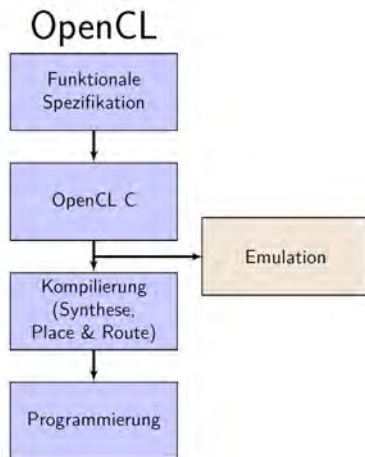


Abbildung 3: Entwicklung von FPGAs mit OpenCL

Die beiden führenden FPGA-Hersteller bieten nun mit OpenCL eine neue Programmiermethodik an, die ebenfalls auf ein Hochsprachenkonzept aufsetzt[1][2]. Es handelt sich dabei um ein Framework für hochgradig parallelisierbare Probleme. Das bedeutet für einen Entwickler, dass dieser das Verhalten nun als Algorithmus in einer C ähnlichen Sprache

beschreiben kann und kein Umdenken zu Logikbausteinen mehr nötig ist. Das entwickelte Programm kann am PC emuliert werden um die funktionale Richtigkeit sicherzustellen, bevor der Ziel-FPGA nach dem Kompilervorgang programmiert wird (s. Abbildung 3). FPGAs sollen dadurch bei dem in der Produktentwicklung wichtigen Maß *Time-to-Market* mit anderen Lösungen mithalten können.

Dieser Anspruch, eine hochperformante Lösung für ein komplexes Problem mit geringem Aufwand zu finden, wird in dieser Arbeit untersucht. Die Evaluation wird anhand eines Beispiels aus der Bildverarbeitung durchgeführt. Der gewählte Algorithmus beansprucht die Ressourcen von FPGAs in hohem Maße und eignet sich so hervorragend zur Untersuchung der OpenCL-Lösung. Durch den Vergleich mit einer bestehenden VHDL-Lösung kann der Entwicklungsprozess optimal verglichen werden. Dabei wird bewertet, wie sich die Entwicklungszeit verändert und welche Hardwarekenntnisse nötig sind, um eine ressourcenoptimierte Lösung zu erhalten. Anschließend wird die entstandene Anwendung nach verschiedenen Automotive-Kriterien wie Safety und Echtzeitfähigkeit bewertet.

Ziel der Arbeit ist die Beurteilung der Umsetzbarkeit von OpenCL auf FPGAs und des Reifegrades des Altera OpenCL SDKs. Zudem wird eine mögliche FPGA-Architektur speziell für den Einsatz von OpenCL untersucht. Es wird bestimmt welches Verhältnis zwischen der Logik, dem Speicher und den Rechen-einheiten bestehen sollte, wenn die FPGA-Anwendung mit OpenCL entwickelt wird. Dazu gilt es herauszufinden welche Strukturen der OpenCL-Compiler effizient umsetzen kann und welche Strukturen häufig genutzt werden. Schließlich soll die optimale FPGA-Architektur für die Bildverarbeitung mit OpenCL gefunden werden. Das Ergebnis wird außerdem Aufschluss darüber geben, ob OpenCL bereits im Automotive-Umfeld eingesetzt werden kann, beziehungsweise aufzeigen, welche Fortschritte noch erzielt werden müssen.

[1] Altera SDK for OpenCL: <https://www.altera.com/products/design-software/embedded-software-developers/opencv/overview.html>

[2] Xilinx SDAccel Development Environment: <http://www.xilinx.com/products/design-tools/software-zone/sdaccel.html>

Bildquellen:

- Abbildung 1: Eigene Darstellung nach <http://www.xilinx.com/fpga/asic.htm>
- Abbildung 2: Eigene Darstellung nach F. Kesel: Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs, De Gruyter Verlag, 2013
- Abbildung 3: Eigene Darstellung