

Diss. ETH No. 16201

On Instruction-Set Generation for Specialized Processors

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
GERO DITTMANN
Dipl.-Ing.
born March 15, 1974
citizen of Germany

accepted on the recommendation of
Prof. Dr. Lothar Thiele, examiner
Prof. Dr. Paolo Ienne, co-examiner
Prof. Dr. Andreas Herkersdorf, co-examiner

2005

Abstract

Owing to the ever-decreasing feature size of today's semiconductor processes, the cost of a mask set has already crossed the one-million-dollar line. Given this investment, a design must be applicable for multiple purposes. This flexibility is commonly provided by programmable elements. A gradual trade-off between the flexibility of general-purpose processor cores and the performance of hard-wired logic can be achieved with *application-specific instruction-set processors* (ASIPs).

Many automated ASIP design methods found in the literature today employ a library of patterns that represent potential specialized instructions. As the libraries tend to grow large their access times become a critical factor. However, no attempts have yet been made to speed up these searches in the pattern libraries. Furthermore, there are no methods available to recognize and exploit structural similarities between patterns.

Another deficiency in today's ASIP design methodologies is their exclusive focus on the data-dominated domain characterized by computation-intensive applications such as digital signal processing. This focus entails a lack of methods for control-dominated domains such as network processing. These domains are characterized by branch-intensive applications with fine-grained timing constraints imposed by frequent interactions with the ASIP environment. The major challenge here is not to speed up the over-all runtime of the applications, but to meet the many timing constraints. This challenge can be addressed by introducing special instructions that speed up the timing-critical paths.

In this thesis we propose a hierarchical organization for pattern libraries that removes the dependency of search times on the library size. In this way, much larger libraries can be handled which removes the need for heuristics to prune patterns from the library. Exact methods become possible. In our experiments we found that searches in our structure are orders of magnitude faster than in a linked-list library. Furthermore, we introduce a method that employs identity operands to find synergies between similar patterns. These similarities can be exploited to achieve leaner instruction sets and for data-path sharing.

On top of these library structures, we propose the first integrated ASIP design methodology for the control-dominated domain. We introduce novel methods to specify fine-grained timing constraints in ANSI C, to include them in an intermediate representation that facilitates compiler optimizations, and to derive an instruction set that enables the ASIP to meet the timing constraints. We present a case study that demonstrates the feasibility of our methods and the quality of the results.

Zusammenfassung

Die stetig schrumpfenden Strukturgrößen moderner Halbleiterprozesse haben dazu geführt, dass die Kosten für einen Belichtungssatz bereits eine Million Dollar überschreiten. Damit sich diese Investition auszahlt, muss ein Design vielfältig einsetzbar sein. Die erforderliche Flexibilität wird üblicherweise mit Hilfe von programmierbaren Bausteinen erreicht. Eine feinstufige Abwägung zwischen der Flexibilität von General-Purpose-Prozessoren auf der einen Seite und der Leistungsfähigkeit von fest verdrahteten Schaltungen auf der anderen Seite wird ermöglicht von *application-specific instruction-set processors* (ASIPs).

Viele der veröffentlichten automatisierten ASIP-Design-Methoden bedienen sich einer Bibliothek von Mustern, die Kandidaten für Spezialbefehle darstellen. Da diese Bibliotheken sehr groß werden können, spielt ihre Zugriffszeit eine entscheidende Rolle. Dennoch wurde bisher nicht untersucht, wie die Suche in einer Musterbibliothek beschleunigt werden kann. Darüberhinaus sind keine Methoden bekannt, um Gemeinsamkeiten in der Struktur zweier Muster zu erkennen und auszunutzen.

Eine weitere Einschränkung heutiger ASIP-Design-Methodologien ist ihre ausschließliche Beschäftigung mit der Klasse der rechenintensiven, datendominierten Anwendungen wie digitaler Signalverarbeitung. Es fehlt daher an Methoden für kontrolldominierte Anwendungsklassen wie der Verarbeitung von Netzwerkprotokollen. Diese Klassen sind gekennzeichnet durch Anwendungen mit vielen Verzweigungen und detaillierten Zeitanforderungen, die von häufigen Interaktionen mit der Umgebung herrühren. Unter diesen Bedingungen besteht die Herausforderung nicht darin, die Gesamtlaufzeit der Anwendungen zu verringern, sondern sämtliche Zeitanforderungen zu erfüllen. Dies kann erreicht werden mit Hilfe von Spezialbefehlen, die die Ausführung von zeitkritischen Pfaden beschleunigen.

Diese Dissertation führt eine hierarchische Organisation für Musterbibliotheken ein, die die Abhängigkeit der Suchgeschwindigkeit von der Größe der Bibliothek aufhebt. Diese Eigenschaft macht wesentlich größere Bibliotheken beherrschbar, wodurch Heuristiken zum Verwerfen von Mustern vermieden werden können. Exakte Methoden werden möglich. Experimente zeigen, dass das Suchen in dieser Struktur um Größenordnungen schneller ist als in einer Bibliothek, die als verkettete Liste aufgebaut ist. Weiterhin wird eine Methode vorgestellt, die mit Hilfe von Identitätsoperanden Synergien zwischen sich ähnelnden Mustern findet. Die Ähnlichkeiten bieten die Möglichkeit, schlankere Befehlssätze zu erreichen und Datenpfade mehrfach zu nutzen.

Aufbauend auf diesen Bibliotheksstrukturen wird die erste integrierte ASIP-Entwicklungsmethodologie für kontrolldominierte Anwendungsklassen vorgestellt. Es werden neuartige Methoden entwickelt, um detaillierte Zeitanforderungen in ANSI-C zu spezifizieren, sie in eine Graphendarstellung für Compileroptimierungen einzubinden und daraus schließlich einen Befehlssatz abzuleiten, der es dem ASIP ermöglicht, die Zeitanforderungen einzuhalten. Abschließend wird anhand einer Fallstudie die Machbarkeit der Methoden und die Qualität der Ergebnisse demonstriert.