Reimund Klemm

HW/SW Codesign für effiziente bitstromorientierte Datenverarbeitung eingebetteter Systeme

Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung
Nr. 57

Reimund Klemm

HW/SW Codesign für effiziente bitstromorientierte Datenverarbeitung eingebetteter Systeme

VOGT

Dresden 2012

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Bibliographic Information published by the Deutsche Bibliothek
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibiograpic data is available in the internet at http://dnb.ddb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2012

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „HW/SW Codesign für effiziente bitstromorientierte Datenverarbeitung eingebetteter Systeme“ von Reimund Klemm überein.

Gesetzt vom Autor

ISBN 978-3-938860-50-2

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

HW/SW Codesign für effiziente bitstromorientierte Datenverarbeitung eingebetteter Systeme

Reimund Klemm

von der
Fakultät Elektrotechnik und Informationstechnik
der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

Doktoringenieurs
(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. René Schüffny		
Gutachter:	Prof.Dr.-Ing. Gerhard Fettweis	Tag der Einreichung:	07.11.2011
	Prof.Dr.-rer.nat Wolfram Hardt	Tag der Verteidigung:	12.04.2012

Zusammenfassung

Aktuelle Standards für den breitbandigen zellularen Mobilfunk wie HSPA oder LTE enthalten zahlreiche Protokolle, die eine umfangreiche Verarbeitung von Bitströmen benötigen. Dazu ist für die mobilen Endgeräte wie Smartphones eine energieeffiziente Verarbeitung notwendig. Zusätzlich ist aufgrund der komplexen und umfangreichen Spezifikationen der einzelnen Protokolle eine soweit als mögliche Verarbeitung in Software wünschenswert.

In dieser Arbeit wird untersucht, wie sich anwendungsspezifische Prozessoren (ASIPs) für energieeffiziente Bitstromverarbeitung in Protokollen verwenden lassen. Diese ermöglichen es kritische Bereiche durch Hardware Erweiterungen zu beschleunigen. In dieser Arbeit werden verschiedene Erweiterungen für die Bitstromverarbeitung vorgestellt und untersucht.

Die eigentliche Verarbeitung erfolgt aber weiterhin durch Software. Für den Softwareentwurf wird dabei in dieser Arbeit ein neuer Ansatz gewählt. Komplexe Bitströme werden mit Hilfe von kurzen Metaprogrammen knapp und präzise beschrieben. Aus der Bitstrombeschreibung wird vom Compiler mittels Metaprogrammierung umfangreicher Verarbeitungscode generiert. Dieser lässt sich vom Compiler unter Ausnutzung gängiger Optimierungen zu sehr energieeffizienten Maschinencode umwandeln. Dabei können Hardwareerweiterungen des ASIPs eingebunden werden, ohne die Software der Bitstromverarbeitung zu ändern.

Ziel dieser Arbeit ist es zu zeigen, wie eine effiziente Bitstromverarbeitung für eingebettete Systeme möglich ist, bei gleichzeitig einer schlanken und plattformunabhängigen Softwareschnittstelle.

Abstract

Current broadband cellular wireless standards like HSPA or LTE contain various protocols that require tremendous amount of bitstream processing. For the handheld devices like smartphones an energy efficient processing is required. Additionally the complex and extensive protocol specifications favor a much as possible software based processing solution.

This work investigates how application specific instruction-set processors (ASIPs) can be employed to do energy efficient bitstream processing. ASIPs allow to accelerate performance critical hot-spots by custom hardware extensions. In this work different bitstream processing hardware extensions are proposed and investigated.

However, the actual processing is still done by software. For the software design this work proposes a new methodology. Complex bitstreams are described by short and precise meta-program constructs. That constructs are transformed by the compiler at compile time to extensive bitstream manipulation code. By using standard optimization features the compiler is able to generate energy efficient assembly code. Additionally custom hardware extensions of the ASIP can be included without changing the software doing the actual bitstream processing.

This work demonstrates energy efficient bitstream processing for embedded systems by proposing a lightweight and platform independent software interface.

Danksagung

Diese Arbeit entstand während vier sehr herausfordernder Jahre am Vodafone Stiftungslehrstuhl in der Hardware Entwurfsgruppe. Ich möchte mich bei meinen früheren Kollegen bedanken Teil dieses sehr erfolgreichen Teams gewesen zu sein.

Für die Betreuung, fachlichen Anregungen und Unterstützung für diese Arbeit möchte ich Professor Gerhard Fettweis herzlich danken.

Danken möchte ich auch meiner Familie, insbesondere meiner Frau Claudia, für die mentale Unterstützung während der Anfertigung der Arbeit.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

3GPP	Third Generation Partnership Project
ADSL	Asynchronous Digital Subscriber Line
ABI	Application Binary Interface
ADC	Analog-to-Digital Converter
ADL	Architecture Description Language
ASIP	Application-Specific Instruction-Set Processor
CISC	Complex Instruction Set Computing
CFG	Control Flow Graph
CKF	Compiler Known Functions
DAC	Digital-to-Analog Converter
DFG	Data Flow Graph
DSP	Digital Signal Processor
eNodeB	E-UTRAN NodeB
EBNF	Extended Backus-Naur Form
GPL	GNU General Public License
H-ARQ	Hybrid Automatic Repeat Request
HDL	Hardware Description Language
HSPA	High-Speed Packet Access
IDE	Integrated Development Environment
IP	Intellectual Property
IP	Internet Protocol
LOC	Lines of Code
LTE	Long Term Evolution
MAC	Media Access Control
MMU	Memory Management Unit

PDPC	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PHY	Physical Layer
RISC	Reduced Instruction Set Computing
RLC	Radio Link Control
RRC	Radio Resource Control
SDU	Service Data Unit
SIMD	Single Instruction Multiple Data
SoC	System-on-Chip
SRAM	Static Random Access Memory
STA	Synchronous Transfer Architecture
STL	Standard Template Library
TCP	Transmission Control Protocol
TIE	Tensilica Instruction Extension
UE	User Equipment (3GPP Bezeichnung für das mobile Funkgerät)
UML	Unified Modeling Language
UTRAN	Universal Terrestrial Radio Access Network
VLIW	Very Long Instruction Word
WiMAX	Worldwide Inter-operability for Microwave Access

Kapitel 1

Einleitung

Im Bereich der Protokollebenen oberhalb der physikalischen Schicht von Mobilfunkstandards findet eine umfangreiche bitstromorientierte Datenverarbeitung statt. Bisher wird dieser Bereich im Mobiltelefon meist auf eingebetteten RISC Prozessoren mit festem Instruktionssatz der Firma ARM verarbeitet. Mit neuen zellularen Breitbandstandards wie Long Term Evolution (LTE) ist mit einem starken Anstieg der Verarbeitungskomplexität für die Protokollebenen zu rechnen.

Diese Arbeit unterbreitet Vorschläge für die bitstromorientierte Datenverarbeitung mittels HW/SW-Codesign von anwendungsspezifischen Prozessoren (ASIPs). Der Schwerpunkt der Arbeit ist dabei die Verbesserung der Entwurfsmethodik für die Softwareentwicklung von ASIP Software am Beispiel der Bitstromverarbeitung. Mithilfe von Metaprogrammierungstechniken wird eine domänenspezifische Bibliothek zur Bitsromverarbeitung (in der Arbeit `Chameleo_Lib` genannt) entwickelt. Diese erlaubt trotz hohem Abstraktionsgrad der Software eine effiziente Abarbeitung durch Ausnutzung von Spezialinstruktionen von ASIPs.

1.1 Mobilfunk SoCs

Heutzutage sind Chiplösungen für Mobiltelefone und Smartphones sehr heterogene Systeme, deren Komponenten aus einer Menge von verschiedenen Hardware-Architekturen bestehen, um den Rechenanforderungen von Mobilfunkstandards und Multimediaanwendungen gerecht zu werden. Deshalb werden die aktuellen Chiplösungen als System-on-Chips (SoC) bezeichnet. Die einzelnen Komponenten eines SoCs sind dabei RISC Prozessoren, DSPs und spezielle Hardwarebeschleuniger, die in einer Gesamtlösung mittels eines Bussystemes oder Netzwerkes auf dem Chip integriert sind.

Schematisch ist dies für ein Beispiel SoC für ein LTE Empfänger in Abbildung 1.1 dargestellt. Die RISC Prozessoren übernehmen meist die Aufgaben der Benutzerschnittstelle, des Betriebssystems, der Verarbeitung der Protokollstapel und der Ablaufsteuerung. Die DSPs werden für die Signalverarbeitungsalgorithmen der physikalischen Schicht für die mobile Datenkommunikation eingesetzt. Besonders rechenintensive und zeitkritische Algorithmen wie die Kanaldekodierung oder Synchronisation werden in vielen SoCs in spezielle Hardwarebeschleuniger abgebildet.

Ein wesentlicher Unterschied der SoCs im Mobilfunkbereich zu Standardprozessoren oder SoCs im Automobilbereich ist die höhere Heterogenität der Systeme. Die zentrale Herausforderung der mobilen Kommunikation ist die Verringerung der Leistungsaufnahme der Gesamtlösung, da die Energieversorgung durch die Batterie begrenzt ist.

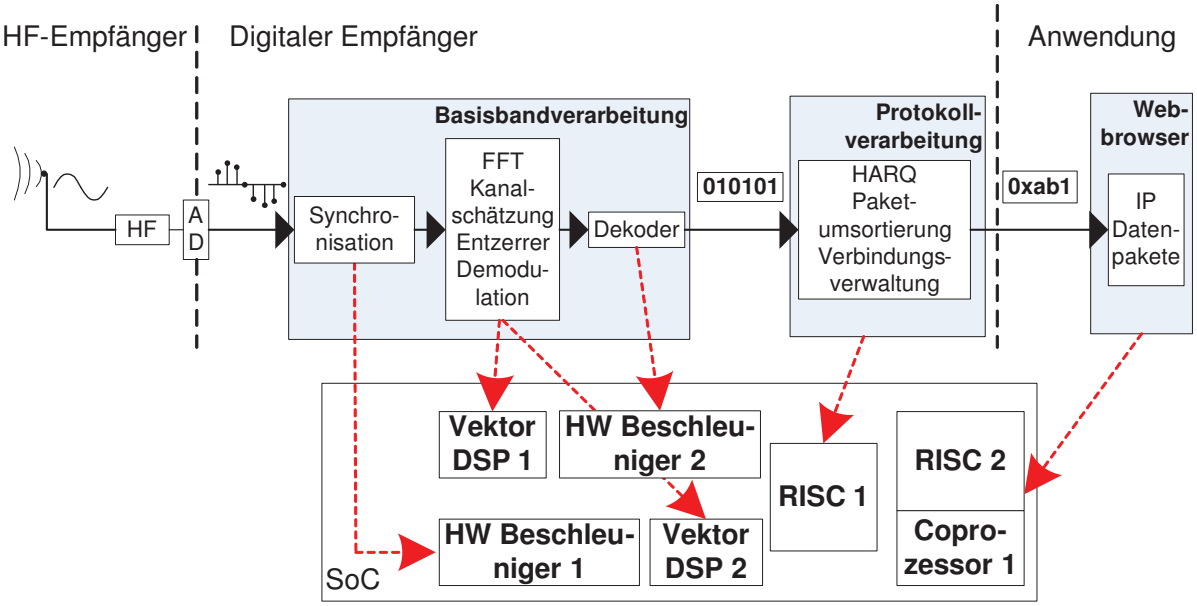

Abbildung 1.1: Vereinfachter LTE-Empfänger mit Abbildung der Algorithmen auf ein SoC

Der Energieverbrauch während der Nutzung des Gerätes ist stark von der Effizienz der Komponente (RISC, DSP oder HW Beschleuniger) für den jeweiligen Algorithmus oder die Anwendung abhängig. Effizienzmaße sind Flächenverbrauch und Leistungsaufnahme der Komponente, sowie die Abarbeitungszeit und Programmgröße der Anwendung. Eine Verarbeitung in einem speziellen HW-Beschleuniger ist in der Regel die effizienteste Lösung hinsichtlich Abarbeitungsgeschwindigkeit und Energieverbrauch zur Laufzeit. Ein wesentlicher Nachteil ist der deutlich erhöhte Verifikationsaufwand und Integrationsaufwand des Gesamtsystems. Programmierbare Komponenten wie RISC oder DSP Prozessoren sind

dabei deutlich flexibler. Durch Modifikationen der Software kann ein SoC an veränderte Anforderungen im Nachhinein angepasst werden.

Einen Mittelweg zwischen Flexibilität durch Softwareanpassungen und Effizienzsteigerung durch Zuschnitt der Hardware auf einen konkreten Algorithmus bieten die anwendungsspezifischen Prozessoren (ASIPs). Diese finden vor allem dort eine zunehmende Verbreitung, wo der Leistungsverbrauch der Gesamtlösung ein entscheidendes Wettbewerbskriterium ist. Dies ist vor allem bei den SoCs für mobile Kommunikation der Fall.

1.2 Applikationsspezifische Prozessoren - ASIPs

Der Hauptvorteil der ASIPs ist die Möglichkeit der Beschleunigung durch Hardwarezuschnitt auf einen Algorithmus in Form von Spezialinstruktionen. Der Ausgangspunkt ist meist ein Prozessordatenpfad wie der eines RISC Prozessors, der die Abarbeitung von Programmcode ermöglicht. Ist zudem ein Compiler für den ASIP vorhanden, können Algorithmen oder kritische Anwendungskerne aus Hochsprachen wie C/C++ als Ausgangspunkt genutzt werden.

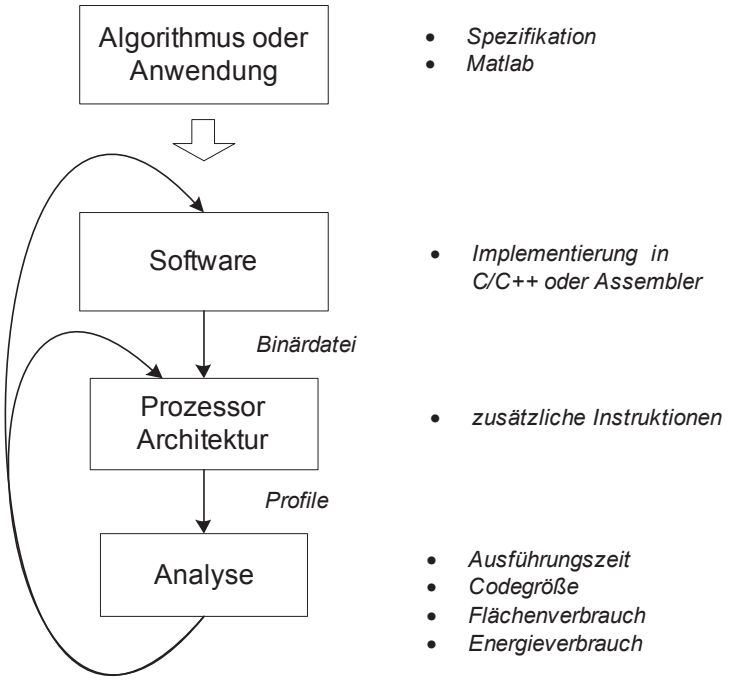

Abbildung 1.2: ASIP Entwurfsfluss

Während des Systemdesigns müssen durch Messung der Laufzeit (*engl. Profiling*) oder

Abschätzung zeitkritische Algorithmenkerne identifiziert werden. Anschließend werden diese durch Hardwareerweiterungen im Prozessor beschleunigt. Danach muss eine Anpassung der Software für die veränderte ASIP Architektur erfolgen. Anschließend kann ein erneutes Profiling stattfinden. Dieser Prozess wird iterativ bis zu dem gewünschten Ergebnis ausgeführt (siehe Abbildung 1.2).

1.3 Ziele dieser Arbeit

Die Ziele dieser Arbeit sind:

- ▷ Untersuchung zum Aufbau von Bitströmen in der Protokollverarbeitung im Mobilfunk
- ▷ kurze und präzise Beschreibung von komplexen Bitströmen in Software, unabhängig von der verwendeten Prozessorplattform
- ▷ Entwurf von angepasster Hardware basierend auf ASIPs zur Beschleunigung der Verarbeitung von Bitströmen im Bereich des Mobilfunks
- ▷ Ausnutzung der angepassten Hardware durch die Software ohne Mehraufwand für den Softwareentwickler

1.4 Gliederung der Arbeit

Die vorgelegte Arbeit gliedert sich in drei Teile. Der erste Teil ist eine Analyse der Problemstellung zur Bitstromverarbeitung und den bisher entwickelten Ansätzen. Dazu wird im 2. Kapitel der Aufbau des High Speed Packet Access (HSPA) und des LTE Protokollstapels genauer erläutert. Dabei wird insbesondere die Problematik der Bitstromverarbeitung herausgestellt. Zudem wird anhand des Xtensa-Entwurfsflusses der Firma Tensilica eine Werkzeugumgebung für ASIPs vorgestellt. Im 3. Kapitel wird der bisherige Stand der Technik zur Bitstromverarbeitung auf eingebetteten Prozessoren wiedergegeben.

Der zweite Teil der Arbeit beinhaltet einen Lösungsvorschlag zur effizienten Bitstromverarbeitung. Dazu werden die bekannten Techniken der aktiven Bibliotheken verwendet, welche im 4. Kapitel genauer erläutert werden. Darauf aufbauend wird die Bibliothek `Chameleo_Lib` im 5. Kapitel eingeführt, die eine kompakte Beschreibung und eine effiziente Abarbeitung von Bitströmen ermöglicht.

Der letzte Teil der Arbeit vergleicht die Bibliothek `Chameleo_Lib` zur Bitstromverarbeitung mit anderen bestehenden Ansätzen. Dabei wird die Ausführungszeit für verschiedene ASIP Konfigurationen, als auch für ARM und x86 Architekturen in Kapitel 6. analysiert. Das 7. Kapitel fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick.