Antonio Carlos Schneider Beck Fl.
Luigi Carro

Dynamic Reconfigurable Architectures and Transparent Optimization Techniques

Automatic Acceleration of Software Execution

Springer

Dynamic Reconfigurable Architectures and Transparent Optimization Techniques

Antonio Carlos Schneider Beck Fl. • Luigi Carro

Dynamic Reconfigurable Architectures and Transparent Optimization Techniques

Automatic Acceleration of Software Execution

Springer

Prof. Antonio Carlos Schneider Beck Fl.
Instituto de Informática
Universidade Federal do Rio Grande
do Sul (UFRGS)
Caixa Postal 15064
Campus do Vale, Bloco IV
Porto Alegre
Brazil
caco@inf.ufrgs.br

Prof. Luigi Carro
Instituto de Informática
Universidade Federal do Rio Grande
do Sul (UFRGS)
Caixa Postal 15064
Campus do Vale, Bloco IV
Porto Alegre
Brazil
carro@inf.ufrgs.br

ISBN 978-90-481-3912-5 e-ISBN 978-90-481-3913-2
DOI 10.1007/978-90-481-3913-2
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2010921831

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Sabrina,
for her understanding and support
To Antônio and Léia,
for the continuous encouragement

To Ulisses, may his journey be full of joy
To Érika, for all our moments
To Cesare, Esther and Beti, for being there

Preface

As Moore's law is losing steam, one already sees the phenomenon of clock frequency reduction caused by the excessive power dissipation in general purpose processors. At the same time, embedded systems are getting more heterogeneous, characterized by a high diversity of computational models coexisting in a single device. Therefore, as innovative technologies that will completely or partially replace silicon are arising, new architectural alternatives are necessary.

Although reconfigurable computing has already shown to be a potential solution when it comes to accelerate specific code with a small power budget, significant speedups are achieved just in very dedicated dataflow oriented software, failing to capture the reality of nowadays complex heterogeneous systems. Moreover, one important characteristic of any new architecture is that it should be able to execute legacy code, since there has already been a large amount of investment into writing software for different applications. The wide spread usage of reconfigurable devices is still withheld by the need of special tools and compilers, which clearly preclude reuse of legacy code and its portability.

The authors have written this book with the aforementioned limitations in mind. Therefore, this book, which is divided in seven chapters, starts presenting the main challenges computer architectures are facing these days. Then, a detailed study on the usage of reconfigurable systems, their main principles, characteristics, potential and classifications is done. A separate chapter is dedicated to present several case studies, with a critical analysis on their main advantages and drawbacks, and the benchmarks used for their evaluation. This analysis will demonstrate that such architectures need to attack a diverse range of applications with very different behaviors, besides supporting code compatibility, that is, the need for no modification in the source or binary codes. This proves that more must be done to bring reconfigurable computing to be used as main stream computing: dynamic optimization techniques. Therefore, binary Translation and different types of reuse, with several examples, are evaluated. Finally, works that combine both reconfigurable systems and dynamic techniques are discussed, and a quantitative analysis of one of these examples is presented. The book ends with some directions that could inspire new fields of research.

The main purpose of this book is to introduce reconfigurable systems and dynamic optimization techniques to the readers, using several examples, so it can be a source of reference whenever the reader needs. The authors hope you enjoy it, as they have enjoyed making the research that resulted in this book.

Porto Alegre *Antonio Carlos Schneider Beck Fl.*
Luigi Carro

Acknowledgements

The authors would like to express their gratitude to the friends and colleagues at Instituto de Informatica of Universidade Federal do Rio Grande do Sul, and to give a special thanks to all the people in the Embedded Systems laboratory, who during several moments contributed for this research for many years.

The authors would also like to thank the Brazilian research support agencies, CAPES and CNPq.

Contents

Acronyms

ADPCM	Adaptive Differential Pulse-Code Modulation
ALU	Arithmetic Logic Unit
AMIL	Average Merged Instructions Length
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction Set Processor
ATR	Automatic Target Recognition
BB	Basic Block
BHB	Block History Buffer
BT	Binary Translator
CAD	Computer-Aided Design
CAM	Content Addressable Memory
CCA	Configurable Compute Accelerator
CCU	Custom Computing Unit
CDFG	Control Data Flow Graph
CISC	Complex Instruction Set Computer
CLB	Configurable Logic Block
CM	Configuration Manager
CMOS	Complementary MetalOxide Semiconductor
CMS	Code Morphing Software
CPII	Cycles Per Issue Interval
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
DADG	Data Address Generator
DAISY	Dynamically Architected Instruction Set from Yorktown
DCT	Discrete Cosine Transformation
DES	Data Encryption Standard
DFG	Data Flow Graph
DIM	Dynamic Instruction Merging
DLL	Dynamic-Link Library
DSP	Digital Signal Processing
DTM	Dynamic Trace Memoization

FFT	Fast Fourier Transform
FIFO	First In, First OutFirst In, First Out
FIR	Finite Impulse Response
FO4	Fanout-Of-Four
FPGA	Field-Programmable Gate Array
FU	Functional Unit
GCC	GNU Compiler Collection
GPP	General Purpose Processor
GSM	Global System for Mobile Communications
HDL	Hardware Description Language
I/O	Input-Output
IC	Integrated Circuit
IDCT	Inverse Discrete Cosine Transform
IDEA	International Data Encryption Algorithm
ILP	Instruction Level Parallelism
IPC	Instructions Per Cycle
IPII	Instructions Per Issue Interval
IR	Instruction Reuse
ISA	Instruction Set Architecture
ITRS	International Technology Roadmap for Semiconductors
JIT	Just-In-Time
JPEG	Joint Photographic Experts Group
LRU	Least Recently Used
LUT	Lookup Table
LVP	Load Value Prediction
MAC	multiplier-accumulator
MAC	Multiply Accumulate
MC	Motion Compensation
MIMD	Multiple Instruction, Multiple Data
MIN	Multistage Interconnection Network
MIR	Merged Instructions Rate
MMX	Multimedia Extensions
MP3	MPEG-1 Audio Layer 3
MPEG	Moving Picture Experts Group
NMI	Number of Merged Instructions
OFDM	Orthogonal frequency-division multiplexing
OPI	Operation per Instructions
OS	Operating System
PAC	Processing Array Cluster
PACT-XPP	eXtreme Processing Plataform
PAE	Processing Array Elements
PC	Program Counter
PCM	Pulse-Code Modulation
PDA	Personal Digital Assistant
PE	Processing Element

PFU	Programmable Functional Units
PRISM	Processor Reconfiguration through Instruction Set Metamorphosis
RAM	Random Access Memory
RAW	Read After Write
RAW	Reconfigurable Architecture Workstation
RB	Reuse Buffer
RC	Reconfigurable Cell
REMARC	Reconfigurable Multimedia Array Coprocessor
RFU	Reconfigurable Functional Unit
RISC	Reduced Instruction Set Computer
RISP	Reconfigurable Instruction Set Processor
ROM	Read Only Memory
RRA	Reconfigurable Arithmetic Array
RST	Reuse through Speculation on Traces
RT	Register Transfer
RTM	Reuse Trace Memory
RU	Reconfigurable Unit
SAD	Sum of Absolute Difference
SCM	Supervising Configuration Manager
SDRAM	Synchronous Dynamic Random Access Memory
SIMD	Single Instruction, Multiple Data
SMT	Simultaneous multithreading
SoC	System-On-a-Chip
SSE	Streaming SIMD Extensions
VHDL	VHSIC Hardware Description Language
VLIW	Very Long Instruction Word
VMM	Virtual Machine Monitor
VP	Value prediction
VPT	Value Prediction Table
WAR	Write After Read
WAW	Write After Write
XREG	Exchange Registers