

Hardware/Software Architectures for Low-Power Embedded Multimedia Systems

Muhammad Shafique • Jörg Henkel

Hardware/Software Architectures for Low-Power Embedded Multimedia Systems

 Springer

Muhammad Shafique
Karlsruhe Institute of Technology
Haid-und-Neu-Str. 7
76131 Karlsruhe
Deutschland
muhammad.shafique@kit.edu

Jörg Henkel
Karlsruhe Institute of Technology
Haid-und-Neu-Str. 7
76131 Karlsruhe
Deutschland

ISBN 978-1-4419-9691-6 e-ISBN 978-1-4419-9692-3
DOI 10.1007/978-1-4419-9692-3
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011931865

© Springer Science+Business Media, LLC 2011

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

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

Contents

1	Introduction	1
1.1	Trends and Requirements of Advanced Multimedia Systems	2
1.2	Trends and Options for Multimedia Processing	4
1.3	Summary of Challenges and Issues	9
1.4	Contribution of this Monograph	9
1.5	Monograph Outline	13
2	Background and Related Work	15
2.1	Video Coding: Basics and Terminology	15
2.2	The H.264 Advanced Video Codec: A Low-power Perspective	17
2.2.1	Overview of the H.264 Video Encoder and Its Functional Blocks	17
2.2.2	Low-Power Architectures for H.264/AVC Video Encoder	22
2.2.3	Adaptive and Low-Power Design of the Key Functional Blocks of the H.264 Video Encoder: State-of-the-Art and Their Limitations	24
2.3	Reconfigurable Processors	28
2.3.1	Fine-Grained Reconfigurable Fabric	29
2.3.2	Leakage Power of Fine-grained Reconfigurable Fabric and the Power-Shutdown Infrastructure	30
2.3.3	Custom Instructions (CIs): A Reconfigurable Processor Perspective	32
2.3.4	Reconfigurable Instruction Set Processors	33
2.3.5	Rotating Instruction Set Processing Platform (RISPP)	35
2.4	Low-Power Approaches in Reconfigurable Processors	43
2.5	Summary of Related Work	45

3 Adaptive Low-Power Architectures for Embedded Multimedia Systems	49
3.1 Analyzing the Video Coding Application for Energy Consumption and Adaptivity	49
3.1.1 Advanced Video Codecs: Analyzing the Tool Set	51
3.1.2 Energy and Adaptivity Related Issues in H.264/AVC Video Encoder	53
3.2 Energy- and Adaptivity Related Issues for Dynamically Reconfigurable Processors	56
3.3 Overview of the Proposed Architectures and Design Steps	59
3.4 Power Model for Dynamically Reconfigurable Processors	63
3.4.1 Power Consuming Parts of a Computation- and Communication-Infrastructure in a Dynamically Reconfigurable Processor	63
3.4.2 The Proposed Power Model	65
3.5 Summary of Adaptive Low-Power Embedded Multimedia System	67
4 Adaptive Low-Power Video Coding	69
4.1 H.264 Encoder Application Architectural Adaptations for Reconfigurable Processors	69
4.1.1 Basic Application Architectural Adaptations	69
4.1.2 Application Architectural Adaptations for On-Demand Interpolation	72
4.1.3 Application Architectural Adaptations for Reducing the Hardware Pressure	75
4.1.4 Data Flow of the H.264 Encoder Application Architecture with Reduced Hardware Pressure	77
4.2 Designing Low-Power Data Paths and Custom Instructions	80
4.2.1 Designing the Custom Instruction for In-Loop Deblocking Filter	81
4.2.2 Designing the Custom Instructions for Motion Estimation	85
4.2.3 Designing the Custom Instruction for Motion Compensation	86
4.2.4 Area Results for the Custom Instructions of H.264 Encoder	87
4.3 Spatial and Temporal Analysis of Videos Considering Human Visual System	87
4.3.1 HVS-based Macroblock Categorization	92
4.3.2 QP-based Thresholding	93
4.3.3 Summary of Spatial and Temporal Analysis of Videos Considering Human Visual System	95
4.4 An HVS-Based Adaptive Complexity Reduction Scheme	95
4.4.1 Prognostic Early Mode Exclusion	97

- 4.4.2 Hierarchical Fast Mode Prediction 99
- 4.4.3 Sequential RDO Mode Elimination 100
- 4.4.4 Evaluation of the Complexity Reduction Scheme 100
- 4.5 Energy-Aware Motion Estimation with an Integrated Energy-Budgeting Scheme 104
 - 4.5.1 Adaptive Motion Estimator with Multiple Processing Stages 105
 - 4.5.2 enBudget: The Adaptive Predictive Energy-budgeting Scheme 111
 - 4.5.3 Evaluation of Energy-Aware Motion Estimation with an Integrated Energy-Budgeting Scheme 117
 - 4.5.4 Comparing Adaptive Motion Estimator with and Without the enBudget Scheme 118
 - 4.5.5 Comparing UMHexagonS with and Without the enBudget Scheme 118
- 4.6 Summary of Low-power Application Architecture 121
- 5 Adaptive Low-power Reconfigurable Processor Architecture 123**
 - 5.1 Motivational Scenario and Problem Identification 123
 - 5.1.1 Summary of the Motivational Scenario and Problem Identification 126
 - 5.2 Run-time Adaptive Energy Management with the Novel Concept of Custom Instruction Set Muting 126
 - 5.2.1 Concept of Muting the Custom Instructions 127
 - 5.2.2 Power-shutdown Infrastructure for the Muted Custom Instructions 128
 - 5.2.3 Run-time Adaptive Energy Management 130
 - 5.2.4 Summary of the Run-time Adaptive Energy Management and CI Muting 132
 - 5.3 Determining an Energy-minimizing Instruction Set 133
 - 5.3.1 Formal Problem Modeling and Energy Benefit Function 133
 - 5.3.2 Algorithm for Choosing CI Implementation Versions 135
 - 5.3.3 Evaluation and Results for Energy-Minimizing Instruction Set 139
 - 5.3.4 Summary of Energy Minimizing Instruction Set 145
 - 5.4 Selective Instruction Set Muting 146
 - 5.4.1 Problem Description and Motivational Scenarios 147
 - 5.4.2 Operational Flow for Selective Instruction Set Muting 148
 - 5.4.3 Analyzing the Energy Benefit Function of Muting 150
 - 5.4.4 Hot Spot Requirement Prediction: Computing Weighting Factors for CIs 152
 - 5.4.5 Evaluation of Selective Instruction Set Muting 153
 - 5.3.6 Summary of Selective Instruction Set Muting 154
 - 5.5 Summary of Adaptive Low-power Reconfigurable Processor Architecture 155

- 6 Power Measurement of the Reconfigurable Processors** 157
 - 6.1 Power Measurement Setup 157
 - 6.2 Measuring the Power of Custom Instructions 158
 - 6.2.1 Flow for Creating the Power Model 158
 - 6.2.2 Test Cases for Power Measurements 160
 - 6.2.3 Results for Power Measurement and Estimation 162
 - 6.3 Measuring the Power of the Reconfiguration Process 164
 - 6.3.1 Power Consumption of EEPROM 165
 - 6.3.2 Power Consumption of the Reconfiguration via ICAP 165
 - 6.4 Summary of the Power Measurement of the Reconfigurable Processors 166

- 7 Benchmarks and Results** 167
 - 7.1 Simulation Conditions and Fairness of the Comparison 168
 - 7.2 Adaptive Low-power Application Architecture 169
 - 7.2.1 Comparing Complexity Reduction Scheme to State-of-the-art and the Exhaustive RDO-MD 169
 - 7.2.2 Comparing the Energy-Aware Motion Estimation with Integrated Energy Budgeting Scheme to State-of-the-art 173
 - 7.3 Adaptive Low-power Processor Architecture 175
 - 7.3.1 Comparing the Adaptive Energy Management Scheme (Without Selective Instruction Set Muting) to RISPP with Performance Maximization [BSH08c] 175
 - 7.3.2 Applying the Adaptive Energy Management Scheme (Without Selective Instruction Set Muting) to Molen [VWG+04] Reconfigurable Processor 176
 - 7.3.3 Comparing the Adaptive Energy Management Scheme (with Selective Instruction Set Muting) to State-of-the-art Hardware-oriented Shutdown 177
 - 7.4 Summary of the Benchmarks and Comparisons 180

- 8 Conclusion and Outlook** 183
 - 8.1 Monograph Summary 183
 - 8.2 Future Work 187

- Appendix** 191

- Bibliography** 211

- Index** 221

Abbreviations and Definitions

ACCoReS	Adaptive Computational Complexity Reduction Scheme
ADI	Arbitrary Directional Intra
ALU	Arithmetic Logic Unit
ASIC	Application Specific Integrated Circuit
ASIP	Application Specific Instruction Set Processor
AVC	Advanced Video Coding
AVS	Advanced Visual Systems
B-MB	Bi-directionally predicted → MB (i.e., a prediction is performed from the previous and future reference frames)
BC	Bus Connector: Connecting a → DPC to the Computation and Communication Infrastructure
BOPF	Buffer Overflow Prevention Factor
BU	Basic Unit, it is a group of → MBs; it defines the granularity at which the rate controller computes a new QP value
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CBR	Constant Bit Rate
CI	Custom Instruction
CIF	Common Intermediate Format (Resolution: 352×288)
CIP	Combined Intra Prediction
cISA	core Instruction Set Architecture: the part of the instruction set that is implemented using the core processor pipeline (i.e., non-reconfigurable); it can be used to implement → Custom Instructions
CLB	Configurable Logic Block: part of an → FPGA, contains multiple → LUTs
CPU	Central Processing Unit
DCSS	Dynamic Clock Supply Stop
DCT	Discrete Cosine Transform
DPC	Data Path Container: a part of the reconfigurable fabric that can be dynamically reconfigured to contain a Data Path, i.e., an elementary hardware accelerator

DVFS	Dynamic Voltage and Frequency Scaling
EAPR	Early Access Partial Reconfiguration
EE	Encoding Engine
EEPROM	Electrically Erasable Programmable Read Only Memory
enBudget	The run-time adaptive Energy Budgeting Scheme
EPZS	Enhanced Predictive Zonal Search
EQ	Energy-Quality
FI	Forecast Instruction: a trigger instruction that indicates a Forecast Block containing a set of \rightarrow CIs with an information of the compile-time analysis (e.g., expected number of executions)
FIFO	First-In First-Out buffer
FIR	Finite Impulse Response
FME	Fractional-pixel Motion Estimation
FMO	Flexible Macroblock Ordering
FM-CI	Fully-Muted Custom Instruction
FPGA	Field Programmable Gate Array: a reconfigurable device that is composed as an array of \rightarrow CLBs and switching matrices
FPS	Frames Per Second
FS	Full Search
GOP	Group of Pictures with one I-Frame followed by a series of P- and/or B-Frames
GPP	General Purpose Processor
HDTV	High Definition Television
HD720p	High Definition 720 Lines Progressive Scan (Resolution: 1280×720)
HEVC	High Efficiency Video Coding
HT	Hadamard Transform
HVS	Human Visual System
I-MB	Intra-predicted \rightarrow MB (i.e., a prediction is performed from the reconstructed pixels of \rightarrow MBs from the current frame; it is also called <i>spatial</i> prediction)
$I4 \times 4$	Macroblock is encoded as Intra with prediction is done at 4×4 block sizes
$I16 \times 16$	Macroblock is encoded as Intra where the whole 16×16 block is predicted
ICAP	Internal Configuration Access Port
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
IHT	Inverse Hadamard Transform
ILP	Integer Linear Programming
IME	Integer-pixel Motion Estimation
IP	Intellectual Property
IPred	Intra Prediction
IQ	Inverse Quantization
ISA	Instruction Set Architecture
ISO	International Organization for Standardization

ISS	Instruction Set Simulator
ITU	International Telecommunication Union
JVT	Joint Video Team
KB	Kilo Byte (also KByte): 1024 Byte
KD	Derivative Gain
KI	Integral Gain
KP	Proportional Gain
LF	Loop Filter
LUT	Look-Up Table: smallest element in an \rightarrow FPGA, part of a \rightarrow CLB; configurable as logic or memory
MAD	Mean of Absolute Differences
MB	Mega Byte (also MByte): 1024 \rightarrow KB
MB	Macroblock, a 16×16 pixel block of a video frame
MBEE	Mean Bit Estimation Error
MC	Motion Compensation
MD	Mode Decision
ME	Motion Estimation
MIPS	Microprocessor without Interlocked Pipeline Stages
MPEG	Motion Picture Experts Group
MPSoC	Multiprocessor System-on-Chip
MSE	Mean Square Error
MV	Motion Vector
MVC	Multiview Video Coding
NM-CI	Non-Muted Custom Instruction
NMOS	N-type Metal-Oxide-Semiconductor Logic
P-MB	Inter-predicted \rightarrow MB (i.e., a prediction is performed from the reconstructed pixels of \rightarrow MBs from the previous frame; it is also called <i>temporal</i> prediction)
$P8 \times 8$	Macroblock is encoded as Inter with sub-block types sizes of 8×8 or below
$P16 \times 16$	Macroblock is encoded as Inter where the whole 16×16 block is predicted
PC	Personal Computer
PID	Proportional-Integral-Derivative
PMOS	P-type Metal-Oxide-Semiconductor Logic
PSM	Programmable Switching Matrix
PSNR	Peak signal-to-noise ratio (units: db)
Q	Quantization
QCIF	Quarter Common Intermediate Format (Resolution: 176×144)
QP	Quantization Parameter
RAM	Random Access Memory
RC	Rate Controller
RD	Rate Distortion
RDO	Rate Distortion Optimization
REMiS	Run-time Adaptive Energy Minimization Scheme

RFU	Reconfigurable Functional Unit: denotes a reconfigurable region that can be reconfigured towards a Custom Instruction implementation
RISPP	Rotating Instruction Set Processing Platform
SAD	Sum of Absolute Differences
SATD	Sum of Absolute Transformed Differences
SI	Special Instruction
SIF	Source Input Format (Resolution: 352×240)
SPARC	Scalable Processor Architecture: processor family from Sun Microsystems; used for the \rightarrow RISPP prototype
SQCIF	Sub-quarter Common Intermediate Format (Resolution: 128×96)
SRAM	Static Random Access Memory
SSE	Sum of Squared Differences
TH	Threshold
UMHexagonS	Unsymmetrical-cross Multi-Hexagon-grid Search
VBR	Variable Bit Rate
VCEG	Video Coding Experts Group
VISA	Virtual Instrument Software Architecture
VLC	Variable Length Coding: a computational kernel that is used in H-264 video encoder
VLIW	Very Large Instruction Word
VM-CI	Virtually-Muted Custom Instruction
XML	Extensible Markup Language
XST	Xilinx Synthesis Technology
YUV	A video format denoting one Luminance (Luma, Y) and two Chrominance (Chroma, UV) Components. A typical resolution given to video encoders is YUV4:20:0, i.e., a sampling method where the two chrominance components have just half the resolution in vertical and horizontal direction as the luminance component

Definitions

Level	define a constraint on key parameters, e.g., specific resolutions and bit rates.
Profile	defines a set of coding tools and algorithms, targeting a specific class of applications.
Residual	Difference of current data to the corresponding prediction data.
Slice	A frame is build up of a number of slices, each containing an integral number of MBs.

List of Figures

Fig. 1.1	Overview of different video services over time	3
Fig. 1.2	a Flexibility vs. efficiency comparison of different architectural options; b evolution trend of Xilinx Virtex FPGAs	5
Fig. 2.1	An overview of the digital video structure (showing group of pictures, frame, slice, MB) and different video resolutions	16
Fig. 2.2	Functional overview of the H.264/AVC video encoder	18
Fig. 2.3	Variable block sizes for inter-predicted MBs (P-MBs) in H.264/AVC	18
Fig. 2.4	A typical composition of a fine-grained reconfigurable fabric with a 2D-Array of CLBs and PSMs along with the internal details of a Spartan-3 Tile	29
Fig. 2.5	State-of-the-art in power-shutdown infrastructure	31
Fig. 2.6	Extending a standard processor pipeline towards RISPP and the overview of the RISPP run-time system	35
Fig. 2.7	Hierarchical composition of custom instructions: multiple implementation versions exist per custom instruction and demand data paths for realization	36
Fig. 2.8	Example control-flow graph showing forecasts and the corresponding custom instruction executions	40
Fig. 2.9	Execution sequence of forecast and custom instructions with the resulting error back propagation and fine-tuning	41
Fig. 2.10	Overview of the hardware infrastructure for computation (data path container) and communication (bus connector) showing the internal composition of a bus connector	42
Fig. 3.1	Overview of an H.324 video conferencing application with H.264/AVC codec	50
Fig. 3.2	Processing time distribution of different functional blocks in the H.324 video conferencing application	51
Fig. 3.3	Percentage distribution of energy consumption of different functional blocks in the H.264 video encoder	54

Fig. 3.4 Distribution of I-MBs in slow-to-very-high motion scenes (test conditions: group of pictures=IPPP..., CAVLC, quantization parameter=28, 30fps) 57

Fig. 3.5 Overview of the adaptive low-power application and processor architectures 59

Fig. 3.6 Highlighting different steps to be performed at design, compile, and run time at both application and processor levels 61

Fig. 3.7 Power-relevant components of the computation- and communication infrastructure to execute CI implementation versions 64

Fig. 3.8 Example for a custom instruction (CI) implementation version 64

Fig. 4.1 Basic application architectural adaptations to construct the benchmark application. **a** Adapting reference software. **b** Improving data structure. **c** Profiling and Designing Custom Instructions 70

Fig. 4.2 Arrangement of functional blocks in the H.264 encoder benchmark application 71

Fig. 4.3 Number of computed vs. required interpolated MBs for two standard test sequences for mobile devices 72

Fig. 4.4 Distribution of different interpolation cases in the carphone video sequence 73

Fig. 4.5 H.264 encoder application architecture with reduced hardware pressure 74

Fig. 4.6 Data flow diagram of the H.264 encoder application architecture with reduced hardware pressure 78

Fig. 4.7 Description and organization of major data structures 79

Fig. 4.8 Steps to create optimized data paths from the standard formulae 81

Fig. 4.9 4-Pixel edges in one macroblock 82

Fig. 4.10 Pixel samples across a 4×4 block horizontal or vertical boundary 82

Fig. 4.11 Custom instruction for in-loop deblocking filter with example schedule 83

Fig. 4.12 The data paths for filtering conditions and filtering operation constituting the for custom instruction for in-loop deblocking filter 84

Fig. 4.13 Custom instruction for SATD4×4 showing the transform and SAV data paths 85

Fig. 4.14 Custom instruction for motion compensation showing different data paths 86

Fig. 4.15 Mode distribution and video statistics in the 7th frame of American Football 89

Fig. 4.16 Optimal coding mode distribution in rafting and American Football sequences at different Quantization Parameter (QP) values 90

Fig. 4.17	Directional groups with respect to the edge direction angle and notion of spatial and temporal neighboring macroblocks	91
Fig. 4.18	Mode distribution of frame 4 in Rafting sequence using the exhaustive RDO-MD for two different QP values: <i>Left</i> : QP=16 and <i>Right</i> : QP=38	94
Fig. 4.19	Overview of the adaptive computational complexity reduction scheme (ACCoReS) showing different processing steps and MB categorizations	96
Fig. 4.20	Processing flow of the hierarchical fast mode prediction	99
Fig. 4.21	Percentage mode excluded in ACCoReS for various video sequences	101
Fig. 4.22	Distribution of mode processing for QP=28	101
Fig. 4.23	Comparison of total SAD computations for various video sequences	102
Fig. 4.24	Frame-level in-depth comparison for Susie sequence	102
Fig. 4.25	Frame-level in-depth evaluation of correct mode prediction	103
Fig. 4.26	MB-level mode comparison with the exhaustive RDO-MD: frame 17 of American Football. <i>Top</i> : ACCoReS [PSNR=33.28 dB], <i>Bottom</i> : Exhaustive RDO-MD [PSNR=34.52 dB]	103
Fig. 4.27	Motion vector difference distribution in Foreman sequence (256 kbps) for various predictors compared to the optimal motion vector (obtained using the full search algorithm)	106
Fig. 4.28	Predictor conditions for motion-dependent early termination	107
Fig. 4.29	Four search patterns used in the adaptive motion estimator and the pixel-decimation patterns for SAD computation	108
Fig. 4.30	Flow of the enBudget scheme for energy-aware motion estimation	111
Fig. 4.31	Energy-Quality (EQ) classes: energy-quality design space exploration showing various pareto points and the pareto curve	113
Fig. 4.32	SAD vs. energy consumption comparison of different motion estimation stages for Foreman sequence	114
Fig. 4.33	Energy and quality comparison for the adaptive motion estimator with and without the enBudget for various video sequences	118
Fig. 4.34	Energy and quality comparison for the UMHexagonS [CZH02] with and without the enBudget for various video sequences	118
Fig. 4.35	Frame-wise energy consumption of the energy-aware motion estimation	119
Fig. 4.36	Macroblock-wise energy consumption map of two exemplary frames in the SusieTableMix_QCIF sequence for a 90 nm technology	120

Fig. 4.37	Energy consumption of the energy-aware motion estimation for various FPGA fabrication technologies for various video sequences	120
Fig. 5.1	Simplified comparison of energy consumption, highlighting the effects of different reconfiguration decisions	125
Fig. 5.2	Infrastructure necessary to exert the proposed CI muting technique	129
Fig. 5.3	Muting the temporarily unused instruction set	129
Fig. 5.4	Overview of the proposed adaptive low-power reconfigurable processor with run-time adaptive energy management along with the design-, compile-, and run-time steps	130
Fig. 5.5	Search space of five CIs with their implementation versions at the corresponding levels and the path of the energy-minimizing instruction set	136
Fig. 5.6	Energy-performance design spaces: evaluation of the energy minimization space using the adaptive energy management scheme under various area and performance constraints for four fabrication technologies for an encoding of 40 QCIF (176×144) frames	141
Fig. 5.7	Comparison of energy components in different fabrication technologies under various area constraints	142
Fig. 5.8	Comparing energy-performance design spaces for different video resolutions when using the energy management scheme under various area and performance constraints for an encoding of 60 video frames	143
Fig. 5.9	CI Execution results for 30 fps on 65 nm showing a detailed breakdown of energy components highlighting the contribution of reconfiguration and leakage energy. The lower graph shows the detailed execution pattern of various CIs executing in different hot spots of the H.264 video encoder along with total energy consumption	144
Fig. 5.10	Comparing the energy requirements of virtually- & fully-muted CIs for two scenarios	147
Fig. 5.11	Time-line showing the execution sequence of hot spots and the situation for a CI muting decision	148
Fig. 5.12	Flow for selecting a muting mode for the custom instruction (CI) set	149
Fig. 5.13	Venn diagram showing the data path requirements of previous, current, upcoming hot spots	149
Fig. 5.14	Calculating the weighting factor for custom instructions w.r.t. the application context	153
Fig. 5.15	Summary of energy benefit of using selective instruction set muting	154
Fig. 6.1	a Measurement setup, b The in-house developed power supply board	158

Fig. 6.2	Flow for creating the measurement-based power model	159
Fig. 6.3	Test case and setup for measuring the power of an idle (empty) framework	159
Fig. 6.4	Different test cases for measuring the power of different components of a custom instruction (CI) implementation version	160
Fig. 6.5	Connection of FIFO between EEPROM and ICAP	164
Fig. 6.6	a EEPROM voltage drop while loading one Data Path Bitstream from EEPROM to FPGA. b VCC_{INT} voltage drop for transferring one Data Path bitstream to ICAP and performing the corresponding reconfiguration	165
Fig. 7.1	Comparing the energy savings and quality loss of the ACCoReS with several state-of-the-art fast mode decision schemes	170
Fig. 7.2	Energy savings and quality loss of the ACCoReS compared to the exhaustive RDO-MD for CIF resolution video sequences	170
Fig. 7.3	Energy savings and quality loss of the ACCoReS compared to the exhaustive RDO-MD for QCIF resolution video sequences	171
Fig. 7.4	Comparing the rate distortion curves for QCIF and CIF sequences	172
Fig. 7.5	Power test with a real battery using Mobile sequence	172
Fig. 7.6	Summary of energy savings of the enBudget scheme compared to various fast adaptive motion estimation schemes	173
Fig. 7.7	Comparing energy saving and PSNR loss of the proposed energy-aware motion estimation and the enBudget scheme with various fast adaptive motion estimators. (* negative PSNR loss actually shows the PSNR gain of the scheme)	174
Fig. 7.8	Energy comparison of the AEM_FM and RISPP_PerfMax schemes for 65 nm	175
Fig. 7.9	Average energy comparison of the AEM_FM and RISPP_PerfMax for three technologies	176
Fig. 7.10	Percentage energy saving of Molen [VWG ⁺ 04] plus AEM_FM over Molen without AEM_FM for three technologies	177
Fig. 7.11	Comparing the energy breakdown of the adaptive energy management scheme (with selective instruction set muting) to [Ge04]-based pre-VM and [MM05]-based pre-FM	178
Fig. 7.12	Energy comparison of the adaptive energy management scheme with [Ge04]-based pre-VM and [MM05]-based pre-FM techniques for varying amount of reconfigurable fabric	179
Fig. 7.13	Energy savings of the adaptive energy management scheme compared to the [Ge04]-based pre-VM technique	180
Fig. A.1	Comparison of produced bits with and without rate control	192

Fig. A.2 The multi-level rate control scheme covering GOP, frame/slice, & BU levels along with image and motion based macroblock prioritization 195

Fig. A.3 Critical Ziegler-Nichols-point for American Football 196

Fig. A.4 Temporal distance based QP calculation for B frames/slices 196

Fig. A.5 Basic unit (BU) level RC with texture and motion based QP adjustments 198

Fig. A.6 RD-curves comparison of the proposed multi-level RC with RC-mode-3 for carphone (QCIF, IPPP) and American Football (SIF, IBBP) 199

Fig. A.7 MBEE comparison of the multi-level RC with three different RC modes 200

Fig. A.8 Frame-wise comparison of the multi-level RC with RC-mode-3 for fast motion combined CIF sequences encoded at 2 Mbps@30 fps 201

Fig. A.9 Frame-wise comparison of the multi-level RC with RC-mode-0 for Susie mixed CIF sequence (*Bright, Dark, Noisy*) at 2 Mbps@30 fps 201

Fig. A.10 Evaluating the image and motion based MB prioritizations (Note: All excerpts are 2× zoomed using nearest neighbor interpolation) 202

Fig. B.1 Simulation methodology showing various steps of the simulation procedure 203

Fig. B.2 Reconfigurable processor simulator with the extensions implemented in the scope of this monograph for run-time adaptive energy-management 204

Fig. B.3 **a** H.264 video encoder executing on the RISPP prototype; **b** Floorplan of the RISPP prototype implementation on the Xilinx Virtex-4 LX 160 FPGA 206

Fig. B.4 H.264 video encoder executing on the TI' DM6437 DSP board 207

Fig. B.5 Flow for porting H.264 Encoder on DM6437 digital signal processor 208

Fig. C.1 The CES video analyzer tool showing the research framework for motion estimation, video merging, and texture analysis 209

List of Tables

Table 2.1	High-level properties of implementation version and custom instruction	39
Table 3.1	Comparing the coding tool set of various video encoding standards	52
Table 4.1	Custom instructions and data paths for the H.264 video encoder	76
Table 4.2	Implementation results for various data paths of the H.264 video encoder	87
Table 4.3	Thresholds and multiplying factors used in ACCoReS	94
Table 4.4	Summary of PSNR, bit rate, and speedup comparison for various video sequences (each encoded using eight different QPs)	101
Table 4.5	Comparing the video quality of different SAD decimation patterns for encoding of Susie CIF video sequence (30fps@256 kbps)	110
Table 4.6	Configuration and energy consumption for the chosen Energy-Quality (EQ) classes	114
Table 4.7	Coefficients and thresholds used by the algorithm of enBudget in Algorithm 4.4	117
Table 4.8	Performance, area, and energy overhead of enBudget	121
Table 5.1	Various custom instruction (CI) muting modes	128
Table 5.2	Parameters and evaluation conditions with their corresponding reference sources	140
Table 5.3	Hardware implementation results for the energy management scheme on the RISPP prototyping platform (see Fig. 6.1 in Sect. 6.1)	145
Table 6.1	Different placement combinations of two transform Data Paths for power measurement	162
Table 6.2	Measured power results for various data paths & HT4 × 4 implementation versions	162

Table 6.3	Parameters of power model for the CI implementation versions	162
Table 6.4	Power consumption and latencies of different implementation versions (using different amount of DPCs) for various custom instructions for 65 nm and 40 nm technologies	163

List of Algorithms

Algorithm 4.1	The Filtering Process for Boundary Strength = 4	82
Algorithm 4.2	Pseudo-Code of Group-A for Prognostic Early Mode Exclusion	97
Algorithm 4.3	Pseudo-Code of Group-B for Prognostic Early Mode Exclusion	98
Algorithm 4.4	Pseudo code of the Run-Time Adaptive Predictive Energy-Budgeting Scheme	116
Algorithm 5.1	Pseudo code of Determining the Energy Minimizing Instruction Set	137
Algorithm 5.2	Pseudo Code for Finding a Data Path for Virtually-Muting Mode	151