

ENERGY OPTIMIZATIONS FOR SCRATCH PAD MEMORY BASED SIMD ARCHITECTURES

NAMITA SHARMA



AMAR NATH AND SHASHI KHOSLA SCHOOL OF IT
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI

APRIL 2015

©Indian Institute of Technology Delhi (IITD), New Delhi, 2015

ENERGY OPTIMIZATIONS FOR SCRATCH PAD MEMORY BASED SIMD ARCHITECTURES

by

NAMITA SHARMA

Amar Nath and Shashi Khosla School of IT
Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of
Doctor of Philosophy

to the



Indian Institute of Technology Delhi

April 2015

Certificate

This is to certify that the thesis titled **Energy Optimizations for Scratch Pad Memory based SIMD Architectures** being submitted by **Ms. Namita Sharma** for the award of **Doctor of Philosophy** in Amar Nath and Shashi Khosla School of IT is a record of bona-fide work carried out by her under my guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Preeti Ranjan Panda

Professor

Department of Computer Science and Engineering

Indian Institute of Technology Delhi

New Delhi - 110 016

Acknowledgments

The satisfaction that accompanies the successful completion of this thesis would be incomplete without making mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts. First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Preeti Ranjan Panda, Department of Computer Science and Engineering, IIT Delhi for providing me an opportunity to work under his precious guidance. It is my fortune to work with Prof. Panda who has been enthusiastically guiding me at every stage of this work and encouraging me to achieve the targets set well in time. Working with him was a great pleasure and vast learning experience. His critical comments during the discussions enabled me to overcome my weaknesses and make valuable improvements. Thanks for your belief in my abilities and to direct my passion towards the right direction.

It is my fortune to have my thesis work in collaboration with IMEC, Belgium. Thanks to Prof. Francky Catthoor for agreeing to serve as my co-supervisor at IMEC. I really admire him for his patience and long future insight he has. I would like to thank him for keeping timely meetings that made this distant collaboration fruitful. I would also like to extend my thanks to my colleagues – Tom Vander Aa, Praveen Raghavan, Eddy De Greef, Amir Amin and Min Li whose timely assistance greatly helped in developing a deep understanding of the project. Special thanks to my PhD friends Prashant Agrawal, Matthias Hartmann, Raf Appeltans, Robert Fasthuber, and Halil Kukner for friendly discussions and support that made my stay at IMEC memorable.

My sincere thanks to Professors Anshul Kumar, G.S. Visweswaran and Vinay Ribeiro for serving on my Student Research Committee. Thanks to my labmates – Vaibhav Jain, Sharat Varma, G. Krishnaiah, Swanti Satsangi, Rajshekhar Kalayappan, Sandeep Chandran, and Prathmesh Kallurkar for their great company. I would also like to thank S.D. Sharma and Vandana Ahluwalia for their timely assistance in lab related issues.

Finally, I dedicate this thesis to my beloved parents Gajendra Sharma and Uma Sharma for their blessings, support, and persistence which has always been a source of inspiration that kept me going to complete this venture. Special thanks to my mom and brother Hardik Sharma who always encouraged me and shared my tough and sweet moments without any hesitation.

Namita Sharma

Abstract

Mobile communication devices have evolved tremendously in the last few decades. These devices are not bound to an application but are useful for a wide range of functionalities. For example, the modern smartphones provide not only the calling facility but also multiple other functionalities such as 3D gaming, Internet access, Bluetooth, and so on. These evolutions require higher data rates, and low latencies. To achieve these features, techniques such as OFDM and MIMO are being introduced leading to co-existence of multiple wireless standards. Separate standards exist for supporting varying connectivity ranges. Relying on hardware implementation to accommodate these features on mobile devices leads to design challenges such as higher area for multiple hardware units, costlier implementation, longer time to market and so on. To overcome these challenges, having the entire baseband running on a programmable architecture is an attractive alternative. This is possible if we have the wireless physical layer functions software defined and run on a re-configurable architecture. This is popularly known as Software Defined Radio (SDR) implementation. With SDR architectures, we overcome the drawbacks of custom hardware implementations by bringing down the implementation cost, reducing the size as resource sharing is applicable, lowering the time-to-market as standard revisions simply require software upgradation rather than building the customized hardware. SDR thus proves to be an attractive alternative as it enables the reuse of design efforts across generations.

Battery life limits the utility of handheld devices such as mobile phones. With more and more applications and features enabled in modern devices the energy consumption is increasing. Current smart phones suffer a major drawback of short battery life, with average lasting time of less than a day. This poses a need to lower the energy consumption so as to increase the battery life. The lower the energy consumption, the longer the battery life will be. Bigger batteries are not a good solution for this problem as the portability of handheld devices decreases with increasing device sizes and weights. Thus, there is a need to have energy efficient implementations for applications running on such architectures. Memory plays a dominating role in the system design leading to significant amount of research in data and memory optimizations. Optimizations related to memory accesses and data storage make a significant difference to the performance and energy of a wide range of data-intensive applications. Such strategies need to

evolve with modern SoC and processor architectures, which lead to new optimization opportunities. In this thesis, we propose some strategies that significantly reduce the memory accesses thereby reducing the overall implementation energy. The proposed strategies are targeted at embedded processor systems with features such as scratch pad memories (SPM), Single Instruction Multiple Data (SIMD) FUs, and vector register files with wide interfaces to both SPM and FUs. The main contributions of this thesis are:

- We study the inter- and intra-kernel data dependencies for the LTE MIMO wireless standard and propose and compare efficient data layouts for the application. Our focus is on the LTE downlink receiver as this is a data- and computation-intensive part of the LTE application with tight energy and latency constraints.
- Array Interleaving, the proposed data layout transformation for the LTE application, is further developed to make it a more broadly applicable compiler transformation strategy for SIMD architectures. Based on the estimations of the benefits and interleaving costs, we perform a global analysis of arrays accesses spanning multiple loops, and take interleaving decisions that are predicted to be globally optimal. We also incorporate the effect of interleaving granularity in our exploration.
- We propose an energy efficient data flow transformation for Givens Rotation based QR Decomposition, a widely used function for matrix inversion and triangularization, for mapping to SIMD architectures. The proposed sequencing strategy is compared with the conventional sequences for matrices of different sizes. We also explore different possible implementations for QRD of multiple matrices using the SIMD feature of the processor.
- We propose a strategy to select an energy optimal Register File (RF) size for FFT processing on input data having large percentage of zeros. FFT is widely used in signal processing and image processing domains to transform the inputs in the time domain to the frequency domain for simplifying the computations. The strategy, based on memory access and compute count estimates, results in an energy efficient RF configuration out of the large number of configurations possible with the variation in number of registers and the SIMD widths.

With the proposed data layout strategy – Array Interleaving, for LTE application, a reduction of 7-15% in memory energy consumption is obtained over a highly hand-optimized code. Through an exploration for inter- and intra- kernel data dependencies in the application we conclude that there exist no dependencies across the Physical Resource Blocks (PRBs) allowing the merging of some kernels in the data processing block, thereby reducing the overall memory access count by around 15%. Experimental evaluation of the proposed layout strategy on

several applications in the wireless communication, multimedia, and image processing domain showed a significant reduction (6% – 34%) in memory energy consumption.

Our proposed data flow transformation strategy for QR Decomposition results in up to 36% reduction in overall energy across different implementations. Up to 75% reduction in memory accesses is achieved over the conventional sequences. A comparison of these sequences with standard tiling transformation shows that tiling results in an energy efficient implementation with respect to conventional sequences but has higher energy consumption compared to the proposed sequencing.

Using our proposed exploration strategy for RF size selection for FFT, we obtain a reduction of up to 59% in data memory energy. This percentage variation is obtained when comparing over the range of possible RF configurations with varying number of registers and SIMD widths.

Contents

Certificate	i
Acknowledgments	iii
Abstract	v
List of Figures	xiii
List of Tables	xvii
1 Introduction and Motivation	1
1.1 Why SDR Architectures?	1
1.2 Energy efficient implementation on SDRs	3
1.3 Application Domain	4
1.4 Motivation for Data Layout Transformation	5
1.5 Energy efficient sequencing strategy for Givens Rotation based QRD	7
1.6 Effect of RF Size on FFT processing	10
1.7 Contributions	12
1.8 Thesis Outline	13
2 Background	15
2.1 Embedded System	15
2.1.1 Components of an Embedded System	16
2.1.2 Pipelining and Parallelization Schemes	18
2.2 Related Architectures	21
2.3 Hardware Based Memory Energy Optimization	24
2.3.1 Register File Context	25
2.3.2 Memory Modeling and Customization	27
2.4 Software Based Memory Energy Optimization	30
2.4.1 Register Allocation	30

2.4.2	Code Transformations	31
2.4.3	Data Layout	34
3	Long Term Evolution Wireless Standard	37
3.1	Related Work	38
3.2	LTE Basics	39
3.2.1	A Time Domain View	39
3.2.2	A Frequency Domain View	40
3.3	Applications Overview	41
3.3.1	LTE 2x2 Application	41
3.3.2	LTE 4x4 Application	42
3.4	Exploration for Interleaving Decision	45
3.4.1	Data Dependence Analysis of LTE2x2	45
3.4.2	Array Interleaving	48
3.4.3	Impact of Interleaving	49
3.4.4	Global Trade-off Analysis and Overall Strategy	52
3.4.5	Memory Access Estimates	53
3.4.6	Examples for Memory Access Computations	55
3.4.7	Interleaving Granularity and Candidate Arrays Selection	56
3.4.8	Decision for Interleaving	57
3.5	Experimental Results	58
3.5.1	Framework	58
3.5.2	The Starting Point	58
3.5.3	Exploration Experiments for LTE	58
3.6	Conclusion	63
4	Array Interleaving – A Data Layout Transformation	65
4.1	Related Work	66
4.2	Illustrative Examples	67
4.3	Problem Formulation and Cost Estimation	70
4.3.1	Problem Definition	71
4.3.2	Memory Access Cost Estimation	71
4.3.3	Estimation of Overheads	76
4.4	Exploration for Interleaving Decision	78
4.4.1	Representation	78
4.4.2	Search Space Pruning	80
4.4.3	Optimal Path Through AIG	82

4.4.4	Effect of Interleaving Granularity	84
4.4.5	Analysis	86
4.5	Experimental Results	87
4.5.1	Compilation and Simulation Framework	87
4.5.2	Exploration Experiments	88
4.5.3	Exploration results for LTE MIMO applications	91
4.5.4	Execution Times	93
4.5.5	Other Applications of Array Interleaving	93
4.6	Conclusion	94
5	Data Flow Transformation for QR Decomposition	95
5.1	Introduction	95
5.2	Related Work	98
5.3	Operation and Memory Access Counts in Conventional QRD sequences	100
5.3.1	Analysis for the Q matrix	101
5.3.2	Analysis for the Input matrix A	102
5.4	Proposed Data Flow Transformation	103
5.4.1	Processing Sequence	103
5.4.2	Comparison with the conventional sequences	106
5.4.3	Comparison with the standard tiling transformation	107
5.5	SIMD Implementation	109
5.5.1	Vectorization within a matrix	109
5.5.2	Vectorization across matrices of same size	110
5.5.3	Operation and Memory Access counts for different implementations	111
5.6	Experimental Results	112
5.7	Conclusion	119
6	Register File Size Exploration for FFT	121
6.1	Introduction	121
6.2	Related Work	123
6.3	Exploration Strategy	124
6.3.1	Approach for non-SIMD scalar registers	125
6.3.2	Approach for SIMD vector registers	131
6.4	Experimental Results	135
6.4.1	Energy Efficient RF Size Selection	136
6.4.2	Comparison of Proposed Strategy with Pruned Radix-2 FFT implementation	142

6.5 Conclusion	146
7 Conclusion and Future Work	147
7.1 Summary of Contributions	147
7.2 Future Directions	149
Bibliography	151
Publications	163
Biography	165

List of Figures

1.1	Challenges faced by the mobile industry	2
1.2	Comparison of the different implementations	3
1.3	Architecture instance	4
1.4	Motivational example from LTE	6
1.5	GR Matrix for a rotation in the example matrix	8
1.6	QRD with illustration for different types of operations	8
1.7	Conventional sequences for QR Decomposition	9
1.8	Percentage of memory energy in total energy consumption	10
1.9	Motivational study	11
1.10	Memory energy variation with RF size	11
2.1	Embedded processor interfaced with off-chip memory	15
2.2	Operation on Scalar and Vector FU in the datapath	16
2.3	Memory hierarchy	17
2.4	Cache / Main memory structure	18
2.5	4 stage pipeline	19
2.6	Illustration for DLP	20
2.7	VLIW Processors with Unified and Clustered RFs	22
2.8	Different inter-connect topologies and a re-configurable unit	23
2.9	ADRES architecture template	24
2.10	Register File organization	26
2.11	(a) A 4-issue processor RF with 8 read ports and 4 write ports (b) A banked architecture with 4 banks, each bank having 2 read and 1 write port.	26
2.12	Very Wide Register (VWR)	27
2.13	Additional caches in the memory hierarchy	27
2.14	Off-chip and on-chip memory	28
2.15	Memory banking	28
2.16	Graph Coloring: A <i>Register Allocation</i> technique	30

2.17	Illustration for Loop Unrolling transformation	31
2.18	Example illustrating Loop Fission transformation	32
2.19	Loop Interchange transformation example	32
2.20	Example for Loop Fusion transformation	33
2.21	Blocking transformation	33
3.1	Downlink Resource Grid for channel bandwidth = 20 MHz, # PRBs=100	39
3.2	Pilot distribution for LTE2x2	40
3.3	High level overview of the LTE 2x2 application	41
3.4	Algorithmic difference between the two LTE MIMO applications	43
3.5	Pilot distribution for LTE4x4	44
3.6	High level overview of the data flow in LTE4x4 application	45
3.7	Data dependence analysis for Data Processing Block	47
3.8	Pilot extraction for LTE MIMO 2x2 application (SIMD width = 4)	48
3.9	Pilot extraction for LTE MIMO 4x4 application (SIMD width = 4)	49
3.10	Interleave operation	50
3.11	Array Interleaving examples	51
3.12	De-interleave operation	52
3.13	Access patterns for loop in Figure 1.4(a)	54
3.14	Memory access computation examples	56
3.15	Non-optimized vs. Optimized storage: variation with channel bandwidth (SIMD width = 8, 100% PRB occupation). NP: non-pilot symbol, PS0 and PS4: pilot symbols 0 and 4, Avg.: Average over all symbols	60
3.16	Non-optimized vs. Optimized storage: variation with SIMD width (BW = 20MHz, 100% PRB occupation)	61
3.17	Non-optimized vs. Optimized storage: variation with #occupied PRBs (BW=20MHz, SIMD width = 8)	62
3.18	Non-optimized vs. Optimized storage: variation due to merging kernels (BW=20MHz, 100% PRB occupation, SIMD width= 8)	63
4.1	Motivational Example	68
4.2	Example with multiple loops	69
4.3	Access pattern with interleaved layout for the example loops in Figure 4.2	70
4.4	Access pattern for loop in Figure 4.1(b)	72
4.5	Access count computation example	75
4.6	Interleaving multiple arrays ($N = 4$) with 2 stages	76
4.7	De-interleave Operation	78

4.8	Array Interleaving Graph	80
4.9	Example loop with an illustration for vector operation	81
4.10	Optimal path through AIG	83
4.11	Access patterns with different interleaving granularity	85
4.12	Interleaving/De-interleaving with different granularities	86
4.13	Non-optimized vs. Optimized storage for SOR	88
4.14	Non-optimized vs. Optimized storage for channel estimation in IEEE 802.11n for different channel bandwidths	89
4.15	Non-optimized vs. Optimized storage for image compression algorithms	90
4.16	Non-optimized vs. Optimized storage for subband coding applications	90
4.17	Non-optimized vs. Optimized storage for LTE2x2 with channel bandwidth=20MHz Cases: (A) Non-interleaved (B) Interleaved ($g=1$) (C) Interleaved ($g=2$) (D) Without remapping	92
4.18	Non-optimized vs. Optimized storage in LTE 4x4: (A) BW=20 MHz, 100% occupied, (B) BW=20 MHz, 1% occupied	92
5.1	GR Matrix for a rotation in the example matrix	97
5.2	QRD with illustration for different types of operations	98
5.3	Conventional Sequencing orders for a 4x4 matrix	99
5.4	Compute complexity of different operations	100
5.5	Proposed sequences for 16×16 matrix	105
5.6	Tiling on 16×16 matrix	108
5.7	SIMD within a matrix ($W = 4$)	110
5.8	Exploring SIMD across multiple (W) matrices	111
5.9	Evaluation of the proposed strategy against the conventional sequences when SIMD is considered within a matrix ($R = 3, W = 8$)	113
5.10	Evaluation of the proposed strategy against the conventional sequences when SIMD is considered across multiple (C) (here $C=W$) matrices ($R = 3, W = 8$)	114
5.11	Fraction of different types of computations	115
5.12	Energy variations for SIMD within and across the matrices for QRD of C ma- trices ($R = 3, W = 8$)	116
5.13	Energy variation with SIMD width on matrix of size 32×32 ($R = 3$)	116
5.14	Impact of varying the number of registers ($W = 8$)	117
5.15	Comparison of the different sequencing strategies with tiling for matrix of size 128×128	118
6.1	DIF implementation for radix-2 FFT (with $N = 8$)	122

6.2	Architectures with scalar and vector registers	125
6.3	Illustration of a 32-point FFT with 4 non-zero terms (represented by X) at the input	126
6.4	Transformed DFG of the butterfly unit (Figure 6.1(c) when $b = 0$)	127
6.5	Illustration for <i>skip_stages</i>	127
6.6	Illustration for merging stages with $RF = 8(\Rightarrow R = 4)$	128
6.7	Illustration of vector loads for two stages of 16-point FFT ($W = 4$)	132
6.8	Data re-arrangement for last $\log_2 W$ stages while using vector registers with $W = 4$	133
6.9	Total energy variation over RF size varying from 8 to 128 registers ($W = 1$) . . .	137
6.10	Total energy variation over RF size varying from 8 to 128 registers for 2K FFT	137
6.11	Total energy variation over RF size varying from 8 to 128 registers for 4K FFT	139
6.12	Total energy variation over RF size varying from 8 to 128 registers for 8K FFT	140
6.13	Total energy variation over RF size varying from 8 to 128 registers for 16K FFT	141
6.14	Comparison of different implementations for non-SIMD architectures	143
6.15	Comparison of different implementations of 2K point FFT for SIMD architectures	144
6.16	Comparison of different implementations of 8K point FFT for SIMD architectures	145

List of Tables

1.1	Memory access and operation counts for conventional rotation sequences	10
4.1	Comparison of execution times	93
5.1	Memory access and operation counts for conventional rotation sequences	103
6.1	Summary of the exploration results using scalar registers	138
6.2	Summary of the exploration results using vector registers for 2K FFT	138
6.3	Summary of the exploration results using vector registers for 4K FFT	140
6.4	Summary of the exploration results using vector registers for 8K FFT	141
6.5	Summary of the exploration results using vector registers for 16K FFT	142
6.6	Comparison of the different FFT implementation strategies for a range of m in N length FFT	146