

# Contents

## Part I Parallel Programming Models and Methodologies

<b>1</b>	<b>Parallel Programming Models</b> .....	3
	Vassilios V. Dimakopoulos	
1.1	Introduction .....	3
1.2	Classification of Parallel Programming Models .....	4
1.3	Shared-Memory Models .....	7
1.3.1	POSIX Threads .....	7
1.3.2	OpenMP .....	8
1.4	Distributed-Memory Models .....	9
1.4.1	Message-Passing (MPI) .....	9
1.5	Heterogeneity: GPGPU and Accelerator Models.....	10
1.5.1	CUDA .....	11
1.5.2	OpenCL .....	11
1.5.3	Directive-Based Models .....	12
1.6	Hybrid Models .....	13
1.6.1	Pthreads + MPI.....	13
1.6.2	OpenMP + MPI .....	14
1.6.3	PGAS .....	14
1.7	Other Parallel Programming Models .....	15
1.7.1	Languages and Language Extensions .....	15
1.7.2	Skeletal Programming .....	16
1.8	Conclusion .....	17
	References .....	17
<b>2</b>	<b>Compilation Tool Chains and Intermediate Representations</b> .....	21
	Julien Mottin, François Pacull, Ronan Keryell, and Pascal Schleuniger	
2.1	Introduction .....	21
2.1.1	Application Domains .....	21
2.1.2	Target Platforms .....	22

2.2	The Tool Chains .....	23
2.3	Intermediate Representations .....	24
2.3.1	High Level Intermediate Representation: SME-C .....	24
2.3.2	Low-Level Intermediate Representation: IR2 .....	25
2.3.3	Source-to-Source Compilers .....	26
2.4	The Tools .....	28
2.4.1	Front-End Tools.....	28
2.4.2	Back-End Tools .....	30
	References .....	32

## Part II HW/SW Architectures Concepts

<b>3</b>	<b>The STHORM Platform .....</b>	<b>35</b>
	Julien Mottin, Mickael Cartron, and Giulio Urlini	
3.1	Introduction .....	35
3.2	Platform .....	35
3.3	SystemC/TLM Platform .....	38
3.3.1	Platform Components .....	38
3.3.2	Simulator Characteristics .....	39
3.4	Cosimulation Platforms .....	40
3.4.1	IP Level Cosimulation Platform .....	40
3.4.2	HWPE Level Cosimulation Platform .....	40
3.5	Fast Simulation Platform .....	41
3.6	Simulation Platform Usage .....	42
	References .....	43
<b>4</b>	<b>The Architecture and the Technology Characterization of an FPGA-Based Customizable Application-Specific Vector Coprocessor (ASVP) .....</b>	<b>45</b>
	Roman Bartosiński, Martin Daněk, Leoš Kafka, Lukáš Kohout, and Jaroslav Sýkora	
4.1	Introduction .....	45
4.2	Related Work .....	47
4.3	Description .....	48
4.3.1	Hierarchy Layering .....	48
4.3.2	Vector Processing Unit .....	50
4.3.3	Space-Time Scheduled Crossbar .....	51
4.3.4	Address Generators .....	51
4.3.5	Data-Flow Unit .....	52
4.3.6	Programming and Simulation .....	52
4.4	Customization Methodology .....	53
4.4.1	Example .....	57
4.5	Memory Interface Characterization .....	59
4.5.1	MPMC Core in Virtex5.....	59
4.5.2	MPMC Core in Spartan6 .....	63

4.6	Technology Characterization .....	64
4.6.1	Analysis of Scaling Limits .....	64
4.6.2	Synthesis Experimental Results .....	66
4.7	Applications .....	68
4.7.1	Methodology .....	68
4.7.2	Finite Impulse Response (FIR) Filter .....	70
4.7.3	Matrix Multiplication (MATMUL) .....	71
4.7.4	Mandelbrot Set (MANDEL) .....	71
4.7.5	Image Segmentation (IMGSEG) .....	71
4.8	Analysis of Weaknesses .....	73
4.8.1	Operation Frequency .....	73
4.8.2	FPGA Area .....	73
4.8.3	Full-Reduction Windup Latencies .....	74
4.8.4	MCU Performance .....	74
4.9	Summary .....	76
	References .....	76

### Part III Run-Time and Faults Management

<b>5</b>	<b>Fault Tolerance .....</b>	<b>81</b>
	Giovanni Agosta, Mickael Cartron, and Antonio Miele	
5.1	Introduction .....	81
5.2	Programming-Model Support Level .....	83
5.2.1	Language-Based Support for Online Dynamic Reconfiguration of Fault-Tolerant Applications .....	84
5.3	Off-Line Analysis and Optimization Level .....	86
5.3.1	Static Scheduling in the Presence of Real-Time Constraints and Uncertainty due to Recovery Actions .....	86
5.4	Runtime/OS Level .....	90
5.4.1	Lightweight Detection Based on Thread Duplication .....	90
5.4.2	Flexible Scrubbing Service for P2012 .....	94
5.4.3	OS Support for Fault Tolerance .....	94
5.4.4	ReDAS, Fault-Management Layer Based on Thread Level Replication .....	95
5.4.5	Run-Time Aging Detection and Management .....	97
5.4.6	Fault Tolerance for Multi-Core Platforms Using Redundant Deterministic Multithreaded Execution .....	99
5.5	Conclusion .....	100
	References .....	101
<b>6</b>	<b>Introduction to Dynamic Code Generation: An Experiment with Matrix Multiplication for the STHORM Platform .....</b>	<b>103</b>
	Damien Couroussé, Victor Lomüller, and Henri-Pierre Charles	
6.1	Introduction .....	103
6.2	Overview of deGoal .....	106
6.2.1	Kernels and Compilettes .....	106

6.2.2	Workflow of Code Generation .....	107
6.2.3	A Tutorial Example .....	108
6.3	An Experiment on Matrix Multiplication .....	113
6.3.1	Implementation of Matrix Multiplication .....	113
6.3.2	Experimental Results .....	115
6.4	Related Work .....	118
6.5	Conclusion .....	120
	References .....	121

## Part IV Case Studies

<b>7</b>	<b>Signal Processing: Radar</b> .....	125
	Michel Barreteau and Claudia Cantini	
7.1	Brief Description of the RT-STAP Algorithm .....	125
7.1.1	Detailed Description of the Computational Phases .....	127
7.1.2	Data-Parallel Cholesky Factorization .....	129
7.2	Related Tool-Chain .....	131
7.2.1	Application and Execution Platform Modeling .....	132
7.2.2	Parallelisation on the STHORM Platform .....	134
7.2.3	IR Code Generation.....	135
7.3	Conclusion .....	138
	References .....	138
<b>8</b>	<b>Image Processing: Object Recognition</b> .....	139
	Marius Bozga, George Chasapis, Vassilios V. Dimakopoulos, and Aggelis Aggelis	
8.1	The HMAX Algorithm .....	139
8.2	DOL/BIP-Based Parallelization .....	141
8.2.1	HMAX System Level Modeling in BIP .....	142
8.2.2	Performance Analysis on the System Model .....	145
8.2.3	Implementation and Experimental Results .....	146
8.3	OpenCL-Based Parallelization .....	146
8.3.1	Basics of OpenCL .....	147
8.3.2	Parallelizing HMAX Using OpenCL .....	148
8.3.3	First Version: Using Global L3 Memory .....	149
8.3.4	Second Version: Liberal Approach .....	150
8.3.5	Third Version: Collaborative Approach .....	151
8.3.6	Experimental Results .....	152
8.4	OpenMP-Based Parallelization .....	153
8.4.1	OpenMP on STHORM .....	153
8.4.2	Parallelizing HMAX Using OpenMP.....	154
8.4.3	Experimental Results .....	156
	References .....	157

<b>9 Video Processing: Foreground Recognition in the ASVP Platform ....</b>	<b>159</b>
Petr Honzík, Roman Bartosiński, Martin Daněk, Leoš Kafka, Lukáš Kohout, and Jaroslav Sýkora	
9.1 Introduction .....	159
9.2 Platform .....	161
9.3 Application .....	163
9.3.1 Requirements .....	163
9.3.2 Motion Detection .....	163
9.3.3 MoG Implementation .....	166
9.3.4 Morphological Opening .....	166
9.3.5 Object Labelling .....	166
9.4 Implementation and Results .....	167
9.4.1 Foreground/Background Segmentation .....	168
9.4.2 Morphological Opening .....	171
9.4.3 Results .....	172
9.5 Summary .....	174
References .....	175

Smart Multicore Embedded Systems

Torquati, M.; Bertels, K.; Karlsson, S.; Pacull, F. (Eds.)

2014, XXVI, 175 p. 77 illus., 54 illus. in color., Hardcover

ISBN: 978-1-4614-8799-9