

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Challenges	1
1.2	Main Motivations	4
1.2.1	Overcoming Some Limits of the Parallelism	4
1.2.2	Taking Advantage of Combinational and Reconfigurable Logic	6
1.2.3	Software Compatibility and Reuse of Existent Binary Code	7
1.2.4	Increasing Yield and Reducing Manufacture Costs	8
1.3	This Book	10
	References	10
<b>2</b>	<b>Reconfigurable Systems</b>	<b>13</b>
2.1	Introduction	13
2.2	Basic Principles	15
2.2.1	Reconfiguration Steps	15
2.3	Underlying Execution Mechanism	17
2.4	Advantages of Using Reconfigurable Logic	20
2.4.1	Application	22
2.4.2	An Instruction Merging Example	22
2.5	Reconfigurable Logic Classification	24
2.5.1	Code Analysis and Transformation	24
2.5.2	RU Coupling	25
2.5.3	Granularity	27
2.5.4	Instruction Types	29
2.5.5	Reconfigurability	30
2.6	Directions	30
2.6.1	Heterogeneous Behavior of the Applications	31
2.6.2	Potential for Using Fine Grained Reconfigurable Arrays	34
2.6.3	Coarse Grain Reconfigurable Architectures	38
2.6.4	Comparing Both Granularities	41
	References	43

<b>3</b>	<b>Deployment of Reconfigurable Systems</b>	45
3.1	Introduction	45
3.2	Examples of Reconfigurable Architectures	46
3.2.1	Chimaera	46
3.2.2	GARP	49
3.2.3	REMARC	52
3.2.4	Rapid	55
3.2.5	Piperech (1999)	57
3.2.6	Molen	61
3.2.7	Morphosys	63
3.2.8	ADRES	66
3.2.9	Concise	68
3.2.10	PACT-XPP	69
3.2.11	RAW	73
3.2.12	Onechip	75
3.2.13	Chess	76
3.2.14	PRISM I	78
3.2.15	PRISM II	78
3.2.16	Nano	80
3.3	Recent Dataflow Architectures	81
3.4	Summary and Comparative Tables	83
3.4.1	Other Reconfigurable Architectures	83
3.4.2	Benchmarks	84
	References	89
<b>4</b>	<b>Dynamic Optimization Techniques</b>	95
4.1	Introduction	95
4.2	Binary Translation	95
4.2.1	Main Motivations	95
4.2.2	Basic Concepts	97
4.2.3	Challenges	99
4.2.4	Examples	100
4.3	Reuse	109
4.3.1	Instruction Reuse	109
4.3.2	Value Prediction	110
4.3.3	Block Reuse	111
4.3.4	Trace Reuse	112
4.3.5	Dynamic Trace Memoization and RST	114
	References	115
<b>5</b>	<b>Dynamic Detection and Reconfiguration</b>	119
5.1	Warp Processing	119
5.1.1	The Reconfigurable Array	120
5.1.2	How Translation Works	121
5.1.3	Evaluation	123

5.2	Configurable Compute Array . . . . .	124
5.2.1	The Reconfigurable Array . . . . .	124
5.2.2	Instruction Translator . . . . .	125
5.2.3	Evaluation . . . . .	128
5.3	Drawbacks . . . . .	128
	References . . . . .	129
<b>6</b>	<b>The DIM Reconfigurable System . . . . .</b>	<b>131</b>
6.1	Introduction . . . . .	131
6.1.1	General System Overview . . . . .	133
6.2	The Reconfigurable Array in Details . . . . .	134
6.3	Translation, Reconfiguration and Execution . . . . .	135
6.4	The BT Algorithm in Details . . . . .	138
6.4.1	Data Structure . . . . .	138
6.4.2	How It Works . . . . .	139
6.4.3	Additional Extensions . . . . .	140
6.4.4	Handling False Dependencies . . . . .	142
6.4.5	Speculative Execution . . . . .	143
6.5	Case Studies . . . . .	145
6.5.1	Coupling the Array to a Superscalar Processor . . . . .	145
6.5.2	Coupling the Array to the MIPS R3000 Processor . . . . .	149
6.5.3	Final Considerations . . . . .	154
6.6	DIM in Stack Machines . . . . .	155
6.7	On-Going and Future Works . . . . .	156
6.7.1	First Studies on the Ideal Shape of the Reconfigurable Array . . . . .	156
6.7.2	Sleep Transistors . . . . .	158
6.7.3	Speculation of Variable Length . . . . .	159
6.7.4	DSP, SIMD and Other Extensions . . . . .	159
6.7.5	Design Space to Be Explored . . . . .	159
	References . . . . .	159
<b>7</b>	<b>Conclusions and Future Trends . . . . .</b>	<b>163</b>
7.1	Introduction . . . . .	163
7.2	Decreasing the Routing Area of Reconfigurable Systems . . . . .	163
7.3	Measuring the Impact of the OS in Reconfigurable Systems . . . . .	165
7.4	Reconfigurable Systems to Increase the Yield . . . . .	166
7.5	Study of the Area Overhead with Technology Scaling and Future Technologies . . . . .	167
7.6	Scheduling Targeting to Low-power . . . . .	168
7.7	Granularity—Comparisons . . . . .	168
7.8	Reconfigurable Systems Attacking Different Levels of Instruction Granularity . . . . .	168
7.8.1	Multithreading . . . . .	168
7.8.2	CMP . . . . .	170

7.9 Final Considerations . . . . .	172
References . . . . .	172
<b>Index . . . . .</b>	<b>175</b>

Dynamic Reconfigurable Architectures and Transparent  
Optimization Techniques

Automatic Acceleration of Software Execution

Beck Fl., A.C.S.; Carro, L.

2010, XVII, 177 p., Hardcover

ISBN: 978-90-481-3912-5