

Contents

1	Introduction	1
1.1	Sources of Variability	1
1.2	Delay Variation	2
1.3	Book Organization	4
	References	6
 Part I Predicting and Preventing Errors		
2	Instruction-Level Tolerance	11
2.1	Introduction	11
2.2	Effect of Operating Conditions	12
2.3	Delay Variation Among Pipeline Stages	13
2.4	Instruction Characterization Methodology and Experimental Results	15
2.4.1	Gate-Level Simulation	15
2.4.2	Instruction-Level Delay Variability	16
2.4.3	Less Intrusive Variation-Tolerant Technique	17
2.4.4	Power Variability	18
2.5	Chapter Summary	19
	References	19
3	Sequence-Level Tolerance	21
3.1	Introduction	21
3.2	PVT Variations	22
3.2.1	Conventional Static Timing Analysis	24
3.2.2	Variation-Aware Statistical STA	26
3.3	Error-Tolerant Applications	27
3.3.1	Analysis of Adaptive Guardbanding for Probabilistic Applications	28

3.4	Error-Intolerant Applications.	30
3.4.1	Sequence-Level Vulnerability (SLV).	30
3.4.2	SLV Characterization	31
3.5	Adaptive Guardbanding	35
3.6	Experimental Results	37
3.6.1	Effectiveness of Adaptive Guardbanding	39
3.6.2	Overhead of Adaptive Guardbanding.	44
3.7	Chapter Summary.	44
	References.	45
4	Procedure-Level Tolerance.	47
4.1	Introduction	47
4.2	Variation-Tolerant Processor Clusters Architecture.	48
4.2.1	Variation-Aware VDD-Hopping	49
4.3	Procedure Hopping for Dynamic IR-Drop	51
4.3.1	Supporting Intra-cluster Procedure Hopping	51
4.4	Characterization of PLV to Dynamic Operating Conditions	54
4.5	Experimental Results	55
4.5.1	Cost of Procedure Hopping	57
4.6	Chapter Summary.	59
	References.	59
5	Kernel-Level Tolerance.	61
5.1	Introduction	61
5.2	Device-Level NBTI Model.	62
5.3	GP-GPU Architecture.	64
5.3.1	GP-GPU Workload Distribution	64
5.4	Aging-Aware Compilation	66
5.4.1	Observability: Aging Sensors	67
5.4.2	Prediction: Wearout Estimation Module	68
5.4.3	Controllability: Uniform Slot Assignment	68
5.5	Experimental Results	70
5.6	Chapter Summary.	73
	References.	73
6	Hierarchically Focused Guardbanding	75
6.1	Introduction	75
6.2	Timing Error Model for PVT.	76
6.2.1	Analysis Flow for Timing Error Extraction.	76
6.2.2	Parametric Model Fitting.	78
6.2.3	TER Classification.	80
6.2.4	Robustness of Classification	81
6.3	Runtime Hierarchically Focused Guardbanding	81
6.3.1	Observability	83
6.3.2	Controllability	84

6.4	A Case Study of HFG on GPUs.	85
6.5	Chapter Summary.	86
	References.	87

Part II Detecting and Correcting Errors

7	Work-Unit Tolerance	91
7.1	Introduction	91
7.2	Architectural Support for VOMP	94
7.3	Work-Unit Vulnerability and VOMP Work-Sharing	95
7.3.1	Intra- and Inter-corner WUV	98
7.3.2	Online WUV Characterization.	103
7.4	VOMP Schedulers	105
7.4.1	Variation-Aware Task Scheduling (VATS)	105
7.4.2	Variation-Aware Section Scheduling (VASS)	108
7.5	Experimental Results	109
7.5.1	Framework Setup	109
7.5.2	VOMP Results for Tasking	110
7.5.3	VOMP Results for Sections.	112
7.6	Chapter Summary.	113
	References.	114
8	Memristive-Based Associative Memory for Error Recovery	117
8.1	Introduction	117
8.2	Energy-Efficient GP-GPUs	119
8.2.1	Associative Memristive-Based Computing.	120
8.3	Collaborative Compilation	122
8.3.1	FPU Memristive-Based Computing.	124
8.4	Experimental Results	125
8.4.1	FPUs with AMM Modules	125
8.4.2	Energy Saving.	126
8.5	Chapter Summary.	129
	References.	129

Part III Accepting Errors

9	Accuracy-Configurable OpenMP	133
9.1	Introduction	133
9.2	Controlled Approximation	135
9.3	Accuracy-Configurable OpenMP Environment	136
9.3.1	Accuracy-Configurable FPUs.	136
9.3.2	OpenMP Compiler Extension for Approximation	137
9.3.3	Runtime Support	138
9.3.4	Application-Driven Hardware FPU Synthesis and Optimization.	139

9.4	Experimental Results	141
9.4.1	Error-Tolerant Applications	142
9.4.2	Error-Intolerant Applications	146
9.5	Chapter Summary	147
	References.	148
10	An Approximation Workflow for Exploiting Data-Level Parallelism in FPGA Acceleration	151
10.1	Introduction	151
10.2	OpenCL Execution Model	153
10.2.1	Mapping OpenCL Programs on FPGAs	153
10.3	GRATER: Approximation Design Workflow	154
10.3.1	Analysis and Pruning	155
10.3.2	Genetic-Based Approximation Algorithm	156
10.4	Experimental Results	159
10.4.1	Experimental Setup	159
10.4.2	Area Savings with Approximate Kernels.	160
10.4.3	Speedup.	160
10.5	Chapter Summary	163
	References.	163
11	Memristive-Based Associative Memory for Approximate Computational Reuse	165
11.1	Introduction	165
11.2	GPU Architecture Using A ² M ² Module.	167
11.2.1	Southern Islands Architecture	167
11.2.2	Approximate Associative Memristive Memory Module	168
11.3	Framework to Support A ² M ²	171
11.3.1	Execution Flow	171
11.3.2	Design Space for A ² M ²	173
11.4	Experimental Results	175
11.4.1	Experimental Setup	175
11.4.2	Energy Saving with Corresponding PSNR	177
11.5	Chapter Summary	178
	References.	179
12	Spatial and Temporal Memoization	181
12.1	Introduction	182
12.2	Spatial Memoization (Concurrent Instruction Reuse)	183
12.2.1	Single Strong Multiple Weak (SSMW) Architecture.	184
12.2.2	Experimental Results.	186
12.3	Temporal Memoization (Temporal Instruction Reuse)	188

- 12.3.1 Temporal Memoization for Error Recovery 188
 - 12.3.2 Experimental Results 189
 - 12.4 Chapter Summary 189
 - References 190
- 13 Outlook 191**
 - 13.1 Domain-Specific Resiliency 191
 - 13.1.1 Software 191
 - 13.1.2 Architecture 192
 - 13.1.3 Circuit 192
 - 13.2 Non-Von Neumann Massively Parallel Architectures 193
- Index 195**

From Variability Tolerance to Approximate Computing in
Parallel Integrated Architectures and Accelerators

Rahimi, A.; Benini, L.; Gupta, R.

2017, XV, 197 p. 86 illus., 48 illus. in color., Hardcover

ISBN: 978-3-319-53767-2