



Contents

Acknowledgements	i
Abstract (English/Français)	v
List of figures	xiii
List of tables	xv
1 Introduction	1
1.1 Instruction Supply in Servers	3
1.2 Thesis and Dissertation Goals	7
1.3 Shared Instruction Streaming Metadata	7
1.4 Unifying Instruction Streaming with BTB	8
1.5 Contributions	9
2 Why a Specialized Processor with a Shared Frontend?	11
2.1 Server Workloads and Processors	11
2.1.1 Common Microarchitectural Characteristics of Server Workloads	12
2.1.2 Specialized Server Processors	13
2.2 Frontend Metadata Overhead and Commonality	15
2.2.1 Instruction Streaming	17
2.2.2 Branch Predictors	21
2.3 Exploiting Metadata Commonality for Shared Frontend	29
2.4 Summary	30
	ix

3	SHIFT: A Design for Sharing Instruction Streaming Metadata	31
3.1	SHIFT with Dedicated History Storage	33
3.2	Virtualized SHIFT	36
3.3	SHIFT Design for Workload Consolidation	41
3.4	Methodology	41
3.5	Evaluation	44
3.5.1	Temporal Stream Lengths	44
3.5.2	Instruction Miss Coverage	45
3.5.3	Performance Comparison	47
3.5.4	LLC Overheads	49
3.5.5	Workload Consolidation	50
3.5.6	Performance Density Implications	51
3.5.7	Power Implications	52
3.6	Discussions	53
3.6.1	Choice of History Generator Core	53
3.6.2	Virtualized PIF	53
3.7	Concluding Remarks	54
4	Confluence: Unifying Instruction-Supply Metadata	55
4.1	Performance Limitations and Area Overheads of Instruction-Supply Mechanisms	57
4.1.1	Conventional Instruction-Supply Path	57
4.1.2	Covering LI-I Misses with Stream-Based Prefetching	59
4.1.3	Putting It All Together	60
4.2	Confluence Design	62
4.2.1	AirBTB Organization	64
4.2.2	AirBTB Insertions and Replacements	65
4.2.3	AirBTB Operation	66
4.2.4	Prefetcher Microarchitecture	67
4.3	Methodology	68

4.3.1	Baseline System Configuration	68
4.3.2	Instruction-Supply Mechanisms	70
4.4	Evaluation	73
4.4.1	Distribution of Branch Types	73
4.4.2	Branch Density in Cache Blocks	74
4.4.3	AirBTB Miss Coverage	75
4.4.4	Dissecting the AirBTB Benefits	77
4.4.5	Competitive BTB Miss Coverage Comparison	78
4.4.6	Competitive Performance Comparison	79
4.4.7	Performance and Area Comparison	82
4.4.8	Performance Benefits without Prefetching	83
4.5	Concluding Remarks	83
5	Related Work	85
5.1	Sharing Core Resources Across Hardware Contexts	85
5.2	Instruction Prefetching	86
5.2.1	Next-Line Prefetchers	86
5.2.2	Discontinuity Prefetcher	87
5.2.3	Branch-Predictor-Directed Prefetchers	87
5.2.4	Speculative-Thread Prefetchers	88
5.2.5	Instruction Streaming	88
5.2.6	Computation Spreading	89
5.2.7	Batching & Time Multiplexing Similar Computations	90
5.2.8	Compiler-Based Techniques	90
5.3	Branch Target Buffer	91
5.3.1	Alternative BTB Organizations	91
5.3.2	Next Cache Line and Set Prediction	92
5.3.3	Compression Techniques	93
5.3.4	Hierarchical BTBs	94

Contents

5.3.5 Prefetching BTB Metadata	94
5.3.6 Synchronizing Predictor and Cache Content	95
6 Concluding Remarks	97
6.1 Thesis Summary	97
6.2 Future Directions	98
Bibliography	103
Curriculum Vitae	115