

Contents

Acknowledgments	v
Abstract	vii
Zusammenfassung	ix
1 Introduction	1
1.1 Motivation	1
1.2 The Von Neumann Bottleneck	2
1.3 Accelerators	5
1.4 Vectorization	7
1.5 Data-Oblivious Algorithms	9
1.6 Outline	10
1.7 Contributions	12
1.8 List of Publications	14
2 Network Training Accelerator	17
2.1 Introduction	17
2.2 Architecture	20
2.2.1 Processing Cluster	20
2.2.2 Network Training Accelerator (NTX)	23
2.2.3 FMAC and FPU	23
2.2.4 Hardware Loops and Address Generation	26
2.2.5 Offloading Support	27
2.3 Programming Model	29
2.3.1 Memory and Tiling	29

2.3.2	Strided Stencil Operations	30
2.3.3	Special Functions (exp, log, div, sqrt)	31
2.3.4	Communication across HMCs	32
2.4	Results	32
2.4.1	Methodology	32
2.4.2	Precision, Sparsity, Compression	36
2.4.3	Voltage and Frequency Scaling	37
2.4.4	Multiple Logic Layers	39
2.4.5	Memory	40
2.4.6	Comparison with NeuroStream	41
2.4.7	Comparison with other Accelerators	43
2.4.8	Deployed Silicon	45
2.4.9	Scaling to multiple HMCs	46
2.4.10	Savings at Data Center Scale	49
2.5	Related Work	51
2.5.1	Accelerators for Inference	51
2.5.2	Accelerators for Training	53
2.5.3	GPUs	55
2.5.4	State of the Art since Publication	55
2.6	Summary and Conclusions	57
3	Stream Semantic Registers	59
3.1	Introduction	59
3.2	Architecture	62
3.2.1	Overview	63
3.2.2	Core	63
3.2.3	Data Mover	68
3.2.4	Interrupts and Exceptions	69
3.2.5	Memory System	71
3.3	Programming Model	73
3.3.1	Pattern Configuration	75
3.3.2	Automated Code Generation in LLVM	75
3.4	Results	76
3.4.1	ISA-level Impact	77
3.4.2	Kernels	81
3.4.3	Methodology	82
3.4.4	Single-Core Comparison	82
3.4.5	Multi-Core Comparison	87

3.4.6	Amdahl's Law and Strong Scaling	91
3.4.7	Automated Code Generation	92
3.4.8	Comparison to Other Cores	92
3.4.9	Comparison to Commercial Processors	97
3.4.10	Comparison to Vector Processors	98
3.5	Related Work	103
3.5.1	Streaming Acceleration	103
3.5.2	Loop Acceleration	104
3.5.3	Data Address Generator	105
3.5.4	Vector Processors	106
3.6	Summary and Conclusions	107
4	Floating-Point Repetition	109
4.1	Introduction	109
4.2	Architecture	113
4.2.1	Snitch Core	113
4.2.2	FPU Subsystem	116
4.2.3	Cluster	116
4.2.4	Stream Semantic Registers (SSR)	117
4.2.5	FPU Sequence Buffer	117
4.3	Programming Model	122
4.3.1	Stream Semantic Registers	122
4.3.2	FPU Sequencer	122
4.4	Results	125
4.4.1	Microkernels	125
4.4.2	Single Core	127
4.4.3	Multiple Cores	130
4.5	Related Work	139
4.6	Summary and Conclusions	141
5	Simulation and Emulation	143
5.1	Introduction	143
5.2	Related Work	147
5.3	Cycle-Accurate Simulation	148
5.4	FPGA Emulation	150
5.5	Binary Translation Simulation	153
5.5.1	Assumptions	154
5.5.2	Parsing RISC-V Binaries	154

5.5.3	Branch Target Prediction	156
5.6	Translation Procedure	157
5.6.1	Register Reads and Writes	158
5.6.2	Arithmetic Operations	162
5.6.3	Atomic Operations	164
5.6.4	Loads and Stores	164
5.6.5	Jumps and Branches	169
5.6.6	Control and Status Registers	172
5.6.7	Floating-Point Operations	174
5.6.8	Stream Semantic Registers	178
5.6.9	Floating-Point Repetition	181
5.6.10	Data Movement Operations	182
5.6.11	Tracing	183
5.7	Runtime	185
5.8	Hardware Emulation	187
5.8.1	Memory	187
5.8.2	TCDM	188
5.8.3	Multiple Cores	188
5.8.4	Multiple Clusters	189
5.9	Results	189
5.9.1	Cycle-Accurate Simulation	189
5.9.2	FPGA Emulation	191
5.9.3	Compile Time	191
5.9.4	Simulation Performance	195
5.10	Summary and Conclusions	200
6	Scale-Out Case Study	205
6.1	Introduction	205
6.2	Chiplet Architecture	206
6.2.1	Memory Hierarchy	209
6.2.2	Compute Cluster	212
6.3	Programming	212
6.3.1	Stream Semantic Registers (Xssr)	212
6.3.2	Floating-Point Repetition (Xfrep)	213
6.3.3	Data Movement (Xdma)	214
6.3.4	Typical SSR/FREP Execution	215
6.4	Prototype	216
6.5	Silicon Performance	218

6.5.1	Efficiency	218
6.5.2	Roofline	218
6.5.3	Applications	221
6.6	Summary and Conclusions	223
7	Software Development	225
7.1	Introduction	225
7.2	Related Work	227
7.3	Compiler Support	228
7.3.1	Overview	228
7.3.2	Unmodified Compiler	231
7.3.3	Assembler and Disassembler Support	239
7.3.4	SSR Intrinsics	242
7.3.5	FREP Hardware Loops	248
7.3.6	DMA Library	250
7.3.7	Automated SSR Inference	251
7.4	Programming Model	253
7.5	Data Movement Core	255
7.5.1	Kernel Integration	256
7.5.2	Double Buffering	258
7.6	Kernel Libraries	260
7.7	Data-Centric Parallel Programming	263
7.8	Summary and Conclusions	264
8	Conclusions	269
8.1	Summary	269
8.2	Main Results	270
8.3	Outlook	273
A	Chip Gallery	275
A.1	Hecate	276
A.2	Artemis	277
A.3	Selene	278
A.4	Diana	279
A.5	Kosmodrom	280
A.6	Scarabaeus	281
A.7	Billywig	282
A.8	Xavier	283

A.9 Baikonur	284
A.10 Thestral	285
B Notation and Acronyms	287
Acronyms	287
Bibliography	291
Curriculum Vitae	311