

# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Zusammenfassung</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Outline . . . . .	6
1.2 Contributions and Publications . . . . .	8
<b>2 Energy-Efficient Design of Embedded Context Recognition</b>	<b>13</b>
2.1 Introduction . . . . .	14
2.2 Related work . . . . .	17
2.3 Smartwatch System Architecture . . . . .	20
2.3.1 MSP430 Core . . . . .	21
2.3.2 PULP Accelerator . . . . .	22
2.3.3 Sensors . . . . .	25
2.4 Context Classification . . . . .	26
2.4.1 Feature Extraction on the MSP430 . . . . .	27
2.4.2 Artificial Neural Networks . . . . .	29
2.4.3 Convolutional Neural Networks . . . . .	30
2.4.4 Visual Feature Extraction on PULP . . . . .	31

2.4.5	Sensor fusion and Classification . . . . .	31
2.4.6	C4.5 Decision Tree Algorithm . . . . .	32
2.5	Results . . . . .	36
2.5.1	Context classification . . . . .	36
2.5.2	Battery Lifetime Estimation . . . . .	42
2.6	Conclusions . . . . .	42
<b>3</b>	<b>Embedded BNN Enabling Sound Event Detection</b>	<b>45</b>
3.1	Introduction . . . . .	46
3.2	Related Works . . . . .	48
3.3	Feature Extraction and BNN . . . . .	49
3.3.1	Spectrogram-based CNN and MFCC . . . . .	49
3.3.2	First Layer and Binarization . . . . .	51
3.3.3	Binary Neural Networks BNNs . . . . .	52
3.3.4	BNN Implementation . . . . .	54
3.3.5	Batch Normalization and Binarization . . . . .	55
3.3.6	Last Layer and Prediction . . . . .	56
3.3.7	Neural Network Architecture . . . . .	56
3.4	Embedded Implementation . . . . .	57
3.5	Experimental Results . . . . .	59
3.5.1	Dataset . . . . .	59
3.5.2	Accuracy . . . . .	59
3.5.3	Energy Efficiency . . . . .	60
3.5.4	Execution Time and Power Consumption . . . . .	61
3.6	Conclusions . . . . .	64
<b>4</b>	<b>Extending the RISC-V ISA for Efficient RNN-based 5G Radio Resource Management</b>	<b>67</b>
4.1	Introduction . . . . .	68
4.2	Related Works . . . . .	70
4.2.1	Generic Software-Programmable Platforms . . . . .	70
4.2.2	ML Compute Platforms . . . . .	70
4.2.3	RISC-V and RI5CY . . . . .	72

4.2.4	Benchmark Suite and Neural Networks . . . . .	73
4.2.5	Neural Networks in RRM . . . . .	81
4.2.6	Recurrent Neural Networks RNN . . . . .	81
4.2.7	Long Short-Term Memory . . . . .	82
4.2.8	Reinforcement Learning and Q-Learning . . . . .	83
4.3	HW/SW Extension and Optimizations . . . . .	84
4.3.1	Baseline Implementation (SW) . . . . .	84
4.3.2	SIMD, HWL and post-increment load (HW) . . . . .	85
4.3.3	Output Feature Map Tiling (SW) . . . . .	87
4.3.4	Tanh and Sigmoid Extension (HW) . . . . .	88
4.3.5	Load and Compute VLIW instruction (HW) . . . . .	93
4.4	Core Implementation Results . . . . .	95
4.5	Conclusion . . . . .	98
<b>5</b>	<b>YodaNN: BWN HW Acceleration</b>	<b>99</b>
5.1	Introduction . . . . .	101
5.2	Related Work . . . . .	103
5.2.1	Co-Design of DNN Models and Hardware . . . . .	103
5.2.2	CNN Acceleration Hardware . . . . .	104
5.2.3	Binary Weight Neural Networks . . . . .	106
5.3	Architecture . . . . .	107
5.3.1	Dataflow . . . . .	113
5.3.2	BinaryConnect Approach . . . . .	115
5.3.3	Latch-Based SCM . . . . .	117
5.3.4	Considering I/O Power in Energy Efficiency . . . . .	118
5.3.5	Support for Different Filter Sizes, Zero-Padding, Scaling and Biasing . . . . .	119
5.4	Results . . . . .	123
5.4.1	Computational Complexity and Energy Efficiency Measure . . . . .	123
5.4.2	Experimental Setup . . . . .	125
5.4.3	Fixed-Point vs. YodaNN . . . . .	126
5.5	Latch-based memory vs SRAM . . . . .	127

5.5.1	Real Applications . . . . .	129
5.5.2	Comparison with State-of-the-Art . . . . .	133
5.6	Conclusion . . . . .	134
<b>6</b>	<b>XNORBIN: BNN Hardware Acceleration</b>	<b>137</b>
6.1	Introduction . . . . .	137
6.2	BNN and related HW optimization . . . . .	138
6.3	Architecture . . . . .	141
6.3.1	Data Organization and Data Reuse . . . . .	145
6.3.2	Scheduling . . . . .	146
6.4	Scalability . . . . .	149
6.5	Results . . . . .	151
6.5.1	Physical Implementation . . . . .	151
6.5.2	Experimental Results . . . . .	151
6.6	Analysis Summary . . . . .	155
6.7	Conclusion . . . . .	155
<b>7</b>	<b>Hyperdrive: Solving the I/O Bottleneck in BWN HW Accelerators</b>	<b>159</b>
7.1	Introduction . . . . .	160
7.2	Hyperdrive Architecture . . . . .	162
7.3	Computational Model . . . . .	166
7.3.1	Binary Weights for Residual Networks . . . . .	166
7.3.2	Principles of Operation . . . . .	169
7.3.3	CNN Mapping . . . . .	170
7.3.4	Supported Neural Network Topologies . . . . .	175
7.4	Scalability to Multiple Chips . . . . .	177
7.4.1	Access Pattern and Storing Scheme of the Border Memories . . . . .	179
7.4.2	Border and Corner Exchange . . . . .	180
7.4.3	Border and Corner Memory . . . . .	180
7.4.4	Interface Implementation . . . . .	181
7.5	Experimental Results . . . . .	182

7.5.1	Implementation Results . . . . .	183
7.5.2	Benchmarking . . . . .	187
7.5.3	I/O in Multi-Chip Setup . . . . .	188
7.5.4	Comparison with State-of-the-Art . . . . .	191
7.6	Conclusion . . . . .	193
<b>8</b>	<b>Summary and Conclusion</b>	<b>195</b>
8.1	Overview of the Main Results . . . . .	196
8.2	Outlook . . . . .	200
<b>A</b>	<b>Notations and Acronyms</b>	<b>203</b>
	Operators . . . . .	203
	<b>Bibliography</b>	<b>209</b>
	<b>Curriculum Vitae</b>	<b>231</b>