

Contents

1	Introduction	1
2	Preliminaries	7
2.1	Background	7
2.1.1	Reversible Functions	7
2.1.2	Reversible Circuits	9
2.1.3	Quantum Circuits	13
2.2	Decision Diagrams	16
2.2.1	Binary Decision Diagrams	17
2.2.2	Quantum Multiple-valued Decision Diagrams	19
2.3	Satisfiability Solvers	21
2.3.1	Boolean Satisfiability	21
2.3.2	Extended SAT Solvers	22
3	Synthesis of Reversible Logic	27
3.1	Current Synthesis Steps	28
3.1.1	Embedding Irreversible Functions	28
3.1.2	Transformation-based Synthesis	30
3.2	BDD-based Synthesis	31
3.2.1	General Idea	32
3.2.2	Exploiting BDD Optimization	34
3.2.3	Theoretical Consideration	37
3.2.4	Experimental Results	39
3.3	SyReC: A Reversible Hardware Language	46
3.3.1	The SyReC Language	47
3.3.2	Synthesis of the Circuits	50
3.3.3	Experimental Results	53
3.4	Summary and Future Work	56
4	Exact Synthesis of Reversible Logic	57
4.1	Main Flow	58
4.2	SAT-based Exact Synthesis	61

4.2.1	Encoding for Toffoli Circuits	61
4.2.2	Encoding for Quantum Circuits	65
4.2.3	Handling Irreversible Functions	68
4.2.4	Experimental Results	70
4.3	Improved Exact Synthesis	76
4.3.1	Exploiting Higher Levels of Abstractions	77
4.3.2	Quantified Exact Synthesis	81
4.3.3	Experimental Results	84
4.4	Summary and Future Work	91
5	Embedding of Irreversible Functions	93
5.1	The Embedding Problem	94
5.2	Don't Care Assignment	96
5.2.1	Methods	96
5.2.2	Experimental Results	99
5.3	Synthesis with Output Permutation	100
5.3.1	General Idea	102
5.3.2	Exact Approach	104
5.3.3	Heuristic Approach	105
5.3.4	Experimental Results	106
5.4	Summary and Future Work	111
6	Optimization	113
6.1	Adding Lines to Reduce Circuit Cost	114
6.1.1	General Idea	114
6.1.2	Algorithm	116
6.1.3	Experimental Results	117
6.2	Reducing the Number of Circuit Lines	124
6.2.1	General Idea	125
6.2.2	Algorithm	127
6.2.3	Experimental Results	130
6.3	Optimizing Circuits for Linear Nearest Neighbor Architectures	131
6.3.1	NNC-optimal Decomposition	133
6.3.2	Optimizing NNC-optimal Decomposition	134
6.3.3	Experimental Results	138
6.4	Summary and Future Work	138
7	Formal Verification and Debugging	143
7.1	Equivalence Checking	144
7.1.1	The Equivalence Checking Problem	145
7.1.2	QMDD-based Equivalence Checking	145
7.1.3	SAT-based Equivalence Checking	148
7.1.4	Experimental Results	150
7.2	Automated Debugging and Fixing	154
7.2.1	The Debugging Problem	155
7.2.2	Determining Error Candidates	157

- 7.2.3 Determining Error Locations 161
- 7.2.4 Fixing Erroneous Circuits 165
- 7.2.5 Experimental Results 167
- 7.3 Summary and Future Work 173
- 8 Summary and Conclusions 175**
- References 177**
- Index 183**



<http://www.springer.com/978-90-481-9578-7>

Towards a Design Flow for Reversible Logic

Wille, R.; Drechsler, R.

2010, XIII, 184 p., Hardcover

ISBN: 978-90-481-9578-7