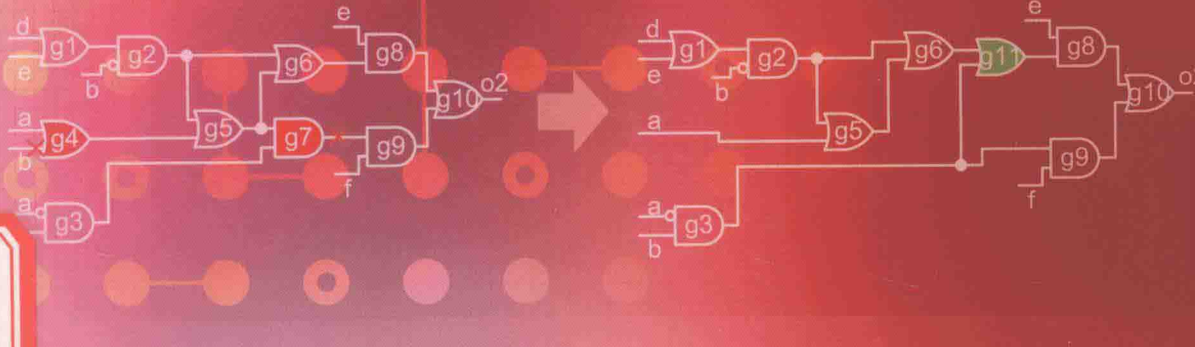
Tak-Kei Lam
Wai-Chung Tang
Xing Wei
Yi Diao
David Yu-Liang Wu

Boolean Circuit Rewiring

Bridging Logical and Physical Designs



WILEY

BOOLEAN CIRCUIT REWIRING:

BRIDGING LOGICAL AND PHYSICAL DESIGNS

Tak-Kei Lam

The Chinese University of Hong Kong, Hong Kong, P. R. China

Wai-Chung Tang

Queen Mary University of London, UK

Xing Wei

Easy-Logic Technology Ltd. Hong Kong, Hong Kong, P. R. China

Yi Diao

Easy-Logic Technology Ltd. Hong Kong, Hong Kong, P. R. China

David Yu-Liang Wu

Easy-Logic Technology Ltd. Hong Kong, Hong Kong, P. R. China

WILEY

This edition first published 2016

© 2016 John Wiley & Sons Singapore Pte Ltd

Registered office
John Wiley & Sons Singapore Pte Ltd, 1 Fusionopolis Walk, #07-01 Solaris South Tower, Singapore 138628.

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. It is sold on the understanding that the publisher is not engaged in rendering professional services and neither the publisher nor the author shall be liable for damages arising herefrom. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Library of Congress Cataloging-in-Publication Data applied for.

ISBN: 9781118750117

A catalogue record for this book is available from the British Library.

Set in 10/12pt, TimesLTStd by SPi Global, Chennai, India.

Printed and bound in Singapore by Markono Print Media Pte Ltd

1 2016

List of Figures

Figure 1.1 Basic logic gates 3
Figure 1.2 Directed acyclic graph representation of a circuit 3
Figure 1.3 Stuck-at fault 5
Figure 1.4 Untestable stuck-at fault 5
Figure 1.5 Unjustified AND/OR gates and their complete justifications 7
Figure 1.6 Example of recursive learning 9
Figure 2.1 Example of rewiring 12
Figure 2.2 Redundancy addition and removal 13
Figure 2.3 Identification of alternative wires in REWIRE 15
Figure 2.4 Mutual target and alternative wires 16
Figure 2.5 Uncontrollability implication for AND, OR, and NOT gates 16
Figure 2.6 Unobservability implication for AND, OR, and NOT gates 17
Figure 2.7 Examples of delete-first rewiring/error cancellation 18
Figure 2.8 General abstract view of error cancellation 19
Figure 2.9 Example of rewiring by logic addition 20
Figure 2.10 Examples of dynamic dominators 21
Figure 2.11 Example of an irredundant alternative wire 22
Figure 2.12 Error propagation of $sa0(b \rightarrow g1)$ and the error frontiers 23
Figure 2.13 Multiple alternatives wires 23
Figure 2.14 Sample graph-based alternate wiring patterns 25
Figure 2.15 DAG representations of the sample GBAW patterns 26
Figure 2.16 Implementation I of $h = ab + cd$ 27
Figure 2.17 Karnaugh maps of $SPFD_g$ 28
Figure 2.18 Karnaugh maps of $SPFD_e$ 28
Figure 2.19 Karnaugh maps of $SPFD_f$ 29
Figure 2.20 Karnaugh maps of $SPFD_i$ 29
Figure 2.21 Implementation II of $h = ab + cd$ 30

Figure 2.22 New Karnaugh maps of $SPFD_e$ 30
Figure 2.23 New Karnaugh maps of $SPFD_j$ 31
Figure 2.24 SPFD-based rewiring 31
Figure 2.25 Circuit with timing violation 32
Figure 2.26 Circuit without timing violation after rewiring 33
Figure 3.1 Complete set of transformations for single alternative wire addition 40
Figure 3.2 Example of node merging 44
Figure 3.3 Signatures and ODC masks 46
Figure 3.4 Eight ways to construct an alternative node and their corresponding sufficient conditions 50
Figure 3.5 SPFD: Example 1 52
Figure 3.6 SPFD: Example 2 52
Figure 3.7 SPFD-based local rewiring 56
Figure 3.8 SPFD-based global rewiring 58
Figure 3.9 SPFD-GR: $GR\_Modify\_Logic\_2$ Example 61
Figure 3.10 SPFD-ER: Example 62
Figure 4.1 General abstract view of error cancellation 68
Figure 4.2 Source mandatory assignments 69
Figure 4.3 Rectification network 70
Figure 4.4 Simplified rectification network 71
Figure 4.5 Destination nodes of alternative wires 72
Figure 4.6 Source mandatory assignments of fault $sa1(g1 \rightarrow g2)$ 73
Figure 4.7 Removal of semi-redundant and redundant SMAs 74
Figure 4.8 Substitution rules of source mandatory assignments for AND and OR gates 75
Figure 4.9 IRRA rewiring 76
Figure 4.10 Identification of SMA under recursive learning 79
Figure 4.11 Examples of dynamic dominators 81
Figure 4.12 Examples of error cancellation rules 82
Figure 4.13 Generalized wire addition and removal 82
Figure 4.14 Rectification network 83
Figure 4.15 Destination identification 86
Figure 4.16 Empirical runtime of IRRA 88
Figure 4.17 Empirical runtime of ECR 90
Figure 4.18 An error cancelled at dominator that does not have side input 92
Figure 4.19 Error propagation of w_t_error 97
Figure 4.20 Error flow graphs (a) without semi-MA and (b) with semi-MA 98

此为试读，需要完整PDF请访问：www.ertongbook.com

Figure 4.21 Structural view of error cancellation 100

Figure 4.22 Example of flow-graph-based error cancellation rewiring (FECR) 103

Figure 4.23 Example of flow-graph-based error cancellation rewiring (FECR) 104

Figure 4.24 Example of B-cut and E-cut 108

Figure 4.25 Relationship between dominators and error cuts 109

Figure 4.26 Error propagation using dominators 110

Figure 4.27 Example of E-$cuts$ and E-frontier shifting 111

Figure 4.28 Error propagation using error frontier 112

Figure 4.29 Flow of error-frontier-based error propagation 113

Figure 4.30 Locating new single-node frontier 113

Figure 4.31 Locating new multi-node frontier 114

Figure 4.32 Locating new error frontier with inconsistent path$_{node}$-MA propagation 115

Figure 4.33 Relationship between (a) dominator-based propagation and (b) error-frontier-based propagations 117

Figure 4.34 Construction of an error graph 118

Figure 4.35 Cut enumeration 119

Figure 4.36 Simplification of rectification networks by node merging 120

Figure 4.37 Structural view of n-to-m error cancellation 122

Figure 4.38 Sub-circuit for rewiring 124

Figure 4.39 Rewiring ability of CECR for different window sizes 128

Figure 5.1 Example of ODCs and node merging 134

Figure 5.2 Typical greedy decaying effect in bin-packing router (Wu and Marek-Sadowska 1995) 136

Figure 5.3 Example of orthogonal greedy coupling (Wu and Marek-Sadowska 1995) 137

Figure 5.4 Difference between node merging and rewiring 138

Figure 5.5 Example of coupling rewiring and node merging 140

Figure 5.6 Rewiring for perturbation 142

Figure 5.7 Reduction of HPWL by logic rewiring 146

Figure 5.8 Estimation of HPWL of a single fanout net 147

Figure 5.9 Estimation of HPWL of multiple fanout nets 147

Figure 5.10 HPWL increase due to AW addition 148

Figure 5.11 Revised HPWL estimation when TW and AW have the same source 148

Figure 5.12 Revised HPWL estimation when AW is in the fanin cone of TW 148

Figure 5.13 Figure Slack distribution graph of an industrial design 152

Figure 5.14 Optimizing circuit from slack mountain boundary resulting in more delay improvement 153

Figure 5.15 Example of AW selection 155

Figure 5.16 Fanout gates of TW and AW 156
Figure 5.17 Slack mountain being smoothed down by timing optimization (valley areas are not shown for clarity) 159
Figure 5.18 Example of ECO path optimization 160
Figure 5.19 NEGO-ROUT flow 163
Figure 5.20 RESTRUCTURING flow 164
Figure 5.21 Framework flow 166
Figure 5.22 Rewiring improvement upon already optimal graph-based tech map 169
Figure 5.23 Conventional FPGA EDA flow 184
Figure 5.24 Alternative function construction using MAs 186
Figure 5.25 Layout-driven synthesis using alternative functions 188
Figure 5.26 Example of external/internal wires 189
Figure 5.27 Example of postlayout logic perturbation by ATPG-based rewiring 189
Figure 5.28 Example of multiple-wire addition 190
Figure 5.29 Example of extra LUT addition 190
Figure 5.30 Rules for identifying alternative candidates 191
Figure 5.31 Example of an “existing” alternative wire ($K = 3$) (Rule 2) 192
Figure 5.32 Example of gate duplication in technology mapping ($K = 4$) 193
Figure 5.33 Example of destination LUT expansion ($K = 4$) (Rule 4) 193
Figure 5.34 Work flow of rewiring-based FPGA router 196
Figure 5.35 Example of clock gating implementation 200
Figure 5.36 Flip-flops that are mutually unobservable 201
Figure 5.37 Examples of clock gating 202
Figure 5.38 Rewiring strategy I 203
Figure 5.39 Rewiring strategy II 204
Figure 5.40 Rewiring strategy III 204

List of Tables

Table 1.1	Behavior of the basic Boolean operators	2
Table 2.1	Truth table and minterms of logical AND function	27
Table 2.2	SPFDs of the gates in Figure 2.16	28
Table 2.3	SPFDs of the gates in Figure 2.21	30
Table 3.1	Comparison of rewiring ability of SPFD-based rewiring algorithms for four-LUT FPGA designs	64
Table 3.2	Comparison of rewiring ability for different atomic SPFD assignment methods	64
Table 4.1	CPU time analysis for IRRA	88
Table 4.2	CPU time analysis for ECR	89
Table 4.3	Comparison on combinational benchmarks between IRRA and ECR	93
Table 4.4	Comparison on sequential benchmarks between IRRA and ECR	94
Table 4.5	Comparison on combinational benchmarks between NAR and ECR	95
Table 4.6	FPGA technology mapping for ILR combined with different rewiring engines on optimized benchmarks, LUT size = 4	96
Table 4.7	Comparison between ECR and FECR	106
Table 4.8	Comparison between ECR, FECR, and CECR	125
Table 4.9	Detailed statistics for CECR	126
Table 4.10	Effectiveness of windowing technique	127
Table 4.11	Effectiveness of different windowing sizes	127
Table 5.1	Experimental results of area reduction by using approaches in Plaza et al. (2007) and Chen and Wang (2010) and our approach	144
Table 5.2	Iterations of optimization in our approach	145
Table 5.3	Benchmark information	150
Table 5.4	Real versus estimated HPWL reduction	150
Table 5.5	Effectiveness of our framework for wirelength reduction of placement and global routing	151

Table 5.6 Effectiveness of our algorithm compared to the traditional peak-first algorithm and a recent work 158
Table 5.7 Comparison of our algorithm and the DCP algorithm 168
Table 5.8 Results of depth-ILR followed by area-ILR with FlowSYN ($K = 5$) 174
Table 5.9 Result of area-ILR over DAOMap ($K = 4$) 175
Table 5.10 Result of area-ILR over DAOMap ($K = 6$) 175
Table 5.11 Result of Area-ILR over IMap ($K = 4$) 176
Table 5.12 Result of Area-ILR over IMap ($K = 6$) 177
Table 5.13 Result of Area-ILR over imfs + lutpack in ABC($K = 6$) 178
Table 5.14 Result of Area-ILR on Commercial Benchmarks ($K = 4$) 179
Table 5.15 Effect of area-ILR on VPR Delay performance ($K = 4$) 181
Table 5.16 Effect of area-ILR on delay performance in commercial FPGAs ($K = 6$) 182
Table 5.17 Result of area-ILR on circuits synthesized by BDS-pga ($K = 4$) 183
Table 5.18 $q(i)$: Number of terminals versus net weight 195
Table 5.19 Experimental results on rewiring-based FPGA router 197
Table 5.20 Experimental results on rewiring-based technology mapping and routing 197
Table 5.21 Postlayout optimization by SPFD-ER 198
Table 5.22 Total cell areas of the circuits 205
Table 5.23 Power dissipation statistics 206

Preface

Our group has been working on wire-based logic restructuring (rewiring) for over a decade. Over the years, we have published numerous conference and journal papers on rewiring. As a recent major milestone, we have developed a rewiring scheme that reaches a near-complete rewiring rate (96%). This result demonstrates the high power of this kind of logic transformation techniques and the great potential of applying them on modern electronic design automation (EDA) tools.

Because of the aggressive and continuous scaling down of transistor sizes, to 45, 22 nm, and even below 16 nm, wires have become a dominant factor affecting circuit performance. Hence, rewiring is particularly suitable for today's nanometer technologies.

We could not find a book that focuses on and discusses rewiring techniques. Since rewiring techniques have become much more practical in nanometer technologies, we felt there was a need to publish a reference book to provide readers with the key ideas.

This book is of introductory to intermediate level. We hope this book will help in popularizing science, and the readers will find this book interesting and informative.

Tak-Kei Lam, Wai-Chung Tang, Xing Wei,
Yi Diao, and David Yu-Liang Wu

Introduction

The concepts of various major rewiring techniques are explained throughout the book gradually. First, readers will be presented with the basic ideas of rewiring. Next, the technical details of each kind of rewiring technique will be discussed in detail. Finally, the applications of rewiring techniques in various electronic design automation (EDA) areas will be introduced.

Intended Audience

Students studying computer/electronic engineering, academic staff, and even EDA engineers are the intended readers of this book. The readers should have some basic knowledge of Boolean algebra, logic gates, and graph theory. For readers without the related advanced knowledge, essential concepts will be introduced and explained throughout the book.

Type Conventions

The following conventions are used in this book:

- Mathematical symbols and names of circuit elements, such as a and α, are typeset in this font.
- `Codes are typeset in this font.`

Acknowledgments

Many people have contributed to this book in the forms of research output, implementations of algorithms, suggestions for content, and, last but not least, simply being encouraging. This book could never have been completed without their generous effort. We are very grateful to the following people for all they have done:

- The authors of RAMBO, REWIRE, RAMFIRE, GBAW, IRRA, NAR, ECR, FECR, CECR, SPFD-based and all other rewiring techniques.
- The authors of the typesetting system LaTeX and the plugins.

此为试读，需要完整PDF请访问：www.ertongbook.com

Contents

List of Figures ix

List of Tables xiii

Preface xv

Introduction xvii

1 Preliminaries **1**
1.1 Boolean Circuits 1
1.2 Redundancy and Stuck-at Faults 4
1.3 Automatic Test Pattern Generation (ATPG) 6
1.4 Dominators 6
1.5 Mandatory Assignments and Recursive Learning 7
1.6 Graph Theory and Boolean Circuits 8
References 10

2 Concept of Logic Rewiring **11**
2.1 What is Rewiring? 11
2.2 ATPG-based Rewiring Techniques 12
2.2.1 Add-First 12
2.2.2 Delete-First 18
2.3 Non-ATPG-based Rewiring Techniques 24
2.3.1 Graph-based Alternate Wiring (GBAW) 24
2.3.2 SPFD 25
2.4 Why are Rewiring Techniques Important? 31
References 33

3 Add-First and Non-ATPG-Based Rewiring Techniques **37**
3.1 Redundancy Addition and Removal (RAR) 37
3.1.1 RAMBO 37
3.1.2 REWIRE 38
3.1.3 RAMFIRE 41
3.1.4 Comparison Between RAR-Based Rewiring Techniques 43

3.2 Node-Based Network Addition and Removal (NAR) 43
3.2.1 Node Merging 43
3.2.2 Node Addition and Removal 48
3.3 Other Rewiring Techniques 51
3.3.1 SPFD-Based Rewiring 51
References 65

4 Delete-First Rewiring Techniques 67
4.1 IRRA 69
4.1.1 Destination of Alternative Wires 71
4.1.2 Source of Alternative Wires 72
4.2 ECR 76
4.2.1 Destination of Alternative Wires 80
4.2.2 Source of Alternative Wires 85
4.2.3 Overview of the Approach of Error-Cancellation-Based Rewiring 86
4.2.4 Complexity Analysis of ECR 87
4.2.5 Comparison Between ECR and Other Resynthesis Techniques 90
4.2.6 Experimental Result 92
4.3 FECR 96
4.3.1 Error Flow Graph Construction 97
4.3.2 Destination Node Identification 98
4.3.3 Source Node Identification 102
4.3.4 ECR is a Special Case of FECR 104
4.3.5 Complexity Analysis of FECR 105
4.3.6 Experimental Result 105
4.4 Cut-Based Error Cancellation Rewiring 107
4.4.1 Preliminaries 107
4.4.2 Error Frontier 109
4.4.3 Cut-Based Error Cancellation Rewiring 117
4.4.4 Verification of Alternative Wires 121
4.4.5 Complexity Analysis of CECR 122
4.4.6 Relationship Between ECR, FECR, and CECR 122
4.4.7 Extending CECR for n-to-m Rewiring 123
4.4.8 Speedup for CECR 124
4.4.9 Experimental Results 125
References 129

5 Applications 133
5.1 Area Reduction 133
5.1.1 Preliminaries 134
5.1.2 Our Methodology ("Long tail" vs "Bump tail" Curves) 135

5.1.3 Details of our Approach 140
5.1.4 Experimental Results 143
5.2 Postplacement Optimization 145
5.2.1 Wire-Length-Driven Rewiring-Based Postplacement Optimization 145
5.2.2 Timing-Driven Rewiring-Based Postplacement Optimization 151
5.3 ECO Timing Optimization 158
5.3.1 Preliminaries 160
5.3.2 NEGO-ROUT *Operation* 161
5.3.3 Path-Restructuring Operation 164
5.3.4 Experimental Results 166
5.4 Area Reduction in FPGA Technology Mapping 167
5.4.1 Incremental Logic Resynthesis (ILR): Depth-Oriented Mode 170
5.4.2 Incremental Logic Resynthesis (ILR): Area-Oriented Mode 171
5.4.3 Experimental Results 173
5.4.4 Conclusion 183
5.5 FPGA Postlayout Routing Optimization 184
5.5.1 Optimization by Alternative Functions 185
5.5.2 Optimization with Mapping-to-Routing Logic Rewirings 187
5.5.3 Optimization by SPFD-Based Rewiring 198
5.6 Logic Synthesis for Low Power Using Clock Gating and Rewiring 199
5.6.1 Mechanism of Clock Gating 199
5.6.2 Rewiring-Based Optimization 203
References 207

6 Summary **211**

Index **213**

1

Preliminaries

1.1 Boolean Circuits

A Boolean variable is a variable whose value can only be either 0 (false) or 1 (true) or unknown. Every Boolean variable has two literals. They are the normal form and the negation/complement of the variable. The negation of a variable always evaluates to the opposite value of the variable. Suppose v is a Boolean variable; then its negation is $\bar{v}$. When v is 1, $\bar{v}$ is 0; when v is 0, $\bar{v}$ is 1. The literals of variable v are then v and $\bar{v}$.

A function consisting of Boolean variables is known as a Boolean function. It is a mapping between Boolean spaces. For example, the function $f : B^m \rightarrow B^n$ is a mapping between the input space of m Boolean variables and the output space of n Boolean variables. We use $f(x_1, x_2, \ldots, x_{m-1}, x_m)$ to indicate the input variables or input values of the Boolean function f.

The mapping between Boolean spaces is achieved by Boolean operators. The basic Boolean operators (operations) AND, OR, NOT, XOR, and XNOR are denoted as $\cdot$, $+$, $\sim$, $\oplus$, and $\bar{\oplus}$, respectively, in this book. We may omit the symbol $\cdot$ for clarity. The behavior of the basic operators is listed in Table 1.1. Complex Boolean operators can be derived from these basic operators. In fact, only AND and NOT, or only OR and NOT, are sufficient to derive all other Boolean operations.

An example of Boolean function is $f(a, b) = a \cdot b$, which computes the logical conjunction of variables a and b. A Boolean function may contain literals. The Boolean function $f(a, b) = a \cdot \bar{b}$ is such an example that computes the logical conjunction of variable a and the negation of variable b. It may be surprising for readers who are not familiar with Boolean algebra to see a function $f(a, b, c) = a \cdot b$. This function is actually nothing special but is normal and valid. It just means that, among the three variables, the value of variable c is "don't care." That is to say, the value of c can be either 0 or 1, and $f(a, b, c) = ab = ab\bar{c} + abc$. For another example, the function $f(a, b, c) = (a + b)$ can be expanded into $f(a, b, c) = (a + b) = (a + b)\bar{c} + (a + b)c$.

Observability don't cares (ODCs) (Damiani and De Micheli 1990) of a Boolean variable are the conditions under which the variable is not affecting any of the primary outputs. For example, if an input i of an AND gate has the controlling value 0, its set of other inputs J are unobservable no matter what values they have. The ODC of J is $\bar{i}$. Satisfiability don't cares

Boolean Circuit Rewiring: Bridging Logical and Physical Designs, First Edition.
Tak-Kei Lam, Wai-Chung Tang, Xing Wei, Yi Diao and David Yu-Liang Wu.
© 2016 John Wiley & Sons Singapore Pte Ltd. Published 2016 by John Wiley & Sons Singapore Pte Ltd.

Table 1.1 Behavior of the basic Boolean operators

Operator	When will it returns true?
AND $\cdot$	All of its operands are true
OR $+$	Any one of its operands is true
NOT $\sim$	Its operand is false
XOR $\oplus$	Both of its operands have different values
XNOR $\bar{\oplus}$	Both of its operands have the same values

(SDCs) of a circuit node represent the local input patterns at the node that cannot be generated by the node's fanins. As a trivial example of SDC, if we connect all inputs of a two-input AND gate to a common signal, the values of its inputs can never be $\{1, 0\}$ or $\{0, 1\}$.

Many rules in ordinary algebra, such as commutative addition and multiplication, associative addition and multiplication, and variable distribution, can be applied into Boolean algebra. Therefore, function $f(a, b, c) = (a + b) = (a + b)\bar{c} + (a + b)c = a\bar{c} + b\bar{c} + ac + bc$. For each of the conjunction term, it can be expanded by connecting it with all combinations of the literals of the missing variables by conjunction. Some additional important rules that are obeyed in Boolean algebra only include $a \cdot a = a$ and $a + a = a$. Regarding our example, it can be expanded as follows:

$$\begin{aligned}
f(a, b, c) &= (a + b) \\
&= (a + b)\bar{c} + (a + b)c \\
&= (a\bar{c} + b\bar{c}) + (ac + bc) \\
&= (ab\bar{c} + a\bar{b}\bar{c} + ab\bar{c} + \bar{a}b\bar{c}) + (abc + a\bar{b}c + abc + \bar{a}bc) \\
&= ab\bar{c} + a\bar{b}\bar{c} + \bar{a}b\bar{c} + abc + a\bar{b}c + \bar{a}bc
\end{aligned}$$

Other rules can be derived from the basic rules easily. Since Boolean algebra is a vast area of study, even the elementary topics can cover a whole book. In this book, we shall not cover every detail.

Boolean functions can be realized in hardware using logic gates. A Boolean circuit or Boolean network is composed of gates and implements some Boolean functions. We simply use circuits or networks to represent Boolean circuits when the meaning is clear in the context. Figure 1.1 illustrates the logic gates implementing the basic Boolean functions. An example circuit composed of AND, OR, and NOT gates is shown in Figure 1.2(b). For gate e, its inputs are a and b, so it is implementing the function $a + b$. Regarding gate f, its function is $a \cdot \bar{b}$.

A less famous Boolean operator is the cofactor. The cofactor of a Boolean function $f(x_1, x_2, \ldots, x_{n-1}, x_n)$ with respect to a variable x_k is $f|_{x_k} = f(x_1, x_2, \ldots, x_{k-1}, 1, x_{k+1}, \ldots, x_{n-1}, x_n)$. Suppose $f = ab + c$, $f|_a = b + c$, and $f|_c = ab + 1 = 1$. Similarly, the cofactor of a Boolean function $f(x_1, x_2, \ldots, x_{n-1}, x_n)$ with respect to the complement of a variable x_k is $f|_{\overline{x_k}} = f(x_1, x_2, \ldots, x_{k-1}, 0, x_{k+1}, \ldots, x_{n-1}, x_n)$. Suppose $f = ab + c$, $f|_{\bar{a}} = 0 \cdot b + c = c$. Every Boolean function can be expressed using Shannon's expansion. For example, $f(x) = x \cdot f|_x + \bar{x} \cdot f|_{\bar{x}}$. An example of a function with multiple inputs is $f(x, y, z, w) = xyf|_{xy} + \bar{x}yf|_{\bar{x}y} + x\bar{y}f|_{x\bar{y}} + \bar{x}\bar{y}f|_{\bar{x}\bar{y}}$

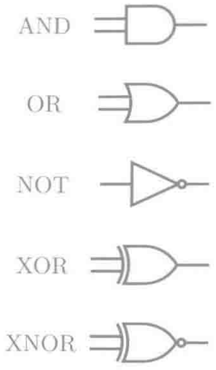
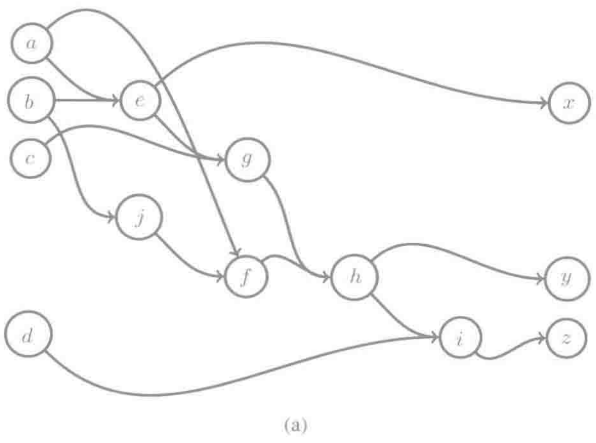


Figure 1.1 Basic logic gates



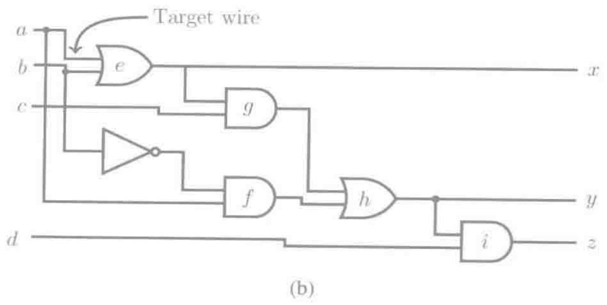


Figure 1.2 Directed acyclic graph representation of a circuit. (a) A directed acyclic graph; (b) a Boolean circuit

此为试读，需要完整PDF请访问：www.ertongbook.com