

# **Index Generation Functions**

# Synthesis Lectures on Digital Circuits and Systems

## Editor

**Mitchell A. Thornton**, *Southern Methodist University*

The *Synthesis Lectures on Digital Circuits and Systems* series is comprised of 50- to 100-page books targeted for audience members with a wide-ranging background. The Lectures include topics that are of interest to students, professionals, and researchers in the area of design and analysis of digital circuits and systems. Each Lecture is self-contained and focuses on the background information required to understand the subject matter and practical case studies that illustrate applications. The format of a Lecture is structured such that each will be devoted to a specific topic in digital circuits and systems rather than a larger overview of several topics such as that found in a comprehensive handbook. The Lectures cover both well-established areas as well as newly developed or emerging material in digital circuits and systems design and analysis.

## Index Generation Functions

Tsutomu Sasao

2019

## Microcontroller Programming and Interfacing with Texas Instruments MSP430FR2433 and MSP430FR5994 – Part II, Second Edition

Steven F. Barrett and Daniel J. Pack

2019

## Microcontroller Programming and Interfacing with Texas Instruments MSP430FR2433 and MSP430FR5994 – Part I, Second Edition

Steven F. Barrett and Daniel J. Pack

2019

## Synthesis of Quantum Circuits vs. Synthesis of Classical Reversible Circuits

Alexis De Vos, Stijn De Baerdemacker, and Yvan Van Rentergen

2018

## Boolean Differential Calculus

Bernd Steinbach and Christian Posthoff

2017

**Embedded Systems Design with Texas Instruments MSP432 32-bit Processor**

Dung Dang, Daniel J. Pack, and Steven F. Barrett  
2016

**Fundamentals of Electronics: Book 4 Oscillators and Advanced Electronics Topics**

Thomas F. Schubert and Ernest M. Kim  
2016

**Fundamentals of Electronics: Book 3 Active Filters and Amplifier Frequency**

Thomas F. Schubert and Ernest M. Kim  
2016

**Bad to the Bone: Crafting Electronic Systems with BeagleBone and BeagleBone Black, Second Edition**

Steven F. Barrett and Jason Kridner  
2015

**Fundamentals of Electronics: Book 2 Amplifiers: Analysis and Design**

Thomas F. Schubert and Ernest M. Kim  
2015

**Fundamentals of Electronics: Book 1 Electronic Devices and Circuit Applications**

Thomas F. Schubert and Ernest M. Kim  
2015

**Applications of Zero-Suppressed Decision Diagrams**

Tsutomu Sasao and Jon T. Butler  
2014

**Modeling Digital Switching Circuits with Linear Algebra**

Mitchell A. Thornton  
2014

**Arduino Microcontroller Processing for Everyone! Third Edition**

Steven F. Barrett  
2013

**Boolean Differential Equations**

Bernd Steinbach and Christian Posthoff  
2013

**Bad to the Bone: Crafting Electronic Systems with BeagleBone and BeagleBone Black**

Steven F. Barrett and Jason Kridner  
2013

### Introduction to Noise-Resilient Computing

S.N. Yanushkevich, S. Kasai, G. Tangim, A.H. Tran, T. Mohamed, and V.P. Shmerko  
2013

### Atmel AVR Microcontroller Primer: Programming and Interfacing, Second Edition

Steven F. Barrett and Daniel J. Pack  
2012

### Representation of Multiple-Valued Logic Functions

Radomir S. Stankovic, Jaakko T. Astola, and Claudio Moraga  
2012

### Arduino Microcontroller: Processing for Everyone! Second Edition

Steven F. Barrett  
2012

### Advanced Circuit Simulation Using Multisim Workbench

David Báez-López, Félix E. Guerrero-Castro, and Ofelia Delfina Cervantes-Villagómez  
2012

### Circuit Analysis with Multisim

David Báez-López and Félix E. Guerrero-Castro  
2011

### Microcontroller Programming and Interfacing Texas Instruments MSP430, Part I

Steven F. Barrett and Daniel J. Pack  
2011

### Microcontroller Programming and Interfacing Texas Instruments MSP430, Part II

Steven F. Barrett and Daniel J. Pack  
2011

### Pragmatic Electrical Engineering: Systems and Instruments

William Eccles  
2011

### Pragmatic Electrical Engineering: Fundamentals

William Eccles  
2011

### Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment

David J. Russell  
2010

### Arduino Microcontroller: Processing for Everyone! Part II

Steven F. Barrett  
2010

## Arduino Microcontroller Processing for Everyone! Part I

Steven F. Barrett

2010

## Digital System Verification: A Combined Formal Methods and Simulation Framework

Lun Li and Mitchell A. Thornton

2010

## Progress in Applications of Boolean Functions

Tsutomu Sasao and Jon T. Butler

2009

## Embedded Systems Design with the Atmel AVR Microcontroller: Part II

Steven F. Barrett

2009

## Embedded Systems Design with the Atmel AVR Microcontroller: Part I

Steven F. Barrett

2009

## Embedded Systems Interfacing for Engineers using the Freescale HCS08 Microcontroller II: Digital and Analog Hardware Interfacing

Douglas H. Summerville

2009

## Designing Asynchronous Circuits using NULL Convention Logic (NCL)

Scott C. Smith and JiaDi

2009

## Embedded Systems Interfacing for Engineers using the Freescale HCS08 Microcontroller I: Assembly Language Programming

Douglas H. Summerville

2009

## Developing Embedded Software using DaVinci & OMAP Technology

B.I. (Raj) Pawate

2009

## Mismatch and Noise in Modern IC Processes

Andrew Marshall

2009

## Asynchronous Sequential Machine Design and Analysis: A Comprehensive Development of the Design and Analysis of Clock-Independent State Machines and Systems

Richard F. Tinder

2009

### An Introduction to Logic Circuit Testing

Parag K. Lala  
2008

### Pragmatic Power

William J. Eccles  
2008

### Multiple Valued Logic: Concepts and Representations

D. Michael Miller and Mitchell A. Thornton  
2007

### Finite State Machine Datapath Design, Optimization, and Implementation

Justin Davis and Robert Reese  
2007

### Atmel AVR Microcontroller Primer: Programming and Interfacing

Steven F. Barrett and Daniel J. Pack  
2007

### Pragmatic Logic

William J. Eccles  
2007

### PSpice for Filters and Transmission Lines

Paul Tobin  
2007

### PSpice for Digital Signal Processing

Paul Tobin  
2007

### PSpice for Analog Communications Engineering

Paul Tobin  
2007

### PSpice for Digital Communications Engineering

Paul Tobin  
2007

### PSpice for Circuit Theory and Electronic Devices

Paul Tobin  
2007

### Pragmatic Circuits: DC and Time Domain

William J. Eccles  
2006

**Pragmatic Circuits: Frequency Domain**

William J. Eccles

2006

**Pragmatic Circuits: Signals and Filters**

William J. Eccles

2006

**High-Speed Digital System Design**

Justin Davis

2006

**Introduction to Logic Synthesis using Verilog HDL**

Robert B. Reese and Mitchell A. Thornton

2006

**Microcontrollers Fundamentals for Engineers and Scientists**

Steven F. Barrett and Daniel J. Pack

2006

© Springer Nature Switzerland AG 2022  
Reprint of original edition © Morgan & Claypool 2020

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, photocopy, recording, or any other except for brief quotations in printed reviews, without the prior permission of the publisher.

Index Generation Functions

Tsutomu Sasao

ISBN: 978-3-031-79910-5      paperback

ISBN: 978-3-031-79911-2      ebook

ISBN: 978-3-031-79912-9      hardcover

DOI 10.1007/978-3-031-79911-2

A Publication in the Springer series

*SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS*

Lecture #57

Series Editor: Mitchell A. Thornton, *Southern Methodist University*

Series ISSN

Print 1932-3166    Electronic 1932-3174



# Index Generation Functions

Tsutomu Sasao  
Meiji University, Japan

*SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS #57*

## ABSTRACT

Index generation functions are binary-input integer valued functions. They represent functions of content addressable memories (CAMs). Applications include: IP address tables; terminal controllers; URL lists; computer virus scanning circuits; memory patch circuits; list of English words; code converters; and pattern matching circuits.

This book shows memory-based realization of index generation functions. It shows:

1. methods to implement index generation functions by look-up table (LUT) cascades and index generation units (IGU),
2. methods to reduce the number of variables using linear transformations, and
3. methods to estimate the sizes of memories,

with many illustrations, tables, examples, exercises, and their solutions.

## KEYWORDS

affine equivalence, code converter, constant-weight code, computer virus scanning, equivalence class, functional decomposition, hash function, incompletely specified function, index generation function, index generation unit, IP address table, irreducible index generation function, linear decomposition, linear transformation, LUT cascade, minimization of variables, m-out-of-n code, random function, SAT solver, symmetric function, upper bound, URL list

# Contents

	<b>Preface</b> .....	<b>xv</b>
	<b>Acknowledgments</b> .....	<b>xvii</b>
<b>1</b>	<b>Introduction</b> .....	<b>1</b>
1.1	Motivation .....	1
1.2	Organization of the Book .....	2
<b>2</b>	<b>Applications</b> .....	<b>5</b>
2.1	IP Address Table .....	5
2.2	Terminal Access Controller .....	5
2.3	URL List .....	7
2.4	Computer Virus Scanning Circuit .....	7
2.5	Memory Patch Circuit .....	8
2.6	List of English Words .....	8
2.7	Code Converter .....	9
2.8	Remarks .....	9
<b>3</b>	<b>Definitions and Basic Properties</b> .....	<b>11</b>
3.1	Logic Functions .....	11
3.2	Functional Decomposition .....	11
3.3	Symmetric Functions .....	14
3.4	Linear Functions .....	15
3.5	Constant-Weight Code .....	15
3.6	Euler's Number $e$ and its Property .....	16
3.7	Remarks .....	17
3.8	Exercises .....	18
<b>4</b>	<b>Index Generation Functions and Their Realizations</b> .....	<b>19</b>
4.1	Index Generation Function .....	19
4.2	LUT Cascade Realization .....	21

4.3	Index Generation Unit (IGU) . . . . .	26
4.4	Remarks . . . . .	29
4.5	Exercises . . . . .	29
<b>5</b>	<b>Minimization of Primitive Variables . . . . .</b>	<b>31</b>
5.1	Minimization Algorithm . . . . .	31
5.2	Detection of Essential Variables . . . . .	33
5.3	Random Index Generation Functions . . . . .	36
5.4	Remarks . . . . .	37
5.5	Exercises . . . . .	37
<b>6</b>	<b>Linear Transformations of Input Variables . . . . .</b>	<b>41</b>
6.1	Linear Decomposition . . . . .	41
6.2	Reduction by Linear Transformations . . . . .	43
6.3	Heuristic Method to Find Linear Transformations . . . . .	45
6.4	Experimental Results . . . . .	52
6.4.1	$m$ -out-of- $n$ Code to Index Converter . . . . .	52
6.4.2	Random Index Generation Functions . . . . .	52
6.4.3	IP Address Tables . . . . .	53
6.4.4	Lists of English Words . . . . .	53
6.5	Remarks . . . . .	54
6.6	Exercises . . . . .	55
<b>7</b>	<b>Iterative Reduction of Compound Variables . . . . .</b>	<b>59</b>
7.1	Improved Upper Bound . . . . .	59
7.2	Illustrative Examples . . . . .	62
7.3	Iterative Method to Reduce Compound Variables . . . . .	70
7.4	Comparison of Minimization Methods . . . . .	71
7.4.1	Random Index Generation Functions . . . . .	71
7.4.2	$m$ -out-of- $n$ Code to Index Converters . . . . .	72
7.4.3	IP Address Tables . . . . .	73
7.4.4	Lists of English Words . . . . .	74
7.4.5	URL List . . . . .	75
7.4.6	Computation Time . . . . .	75
7.5	Remarks . . . . .	76
7.6	Exercises . . . . .	76

<b>8</b>	<b>Irreducible Index Generation Function</b>	<b>77</b>
8.1	Irreducible Index Generation Function	77
8.2	Minimum-Weight Irreducible Index Generation Functions	79
8.3	Normal Minimum-Weight Irreducible Index Generation Functions	80
8.3.1	Equivalence Classes	80
8.3.2	Normal Function	81
8.3.3	A Fast Method to Detect Irreducible Index Generation Functions	82
8.3.4	Improved Upper Bound	83
8.4	Remarks	87
8.5	Exercises	88
<b>9</b>	<b>SAT-Based Method to Find Linear Transformations</b>	<b>91</b>
9.1	SAT-Based Formulation	91
9.2	Reduction of Search Space for General Functions	94
9.3	Reduction of Search Space for cf-Symmetric Functions	96
9.4	Experimental Results	99
9.4.1	Minimization System	99
9.4.2	Randomly Generated Functions	100
9.4.3	$m$ -out-of- $n$ Code to Index Converters	100
9.5	Remarks	101
9.6	Exercises	102
<b>10</b>	<b>Statistical Approach</b>	<b>103</b>
10.1	Hash Function	103
10.2	Number of Vectors Realized by Main Memory	104
10.3	Hybrid Method	106
10.4	Super Hybrid Method	109
10.5	Parallel Sieve Method	112
10.6	Remarks	114
10.7	Exercises	114
<b>11</b>	<b>Realization Using Four IGUs</b>	<b>117</b>
11.1	Realization Using Four IGUs	117
11.2	Selection of Linear Transformations	121
11.3	Experimental Results	124
11.3.1	Realization with 4IGUs	124

11.3.2	Effect of Independent Linear Transformations . . . . .	125
11.4	Remarks . . . . .	126
<b>12</b>	<b>References on Index Generation Functions . . . . .</b>	<b>127</b>
12.1	Reduction of Variables . . . . .	127
12.2	Realization with Multiple IGUs . . . . .	127
12.3	Decomposition . . . . .	128
12.4	Analysis . . . . .	128
12.5	Architecture . . . . .	128
12.6	Applications . . . . .	128
12.7	Survey . . . . .	128
12.8	Miscellaneous . . . . .	128
<b>13</b>	<b>Conclusions . . . . .</b>	<b>129</b>
<b>A</b>	<b>Solutions . . . . .</b>	<b>131</b>
A.1	Solutions for Exercises in Chapter 3 . . . . .	131
A.2	Solutions for Exercises in Chapter 4 . . . . .	132
A.3	Solutions for Exercises in Chapter 5 . . . . .	133
A.4	Solutions for Exercises in Chapter 6 . . . . .	135
A.5	Solutions for Exercises in Chapter 7 . . . . .	139
A.6	Solutions for Exercises in Chapter 8 . . . . .	144
A.7	Solutions for Exercises in Chapter 9 . . . . .	145
A.8	Solutions for Exercises in Chapter 10 . . . . .	145
	<b>Bibliography . . . . .</b>	<b>153</b>
	<b>Author's Biography . . . . .</b>	<b>161</b>
	<b>Index . . . . .</b>	<b>163</b>

# Preface

In 2002, the Cluster Project (the first stage) of the MEXT (Ministry of Education, Culture, Sports, Science and Technology of Japan) started. In the MEXT Cluster Project, we received a large amount of research funds for many years. The final goal was to develop a commercial product by doing joint research with industry. At the end of the first stage of the Cluster Project, we encountered the concept of **index generator** as the key device in the network hardware.

An **index generation function** is an integer valued function. It is a mapping from a set of  $n$ -bit vectors to the set of integers from 0 to  $k$ .

For example, in the case of  $n = 32$  and  $k = 255$ , the function maps  $k = 255$  different two-valued vectors into  $k = 255$  distinct integers.

An index generation function can be directly implemented by a Content Addressable Memory (CAM). Unfortunately, CAMs dissipate much power and are expensive. Thus, I invented a better realization than CAM for this function. The **Index Generation Unit** (IGU) can realize index generation functions quite efficiently. Its operation is as fast as a CAM, and can be reconfigured quickly.

The design problem of an IGU can be formulated as a minimization problem of the variables for an incompletely specified function. This method is quite efficient, and we can easily implement a practical pattern matching network by using a field programming gate array (FPGA) and memories. An IGU can easily implement a circuit for  $k > 10^6$ . Furthermore, if we use a linear decomposition, we can drastically reduce the size of the memory. With this idea, many papers have been published, especially with regard to interesting mathematical problems, thus becoming a fruitful research endeavor.

This book considers design methods for index generation functions. Main applications include: IP address table lookup, packet filtering, terminal access controllers, whitelists and blacklists for URLs, virus scan circuits, memory patch circuits, fault map of memory, and pattern matching.

This book also introduces linear decomposition, which efficiently realizes index generation functions.

This book is suitable for both researchers on circuit design and computer-aided design (CAD) tool developers. To read the book, a basic knowledge of logic design is required. It will help with the understanding of the material if the reader has knowledge in linear algebra and statistics. Each chapter contains many examples, and essential chapters contain exercises. Solutions for the exercises are also provided.

Tsutomu Sasao  
November 2019



# Acknowledgments

This research is supported in part by the Grants in Aid for Scientific Research of JSPS, the grants of MEXT knowledge Cluster Project. Many people were involved in this project: Jon T. Butler, Masayuki Chiba, Yukihiro Iguchi, Kazunari Inoue, Hisashi Iwamoto, Munehiro Matsuura, Alan Mishchenko, Hiroki Nakahara, Shinobu Nagayama, Radomir S. Stankovic, and Akira Usui.

Most materials in this book have been presented at various conferences: International Workshop on Logic and Synthesis (IWLS), EUROMICRO Conference on Digital System Design, Architectures, Methods and Tools (DSD), International Symposium on Multi-Valued Logic (ISMVL), International Conference on Computer-Aided Design (ICCAD), ACM/IEEE Design Automation Conference (DAC), Asia South-Pacific Design Automation Conference (ASPDAC), and Workshop on Synthesis and System Integration of Mixed Information technologies (SASIMI), as well as journals: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *The Institute of Electronics, Information and Communication Engineers (IEICE)*, and *Journal of Multi-Valued Logic and Soft Computing*. In many cases, reviewers comments considerably improved the quality of the materials.

Numerous ideas were proposed by the brilliant students of Kyushu Institute of Technology, and Meiji University: Takaaki Nakamura, Masato Maeta, Yuji Urano, Ichido Fumishi, Kyu Matsuura, and Kazuyuki Kai.

Prof. Jon T. Butler read through the entire manuscript repeatedly and made important corrections and improvements. Dr. Alan Mishchenko motivated me to write this book.

Tsutomu Sasao  
November 2019