

Programmable Cellular Logic Arrays

Richard G. Shoup

Carnegie-Mellon University

1970

© Copyright 1970 Richard G. Shoup

Programmable Cellular Logic Arrays

by

Richard G. Shoup

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania
March, 1970

Submitted to Carnegie-Mellon University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-67-C-0058) and is monitored by the Air Force Office of Scientific Research. This document has been approved for public release and sale; its distribution is unlimited.

Carnegie-Mellon University

Programmable Cellular Logic Arrays

Richard G. Shoup

Abstract

In recent years, technological advances have provided the designer of computing hardware with the ability to batch-fabricate large numbers of logic components on a single semiconductor slice. Numerous researchers have investigated the synthesis of various kinds of digital logic by using arrays of identical cells. This dissertation considers cellular arrays whose individual cell functions are determined by parameter flip-flops and logic gates in the cell, rather than by a physical customizing operation during manufacture. Potential advantages of this technique include functional variability after manufacture, more efficient testing, and enhanced failure tolerance.

Arrays may be classified according to their generality, i.e., the number and range of the tasks which they are designed to perform. Two significantly different examples of low-generality arrays are presented and analyzed. One, a shift register array, is shown to be more effective than some conventional techniques for creating shift registers in the presence of numerous manufacturing defects.

A new cell schema is introduced which exhibits properties important in the synthesis of high-generality functions. Techniques are presented for improving the match between an array design and the target class of tasks. As an example, the problem of central processor control logic is approached in terms of a programmable logic array. A small computer is implemented in detail using these techniques. This method of synthesizing control is compared to a more conventional microprogramming approach.

Acknowledgements

I wish to thank Professor C. Gordon Bell for his advice and guidance during this work. His enthusiastic interest and his many helpful suggestions were sincerely appreciated.

I am also indebted to Dr. Jesse T. Quatse, who originally suggested this topic and provided guidance and direction during the important early phases of the work.

Finally, special thanks are due to my wife, Nancy, for her help in typing and proofreading and for her continuing personal support during this long effort.

Table of Contents

Abstract	ii
Acknowledgements	iv
Table of Contents	v
List of Tables	vi
Table of Figures	vii
1. Introduction	1
1.1 Background and definitions	2
1.2 Programmable logic and the LSI problem	7
1.3 The n-space of cellular logic arrays	18
1.4 Overview of this work	29
2. Assumptions and preliminary topics	33
2.1 Evaluations and the standard logic system	35
2.2 Yields and distribution of defects	42
2.3 Failure modes, testing, and diagnosability	50
3. Low-generality arrays	56
3.1 The arithmetic/parity arrays	58
3.2 A shift register array	82
4. High-generality arrays	101
4.1 A previous PL array	102
4.2 High array properties and a generic cell type	107
4.3 Connection functions	114
4.4 Cell nucleus functions	122
4.5 An application and analysis	138
5. Conclusions and further research	169
References	176

List of Tables

3.1 Evaluation of arrays A1 and A2	76
4.1 The equivalence class of $A \wedge B \wedge \bar{C}$	127
4.2 Three-variable function classes	129
4.3 Overlapped microcode for Rosin machine	156
4.4 Inbus and outbus signals for the Rosin machine	159
4.5 States of the Rosin machine	160

Table of Figures

1.1 Idealized fabrication of logic blocks	4
1.2 Level of integration vs. number of unique parts	9
1.3 Several interconnection structures	24
2.1 Standard set of logic elements	36
2.2 Examples of cell complexity	39
2.3 Representative slice from Petritz data	45
2.4 Experimental cell yield vs. cell size	46
2.5 Cluster defects on a hypothetical slice	49
2.6 A special case of cell failure	53
3.1 Array A1	61
3.2 Complete cell A1	62
3.3 Primary functions of cell A1	64
3.4 Secondary functions of cell A1	65
3.5 Control logic of array A1	66
3.6 Functional diagrams of array A2	69
3.7 Complete cell A2	70
3.8 Primary functions of A2	71
3.9 Secondary functions of A2	72
3.10 Control logic of array A2	73
3.11 Minimal synthesis of the primary functions	77
3.12 Minimal synthesis of the secondary functions	78
3.13 Functional diagram of array S1	83
3.14 Cell S1	84
3.15 Example of array S1	85
3.16 Control logic of array S1	87

3.17 Shift register array examples	93
4.1 Four cells in the Wahlstrom array	103
4.2 Complete Wahlstrom cell	104
4.3 A flip-flop in the Wahlstrom array	106
4.4 A high-g PL cell schema	112
4.5 Adjacent 3-input 1-output cells	116
4.6 Parity of paths in arrays	118
4.7 Busing schemes	120
4.8 Functions of 3 variables	126
4.9 Cell function F1	131
4.10 Cell function F2	131
4.11 Cell function F3	132
4.12 Cell function F4	132
4.13 Cost vs. number of functions for example cells	134
4.14 Data and control submachines	139
4.15 Control machines	141
4.16 Functional diagram of cell G1	146
4.17 C section of cell G1	147
4.18 F section of cell G1	148
4.19 Control parameter section of cell G1	148
4.20 Detail of control parameters of cell G1	150
4.21 Timing of typical G1 logic chain	153
4.22 Block diagram of Rosin machine	155
4.23 Flowchart and instruction set for Rosin machine	157
4.24 PL control array for Rosin machine	161