

AAEL
TK
7871.99
.M44
M331
1992



LAYOUT MINIMIZATION OF CMOS CELLS

**THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE**

**VLSI, COMPUTER ARCHITECTURE AND
DIGITAL SIGNAL PROCESSING**

Consulting Editor
Jonathan Allen

Latest Titles

- Iterative Identification and Restoration of Images*, R. L. Lagendijk, J. Biemond
ISBN: 0-7923-9097-0
- VLSI Design of Neural Networks*, U. Ramacher, U. Ruckert
ISBN: 0-7923-9127-6
- Synchronization Design for Digital Systems*, T. H. Meng
ISBN: 0-7923-9128-4
- Hardware Annealing in Analog VLSI Neurocomputing*, B. W. Lee, B. J. Sheu
ISBN: 0-7923-9132-2
- Neural Networks and Speech Processing*, D. P. Morgan, C. L. Scofield
ISBN: 0-7923-9144-6
- Silicon-on-Insulator Technology: Materials to VLSI*, J. P. Colinge
ISBN: 0-7923-9150-0
- Microwave Semiconductor Devices*, S. Yngvesson
ISBN: 0-7923-9156-X
- A Survey of High-Level Synthesis Systems*, R. A. Walker, R. Camposano
ISBN: 0-7923-9158-6
- Symbolic Analysis for Automated Design of Analog Integrated Circuits*,
G. Gielen, W. Sansen,
ISBN: 0-7923-9161-6
- High-Level VLSI Synthesis*, R. Camposano, W. Wolf,
ISBN: 0-7923-9159-4
- Integrating Functional and Temporal Domains in Logic Design: The False Path
Problem and its Implications*, P. C. McGeer, R. K. Brayton,
ISBN: 0-7923-9163-2
- Neural Models and Algorithms for Digital Testing*, S. T. Chakradhar,
V. D. Agrawal, M. L. Bushnell,
ISBN: 0-7923-9165-9
- Monte Carlo Device Simulation: Full Band and Beyond*, Karl Hess, editor
ISBN: 0-7923-9172-1
- The Design of Communicating Systems: A System Engineering Approach*,
C. J. Koomey
ISBN: 0-7923-9203-5
- Parallel Algorithms and Architectures for DSP Applications*,
M. A. Bayoumi, editor
ISBN: 0-7923-9209-4
- Digital Speech Processing: Speech Coding, Synthesis and Recognition*,
A. Nejat Ince, editor
ISBN: 0-7923-9220-5
- Assessing Fault Model and Test Quality*, K. M. Butler and M. R. Mercer
ISBN: 0-7923-9222-1
- Optimal VLSI Architectural Synthesis: Area, Performance and Testability*
C. H. Gebotys and M. I. Elmasry
ISBN: 0-7923-9223-X

LAYOUT MINIMIZATION OF CMOS CELLS

by

Robert L. Maziasz
and
John P. Hayes

The University of Michigan



Kluwer Academic Publishers
Boston/Dordrecht/London

engn

TK

7871.99

.M44

M331

1992

Distributors for North America:

Kluwer Academic Publishers

101 Philip Drive

Assinippi Park

Norwell, Massachusetts 02061 USA

Distributors for all other countries:

Kluwer Academic Publishers Group

Distribution Centre

Post Office Box 322

3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

Maziasz, Robert L., 1952-

Layout minimization of CMOS cells / by Robert L. Maziasz and John P. Hayes.

p. cm. -- (The Kluwer international series in engineering and computer science : SECS 160)

Includes bibliographical references and index.

ISBN 0-7923-9182-9 (alk. paper)

1. Metal oxide semiconductors, Complementary--Design and construction--Data processing. 2. Computer-aided design.

I. Hayes, John P. (John Patrick), 1944- . II. Title.

III. Series.

TK7871.99.M44M33 1992

621.381'52--dc20

91-34031

CIP

Copyright 1992 by Kluwer Academic Publishers

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, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher, Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061.

Printed on acid-free paper.

Printed in the United States of America

Enqin
728-1055
Enqin
4.10.92

To our wives

TABLE OF CONTENTS

PREFACE	xi
CHAPTER	
I. INTRODUCTION	1
1.1 Problem and Motivation	1
1.2 Layout Styles	6
1.2.1 Semi-Custom Layouts	6
1.2.2 Unstructured Methods	9
1.2.3 Programmable Logic Arrays	10
1.2.4 Gate Matrix Layouts	12
1.2.5 Functional Cells	13
1.3 Functional Cell Optimization	17
1.3.1 Layout Problem	17
1.3.2 Prior Work	21
1.4 Proposed Approach	22
1.4.1 Philosophy	22
1.4.2 Outline of the Book	24
II. FUNCTIONAL CELL LAYOUT METHODS	27
2.1 Functional Cell Design	27

2.2	Survey of Prior Methods	33
2.2.1	Static Cells	33
2.2.2	Dynamic Cells	43
2.3	Critique of Prior Work	44
III.	SERIES-PARALLEL CELL WIDTH MINIMIZATION	49
3.1	Graph Optimization Problems	50
3.2	Theory of Dual Trail Covering	55
3.3	Optimal Trail Covering without Reordering	63
3.4	Optimal Trail Covering with Reordering	67
3.5	Analysis of Complete Class of Practical Cells	68
3.6	Minimum-Width Rows of Cells	74
IV.	PLANAR CELL WIDTH MINIMIZATION	77
4.1	Nonseries-Parallel Composition	78
4.1.1	Graph Composition	79
4.1.2	Trail Covering	91
4.2	P-TrailTrace Algorithm	93
4.2.1	Description.	93
4.2.2	Design Example	96
4.2.3	Optimality.	97
4.2.4	Time Complexity.	99
4.3	Complete Study of Practical Planar Cells.	99
V.	SINGLE CELL WIDTH AND HEIGHT MINIMIZATION	105
5.1	Layout Problem.	105
5.1.1	Constraints and Assumptions.	106
5.1.2	Comparison to Other Assumptions.	107
5.2	Extension of Series-Parallel Cell Theory.	109

5.3	HR-TrailTrace Algorithm.	111
5.3.1	Description	112
5.3.2	Design Example.	115
5.3.3	Optimality	120
5.3.4	Time Complexity	123
5.4	Complete Study of Practical Cells.	124
5.5	Planar Cell Layout.	132
VI.	CELL ARRAY WIDTH AND HEIGHT MINIMIZATION	133
6.1	Layout Problem	133
6.2	HRM-TrailTrace Algorithm	135
6.2.1	Description	135
6.2.2	Design Example	137
6.2.3	Optimality	142
6.2.4	Time Complexity	148
6.3	Experimental Results	149
6.3.1	Commercial Circuits	149
6.3.2	Published Layouts	151
VII.	CONCLUSIONS	153
7.1	Contributions	153
7.2	Practical Applications and Extensions	155
	BIBLIOGRAPHY	157
	INDEX	167

PREFACE

The layout of an integrated circuit (IC) is the process of assigning geometric shape, size and position to the components (transistors and connections) used in its fabrication. Since the number of components in modern ICs is enormous, computer-aided-design (CAD) programs are required to automate the difficult layout process. Prior CAD methods are inexact or limited in scope, and produce layouts whose area, and consequently manufacturing costs, are larger than necessary. This book addresses the problem of minimizing exactly the layout area of an important class of basic IC structures called CMOS cells.

First, we precisely define the possible goals in area minimization for such cells, namely width and height minimization, with allowance for area-reducing reordering of transistors. We reformulate the layout problem in terms of a graph model and develop new graph-theoretic concepts that completely characterize the fundamental area minimization problems for series-parallel and nonseries-parallel circuits. These concepts lead to practical algorithms that solve all the basic layout minimization problems exactly, both for a single cell and for a one-dimensional array of such cells. Although a few of these layout problems have been solved or partially solved previously, we present here the first complete solutions to all the problems of interest.

We have implemented our algorithms by means of computer programs, and demonstrate that they efficiently generate minimum-area layouts for all possible cells of practical size. An analysis of these layouts permits us to make the first comprehensive evaluation of the quality and scope of prior layout methods. We show that most previous layout optimization algorithms are limited in scope, and cannot find optimal layouts for a large percentage of practical circuits. We also compare our optimal layouts to those of nonoptimal heuristic methods, and demonstrate that our exact algorithms produce significantly smaller cells.

This book will be of interest to designers of circuits or CAD tools, and others concerned with generating highly compact layouts for ICs. Chapter 1 introduces the subject of IC layout, and surveys traditional layout styles. Chapter 2 gives a brief tutorial on the cell layout problem, and evaluates the effectiveness of prior cell layout

techniques at minimizing area. Chapter 3 develops our general theory of cell layout and presents algorithms for generating minimum-width cells for series-parallel circuits; Chapter 4 generalizes these results for the nonseries-parallel case. Chapter 5 presents an algorithm that generates minimum-width and -height cells, and these results are extended to an array of cells in Chapter 6. Chapter 7 summarizes the book and suggests applications and extensions to this work.

Most of the material in this book was developed over the past few years as part of the first author's Ph.D. dissertation research at the University of Michigan. This research was supported by grants from the Office of Naval Research and the National Science Foundation, and by fellowships from the Burroughs (now part of Unisys) and Schlumberger Corporations. We wish to express our gratitude to these organizations. We also thank William P. Birmingham, Richard B. Brown, Philip J. Hanlon and Ronald J. Lomax of the University of Michigan for their suggestions.

LAYOUT MINIMIZATION OF CMOS CELLS

CHAPTER I

INTRODUCTION

We begin by discussing the general design problem for integrated circuits, and the role of the cell layout problem in relation to it. Next, we survey the popular cell layout styles for integrated circuits. The cell layout area minimization problem we address is defined and prior work on it is evaluated. We conclude with a discussion of our approach to exact layout optimization.

1.1 PROBLEM AND MOTIVATION

Since the introduction of the first commercial integrated circuit (IC) in 1961, the capacity of digital integrated circuits has been increasing at a rapid pace. Commercial IC chips in the 1960's had about 100 transistors. Through dramatic improvements in fabrication technology, very large scale integrated (VLSI) circuits with over ten million transistors are being fabricated at present, and the number of transistors per chip tends to double every year or two. The design of chips with such huge numbers of transistors presents major problems in design effort and cost. Computer-aided design (CAD) tools were developed to help designers cope with these problems. Such tools now play a key role in managing design complexity and improving designer productivity.

The design of a VLSI chip is a complicated process requiring many intermediate steps. Typically, several hierarchical levels of representation are used in the design process; see Fig. 1.1. One of the higher levels is the *register transfer level*, where the variables and the data operators represent the hardware registers and functional blocks of the data-processing section of a chip. The control section implements the sequencing of operations implied by the register transfer description. Next comes the *logic level*, at which the functional blocks are basic logic units called *gates*, such as the AND and NOR gates shown in Fig. 1.1(b); these gates implement the carry function of the adder in Fig. 1.1(a). At the *transistor level*, the logic gates are decomposed into transistor circuits. The NOR gate of Fig. 1.1(b) is translated into the transistor circuit of Fig. 1.1(c), where a *transistor* is a three-terminal device with

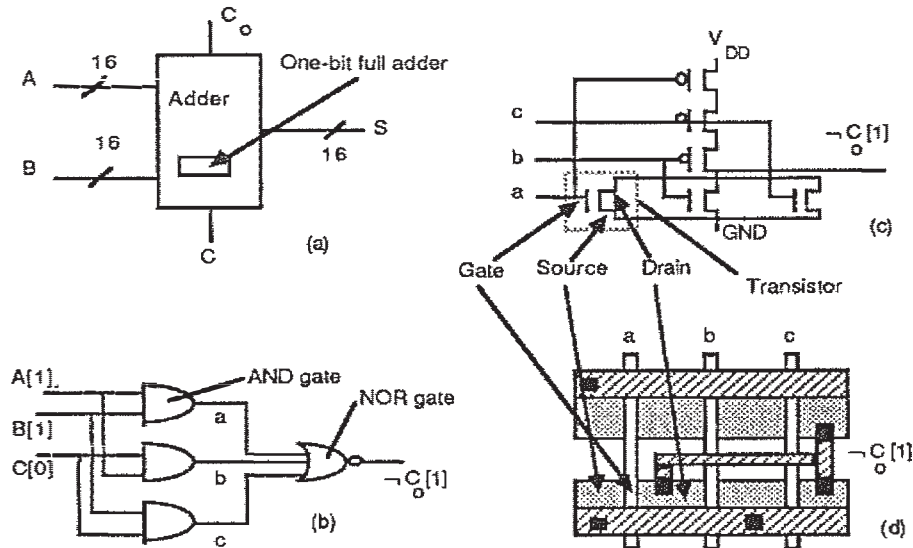


Fig. 1.1. Levels of design: (a) register transfer level (16-bit binary adder); (b) Logic level (carry circuit of a 1-bit adder); (c) transistor level (3-input NOR gate); (d) layout level (NOR gate in the CMOS functional cell style).

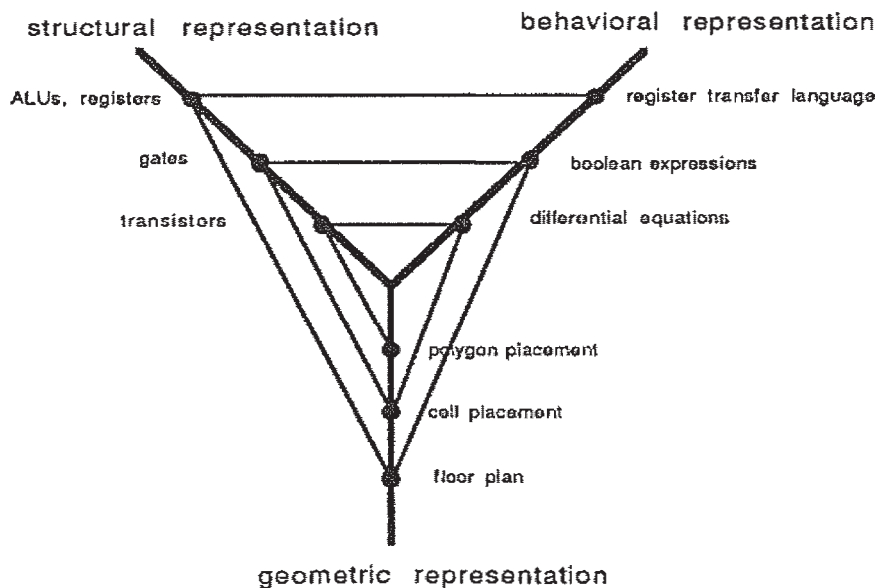


Fig. 1.2. Y chart representation of the integrated circuit design process [Ga82].

drain, source and gate terminals. The lowest design level is the *layout level*, where transistors are represented as geometric figures with dimensions such as length, width and position. The geometric representation or *layout* of a basic logic unit is called a *cell*. The layout of Fig. 1.1(d) is a cell implementing the NOR gate in Fig. 1.1(c). A large circuit may be laid out in the form of a set of interconnected cells. As in Fig. 1.1(d), the individual cells of the entire circuit are usually rectangular in shape. The layout design is transferred to mask-making facilities in this form for chip fabrication.

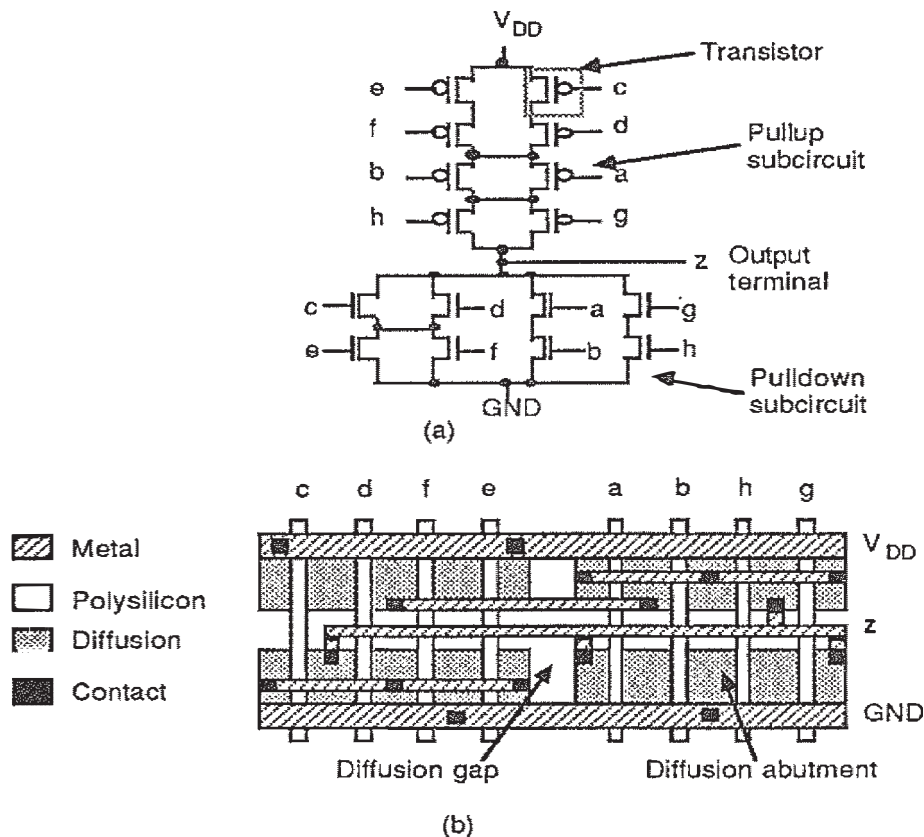


Fig. 1.3. A logic function implemented by a functional cell; (a) transistor circuit; (b) layout.

Each of the above design levels has behavioral, structural and geometric aspects. The Y chart in Fig. 1.2 illustrates these aspects, showing the order in which they are considered in a typical hierarchical, top-down design process [Ga85]. At the higher

levels, the behavioral aspects are more prominent, and the structural and geometric aspects are less important and less well defined. For example, at the register transfer level, the system is represented as registers and functional units such as an adder. These parts and their interconnection constitute the structure of the system. The behavior may be described by a register transfer language statement, such as

$$\text{register\_A} := \text{register\_B} + \text{register\_C}.$$

Each of the functional units and interconnections at this level can be roughly sized, shaped, placed and oriented in what is called a floor plan [MC80]; see Fig. 1.2. At the transistor level, all structure is explicit; the parts are low-level design components, such as transistors and wires, and the logical behavior of each part is quite simple. At the layout level, the geometric figures composing each transistor and wire are assigned size and shape; each figure represents one of various materials used in fabricating an IC, such as diffusion, polysilicon and metal shown by different shadings in Fig. 1.3(b). Each of these materials occupies a different physical layer of the IC. At the layout level, structure and geometry are most prominent.

Each of the above design steps can be assisted or even be fully automated by CAD tools. Hardware description languages are needed at each level of representation to model the structure, behavior and geometry of the design as it is refined in top-down fashion. *Synthesis tools* are needed to translate the design from a higher level of representation, like the register transfer level, to a lower one, such as the layout level. *Analysis tools* are useful at each level of representation to analyze the quality of the design in terms of correctness and performance. Examples of CAD tools that aid IC designers are logic synthesizers, placement and routing programs, cell generators and simulators. A logic synthesizer can take a design described at the register transfer level and translate it into a set of logic (boolean) equations, accounting to some extent for speed, power and area requirements. These equations can be mapped directly into basic logic units such as AND, OR or NOT gates. A cell generator can produce a cell for each logic unit. The cells can be placed and routed on the chip by placement and routing tools. Simulation at the register transfer, gate and transistor levels can predict the performance of the design at each of these intermediate levels. The designer can use this information to decide whether the behavior of the design meets its specifications.

The *layout style* used in a chip design refers to the rules governing the transistor circuit structure (the circuit type), and the rules for mapping the circuit into the layout (the cell type), and has a profound effect on the amount of circuitry that fits on a chip of a given size. There are two major layout styles, which we discuss in detail in the next section. *Semi-custom* methods select manually designed cells from a cell library; these cells are well characterized in advance for speed and power. In

full-custom methods, on the other hand, a new type of cell is designed as needed for each logical unit. This can be done by human designers or by automatic cell generators, and requires simulation to analyze the performance of the cell.

This book is concerned with the translation of a design from the transistor circuit level to the layout level. Figure 1.3 shows the main concepts involved in this process. A transistor has three external terminals called the drain, source and gate; Figs. 1.1(c) and (d) show the correspondence between the transistor terminals and the layout. The conducting path between the drain and source terminals is open or closed depending upon the voltage level on the transistor's gate terminal; thus a transistor acts like an on-off switch. The transistor circuit of Fig. 1.3(a) is of a type called complementary MOS or CMOS, and uses two different types of transistors: p-channel MOS transistors in the pullup subcircuit and n-channel MOS transistors in the pulldown subcircuit. Each of these subcircuits has two designated terminals: the pullup subcircuit, whose designated terminals are the logical high voltage, referred to as V_{DD} or 1, and the output z ; and the pulldown subcircuit, whose designated terminals are the output z and the logical low voltage or ground, denoted as GND or 0. (The terminals to which the input signals $a, b, \dots, h$ are applied are not considered designated terminals.) The transistors composing the pullup and pulldown subcircuits create conducting paths connecting the output z either to V_{DD} or GND. Thus z implements a logical function that maps each combination of the input variables onto either 0 or 1.

An important layout task is to place and orient the transistors on the chip to minimize the layout area of a circuit. Figure 1.3(b) shows a cell implementing the transistor circuit of Fig. 1.3(a) in a very area-efficient layout style called a *functional cell* [Uv81]. The various types of shading indicate the materials used to fabricate the transistor terminals and the interconnections between these terminals. The drain and source terminals of a transistor are made from diffusion and the gate terminal is polysilicon. The pullup transistors of the circuit in Fig. 1.3(a) are placed side-by-side in the top diffusion row of the cell of Fig. 1.3(b) such that their drain and source terminals are adjacent, and the pulldown transistors are placed similarly in the lower row. Transistors in the two subcircuits sharing a common gate signal are placed in the same vertical column, which makes for easy routing of the signal between them using a polysilicon column. If the transistors are placed and oriented appropriately, the width of the cell can be reduced by arranging adjacent transistors to share the same drain/source terminal via diffusion abutment, as shown in Fig. 1.3(b). This obviates the need to have separate diffusion terminals which must be isolated by an area-wasting diffusion gap, as also illustrated in Fig. 1.3(b). A judicious placement and orientation of the transistors can result in fewer metal rows for interconnections, thus reducing cell height. Therefore, it is a basic goal of a layout generation tool to lay out the transistors of a circuit and their interconnections to minimize cell area.

We have chosen to address the topic of cell layout generation because it is so important in the overall chip design process. A significant reduction in the chip area required for a given circuit can be achieved by using full-custom layout, and a dramatic increase in productivity over hand layout can be realized by automatically generating these layouts. Layout generators exist for several full-custom layout styles; these styles are surveyed in the next section. Based on the level of recent research activity, the most promising of these styles is the functional cell. However, existing functional cell methods often produce layouts that have far from optimal area. This is due to the fact that most use heuristic or approximate approaches, rather than exact algorithms. An algorithmic layout methodology is needed that achieves minimal area in the functional cell layout style.

Therefore, the general problem being addressed in this book is that of generating minimum-area cell layouts of CMOS digital integrated circuits in the functional cell style. We present a complete solution to the problem of exact layout-area minimization under precisely specified conditions. We will demonstrate that exact area minimization of all practical-sized static CMOS circuits in the functional cell style is both computationally feasible and produces significantly smaller layouts than previous methods. We analyze the properties of the important class of dual planar complementary CMOS circuits, and present efficient algorithms for obtaining the optimal area layout of these circuits. These algorithms, which we have also programmed, are based on a complete theory of cell layout, which significantly extends prior work by Uehara and vanCleemput [Uv81] and others.

1.2 LAYOUT STYLES

Digital integrated circuits are often classified according to their layout styles, which are either semi-custom, such as gate arrays and standard cells, or full-custom. In this section we survey both semi-custom methods and the following full-custom styles: unstructured methods, programmable logic arrays, gate matrix methods, and functional cells.

1.2.1 Semi-Custom Layouts

Gate arrays constitute a popular approach to the design and layout of digital circuits [Ra80]. A conventional *gate array* consists of rows and columns of cells separated by routing areas for wires interconnecting the gates; see Fig. 1.4(a). These cells are groups of unconnected transistors that can be wired together to form various types of gates, such as AND (*), OR (+), NOT (—), NAND or NOR. Fig. 1.4(b) shows the structure of a CMOS cell, which has pMOS and nMOS transistors that are interconnected by metal wires (shown by the dotted lines) to form a two-input

CHAPTER II

FUNCTIONAL CELL LAYOUT METHODS

In this chapter, we discuss one-dimensional functional cell design in depth and present a detailed survey and evaluation of all important prior methods for generating layouts in this style.

2.1 FUNCTIONAL CELL DESIGN

First we discuss the general principles of functional cell design. We begin by introducing the original set of assumptions defining this layout style, followed by an extended illustration of these fundamental ideas. Uehara and vanCleemput originally proposed the style of interest, which is illustrated by Fig. 1.3(b) [Uv78]. To simplify the resulting design problem, they made various assumptions, which we list explicitly in Fig. 2.1.

-
1. Static CMOS technology is used.
 2. The pMOS and nMOS subcircuits are series-parallel connections of transistors.
 3. The pMOS and nMOS subcircuits are geometric duals of each other.
 4. The circuit height h is limited to four for performance reasons.
 5. Each cell consists of two horizontal diffusion rows for pMOS and nMOS transistors. This implies that single p- and n-wells can be used.
 6. Complementary pMOS and nMOS transistor pairs are vertically aligned. This allows their gate terminals to be interconnected by vertical polysilicon columns without use of crossovers.
 7. Transistor drain and source terminals are connected by diffusion if the terminals are physically adjacent in the layout, or by metal if they are not adjacent. Only one metal layer is used.
-

Fig. 2.1. The assumptions defining the Uehara-vanCleemput layout style.

Figure 2.2 illustrates the first four assumptions of Fig. 2.1. A static CMOS gate is a transistor circuit composed of two 2-terminal subcircuits of MOS

transistors, a pullup subcircuit of pMOS transistors, the designated terminals of which are V_{DD} and the circuit's output z , and a pulldown subcircuit of nMOS transistors, whose terminals are z and GND. These subcircuits are functional duals, meaning that when there is a conducting path between the terminals of one subcircuit, there is not such a path between the terminals of the other subcircuit. The circuit of Fig. 2.2 is often referred to as a static CMOS "complex" gate. A pMOS transistor is shown in the pullup subcircuit. The circuit between the source and drain terminals of a pMOS transistor is closed, creating a path for current to flow, if the voltage on the gate terminal is low (at the logical 0 voltage, assuming the positive convention); otherwise the circuit is open, preventing the flow of current. Conversely, the circuit between the source and drain terminals of an nMOS transistor in the pulldown circuit is closed, if the voltage on the gate terminal is high (at the logical 1 voltage); otherwise the circuit is open. For example, if the signals c , d , a and g are low, then the path through the pMOS transistors with these signals on their gate terminals connects V_{DD} to the output terminal, and the output signal is logical 1. Note that under these conditions, no path from the output z to GND can be closed because the nMOS dual transistors labeled c , d , a and g open all possible paths to GND. If signals a and b are high, then the output is connected to GND through transistors a and b , and all paths from V_{DD} to the output are open resulting in a logical 0 output. The source terminal of an nMOS transistor in Fig. 2.2 is the

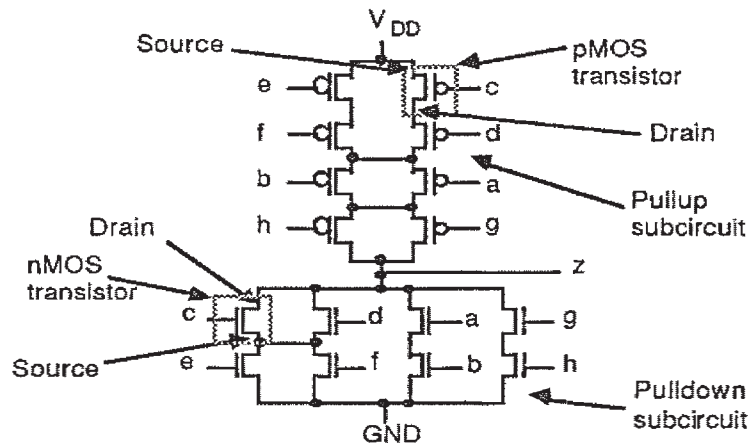


Fig. 2.2. A static CMOS complex gate.

one that is closer to GND, whereas the source terminal of a pMOS transistor is the one closer to the V_{DD} terminal; these two cases are illustrated in Fig. 2.2 for the transistors labeled c .

The second assumption of Fig. 2.1 is that the pullup and pulldown subcircuits are series-parallel. The circuit of Fig. 2.2 is of this type, where every transistor in either the pullup or pulldown subcircuit is either in series or in parallel with some other transistor. Transistors *a* and *b*, for example, are in series in the pulldown subcircuit and in parallel in the pullup subcircuit. The third assumption is that the pullup and pulldown subcircuits of the circuit are geometric duals [Ha69]. Dual subcircuits reverse the roles of the series and parallel operations; every transistor in series (parallel) with another transistor in one subcircuit is in parallel (series) with its dual transistor in the other subcircuit. Furthermore, dual transistor pairs in the subcircuits have the same signal label on their gate terminals. The pullup subcircuit of Fig. 2.2 the geometric dual of the pulldown subcircuit. The height restriction of the fourth assumption is illustrated by the longest path of transistors *c*, *d*, *a* and *g* in the pullup subcircuit. Since no path in either subcircuit is longer than this, the circuit height is four.

The remaining three assumptions relate to the layout of transistors in the functional cell. First, we briefly discuss the role of the three layers: metal, polysilicon and diffusion. Diffusion and polysilicon are primarily used for creating transistors, which occur where these layers cross, as shown in Fig. 1.9. They can also form interconnections, but they have higher resistance and capacitance than the metal layer, whose primary role is interconnection. The metal layer is on a different plane than either the diffusion or polysilicon layer, and requires a contact to make an electrical connection with these other layers. Two such contacts are shown in Fig. 2.3(b) along the bottom row of metal, interconnecting the diffusion regions of the pulldown regions to GND. Although we only use one metal layer, some IC manufacturing processes offer two or more layers.

The fifth assumption of two rows of transistors is illustrated in Fig. 2.3(b), where the pMOS transistors are placed in the upper row and the nMOS transistors in the lower row. The sixth assumption about interconnecting the gate terminals of a dual transistor pair with a vertical polysilicon column is illustrated in Fig. 2.3(a), where the pMOS and nMOS dual transistors labeled *c* in Fig. 2.2 are vertically aligned in the leftmost column also labeled *c*, and their gate terminals are connected.

Assumption 7 of Fig. 2.1 defines the methods of interconnecting those source and drain terminals in the layout that are connected in the transistor circuit. First, we outline how drain-source terminals of transistors are interconnected or electrically isolated in the functional cell, and how this affects cell size. Terminals connected in the transistor circuit are connected in the cell either by metal wires or diffusion abutment; those not connected in the circuit are isolated either by not placing such terminals adjacent to each other or, if placed adjacent, by inserting a diffusion gap between them. The position and orientation of transistors in the cell determines

which connection or isolation methods are used, and consequently determines the width and height of the cell.

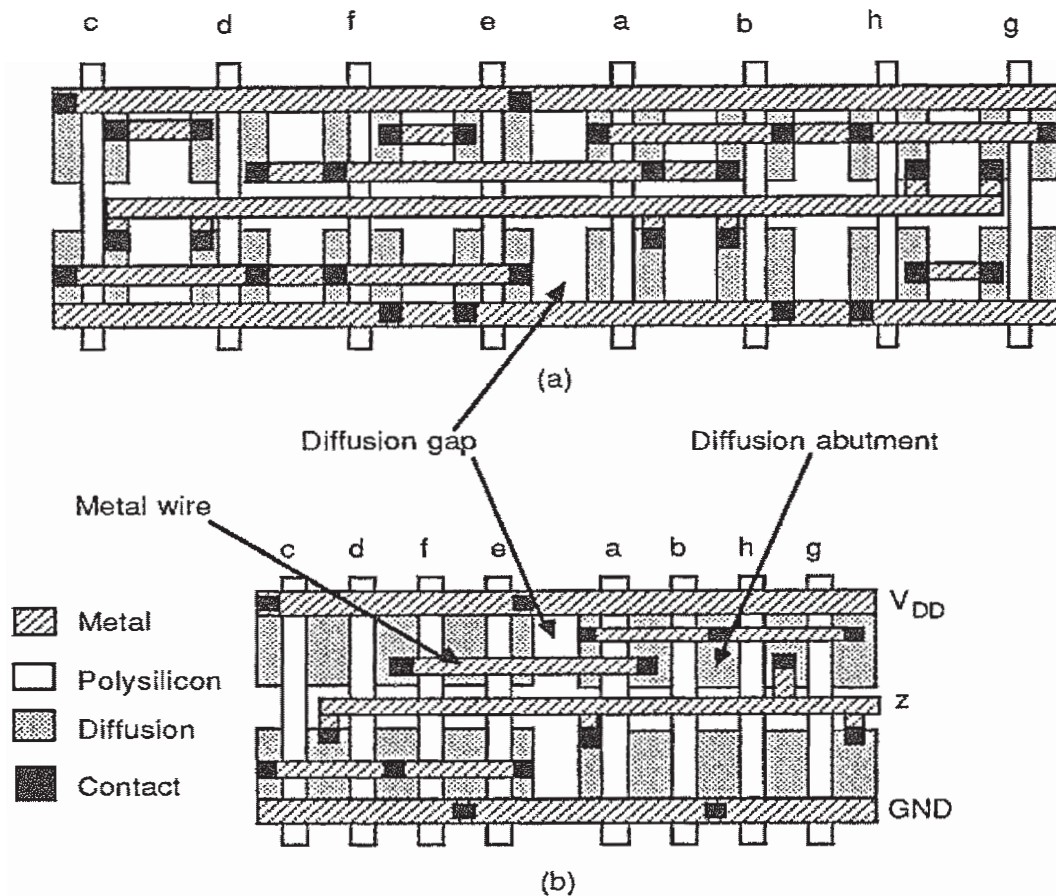


Fig. 2.3. A functional cell: (a) with no diffusion abutment; (b) with diffusion abutment.

The rules for the various interconnection methods allowed by the seventh assumption are now given. Drain-source terminals that are connected in the transistor circuit and that are also adjacent in the cell are connected by diffusion abutment; this case is illustrated in Fig. 2.3(b), where transistors *b* and *h* are connected in this way. Transistors connected in the circuit but nonadjacent in the

cell are connected by horizontal metal wires; Fig. 2.3(b) shows a metal wire in the pullup subcircuit connecting the diffusion region between columns *d* and *f* to the region between columns *a* and *b*. Drain or source terminals of two transistors not connected in the circuit but adjacent in the cell are isolated by an area-wasting diffusion gap. In Fig. 2.2, transistors *e* and *a* are not connected in the pulldown circuit, but are adjacent in the layout of Fig. 2.3(b); therefore, a diffusion gap is required between them.

We illustrate the layout process with Fig. 2.3. Assume we choose to place the dual transistor pairs in the cell in the left-to-right order of *c*, *d*, *f*, *e*, *a*, *b*, *h* and *g*. Each transistor is *oriented* in one of two ways, such that its drain terminal is either to the left or the right of its polysilicon gate terminal. First, let us consider the cell of Fig. 2.3(a) which uses only the metal layer to interconnect drain-source transistor terminals. A diffusion gap is placed between each pair of transistors, whether the adjacent terminals are to be electrically connected or not. Note that if only metal interconnections were used, all placements of transistors would result in a cell of the same width *W*. In this case, $W = T + G + 1 = 8 + 7 + 1 = 16$ units (columns). However, by choosing a good placement and orientation of the transistor pairs, and using diffusion abutment, the width of the cell can be reduced. Figure 2.3(b) shows a layout having the same transistor placement and orientation as that in Fig. 2.3(a), but using abutment wherever applicable. The cell width $W = 8 + 1 + 1 = 10$ columns. The height of the cell of Fig. 2.3(b) in terms of the number of metal rows is seven.

The height minimization problem depends on the number of metal interconnection layers used within a cell. Traditionally, one layer of metal is provided for interconnection purposes by chip fabrication processes, although more recently two or more layers have become available. Some layout styles use more than one metal layer for routing within a cell, whereas other styles employ only one metal layer in the cell even when more are available, reserving the other layers of metal for connections between cells. If more than one layer of metal is used within a cell, then in some cases cell height can be made smaller than the height of a cell using only one metal layer. The height problem is also affected by the number of contacts required between a metal layer and the other layers to ensure that a given fabrication process produces a reasonable percentage of working chips and that the total contact resistance is low enough for good circuit performance. For these reasons, multiple contacts become particularly important for submicron processes. The height of a cell may be larger if multiple contacts are required, when compared to a cell using single contacts. The vast majority of prior layout methods use a single layer of metal for cell interconnection; moreover, almost all of them use a single contact between metal layers and the other layers.

← ?