
PRINCIPLES OF CMOS VLSI DESIGN

A Systems Perspective

Neil H. E. Weste

AT&T Bell Laboratories

Kamran Eshraghian

University of Adelaide

▲ ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts • Menlo Park, California
Don Mills, Ontario • Wokingham, England • Amsterdam
Sydney • Singapore • Tokyo • Mexico City
Bogotá • Santiago • San Juan

M. C. Varley, Sponsoring Editor

Hugh Crawford, Manufacturing Supervisor

Mary Crittendon, Project Supervisor

Richard Hannus, Cover Designer

Maureen Langer, Text Designer

Elydia P. Siegel, Production Editor

This book is in the Addison-Wesley **VLSI Systems Series**
Lynn Conway and Charles Seitz, *Consulting Editors*

To A
and I

Library of Congress Cataloging in Publication Data

Weste, Neil.

Principles of CMOS VLSI design.

Bibliography: p.

Includes index.

1. Integrated circuits—Very large scale integration

—Design and construction. 2. Metaloxide semiconductors,

Complementary. I. Eshraghian, Kamran. II. Title.

III. Title: Principles of C.M.O.S. V.L.S.I. design.

TK7874.W46 1985 621.381'73 84-16738

ISBN 0-201-08222-5

Reprinted with corrections June, 1988

Copyright © 1985 by AT&T Bell Laboratories, Incorporated,
and Kamran Eshraghian

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, without the prior written
permission of the publisher. Printed in the United States of America.
Published simultaneously in Canada.

1J-HA-898

CONTENTS

PART 1	
INTRODUCTION TO CMOS TECHNOLOGY	1

1	
INTRODUCTION TO CMOS CIRCUITS	3
1.1	Introduction 4
1.2	MOS transistors 5
1.3	MOS transistor switches 7
1.4	CMOS logic 8
1.4.1	The inverter 8
1.4.2	Combinational logic 9
1.4.3	The NAND gate 11
1.4.4	The NOR gate 12
1.4.5	Compound gates 14
1.4.6	Multiplexers 14
1.4.7	Memory 17
1.5	Alternate circuit representations 18
1.5.1	Behavioral representation 19
1.5.2	Structural representation 19
1.5.3	Physical representation 23
1.6	CMOS–nMOS comparison 28
1.7	Summary 29
1.8	Exercises 29

INTRODUCTION TO CMOS CIRCUITS

1

1.1 Introduction

Over the past few years, Complementary Metal Oxide Silicon (CMOS for short) technology has played an increasingly important role in the world integrated circuit industry. Not that CMOS technology is that new. In fact the basic principle behind the MOS field effect transistor was proposed by J. Lilienfeld as early as 1925 and a similar structure closely resembling a modern MOS transistor was proposed by O. Heil in 1935. Material problems foiled these early attempts. Experiments with early field effect transistors led to the invention of the bipolar transistor. The success of the latter device led to a decline in interest in the MOS transistor. MOS devices remained an oddity until the invention of the silicon planar process in the early 1960s. Material and quality control problems dogged the introduction of the MOS device into commercial uses until around 1967 [Cobb70]. Even then, single polarity p-type transistors or n-transistors were favored. The use of both polarity devices on the same substrate was initially used for very low power applications such as watches. As the processing technology required in the fabrication of CMOS circuits was more complex than that for single polarity transistors, CMOS was sparingly applied to general system designs. As nMOS production processes became more complicated, the additional complexity of the basic CMOS process decreased in importance. Additionally, system designers were being faced with very large chip sizes and power consumptions. For this, and other reasons that will become evident during the course of this book, CMOS technology has increased in its level of importance as a VLSI technology.

The purpose of this book is to provide the designers of hardware or software systems with an understanding of CMOS technology, circuit design, layout, and system design sufficient to feel confident with the technology. The text deals with the technology down to the layout level of detail, thereby providing a bridge from a circuit to a form that may be processed. At the present time, relatively automated design approaches can take a logic schematic and automatically convert these to a chip layout. However, these approaches do not really capitalize on the fundamental objects available in an IC — transistors. Hopefully, with texts such as this, software systems may be constructed that capture an expert's knowledge so that arbitrarily structured and architected silicon systems of enormous complexity can be built rapidly and accurately.

The book is divided into three main sections. Chapters 1–5 essentially provide a circuit view of CMOS IC design. In the first chapter, a rather idealized view of CMOS technology will be taken and some basic forms of logic and memory will be introduced. This

is aimed
without
depth
of the
phenom
backgr
3. The
interest
are als
subject
operati
marizin
of basi
clockin
circuit
6–8. T
Chapte
where
on testi
in Chap
implem
of some
system.
using th
with a
covered
five exa
these ch
to a ful
tionship

1.2

An MOS
several l
terials. A
consist o
separate
of transi
nMOS,
silicon by
negative
the dual

is aimed at providing an unencumbered picture of the technology without delving into unnecessary detail. Chapter 2 deals at a greater depth with the operation of the MOS transistor and the DC operation of the CMOS inverter and a few other basic circuits of interest. The phenomenon commonly known as *latch-up* is also discussed. Some background to CMOS processing technology is presented in Chapter 3. The basic processes in current use are described along with some interesting process enhancements. Some representative design rules are also presented in this chapter. Chapter 4 treats the important subject of performance estimation and characterization of circuit operation. This covers speed and power dissipation. A section summarizing some first order scaling effects is also included. A summary of basic CMOS circuit forms is provided in Chapter 5. Various clocking schemes are discussed with the emphasis being placed on circuit design and layout. The second section comprises Chapters 6–8. These chapters present a *sub-system* view of CMOS design. Chapter 6 focusses on a range of current design methods, identifying where appropriate the factors in common with CMOS. A section on testing is also included. Symbolic layout techniques are discussed in Chapter 7, with particular emphasis placed on a design system implemented by one of the authors. This is included to give an idea of some of the components required in a custom CMOS design system. Chapter 8 is a rather hefty chapter on sub-system design, using the circuits discussed in Chapter 5. Discussion is commenced with a variety of adder designs. RAMs, ROMs, and PLAs are then covered. The final section is contained in Chapter 9. It consists of five examples of CMOS ICs of varying complexity. The purpose of these chapters is to illustrate the architectural decisions that lead to a full custom chip design. Where appropriate the specific relationships to CMOS technology are noted.

1.2 MOS transistors

An MOS (Metal-Oxide-Silicon) structure is created by superimposing several layers of conducting, insulating, and transistor forming materials. After a series of processing steps, a typical structure might consist of levels called diffusion, polysilicon, and metal that are separated by insulating layers. CMOS technology provides two types of transistors (also called *devices* in this text), an n-type transistor (nMOS) and a p-type transistor (pMOS). These are fabricated in silicon by using either negatively doped silicon that is rich in electrons (negatively charged) or positively doped silicon that is rich in holes (the dual of electrons and positively charged). Typical physical

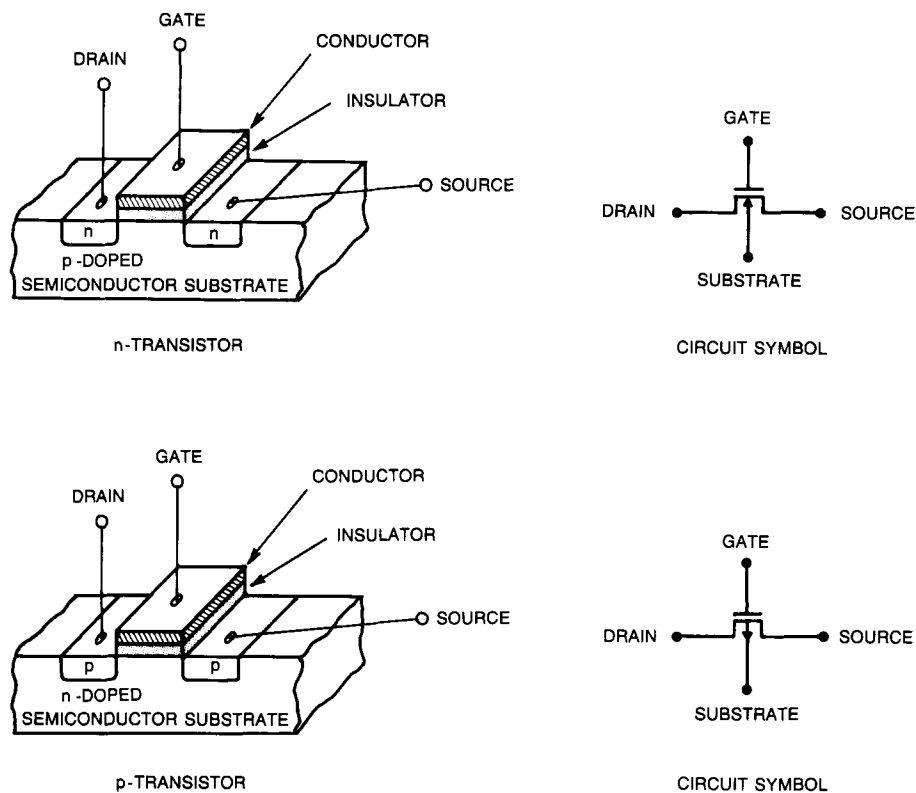


FIGURE 1.1. MOS transistor physical structures

structures for the two types of MOS transistors are shown in Fig. 1.1. For the n-transistor, the structure consists of a section of p-type silicon separating two diffused areas of n-type silicon. The area separating the n regions is capped with a sandwich consisting of an insulator and a conducting electrode called the GATE. Similarly, for the p-transistor the structure consists of a section of n-type silicon separating two p-type diffused areas. The p-transistor also has a gate electrode. For the purpose of introduction, we will assume that the transistors have two additional connections, which are designated the DRAIN and the SOURCE, these being formed by the n (p in the case of a p-device) diffused regions. The gate is a control input — it affects the flow of electrical current between the drain and source. In fact the drain and source may be viewed as two switched terminals. They are physically equivalent and the name assignment depends on the direction of current flow. For now, we will regard them as interchangeable.

1.3 MOS transistor switches

The gate controls the passage of current between the drain and source. Simplifying this to the extreme allows the MOS transistors to be viewed as simple on/off switches. In the following discussion, we will assume that a '1' is a high voltage that is normally set to 5 volts and called POWER or V_{DD} . The symbol '0' will be assumed to be a low voltage that is normally set to 0 volts and called GROUND or V_{SS} . The strength of the '1' and '0' signals can vary. The "strength" of a signal is measured by its ability to sink or source current. In general, the stronger a signal, the more current it can source or sink. Where the term output and input are used, the output will be the source of stronger '1's and '0's than the input. The power supplies (V_{DD} and V_{SS}) are the source of the strongest '1's and '0's.

The nMOS switch (N-SWITCH) is shown in Fig. 1.2a. The schematic representation is shown along with a switch representation. The gate has been labeled with signal s , the drain a , and the source b . In an N-SWITCH, the switch is closed or 'ON' if the drain and source are connected. This occurs when there is a '1' on the GATE. The switch is open or 'OFF' if the drain and source are disconnected. A '0' on the GATE ensures this condition. These conditions are summarized in Fig. 1.2b. An N-SWITCH is almost a perfect switch when a '0' is to be passed from an output to an input (say a to b).

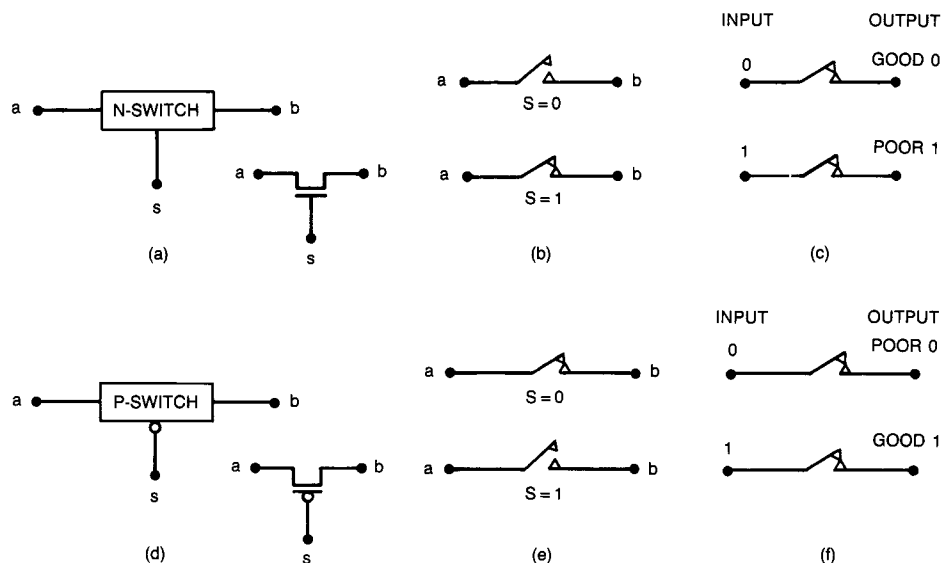


FIGURE 1.2. MOS transistors viewed as switches

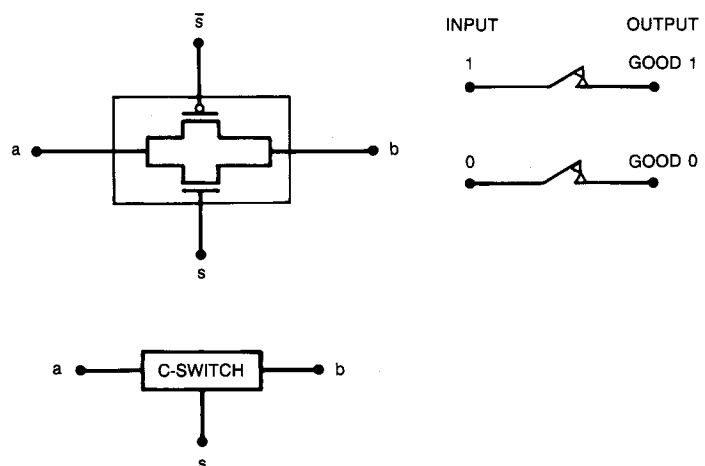


FIGURE 1.3. A complementary switch

However, the N-SWITCH is an imperfect switch when passing a '1'. In doing this, the '1' voltage level is reduced a little. (This is explained in Section 2.5.) These cases are shown in Fig. 1.2c. The pMOS switch (P-SWITCH) is shown in Fig. 1.2d. It has different properties than the N-SWITCH. The P-SWITCH is closed or 'ON' when there is a '0' on the gate. The switch is open or 'OFF' when there is a '1' on the gate. Fig. 1.2e depicts these conditions. Notice that the pMOS and nMOS switches are ON and OFF for complementary values of the gate signal. We denote this difference for a P-SWITCH by including the inversion bubble in the notation. A P-SWITCH is almost perfect for passing '1' signals but is an imperfect switch when passing '0' signals. This is illustrated in Fig. 1.2f.

By combining an N-SWITCH and a P-SWITCH in parallel (Fig. 1.3), we obtain a switch in which '0's and '1's are passed in an acceptable fashion. We term this a complementary switch or C-SWITCH. In a circuit where only a '0' or '1' has to be passed, the appropriate sub-switch (N or P) may be deleted, reverting to a P-SWITCH or N-SWITCH. Note that a double rail logic is implied for this switch (the control input and its complement are routed to all switches where necessary. The control signal is applied to the n-transistor and the complement to the p-transistor.).

TABLE 1.1. Inverter truth table

INPUT	OUTPUT
0	1
1	0

1.4 CMOS logic

1.4.1 The inverter

Table 1.1 outlines the necessary states to implement a logical inverter. If we examine this table we find that when there is a '0' on the

FIGURE

input from there This a '0' s that a '0's is nee to pas sched fully down pull-

1.4.2

If two page closer Fig. 1 for P- if bot

W comp is a Fig. 1 contr closer B nation

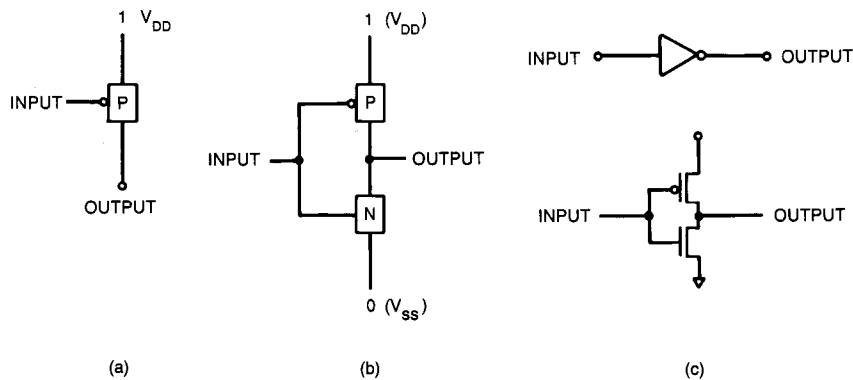


FIGURE 1.4. Construction of a CMOS inverter

input, there is a '1' at the output. This suggests a P-SWITCH connected from a '1' source (V_{DD}) to the output as shown in Fig. 1.4a. When there is a '1' on the input a '0' has to be connected to the output. This suggests the addition of an N-SWITCH between the output and a '0' source (V_{SS}). The completed circuit is shown in Fig. 1.4b. Note that as the lower switch only has to pass a '0' (the V_{SS} source of '0's is stronger than the output of the inverter), only an N-SWITCH is needed. By similar reasoning, the upper switch, which only has to pass a '1', needs only a P-SWITCH. The transistor schematic and schematic icon forms for this are shown in Fig. 1.4c. In general, a fully complementary CMOS gate always has an N-SWITCH (pull-down) array to connect the output to '0' (V_{SS}) and a P-SWITCH (pull-up) array to connect the output to '1' (V_{DD}).

1.4.2 Combinational logic

If two N-SWITCHES are placed in series, as shown in Fig. 1.5a on page 10, then the composite switch constructed by this action is closed (or ON) if both switches are closed (or ON) as illustrated in Fig. 1.5a. This yields an "AND" function. The corresponding structure for P-switches is shown in Fig. 1.5b. The composite switch is closed if both inputs are set to '0'.

When two N-SWITCHES are placed in parallel (Fig. 1.5c), the composite switch is closed if either switch is closed (either input is a '1'). Thus an 'OR' function is created. The switch shown in Fig. 1.5d is composed of two P-SWITCHES placed in parallel. In contrast to the previous case, if either input is a '0' the switch is closed.

By using combinations of these constructions, CMOS combinational gates may be constructed.

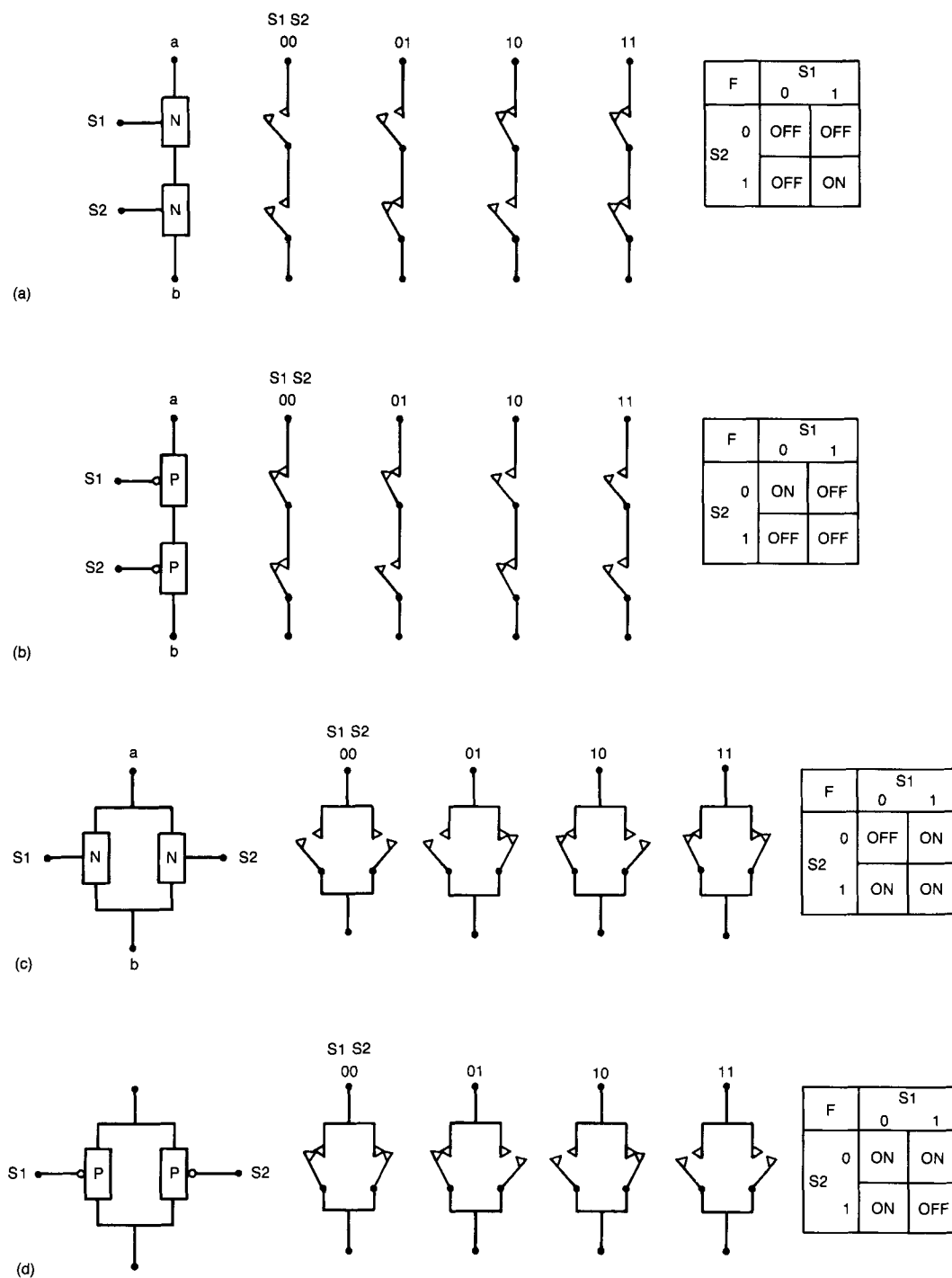


FIGURE 1.5. Series and parallel CMOS switch combinations

1.4.3

Fig. 1.4.3 shows two constructions for the term (p1 + p2)q. The first is realized by the circuit in Fig. 1.4.3(a), which is a series combination of a parallel combination of two PMOS transistors (p1 and p2) and an NMOS transistor (q).

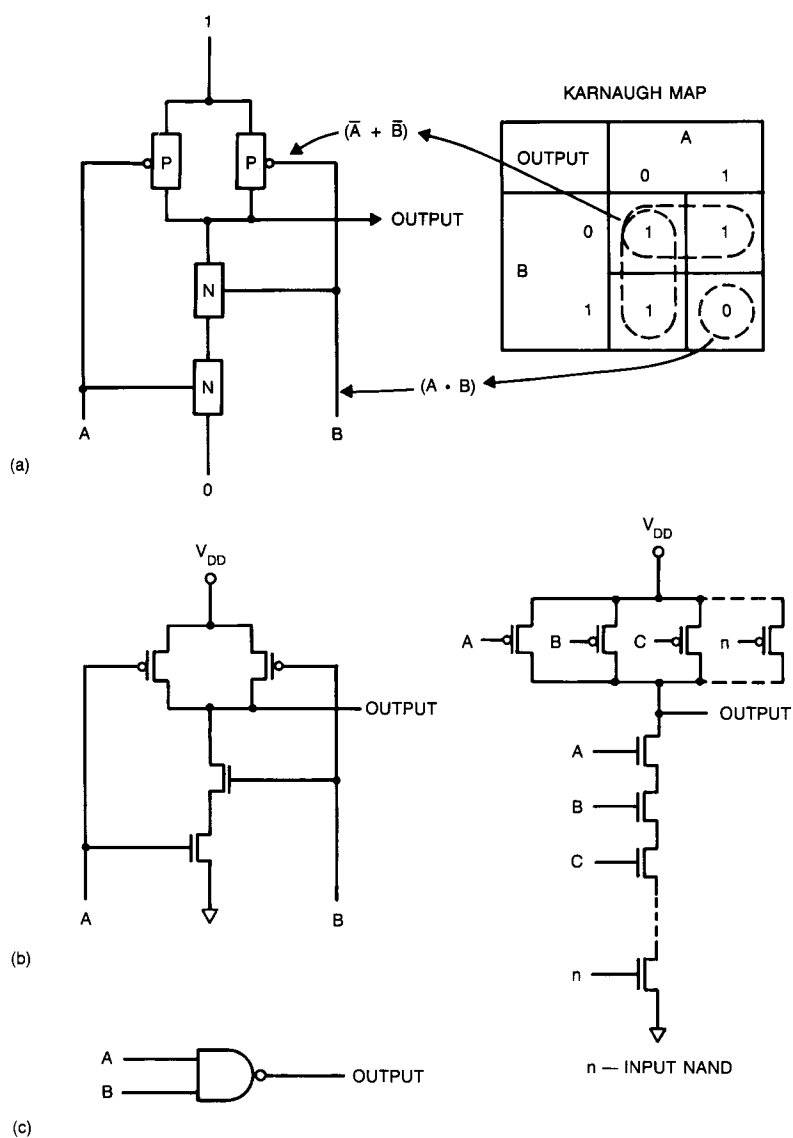


TABLE 1.2. NAND gate truth table

A INPUT	B INPUT	A N-SWITCH	B N-SWITCH	A P-SWITCH	B P-SWITCH	OUTPUT
0	0	OFF	OFF	ON	ON	1
0	1	OFF	ON	ON	OFF	1
1	0	ON	OFF	ON	OFF	1
1	1	ON	ON	OFF	OFF	0

are obtained automatically by the operation of the p-device. The p-structure is the logical dual of the n-structure. This property is used in most complementary CMOS logic gates (but not necessarily in dynamic gates or static gates that dissipate static power). The truth table and SWITCH states are shown in Table 1.2. By inspection, one may see that this implements the NAND function.

Some further points may be noted from this example. Firstly, note that for all inputs there is always a path from the '1' or '0' (V_{DD} or V_{SS} supplies) to the output and the full supply voltages appear at the output. The latter feature leads to a "fully restored" logic family. This simplifies the circuit design considerably. In comparison to nMOS, where the load and driver transistors have to be ratioed, the transistors in the CMOS gate do not have to be ratioed for the gate to function correctly. Secondly, there is never a path from the '1' to the '0' supplies for any combination of inputs (again in contrast to nMOS). As we will learn in subsequent chapters, this is the basis for the low static power dissipation in CMOS. The circuit and logic schematics for the 2-input NAND gate are shown in Fig. 1.6b and Fig. 1.6c. Note that larger input NAND gates are constructed by placing one N-SWITCH in series on the n side and one P-SWITCH in parallel for each additional input to the gate.

1.4.4 The NOR gate

A 2-input NOR gate is shown in Fig. 1.7a. It is composed from sections introduced in Fig. 1.5b and Fig. 1.5c, according to the Karnaugh map. Note that the N and P switch combinations are the dual or complement of that for the NAND gate. The truth table is shown in Table 1.3. This implements a logical NOR operation. The corresponding schematics are shown in Fig. 1.7b and Fig. 1.7c. In comparison to the NAND gate, extra inputs are accommodated in the NOR structure by adding N-SWITCHES in parallel and P-SWITCHES in series with the corresponding switch structures.

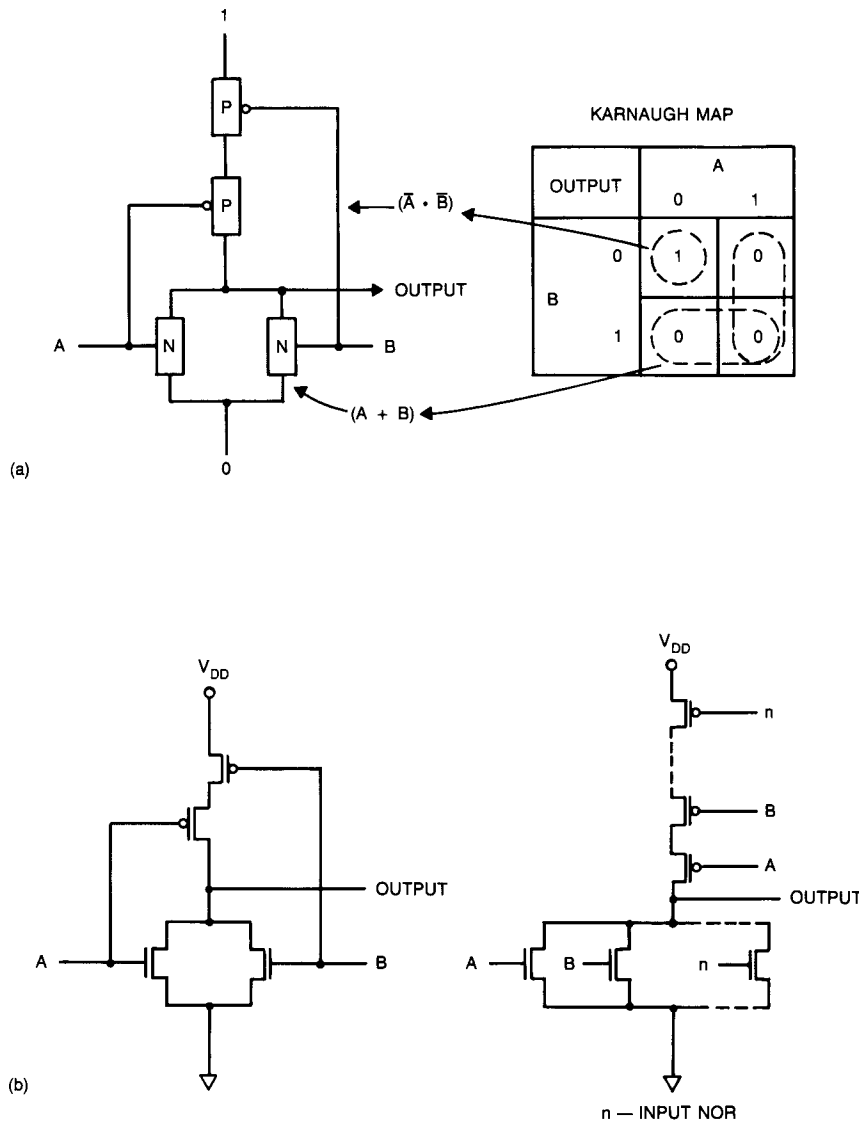


FIGURE 1.7. A CMOS NOR gate

TABLE 1.3. NOR gate truth table

A INPUT	B INPUT	A N-SWITCH	B N-SWITCH	A P-SWITCH	B P-SWITCH	OUTPUT
0	0	OFF	OFF	ON	ON	1
0	1	OFF	ON	ON	OFF	0
1	0	ON	OFF	OFF	ON	0
1	1	ON	ON	OFF	OFF	0

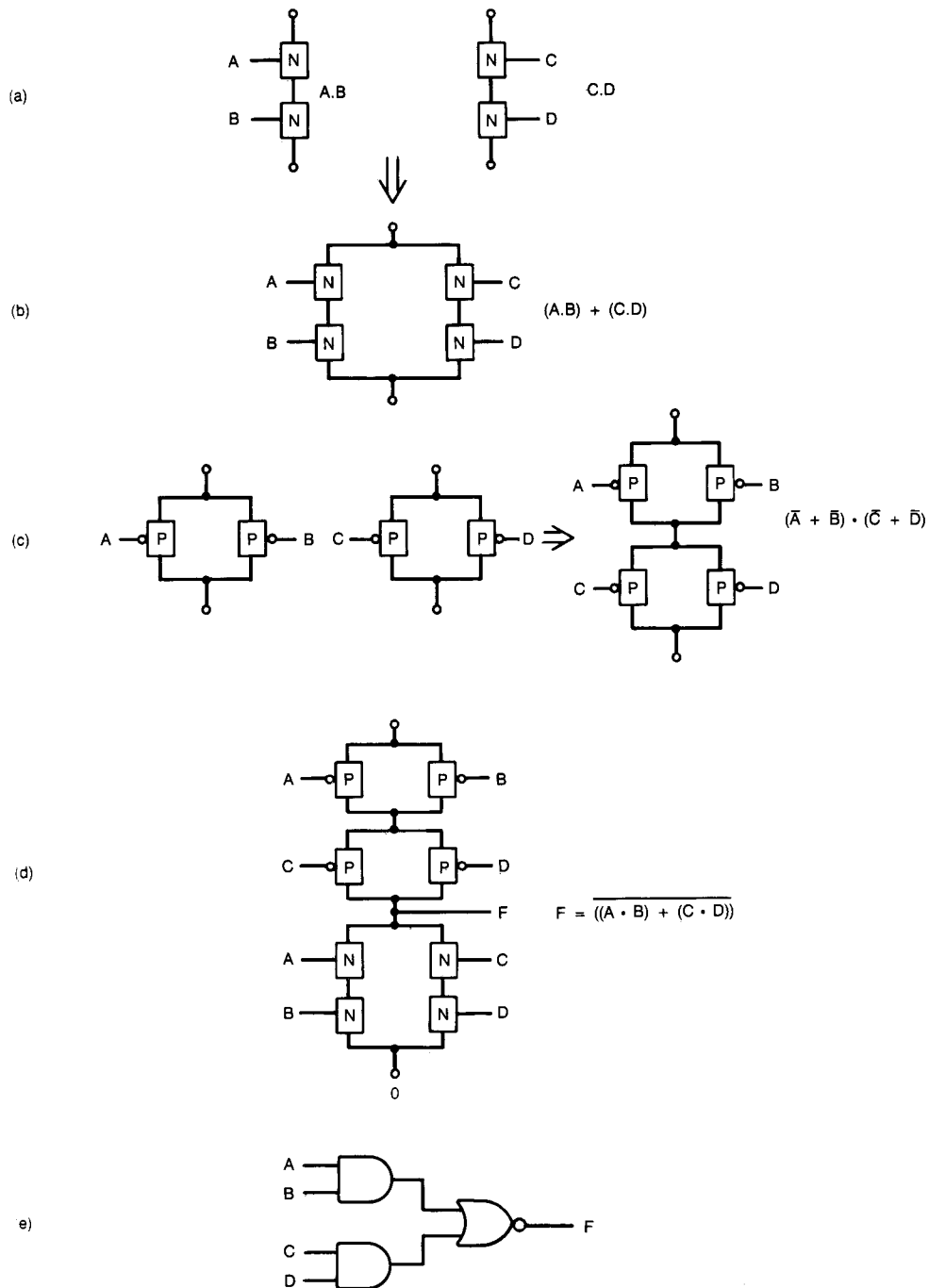
1.4.5 Compound gates

A compound gate is formed by using a combination of series and parallel switch structures. For example, the derivation of the switch connection diagram for the function $F = \overline{((A.B) + (C.D))}$ is shown in Fig. 1.8. The decomposition of this function and generation of the diagram may be approached as follows. For the n-side take the uninverted expression $((A.B) + (C.D))$. The AND expressions $(A.B)$ and $(C.D)$ may be implemented by series connections of switches as shown in Fig. 1.8a. Now taking these as subentities and ORing the result requires the parallel connection of these two structures. This is shown in Fig. 1.8b. For the p-side we invert the expression used for the n-expansion, yielding $((\bar{A} + \bar{B}) \cdot (\bar{C} + \bar{D}))$. This suggests two OR structures, which are subsequently connected in series. This progression is evident in Fig. 1.8c. The final step requires connecting one end of the p-structure to '1' (V_{DD}) and the other to the output. One side of the n-structure is connected to '0' (V_{SS}) and the other to the output in common with the p-structure. This yields the final connection diagram (Fig. 1.8d). The schematic icon is shown in Fig. 1.8e, which shows that this gate may be used in a 2-input multiplexer. If $C = \bar{B}$, then $F = \bar{A}$ if B is true, while $F = \bar{D}$ if B is false.

The Karnaugh map for a second function $F = \overline{((A + B + C) \cdot \bar{D})}$ is shown in Fig. 1.9a on page 16. The sub-function $(A + B + C)$ is implemented as three parallel switches. This structure is then placed in series with a switch with D on the input. The p-function is $(\bar{D} + \bar{A}.\bar{B}.\bar{C})$. This requires three switches in series connected in turn in parallel with a switch with D on the input. The completed gate is shown in Fig. 1.9b on page 16. In general, CMOS gates may be implemented by analyzing the relevant Karnaugh map for both n- and p-logic structures and subsequently generating the required series and parallel combinations of transistors.

1.4.6 Multiplexers

Complementary switches may be used to select between a number of inputs, thus forming a multiplexer function. Fig. 1.10a on page 17 shows a connection diagram for a 2-input multiplexer. As the

FIGURE 1.8. Construction of function $F = ((A \cdot B) + (C \cdot D))$

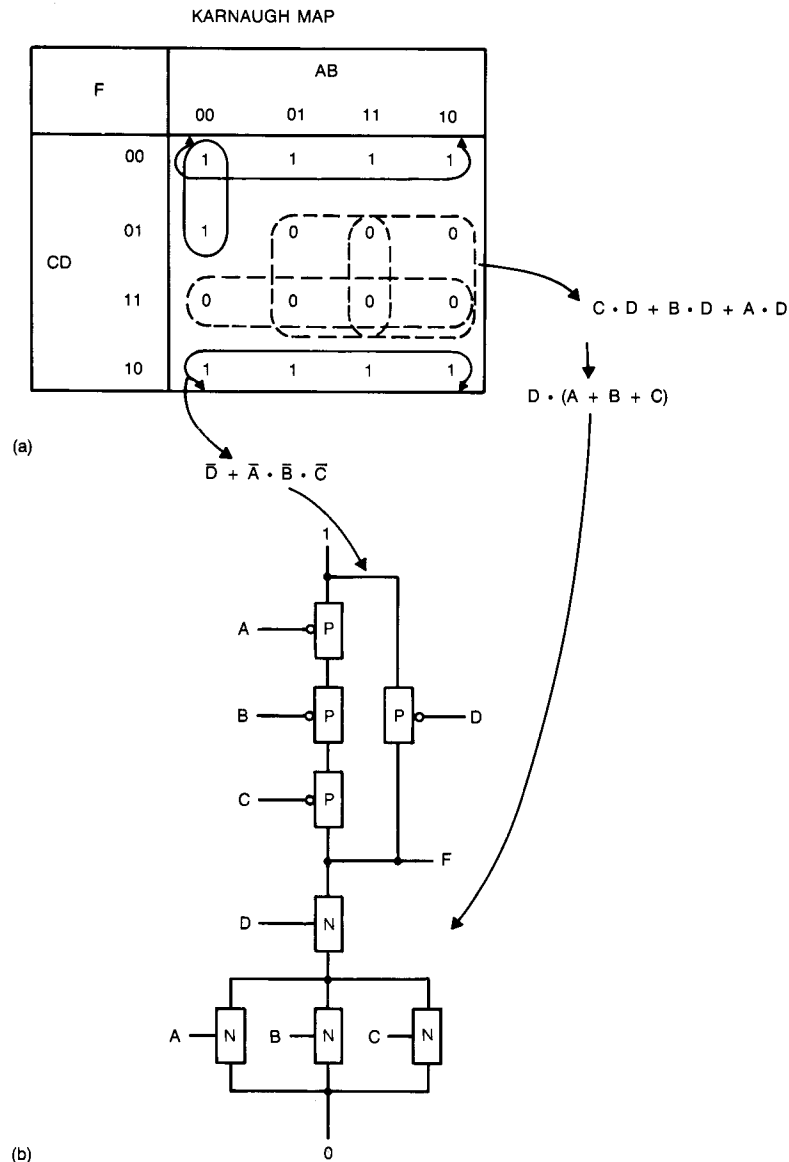


FIGURE 1.9. Construction of function
 $F = ((A + B + C) \cdot D)$

switches have to pass '0's and '1's equally well, complementary switches with n- and p-transistors are used. The truth table for the structure in Fig. 1.10 is shown in Table 1.4. The complementary switch is also called a transmission gate or pass gate (complementary). A commonly used circuit symbol for the transmission gate is shown in Fig. 1.10b. The multiplexer connection in terms of this symbol and transistor symbols is shown in Fig. 1.10c.

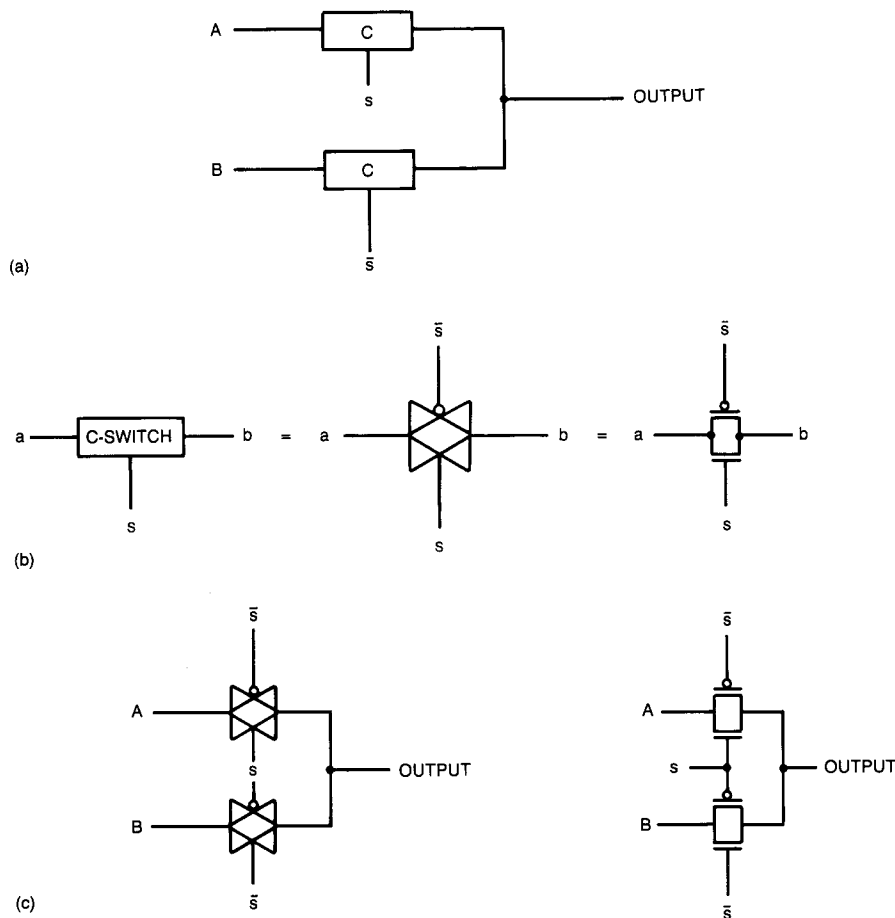


FIGURE 1.10. A CMOS 2-input multiplexer

1.4.7 Memory

We have now constructed a sufficient set of CMOS structures to enable a memory element to be constructed. A simple flip-flop using one 2-input multiplexer and two inverters is shown in Fig. 1.11a.

TABLE 1.4. Two input multiplexer truth table

A	B	S	$\bar{S}$	OUTPUT
X	0	0	1	0(B)
X	1	0	1	1(B)
0	X	1	0	0(A)
1	X	1	0	1(A)

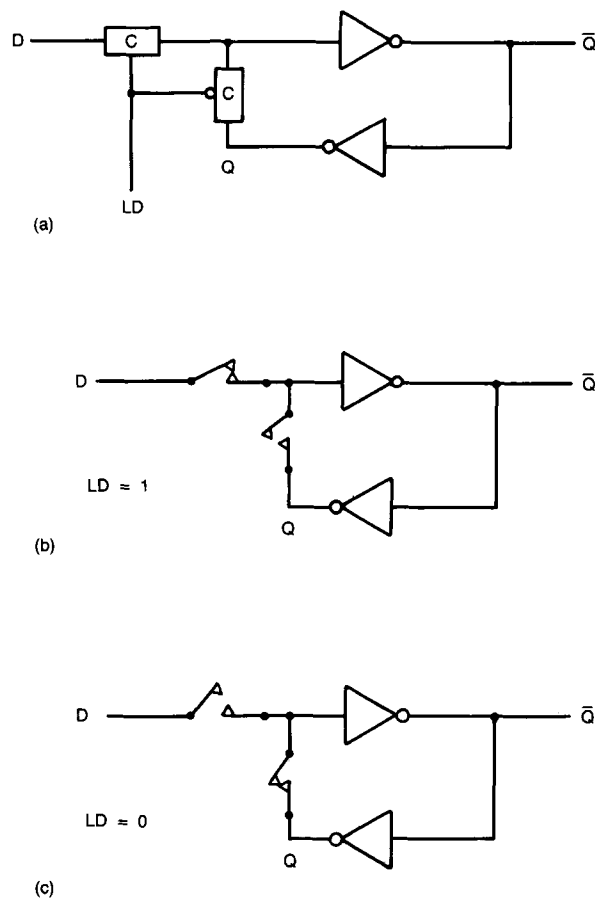


FIGURE 1.11. Connection of components for a simple CMOS flip-flop

When $LD = '1'$, $\bar{Q}$ is set to $\bar{D}$ and Q is set to D (Fig. 1.11b). When LD is switched to '0', a feedback path around the inverter pair is established (Fig. 1.11c). This causes the current state of Q to be stored. While $LD = '0'$ the input D is ignored.

1.5 Alternate circuit representations

In this section we will examine some alternate representations for the circuits developed so far. Generally, a design can be expressed in terms of behavioral, structural, and physical properties.

1.5.1 Behavioral representation

A behavioral representation describes how a particular design should respond to a given set of inputs. In Section 1.4 the behavior of a gate was defined in terms of its boolean function,

$$F = \overline{((A + B + C).D)}.$$

This is a technology independent behavioral specification at the logic level. No notion of how to implement this function is implied, nor is any speed performance implied. Higher levels of behavioral description are possible. For instance, an add operation may be summarized in a high level language by

$$\text{sum} = a + b$$

Here no method of addition is implied and the word length is assumed to be that of the machine. A further example of the behavior of the flip-flop designed previously is as follows:

```
IF(LD == 1)
THEN
  Q = D;
```

Note there may be some ambiguity associated with this style of behavioral representation. It could also represent a multiplexer without implying storage of state. Higher levels of behavioral specification can specify the types of registers involved in a design and the transfers that occur between them. Even less information about implementation is implied. At some stage it is possible to express behavior as an algorithm written in a high level language. The aim of most modern design systems is to convert some such specification into a system design in a minimum time and with maximum likelihood that the system will perform as desired.

1.5.2 Structural representation

A structural specification specifies how components are interconnected to perform a certain function (or achieve a designated behavior). We will use as an example of a complete structural description language, MODEL, a language conceived by Lattice Logic Ltd. [Latt82]. The specification for the inverter is

```
Part   inv  (in)-> out
      Nfet  out  in  vss
      Pfet  out  in  vdd
End
```

The first line declares a part called `inv` followed by a list of inputs, in this case `in`. The outputs appear on the other side of the symbol `->`, in this case `out`. Following this is a list of transistors with their type and connections in the form

```
Transistor-type  drain-conn  gate-conn  source-conn
      Nfet              out         in         vss
```

Thus the first statement describes an n-transistor with `drain = out`, `gate = in`, `source = vss`. The second statement describes a p-transistor with `drain = out`, `gate = in`, `source = vdd`.

The description for a 2-input NAND gate would be

```
Part nand2 (a,b) -> out
Signal i1
  Nfet i1 a vss
  Nfet out b i1
  Pfet out a vdd
  Pfet out b vdd
End
```

In this description the internal signal `i1` is declared by the keyword `Signal`. A diagram of this appears in Fig. 1.12a. It is worthwhile to compare this description with a possible behavioral description.

```
out = -(a&b)
```

or

```
out = (not (and a b ))
```

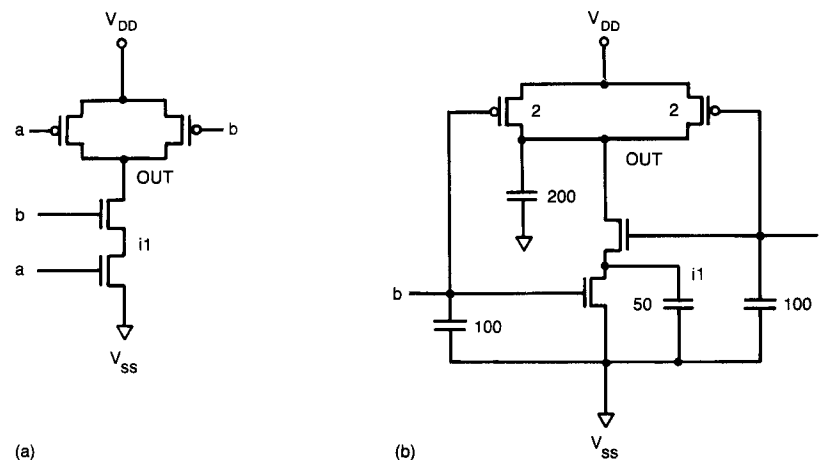


FIGURE 1.12. Graphical versions of structural descriptions for a CMOS NAND gate (schematics)

Note that
code frag
In the MC
descriptio
sizing tha
be added
result and
expanded

Here the c
the p-tran
size param
subsequen
formance o
ensures th
reference i
The struct
that affect
a circuit o
The speci

```
.SUBCKT
M1 I1
M2 OUT
M3 OUT
M4 OUT
CA A VS
CB B VS
COUT OUT
.ENDS
```

In this des
capacitanc
specified.
extra rout

Note that we can infer all of the transistor connections from these code fragments. However, the intermediate node *il* is "hidden." In the MODEL description, we may augment the purely structural description with some parameters such as capacitance and transistor sizing that will affect performance. Although notation for this can be added to the behavioral representation, rather baroque forms result and the simple elegance of the logical statement is lost. An expanded MODEL description might be

```
Part nand2 (a,b) -> out
Signal il
  Nfet  il a vss
  Nfet  out b il
  Pfet  out a vdd size = 2
  Pfet  out b vdd size = 2
  Capacitance il 50
  Capacitance  a 100
  Capacitance  b 100
  Capacitance out 200
End
```

Here the capacitance (in some units) has been specified. In addition, the p-transistor sizes have been modified according to some notional size parameter. This is shown in Fig. 1.12b. As we will learn in subsequent chapters, this type of information is crucial to the performance of CMOS circuits. In other words, the behavioral description ensures that the function may be correctly implemented but no reference is necessarily made to speed or other operational parameters. The structural description allows the specification of all components that affect performance. The circuit simulator SPICE [Nage75] uses a circuit description for the specification of transistor connectivity. The specification of the NAND gate might look like the following:

```
.SUBCKT NAND2 VDD VSS A B OUT
MN1 I1 A VSS VSS NFET W=8U L=4U AD=64P AS=64P
MN2 OUT B I1 VSS NFET W=8U L=4U AD=64P AS=64P
MP1 OUT A VDD VDD PFET W=16U L=4U AD=128P AS=128P
MP2 OUT B VDD VDD PFET W=16U L=4U AD=128P AS=128P
CA A VSS 50fF
CB B VSS 50fF
COUT OUT VSS 100fF
.ENDS
```

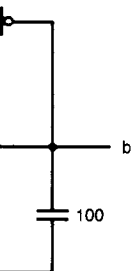
In this description the internal model in SPICE calculates the parasitic capacitances inherent in the actual device using the device dimensions specified. The capacitance statements in the above description add extra routing capacitance.

t of inputs,
the symbol
with their

source-conn
vss

drain =
t describes
e = vdd.

e keyword
worthwhile
description.



Defining a transmission gate in MODEL, we have

```
Part tg (a,c,cb) -> b
  Nfet a c b
  Pfet a cb b
End
```

We can now define the flip-flop (also called a D latch) as follows (a signal appended with "bar" is a complemented signal):

```
Part flipflop (in, ld, ldbar, q, qbar)
Signal a
  tg (in, ld, ldbar) -> a
  inv (a) -> qbar
  inv (qbar) -> q
  tg (q, ldbar, ld) -> a
End
```

Now we can use the flip-flop and other similarly constructed parts to hierarchically build larger and larger circuits.

A more familiar type of structural description is the schematic diagram shown in Fig. 1.13. Here we have the graphical hierarchy

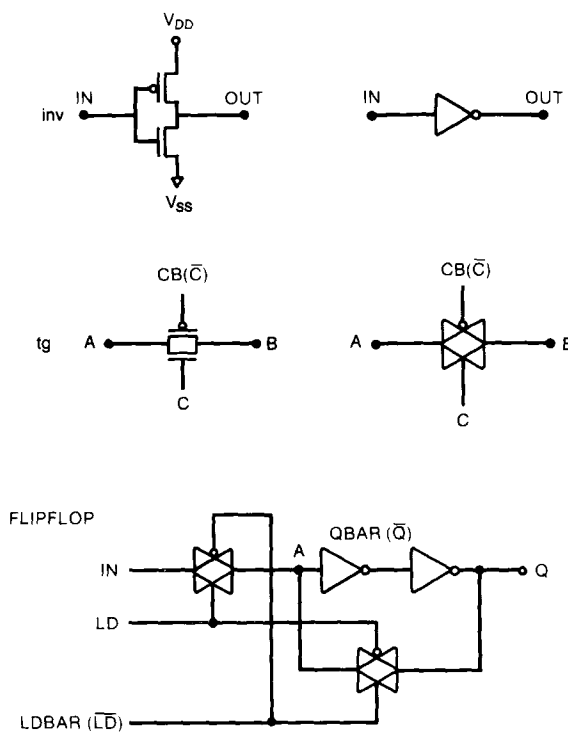


FIGURE 1.13. Schematic representation of CMOS flip-flop

of parts corresponding to the MODEL descriptions that we have developed. The specifications range from the circuit level (transistors), to the logic level (gates), to the functional block level (memory and collections of gates).

To all intents, both types of description are interchangeable, with preference for use dependent on the user. The schematic description is more immediately descriptive — “a picture is worth a thousand words.” However, the language representation has some particular benefits, especially if the high level constructs such as looping, conditionals, and parameter passing are available. For instance, if we wanted to change the size of transistors in the inverter we might say

```
Part inv (in) [n] -> out
      Nfet out in vss      size = n
      Pfet out in vdd      size = 2*n
End
```

Here, n is a parameter passed to the inverter description to specify the size of the transistors. Emerging design systems show promise of dealing with language and graphical aspects of a design in a consistent fashion.

1.5.3 Physical representation

The physical specification for a circuit is used to define how the particular part has to be constructed to yield a specific structure and hence behavior. In an IC process, the lowest level of physical specification is the photo-mask information required by the various processing steps required on the fabrication process (see Chapter 3). At this stage, we will not dwell on these details but propose a simple model for the physical nature of a CMOS circuit, assuming that a program can translate our notation directly to the format needed for fabrication.

A typical physical representation for a transistor would consist of two rectangles representing the lithography required to fabricate the transistor. Precise “design rules” specify the size of each rectangle. In addition, for each different process, these rules change and the corresponding dimensions change — not necessarily linearly. Rather than try and remember these rules, we will use a single symbol to represent a transistor in a non-metric format. We will retain a form that reflects the physical nature of the transistor. The physical symbol for an n-transistor is shown in Fig. 1.14a and Plate 1. This mirrors the physical realization in which at least two process levels are overlaid. As we have seen, the gate connection is on one layer of the process and the source and drain on another layer. A similar

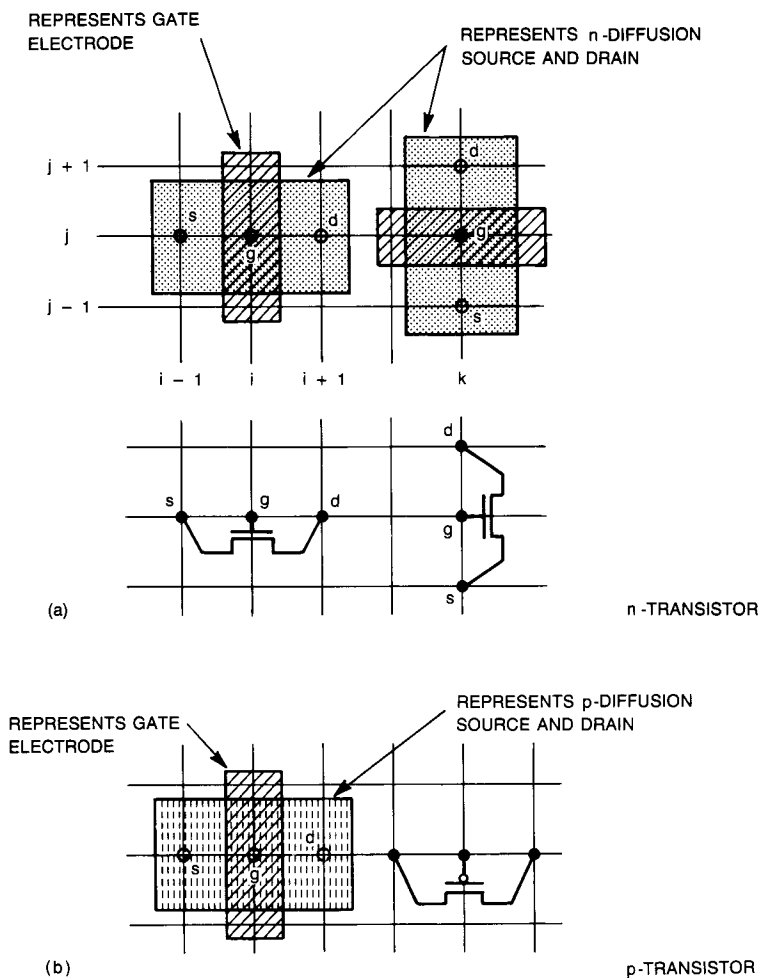
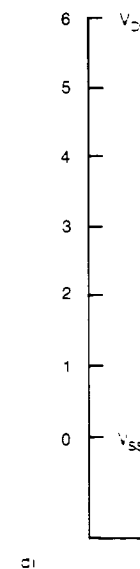
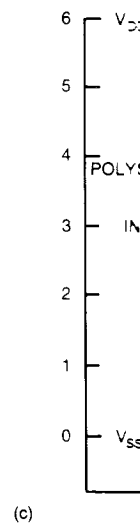


FIGURE 1.14. Physical symbols for transistors and simple circuits

symbol is used for the p-transistor, as shown in Fig. 1.14b and in color in Plate 1. Here, a “horizontal” transistor is shown. These symbols are overlaid on a grid. The transistor symbol occupies three grid points. The center grid point is the connection point for the gate of the transistor. The grid point to the right (or above) is the drain and the grid point to the left (or below) is the source. These two terminals are interchangeable. The schematic symbols are also shown for the n- and p-transistors in terms of grid connection points.

A symbolic layout for an inverter may be constructed using these symbols. It is substantially the same as the schematic but we have had to be careful about the layers in which connections have been made. We have “wires” on four layers. The interaction of these



layers is s
may be m
not be ma
a “contact
A com
1.14c. Tr

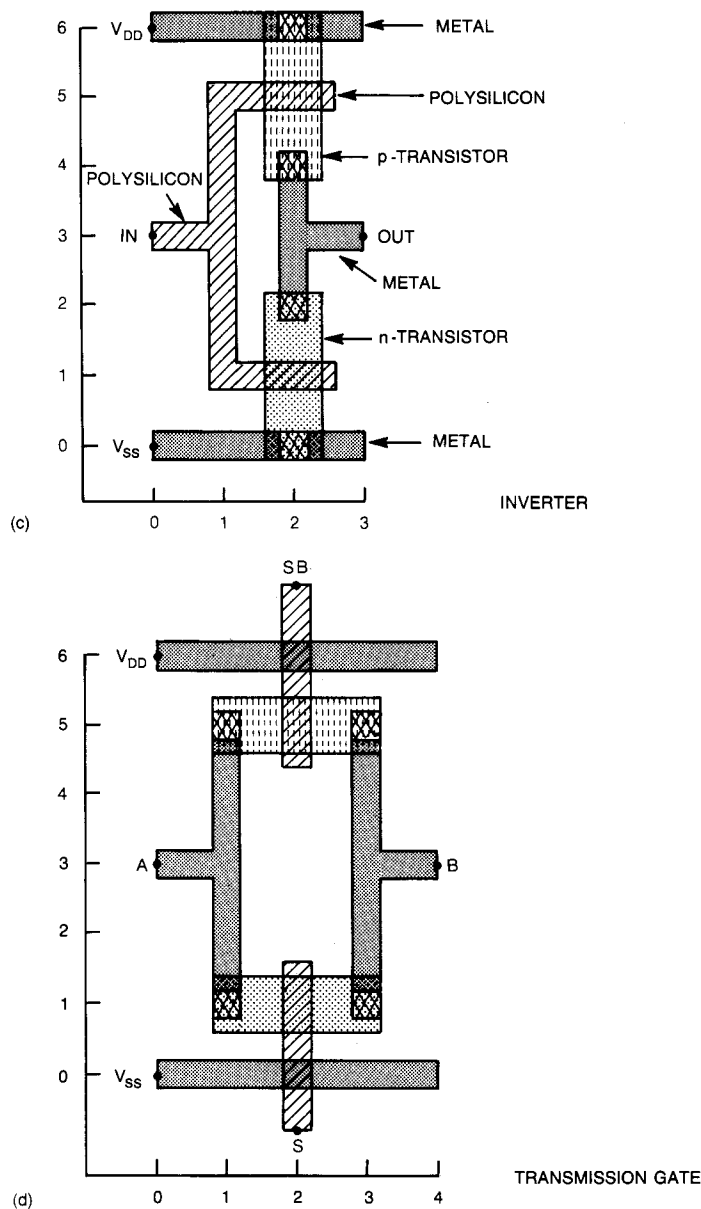


FIGURE 1.14. (Continued)

layers is summarized in Table 1.5. OK denotes that a connection may be made, while an X designates that a direct connection may not be made between the two layers. Any off-diagonal OK requires a "contact" (C) to connect the two layers.

A completed symbolic layout for the inverter is shown in Fig. 1.14c. The symbolic layout for a transmission gate is shown in

TABLE 1.5. Physical layer interactions

	n-DIFFUSION	p-DIFFUSION	POLYSILICON	ALUMINUM
n-diffusion	OK	X	Transistor	OK (C)
p-diffusion	X	OK	Transistor	OK (C)
polysilicon	Transistor	Transistor	OK	OK (C)
aluminum	OK (C)	OK (C)	OK (C)	OK

Fig. 1.14d. Color versions of these are found in Plate 1. This may also be expressed in the form of a language description. The following represents the transmission gate in the ICDL language (see Chapter 7):

```

begin tg
t1:    device n (2,1) or=east
t2:    device p (2,5) or=east
      wire alum (0,0) (4,0)
      wire alum (0,6) (4,6)
      wire poly (2,-1) (2,1)
      wire poly (2,7) (2,5)
      wire alum (1,1) (1,5)
      wire alum (3,1) (3,5)
      wire alum (0,3) (1,3)
      wire alum (3,3) (4,3)
      contact md (1,1)
      contact md (3,1)
      contact md (1,5)
      contact md (3,5)
end

```

It consists of transistor, contact, and wire statements with type qualifiers and grid coordinates. Note that although this is an abridged form of the mask information needed to fabricate the transmission gate, it is substantially larger than the corresponding MODEL structural description. Using a number of such structures, we can construct a physical sub-assembly that constitutes a flip-flop according to Fig. 1.15. Fig. 1.15a shows a simple physical abutment of the two transmission gate cells and two inverter cells. The V_{SS} and V_{DD} supplies have been arranged to feed across the bottom and top of the cells. A feedback line in metal connects the Q output to one side of the input multiplexer. In Fig. 1.15b, the internal circuit details of the cells are displayed in the form of schematic circuit symbols. Finally, Fig. 1.15c shows the symbolic layout representation, which is identical in topology to that shown in Fig. 1.15b. Plate 2 illustrates this example in color.

To a large extent, most CMOS IC design involves the steps illustrated in the preceding sections. Once a behavior is defined, the logic corresponding to that behavior is designed. That leads to

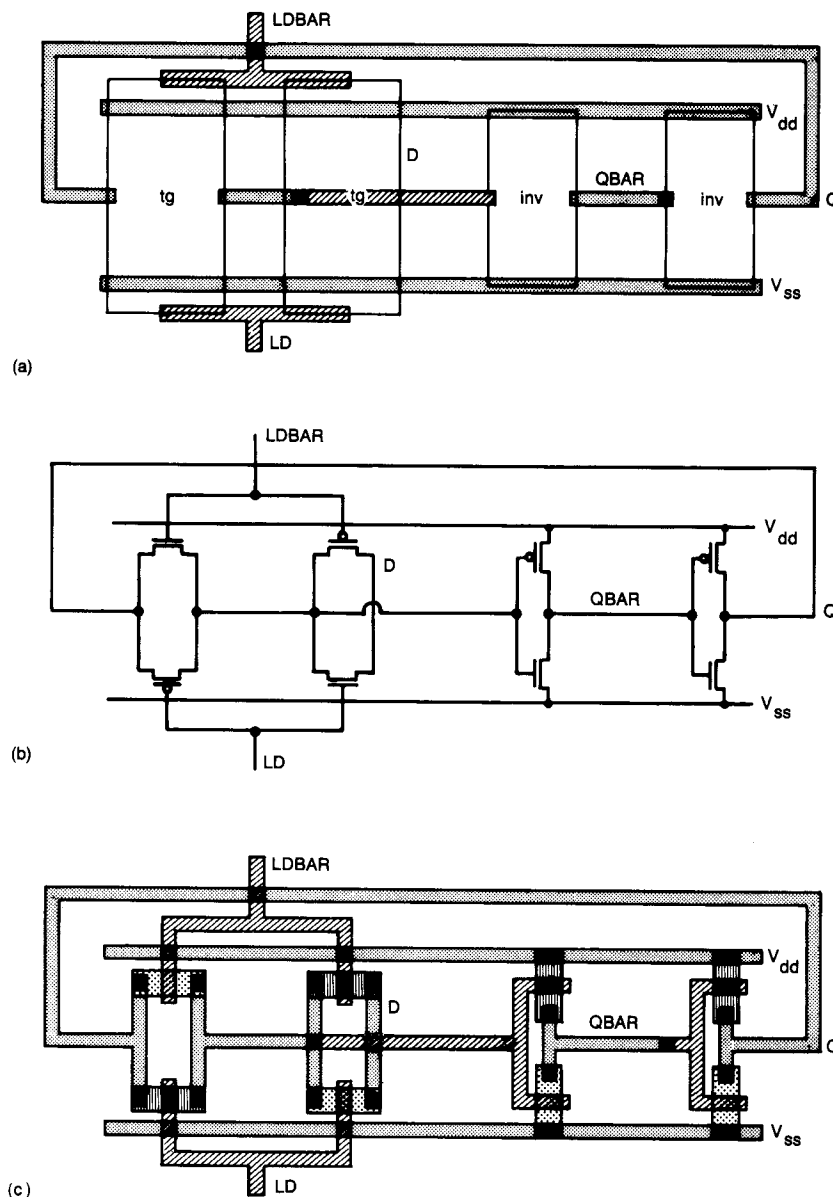


FIGURE 1.15. Physical construction of a CMOS flip-flop

a transistor circuit description. Finally, a layout may be designed for the particular logic function. Quite often, frequently used logic structures may be designed at the layout level and placed in a library. These library elements may then be assembled as illustrated in Fig. 1.15 to build more complex structures. An alternative to this approach is to compose primitive elements such as flip-flops from individual transistors. This yields more efficient layouts. Methods for designing in this manner will be treated in subsequent chapters.

1.6 CMOS–nMOS comparison

Many designers may have been introduced to VLSI design through nMOS design. In order to illustrate the salient features of CMOS, a

TABLE 1.6. CMOS–nMOS quick summary

CMOS	nMOS
(i) Logic Levels	
• Fully restored logic, i.e., output settles at V_{DD} or V_{SS} (GND).	• Output does not settle at V_{SS} (GND) — hence degraded noise margin.*
(ii) Transition Times	
• Rise and fall times are of the same order.	• Rise times are inherently slower than Fall times.*
(iii) Transmission Gates	
• Transmission gate passes both logic levels well. The output of transmission gate can be used to drive the input of other transmission gates.	• Pass transistor transfers logic '0' well but logic '1' is degraded. Pass transistor cannot drive the gate of a second pass transistor.
(iv) Power Dissipation	
• Almost zero static power dissipation. However power is dissipated during logic transition.	• With output of a given gate = '0' power is dissipated in the circuit in addition to power dissipated during logic transitions.*
(v) Precharging Characteristics	
• Both n-type and p-type devices are available for precharging a bus to V_{DD} and V_{SS} . Nodes can be charged fully to V_{DD} or alternatively to V_{SS} in a short time.	• With enhancement mode transistor the best one can do (with normal clocking) is to charge a bus to $(V_{DD} - V_t)$. Generally use of bootstrapping or hot clocking is needed to precharge to V_{DD} .
(vi) Power Supply	
• Voltage required to switch a gate is a fixed percentage of V_{DD} .	• Somewhat dependent on supply voltage.
• Variable range 1.5 to 15 volts.	• Fixed.
(vii) Packing Density	
• Require 2N devices for N inputs for complementary static gates. Less for dynamic gates.	• Require $(N + 1)$ devices for N inputs.
(viii) Pull-up to Pull-down Ratio	
• Load-to-driver device ratio is typically 1:1 or 2:1.	• Load-to-enhancement-driver ratio is typically 4:1 to optimize the logic '0' output level and minimize current consumption.
(ix) Layout	
• CMOS encourages regular layout styles.	• Depletion load and different driver transistor sizes inhibit layout regularity.

"quick sum
stressed th
vary wide
logic style
not valid
main poin
fully resto
is at '1' or
for a fully

1.7 S

This chap
developed
to a basic
and metho
this book v

1.8 E

1.1 Design
Draw

1.2 $F = \bar{A}$
Design

1.3 Design

1.4 A 4-in
busses
which
why o

1.5 Using
layout
proble

1.6 Design
elemen
sequen

“quick summary” is presented in Table 1.6. However, it should be stressed that this is a broad overview and individual points may vary widely in importance. The comparisons are true of ratioed logic styles for nMOS and those items marked with an asterisk are not valid for ratioless nMOS logic design (dynamic circuits). The main points to note are that the output logic levels of CMOS are fully restored, a CMOS gate consumes no DC power when the output is at ‘1’ or ‘0’ level, and 2N devices are required for an N-input gate for a fully complementary gate.

1.7 Summary

This chapter introduced a simple model for an MOS transistor and developed logic that uses p-transistors and n-transistors. This led to a basic discussion of the various levels of representation of circuits and methods of composing these representations. The remainder of this book will elaborate on the material introduced in this chapter.

1.8 Exercises

- 1.1 Design a 4-input NAND gate using CMOS switch elements. Draw the full transistor circuit for the function.
- 1.2 $F = \overline{AB + BC + AC}$ implements a complemented carry function. Design a complementary CMOS gate to perform this function.
- 1.3 Design a 3-input OR gate. To what conclusions do you come?
- 1.4 A 4-input multiplexer structure is needed to multiplex four busses to a register in a microprocessor. Show two ways in which this may be implemented. Can you think of any reasons why one method is preferable to the others?
- 1.5 Using graph paper and colored pencils, complete a symbolic layout for the gates designed in Exercises 1.1, 1.2, and 1.3. What problems do you encounter?
- 1.6 Design and complete a symbolic layout for a CMOS memory element other than that shown in Fig. 1.11. Include waveform sequencing required for operation.