DIGITAL CIRCUITS AND MICROPROCESSORS

Herbert Taub

DIGITAL CIRCUITS AND MICROPROCESSORS

Herbert Taub

Professor of Electrical Engineering
The City College of the
City University of New York

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Hamburg
Johannesburg London Madrid Mexico Montreal New Delhi
Panama Paris São Paulo Singapore Sydney Tokyo Toronto

This book was set in Times Roman by A Graphic Method Inc.
The editors were Frank J. Cerra and Madelaine Eichberg;
the production supervisor was Leroy A. Young.
The drawings were done by J & R Services, Inc.
The cover was designed by Nicholas Krenitsky.
Fairfield Graphics was printer and binder.

DIGITAL CIRCUITS AND MICROPROCESSORS

1 2 3 4 5 6 7 8 9 0 FGFG 8 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Taub, Herbert, date
Digital circuits and microprocessors.

(McGraw-Hill series in electrical engineering.
Computer engineering and switching theory)
Includes index.
1. Digital electronics. 2. Microprocessors.
I. Title. II. Series.
TK7868.D5T36 621.3815'3 81-3729
ISBN 0-07-062945-5 AACR2

PREFACE

This book is an introductory text suitable for a one-semester course. It covers all the basic principles of digital systems and logic design and provides as well an introductory presentation to microprocessor and microprocessor-based systems. The present and growing importance of microprocessors makes it important that these versatile components be introduced into an engineering or computer science curriculum at the earliest opportunity.

The subject of logical variables and boolean algebra is covered in Chapter 1. Logic gates and logical connectives are described and analyzed. The binary number system is introduced here principally to allow some systemization of truth tables. Chapter 2 deals with the standard forms of logic functions and with Karnaugh maps. Chapter 3 considers basic combinational circuits including decoders, encoders, code converters, multiplexers and demultiplexers. It is emphasized that all these components are available as integrated-circuit chips, and a brief discussion is provided of *families* of integrated circuits. Conventions dealing with the characterization of control terminals on chips are explained, and this discussion leads to a consideration of the convention of *mixed logic* which, in certain applications, is gaining popularity. Examples are given of the newer logic symbols which are presently being introduced for combinational and other components. The basic storage element, the flip-flop, is examined in some detail in Chapter 4. A careful distinction is drawn between a latch and a flip-flop. The characteristics required of a flip-flop in order that it be able to function properly in a synchronous system are examined. Assemblages of flip-flops into storage registers, shift registers, and counters are also considered. Again, in this chapter it is emphasized that the components described are available as integrated-circuit packages and examples of such devices are described. Chapter 5 deals with the subject of arithmetic operations, principally addition. The look-ahead carry principle is explained and analyzed. Memory is the subject of Chapter 6. This chapter covers the RAM, both static and dynamic, the ROM, the PLA, serial memories, and memories for bulk storage. Also described are timing considerations in reading from and writing into memories. Chapter 7 introduces the subject of the analysis and design of sequential systems, both synchronous and fundamental mode. The concepts of flow diagrams, state diagrams, and tables are presented, and also described are procedures for eliminating redundant states. While there is some discussion of sequential circuits in Chapter 4 in connection with shift registers and counters, the formal organized and systematized presentation is given in Chapter 7.

The material on controllers in Chapter 8 is written with a view toward microprocessors. A microprocessor consists of a number of storage and working registers, and ALU, and a controller. The controller appers to be endowed with uncanny abilities. It does exactly the right thing at the right time in precisely the right sequence and, having completed one task, proceeds unerringly to the next. Truly enough, the controller is nothing more than a special-purpose sequential circuit involving no difficult concepts. Still, to the beginning student, the vagueness associated with the controller is inevitably a source of uneasiness. It is very difficult to accept the generalizations with which controllers are described when there is no concrete and specific example that can serve as a model. Chapter 8 is written in a manner which will, hopefully, provide some reassurance to the uninitiated. It makes clear at the outset that all of the digital operations which are possible are relatively few and fundamentally simple, and that all are executed in response to the enabling of a gate or set of gates. Next there is presented the architecture of a very simple system which then requires a controller to be effective. A controller is designed in detail, first by using the procedures of Chapter 7 which yield a sequential system with a minimum number of states. Next this initial controller is replaced by a shift-register controller. The shift-register controller uses more hardware but has the great merit that the details of its operation are easily apparent and that required modifications for the purpose of elaboration can be added almost by inspection. Finally a controller is designed in detail to serve a very simple minded (4-instruction) "computer." The design makes clear in an entirely unambiguous manner how a controller can be made to modify its behavior in response to an *instruction*. Also in this chapter the student encounters the concepts of the *program counter*, the *memory address register*, and the *instruction register*. The reader sees, in simple form, the overall architecture which characterizes a microprocessor as well as the typical content of a memory which holds instruction and data for a stored program computation.

Chapter 9 is also written with an eye toward microprocessors. Here there is presented the architecture of a simple (16-instruction) computer. The structure provides a preview of the type of instructions to be encountered in more sophisticated systems. The *jump* and *subroutine call* instructions are presented and some simple programs are written in an assembly language. Also in this chapter the subject of control by *microprogramming* is presented and simple examples given.

Some authors invent a hypothetical microprocessor to have an example through which to introduce the subject. This procedure has the unfortunate feature that the student misses the exposure to a real device while the hypothetical microprocessor eventually turns out to be very nearly as complicated as a real component. Other authors undertake to include a number of real microprocessors in their descriptions and explanations. This approach, all too often, leads to vague generalizations. A third widely used approach, also used in this text, is to concentrate on a single real device. This method allows the analysis to be pointed and specific and, furthermore, a good familiarity with one device provides a background that allows an easy understanding of other devices. In this text, the microprocessor selected for study is the 8080 which is widely known and used and is highly regarded. Even though the 8080 has been updated by the 8085, we have stayed with the 8080 precisely because it is somewhat less sophisticated and, therefore, better suited to an introductory presentation. The 8080, its architecture, instructions, and programming is the subject of Chapter 10. Chapter 11 is devoted entirely to input-output operation of the 8080.

There is some more material in the text than can be covered conveniently in one semester. From the author's prejudicial point of view an effective way of employing the book is to use it for one full semester and for about one fifth of a second semester. Thereafter, for the remainder of the second semester, a new text should be adopted that covers microprocessors and microcomputers generally and in greater depth. On the other hand, it is entirely feasible to cover the book in one semester by omitting some material which is not essential in a first approach. Candidates for omission include the following sections: 1.17, 1.25, 1.26, 2.12, 5.10 through 5.12, 6.10 through 6.17, 7.6 through 7.9, 7.11 through 7.19, 8.12 and 8.13.

A large number of homework problems have been provided. A solutions manual is available that instructors can obtain from the publisher. An answer book is also available.

I am grateful to Professor Mansour Javid, chairman of the Department of Electrical Engineering at the City College of New York, who read a large part of the manuscript and made many valuable suggestions. Mr. Jay Lewis Taub provided a great deal of very effective assistance in the preparation of the manuscript and I am pleased to express to him my most sincere thanks. Mrs. Joyce Rubin's skillful typing of the manuscript is appreciated.

Herbert Taub

CONTENTS

CHAPTER
ONE
ALGEBRA OF LOGICAL VARIABLES

1.1 VARIABLES AND FUNCTIONS

We are familiar with the concept of a *variable* and with the concept of a *function* of a variable. The *field* of a variable, i.e., the range of values which can be assumed by a variable x, can by specified in a limitless number of ways. For example, x may range over all the real numbers from minus to plus infinity; or x may be restricted to the range from -17 to -4; or x may be restricted to the positive integers from 1 to 10; and so on.

A *function* is a *rule* by which we determine the value of a second (dependent) variable y from the (independent) variable x, the dependency of y on x being written $y = f(x)$. Thus, for example, suppose we intend that y is to be determined from x through the rule that x is to be multiplied by itself, that this product is to be multiplied by 5, and that thereafter 3 is to be added. We would then express the functional relationship between x and y by the equation $y = 5x^2 + 3$. In this simple example we determined y by applying the mathematical processes of multiplication and addition. However, when the number of allowable values of x is finite, it is possible to specify a function simply by making a *table* in which y is given for each value of x. When the number of possible values for x is small, it may well be feasible and most convenient to use such a table. Consider that in the example referred to above ($y = 5x^2 + 3$) we restrict x to the integral values $x = 0, 1, 2,$ and 3. Then, as is indicated in Fig. 1.1-1, the functional relationship between y and x can be specified in tabular form.

By an easy extension of these elemental ideas, it is clear that the variables, dependent and independent, need not be numerical. For example, let the independent variable x have as its field the colors of the traffic light at an intersection, and let the dependent variable y represent the expected behavior of a mo-

x	$y = f(x)$
0	3
1	8
2	23
3	48

Figure 1.1-1 A numerical function.

x	$y = f(x)$
Green	Continue
Amber	Slow down
Red	Stop

Figure 1.1-2 A functional relationship.

torist approaching the intersection. Then the functional relationship between x and y is as given in Fig. 1.1-2. The values which can be assumed by x are expressed by the declarative statements "the light is green" or "the light is amber" or "the light is red." Similarly the values which can be assumed by y are "the motorist should continue," etc.

1.2 LOGICAL VARIABLES

A *logical* variable is a variable which has three distinctive properties:

1. The logical variable may assume one or the other of only *two* possible values.
2. The values are expressed by declarative statements, as in the traffic-light example given above.
3. The two possible values expressed by the declarative statements must be such that, on the basis of human reason, i.e., on the basis of logic, they are *mutually exclusive*.

Although, as noted, the variable need not have numerical significance, there is no reason to preclude situations in which the variable does. Thus the variable x may have the two and only two, alternative, mutually exclusive values expressed by the statements "the value of x is 7" and "the value of x is 13." Other properties of the logical variable will appear in the following discussion, in which we return to the example of the traffic light.

Suppose that we postulate that the traffic light can be only green or red. We exclude the possibility that the light may be amber and exclude as well the possibility of an interval when the light is changing and neither green nor red is showing. Then in this case the variable x in the table of Fig. 1.1-2 is a logical variable. Either we shall have that "the light is green," which we can represent as x = green, or we shall have that x = red. Note especially that because of the mutual exclusivity, if we want to indicate that x = red we can indicate it either in that way or by writing x = *not* green. In a simpler notation, the "not" is represented by placing a bar over the value. Thus x = not green can be written $x = \overline{\text{green}}$. Finally we have that $x = \overline{\text{green}}$ = red.

1.3 VALUES FOR A LOGICAL VARIABLE

In the general case of an arbitrary type of variable, say the type of variable which assumes numerical values, the variables may represent anything. Thus x and y may represent temperature or pressure or distance or velocity or time, etc. In considering the functional relationship between variables from a mathematical point of view we have no interest in what is represented by the variables. Thus from the equation $y = 5x^2 + 3$ we have the result that $y = 8$ when $x = 1$ quite independently of what x and y may stand for. The values which can be assumed by the variables are the same in the sense that in both cases the values are numbers.

In the same way, let us assign to the two possible values of our logical variable two names, so that we can consider a variable independently of what it may represent. Any two readily distinguishable names would be suitable, but it would also be useful to have names which convey the notion of mutual exclusivity. For this reason such names suggest themselves as "hot and cold," "in and out," "high and low," etc. Another possible set of names, which we shall comment on further at a later point, uses the values "true" and "false." Thus a logical variable, say A, will either have the value $A =$ true (abbreviated $A =$ T) or the value $A =$ false (abbreviated $A =$ F). If, indeed the fact is that $A =$ true ($A =$ T) we can equally well write that $A = \overline{\text{false}}$ ($A = \overline{\text{F}}$).

Now let us return to our traffic-light example. (Since we are dealing with logical variables, we shall follow the more usual custom of using A and Z for the independent and dependent variables, respectively, rather than x and y.) The functional relationship between the color of the light and the motorist's proper response is given in Fig. 1.3-1a. Suppose that in the matter of the variable A we *arbitrarily* assign the value $A =$ T to the statement "the light is red." Then automatically $A =$ F represents the statement "the light is green." Similarly let us *arbitrarily* associate the value $Z =$ T with the statement "the motorist continues." Then the functional relationship between the color of the light and the behavior of the motorist is equally well given by Fig. 1.3-1b. If the assignment of A and Z with color and motorist behavior were differently made, the pattern of entries T and F in Fig. 1.3-1b would appear different, but of course the functional relationship between color and behavior would not be altered.

A table like that in Fig. 1.3-1b with entries T and F is called a *truth table*.

A	$Z = f(A)$
Green	Continue
Red	Stop

(a)

A	$Z = f(A)$
F	T
T	F

(b)

Figure 1.3-1 A functional relationship in (a) becomes a truth table in (b).

A	$Z = f(A)$
F	F
T	T

(*a*)

A	Z
F	T
T	F

(*b*)

A	Z
F	F
T	F

(*c*)

A	Z
F	T
T	T

(*d*)

Figure 1.4-1 The four functions of a single variable.

1.4 FUNCTIONS OF A SINGLE LOGICAL VARIABLE

All the possible functions $Z = f(A)$ of a single logical variable are given in the four truth tables of Fig. 1.4-1. To assure ourselves that we have missed none of the possible functions we proceed in the following way. In the A column we simply list its two possible values F and T. We now have *two* places in the Z column where we must make entries. In each of the places there are *two* possible entries. Hence the number of distinct possible column Z is $2 \times 2 = 4$. These four are given, and we can be confident that there are no more. In Fig. 1.4-1*a*, since in each row the entry under Z is the same as under A, we write $Z = A$. In Fig. 1.4-1*b* $Z = \bar{A}$. In Fig. 1.1-4*c* $Z =$ F, and in Fig. 1.4-1*d* $Z =$ T. The reader may well decide to take the attitude that Fig. 1.4-1*c* and *d* really do not express functions at all. For in one case Z is false and in the other case Z is true quite independently of the logic value of A.

1.5 FUNCTIONS OF TWO LOGICAL VARIABLES

We consider now the functions $Z = f(A, B)$ of two logical variables A and B. To form such functions we would start out with a truth table as in Fig. 1.5-1. Here we have provided a row for each possible combination of logic values for A and B. There being two variables and two values for each, four combinations are possible. Now, to generate a function we need only make entries in the column for Z. There are four entries to be made, and for each entry there are two possibilities. Hence the total number of distinct columns possible under Z is $2 \times 2 \times 2 \times 2 = 16$, and correspondingly there are 16 possible functions of two variables. As we shall see, and as was the case with the functions of a simple variable, we may want to take the attitude that some of these "functions" are

A	B	$Z = f(A, B)$
F	F	
F	T	
T	F	
T	T	

Figure 1.5-1 An incompleted truth table for two variables.

really not functions at all. We shall eventually consider all the possible functions. For the present we consider some of the functions which are of special interest.

The AND Function

As we have already noted, a logical function is defined by a truth table. The function $Z = f(A, B)$, which is defined by the truth table in Fig. 1.5-2, is called the AND *function*. We express the dependence of Z on A and B by writing

$$Z = A \text{ AND } B \tag{1.5-1}$$

The motivation for this terminology lies in the consideration, which can be verified from the truth table, that $Z = \text{T}$ only when A *and* B are both true. An alternate symbolism for the AND function is

$$Z = A \cdot B \tag{1.5-2}$$

or even more simply

$$Z = AB \tag{1.5-3}$$

Equations (1.5-2) and (1.5-3) suggest that Z is the result of a "multiplication" in which A and B are factors. Of course A and B are not numbers, and multiplication in the usual arithmetic sense is not intended. Nonetheless, as we shall see, the suggestion of multiplication conveyed by the symbolism is deliberate, and the function A AND B is often referred to as the *logical product* of A and B.

A first property of the AND function is that it is *commutative*; i.e., if the order of A and B is interchanged (A and B are commuted), the function Z is unaltered, so that

$$Z = AB = BA \tag{1.5-4}$$

That such is the case is immediately apparent from the truth table of Fig. 1.5-1. If we had arranged the table so that the B column was the leftmost rather than the A column, the entries in the Z column would not change.

A second property of the AND function is that it is *associative*. Suppose that we have three variables A, B, and C and that we first form the logical product AB. Since this product is itself a logical variable, we can form its logical product with C, giving $(AB)C$. On the other hand, suppose we form first BC

A	B	$Z = A$ AND B
F	F	F
F	T	F
T	F	F
T	T	T

Figure 1.5-2 Truth table which defines the AND function.