Counting and Counters

R. M. M. Oberman

Professor in Information Engineering
Technical University
Delft, The Netherlands

M

First published 1981 by
THE MACMILLAN PRESS LTD
London and Basingstoke
Associated companies in Delhi Dublin
Hong Kong Johannesburg Lagos Melbourne
New York Singapore and Tokyo

Typeset in 10/12 Press Roman by
STYLESET LIMITED
Salisbury · Wiltshire
and printed in Hong Kong

ISBN 0 333 30512 4

Counting and Counters

Counting starts with zero and not with one

Preface

During the forty years in which the design of switching circuits has been increasingly developed as a science, the design of counters has, in my experience, always been an excellent proving ground for anyone who has mastered Boolean algebra for the design of gate circuits and has acquired some knowledge of the design of sequential circuits that perform shifting and counting operations.

Counters are sequential circuits with a well-defined basic operation. They can be designed through an operating algorism or a system description; equally, they can be designed using operating tables and Karnaugh maps, from which the final switching equations are derived.

Both methods are demonstrated in the text. However, the design method using algorisms or system concepts has been given preference because in the author's opinion this method provides the designer with the best insight into what he is really doing.

This book is intended to be used at the graduate level of study in digital electronics. Counting has always been and will always be an important operation in any automated technical system or organisation. It is therefore an important tool for the education of students in the science of digital techniques. A knowledge of counters is not only important in itself, it is even more important from the point of view of the insight it provides into the solution of switching problems.

A number of the counter designs discussed in the text have been used as student projects which could be carried out instead of a formal examination. The results of this type of student assessment have been extremely good.

The foregoing remarks about the scope of this book might give the impression that it is intended for teaching purposes only, but this is not so. The book contains more material than should be taught in a single course. Many different types of counting problem, a number of which are published for the first time, have been treated; thus the text will also be useful for many people working on the design of switching circuits. They will need this book as a reference and an explanation of the operation of commercially available counter circuits in integrated form. There are many different types of counter since counting is a part of almost every type of switching circuit.

In this book the concept of counting is treated very broadly. The author considers any circuit running through a cycle of states to be a counter. Its output may have a clear and well-defined relationship with the number of input pulses, but this relationship can also have a pseudo-random character, equally well-defined, but very difficult to state in the form of mathematical equations.

The author wishes to thank his colleagues A. J. van de Goor and A. Snijders for their careful reading of the manuscript and his laboratory staff for testing many of the counter circuits included in the text.

R. M. M. OBERMAN

Contents

1 Counting with Numbers

1.0 NUMBER SYSTEMS

The concept of counting has a very wide field of application in mathematics. In this text, however, the concept of counting will be restricted to the rather narrow field of counting with numbers in a number system.[1]

This type of counting is a step-by-step procedure by means of which all numbers in a number system can be obtained sequentially in the order of ever increasing value. There are two fundamental requirements for the numbers in these systems

(i) Spaces are not allowed between two successive numbers.
(ii) Different representations of the same number are not allowed.

In ordinary number systems, numbers are represented by a group of digits, each having a weight according to the radix of that system. The most well known number systems are the decimal number system for manual use, and the binary system for machine use, having bases of 10 and 2 respectively.

Each number in these systems can be represented by a polynomial of the weights of the various digits

$$A = a_m b^m + a_{m-1} b^{m-1} + \ldots + a_2 b^2 + a_1 b^1 + a_0 b^0 \tag{1.1}$$

In equation 1.1 A is the number to be represented in a number system with base b. The digits have the following range

$$a_i \in \{0, 1, \ldots, (b-1)\} \tag{1.2}$$

This means that the ten digits of the decimal number system range from 0 to 9 inclusive. The most significant digit is always 1 smaller than the base of the number system. This leads to only two digits 0 and 1 in the binary number system. These features are necessary to ensure uniqueness of the numbers in the number system.

Note that the polynomial number representation of equation 1.1 is written in such a way that it conforms to the usual number representation with the most significant digit placed left. In mathematical texts the polynomials are usually written in the reverse order.

1.1 POLYNOMIAL NUMBER REPRESENTATION

Equation 1.1 can be written in short-hand notation as follows

$$A = \sum_{n=0}^{m} a_n \prod_{k=1}^{n} b_k \tag{1.3}$$

with $a_n \in \{0, 1, \ldots, (b_n - 1)\}$ and $b_k = b$.
This definition (and also, of course, equation 1.1) leads to the following polynomial representation of binary numbers

$$A = a_m \cdot 2^m + a_{m-1} \cdot 2^{m-1} + \ldots + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0 \tag{1.4}$$

The text in the following chapters will be concerned mainly with counter circuits consisting of bistable elements so that the counter contents can be described in terms of binary numbers.

Example

$$756_{10} = 1011110100 = 1 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

The uniqueness of the binary number system follows from its definition and that follows easily from the following equation

$$1 \cdot 2^{n-1} + 1 \cdot 2^{n-1} = 2 \cdot 2^{n-1} = 1 \cdot 2^n + 0 \cdot 2^{n-1} \tag{1.5}$$

The addition of 1 to a certain 1 digit of equal weight yields a 0 digit for that weight and a carry digit 1 to the next more significant digit place.

The addition of 1 to a certain number and its consequences are extremely important since this is the fundamental counting operation by means of which the successor of number N, being $N + 1$, is determined. Counters of this type operate according the following basic algorism

$$N_{i \pm 1} = N_i \pm 1 \tag{1.6}$$

In general, the counting increment can be positive or negative, giving respectively up- and down-counting. Moreover the increment or decrement need not always be 1. It can be a constant integer, or even a variable integer. The digits in the numbers are then still binary variables, but their weight may no longer be a power of two.

Factorial representation

A variation on the polynomial representation as defined by equation 1.3 is the factorial number representation with

$$a_n \in \{0, 1, \ldots, (b_k - 1)\} \text{ and } b_k = k \tag{1.7}$$

The resulting number representation is as follows

$$A = a_m \cdot b_m! + a_{m-1} \cdot b_{m-1}! + \ldots + a_2 \cdot 2! + a_1 \cdot 1! \tag{1.8}$$

Digits a_i in the factorial number representation with $b_k = k$ have no constant range as have digits a_i in the ordinary polynomial representation with $b_k = b$.

Example

n°	a_3	a_2	a_1		
0	0	0	0	=	$0 \cdot 3! + 0 \cdot 2! + 0 \cdot 1!$
1	0	0	1	=	$0 \cdot 3! + 0 \cdot 2! + 1 \cdot 1!$
2	0	1	0	=	$0 \cdot 3! + 1 \cdot 2! + 0 \cdot 1!$
3	0	1	1	=	$0 \cdot 3! + 1 \cdot 2! + 1 \cdot 1!$
4	0	2	0	=	$0 \cdot 3! + 2 \cdot 2! + 0 \cdot 1!$
5	0	2	1	=	$0 \cdot 3! + 2 \cdot 2! + 1 \cdot 1!$
6	1	0	0	=	$1 \cdot 3! + 0 \cdot 2! + 0 \cdot 1!$
7	1	0	1	=	$1 \cdot 3! + 0 \cdot 2! + 1 \cdot 1!$
8	1	1	0	=	$1 \cdot 3! + 1 \cdot 2! + 0 \cdot 1!$
9	1	1	1	=	$1 \cdot 3! + 1 \cdot 2! + 1 \cdot 1!$

It is a fact in factorial number representations, just as in the polynomial number representation, that if all the digits of a number have reached the maximum of their range, the sum of these digits plus 1 must yield the weight of the next more significant digit. The proof of this statement is as follows.

Consider a number with p digits

$$\{p \cdot p! + (p-1) \cdot (p-1)! + \ldots 3 \cdot 3! + 2 \cdot 2! + 1 \cdot 1!\} + 1 = (p+1)! \tag{1.9}$$

$$\{p \cdot p! + p! - (p-1)! + (p-1)! - (p-2)! + \ldots + 4! - 3! + 3! - 2! + 2! - 1!\} + 1 = (p+1) \cdot p! = (p+1)! \qquad \text{Q.E.D.} \tag{1.10}$$

Code conversion of numbers in the polynomial representation into numbers with factorial representation and vice versa is not difficult. The only difference is the fact that the variable radix of the various digits has to be taken into account.

Combinatorial representation

Numbers in this representation are determined by the following equation

$$A = \binom{a_p}{p} + \ldots + \binom{a_3}{3} + \binom{a_2}{2} + \binom{a_1}{1} \tag{1.11}$$

with the condition that

$$a_p > \ldots > a_3 > a_2 > a_1 \geqslant 0 \tag{1.12}$$

The usual short-hand notation can also be applied to numbers in the combinatorial representation.

Example

$$100 = \binom{9}{3} + \binom{6}{2} + \binom{1}{1} = 84 + 15 + 1$$

This type of combinatorial representation of numbers is not very well suited to being used in a counting system. However a m-out-of-n code (words with n digits having m 1 digits) with a lexicographic order of the code words (that is the order of ever increasing binary weight) is an example of a variation on the combinatorial number representation.

The number of a word in such a code is determined by the following equation [2]

$$N = \binom{n_m}{m} + \ldots + \binom{n_2}{2} + \binom{n_1}{1} = \sum_{1}^{m} \binom{n_p}{n_b} \tag{1.13}$$

In this equation n_p represents the bit place number of a 1 bit and the number below n_p represents the number n_b of that 1 bit. The bit place numbers range from 0 to n, and the bit numbers from 1 to m inclusive.

Example

place number n_p	6	5	4	3	2	1	0		
given code word	1	0	0	1	1	0	1		
bit number n_b	4	-	-	3	2	-	1		
number weight N	$\binom{6}{4}$ +			$\binom{3}{3}$ +	$\binom{2}{2}$ +		$\binom{0}{1}$	=	
	15 +			1 +	1 +		0	=	17

Note that the number weight of a 1 digit in its initial position is 0.

The transition from one code word to the next (in lexicographic order) is determined by one rule

> The 1 bit in the least significant 01 bit combination is shifted into the place of the 0 bit in that digit pair and when this 01 digit combination has adjacent 1 bits at its right side, then these 1 bits are shifted back to their initial positions with 0 weight.

Example

word n° 18	1 0 0 1 1 1 0
word n° 19	1 0 1 0 0 1 1

Application of this rule leads to a unique code with number weights according to equation 1.13. This can be proved as follows. Let the number of adjacent 1 bits in the group of 1 bits with the least significant 01 combination be m standing on bit places n_p to $n_p - m + 1$ inclusive. The given rule can then be translated into the following general equation

$$N_{i+1} = \binom{n_{p+1}}{m} = \binom{n_p}{m} + \binom{n_p - 1}{m - 1} + \ldots + \binom{n_p - m + 1}{1} + 1 = N_i + 1 \tag{1.14}$$

Equations of this type can be proved in a step-by-step procedure using the following formula

$$\binom{n_{p+1}}{m} = \binom{n_p}{m} + \binom{n_p}{m-1} \tag{1.15}$$

This formula shows that all combinations m-out-of-n_{p+1} are obtained by the sum of all combinations m-out-of-n_p plus all combinations $(m - 1)$-out-of-n_p. In the first set the digit at bit place n_{p+1} is considered to be 0 so that all the m 1 bits are found in the other n_p bit places. In the second set the digit at bit place n_{p+1} is considered to be 1 so that there are only $m - 1$ 1 bits on the other n_p bit places.

Expression 1.12 represents the first step of the proof of equation 1.11. In the second step, expression 1.12 is applied to the second term on the right-hand side of equation 1.11. This leads to the following equation

$$\binom{n_{p+1}}{m} = \binom{n_p}{m} + \binom{n_p - 1}{m-1} + \binom{n_p - 1}{m-2} \tag{1.16}$$

This procedure ends when the following situation is reached

$$\binom{n_{p+1}}{m} = \binom{n_p}{m} + \binom{n_p - 1}{m-1} + \ldots + \binom{n_p - m + 1}{1} + \binom{n_p - m + 1}{0} \tag{1.17}$$

Equation 1.17 is identical to equation 1.14 since $\binom{n_p - m + 1}{0} = 1$, Q.E.D.

Equation 1.14 reduces to the following identity for $m = 1$

$$\binom{n_{p+1}}{m} = \binom{n_p}{1} + \binom{n_p - 1}{0}$$

or

$$n_{p+1} = n_p + 1 \tag{1.18}$$

This proof of the uniqueness of the words in a constant-ratio code shows that it is well suited to counting purposes. However implementing this counting code in hardware is another problem. The complexity of that implementation depends on the switching operations required to obtain a lexicographic sequence of the code words, and this problem will be treated in detail in chapter 4.

Modular representation

This representation is also called the system of residual classes.[3] In this number system addition, subtraction and multiplication are performed without the usual carry problem. Since machine arithmetic is not the subject of this text, this special feature will not be discussed except for the add 1 (or subtract 1) procedure.

In modular representation the numbers are represented by means of a number of positive residues taken modulo p_i from the decimal number to be represented. Modulo p_i must be relatively prime. The system of prime numbers satisfies this requirement.

Example

modulus n° :	4	3	2	1	0
modulus p_i :	11	7	5	3	2
127 ≡	(6	1	2	1	1)

The maximum number that can be represented in this type of number system is $\Pi_{i=0}^{n} p_i$. It follows from the definition of this modular representation that the algorism $N_{i+1} = N_i + 1$ has to be performed as an add-1 operation mod p_i on all digits in a number. Uniqueness is then ensured.

Example

p_i :	11	7	5	3	2
128 = 127 + 1 ≡	(7	2	3	2	0)

Counting with large numbers in modular representation is subdivided into a number of counting processes with smaller numbers so that the counting problem reduces to counting with the residues.

A big problem in the attractive-looking modular representation of numbers is how to convert these numbers back to a binary or decimal representation. This is a rather complicated procedure resulting in the fact that the possible advantages in the actual counting process are not set off by the disadvantages of this code conversion.

1.2 SOME EXTENSIONS OF THE POLYNOMIAL REPRESENTATION

The polynomial representations of numbers using base 10 or base 2 are the most commonly used systems. A couple of other number systems are closely related to the binary number system and belong to those number systems which have a polynomial representation of their numbers.

The number systems discussed in section 1.1 all have digits with positive weights. This is not a necessity in number systems. The following two number systems have positively and negatively weighted digits. Strictly speaking these systems are no longer binary systems but ternary systems. However, the 1 digits in these numbers are alternately positive and negative, so that it is known in advance from any code word which digits are positive and which are negative without further indication. The most significant 1 bit is always positive in a positive number.

The signed-digit number system

The numbers in this system are directly obtained from the ordinary binary system by replacing all sequences of adjacent 1 digits by a pair of 1 digits in which the most significant is positively weighted, and the least significant is negatively weighted, this gives the following two equations

$$a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_{n-m} 2^{n-m} = a_{n+1} 2^{n+1} - a_{n-m} 2^{n-m} \quad (1.19)$$

$$a_n 2^n = a_{n+1} 2^{n+1} - a_n 2^n \quad (1.20)$$

The second equation is identical to the first equation with $m = 0$. The second equation states that even a single (positive) 1 bit at digit place n can be replaced by a positively weighted 1 bit at digit place $n + 1$ and a negatively weighted 1 bit at place n. This is a simple law in binary arithmetic which needs no further proof.

Application of equations 1.19 and 1.20 to a binary word or number gives a signed-digit number with an even number of 1 bits. All these words start with a positive most significant 1 bit, further bits being alternately negative and positive, and they end with a negative least significant 1 bit. In this transformation the number of 1 bits is extended by one.[4]

Table 1-1 shows a 4-bit natural binary code. The corresponding 5-bit signed-digit code is shown in column 3. The number weight in column 1 is valid for all the codes shown in table 1-1.

Note that the weight of the digits in the signed-digit code remains a power of 2.

Table 1–1

number weight	nat. binary				signed digit						reflected binary				
	8	4	2	1	16	±8	±4	±2	-1	binary weight	15	±7	±3	±1	binary weight
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	-1	3	0	0	0	1	1
2	0	0	1	0	0	0	1	-1	0	6	0	0	1	-1	3
3	0	0	1	1	0	0	1	0	-1	5	0	0	1	0	2
4	0	1	0	0	0	1	-1	0	0	12	0	1	-1	0	6
5	0	1	0	1	0	1	-1	+1	-1	15	0	1	-1	+1	7
6	0	1	1	0	0	1	0	-1	0	10	0	1	0	-1	5
7	0	1	1	1	0	1	0	0	-1	9	0	1	0	0	4
8	1	0	0	0	1	-1	0	0	0	24	1	-1	0	0	12
9	1	0	0	1	1	-1	0	+1	-1	27	1	-1	0	+1	13
10	1	0	1	0	1	-1	+1	-1	0	30	1	-1	+1	-1	15
11	1	0	1	1	1	-1	+1	0	-1	29	1	-1	+1	0	14
12	1	1	0	0	1	0	-1	0	0	20	1	0	-1	0	10
13	1	1	0	1	1	0	-1	+1	-1	23	1	0	-1	+1	11
14	1	1	1	0	1	0	0	-1	0	28	1	0	0	-1	9
15	1	1	1	1	1	0	0	0	-1	17	1	0	0	0	8

The polynomial representation of the signed digit code is as follows

$$N = \ldots \pm n_4 2^4 \pm n_3 2^3 \pm n_2 2^2 \pm n_1 2^1 - n_0 2^0 \quad (1.21)$$

with a positive most significant non-zero n coefficient.