Chapter 7: THE BINARY SYSTEM OF LEIBNIZ (1697)

THE DECIMAL SYSTEM

Counting in the decimal system from zero upwards by serially adding 1

Recall that if in adding 1 to the last digit the result exceeds the largest admissible digit which is 9, then we write 0 for the sum and carry the digit 1 to the next column. In writing binary numbers we shall use bold face type to distinguish them from decimals: 10 $\neq$ **10** = 2.

The Overflow Formula for decimals states that 999…9 + 1 = 1000...0 (the number of 9's on the LHS = the number of 0's on the RHS.) The Overflow Formula for the binary numbers state that **111…1 + 1 = 1000…0** (the number of **1**'s on the LHS = the number of **0**'s on the RHS,)

THE BINARY SYSTEM

Table of the first one hundred consecutive binary numbers

0	**1**	**10**	**11**	**100**	**101**	**110**	**111**	**1000**	**1001**
1010	**1011**	**1100**	**1101**	**1110**	**1111**	**10000**	**10001**	**10010**	**10011**
10100	**10101**	**10110**	**10111**	**11000**	**11001**	**11010**	**11011**	**11100**	**11101**
11110	**11111**	**100000**	**100001**	**100010**	**100011**	**100100**	**100101**	**100110**	**100111**
101000	**101001**	**101010**	**101011**	**101100**	**101101**	**101110**	**101111**	**110000**	**110001**
110010	**110011**	**110100**	**110101**	**110110**	**110111**	**111000**	**111001**	**111010**	**111011**
111100	**111101**	**111110**	**111111**	**1000000**	**1000001**	**1000010**	**1000011**	**1000100**	**1000101**
1000110	**1000111**	**1001000**	**1001001**	**1001010**	**1001011**	**1001100**	**1001101**	**1001110**	**1001111**
1010000	**1010001**	**1010010**	**1010011**	**1010100**	**1010101**	**1010110**	**1010111**	**1011000**	**1011001**
1011010	**1011011**	**1011100**	**1011101**	**1011110**	**1011111**	**1100000**	**1100001**	**1100010**	**1100011**

Counting in the binary system from zero forwards by serially adding **1**

$$\begin{array}{r} \mathbf{0} \\ \underline{\mathbf{1}} \\ \mathbf{1} \\ \underline{\mathbf{1}} \\ \mathbf{10} \\ \underline{\mathbf{1}} \\ \mathbf{11} \\ \underline{\mathbf{1}} \\ \mathbf{100} \\ \underline{\mathbf{1}} \\ \mathbf{101} \\ \underline{\mathbf{1}} \\ \mathbf{110} \\ \underline{\mathbf{1}} \\ \mathbf{111} \end{array} \qquad \begin{array}{r} \mathbf{111} \\ \underline{\mathbf{1}} \\ \mathbf{1000} \\ \underline{\mathbf{1}} \\ \mathbf{1001} \\ \underline{\mathbf{1}} \\ \mathbf{1010} \\ \underline{\mathbf{1}} \\ \mathbf{1011} \\ \underline{\mathbf{1}} \\ \mathbf{1100} \\ \underline{\mathbf{1}} \\ \mathbf{1101} \\ \underline{\mathbf{1}} \\ \mathbf{1110} \end{array} \qquad \begin{array}{r} \mathbf{1110} \\ \underline{\mathbf{1}} \\ \mathbf{1111} \\ \underline{\mathbf{1}} \\ \mathbf{10000} \\ \underline{\mathbf{1}} \\ \mathbf{10001} \\ \underline{\mathbf{1}} \\ \mathbf{10010} \\ \underline{\mathbf{1}} \\ \mathbf{10011} \\ \underline{\mathbf{1}} \\ \mathbf{10100} \\ \underline{\mathbf{1}} \\ \textit{continue} \end{array}$$

Note. If in adding **1** to the last digit the result exceeds the largest admissible digit which is **1**, then we write **0** for the sum and carry the digit **1** to the next column.

We shall use the abbreviations **1000…0** = $\mathbf{10}_n$ (n copies of **0** following **1**) and **111…1** = $\mathbf{1}_n$ (n copies of **1**). We thus have: $\mathbf{10}_0 = 1$, $\mathbf{10}_1 = \mathbf{10} = 2$, $\mathbf{10}_2 = \mathbf{100} = 4$, $\mathbf{10}_3 = \mathbf{1000} = 8$, $\mathbf{10}_4 = \mathbf{10000} = 16$, $\mathbf{10}_5 = \mathbf{100000} = 32$, $\mathbf{10}_6 = \mathbf{1000000} = 64$, $\mathbf{10}_7 = \mathbf{10000000} = 128$, $\mathbf{10}_8 = 256$, $\mathbf{10}_9 = \mathbf{1000000000} = 512$, $\mathbf{10}_{10} = 1024$,…
We also have: $\mathbf{1}_1 = \mathbf{1} = 1$, $\mathbf{1}_2 = \mathbf{11} = 3$, $\mathbf{1}_3 = \mathbf{111} = 7$, $\mathbf{1}_4 = \mathbf{1111} = 15$, $\mathbf{1}_5 = \mathbf{11111} = 31$, $\mathbf{1}_6 = \mathbf{111111} = 63$, $\mathbf{1}_7 = \mathbf{1111111} = 127$, $\mathbf{1}_8 = \mathbf{11111111} = 255$, $\mathbf{1}_9 = \mathbf{111111111} = 511$, $\mathbf{10}_{10} = \mathbf{1111111111} = 1023$,… These abbreviations will also be used in combination: $\mathbf{1}_k\mathbf{0}_n$ = **111…1000…0** (k copies of **1** followed by n copies of **0**), e.g., $\mathbf{1}_2\mathbf{0}_2 = 12$, $\mathbf{1}_2\mathbf{0}_3 = 24$, $\mathbf{1}_3\mathbf{0}_2 = 28$.

The Overflow Formula for the binary system states that $\mathbf{1}_n + \mathbf{1} = \mathbf{10}_n$ (compare with the Overflow Formula for decimals).

Exercises:

1. Count *backwards* from **10000** to **1**.

2. Find the values of $\mathbf{10}_{11}$, $\mathbf{10}_{12}$ and $\mathbf{1}_{11}$, $\mathbf{1}_{12}$.

3. Find the values of $\mathbf{1}_2\mathbf{0}_5$, $\mathbf{1}_2\mathbf{0}_3\mathbf{1}_2$, $\mathbf{10}_2\mathbf{1}_4$.

4. Prepare a table for the values of $\mathbf{10}_n$ for $n = 0$ through 20.

5. Find the values of $\mathbf{1}_n$ for $n = 11, 12, 13, 14, 15$.

6. Show that $\mathbf{10}_n = 2^n$.

The milestones in the binary system are: $\mathbf{10}_0 = 1$, $\mathbf{10}_1 = 2$, $\mathbf{10}_2 = 4$, $\mathbf{10}_3 = 8$, $\mathbf{10}_4 = 16$, $\mathbf{10}_5 = 32$, $\mathbf{10}_6 = 64$, $\mathbf{10}_7 = 128$, $\mathbf{10}_8 = 256$, $\mathbf{10}_9 = 512$,…, in general: $\mathbf{10}_n = 2^n$. These milestones have interesting applications. They can be used for counting:

(1) the number of binary numbers of at most n digits
(2) the number of ways we can pick a selection of balls from a set of n balls
(3) the number of ways to split a set of n balls into two sets.

Exercises:

7. How many binary numbers have at most n digits? How many have exactly n digits? (*Hint:* start with the fact that that $\mathbf{10}_n$ counts the number of binary numbers from **1** through $\mathbf{10}_n$ inclusive.)

8. In how many different ways can we pick a selection from a set of n different balls? (*Hint:* Start with the binary number $\mathbf{1}_n$ = **111…1** and identify the digits **1** with the balls. Replace **1** by **0** if you *do not* pick that particular ball.)

9. In how many ways can we split a set of n balls into two sets. (*Hint:* consider the fact that in picking a selection from the n balls, you willy-nilly pick another as well.)

10. A group of 5 children want to play a ball-game. In how many ways can they divide themselves into two teams? (Each team must have at least 1 player.)

CONVERSION OF DECIMALS INTO BINARY NUMBERS

We learn two methods to do the conversion: the *long* and the *short* method. In checking the conversion the *sum formula* for the powers of 2 is helpful:

$$1 + 2 + 4 + 8 + \dots + 2^n = 2^{n+1} - 1$$

or, more generally

$$2^n + 2^{n+1} + \dots + 2^{n+m} = 2^n(2^{m+1} - 1)$$

First we take a look at long method. We start by determining the number of digits the decimal number N will have in the binary system. We do this by determining the two adjacent milestones (powers of 2) enclosing N. Of course, if N is a power of 2, then the conversion is obvious, e.g., $N = 2 =$ **10**; $N = 8 = 2^3 =$ **1000**. Otherwise we take the difference $N - 2^n > 0$ with the largest n possible and repeat the process.

Example 1. $N = 255$. $2^7 = 128 < 255 < 256 = 2^8$; N has 8 digits, 1st is **1**.

$255 - 128 = 127$;	$2^6 = 64 < 127 < 128 = 2^7$;	2nd is **1**.
$127 - 64 = 63$;	$2^5 = 32 < 63 < 64 = 2^6$;	3rd is **1**.
$63 - 32 = 31$;	$2^4 = 16 < 31 < 32 = 2^5$;	4th is **1**.
$31 - 16 = 15$;	$2^3 = 8 < 15 < 16 = 2^4$;	5th is **1**.
$15 - 8 = 7$;	$2^2 = 4 < 7 < 8 = 2^3$;	6th is **1**.
$7 - 4 = 3$;	$2^1 = 2 < 3 < 4 = 2^2$;	7th is **1**.
$3 - 2 = 1$;		the 8th and last digit is **1**.

$N =$ **11111111**. Check: $128+64+32+16+8+4+2+1 = 2^8 - 1 = 256 - 1 = 255$.

Example 2. $N = 254$. The first six steps are very similar to those of Example 1, after which we get:

$6 - 4 = 2$;	$2^1 = 2 = 6 - 4$;	the 7th digit is **1**.
$2 - 2 = 0$;		the 8th and last digit is **0**.

$N =$ **11111110.** Check: $128+64+32+16+8+4+2 = 2(2^7 - 1) = 2(127) = 254$.

Example 3. $N = 135$. $2^7 = 128 < 135 < 256 = 2^8$; N has 8 digits, the 1st is **1**.

$135 - 128 = 7$;	$2^6 = 64 > 7$;	the 2nd digit is **0**.
	$2^5 = 32 > 7$;	3rd is **0**.
	$2^4 = 16 > 7$;	4th is **0**.
	$2^3 = 8 > 7$;	5th is **0**.
	$2^2 = 4 < 7 < 8 = 2^3$;	6th is **1**.
$7 - 4 = 3$;	$2^1 = 2 < 3 < 4 = 2^2$;	7th is **1**.
$3 - 2 = 1$;		the 8th and last digit is **1**.

$N =$ **10000111.** Check: $128 + 4 + 2 + 1 = 135$.

Exercise 4. $N = 51$. $2^5 = 32 < 51 < 64 = 2^6$; N has 6 digits, 1st is **1**.
$51 – 32 = 19$; $2^4 = 16 < 19 < 32 = 2^5$; 2nd is **1**.
$19 – 16 = 3$; $2^3 = 8 > 3$; 3rd is **0**.
$2^2 = 4 > 3$; 4th is **0**.
$2^1 = 2 < 3 < 4 = 2^2$; 5th is **1**.
$3 – 2 = 1$; the 6th and last digit is **1**.
N = **110011**. Check: 32 + 16 + 2 + 1 = 51.

Example 5. $N = 2011$. $2^{10} = 1024 < N < 2048 = 2^{11}$; N has 11 digits, 1st is **1**.
$2011 – 1024 = 987$; $2^9 = 512 < 987 < 1024 = 2^{10}$; 2nd is **1.**
$987 – 512 = 475$; $2^8 = 256 < 475 < 512 = 2^9$; 3rd is **1**.
$475 – 256 = 219$; $2^7 = 128 < 219 < 256 = 2^8$; 4th is **1**.
$219 – 128 = 91$; $2^6 = 64 < 91 < 128 = 2^7$; 5th is **1**.
$91 – 64 = 27$; $2^5 = 32 > 27$ 6th is **0**.
$2^4 = 16 < 27 < 32 = 2^5$; 7th is **1**.
$27 – 16 = 11$; $2^3 = 8 < 11 < 16 = 2^4$; 8th is **1**.
$11 – 8 = 3$; $2^2 = 4 > 3$ 9th is **0**.
$2^1 = 2 < 3 < 4 = 2^2$; 10th is **1.**
$3 – 2 = 1$; 11th is **1**.
N = **11111011011.** *Check*: 1024+512+256+128+64+16+8+2+1 = 2011.

While both the long and short methods are always applicable, the short method is especially useful if N is close to a power of 2. If N follows 2^n closely, then we count *forward* from 2^n to N; if N precedes 2^n but by not much, then we count *backward* from 2^n to N. To illustrate the short method, let us recalculate Example 3 and 2.

Example 6. $N = 135$ follows $2^7 = 128 =$ **10000000** by 7 steps. Accordingly, we count forward 7 times: **10000001, 10000010, 10000011, 10000100, 10000101, 10000110, 1000111** = ***N*.** Check: compare with Example 3 above.

Example 7. $N = 254$ precedes $2^8 = 256 =$ **100000000** by 2 steps. Accordingly, we count backward in two steps: **11111111, 1111110 =** N. Check: compare with Example 2 above.

Exercises:

11. Convert $N = 253$ into a binary number by using the short method, and check in two different ways.

12. Convert $N = 515$ into a binary number by using the long method and check your calculation in two different ways.

13. Convert N into a binary number by using the long method and check:
(*i*) $N = 21$
(*ii*) $N = 73$
(*iii*) $N = 273$