

4-26-2015

Fun with Fibonacci

John E. Baker

Natural Maths, john@naturalmaths.com.au

Steve Sugden

Queensland University of Technology, ssugden@bond.edu.au

Follow this and additional works at: <http://epublications.bond.edu.au/ejsie>



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Recommended Citation

Baker, John E. and Sugden, Steve (2015) Fun with Fibonacci, *Spreadsheets in Education (eJSiE)*: Vol. 7: Iss. 2, Article 1.

Available at: <http://epublications.bond.edu.au/ejsie/vol7/iss2/1>

This In the Classroom Article is brought to you by the Bond Business School at [epublications@bond](mailto:epublications@bond.edu.au). It has been accepted for inclusion in *Spreadsheets in Education (eJSiE)* by an authorized administrator of [epublications@bond](mailto:epublications@bond.edu.au). For more information, please contact [Bond University's Repository Coordinator](mailto:epublications@bond.edu.au).

Fun with Fibonacci

Abstract

We introduce a function which measures the number of distinct ways in which a number can be expressed as the sum of Fibonacci numbers. Using a binary table and other devices, we explore the values that can take and reveal some interesting patterns. The article shows how standard spreadsheet functionalities make it possible to reveal quite striking patterns in data, and is intended to be used in the classroom.

Keywords

Mathematics investigation, Fibonacci numbers, Zeckendorf

Distribution License



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Fun with Fibonacci

John Baker

Natural Maths

john@naturalmaths.com.au

Steve Sugden

Queensland University of Technology

stephen.sugden@qut.edu.au

Abstract

We introduce a function $Z(k)$ which measures the number of distinct ways in which a number can be expressed as the sum of Fibonacci numbers. Using a binary table and other devices, we explore the values that $Z(k)$ can take and reveal some interesting patterns. The article shows how standard spreadsheet functionalities make it possible to reveal quite striking patterns in data, and is intended to be used in the classroom.

Keywords: Mathematics investigation, Fibonacci numbers, Zeckendorf.

1. Introduction

In 1952, a new feature of Fibonacci numbers was articulated by E. Zeckendorf [4]. He showed that every number can be expressed as the sum of non-consecutive Fibonacci numbers in a unique way. His theorem is described in MathWorld as:

$$k = \sum_{i=2}^n e_i f(i) \quad \text{for all } k \in \mathbb{Z}$$

where $e_i = 0$ or 1 and $e_j e_{j+1} = 0$ and the Fibonacci numbers are $1, 1, 2, 3, 5, 8, \dots$ starting with $f(1) = 1, f(2) = 1, f(3) = 2, f(4) = 3$, and so on. The requirement that $e_j e_{j+1} = 0$ means that adjacent Fibonacci numbers may not be used. This is no real restriction since, if we wished to include both $f(j)$ and $f(j+1)$, then we can just use $f(j+2)$ instead, as this is always equal to the sum of the two predecessors.

A previous article in this journal [1] described an investigation of Goldbach's Theorem, where the centre of interest was in finding the number of ways in which an even number can be expressed as the sum of prime numbers. In this article we apply that concept to Zeckendorf's Theorem and explore the number of ways in which a number can be expressed as the sum of Fibonacci numbers. During the investigation, certain features of the spreadsheet are used to help locate these sums and the patterns that the spreadsheet reveals are, we believe, only made visible because of these features. The features that were found particularly powerful were *conditional formatting* and *data table filtering*. When combined with a simple method of constructing a binary table the results that we found were made visible in a remarkable way. The process showed how results described by Klarner [3] can be displayed visually and led to further results about the expansions of a natural

number as the sum of Fibonacci numbers that are described in our companion paper [6].

2. Finding the Zeckendorf Expansion

Finding the Zeckendorf expansion is made very straightforward when the VLOOKUP function is used. In Wikipedia [5], the method of finding this unique expansion is described as a 'greedy' algorithm. This is just what the VLOOKUP function does because if your number is not in the list that you are searching, VLOOKUP returns the largest number that is not greater than the number sought.

2.1. Example

Suppose we want the Zeckendorf expansion for the number 20. VLOOKUP scans the list of distinct Fibonacci numbers 1, 2, 3, 5, 8, 13, 21, 34 etc., and returns 13 as the largest value not greater than 20. We then subtract 13 from 20 to obtain 7 and repeat the process, generating the sequence 13, 5, 2. This is done in the spreadsheet by very simple formulas, using just subtraction and the VLOOKUP function. In this manner, a sub-sequence of the Fibonacci numbers is generated which adds exactly to the original number; in this case, $20 = 13 + 5 + 2 = f(7) + f(5) + f(3)$.

Thus the spreadsheet to find the expansion of any number up to, say, 10,000 is shown in Figure 1. The corresponding file, *Zeckendorf.xlsx*, may be downloaded with the present article. The number to be expressed as the sum of Fibonacci numbers is selected by moving the slider that covers cell D1 and the formulas used are derived from:

= IF(D3 = 0, 0, VLOOKUP(D3, Fibonacci, 1)) in cell E3

= D3 – E3 in cell D4

Since these formulas are filled-down to row 28, for smaller values of k in D3, they will generate many zeroes. Such zeroes are conveniently hidden from view by a standard "trick" use of conditional formatting (white font on white background). For any slider-selected value in D3, the model generates the terms of the (unique) Zeckendorf expansion in Column E. The slider is limited to 10,000 but you can also type larger values directly into cell D3. However, to find all possible ways in which a number k can be expressed as the sum of Fibonacci numbers, including both $f(1)$ and $f(2)$ terms at the beginning, a different approach is needed. We consider this problem in the remainder of the paper.

	A	B	C	D	E	F	G
1							
2		n	Fibonacci	k	Zeckendorf		Checksum
3		2	1	9876	6765		9876
4		3	2	3111	2584		
5		4	3	527	377		
6		5	5	150	144		
7		6	8	6	5		
8		7	13	1	1		
9		8	21				
10		9	34				
11		10	55				
12		11	89				
13		12	144				

Figure 1: Spreadsheet for finding a Zeckendorf expansion of k

3. Using a binary truth table to find expansions of k

Suppose we wish to find all the subsets of a given set S . This is only feasible for sets of very modest cardinality, since if S has m elements, then it will have 2^m subsets. We create a binary truth table that lists 0s and 1s in a form that can be used to select elements from S . To create the binary table, we use the fact that a number in position n will change only if there is a carry from the previous column, which happens when there is a change from 1 to 0. Full details for creating a binary truth table are given in Appendix B. The supplied Excel models also contain this logic.

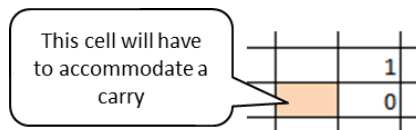


Figure 2: Creating a binary table

The first few rows of the truth table are shown in Figure 2, and the calculations for it are given in the Appendix. The addition of conditional formatting makes the pattern of 1s and 0s much clearer and, if the top row were to be replaced with powers of 2, the composition of the table as a listing of binary forms would be made even more obvious.

Our goal is to express a number k in the form:

$$k = \sum_{i=1}^n e_i f(i)$$

and to find in how many different ways this can be done. To do this, we need to replace the top row of the table with Fibonacci numbers and to form the SUMPRODUCT of the top row with the remaining rows of the table as that will furnish a complete listing of all possible subsets of Fibonacci numbers and the product will give the sum of each subset. This process is shown in Figure 3, where the formula in cell M3 is

= SUMPRODUCT(\$C\$2:\$L\$2, C3:L3)
or better still

= SUMPRODUCT(fibs, C3:L3), where the top row has been given the name *fibs*.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		n	55	34	21	13	8	5	3	2	1	1	k
3		0	0	0	0	0	0	0	0	0	0	0	0
4		1	0	0	0	0	0	0	0	0	0	1	1
5		2	0	0	0	0	0	0	0	0	1	0	1
6		3	0	0	0	0	0	0	0	0	1	1	2
7		4	0	0	0	0	0	0	0	1	0	0	2
8		5	0	0	0	0	0	0	0	1	0	1	3
9		6	0	0	0	0	0	0	0	1	1	0	3
10		7	0	0	0	0	0	0	0	1	1	1	4
11		8	0	0	0	0	0	0	1	0	0	0	3
12		9	0	0	0	0	0	0	1	0	0	1	4
13		10	0	0	0	0	0	0	1	0	1	0	4
14		11	0	0	0	0	0	0	1	0	1	1	5
15		12	0	0	0	0	0	0	1	1	0	0	5
16		13	0	0	0	0	0	0	1	1	0	1	6
17		14	0	0	0	0	0	0	1	1	1	0	6
18		15	0	0	0	0	0	0	1	1	1	1	7
19		16	0	0	0	0	0	1	0	0	0	0	5

Figure 3: Expansions of k as the sum of Fibonacci numbers

The SUMPRODUCT formula is used for each row of the table, showing, for example, that in rows 5, 6 and 8, the expansions of 3 are:

$$\text{Row 5: } f(1) + f(3) = 3$$

$$\text{Row 6: } f(2) + f(3) = 3$$

$$\text{Row 8: } f(3) = 3$$

3.1. The Binary Table suggests a formula

Even though it is not complete, the binary table suggests a formula, for when we look at the k-values for those rows that are only 1s, we find that:

$$\text{Row 1: } f(1) = 1$$

$$\text{Row 3: } f(1) + f(2) = 2$$

$$\text{Row 7: } f(1) + f(2) + f(3) = 4$$

$$\text{Row 15: } f(1) + f(2) + f(3) + f(4) = 7$$

Add 1 to each of these row totals and the sequence is 2, 3, 5, 8, which are the Fibonacci numbers. The formula that the table uncovers is:

$$\sum_{i=1}^n f(i) = f(n + 2) - 1 \tag{Equation 1}$$

This is a well-known result (Identity 22 in [2]) and it is one that we used many times in our investigation.

For the next step, we need to make a definition. Let $Z(k)$ = the number of distinct ways in which the number k can be expressed as a sum of Fibonacci numbers.

4. Creating the $Z(k)$ values

The binary table was extended to include the Fibonacci numbers 1 – 377 and, as this meant that the binary forms would go from $0 - 2^{14}$, the table extended down 16,384 rows. This enabled us to find the $Z(k)$ values for $0 \leq k < 610$ as any expansion that includes the next Fibonacci number, 610, will be an expansion of a number that is ≥ 610 .

The values of k were listed in Column U and the last column of the binary table was given the name **z_values**. This enabled the table of $Z(k)$ values to be created with the formula in cell V3:

$$= \text{COUNTIF}(\text{z_values}, \text{U3})$$

The data that this new table generated was then plotted as a scatter chart, as shown in Figure 5.

The first impression is that these data are grouped into intervals that are demarcated by a series of low values. It also appears that each group reaches a maximum and then falls back before the next group is reached. Using a feature of Excel whereby you can move the cursor to a point on a graph and the coordinates of that point are displayed, the values of $(k, Z(k))$ can be explored.

T	U	V	W
	k	Z(k)	
	0	1	
	1	2	
	2	2	
	3	3	
	4	3	
	5	3	
	6	4	
	7	3	
	8	4	
	9	5	
	10	4	
	11	5	
	12	4	
	13	4	
	14	6	
	15	5	
	16	6	
	17	6	

Figure 4: Calculation of $Z(k)$

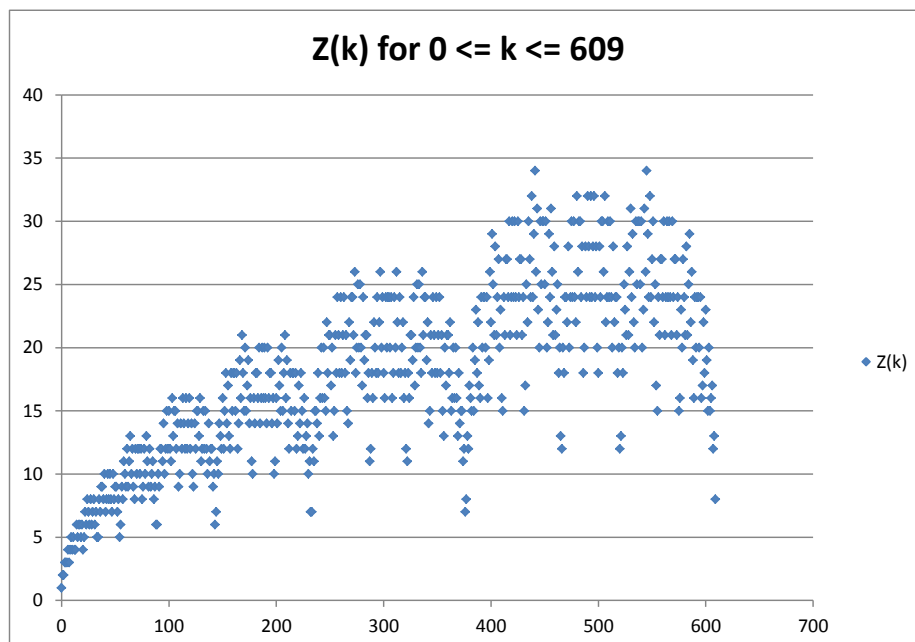


Figure 5: A plot of $Z(k)$

The low points appeared to be close to or exactly equal to a Fibonacci number, and so the $Z(k)$ values for the Fibonacci numbers were added to the chart and the visual impression was quickly confirmed.

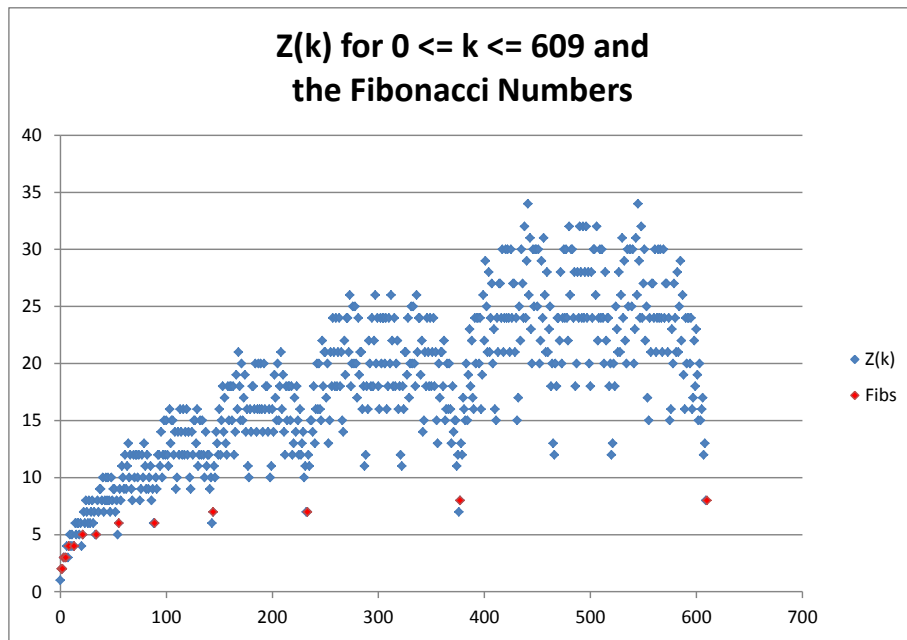


Figure 6: A plot of $Z(k)$ with $Z(f(n))$ added

The table of $Z(f(n))$ values for the Fibonacci numbers itself is quite suggestive and indicates that the following formulae for $Z(f(n))$ are:

$$Z(f(2n)) = n + 1 \quad \text{Equation 2}$$

$$Z(f(2n + 1)) = n + 1 \quad \text{Equation 3}$$

In order to see why this should be the case, we turned to the binary table and used the filter feature to select just those rows whose k -value was a Fibonacci number. For example, the binary table reduces to the following when k -values of 13, for which $n = 7$, are selected (see Figure 7).

You can see that the bottom row is the singleton $f(7) = 13$, while the preceding rows are derived from the ones below by applying the standard Fibonacci formula

$$f(n) = f(n - 1) + f(n - 2)$$

n	$f(n)$	$Z(f(n))$
1	1	2
2	1	2
3	2	2
4	3	3
5	5	3
6	8	4
7	13	4
8	21	5
9	34	5
10	55	6
11	89	6
12	144	7
13	233	7
14	377	8
15	610	8

n	55	34	21	13	8	5	3	2	1	1	k
43	0	0	0	0	1	0	1	0	1	1	13
44	0	0	0	0	1	0	1	1	0	0	13
48	0	0	0	0	1	1	0	0	0	0	13
64	0	0	0	1	0	0	0	0	0	0	13

Figure 7: The binary table for 13

n	55	34	21	13	8	5	3	2	1	1	k
85	0	0	0	1	0	1	0	1	0	1	21
86	0	0	0	1	0	1	0	1	1	0	21
88	0	0	0	1	0	1	1	0	0	0	21
96	0	0	0	1	1	0	0	0	0	0	21
128	0	0	1	0	0	0	0	0	0	0	21

Figure 8: The binary table for 21

The distinctive pattern of the binary table for $f(7) = 13$ is repeated for 21 (see Figure 8), with an important difference:

The table for 13 has only 1 row that uses $f(1)$ and $f(2)$ whereas these are split into two rows for 21.

This visual impression can be formalised into a proof that Equations 2 and 3 are indeed true for all values of n .

4.1. Theorem

Let $Z(f(n))$ be the number of ways of representing $f(n)$ as the sum of Fibonacci numbers. Then $Z(f(1)) = 2$ and $Z(f(n)) = \lfloor (n/2) \rfloor + 1$ for $n \geq 2$. Equivalently, $Z(f(2n)) = n + 1$ and $Z(f(2n + 1)) = n + 1$ for $n \geq 1$, with $Z(f(1)) = 2$.

4.2. Proof

Our proof is inspired by the binary table, in which the patterns are very suggestive. The result is clear for $n = 1$ and $n = 2$, so we consider only $n \geq 3$ and begin by noting that one representation of $f(n)$ is just $f(n)$ itself. Any other representation clearly cannot include $f(n)$, but must include $f(n - 1)$, for if not, the greatest value that could be obtained would be the sum of the first $n - 2$ Fibonacci numbers, and this falls short of $f(n)$ since it always equals $f(n) - 1$. Since $f(n - 1)$ must be present, the remaining terms must sum to $f(n - 2)$, by virtue of the Fibonacci recurrence. The number of ways this can happen is precisely $Z(f(n - 2))$. Thus, we have the beautiful recurrence (already apparent from our Excel model) $Z(f(n)) = 1 + Z(f(n - 2))$ for $n \geq 3$. Coupled with $Z(f(1)) = Z(f(2)) = 2$, the result follows swiftly. ■

4.3. A final comment on logarithms

Another interesting aspect of all this is that, viewed as a function whose domain is the set of Fibonacci numbers, equations 2 and 3 are inferring that Z appears to be a kind of “inverse Fibonacci” function. That is, $Z(f(n))$ is something like an identity function, or at least a multiple of identity (linear). Since we know that the direct formula for $f(n)$ involves exponential functions of n , then its inverse would be some kind of logarithm, or a closely related function. Indeed, a cursory glance at the red points in Figure 5 certainly evokes memories of the graph of a logarithmic function.

5. Conclusion

Yet again, it has been shown that, with relatively modest effort, the modern electronic spreadsheet becomes a fertile source for visualising patterns which represent mathematical theorems. Students can see the patterns and will find that certain spreadsheet features provide powerful tools for the investigation of a mathematical topic. Of these, the two most important would be:

- Conditional Formatting
- Charting the Data

But there are other features that can be of great value. For example, the COUNTIF function can reveal trends in a way that would be most difficult to see otherwise and the ability to filter a data set enables you to single out certain data for inspection. We also introduced the concept of a binary table as a mechanism for making sure that every possible outcome was catered for.

So often, a mathematical investigation requires that data be generated from which generalisations can be formed, or limits such as maxima and minima can be found. In this case it has been these types of activity that the spreadsheet has made possible and accessible.

The reader interested in further results along the lines presented here is referred to our companion paper [6].

6. References

1. Baker, John (2007) "Excel and the Goldbach Comet," *Spreadsheets in Education (eJSiE)*: Vol. 2: Iss. 2, Article 2.
2. Chandra, Pravin and Weisstein, Eric W. "Fibonacci Number." From MathWorld--A Wolfram Web Resource.
<http://mathworld.wolfram.com/FibonacciNumber.html>
3. Klarner, David (1966) "Representation of N as a Sum of Distinct Elements from Special Sequences" *Fibonacci Quarterly* 4:4.
4. Zeckendorf, E. (1972). "Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas". *Bull. Soc. R. Sci. Liège* (in French) 41: 179–182.
5. Zeckendorf's Theorem http://en.wikipedia.org/wiki/Zeckendorf's_theorem
Retrieved on 13-Jan-2014.
6. Baker, John and Sugden, Stephen (2014). The role of spreadsheets in an investigation of Fibonacci numbers. *Spreadsheets in Education (eJSiE)*: Vol. 7: Iss. 2, Article 2.

Appendix A – some classroom exercises & activities

1. Construct an Excel model to tabulate the first 20 terms of the Fibonacci sequence by using its defining recurrence.
2. Extend your model of the previous question to tabulate the partial sums of the Fibonacci sequence. Compare the sum to n terms with $f(n + 2) - 1$ and verify that these match.
3. Create an Excel model to tabulate partial sums of the odd terms of the Fibonacci sequence. Verify by parallel tabulation that

$$\sum_{j=1}^n f(2j - 1) = f(2n)$$

4. Create an Excel model to tabulate partial sums of the even terms of the Fibonacci sequence. Verify by parallel tabulation that

$$\sum_{j=1}^n f(2j) = f(2n + 1) - 1$$

5. Explain the notation used in section 1, i.e.,

$$k = \sum_{i=2}^n e_i f(i) \text{ for all } k \in Z$$

6. Again with reference to section 1, explain the equations " $e_i = 0$ or 1 " and " $e_j e_{j+1} = 0$ ".
7. Using only pencil and paper, compute the Zeckendorf representation for the integer 44.
8. Construct a 5-bit binary table in Excel. It gives the binary forms for the integers 0 to 31 in a 32 by 5 table, and is also used as the "input" combinations for a 5-bit truth-table. Building the table may be done in several ways. the simplest is by repetitive "copy-and-paste". Another method is to use a clever IF statement in Excel. You will find full details of the steps to be used for both of these methods in Appendix B.
9. Using your binary table of the previous question, extend your excel model to generate the integers 0 to 31 in rows 1 to 32 of some column after column E of your table. Hint: use Horner's multiplication method. This also serves as a check that your binary truth table is, in fact, correct.
10. Tabulate the function $\lfloor (n/2) \rfloor + 1$ in Excel for $2 \leq n \leq 20$ and show that this generates the correct values for $Z(f(n))$. Use the function FLOOR in Excel.
11. Create an Excel model to generate a binary (truth) table with 5 columns. How many rows will this table have? The first row will be 5 zeroes and the last will be 5 ones. Use Excel formulas and functions to do this.
12. A *bit* (binary digit) is either zero or one, and a *bitstring* is a sequence of such values. A Zeckendorf bitstring is given in a contiguous range (just a row or column of zeroes and ones) in Excel for the representation of an integer n .

Explain how to use the SUMPRODUCT function to generate n from this sequence of bits.

- Given a positive integer $n > 1$, we know that $f(n)$ has $\left\lfloor \left(\frac{n}{2}\right) \right\rfloor + 1$ Zeckendorf representations in terms of Fibonacci numbers. Describe an algorithm to generate each of these representations. Implement it in Excel.

Appendix B How to construct a binary table in Excel

The truth table generator, TableGen.xlsm, is supplied with this paper. It will create binary tables for you, or, you can even record your own macro to do this if you like. Here are the steps. For convenience and clarity, the two methods are found on two separate worksheets of this workbook.

Method 1 (Repetitive copy-and-paste)

- In range A1:B2 put the following small table.

	A	B	C
1	0	0	
2	0	1	
3	1	0	
4	1	1	
5			

- This is a binary truth-table for $n = 2$ variables. The rows correspond to bit-strings for integers from 0 to $2^n - 1 = 3$.
- The method works recursively: you get the table for $n + 1$ by extending the table for n . We now describe the extension process, which is repeated until you have the desired value of n . Note that the number of rows in the table doubles at each step, that is, as n increases by 1, the row count doubles. Here we go.
- Click on column A heading and go Insert. This inserts a new column at the extreme left of the current binary table.
- Click in cell A1 and type "0" there (no quotes).
- Double-click the fill-handle of A1. The range A1 down to the current bottom of the table (this will be row 4 for the first time through, row 8 for the second, then 16, and so on) will fill with "0".
- Ctrl-Home
- Right-arrow
- Ctrl-shift-right arrow. The entire first row of the table except A1 should now be selected.
- Ctrl-shift-down arrow. The entire table except for column A should now be selected.
- Ctrl-C (copy the selection to the clipboard).
- Ctrl-down arrow
- Down arrow

14. Ctrl-V (paste from clipboard)
15. Ctrl-left arrow
16. You should now be just below a block of zeroes in column A. Type "1" (no quotes) in that cell.
17. Double-click its fill-handle and you are done.
18. If you want to insert another column, go back to step 4 and repeat the whole process from there.

Method 2

With this technique, one must first decide on the table size. For the sake of definiteness, let's choose $n = 4$, so we will be working in columns A through D. You can adapt the following steps to cater for larger tables, if desired.

1. Fill the top row, i.e. range A1:D1, with zeroes.
2. Put the formula " $= 1 - D1$ " in cell D2. This will toggle the values in Column D from 0 to 1 and back again.
3. Fill this formula down from D2 to D16.
4. For the interior of the table, adopt the following rule: If the bit to your right has just decreased then you toggle to get the next bit in your column. This leads to the formula " $= IF(D1 < D2, 1 - C1, C1)$ ".
5. Fill the range C2:A16 with this formula. That's it!