Electrical Circuit Enumeration

James Thomas Parker, MS
Post Graduate NC State University
gjtp3986@gmail.com

Abstract: This paper uses the art of combinations and permutations to allow a one-to-one mapping between the natural numbers and all unique electrical circuits that can be created on a wire wrap board. It has been shown by exhaustive computer analysis to create all unique circuits possible using 1 to 9 pins or connections points where all the points are different.

In order to map integers into circuits, it is first necessary to get an exact count of how many circuits are possible. Let's consider a bare wire wrap board with n pins on the back side. One could just wire wrap every pin to every other pin and that would create an upper limit of n! possible circuits where every pin must connect to a pin higher than itself. But counting the number of possible electrical circuits is not that simple. For example, pin 1 could connect to pin 2 and pin 2 could connect to pin 3 and this would represent one of the possible n! circuits. But pin 1 could connect to pin 3 and pin 2 could connect to pin 3 and this would be a different one of the possible n! circuits but electrically, it would be an identical circuit.

Instead of counting pins, we must count nodes. We assume a fixed number of nodes and determine how many ways the pins can attach themselves to the nodes assuming the pins each have a unique identity. In a real wrap board, each pin may not be completely unique in its function but for our current illustration, we will assume each pin goes to a unique component on the wire wrap board.

Consider the wire wrap board in figure X. We stretch the two dimensional board into a single dimension and number the pins from left to right. The horizontal bars above the pins are nodes much like computer busses. They are not numbered because they have no identity. They are completely interchangeable.

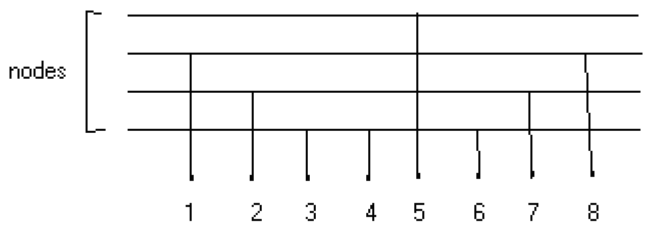

Figure X – Wire Wrap Board

It took a long time to figure out that each bar must be touched by at least one pin. If a bar is open, one should be counting circuits with fewer bars. An open pin connects to a bar by itself but it must touch a bar. Pin 5 in figure X is an open pin because it connects to no other pin but it must still connect to a bar.

To count the number of circuits in figure X we introduce the function Phi(n,m) where n represents the number of pins, m represents the number of bars, and Phi represents the number of possible circuits where every bar and every pin is used. To count the number of circuits in figure X we use Phi recursively.

First of all, if m=1, Phi will always equal 1 because if there is only one bar, all the pins must connect to that bar. If m is any other value, there are m^n ways the pins can connect to the bars but now we have allowed identity of the bars so we must divide out count by m! There is still a problem. Sometimes in considering m^n connections, a bar will be unused. Sometimes two bars will be unused, etc. To count the number of circuits in figure X that had one bar unused we consider combinations(4,3)*Phi(8,3) since there are 8 pins and we are only using 3 bars. Where combinations(a,b)=a!/((a-b)!*b!) But for the moment we are going to give those three bars an identity since m^n assumes an identity of the bars. They can be permuted 3! ways and our count becomes 3!*combinations(4,3)*Phi(8,3). This count gets subtracted from m^n or thus 4^8. But we are not finished removing all the circuits with open bars. We could have two bars open. In short we must sum all of the possibilities and subtract from m^n.

This gives the function m^n – sum(i to m-1 combinations(m,m-i)*(m-i)!*Phi(n,m-i))

Finally, the bars have no identity so their permutations must be divided out of Phi. Combining and reducing the function, we get:

Phi(n,m) = 1 if m=1

Phi(n,m) = (1/m!)(m^n-sum(i=1 to m-1((m!/i!)*Phi(n,m-i)))

Understanding Phi is the first step in enumerating electrical circuits. The second step is understanding the function Pi.

Pi(n,m,p)=Phi(n,m)-(all circuits that have any bars with p pins or less)

The Pi function is directly needed to convert integers into circuits. Phi is mainly used with Pi. Although the bars have no identity, for enumeration purposes we always place the bar with the fewest number of pins on top and arrange the other bars with the pin count in ascending order.

In figure X, the top bar has one pin, the next two bars have two pins, and the bottom bar has three pins. This has the pin count in ascending order. When mapping the integers into circuits, the lowest integers will have one pin on the top bar. The number of possible circuits with only one pin on top for figure X is 8*Pi(7,3,1). Each of the 8 pins can be placed on the top bar and all the possibilities are multiplied by the number of circuits with 7 pins and 3 bars left, excluding any bars that have only one pin. By the process of division, an integer in the proper range can determine if only one pin is on the top bar and which pin it is.

The function Pi uses the third argument p to determine how many circuits exist excluding bars with p pins or less. First, it uses Phi to determine how many circuits are possible overall. Then it subtracts out all the possibilities with circuits up to p pins. Using figure X, we calculate Phi(8,4) and subtract out 8*Pi(7,3,1) to partially calculate the function Pi(8,4,1). We left out the possibility that the second bar may also have only one pin so we have to subtract out combinations(8,2)*Pi(6,2,1) to allow for that possibility. Finally we must allow for 3 bars to contain only 1 pin each. In general, we calculate the sum:

Pi(n,m,1)=Phi(n,m)-sum(i=1 to m-1 (n!*Pi(n-i, m-i, 1))/((n-i)!*i!))).

It would be nice if the computation would end there but the third argument of Pi may be greater than 1. To allow for p to be greater than 1, we must do a double summation. Phi(n,m) is still calculated and then we subtract out a double summation with the inner summation the same as above and a new outer summation of sum(k=1 to p) to consider all the possibilities of more than 1 pin on the top bars. The combinations of possible pin configurations is multiplied by Pi(n-ki,m-i,k) as one would expect with k pins on the top bars. The only other consideration is that the combinations become a little more difficult to calculate when k is greater than 1. The k pins can be permuted and the i bars can also raise the permuation count to the i power so our final formula is:

Pi(n,m,p) = Phi(n,m) – sum(k=1 to p sum(i=1 to m-1 (n!*Pi(n-k*i,m-i,k))/((n-i*k)!*i!*(k!)^i)))

Using Pi, an integer can determine how many pins are on each bar and then the integer further breaks down to determine which pins actually go on each bar. To restrict the enumeration, the pins are arranged in groups. Bars with 1 pin only are on top. Bars with two pins go next, etc. Each group of bars with the same number of pins has the head pins (the left most pin) in ascending order. The tail pins follow each head pin in ascending order. For example consider the following: The pins could be numbered as follows:

6 8
1 3 5
2 4 7

Notice the head pins ascend as you go down the bars. The tails follow each head in ascending order. It isn't really necessary to arrange the pins in some sort of order as long as the same pins belong to each other in a node (bar).

The same circuit could be represented by

5 1 3
8 6
7 4 2

It's just the algorithm used represents this circuit by the first pin pattern given.
For detailed logic examples, run the demonstration program given as a reference.

[1] Demonstration program by James T. Parker found on web 5/27/18
mathshowcase.com/back2ckt.cpp