The Filter Wizard
issue 4: An E96 Formula: How can you resist it?
Kendall Castor-Perry

Spreadsheets are brilliant for filter design, and I write much less design software these days. Don't get me wrong, I love the BASIC interpreter that I use, BBC BASIC for Windows (www.bb4w.com). It's a compact but complete Windows programming environment, sharing a direct lineage with the BASIC dialect found in the original BBC Micro, which if you are young, or not British, you may never even have heard of. It's so compatible that I can run filter synthesis, analysis and optimization programs I wrote up to 25 years ago *completely unmodified* – just orders of magnitude faster than on the Beeb's 2MHz 6502. If I post BASIC programs relevant to this column, that's what I'll use. But this piece isn't about BASIC; it's about spreadsheets.

These days, I mostly code design equations for filter circuits directly into Excel. And, for the Microscoffs out there: yes, you *can* use other spreadsheets, including free ones. In fact, this work began in the far-off Quattro Pro days. Calc (from the OpenOffice suite) seems to have lost the 'general' number format, which makes formatting some sheets much harder work than it should be, IMHO.. The Google Docs spreadsheet seems to work, though I've not tried it out on complex stuff. Both also have a 'solver' capability, which is great for circuit optimization, and I'll cover that in more detail in some other columns.

Anyway, how do you get from the many decimal places in each calculated component value, to a design that uses components that you can actually *get*? You have to turn them into 'preferred values'. I do this manually for capacitors – it's a habit from my formative years where you *purchased* accuracy in resistors but *selected* it into capacitors. Accurate capacitors aren't readily available in a structured, closely spaced sequence. In fact, I'm starting to find that accurate capacitors aren't available, period. Active filter design is being deprived of its life-blood – wail, gnash, sob. But you have to work with what you can get, and at least you can get closely spaced fractional-percent resistors in the E96 range to go with the few stable capacitors still on the market.

Even with the bounty of 96 values in each resistance decade, the value spacing is still wide compared to the tolerance (0.1% resistors are readily available at quite low cost). Designing an entire filter and then replacing all the theoretical values with the closest E96 value in one go is sub-optimal when there are many components. You can get sizeable implementation errors, especially in sensitive 'sharp' filters with values in close ratio.

A much better approach (my column, my opinion) is to form the design process into a series of steps, at each of which a new component value is calculated. This is then quantized onto the available component grid and fixed before moving onto the next component calculation, which takes the already-chosen values into account. The spreadsheet paradigm is ideal for automating this approach. But still, how to determine *which* E96 component value to use?

We could loop over the choices available from a big list, until we find the closest fit. But spreadsheets don't do looping very well, unless you're prepared to write some code in VBA, and that rather defeats the object. If I'm going to write a BASIC program, I'll write it in BBC BASIC! So, what I needed many years ago was a tractable, non-iterative cell formula for turning a number, representing a resistor value, into the nearest E96 component value, so that each component can be fixed and then used in subsequent calculations.

The E96 value multipliers are distributed logarithmically over a decade. Each successive value is higher than the previous by the one-ninety-sixth root of 10. Round off the results of the resulting geometric progression:

1, 1.024275221, 1.049139729, 1.074607828, 1.100694171...

to three significant digits, and Hey Presto! The key multipliers in the E96 series fall straight out. The people who devised this series must have sensed that we'd want to automate our design process one day! All that is needed to convert an actual resistance value is to extract the power of ten for the actual decade the resistor is in, find the E96 multiplier, and then put the whole value back together.

The scheme also appears to work well for E48, but I'm not sure the rounding always gives the accepted value for the E192 scale (I've never needed 'em so haven't tested this to any depth). And of course the more workaday scales, E24 and below, don't fully overlap with the E96 range.

So, here's a cell formula that turns a positive real resistance into the nearest E96 component; copy it straight out and paste it in wherever you need it. The value to be converted is in cell E11, in this example:

```
=( 10^TRUNC(LOG(E11))*0.01*TRUNC((0.5 + 100*10^(1/96)^TRUNC(0.5 +
96*(LOG(E11) - TRUNC(LOG(E11)))))) )
```

This has served me well in all sorts of filter design spreadsheets over many years; enjoy! Just occasionally, the formula selects the value 'on the wrong side' of the closest one. Changing the offset in the innermost TRUNC() function biases the selection to one side (sometimes useful if you always want next-higher or next-lower for a particular scheme). So these days I sometimes use a more laborious conditional version which tests which is closer:

```
=IF( ( 10^TRUNC(LOG(E11)) * 0.01 * TRUNC((0.5 +
100*10^(1/96)^TRUNC(0.75 + 96*(LOG(E11) - TRUNC(LOG(E11))))))) *
( 10^TRUNC(LOG(E11)) * 0.01 * TRUNC((0.5 + 100*10^(1/96)^TRUNC(0.25 +
96*(LOG(E11) - TRUNC(LOG(E11)))))) ) > E11*E11 , ( 10^TRUNC(LOG(E11)) *
0.01 * TRUNC((0.5 + 100*10^(1/96)^TRUNC(0.25 + 96*(LOG(E11) -
TRUNC(LOG(E11)))))) ) , ( 10^TRUNC(LOG(E11)) * 0.01 * TRUNC((0.5 +
100*10^(1/96)^TRUNC(0.75 + 96*(LOG(E11) - TRUNC(LOG(E11))))))) )
```

You can pre-calculate some cells if you’re a bit frightened of long formulae; adapt them to your own needs. Anyone have some other useful cell formulae for circuit design? -
Kendall