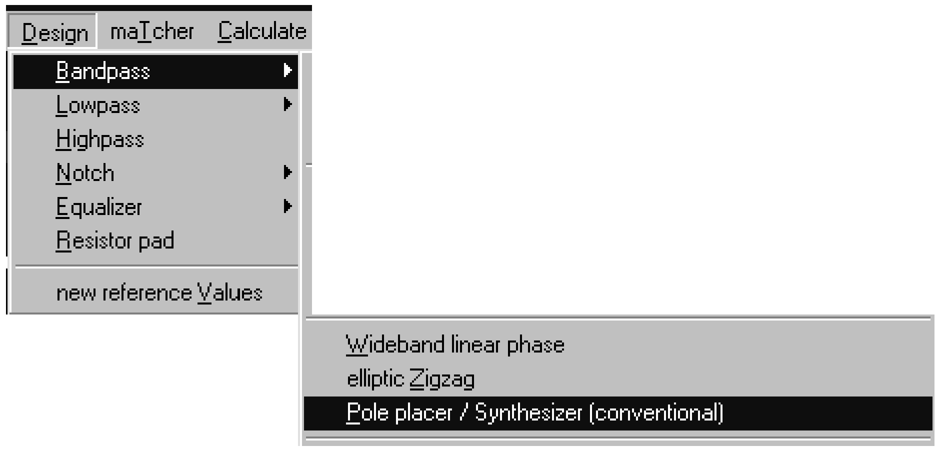
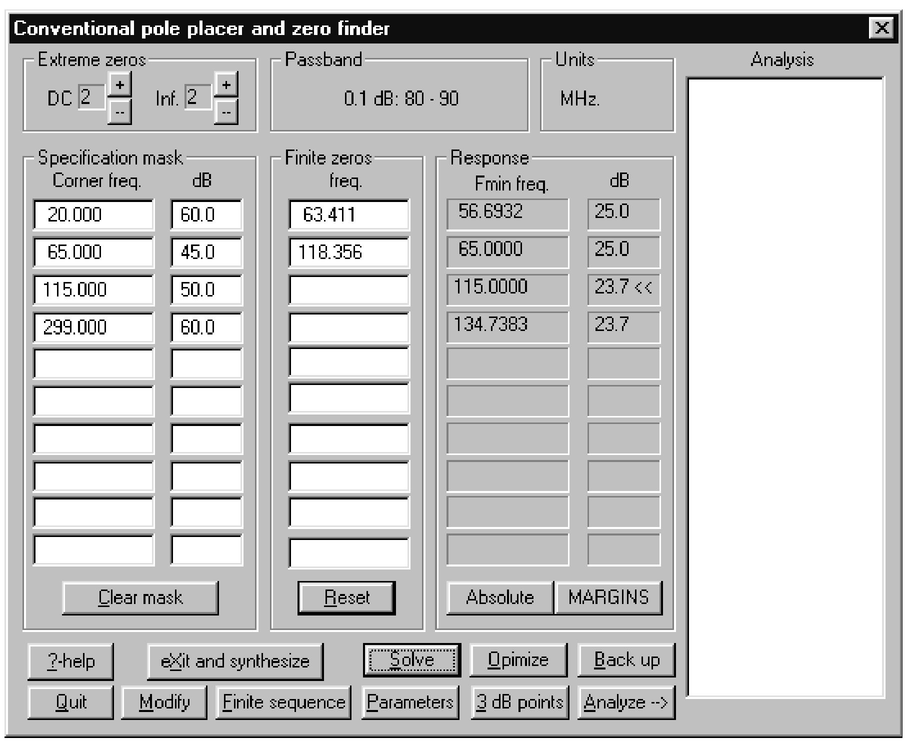
Conventional Bandpass Pole Placer



Pole placer / Synthesizer (conventional)

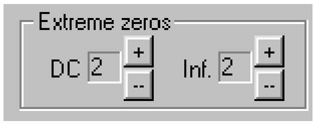
The pole placer will synthesize conventional (not parametric) bandpass filters to a manually entered set of parameters, or will generate these parameters in response to a passband / stopband "mask" specification. Up to 10 finite transmission zeros ("notch" sections) may be generated above or below the passband in any order desired. In addition, any combination of zeros at the extreme frequencies may be specified so long as the total is an even number of at least 4.



Any design saved that was done using the pole placer will generate a file that contains the mask, the "factors" necessary to synthesize the filter and any manually entered structure codes that were used. The file will have the active file name with the extension ".PPD" (such as DEFAULT.PPD). A recalled design can be routed directly to the placer screen without keying in the mask again allowing the placer to go directly into the evaluation mode where the margins are displayed.

The pole placer dialog box is shown above with the specification masked of the example to follow entered.. The dialog box is divided into several areas dedicated to various aspects of the pole placed filter design problem.

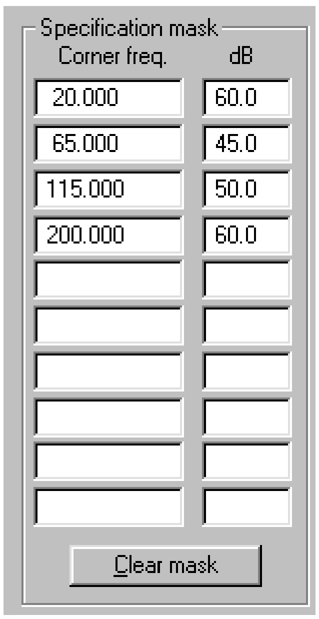
Extreme zeros



Zeros at DC and Infinity can be set here. The lower limit is a total of 4. The total must also add to an even number. Use the [+] and [-] buttons to adust the numbers up or down. When doing a lowpass reference only the zeros at Infinity will function.

The passband specifications from the parameters menu are displayed here along with the units (Notation) for all variables.

Poles and zero locations to mask



Enters the pole placer section which is used to establish the necessary arrangement of transmission zeros to meet a set of specifications. These specifications take the form of a "mask" consisting of segments in frequency over which a certain stopband rejection is needed. The mask is keyed in from the keyboard but the last ones entered may be reused if they are still appropriate.

Shown below is a typical stopband mask. This one represents a filter with the following specifications:

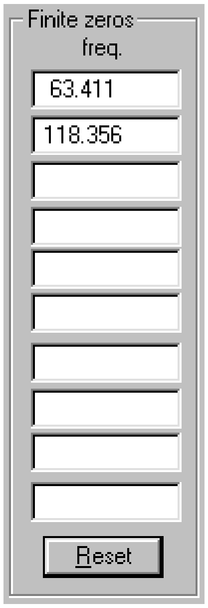
```
Passband:  .1 dB ripple bandwidth. 80 - 90 Mhz.
        (Passband data must be entered on the parameters menu
         before keying in the mask)
Stopbands:
        > 60 dB DC to 20 MHz and above 200 MHz.
        > 45 dB 20 - 65 MHz
        > 50 dB 115 - 200 MHz.
        > 60 dB 200 Up
```

Break the stopbands into rectangular sections. In the lower stopband (LSB), a section is defined by the highest FREQUENCY, ATTEN. (dB) of that section .In the upper stopband (USB), a section is defined by the lowest FREQUENCY, ATTEN. (dB) of that section. All section and zero frequencies will be sorted to ascending order. This specification mask could be drawn graphically like this:

```
               ─────────────────────────────────────────── 0 dB.
                           │                 │
                           │  ┌───.1 dB ───┐ │
                           │  80         90  │
        (LSB)              │  MHz.      MHz. │         (USB)
                           │                 │
                           │                 │
         20 MHz.           │                 │
              ┌── 45 dB. ──┘                 │         200 MHz.
              │        65 MHz.               └──── 50 dB. ───┐
 ── 60 dB. ───┘                         115 MHz.             └── 60 dB. ──
 Section 1    │ Section 2  │                 │    Section 3  │   Section 4
```

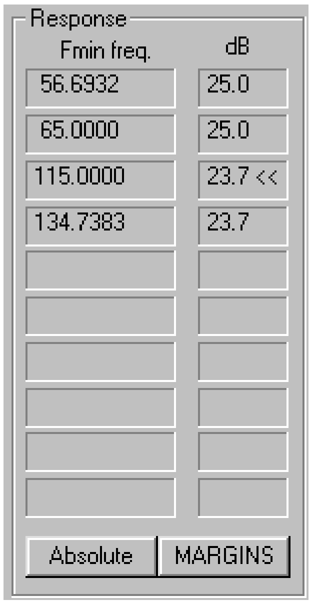
The stopband mask corner frequencies should be entered into the “Corner freq.” edit boxes first, from top to bottom by pressing the <Tab> key to move downward. The attenuation at each frequency will be assumed to be 30dB. Simply move to the top of the “dB” edit boxes and correct the attenuation requirements from top to bottom using the <Tab> key as with the frequency inputs.

The [Clear Mask] button clears all the inputs and lets you start over with a new specification mask.



The Finite zero area of the display shows the frequency of each of the finite zero notch frequencies. These are edit boxes and will allow the user to adjust the frequency of each zero as needed. They will also be adjusted automatically by the [Optimize] button when working with a specification mask. Finite zero frequencies may be input directly also. As with the mask edit boxes, the <Tab> key will move from top down as you enter zero frequencies.

The [Reset] button will clear all the zero frequencies. Use this to start a new possible solution to a specification mask. Press [Reset] and try a different set of the extreme zeros (DC and infinite) and press the [Solve] button again.



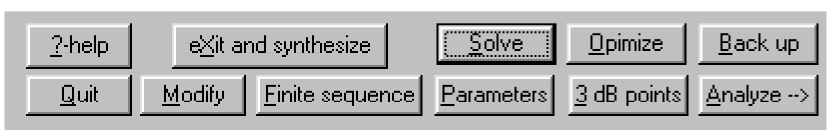
The Response area displays the attenuation below each of the corner frequencies of the stopband mask The display can be in one of two modes indicated by the red “light” on either side of the mode buttons.

[Absolute] displays the attenuation exactly as it would be displayed by the analysis program. That is, in absolute dB units.

[Margins] displays the attenuation relative to the attenuation requested by the stopband mask. A negative number means the attenuation is below that required. A positive number indicates the specification is being exceeded by that margin.

The corner frequency with the smallest margin (the worst case) is identified with arrows (<<).

The control buttons



[Optimize]

Adjust the frequency of all the finite zeros to equalize all the attenuation margins. Repeated applications will often improve the margins in the stopbands.

[Modify]

This is a direct path to the finite zeros. It will direct the input to the last finite zero midified.

[Back up]

This will undo the optimize operation. Do this before you add another finite zero, either manually, by the finite zero edit boxes, or by the [Solve] button.

[Analyze–>]

Used to generate tabulated data on stopbands with infinite Q. The results will be tabulated into the area to the far right.

[Solve]

This instructs the program to find a solution by initially adding finite zeros until the stopband margin is slightly shy of the mask requirements allowing you to decide the best way to finish the design. This operation is automatically followed by an optimization of multiple loops. At this point, using the [Solve] button again increments the extreme zeros each by 1 effectively adding a single all-pole type "tank" at the end of the network. Using the manual [Back up] button will restore the initial condition allowing another finite zero to be added instead. The condition that determines if a finite zero is added or if the extreme zeros are incremented is if two or more of the attenuation margins are equal as is the case after an optimization has been performed.

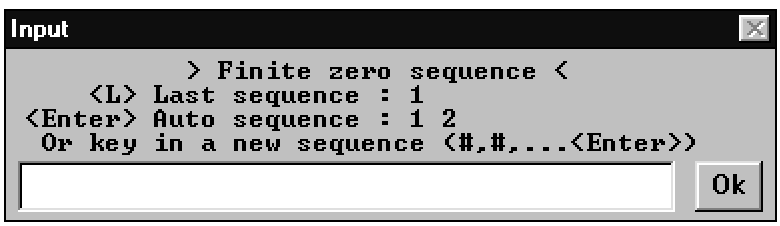
[3 dB points]

Causes the program to find and display the infinite Q 3 dB points. This is most useful when working with singly terminated designs as it indicates the crossover frequencies for adjacent channels when designing contiguous multiplexers. The results will be tabulated in the analysis area to the right.

[Parameters]

Allows the passband to be changed by bringing up the parameters menu.

[Finite sequence]



The finite sequence refers to the order in which two or more finite zero notch sections appear in the filter. This brings up a dialog box to allow the user to chose the sequence. It will only appear if there are two or more finite zeros.

```
        <L> Last sequence  : 2 1 3
             This is the last sequence used or recorded in the ".PPD" file.
    <Enter> Auto sequence  : 2 1 3
   This is the computer selected default sequence.
         Any desired sequence you like may be keyed in.
```

[eXit and synthesize]

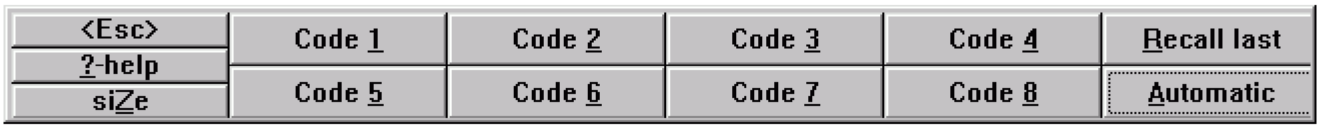
A filter is synthesized by building it up from sub-networks. The program will pick these sub-networks automatically by simply pressing the [Automatic] button. On occasion, you may want a different network than the one the computer picks. This option allows any topology you require to be specified by picking "Sub-networks" in the correct order manually.

These sub-networks and their reference code numbers are identified in the following window:

```
═════════════════════════════> Sub-network codes <═════════════════════════
 CODE  Schematic    CODE  Schematic   CODE  Schematic    CODE  Schematic

   1  ─── C ───       2  ───┬───        3  ─── L ───       4  ───┬───
                            L                                    C
                         ───┴───                              ───┴───
 ─── C ──┬─────                     ─── L ──┬─────
   5     L            6   ┌─ L ─┐     7     L           8   ┌─ L ─┐
         │           ──┬──┤     ├──         │          ──┬──┤     ├──
         C f-          L  └─ C ─┘  f-       C f+         C  └─ C ─┘  f+
 ────────┴─────      ──┴────────────    ────────┴─────   ──┴────────────
```

```
ENTER THE CODES FOR THE FILTER STRUCTURE     (R)-Recall last manual entry.
(Start at termination end.)  Zeros: DC= 3  Inf.= 1  Finite order= +-+
(Cr = Auto.) >
```



The job of picking these codes correctly is not easy and requires careful planning before proceeding. The codes must be picked so as to not only generate a filter with the right response, but one that can also be synthesized within the limits of the pole removal scheme used (H. J. Orchard) as well.

Extreme poles - DC and infinity

A good procedure to find the extreme zeros of a filter is to draw the desired filter circuit diagram and then draw its equivalent circuit at both DC and infinite frequency. This will allow the "extreme zeros of transmission" (the number of times an open circuit is followed by a short circuit or vice-versa).

```
For the following all-pole mesh filter:
        ─── C ── L ──┬── C ── L ──┬── C ── L ──┬── C ── L ──
                     C            L            C
        ─────────────┴────────────┴────────────┴────────────

Redrawn at DC, The circuit will look like this:
        ───   ───────┬──   ───────┬──   ───────┬──   ───────
                                  │
        ─────────────┴────────────┴────────────┴────────────
```

Each capacitor looks like an open circuit at DC. An inductor is equivalent to a short circuit. Note that there are 3 open-short-open "inversions" at DC. Multiple "opens" or "shorts" in a row count as only one. This counts as 3 "Zeros" at DC.

```
At infinite frequency, the equivalent looks like this:
        ────────    ─┬───────    ─┬───────    ─┬───────    ──
                     │                         │
        ─────────────┴────────────┴────────────┴────────────
```

Note that we have 5 "inversions" of open-short-open-short-open. This indicates 5 zeros at infinite frequency. This filter will have a sharper upper skirt than lower skirt.

```
The sub-network codes that generate extreme zeros are the following:
    1  ─── C ───       2  ───┬───       A zero at DC. each
                             L
                          ───┴───

    3  ─── L ───       4  ───┬───       A zero at Infinity each.
                             C
                          ───┴───
```

Finite Zeros

Finite zeros of transmission are simply "notches" in the stopband. These can be either above or below the passband and must total no more than 9. Finite zeros above the passband are referred to by the program as "f+". Below the passband they are indicated as "f-". The sub-network codes that provide finite zeros are:

```
 ─── C ──┬──────                           A Finite zero BELOW the
  5      L            6    ┌─ L ─┐           passband (f-) each.
         │           ──┬───┤     ├──
         C f-          L   └─ C ─┘ f-
 ────────┴──────     ──┴────────────

 ─── L ──┬──────                           A Finite pole ABOVE the
  7      L            8    ┌─ L ─┐           passband (f+) each.
         │           ──┬───┤     ├──
         C f+          C   └─ C ─┘ f+
 ────────┴──────     ──┴────────────
```

Sequence of sub-networks:

The sub-networks can be arranged in any order necessary to synthesize the filter topology you want, with a few limitations. Here are the rules:

1-No single type of component may be repeated twice without being separated by some other. That is, no parallel capacitors or series inductors even when they are part of different sub-networks. This makes the following example sequence illegal (and illogical as well):

```
3 followed by 7:    ─── L ───*─── L ──┬──────
                 code: 3     Code: 7  L
                                      │
                                      C f+
                    ──────────────────┴──────
```

2-Each individual end of the final filter must act the same at DC as it does at infinite frequency. That is, both DC and Infinity must be short circuits, or both must be open circuits (infinite Zo). This sequence would therefore be illegal:

```
1 followed by 8:             Code: 8 ┌─ L ─┐
                      ─── C ───*───┬───┤     ├──
                    Code: 1        C   └─ C ─┘ f+
                      ─────────────┴────────────
```

This network would appear as an "open" at DC and as a "short" at infinity. When the codes are keyed in manually a test is made to see that this condition is met from both ends of the filter. In case of an error an arrow will point to the offending element.

For example:

```
(Cr = Auto.) > 1 2 5 8 1 4 2 1 2 1
   From the left ——> ^  In this case, Sub-network code 8 is the offender.
```

3-A network must not end with a finite zero. The finite zero is called a "partial removal" and requires that a "full removal" be performed further along the network.

4-The total of the extreme zeros must be an even number. 3 zeros at DC plus 5 zeros at infinity equal 8, an even number.

5-Sub-networks that provide finite zeros above the passband (f+) and below (f-), must be specified in the same order as they are shown below the lower right corner of the Sub-network window. "+-+" in the example below.

```
└──────────────────────────────────────────────────────────────────────────────┘
ENTER THE CODES FOR THE FILTER STRUCTURE     (R)-Recall last manual entry.
(Start at termination end.)  Zeros: DC= 3  Inf.= 1  Finite order= +-+
(Cr = Auto.) >                                    ----------------------
```

Redundant elements

It must also be kept in mind that the program synthesizes the minimum component network that has the requested arrangement of zeros. This will make the filter you get look like something entirely different than you expected, particularly with all-pole filters. To arrive at the minimum component equivalent of the topology you want you should make a study of the Norton transform command in the circuit editor and be aware of the extra parts that this transformation generates. The minimum component circuit will be the circuit that will result if all the components that would be generated by all the transforms in your final topology were REMOVED!

For example, let's look at the mesh filter used earlier to explain the idea of extreme zeros. The center element of each transform that generated each extra part is shown below the schematic and the extra part associated with each is identified above it:

```
        Associated "extra" part: 1               2     3
                                 |               |     |
                  ─── C ── L ─┬─ C ── L ─┬─ C ── L ─┬─ C ── L ──
                              C          L          C
                  ____________┴__________┴__________┴_____________
                              |          |          |
                              1          2          3
                       (Transform number center element)
        Removing these 3 "extra" series parts yields this:
                  ─── C ── L ──┬── L ──┬─ C ──┬── L ───
                               C       L      C
                  _____________┴_______┴______┴________
        The codes for this network would then be:
(Cr = Auto.) > 1 3 4 3 2 1 4 3 <Cr>
```

After the filter is synthesized, the transforms can be done at each shunt element using the circuit editor to arrive at the final design. This is usually necessary because of the source to termination impedance ratio that an asymmetrical minimum component filter requires.

Network transformations

After a filter has been synthesized, it is usually necessary to do a considerable number of transformations to reduce the parts value spread and the source / termination impedance ratio. At higher frequency, you must also see to it that there are capacitors to ground at every point in the circuit that could be affected by distributed capacity. The job of doing these transformations can test your imagination as the procedures involved can be rather elusive. The possibilities are endless, but below is a random selection of situations that can develop within typical networks as you go through a typical transformation session. It is hoped that these examples will give you an idea of how to proceed. A typical problem is first described with the schematic shown to the left. One possible solution is shown to the right after the procedure described below it has been applied.

Problem: The series connected inductor at branch 10 has a desirable value but the L/C ratio at branches 7 and 8 is much too high, That is, L8 is huge compared to L10:

```
                                             │
                                             │              ┌─┴─┐
      │                ┌─┴─┐             4.  │              C   L
  4.  │                C   L                 │              └─┬─┘
      │                └─┬─┘                 │                │
  6   ├──────── C ───────┤               6   ├──────── C ─────┤
  7.  ├──── C ─── L ─────┤               7.  ├──── C ─── L ───┤
      │                  │                   │                │
  9   │                  C     ---->     9   │                C
      │                  │                   │                │
 10   │                  L              10   ├──────── C ─────┤
      │                  │                   │                │
                                        11   │                L
                                             │                │
```

Solution: The ratio of the inductor values at branches 8 and 10 is determined and a Norton transform is performed at branch 9 using this ratio. The ratio used must be < 1.0 or the resulting branch 10 will be negative. That is, L8 must be much greater than L10.

Problem: The parts values at branches 1,2 and 6-8 are reasonable but the L/C ratio at the series trap C4 and L5 is much too high. That is, L5 is huge compared to L1 and L8:

```
                                         0   ┌─ Termination ──┐
                                         1   ├──────── L ─────┤
                                         2   ├──────── C ─────┤
  0   ┌─ Termination ─┐                      │                │
  1   ├──────── L ────┤                  3   │                C
  2   ├──────── C ────┤                      │                │
      │               │                  4   ├──────── C ─────┤
  3   │               C                      │              ┌─┴─┐
      │               │        ---->     5.  │              C   L
      │             ┌─┴─┐                    │              └─┬─┘
  4.  │             C   L                7   ├──────── C ─────┤
      │             └─┬─┘                    │                │
  6   ├──────── C ────┤                  8   │                C
  7.  ├──── C ─── L ──┤                      │                │
                                         9   ├──────── C ─────┤
                                        10.  ├──── C ─── L ───┤
```

Solution: Split C3 into two series capacitors with the **Split** command, then move one of the two to the other side of the notch (C4 and L5). Perform a Norton transform at each series capacitor using the ratio of L1 / L5 and L5 / L1 at the other. Later, you can come back and perform another transform at these same two series capacitors to provide for an even nicer parts value spread all through the network or even force the new shunt capacitors (C4 and C7) to standard values.

Problem: The series input end configuration is not compatible with the impedance matcher (No L/C resonant circuit is recognizable):

```
                                                  │                  ┌─┴─┐
                      │            ┌─┴─┐      6.  │                  C   L
   6. │                            C   L          │                  └─┬─┘
      │                            └─┬─┘      8   ├──── C ─────────────┤
      │                              │            │                    │
   8  │                              C  ---->  9  │                    C
      │                              │            │                    │
   9  ├───── C ──────────────────────┤       10   ├──── C ─────────────┤
      │                              │            │                    │
  10  │                              L       11   │                    L
      │                              │            │                    │
  11  └──── Source ──────────────────┘       12   └──── Source ────────┘
```

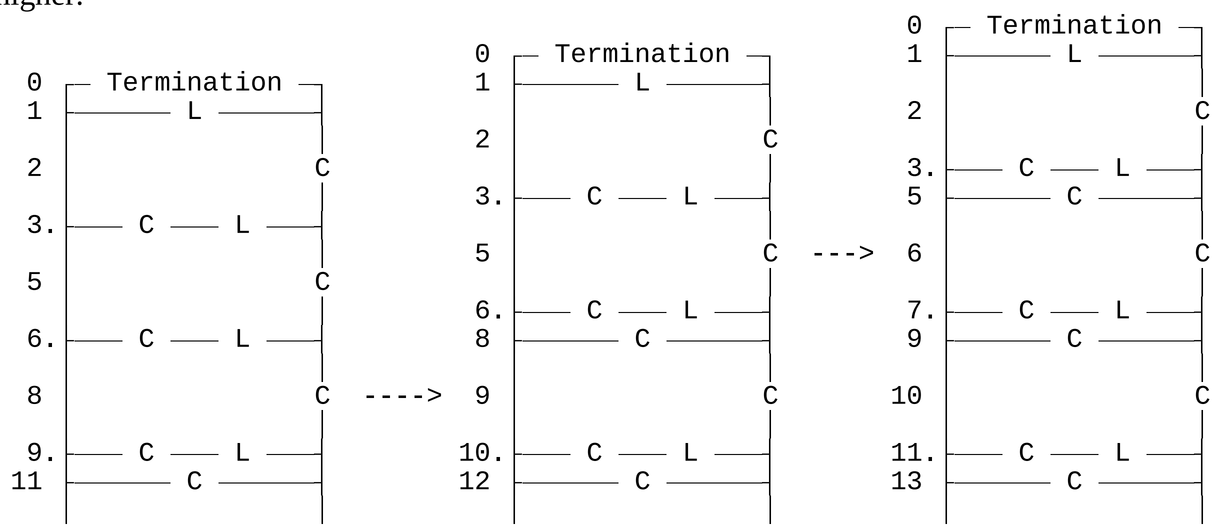
Solution: A Norton transform applied at the series capacitor C8 using a ratio < 1.0 will generate a "pi" configuration that the matcher module can easily handle. Note that this "pi" circuit could also be converted to a "T" which the matcher can also handle. The impedance at this end of the filter must be less than the system impedance into which it is to be matched.

Problem: The parallel input end configuration is not compatible with the impedance matcher (No L/C resonant circuit is recognizable):

```
                                                  │                    │
   8  │                        C             9    │                    C
      │                        │                  │                    │
   9. ├── C ── L ──────────────┤             10.  ├── C ── L ──────────┤
  11  ├───── C ────────────────┤             12   ├───── C ────────────┤
      │                        │                  │                    │
  12  │                        C    ---->    13   │                    C
      │                        │                  │                    │
  13  ├───── L ────────────────┤             14   ├───── C ────────────┤
  14  └──── Source ────────────┘             15   ├───── L ────────────┤
                                             14   └──── Source ────────┘
```

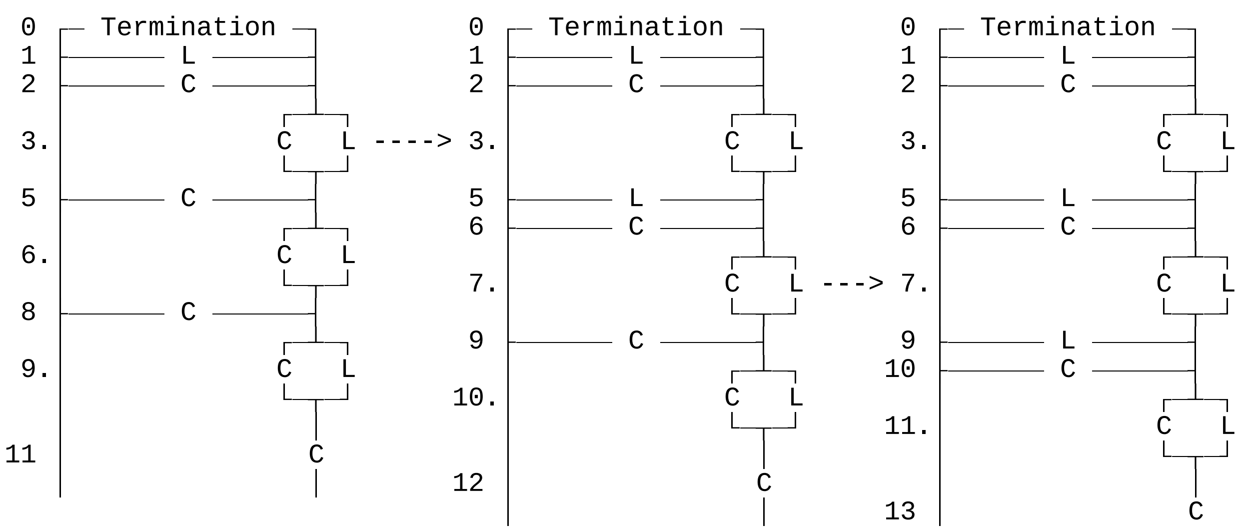
Solution: A Norton transform at the series capacitor C12 will generate the need capacitor in parallel with the shunt inductor. The transformation ratio must be > 1.0.

Problem: Several lower stopband finite zeros (f-) in a row are raising the termination successively higher.



Solution: Applying Norton transforms at each series capacitor using a ratio < 1.0 moving toward the termination will equalize each parallel "notch" L/C ratio.

Problem: Several upper stopband finite zeros in a row ended by an L/C "tank" at the end.



Solution: Working from the termination backward toward the source end, perform Norton transforms at each series connected L/C "notch". The impedance ratio must be > 1.0. This will generate a new inductor each time forming the necessary L/C "tank" to support the next transformation. The **Iterate** editor command is also useful in this situation to automatically reduce the value spread.

Problem: Previous transforms have generated a shunt three element network where the inductor (L8 in the left schematic below) is very large compared to other inductors in the filter.

```
                                 |                        |                  |
                  |          |   5  |                  C      5  |                  C
 5   |                  C        |                  |        |                  |
     |                  |        6  |------ C ------|      6  |------ C ------|
 6   |------ C ------|  ----> 7  |------ C ------|  -----> 7  |--- C -┐        |
 7.  |--- C --- L ---|       8.  |--- C --- L ---|      8.  |--- L -┴-- C ---|
     |                  |        |                  |        |                  |
 9   |                  C    10  |                  C    10  |                  C
     |                  |        |                  |        |                  |'
```

Solution: Apply a dipole transform at the lowest branch number of the three (6). The inductor in the equivalent network will be much smaller. By splitting the single shunt capacitor (C6) into two (**Split** command), the size of the transformed inductor may be adjusted by varying the value of C7 (middle schematic).

This method can also be used to absorb the distributed capacity associated with the inductor. If the value of the capacitor involved in the transform (C7 in the middle schematic above) is very small compared to the series capacitor (C8), its value will be nearly equal to the resulting capacitor across the inductor in the equivalent network (C7 in the right schematic above). The split command (**Split**) allows the value of one of the two split capacitors to be specified. When building the filter, simply leave out C7 as it is already part of the inductor L8.

The exact relationship is:

```
  ┌── L1 ──┐                      ┌── L2 ── C3 ──┐
 ─┤        ├── C2 ──             ─┤              ├─
  └── C4 ──┘                      └───── C1 ─────┘
```

$$C4 = C3 \left[\frac{1}{Sqrt(L1/L2)} - 1 \right]$$