Ladpac-2008

November 3, 2008, Wes Hayward, w7zoi (w7zoi@arrl.net)
(updates 7Nov08, 14Nov08, 18May09)

Contents

1. General Information.

Ladpac is short for "Ladder Package." It's a package of programs primarily aimed at the design and analysis of ladder filters. A few other programs are included even though they are slightly outside the central ladder theme.

1.1. System Requirements:

This software was written for use on a computer using Windows 95, 98, ME, XP, NT, and Vista. The system should have 5 Meg of RAM or more, with another 5 Meg available on a hard disc for storage of programs and files. An Intel Pentium processor with a speed of 300 MHz or greater is recommended, although we often use the programs on a slower machine. A monitor set for a color display with a resolution of 1024 x 768 is recommended. An earlier version, Ladpac-2002, was constrained to 640 x 480 pixels. A few of the earlier programs from that collection are included for those folks using such a computer.

1.2. Installation.

The installation requires nothing more than putting all programs into one directory. They can be copied directly from the distribution CD to a directory of your choice. Alternatively, run setup.exe and follow the on-screen directions to install the software.

Many of the programs within LADPAC will read from or write to the disc. The filter design programs will generate output files that can then be read by the analysis program, GPLA. Circuits can be further edited with Ladbuild.exe. It is generally useful to have the programs and the circuit files in the same directory.

1.3. General characteristics.

The various programs have many similarities in the way they function. We will list some of the common characteristics as an aid to understanding their use. Generally, the methods that we expect from the Microsoft Windows © environment are used.

1.3.1. The programs were written in Delphi 3.0 from Borland. This was our introduction to this environment, one that turned out to be a very positive experience. A very small subset of the available performance of Delphi is used here.

1.3.2. All programs have data in them as a default and will run, generating useful (we hope) data with but a few mouse clicks. For example, when the biasing program, BiasNPN is run, a schematic diagram appears. Move the mouse cursor to the "Calculate" button and (left) click. This produces a readout of the currents and voltages within the circuit.

1.3.3. Input parameters, shown initially as default values, are all changed merely by moving the mouse to the box with the input data. Again using BiasNPN as an example, move the mouse cursor over the 510 Ohm emitter resistor box and double click. This causes the entire "510" entry to be highlighted. It is then replaced by typing a new value. The results are updated by again clicking "Calculate." As an alternative to direct editing, the emitter resistor in BiasNPN can be increased or decreased by 5% by clicking on small boxes next to the value window.

1.3.4. The window containing the operating program can be minimized without disturbing any of the data in it. The buttons at the upper right of the window are used, just as with any windows program. It is often useful to have more than one of the LADPAC programs running at a time.

1.3.5. Many of the programs use a sequence of commands. These are numbered and should be executed in order. If you wish to try something different, go back and change an input parameter. Then click on all of the buttons from the change forward.

1.3.6. Most of the programs use more than one window. The other windows are activated by clicking on an indicated button in the active window. For example, clicking on the "About Bias" in the upper left part of Bias-NPN08 will cause a window to be shown with program copyrights and the like. A button in that window causes a return to the main window.

1.3.7. Many programs allow a file to be saved for further analysis. This is done by clicking on File in the upper left corner of the display window, a characteristic of virtually all Windows programs. Once into the File menu, click on the SAVE-AS button. This will bring up a dialog box showing a listing of files of the type related to the program. You can use an open-ended format for naming files. (If you save files in the older DOS format [8max.3max characters], they can be analyzed with G87.exe, a DOS program distributed with "**Introduction to Radio Frequency Design**," 1994, ARRL.)

An additional “save” format is present in some of the programs. This is a button within the main program that allows a file to be save as a default file named “Startfile.cir”. This is the often changed file that is the starting point for the **General Purpose Ladder Analysis** program, GPLA08. GPLA is a central cornerstone for all of Ladpac and is often used. It is very handy to design a filter, save it as the default, and then start GPLA for an immediate and quick analysis.

1.3.8. Files can be opened from the File menu in those programs where files are read. Opening is much like the SAVE-AS operation described.

1.3.9. When a program is finished, it can be closed with the usual "X" in the upper right of the window, by clicking on "exit" in the File menu, or by clicking in an Exit button within the program.

1.3.10. We urge you to open any of the programs and merely play with them to gain intuition about their operation, and more important, about circuit behavior. Use the programs interactively by designing a filter in one program, then examine the result in the LadBuild editor, and finally analyze the results in the General Purpose Ladder Analysis program.

1.3.11. The units used in LADPAC are Ohms, picofarad, nanohenry, and MHz. Crystal filters use Hz for frequency, but always referenced to a crystal nominal frequency in MHz.

1.3.12. It may be useful to print these instructions so you can read them while running the programs.

1.3.13. The programs within LADPAC include no specific printing routines. However, obtaining a printout is relatively easy within the Microsoft Windows environment. If you wish to make a record of any screen shown at a point in time, press the print-screen command. (Printers should be off at that time.) This causes the entire screen image to be copied to the Windows clipboard. Then open the standard Windows "Paint" program. When that is open, paste into it. (Click Edit, followed by clicking on Paste, or key in Ctrl-V.) This view can then be saved, as it stands. Alternatively, the PAINT program can be used to crop the display. The result can be saved as a .GIF or .JPG file and sent over the internet, or can be printed. (Figures in this manual were done this way.)

1.3.14. Above all else, remember that these programs are all simple examples using simple models. The results represent the limitations of the circuit models which may be nothing more than an approximation to physical reality; that is the way of all science. These programs are intended to supplement the data within the book "**Experimental Methods in Radio Frequency Design**." They are not intended to be used alone without the additional information provided by the text.

1.3.15. These programs are a part of the text and should not be distributed.

2. Lowhi08 Low Pass and High Pass Filters.

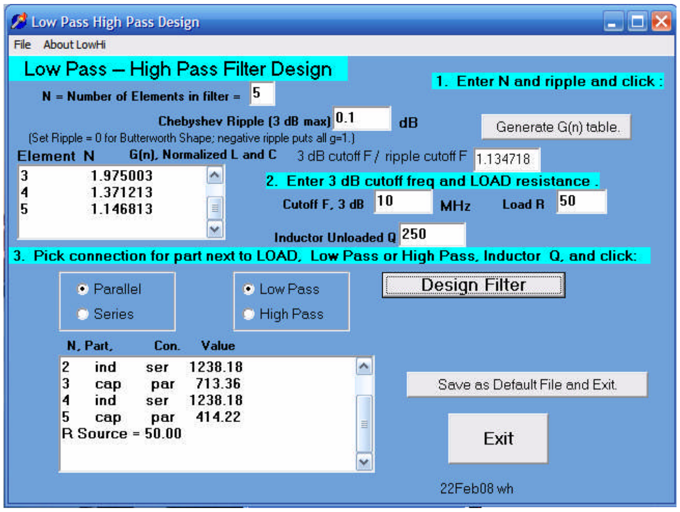

Fig 1.

This program is used to generate LC low pass and high pass filters. The program is started by double clicking it from Windows Explorer with focus on the file containing the Ladpac programs and following the numbered instructions. You pick the number of elements in the filter, and the allowed passband ripple. Default values of N=5 and 0.1 dB are built into the program. Clicking the first button, labeled "Generate G(n) table," then generates the normalized data for that Chebyshev filter. A Butterworth filter will be designed if a ripple of 0 is entered. Although obscure, a negative value can also be entered. This forces all G(n) values (normalized component values)to be 1. This is useful for the design of things like half-wave filters.

This program uses the 3 dB cutoff to specify a filter. Some others are based upon the ripple cutoff. A value is calculated and displayed in the program showing the ratio of the 3 dB cutoff to the ripple cutoff frequencies. This value will allow the user to change the design basis, if desired. Consider an example: The figure above shows that a 0.1 dB ripple Chebyshev filter will have a ripple cutoff that is below the 3 dB cutoff by a factor of 1.1347. If a low pass with a 10 MHz ripple cutoff is needed, a N=5 filter with a 3 dB cutoff of 11.347 MHz should be designed.

Specific data is entered about the filter you wish to design. This includes the 3 dB cutoff frequency and the terminating impedance at the load end of the filter. A box is also available for unloaded inductor Q. Although this has no impact on the design, it is useful for later analysis. You will then select between a low and a high pass response, and decide if the first component at the load will be parallel or series connected. Then click on "Design Filter" and the resulting component values are shown in the "memo box." If you had elected to design an even ordered Chebyshev filter, the termination resistance at the source will also be calculated and shown.

A design can be saved to disc for later analysis upon completion of the design. See the comments above (Section 1.3.7) in the General Comments area for details about saving programs.

This program is restricted to filters up to the 30th order, much more than most of us need. The filters designed here are a subset of a greater possible collection of Chebyshev and Butterworth filters with a wide variety of terminations. EMRFD should be consulted for design of other filter shapes.

3. GPLA08: General Purpose Ladder Analysis.

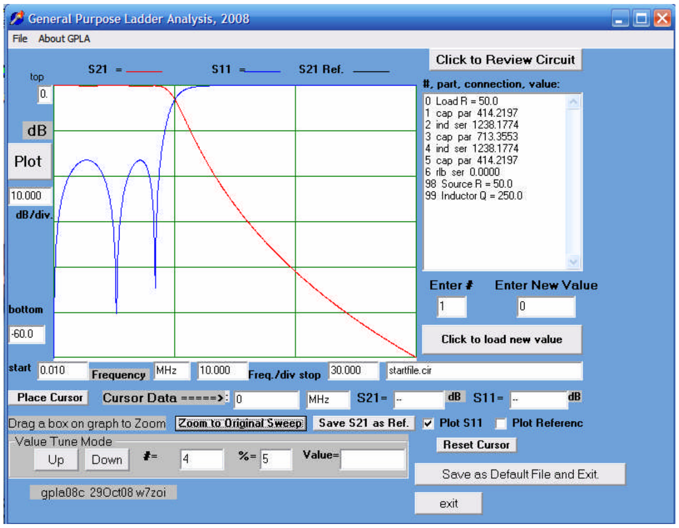

Fig 2.

GPLA is the central cornerstone program for the LADPAC collection. It allows the user to calculate the gain and input impedance match magnitude for a wide variety of filters that fit into the "ladder" category. The elements that can be placed in the ladder include capacitor, inductor, resistor, crystal, and wire. The "wire" is nothing more than a default connection that extends to a termination. Also included are duplicate capacitors, inductors, and resistors, components that are linked or "ganged" to other parts in a filter.

We assume that the user will be following along with this description with his or her version of Ladpac08 as this discussion is read.

The default filter circuit is whatever happened to be included in the Startfile, described earlier. A five element low pass filter with 0.1 dB ripple was designed and saved in the default Startfile. We will use this data as a "guided tour," a means of illustrating the program operation.

Begin by calling the program by double clicking GPLA08 from Windows Explorer to show a screen filled with buttons. Start by clicking on the "Plot" button on the left side of the window. The response for the default circuit immediately appears, showing the gain (S21) in red and the return loss at the source (S11) in blue. If you do not wish to show the return loss, un-click the checked S11 box near the bottom right of the screen. We have chosen to label the parameters with the scattering parameters rather than some of the older terms that we sometimes see in classic literature. Scattering parameters are the things that we usually measure. GPLA only shows magnitudes of S21 and S11.

The default circuit from lowhi08 is a low pass with 10 MHz cutoff. The sweep goes from 0 to 30 MHz. You can see the response in greater detail by changing the sweep parameters below the plot to show the range from 0 to 10 MHz with 1 MHz per division graphed. This is done by editing the values in the "Frequency" row, below the plot. We change the vertical parameters at the left to plot S21 to -1 dB with 0.2 dB steps. If a positive number is entered to represent the screen bottom, it will be converted to a negative one. The top is always 0 dB. After the editing, click "Plot" again to see the update.   This is shown in the following figure.

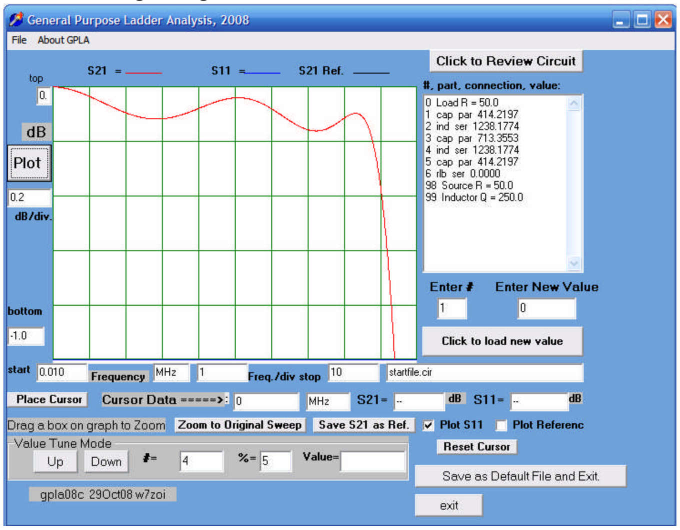

Fig 3. The response of our low pass filter over a 10 MHz by 1 dB range.   The downward slope with increasing frequency results from the finite inductor Q.

Zoom capabilities are built into GPLA.   While viewing the response, move the mouse cursor onto the graph. Assume you wish to view the response between 7 and 9 MHz to the -0.5 dB point. Move the mouse to 7 MHz and press the left mouse button. Then drag the mouse to the right, keeping the button pressed, until the mouse is at 9 MHz and -0.5 dB. Release the left mouse button to create a new sweep. It would also have worked if you had started the new sweep rectangle at 9 MHz with an end at 7 MHz. You can continue to zoom. When there is no more insight to be extracted, click on the "zoom to original sweep" button below the graph. Because the top of the screen is always 0 dB, the rectangle used during zoom is always close to the top, even when the mouse does not start there, making zooming an easy operation. Try to stay inside the graph border when marking a region for zooming.

To place a cursor on the graph, click on the "Place Cursor" button below the response. Then put the mouse cursor inside the graph and click at the frequency of interest. A vertical cursor line (dashed red) is then drawn on the graph, and the frequency and gain response are shown below the graph on the cursor line. Cursor frequency is maintained with zoom or value change actions that may follow. The cursor can be replaced or moved with another "Place Cursor" action, or eliminated with the "Reset Cursor" button.

An alternative way to place a cursor is to merely edit the frequency in the Cursor Data box. Type "7.5" in this box and then click on the PLOT button. This generates a cursor exactly at 7.5 MHz.

We can examine the components in our filter by clicking on the "Click to Review Circuit" box. This then shows the circuit elements including the terminations and the global inductor Q. Any of these values may be changed. If, for example, you wish to change the Q of the inductors in the low pass filter, enter 99 for N and a new value in the "Enter New Value" box.   Try a very large value of perhaps 10,000. Then, clicking the "Click to load new value" button will change the circuit and reprint the circuit description. The loss now appears to disappear at the peaks of the Chebyshev response.

Two independent 1238 nH inductors are shown as parts 2 and 4. It is often useful to lock the values together, making part 4 a duplicate of 2. Then, if part 2 tuned or changed, part 4 will also change. The change is performed in a different program, *ladbuild08*. We will deal with this program later.

We have made several changes that could complicate our description. Let's fix that by clicking on the Exit in the File menu. Restart the program. The sweep is back to the previous 0 to 30 by 10 MHz steps. Let's now save this circuit by calling the File | Save-As menu, using the file name gpladefault.cir, and click on the "Save" button. The circuit is now on the disc for later use by GPLA. It is also available for viewing in the "Ladbuild08" program, discussed later.

You can now open the File | Open menu and open the file. This causes the existing plot to be erased. When a file is opened, the name of that file appears in a data box in the "Frequency" row. If the file description is so long that it cannot all be seen in the box, place a cursor within the box with the mouse. Then move the cursor with keyboard arrow keys or the “end” key.

GPLA includes a "tune" capability, controlled in a box at the bottom of the GPLA window. The default has part #4 selected for tuning in 5% steps. Edit this to tune part #3 and then click on the "Up" button, causing the capacitor value of part number 3 to go up by 5%. This causes a new sweep to be generated. If circuit data was present in the review window, that data would disappear with the new sweep. But the new value of part 3 is now shown in the Tune box.

Often it is useful when tuning to know what the gain response had been at an earlier time. Click on the box labeled "Save S21 as ref." This is located just above the Value Tune Mode box. This click causes whatever gain data is present in the plot to be stored in computer memory as a reference. Near this "save" button is another button labeled "plot reference." Check this box and then click on the Plot button. Note that the reference is now plotted in black while the newly tuned response is shown in red. (Click the Tune/Up button to generate a different plot.) S11 data is not saved with a reference.

If you zoom in after having started a sweep with a comparison plot, the reference disappears. Activating the "Plot Reference" button again still generates the original plot. The reference values are not altered by a zoom operation, so the reference only makes sense with the frequency span used when the reference was saved.

The reference display is very useful for comparing two different plots. We already have our 5 element filter saved as “gpladefault.cir”. Return for the moment to Lowhi08 and design a low pass filter with an identical cutoff frequency of 10 MHz, but with 9 elements. Save this with the File|Save-As function with the name “lpf9.cir”.

Return to GPLA. Use the File|Open function to load the original file, *gpladefault*. The response is plotted. Click on the “Save S21 as ref” button. Now return to the File|Open function and load the newer response *lpf9*. Click on the “plot reference” button and both responses are now shown. The 9 element response has a much more dramatic cutoff response while having virtually the same response within the passband.

There is a potential trap when comparing filters. The user must take care to be sure that the same frequency span is used in both plots. Otherwise, the comparison will not make a lot of sense.

GPLA will also plot the response of crystal filters. An included file is named X5.cir. Load this into GPLA and click the Plot button. You may have to "un-click" the "show reference" plot button, for early data may still be present. The frequency has now shifted to units of Hz instead of MHz. This happens whenever a crystal is present in the collection of components. The frequencies shown in plots and tables are referenced against the nominal crystal series resonant frequency which shows up in the data in the circuit review box.

Reviewing the circuit of a crystal filter shows a value for each crystal. These are all 0 for this example. These values are frequency offsets for each crystal. While this would, ideally, be the offset with respect to the ideal series resonance, it is sufficient in practice to insert a value that is an offset with respect to an average or a maximum. The circuit we just loaded, *X5.cir*, is for a filter with 5 crystals. The offset value for each crystal is 0, the result generated by XLAD, the crystal filter design program (discussed later.) The designer can now insert practical measured values. Later, we will discuss the tuning of such filters when offsets exist.

The "global" crystal parameters are listed for the parts in the crystal filter and can be changed with editing. For example, try changing the crystal Q to a much lower value of 20000.

Some details: GPLA was written using a scheme known as "the ladder method," with details presented in "Introduction to Radio Frequency Design," ARRL, 1994. This method is generally quite fast compared to matrix methods. (This made a ***<u>big</u>*** difference when our computers were slower, but is of little consequence now.) The ladder method is not as powerful as other techniques, and is not suitable for amplifier analysis. But the method is exact within the constraints of our ideal L, C, and R component models. We have chosen to assign a uniform unloaded Q to all inductors within a circuit. Capacitors are assumed to be lossless. All crystals within a circuit are assumed to have an identical unloaded Q, and identical parallel C values. These assumptions seem to work well for filters in the HF spectrum built from traditional leaded components. Surface mount capacitors are sometimes quite lossy (without regard for temperature coefficient) and that loss may compromise filter modeling.

4. DTC08, TTC08, and QTC08: Double, Triple, and Quad Tuned Circuits.

Three programs are more than cousins; they are almost identical to each other. DTC08 and TTC08 serve to design double and triple tuned circuits. Each program will design two filters at the same time. This is illustrated in Fig 4. The third program, qtc08, designs a quad tuned circuit.

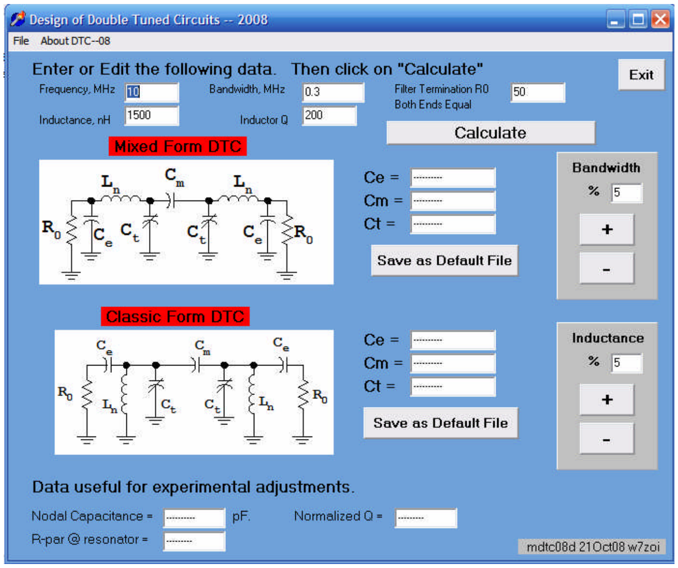

Fig 4. Two forms of the double tuned circuit are designed with one program, dtc08.

Like other programs within LADPAC-2008, these begin with built-in defaults. In this DTC example, it's a 10 MHz bandpass filter with a 0.3 MHz bandwidth. It uses 1500 nH inductors with unloaded Q of 200. These values can all be changed with editing.

In both dtc08 and ttc08, the coupling and loading coefficients are set to generate Butterworth filters.

The unusual character of the 2008 programs when compared with Ladpac-2002 is the feature that designs two programs at the same time. One is what we are terming a "mixed form" bandpass filter. The ends of the filter are loaded by terminations as if we were building a filter with series tuned circuits. Such circuits tend to degrade into a low pass filter in the stopband. But the coupling between resonators occurs as if the end resonators were parallel tuned circuits, a topology that behaves like a high pass filter within the stopband. We find that the mixed form is the preferred topology if filter shape symmetry is important. Such an application might be a spectrum analyzer filter. The topology is a handy one even when symmetry is not a consideration.

The programs are easy to use by merely following the numbered instructions. We begin by describing the filter in terms of center frequency, bandwidth, inductor characteristics, and termination. After the parameters are in place, click on Calculate . The program now displays all L and C values parameters for the two filter forms.

One of the two designs may be just what is needed. That filter may now be saved for later analysis. There are several ways to do this. The traditional one is to use the File|Save-As function. This *almost* works like you would expect. Assume we perform this operation and pick a filter name of "dtc10" indicating the 10 MHz center frequency. (We are using the default data.) What happens is that two files are created. One has the phrase "_classic" appended to the name while the other has "_mixed" attached. (These forms cannot be used with the older versions of Windows or with the old DOS G87 variant of GPLA.)

The other way to save a file is to just click on the boxes that save a design to the default file "Startfile." Only the last one clicked is saved if you click both boxes.

Some extra data is presented in the lower left part of the screen. These are the nodal capacitance, the resistance load appearing across a parallel resonator at the ends (classic form filter), and the normalized Q. Normalized Q is useful in predicting filter passband insertion loss. The nodal capacitance is just the C required to resonate the chosen inductor at the filter center. The parallel R is useful when matching to the ends of a modification of a classic form filter that uses link coupling. Reference: http://w7zoi.net/Transformer_Coupled_LC_Bandpass_Filters.pdf

There is a degree of flexibility in these designs that is not always apparent. The designer may exercise considerable freedom when picking inductor values. The limiting criterion is that the terminating resistance picked (often 50 Ohms) for the design must be less than the terminating resistance listed in the lower left corner. The filter bandwidth and the inductor value may both be adjusted to realize standard capacitor values. This is accommodated with the corresponding buttons on the right side of the program screens. For example, the default double tuned circuit uses 1500 nH and has a bandwidth of 300 kHz. Increasing L and B to 1914 nH and 331 kHz allows the mixed form filter to be built with standard components of 1500 pF and 3.3 pF for the end and coupling capacitors, respectively.

The Quad tuned circuit program, qtc08.exe, shown in Fig 5, does not include the classic design form. The designer can use the data within EMRFD to generate his own classic form, if desired.

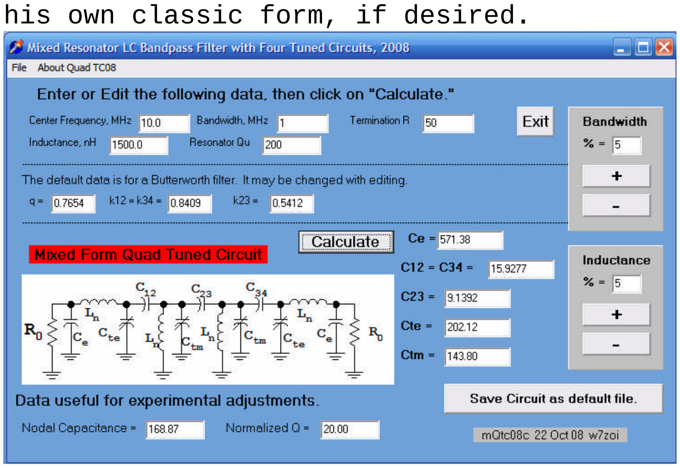

Fig 5. Quad Tuned Circuit. This circuit is especially useful for filters that must have good symmetry.

It is often useful for a filter to have different terminations at the two ends. This is easily realized with the various programs available in LADPAC. Assume we need a 10.1 MHz filter with a 50 Ohm input termination, but an output termination of 3000 Ohms, perhaps the input of an integrated circuit. We first design the filter (Classic form DTC) for 50 Ohms. We use an inductance of 2300 nH with a bandwidth of 0.36 MHz to generate a filter with a coupling capacitor of 2.7 pF. The capacitor to the 50 Ohm termination is 27 pF. The tuning capacitor across the inductor is 80 pF. (Jot this schematic down on a scrap of paper as it is designed.) We now return and change the termination to 3000 Ohms. Re-click Calculate and see a new 4.5 pF capacitor that couples to the 3K load. (Use 4.7 pF when building the filter.) This variation is tuned with 102 pF capacitors. The final circuit is shown in Fig 6.

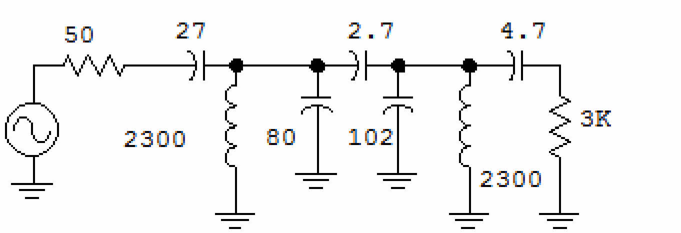

Fig 6. A Double Tuned Circuit with different terminations.

The double tuned circuit with differing terminations can be "built" in the ladder builder editor program discussed later. It can then be analyzed with GPLA.

5. Ladbuild08: A Ladder Editing Program.

Ladbuild08 is an editor that allows modification of existing circuits or construction of new ones for use by GPLA. As the name suggests, the program purpose is to *build ladders*.

The original version of Ladbuild is included on the distribution CD. This only deals with 30 components, but the program fits in on a smaller computer screen.

Once a circuit is drawn in LadBuild08, it can be saved to disc for analysis in GPLA. The components allowed are capacitors, inductors, resistors, duplicate capacitors, duplicate inductors, duplicate resistors, crystals, a return loss bridge, and wires. All crystals within a circuit must have an identical nominal frequency, unloaded Q, and parallel capacitance. However, each crystal frequency can be offset in frequency from the specified series resonance with the offset, in Hz, being the "value" related to the part. These crystal constraints are generally reasonable for folks designing and building crystal filters. The ladbuild screen is shown in Fig 7.

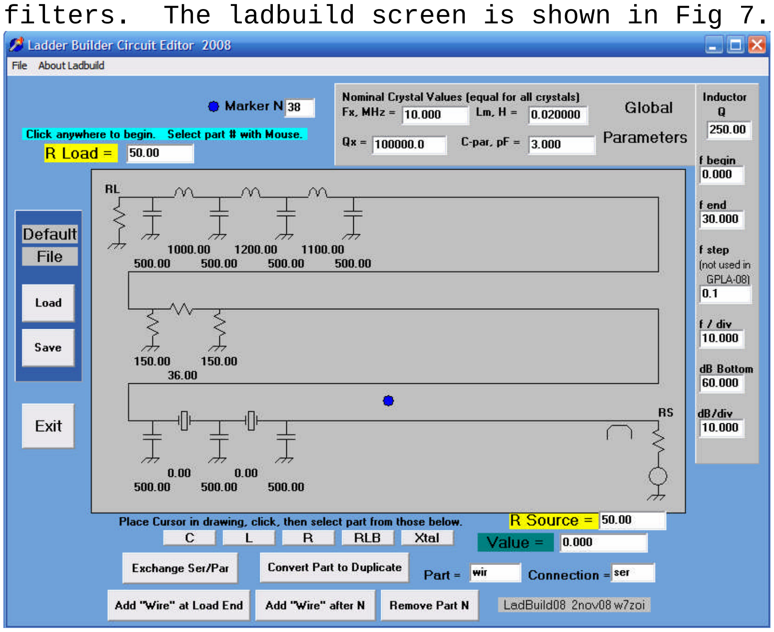

Fig 7. Ladbuild08. The circuit shown is merely a built-in starting point. Use it for learning. The blue dot indicates that a component is selected. The corresponding number is at the top of the screen, here #38. The value of this "wire" is 0.000.

A blank screen appears when LadBuild08 starts. However, clicking anywhere within the drawing area with the mouse causes a default (essentially nonsense) circuit to appear. Clicking within the circuit causes a cursor dot to appear over the corresponding part. The number of that part is shown at the top of the drawing while the component value is shown at the bottom. The component values are also presented within the schematic below the component symbols.

A selected component can be altered by keying in a new value at the drawing bottom. It can also be changed to a completely new component type by clicking on the desired new part (C, L, R, RLB, or Xtl.) Values are "fixed" by clicking on another part with the mouse. Components can be toggled from a series or parallel placement with a button near the screen bottom. A "wire" can be inserted into the circuit at either the load end or just after a marked part. A part can be deleted from the circuit, or changed to become a duplicate of another part.

Files are loaded and/or saved with the same methods used with the other LADPAC programs through the File menu. A sequence is shown in Figures 8, 9, and 10 where we deal with the problem of an unequally terminated double tuned circuit designed in Fig 6. The default file is loaded with the buttons to the left of the drawing area.

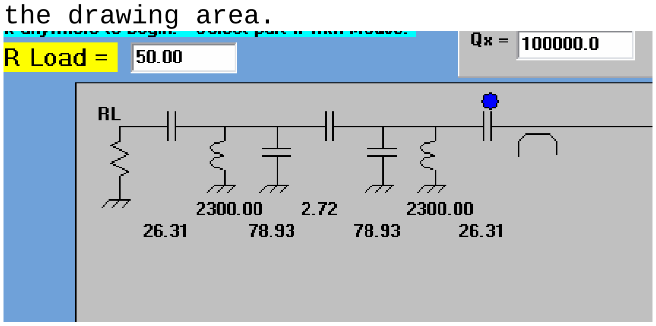

Fig 8. This is what we see when we load Startfile into the editor after having generated the 50 Ohm variation of the Fig 6 circuit.

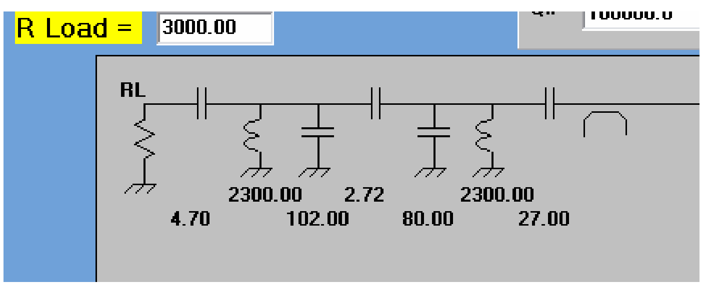

Fig 9. The editor is then used to change component values. Notice that the load resistance, R Load, is at the apparent beginning of the filter rather than at the end. After the changes, Startfile was saved with the Save button to the left of the schematic.

The edited and saved file can now be used for analysis. Starting GPLA and clicking on Plot produces the following result:

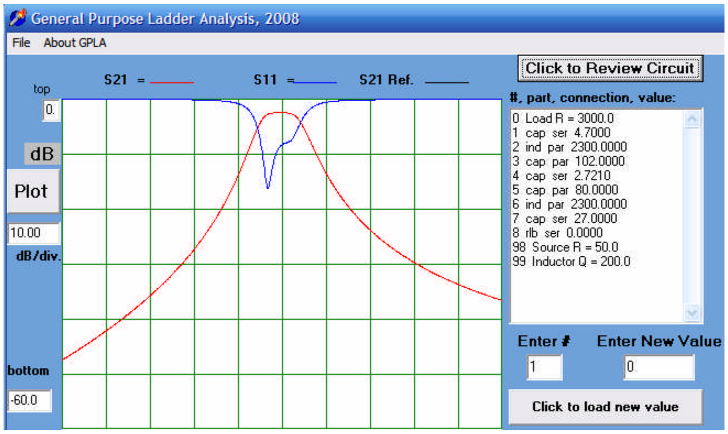

Fig 10. GPLA Analysis of the double tuned circuit built in Ladbuild08. Note in the listing to the right of the plot that the load R is 3000 while the source R is 50.

The duplicate elements are especially useful, for they allow values to track each other. It is not necessary that a duplicated part have the same connection as the parent. But, be careful to not declare a duplicate of another duplicate when using this feature. The "value" for a duplicate is the number of the part to be duplicated. See the crystal filter case study for an application.

Several global parameters are present on the screen, grouped around the edges of the schematic. Ladbuild08 includes some defaults including crystal parameters. If the circuit being edited contains no crystals, no crystal data will be saved. Don't be concerted about screen data that is present. There is one parameter in the frequency steps that is only used in G87, which is an older DOS program.

Note that the numbering of parts we use throughout the series of programs begins with the load end of the filter. This was a natural consequence of the ladder method (IRFD, p 51) and is kept as a reminder about the formality of the process.

Like all of the other programs within LadPac, Ladbuild08 includes a default set of values, which is mostly just for illustration. Hopefully, it is useful when learning how to use the program. If the user wants to start fresh, a blank screen is created with File|New. 50 Ohm terminations are used to begin the NEW screen.

This displayed screen for Ladbuild08 is larger than most of the other programs in Ladpac. The 2002 version of the program, which will only accommodate 30 components, is included for users with a computer display with less resolution. Operation is identical except that the default parameters are in separate windows, so don't forget them.

6. XLAD08: Crystal Ladder bandpass filter design.

This program is for the design of crystal filters of the lower sideband ladder type as described in EMRFD, chapter 3. Filters used in several places in the book were designed with this program. The program is structured for use with tables of normalized coupling and loading, or k and q. A good source for this data is the classic text by Zverev, "**Handbook of Filter Synthesis**," Wiley, 1967. Further, a subroutine is included for calculation of k and q parameters for Butterworth and Chebyshev filter shapes. This latest 2008 version includes the calculation of Min-Loss (Cohn) filter k and q. Figure 11 shows the screen presented by the program.

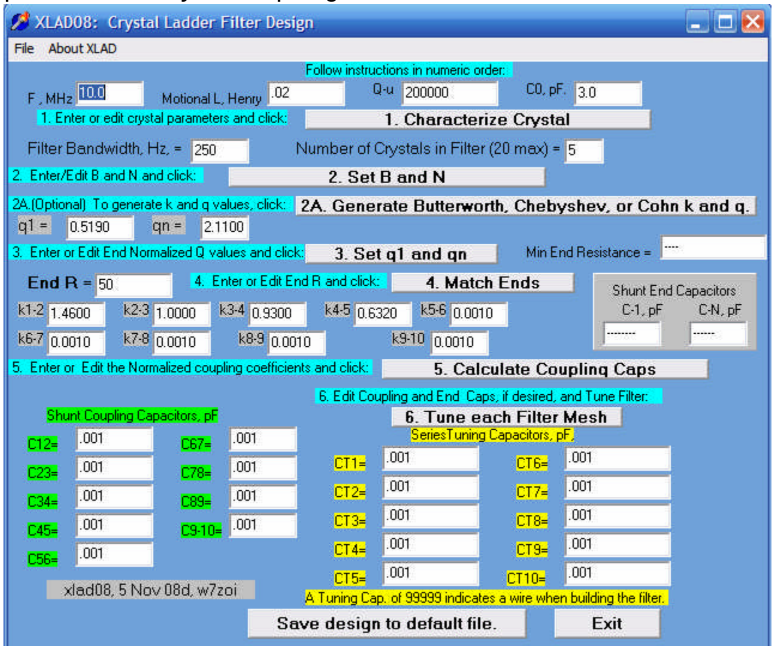

Fig 11, the screen presented by XLAD. Program operation begins with step 1 and continues in sequence.

One method useful for measuring crystal motional inductance and capacitance uses a frequency counter and an oscillator with a switched capacitor in series with the crystal. Parallel capacitance is measured with one of several methods operating at low frequency, well away from crystal resonance. The measurements are detailed in EMRFD Chapter 7.

Some assumptions are made for the filter design. First, we assume that all crystals are on the same, or close to the same frequency. A later program, Meshtune08, will let us account for departures. Second, we assume that all crystals have identical unloaded Q and parallel capacitance. This is a reasonable assumption when the crystals are from the same manufacturing process. This represents the example when a batch of crystals of one type from one manufacturer is purchased from a catalog.

Filter design is a six or seven step process. The first step is to enter global crystal data at the top of the window, followed by clicking on button 1. Next, a filter bandwidth and order (number of crystals in filter) is picked and entered into the program followed by a click of button 2.

The next step, 2A, will not be used if you plan to enter k and q data from a table. However, if a Butterworth or Chebyshev filter is to be designed, click on button 2A. This activates a different screen where the k and q data will be generated. Normalized k and q data for up to 20 element filters can be generated here. However, the basic program is confined to 10th order filters. Don't use step 2A if you already have k and q data in place that you plan on using, for activating 2A erases that k data.

The default k and q data built into the program starting screen when it starts is for a Gaussian-to-6 dB filter shape with 5 crystals. This is a filter we have built and used for spectrum analysis purposes as well as in CW receivers. This shape lacks the component symmetry of Butterworth and Chebyshev filters, but offers excellent performance in these applications. *These k and q only work for N=5 and you will need other data for other filters.* This means that if you first do a N=5 design and then decide to try a N=6 filter design, you will have to enter completely new k and q values.

Step 3 enters normalized end section q data, followed by a click of button 3. This produces a readout of the minimum terminating resistance that can be used for the filter, information needed in the next step.

The designer next picks a terminating resistance to enter, followed by clicking button 4. If you pick a value that is too low, an error will be issued. When this step is successful, shunt end capacitors will be calculated and displayed in the box at the mid right part of the window. The two capacitors will be equal for equally terminated Butterworth and Chebyshev filters, but may not be for other shapes.

The next step enters normalized coupling data (k) followed by clicking on button 5. This data will already be in place if you have performed Chebyshev or Butterworth calculations via step 2A. Clicking on button 5 causes the shunt coupling capacitors between crystals to be calculated and displayed.

The final step is normally no more than a click of button 6 to tune the filter. This action calculates the tuning capacitors that will be in series with each crystal. These capacitors are required to force each mesh (i.e., loop) in the filter to have the same resonant frequency when isolated from the other meshes. Tuning with series capacitors is necessary because the coupling capacitors detune each mesh.

The program will evaluate the resonant frequency of each mesh before inserting series tuning capacitors. One mesh will be the highest in frequency and will not have to be "raised" with a tuning capacitor. The program merely inserts a bogus value of 99999 pF for the related tuning capacitor, indicating that it is merely replaced by a wire when the filter is built. The user can modify the circuit with Ladbuild to eliminate these bogus capacitors, but that is not necessary for GPLA analysis.

A subtle, but highly practical feature is built into the program that may not be apparent. ***The tuning capacitors are calculated by reading the values that are presently in the END capacitor and coupling capacitor edit windows.*** If you wish to use standard value capacitors that are close to those calculated by the program, you can edit the practical values into the edit windows before clicking on the tuning button 6. You may even wish to change the tuned values to close practical substitutes before saving. (This freedom is not found in other programs that we have found.)

The plots reveal another subtle, yet profound truth of crystal ladder filters: The filter frequency passband is always above the series resonant frequency of the nominal crystal. The exact filter edge is not related to the filter parameters with a simple relationship. The designer can change the position by changing tuning capacitors throughout the filter. This makes it **very** difficult to order a batch of crystals with the idea of exactly hitting a design frequency, which would be needed to build a filter for an existing piece of commercial equipment. Instead, the usual practice has the designer/builder ordering available crystals, measuring them, designing a filter, and even building it, before exactly establishing the filter frequency. The builder of homebrew equipment usually builds filters and then designs and constructs the rest of the project.

While XLAD is restricted to filters with at most 10 crystals, it is possible to design filters up to order 20. An example with 14 crystals has been built. The design process is summarized at the end of this manual.

7. Meshtune08: A program to adjust the frequency of individual meshes in a crystal ladder filter.

Linear circuits are usually analyzed with an emphasis on nodes or meshes. A node is a point in a circuit where components join. A mesh is a connection of several components in a loop. In a ladder filter, proper tuning results when each mesh is resonant at the same frequency. Xlad08 achieves this with internal calculations for the case when all crystals are identical. Meshtune08 lets us deal with individual meshes. This program will not be needed during the design of simple filters. The program screen is shown in Fig 12.

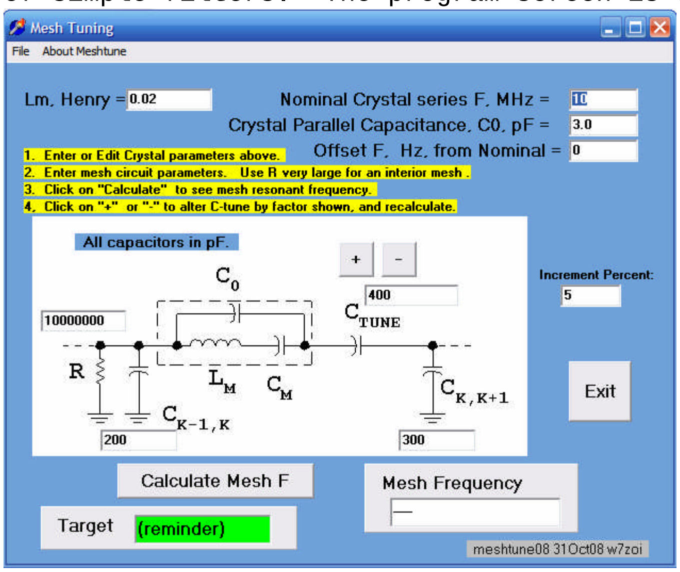

Fig 12, Meshtune08 screen.

Meshtune08 operation begins by entering the crystal data. Note that this data set includes an offset with respect to the nominal. It is really the user's choice as to how nominal is defined. The normal practice when building a filter is to accumulate a collection of nominally identical crystals. They are all marked with a sequential number and then measured for oscillation frequency in a convenient oscillator. Most folks (myself included) find it useful to enter the data into a spread sheet program. After all data is in the program, a "sort" is performed to put the data points in order according to frequency. It is then easy to select closely matched crystals for a filter. This may not always be practical and we may sometimes want to use crystals that are not close to each other in frequency. This is the reason for this program.

Begin by entering data for one of the meshes in the filter. If this is an end mesh, the resistance value for the end termination is entered. If it is an interior mesh, a very large R value (usually 1 meg or higher) is entered. If you are evaluating a mesh with a resistor on the right hand side in your drawing, do a mental flip. That is, place the terminating resistor and its capacitor together on the left in *Meshtune* with the un-terminated capacitor on the right.

Once the crystal data is in place, click on the “Calculate Mesh F” button. This will cause all data (crystal parameters and circuit R and C) to be read, will calculate the mesh frequency and will display it in the box. For example, with the default parameters shown, the mesh is resonant 664 Hz above nominal. The user can enter new data in the edit boxes and calculate again. Alternatively, the + or – buttons can be pressed to increment the series capacitor value.

As an exercise, use the default data and examine tuning capacitors ranging from a short circuit (use 99999 pF) down to 10 pF. This mesh frequency will vary from 515 Hz above nominal to 5176 Hz above nominal. Now, assume that the mesh is to be tuned to a frequency 1000 Hz above nominal. Start by entering 1000 pF for the tuning C, which, with the Calculate button, will show F=575 Hz. Now, enter 1000 in the “Target” box in the lower left of the window. This box is nothing more than a scratch pad, a place to note a value. Click on the “-“ box and note that the tuning C drops down, but the frequency increases to 578. Note that there is now a dotted box around the “-“ box. This indicates that it is ***active***. So, if you merely press the enter key, that active key will continue to be clicked in fast succession. It will probably be necessary to click the “+” key and to change the “Increment Percent” value before arriving at a tune C value of 121.4 pF.

We will illustrate the use of *Meshtune* with an example. Going into XLAD, we design a 4th order 0.5 dB Chebyshev bandpass filter with a bandwidth of 1000 Hz. A 5 MHz crystal is used with motional L=0.11 H, Qu=200000, C0=3 pF. The filter is terminated in 500 Ohms at each end. The end capacitors are 13.8 pF while the coupling capacitors are 54, 64, and 54 pF. The filter is tuned with 81 pF capacitors in the two end meshes with "wires" as the inner capacitors. This filter is in memory as c4a.cir. We will use the actual tuning C values for the initial analysis. The filter circuit is shown in Fig 13.

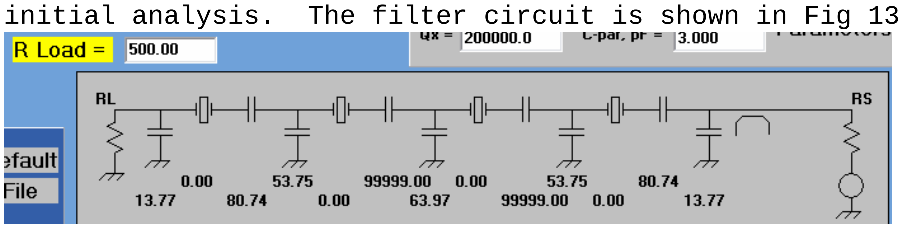

Fig 13.

Meshtune08 is now used to examine each mesh, one at a time. Because all crystals are on the same frequency, all meshes are on the same frequency, which comes out as 713 Hz above the nominal 5 MHz crystal series resonance. The actual series resonance of the crystals is not at 5.0000 MHz for these crystals. Rather, it's about 1.5 kHz below this. However, this difference is of minimal consequence. Only the relative spacing is significant.

The schematic in *Meshtune* shows the terminated side closest to the crystal with the tuning capacitor next to following coupling capacitor. However, the tuning capacitor and crystal can be exchanged with no consequence. Both networks have the same impedance.

First we test a rule of thumb that says a crystal frequency difference of 10% of a filter bandwidth will have little effect. Examining 10% errors tends to confirm this rule for this particular filter. Next we pick more severe errors of +200, -100, +100, and 0 for the 4 meshes, left to right in LadBuild. We again check resonance for all 4 meshes in *Meshtune* and get the expected mesh frequencies of 913, 613, 813, and 713 Hz. This filter is saved as c4b.cir and may be examined with GPLA. These errors, up to 20%, do compromise filter shape, although not so much as to make the filter dysfunctional.

Next we eliminate all tuning capacitors through the filter. The inner meshes had none anyway, so they do not change. The outer meshes drop in frequency. The severe crystal offsets of +200, -100, 100, and 0 are still in place. The four mesh frequencies are now, in sequence, 670, 613, 813, and 471 Hz in a filter now saved as c4c.cir. The shape distortion observed with GPLA is worse.

Adding series tuning to a mesh increases the resonant frequency of that mesh. So we pick the highest mesh as one to leave alone. This is mesh #3 at 813 Hz. The three remaining meshes are now entered into *Meshtune* and the series capacitors are adjusted until they are all at 813 Hz. This value is entered into the "target" edit box, which has no function other than serving as a useful notepad. The filter with tuning capacitors of 134, 92, none, and 56.5 pF is now saved as c4d.cir. This shape again looks usable.

We now use GPLA to compare the first and the last filter. We begin by loading c4a.cir and examine the display. We click the button to "Save S21 as Ref." We now load c4d.cir, click the button to cause the reference to be plotted, and click the Plot/Tune button. The two filter shapes are virtually identical, although the latest one is perhaps 100 Hz higher in frequency. Bandwidth is 1060 Hz. The two plots are in Fig 14. (We did modify the files as they were loaded. In each case, we changed the frequency range to be 0 to 1500 Hz and the gain range to be from -30 to 0 dB. This produced a better comparison. But be sure to do this with both files.)

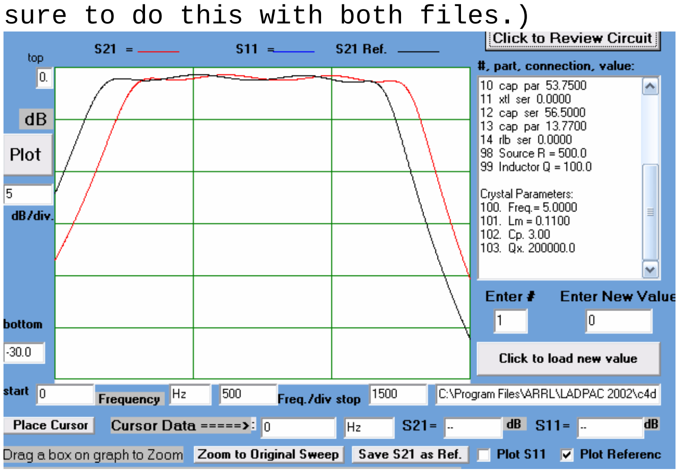

Fig 14. Two crystal filters. One uses crystals that are NOT well matched. See text for details. The two files are saved as c4a and c4d.

A final experiment is aimed at moving the filter to a higher frequency. We move the "target" from 813 up to 1200 Hz and now tune on all 4 meshes in the filter. The final result, saved as c4e.cir, has tuning capacitors of 35.5, 29.6, 46.4, and 25 pF. The shape and loss look much like the original filter, although the 3 dB bandwidth has now dropped to 940 Hz.

Clearly, the careful designer/builder has great freedom in building crystal filters. However, minor variations will alter the results, emphasizing the approximations that were used in the original filter design program, XLAD. As a filter is moved further from the lowest possible frequency, tuning becomes much more sensitive to small variations in values.

8. BiasNPN08: Biasing of an NPN transistor.

This simple program evaluates the bias for an NPN transistor operated from a single power supply. Biasnpn08 uses only resistors. The model is a very simple one of a current generator with constant beta. A diode with a constant voltage drop is included within the base. See EMRFD Chapter 2. The screen is in Fig 15.

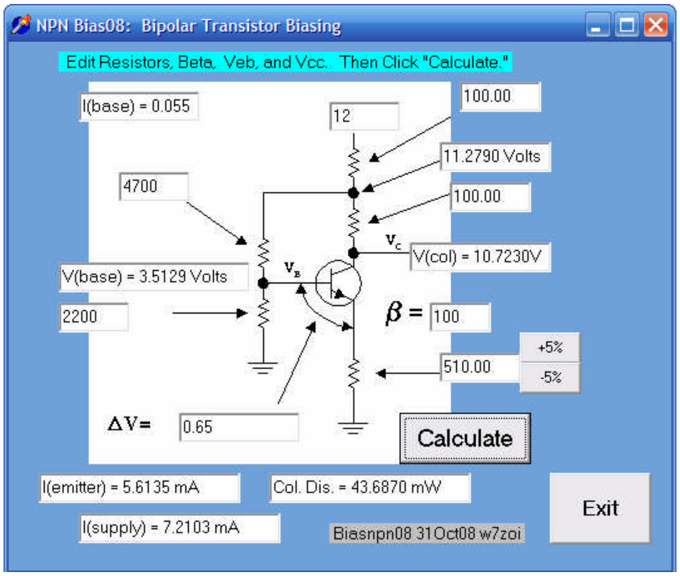

Fig 15. Bias Program.

Program operation is straightforward. When the program is called, merely click on "Calculate" to show the results with the default resistors shown. Results displayed include supply and emitter current, collector dissipation, base current, and voltages throughout the circuit. The transistor parameters, supply voltage, and any of 5 resistors can be changed in the circuit by editing the appropriate boxes. The equations used in this program are found in chapter 1 of Introduction to Radio Frequency Design.

We have added error trapping to the program, allowing 0 to be inserted for resistor values without causing program instability. The emitter resistor may be adjusted in 5% steps with mouse clicks.

9. FBA08: Design of single transistor feedback amplifier.

The reader will detect negative feedback as a central theme through much of EMRFD. A favorite radio frequency amplifier circuit uses a single transistor with negative feedback in two forms: emitter degeneration and parallel feedback from collector to base. This amplifier is one that offers good stability and bandwidth with well defined input and output impedances. This program, FBA08, analyzes these circuits. See Fig 16.

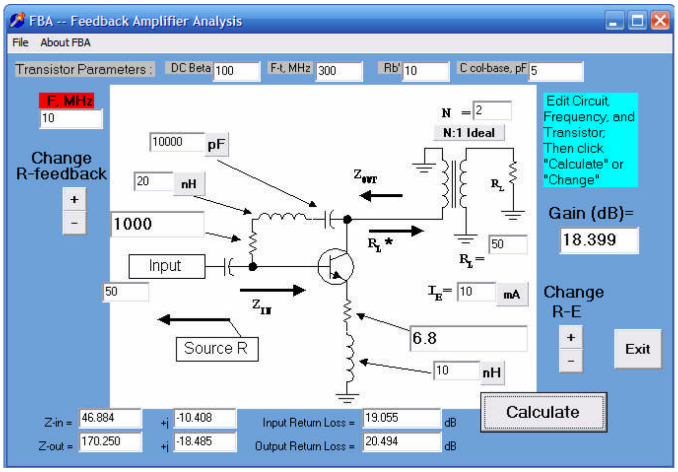

Fig 16. Feedback amplifiers are designed with a simple program.

The model used is the familiar "hybrid-pi" where the high frequency effects of reduced gain with increasing frequency are modeled with a capacitor between base and emitter. The model is extended to include a base spreading resistance and a collector-to-base capacitance (a.k.a., Miller capacitor.) External inductance in the emitter and in the feedback from collector to base are also included and can be especially significant at VHF.

This program performs a small signal analysis, showing the transducer power gain (magnitude only), input and output impedance with the terminations shown, along with related return losses. The feedback elements, emitter current, terminations, frequency, and transistor characteristics can all be altered. After changes are made, re-click on "Calculate" to show the results. The default transistor built into the program approximate the familiar 2N3904.

The schematic shows the familiar transformer in the output. The circuit is specified by the external load resistance and N, a turns ratio for the transformer. Negative feedback is obtained directly from the collector rather than from a tap on the transformer, for that scheme can compromise stability with really "hot" microwave transistors.

The output impedance calculated by the program is that seen looking from the transformer into the collector node. The values we would measure at the transformer secondary are transformed accordingly. The return loss would be the same so long as the transformer is ideal.

A pair of buttons are included for both the parallel feedback resistor and the emitter degeneration resistor. This allows the respective resistors to be incremented or decremented by 5% with a mouse click.

It is useful to open FBA and NPNBIAS in two adjacent windows. This allows one to design amplifiers that combine biasing with RF feedback, another special amplifier topology used through the book. Feedback amplifiers are discussed in Chapter 2 and applied to receivers and transmitters in Chapter 6.

Equations are given in the book for calculating feedback amplifier parameters. Those equations are scalar (no reactive components) and are simpler than those used in this program.

10. CASCADE08: Noise figure and third order IMD evaluation of a cascade.

This program is used in the design of systems to evaluate the effect of cascading several stages. Noise figure, net gain, and third order intermodulation intercepts are calculated for the cascade. System parameters that are evaluated include receiver two tone dynamic range, MDS, and receiver factor, all discussed in Chapter 6. The screen is shown in Fig 17.

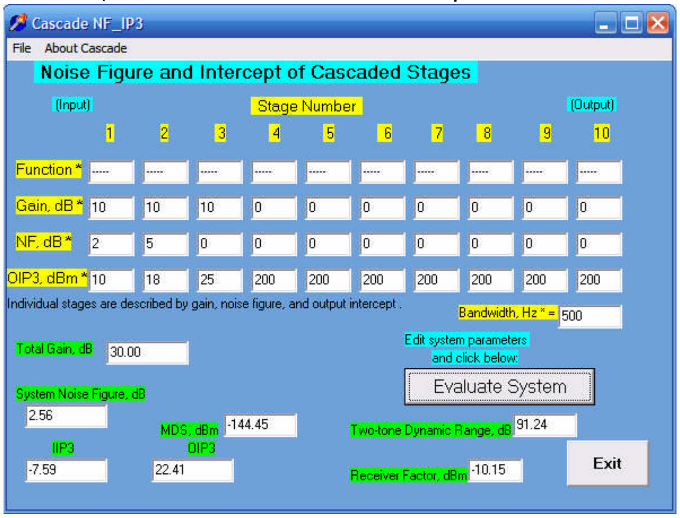

Fig 17. Cascade.

Several example receiver designs are presented in Chapter 6. We urge the user to use these with the program as a means of study. A file included on the CD is titled gp_rx_fe.cas, standing for "general purpose receiver front-end." This design is used with some receivers in the book.

Each stage in a cascade is described by entering its gain, noise figure, and output intercept. This data, for a single stage, occupies a column. A short text description of that stage is shown at the top. The name has no significance in program operation other than reminding the user of prior thoughts. Some stages, such as filters and pads, are assumed to be free of IMD. This is represented to the program with a extremely high OIP3 for that stage, usually +100 dBm.

The model used in CASCADE.EXE for IMD combination is coherent addition of distortion voltages. This assumes that distortion products from all stages add in phase. This represents a worst case. It is also consistent with the behavior we have measured in many receivers, power amplifier chains, and RF Instruments.

Files for various designs can be Opened and Saved through the File operation on the tool bar.

11. Zmat08: Impedance Matching

We frequently encounter the need to transform impedances. Although this can be a complicated general problem, it is straight forward for the special case of one real impedance transforming to another real one. Transformations between real impedances are treated with the program zmat08. The program window is seen in Fig 18.

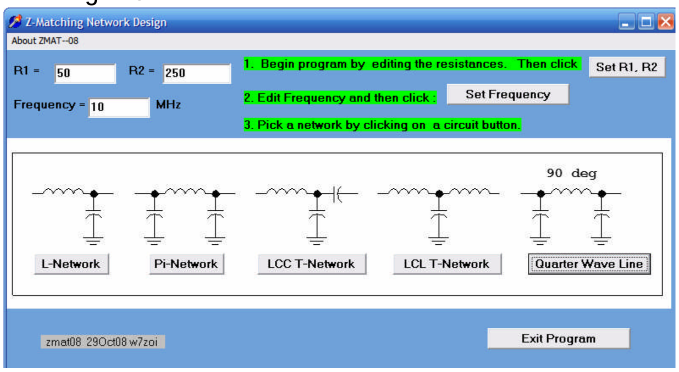

Fig 18. Impedance transformations with several popular networks.

Program operation is straight forward. The user begins by establishing the impedances and the frequency. The default values are edited and the appropriate boxes are checked with the mouse. The user then clicks on one of the five networks available in the program. If the LCC T-Network is picked, the screen shown in Fig 19 appears.

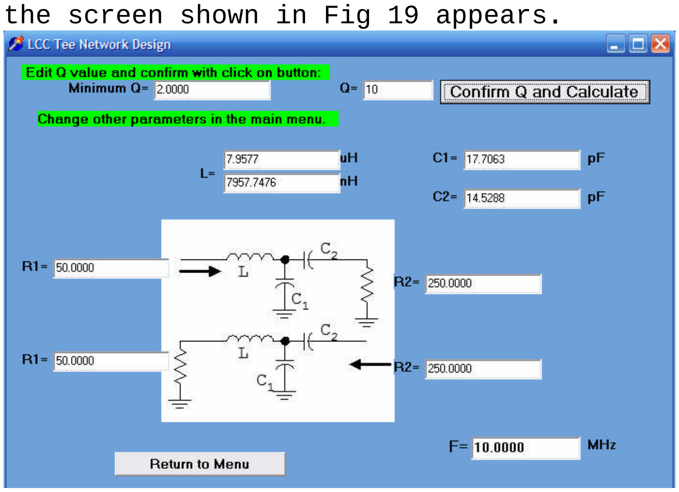

Fig 19. The screen associated with the design of an LCC Tee network.

Two schematic diagrams are shown above. The example presented transforms between 50 and 250 Ohms. The network terminated in 250 Ohms at one port produces a 50 Ohm input impedance at another. The same network terminated in 50 Ohms, as shown, produces a 250 Ohm impedance at the other end. But the user should not get the two confused. That is, placing 50 Ohms on the high Z port will produce much different result than expected, and certainly not 250 Ohms.

The main screen is the place where frequency is entered. Although it is shown on the screens for the individual networks, the user must return to the main menu to change frequency.

Many of the networks require that the user specify a Q. In the case of impedance matching networks, this parameter is the ratio of a reactance to a resistance, and may only be approximately related to bandwidth. If the specified Q is not compatible with the transformation, the program will change the Q, but will warn the user that this is happening.

The usual equations that we see for impedance transforming networks occur in a form where one resistor value must exceed the other. The user can be more cavalier with zmat08 for the program will take care of those details.

All of the networks included may be inserted and studied within GPLA08. They must be entered through Ladbuild08. Inductors and/or capacitors can be added to terminating resistance to approximate situations with complex impedances that must be transformed. Such situations are best treated with the Smith Chart.

12. Q-Measurement

This program is one used at w7zoi, for this is a Q measurement scheme that I often apply. It seemed reasonable to include it with the present collection. The details for using the method are outlined in Chapter 7 of EMRFD. The screen is shown in Fig 20.

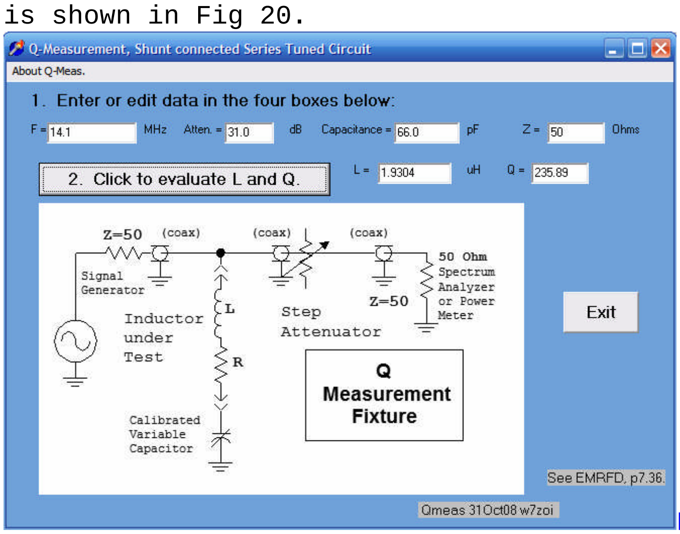

Fig 20. Q and inductance measurement.

The evaluation of an inductor requires that it be resonated with a variable capacitor and then placed in a 50 Ohm environment as a shunt element. It is especially important that the output of this test set up be a clean 50 Ohms. A spectrum analyzer with at least 10 dB of input attenuation works well. The variable capacitor is adjusted for a large dip at the frequency of interest. The attenuation related to the dip is measured with as much accuracy as possible and is recorded. Then the inductor is removed from the test set and the capacitor is measured, usually with an LC meter by Almost All Digital Electronics. The C and frequency data are entered in the program and the button is clicked to evaluate the inductance and Q.

This method has produced good correspondence with other methods for Q determination.

13. Spurtune08. Evaluation of Mixer Spurious Output Frequencies.

Most of the circuits that we build include at least one mixer. A mixer is a nonlinear circuit that accepts two input frequencies: A local oscillator at frequency L and an RF, or radio frequency, at frequency R. Then the mixer output will occur at a sum or difference frequency of L+/-R. But there are other outputs, some strong and others not. These are undesired spurious outputs, or "spurs." They occur at n•L+/-m•R where n and m are integers. The spurious outputs are a direct result of device nonlinearity. Classic literature offers ways to examine the output spectrum of mixers. See, for example, Rohde and Bucher, "Communications Receivers: Principles and Design," McGraw-Hill, 1988, pp106-115. Although the charts in this and other references are very useful, they can also be confusing.

Presented in this program is an alternative method for analysis, one that we first presented with the ARRL edition of **Introduction to RF Design**. The program starts with a spectrum display such as the one we would see in an analyzer. The user specifies the output spectrum as a center frequency and span. Both a RF and a LO are specified as a beginning frequency and a tuning step. After these initial specification, a plot is initiated from a button. The initial view of Spurtune08 after Plot has been clicked is shown in Fig 21.

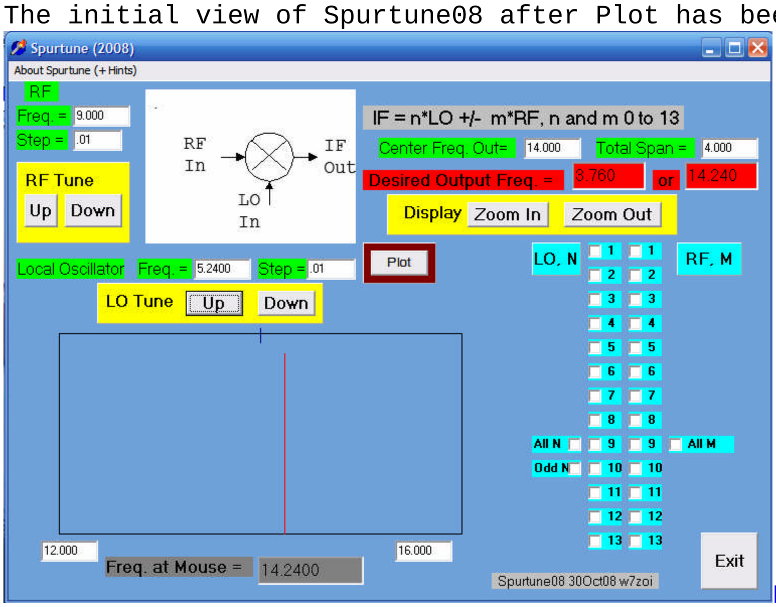

Fig 21. Spurtune.

The example shown is that built into the program. A local oscillator at 5 MHz mixes with an RF signal at 9 MHz to produce an output at 14 MHz. The output shown on the "spectrum analyzer" is a single red line, which is the desired output.

The output signal shown can be tuned by control of either the RF or the LO input signals. Assume that the LO is to be tuned. Click on the Up button in the LO Tune box and the red line will move up slightly. The new output frequency can be determined in two ways. First, a large red label says "Desired Output Frequency." The output is then shown in that line. Both the difference and sum frequencies are shown. Additionally, the mouse can be moved over the spectrum display. The corresponding frequency is shown in a gray box below the spectrum.

There is a subtle detail that is shown in the display that turns out to be very useful. The LO Tune **UP** button had been clicked to increase that parameter. Since that was the last button clicked, it is the active one, which is shown by the dotted line around the "up" lettering. If the user presses **Enter** on the Keyboard, the active function will be repeated. Moreover, if an arrow key (any of them) is pressed, the active status is toggled to the DOWN Key. This makes it very easy to quickly tune the analyzer. (Give it a try.)

This display does not yet show us anything of interest. No spurious outputs are presented. Spurious outputs are generated and displayed only when the user selects the **n** and **m** numbers that will be used. n and m up to 13 can be selected with the check box section to the right of the spectrum display. They can be turned on individually, or in groups. To see what can happen, select "all n" and "all m", and then click on LO **UP**. Now press the Enter key and watch the action. Figure 22 shows one possible result with spurs.

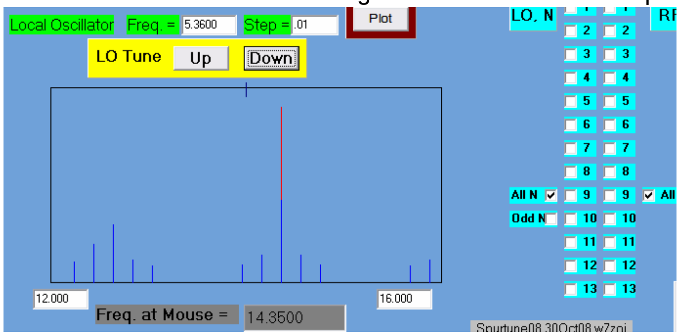

Fig 22. Spectrum with some spurious responses present.

Notice in Fig 22 that the vertical height of the spurs varies. This is what happens in the real world. Spurs resulting from low orders of n and m will tend to produce the strongest response while the higher orders yield weaker signals. In this regard, we have built an algorithm within the program that sets the height in proportion to the reciprocal of (n+m). A characteristic of switching mode mixers, such as diode rings and some FET circuits, is that the spurious outputs tend to emphasize the odd orders of the local oscillator. In that regard, a feature has been included to allow all odd orders of **n** with a single click.

14. Padcap08 : Padding capacitors for tuned circuits.

Often we wish to build a tuned circuit for us in, for example, an oscillator and the only variable capacitor in the junk box is one with a standard capacitance of 365 pF. Yet the inductor we have available for the oscillator requires, for example, only a 30 pF variation to tune the range of interest. A fixed capacitor may be added in series with the variable to obtain a desired smaller range. However, that often produces severely nonlinear tuning. What is really needed is to add a series fixed capacitor and additional fixed capacitance in parallel with the variable. The problem is an easy one, using nothing other than the basic condition for resonance. Still, a computer program is handy for the application. The one we have generated is shown in Fig 23.

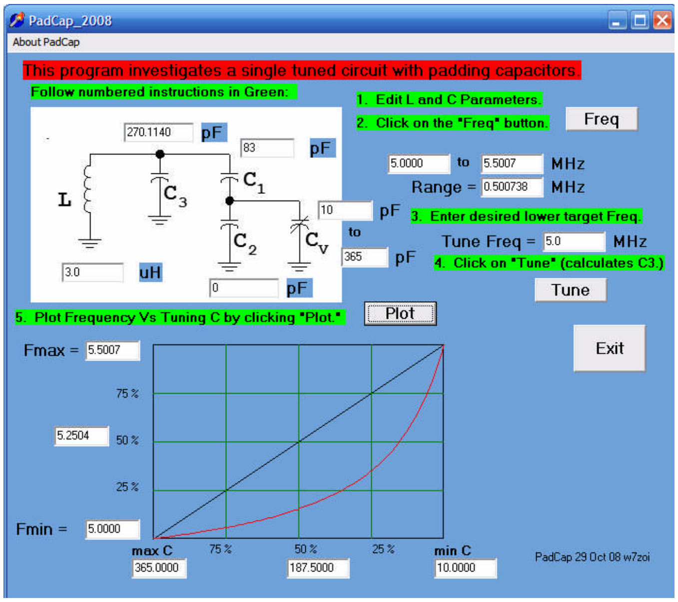

Fig 23. Padcap08 analyzes the single tuned circuit for minimum and maximum frequency for a specified variable.

Operation is straight forward and follows the instructions in green. Enter the inductor and the various capacitor values and then click on "Freq" to see the resulting frequencies. If you want to start at a given lower frequency, enter that value and click on "Tune." This will adjust C3 to realize that value. After the initial analysis, click on the "plot" button to cause a graph to appear.

The example shown in Fig 23 is the extreme case for a 3 uH inductor of using no extra capacitance across the variable (C2), but adjusting the series capacitor (C1) for a 500 kHz tuning range with the 10 to 365 pF variable. Note the extreme nonlinearity. The higher C half of the variable capacitor only tunes the circuit over about 12% of the total range. If C1 is increased and a capacitor is added at C2, improved linearity results. Note that the graph merely uses capacitance for the horizontal axis rather than rotation angle. Shaped capacitor plates will do very interesting things.

15. Designing a 14 element crystal filter with XLAD: A case study

This section is included to illustrate methods that allow a high order filter to be designed beyond the limits of the program. Here a 14 pole filter is designed even though the program is only configured for 10th order crystal filters.

The first step will be a simple and quick *thought-experiment* to illustrate a property of most crystal filters: They are usually symmetric with regard to part values. We will use 5 MHz crystals for this filter, and will continue with those parts for the 14 element circuit. The crystals have Lm=0.11 H, C0=3 pF, and Qu=240000. (These are not microprocessor clock parts, but are from the writer’s junk box.)   Xlad08 is used to design a filter with 2000 Ohm terminations and 6 crystals. This filter appears in Fig 24.

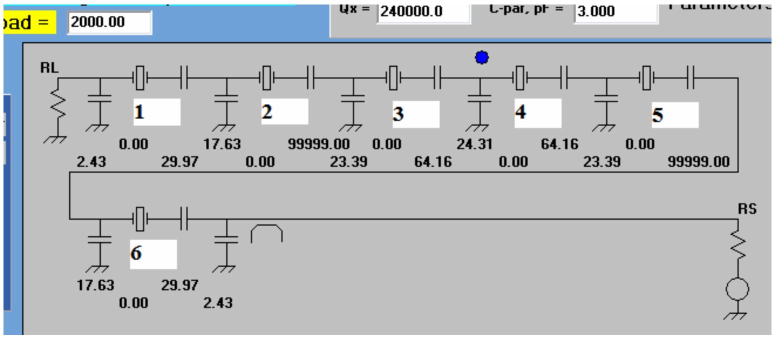

Fig 24. The meshes are numbered starting at the load. Mesh 1 and mesh 6 are identical with 2000 Ohm terminations paralleled by 2.4 pF capacitors, identical crystals, 17.6 pF coupling capacitors and 30 pF series tuning capacitors. Mesh 2 is identical to mesh 5, while mesh 3 is the same mesh 4.

A symmetry like that of Fig 24 occurs in all Butterworth and Chebyshev crystal filters designed with so called pseudo-exact method, which is the scheme used with XLAD. This symmetry disappears with predistorted filters such as those popularized in the classic Zverev text.

We will take advantage of this symmetry for this 14 element design. The filter will be designed for a 2000 Hz bandwidth with the 5 MHz crystals described.

Launching XLAD08, we enter crystal data and click on button #1. Enter 2000 Hz for the bandwidth and N=14 and click button #2. Continue with 2A, calculating the k and q for a N=14 Chebyshev filter with a ripple of 0.1 dB. (This routine is good for k and q calculation for N>20.) Return to the basic program, now with the first 10 k values in place for a 14th order filter. The proper q values are there for the 14 element filter.

A trap has been built into the program to warn the user if there are more than 20 crystals.

Continue the process. It all works fine until step 6 is reached. The tuning values shown will not be valid and an error message will so indicate. There is a way of fooling the program though. Having obtained some of the N=14 coupling capacitors, return to step 1 and click the button. Change the N value from 14 to 10 and click button 2. **DO NOT CLICK 2A**. Continue, clicking buttons 3 through 6. These values should be valid, although incomplete.

Now save the present circuit to the default file. Open the editor, ladbuild08, and load the default to see a filter with 10 crystals. The first thing to do is to go to the 99999 pF capacitor in the first mesh and remove it. That high value is just a “shortcut” to indicate that no tuning capacitor is needed. Then go to the last capacitor (source end) and add wires until the return loss bridge is the final element. There will be a long chain of wires. Build four additional meshes without regard for values. The last one, at the source end of the filter, should not need a series tuning capacitor.

We now take advantage of the symmetry of the filter and use the “duplicate” function. Place the cursor over the last capacitor at the source end and click to select component #41. Then click on the button that converts this part to a duplicate. In the part value box, enter 1, for you wish to duplicate the first capacitor at the load end. The process is repeated until you read the 25.03 pF coupling capacitor, which is the center of the filter. The final 14th order filter circuit is shown in Fig 25.

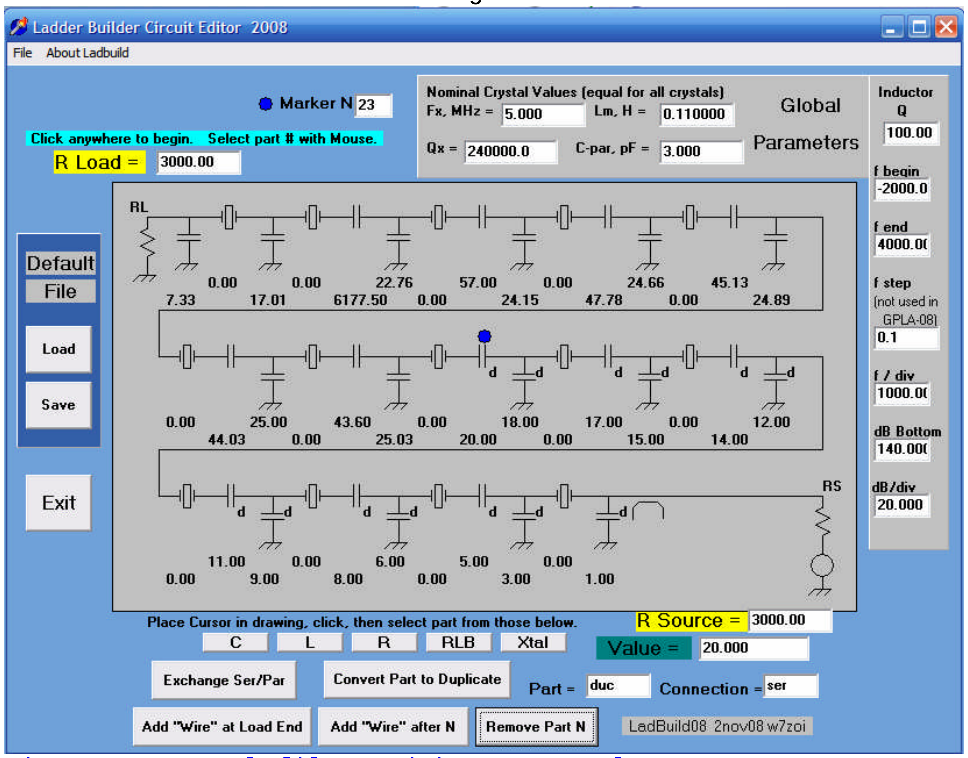

Fig 25. Crystal filter with 14 crystals.

Save the filter as the default file. Then run the analysis program, gpla08. This will produce the response shown in Fig 26. After running the program the first time, we changed the sweep and response limits to show filter behavior to the -140 dB level.

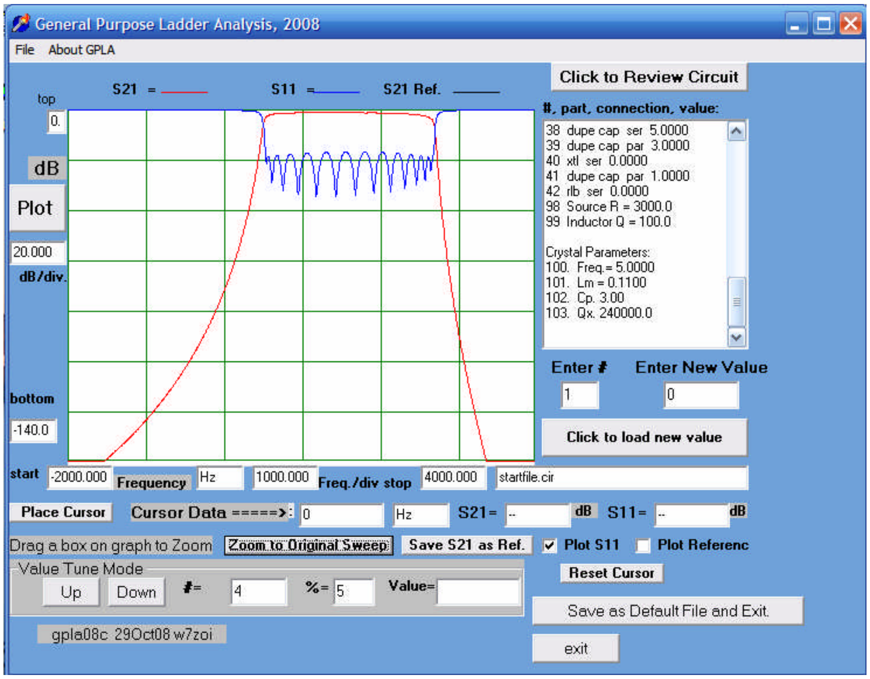

Fig 26. Response of the filter. This circuit is saved on disc as cheb14final.cir. This filter is a spectacular one with very steep skirts. Still, the frequency response asymmetry typical of the *lower sideband ladder* topology is evident. Similar filters built at high frequency (10 MHz) or at 5 MHz with narrower bandwidth will have a response with improved symmetry.

Not all filters are symmetrical with regard to component value. But they can still be designed with xfil08. If the filters are asymmetric, they will have asymmetric coupling coefficients. The coupling coefficients can still be loaded into the program, allowing the coupling capacitors to be calculated. If the filter is to use more than 10 crystals, k and q parameters can be loaded out of order to do the calculations. In that situation, it will probably be necessary to do all tuning with meshtune08.

Correspondence

Correspondence regarding these programs should be sent to w7zoi@arrl.net .