

Mini MRI Senior Project

Winter 2004

Junior Salles

Introduction

In this senior project I was mostly engaged in the design and implementation of the receiver end of our magnetic resonance device. Most of my effort was put on the various stages of implementation of the receiver's frequency filters. Later on, I also worked with the coils matching network and other miscellaneous tasks that needed to be done.

Our first weeks into this project were mostly taken by the business lecture assignments we had to do, although I started working on the frequency filters since one of our early meetings. In the first four weeks counting only business lecture assignments, I spent about 2 ½ hours/week in the senior project.

Filter Design

Using a design similar to the one that the software radio senior project uses, we noticed that we needed a 2.5MHz low pass filter and a band pass filter centered at 42.5MHz. We decided to replace the band pass filter with a 43MHz low pass and a 41MHz high pass because we thought they were simpler to design.

Once we determined that I was going to be responsible for the filters, David Thayer suggested me to use the Matlab code that David Casbeer, a student who worked in this project on the winter '03 semester, had created for filter design. The code is capable of designing a Chebyshev and a Butterworth filter and outputs the component values according to the filter order the user chooses. I used the Chebyshev filter design because Ryan said it is better for analog filters. The major usefulness of this code is to define de place of each component on the design and approximate component values, although these values must be rounded and sometimes changed in order to achieve the desired frequency. Appendix A contains a copy full of the Matlab code.

After working with these values I was able to create a final design and run PSpice simulations for the 2.5MHz low pass filter (figure 1 & 2), the 43MHz low pass (figure 3 & 4) and the 41MHz high pass (figure 5 & 6).

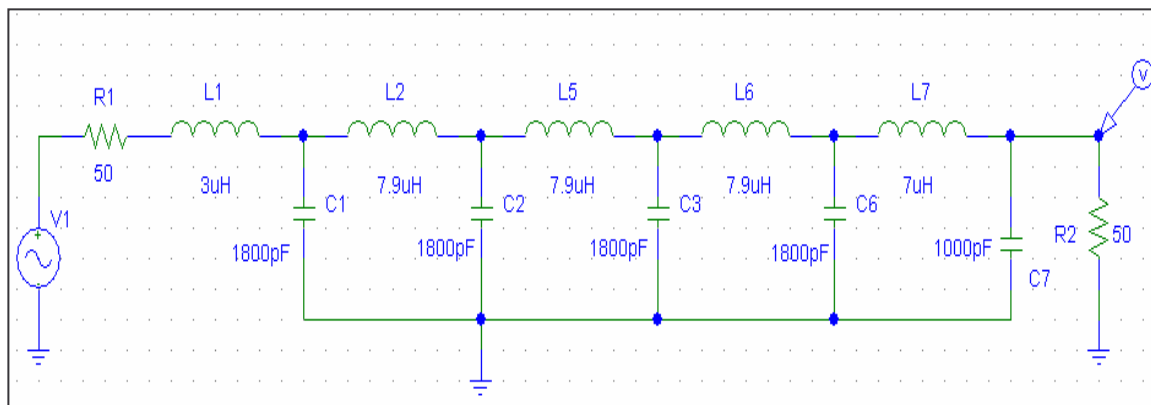


Figure 1: Design of 2.5MHz low pass filter

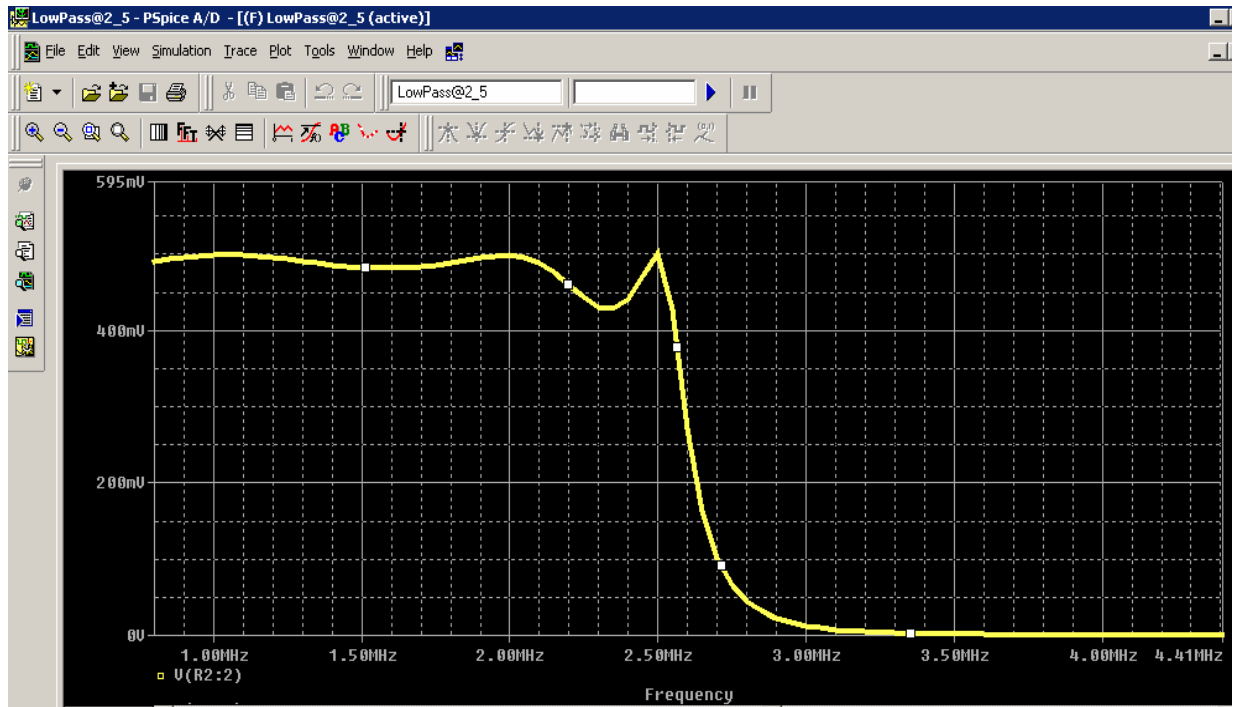


Figure 2: Simulation of 2.5MHz low pass filter

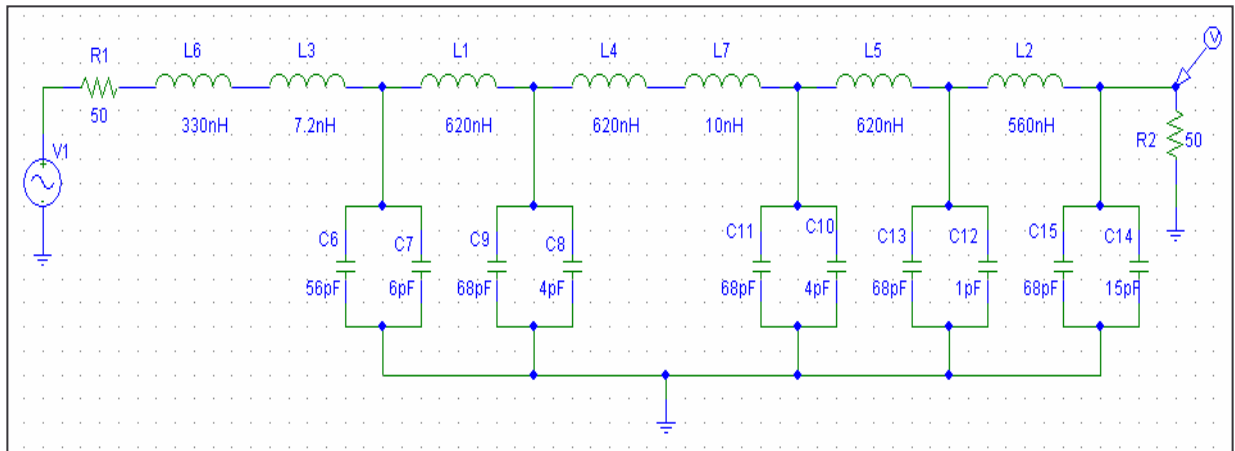


Figure 3: Design of 43MHz low pass filter

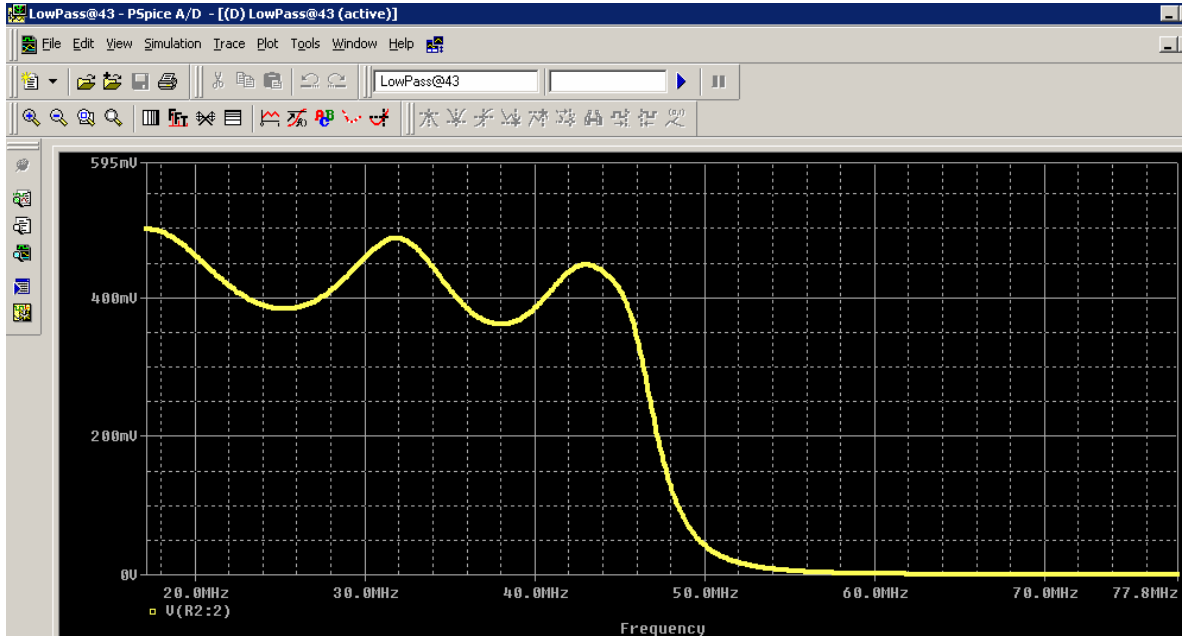


Figure 4: Simulation of 43MHz low pass filter

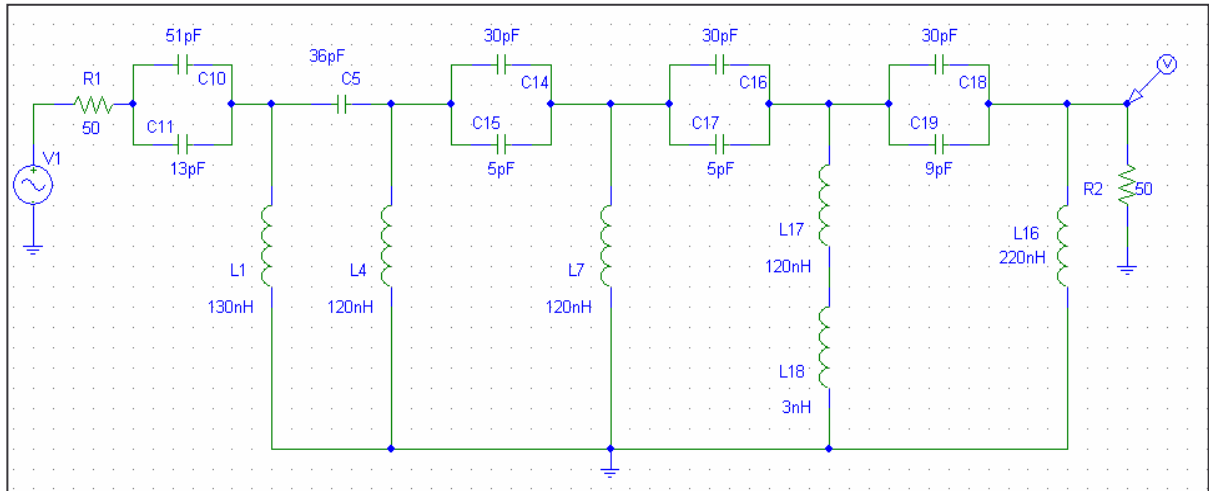


Figure 5: Design of 41MHz high pass filter

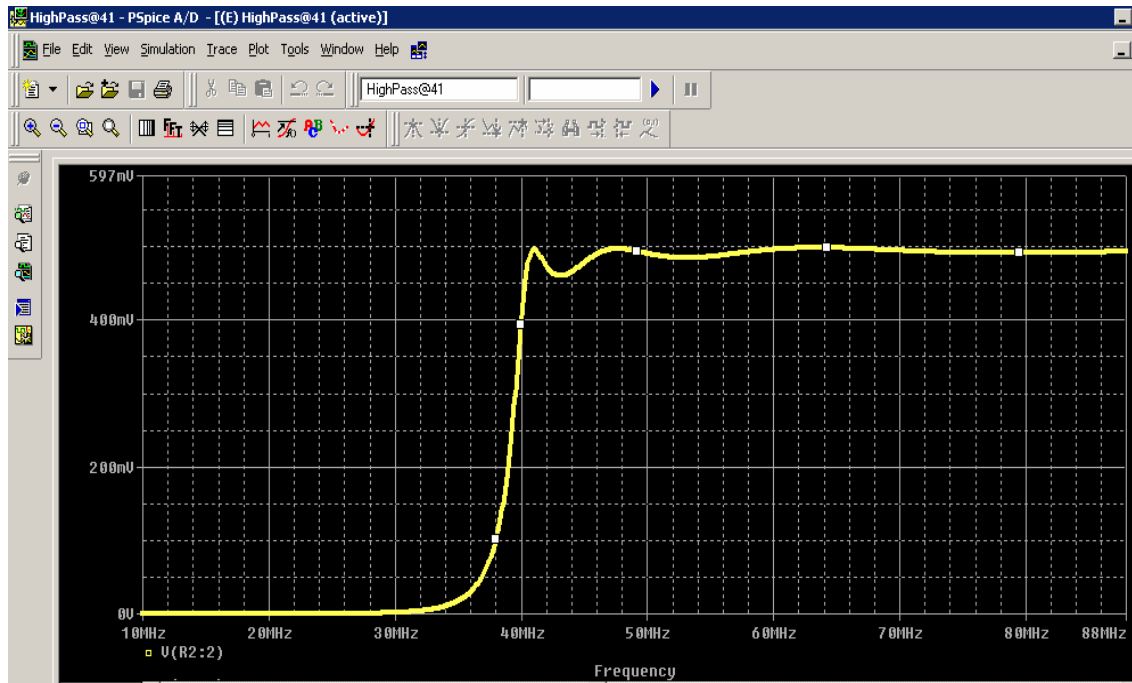


Figure 6: Simulation of 41MHz high pass filter

I had to change the component values constantly because I wanted to make sure that the capacitors we had in the design could be found in the shop. Since we had decided to order inductors from Coilcraft, we knew we could find pretty much any inductor needed. The design process was very time consuming and I spent around 14 hours/week more on that.

Filter Layout

Since these simulations results were what we expected, we decided to mill test boards in order to test the filters and see if they would give acceptable responses to the desired frequencies. So, I went to the shop and at Coilcraft's homepage to find the dimensions of each capacitor and inductor and created all the capacitors and inductors footprints and the filter layouts in ADS. These layouts are shown below (figure 7, 8 & 9).

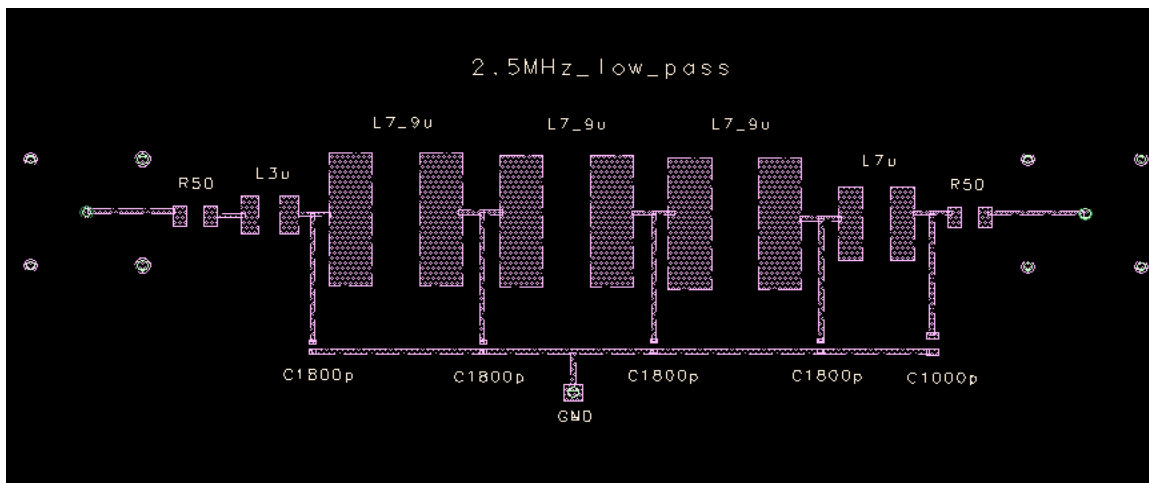


Figure 7: Layout of 2.5MHz low pass filter



Figure 8: Layout of 43MHz low pass filter

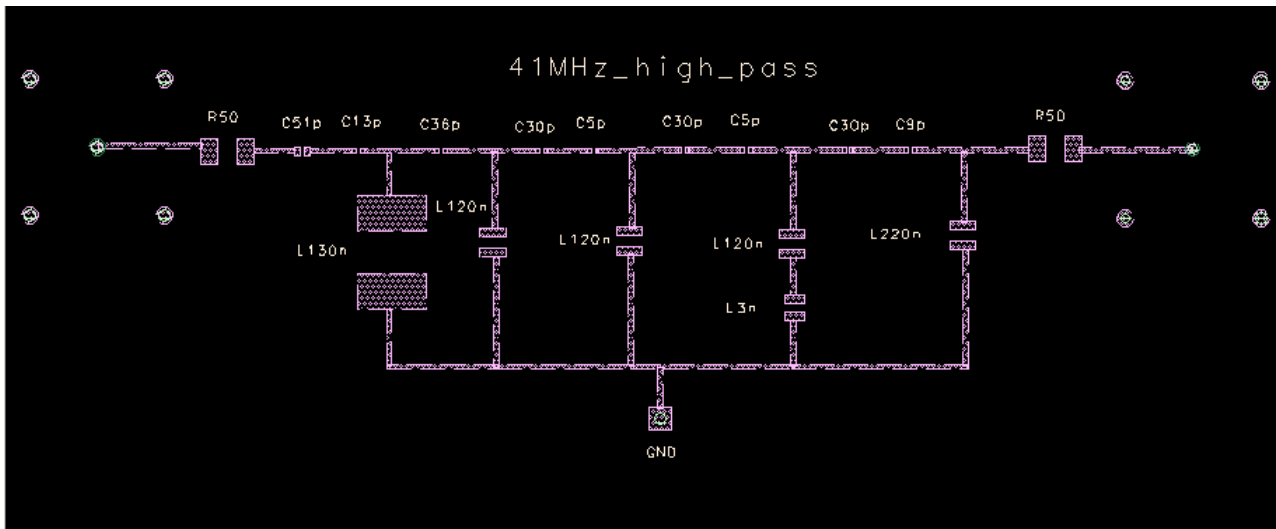


Figure9: Layout of 41MHz high pass filter

The filter layout process took about 2 weeks and I spent around 5 hours/week working on that.

Filter Testing

After the filter boards were milled, James and Dave populated them and we started testing them on the spectrum analyzer. We wanted to make sure the filters were working properly before finishing the PC board layout and sending it to be milled. It turned out that the 2.5MHz filter worked really good, just like simulated on PSpice. However, the 43MHz and the 41MHz gave us responses that were not exactly what we saw on the simulations, but they were still acceptable. Before the PC board layout was sent to be milled, Ryan asked me

to “proofread” the filters layout. I noticed that one of the filters had a layout mistake, which Ryan fixed before sending the layout to Advanced Circuits.

This process took 2 more weeks and I spent about 4 hours/week working on this testing.

PC Board

Once the PC board came back and Dave populated it, we started testing the filters. Again, we noticed that the 2.5MHz low pass filter worked great, but the 43MHz low pass and the 41MHz high pass did not work well. We tried to test them separately, together, with and without the 50 Ω resistors, but in all cases they showed an “all pass” response. We then started deliberating the reasons the filters did not work. One possibility was the missing ground plane on the PC board. This might have affected the capacitor functionality since they were so small and possibly so sensitive in these filters. That is our major assumption, once the 2.5MHz worked fine because its capacitors were much bigger and therefore, not so sensitive to the missing ground plane. In this testing process I spent around 5 hours/week. As a matter of suggestions, I think that in the future students could avoid the hassle of high frequency filter design uncertainties by simply buying a band pass filter. Another suggestion is to buy the capacitors from a credited source where students can know component specifications before buying them. The components we got from CB 416 (shop) do not have spec sheets, so we faithfully assumed they would work. One thing we should have done to be more specific in our final assumptions were to measure the capacitances on the filter components on the PC board. This would help us determine if the capacitors were actually as sensitive as we thought they were.

Matching Network

As we advanced in our testing process we wanted to assemble the whole hardware to check if it would work. So, Dave asked me and Chad to create a matching network for the inner and outer coils respectively. I got a complex impedance analyzer from Dr. Manwaring’s project in order to obtain real and imaginary values that would correspond to our frequency range. We plugged these values in ADS and got designs that would possibly match our coils impedance. We then ran simulations in such designs and obtained results that Dave considered acceptable. After that I created component footprints and put together a small board layout to build the matching network. Below (figure 10) is the inner coil matching network layout.

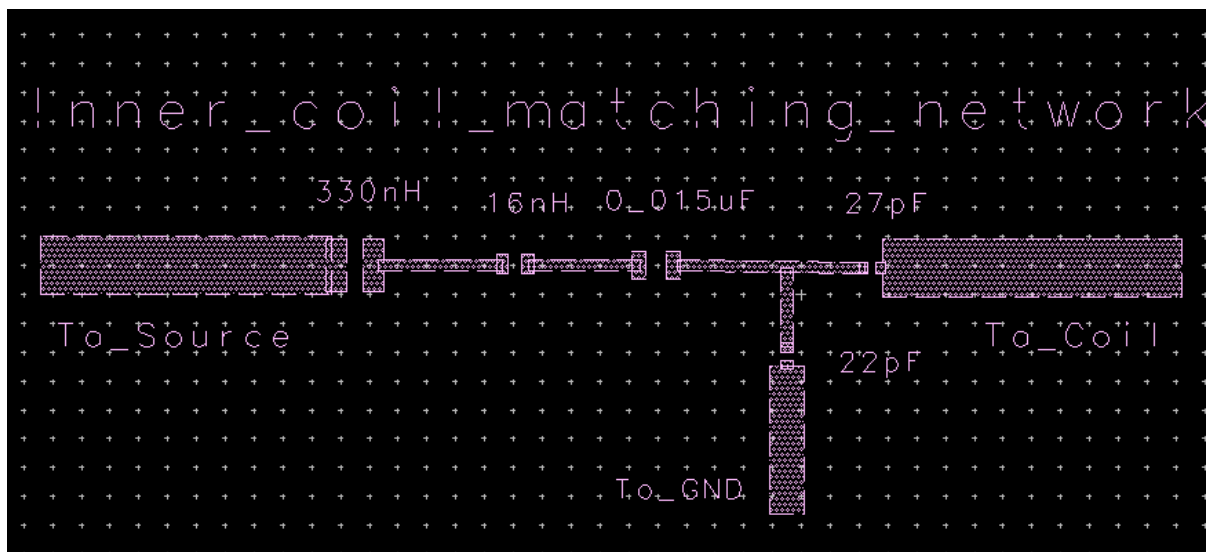


Figure 10: Inner coil matching network layout.

We really wanted to test if this network would work, but because of the problems with the power amplifier, we could not test the whole device and therefore we could not know if this really works. I worked in this network for about 1 ½ weeks and it took me around 6 hours/week to do it.

Other Tasks

Besides these major processes, I also did some other smaller tasks in this project. For example, I created footprints for all capacitors and inductors used in the whole project and either ordered them from Coilcraft or got them in the shop. I also created some slides for both presentations to describe our hardware functionality. Dave mentioned during the semesters that we probably needed signs on the lab to warn visitors about the cataract risk from the power amplifier, the strong magnetic field our magnet has and the risks of RF burn. So I spent around 7 hours in the middle of the semester to make and print these signs.

The week of the senior project exposition in the Wilkinson Center we all gave an extra effort to try to make the device to work. We also set up a stand in the Wilkinson Center for our project even knowing it probably would not work. We still wanted to expose it at least to serve as instruction to people that saw the equipment and asked question about NMR. During this week I spent around 9 hours doing whatever was needed to have our project exposed. If we did not have problems with the power amplifier, maybe we could have gotten acceptable results from our efforts.

And at the end, when we had to prepare our last presentation to Dr. Oliphant, I was asked to prepare a slide with our filters simulations. This task and others like preparing the lab and getting the necessary equipments took me about 4 hours.

Conclusion

I really enjoyed the time spent on this project this semester. I believe it was the class where I learned the most in my college career. The challenge of going to figure out things for myself really helped me to learn and grasp concepts that I had not understood yet. The chance to work in group also prepared me for future experiences in my career in engineering. I finish this project a little frustrated for not being able to see our device working, but happy for everything I learned. I worked hard and for a long time, but it was worth every minute. Below is an approximate description of the time I spent in this project.

Task	Time spent
Business lecture assignments	10
Filter Design	56
Components footprints	9
Filter Layout	10
Filter Testing	8
Matching networks	13
Searching project components	9
Signs	7
Presentations	12
Documentation	6
Total hours	140


```

    lc(i)
end;
elseif type == 'h'
    for i = 1:length(lc)
        lc(i) = abs(1/lc(i))/cutoff;
        lc(i)
    end;
end;

%Other function below %
%% Hopefully this will compute a ChebyChev filter
%% Start date: 5/2/03
%% Number's to use before making a function
clear;
order = 10;
ripple = 1;
R = 50; %% Load resistances
type = 'h'; %% 'l' for low or 'h' for high pass
cutoff = 1.6e6 *(2*pi);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculates the Chebychev Polynomials
% I could have used matlab to compute the transfer
% function for me, but I want to try to solve for
% the transfer function dynamically myself, and
% then get the transmittance to solve for the
% input impedance.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NOTE: Put a table into your writeup.

chebn2 = [1]; %Co(w) = cos(0*acos(w)) = 1;
chebn1 = [1 0]; %C1(w) = cos(1*acos(w)) = w;
if order == 1
    cheb = chebn1;
else
    for q = 1:order-1
        cheb = 2*[chebn1 0] - [0 0 chebn2];
        chebn2 = chebn1;
        chebn1 = cheb;
    end;
end;
%% cheb = the cheby polynomial for order specified
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Reflection Coefficient: rho(s) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% rho(s) = 1 - abs(TF)^2;
eps = 10^(ripple/10)-1;
chebsqr = conv(cheb,cheb);
rhoSQnum = eps*chebsqr;
rhoSQden = [zeros(1,order*2) 1] + eps*chebsqr;

%% This is abs(rho(s))^2

rnum = roots(rhoSQnum)*-i; %% *i because w^2 = -s^2 ???
rden = roots(rhoSQden)*-i;
lhnumr = [];
lhdenr = [];
check = 1;
check2 = 1;

```

```

%% Find lhplane zeros & poles
%% In other words the negative real parts of poles
for q = 1:order*2
    if (real(rnum(q)) <= 0)
        for z=1:length(lhnumr)
            a=imag(rnum(q));
            s=imag(lhnumr(z));
            s=a-s;
            if (abs(s)<=.0001)
                check = 0;
            end;
        end;
        if check == 1
            lhnumr = [lhnumr rnum(q)];
        end;
    end;
    check = 1;

    if real(rden(q)) <= 0
        if (check2 == 1) | (real(rden(q)) < 0)
            lhdenr = [lhdenr rden(q)];
        end;
        if real(rden(q)) == 0%% Note: Not sure which pole not
            check2 = 0;    %% to take neg or pos imag part
        end;
    end;
end;
lhnumr';%%%%

rhonum = poly(lhnumr);
rhoden = poly(lhdenr);
if type == 'h'
    rhonum = fliplr(rhonum);
    rhoden = fliplr(rhoden);
end;

%% We have numerator and denominator of rho(s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Now calculate the input impedance with rho(s)

Znum = (rhoden+rhonum)*R;
Zden = rhoden-rhonum;
if type == 'l'
    Zden = Zden(2:length(Zden));
elseif type == 'h'
    Zden = Zden;
end;

%This step calculates the C & L values by taking the poles out
% of the reflection coeff equation.
if type == 'h'
    Znum = fliplr(Znum);%flips them to perform long division
    Zden = fliplr(Zden);%as shown on page 151
    Zden = Zden(2:length(Zden));
end;

for c = 1:order
    [p,r] = deconv(Znum,Zden);
    lc(c) = p(1);
    Znum = Zden;
    Zden = r(3:length(r));
end;

```

```

if type == 'l'
    for i = 1:length(lc)
        lc(i) = abs(lc(i))/cutoff;
        lc(i)
    end;
elseif type == 'h'
    for i = 1:length(lc)
        lc(i) = abs(1/lc(i))/cutoff;
        lc(i)
    end;
end;

%Calculates the transfer function(TF) of the circuit with the
% C & L's that we solved for earlier. Then plots the TF.
index = length(lc);
i5 = [0 1/R] + [lc(index) 0];
v4 = [0 0 1] + [lc(index-1)*i5 0];
if order > 2
    index = index-2;
    i4 = lc(index)*[v4 0] + [0 0 i5];
    if order == 4
        v3 = conv([lc(index-1) R],i4) + [0 0 v4];
    else
        v3 = lc(index-1)*[i4 0] + [0 0 v4];
    end
end
if order > 4
    index = index-2;
    i3 = lc(index)*[v3 0] + [0 0 i4];
    if order == 6
        v2 = conv([lc(index-1) R],i3) + [0 0 v3];
    else
        v2 = lc(index-1)*[i3 0] + [0 0 v3];
    end
end
if order > 6
    index = index-2;
    i2 = lc(index)*[v2 0] + [0 0 i3];
    if order == 8
        v1 = conv([lc(index-1) R],i2) + [0 0 v2];
    else
        v1 = lc(index-1)*[i2 0] + [0 0 v2];
    end
end
if order > 8
    index = index-2;
    i1 = lc(index)*[v1 0] + [0 0 i2];
    v = conv([lc(index-1) R],i1) + [0 0 v1];
end;

if order == 'l'
if order <=10
    figure(1);
end;

switch order
case 2
    freqs(1,v4);
case 4
    freqs(1,v3);

```

```
case 6
  freqs(1,v2);
case 8
  freqs(1,v1);
case 10
  freqs(1,v);
end
end;
```