

Design of circuit for SiC FET gate driver and its PCB implementation

Rahul Radhakrishnan

Special Problem with Professor Kuang Sheng

Fall 2005

Aknowledgements

I would like to thank Professor Kuang Sheng for his kind guidance throughout this project, on matters both theoretical and its practical implementation, which has played a major hand in completing this work. On many occasions, he had pointed out things that had been staring at me, but I had failed to recognize.

Genuine thanks are also extended to Mr.Tony Farthing Jr. whose help in the early stages was vital in getting me started on the software.

Iam also obliged to Mr.Abhijit Solanki for his advice and company in the lab.

Contents

1.	Introduction
2.	Design of Circuit elements
3.	Study of OrCAD Layout
4.	Creation of PCB footprints
5.	Design of the board template file
6.	Netlisting and creation of board design
7.	Circuit board fabrication
8.	Conclusion

List of figures:

1. Screenshot from OrCAD Capture showing the circuit design.
2. Output of the second monostable section.
3. The output of the circuit.
4. Front and back sides of the layout design
5. Camera shots of the PCB.

Introduction

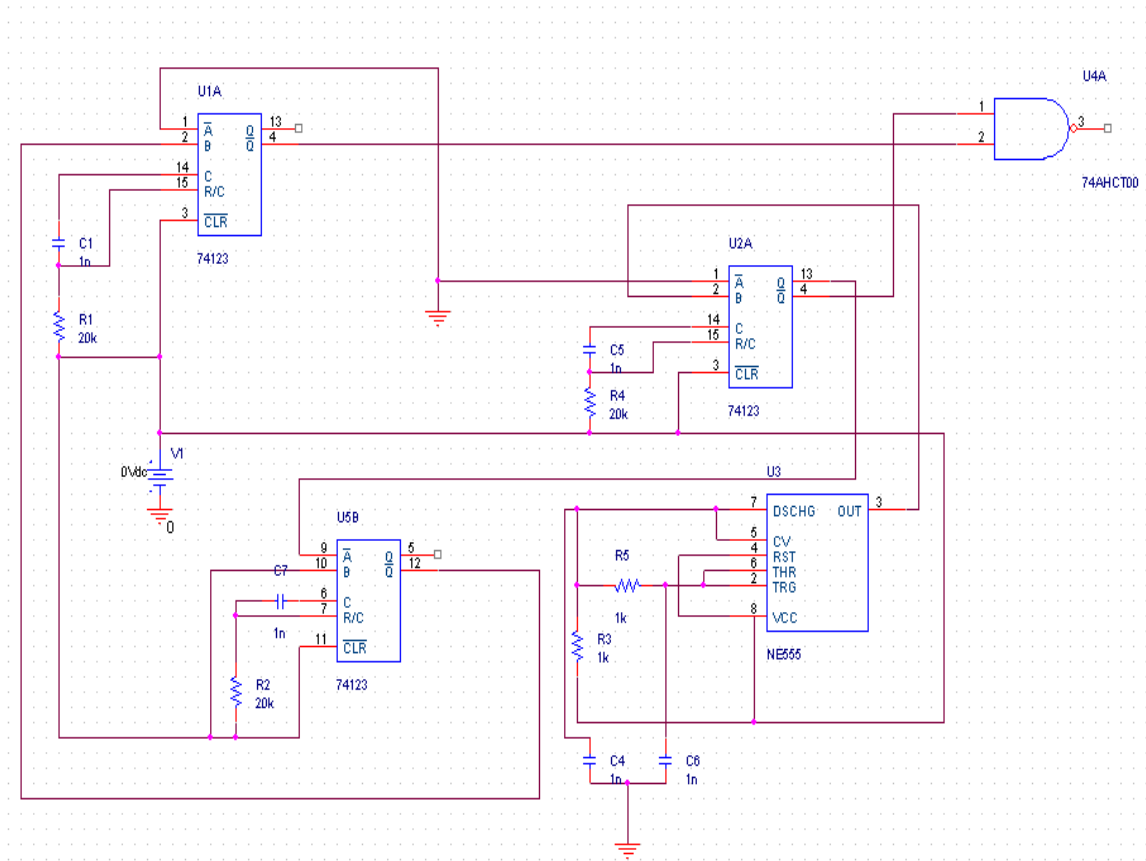
The project involves the design of circuit variables and PCB implementation of a gate driver circuit for testing a SiC MOSFET. The standard test procedure involves generating two pulses in a rapid burst followed by a long lean period, and repeating this pattern with a frequency. In this case, it is required to have a pulse of variable width from 3 to 200 μ s, followed by a period of inactivity lasting .5 to 10 μ s, and then another pulse of width that can be varied between .5 and 10 μ s. The variation will be achieved by means of variable resistors in the circuit, and the whole variations should be available when one moves over the whole range of the variable resistor.

The logic of the circuit involves cascading sections of monostable multivibrators. At first, a 555 timer is set up to run at the frequency at which the succession of pulses is to repeat. So, the 555 is in its astable mode of operation, and by means of a variable resistor, the frequency of the clock generated can be tailored to the required range. The duty factor of the astable clock is very low as we need a very high burst frequency when compared to the frequency of the train. The monostable sections used are downward edge triggered. So, if the output of the 555 timer is given to the monostable, it will trigger at the negative going edge of its input. Then the monostable output would be a pulse which lasts for an interval depending on the resistors and capacitors in the monostable circuit, starting at the time the 555 moved to zero. The output of this section is fed to another monostable multivibrator, which will trigger when its input goes to zero, and last for a certain time depending on R and C. If this output is fed into a third monostable section, then we have a pulse as its output too. So, the output of the third section is a pulse which is separated from the first section by the pulse width of the second section.

Now, if the first and third section were to be sent to a NAND gate, then the output will be low, only when both the inputs are high. So, a logical 1 can be achieved when and only when both the inputs are high i.e. two short pulses of widths set by the monostable sections#1 and #3, which are separated by the width of the monostable section#2. This is repeated with the frequency of the 555 operating in the astable mode.

The circuit is modeled in OrCAD Capture, and the PCB layout is created through OrCAD layout. Finally, the layout of the board is printed and used as a mask to etch away the rest of a copper board except the parts that will form the tracks. Holes are then drilled at the footprint locations. Now, circuit board is fabricated; the components can be soldered on and the PCB is ready to use.

Fig: 1 Screenshot from OrCAD Capture showing the circuit design



Design of circuit elements

Going with the flow of the signal, the first task is to design the two resistors and one capacitor required to operate the 555 timer in the astable multivibrator mode. They are R_3 , R_5 and C_4 in the fig:1.

Now, the frequency in astable mode is given by

$$F = 1.44 / ((R_a + 2 * R_b) * C)$$

Also, we need to keep the duty cycle low. i.e.

$$R_b / (R_a + R_b) \ll 1$$

With these considerations, and choosing standard values, we can have

$$C_4 = 1 \mu F, R_5 = 220 \Omega - 20k \Omega, R_3 = 22 \Omega$$

Now T_1 , which is the width of the first pulse and is decided by the first monostable section, is can range from 3 to 200 μ s. Inserting in the equation for the monostable,

$$T_2 = 0.28 * R_4 * C_5 (1 + 0.7 / R_4)$$

So, we can design with standard components as

$C_5 = 1nF$, and R_4 is a 20k Ω potentiometer, which can be brought down as low as 100 Ω .

The same equation holds for the other two monostable sections too. So, we have the design as-

$C_7 = 2nF$, and R_2 is again the 20k Ω potentiometer for the second monostable section; and $C_1 = 1nF$, with R_1 being the 20k Ω potentiometer for the third section.

That completes the design of the circuit components. The monostable IC used here is Texas Instruments 74123, and the NAND gate is TI's 741hct00, which is a quad, 2-input gate.

Now the design can be simulated to obtain the expected output waveforms. The simulation tool in the OrCAD family is PSPICE. But, PSPICE can simulate only with ICs which have a PSPICE model. The NAND gate, and the multivibrator don't have a PSPICE model. One way ahead would be to create such a model, which would be similar to a simulation with the same equations that define the model. So, the 2-port reduced equations of the various sections are used in a MATLAB program to generate the output waveforms expected. The MATLAB code for the program is as follows.

```

Ra= 2.75k;
Rb= 22;
C4= 10^-6;
f= 1.44/((Ra+2*Rb)*C4);
F= 1/f*10^7;
R4=20000;
C5= 10^-9;
t1= 0.28*R4*C5*(1+0.7/R4)*10^7;
R2=20000;
C7= 10^-9;
t2= 0.28*R2*C7*(1+0.7/R2)*10^7;
R1=20000;
C1= 2*10^-9;
t3= 0.28*R1*C1*(1+0.7/R1)*10^7;

```

```

for t=1:(3*(F-mod(F,1)))
    b(t)=t*10^-7;

```

```

    if t<=t1
        a(t)=5;
    elseif t<=(F-mod(F,1))
        a(t)=0;
    elseif t<=(F-mod(F,1)+t1)
        a(t)=5;
    elseif t<=(2*(F-mod(F,1)))
        a(t)=0;
    elseif t<=(2*(F-mod(F,1))+t1)
        a(t)=5;
    else
        a(t)=0;
    end

```

```

end
% Output of the first section
figure(1);
plot(b,a)
axis([0 2*10^-3 0 6])

```

```

for t=1:(3*(F-mod(F,1)))
    b(t)=t*10^-7;

```

```

    if t<=t1
        c(t)=0;
    elseif t<=(t1+t2)
        c(t)=5;
    elseif t<=(F-mod(F,1)+t1)
        c(t)=0;
    elseif t<=(F-mod(F,1)+t1+t2)
        c(t)=5;
    elseif t<=(2*(F-mod(F,1))+t1)
        c(t)=0;
    elseif t<=(2*(F-mod(F,1))+t1+t2)
        c(t)=5;
    else
        c(t)=0;
    end

```

```
end
```

```
end
```

```
% Output of second section.
```

```
figure(2);
```

```
plot(b,c)
```

```
axis([0 2*10^-3 0 6])
```

```
for t=1:(3*(F-mod(F,1)))
```

```
    b(t)=t*10^-7;
```

```
    if t<=t1+t2
```

```
        d(t)=0;
```

```
    elseif t<=t1+t2+t3
```

```
        d(t)=5;
```

```
    elseif t<=(F-mod(F,1))+t1+t2)
```

```
        d(t)=0;
```

```
    elseif t<=(F-mod(F,1))+t1+t2+t3)
```

```
        d(t)=5;
```

```
    elseif t<=(2*(F-mod(F,1))+t1+t2)
```

```
        d(t)=0;
```

```
    elseif t<=(2*(F-mod(F,1))+t1+t2+t3)
```

```
        d(t)=5;
```

```
    else
```

```
        d(t)=0;
```

```
    end
```

```
end
```

```
% Output of section 3.
```

```
figure(3);
```

```
plot(b,d)
```

```
axis([0 2*10^-3 0 6])
```

```
for t=1:(3*(F-mod(F,1)))
```

```
    b(t)=t*10^-7;
```

```
    if a(t)==5
```

```
        e(t)=5;
```

```
    elseif d(t)==5
```

```
        e(t)=5;
```

```
    else
```

```
        e(t)=0;
```

```
    end
```

```
end
```

```
%Output of NAND gate.
```

```
figure(4);
```

```
plot(b,e)
```

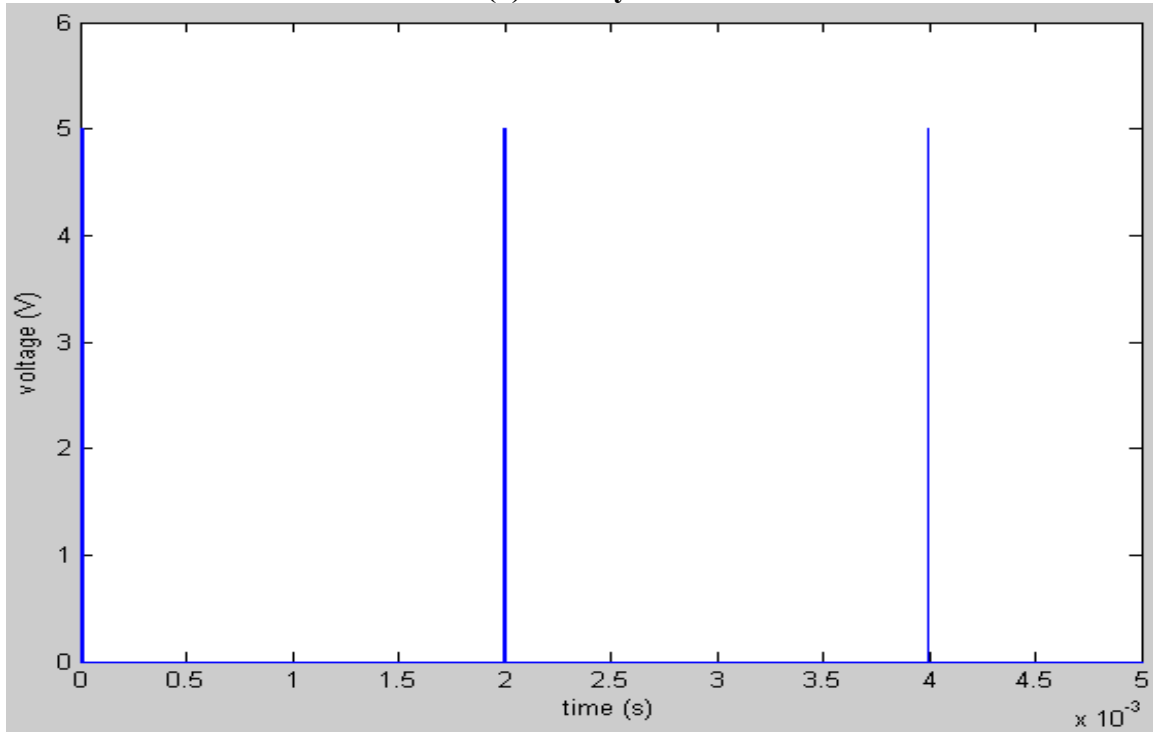
```
axis([0 2*10^-3 0 6])
```

The output waveforms are-

Fig:2 Output of the second monostable section

(The first pulse in the output will end at the beginning of this pulse and the second pulse will begin at the end of this pulse)

(a) Two cycles



(b) Blow up of the pulse

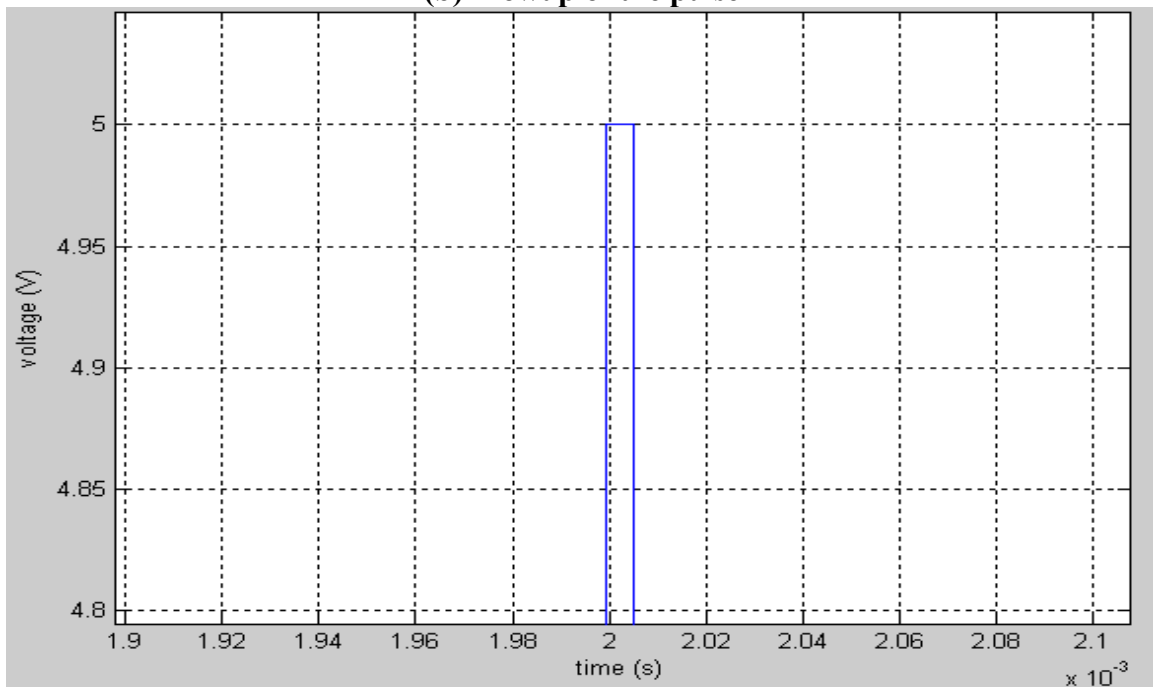
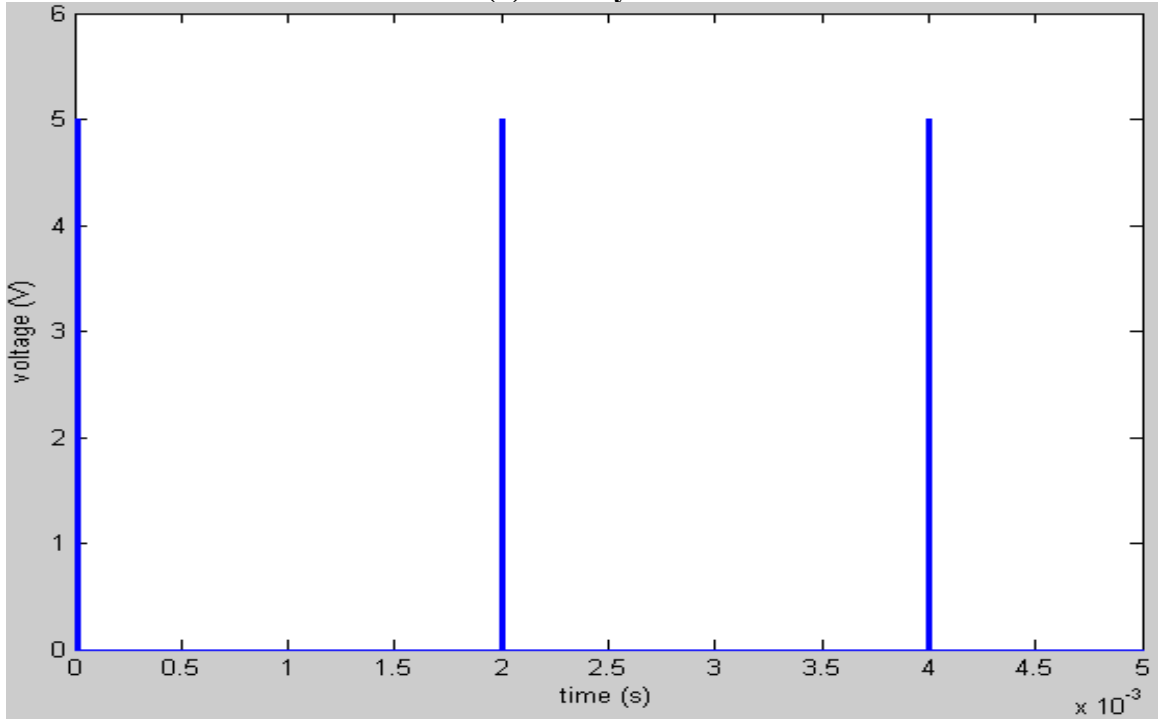


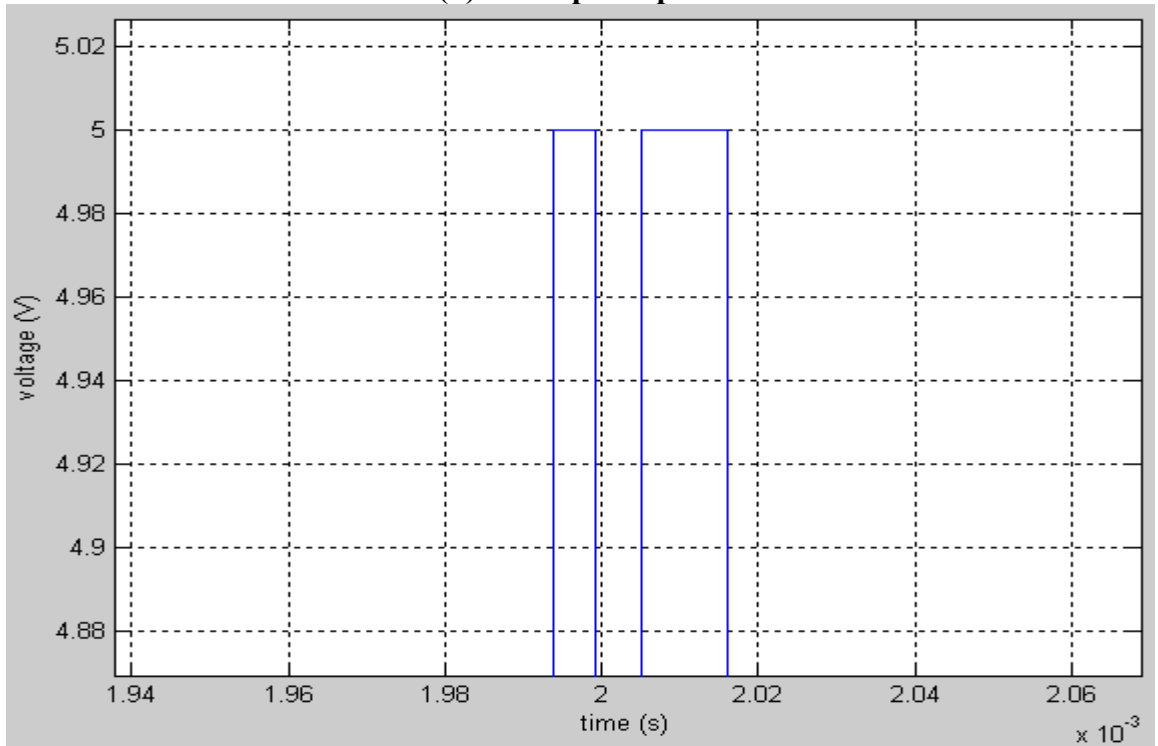
Fig:3 The output of the circuit

(There are two pulses of lengths in time t_1 and t_3 , separated by t_2 , and periodically repeating)

(a) Two cycles



(b) Blowup of a pulse



Study of OrCAD layout

OrCAD is a set of tools from Cadence for the design and layout of printed circuit boards (PCBs). OrCAD consists of two tools. Capture is used for design entry in schematic form, and Layout is a tool for designing the physical layout of components and circuits on a PCB. Work begins with the creation of a new project. Once the circuit has been modeled in Capture, one proceeds to Layout to create its PCB implementation. The work in this project began with the Capture model available, and hence concentrated on Layout.

In Layout, one has to match a unique footprint to every element in the design. Footprints are either available from the library, or have to be created. In this endeavor, footprints were created for all the ICs and discrete components from scratch. Then each footprint is enclosed within an obstacle, which indicates the physical space around the component that acts as a buffer and will not be violated by other components. Once each footprint is matched to its corresponding part, we need the board template file which provides Layout the dimensions of the circuit board where the whole circuit has to be fit.

Layout works with a progression of spreadsheets which hold all the details pertinent to the design. For the footprints, there is a padstacks spreadsheet which contains the geometrical details of the pins. Then there is a footprints spreadsheet which elucidates the placement of the padstacks on the plane. Then, in layout the footprints are aligned in space on their own, which is considered in the layout spreadsheet. This hierarchy of spreadsheets, once mastered makes Layout easier.

Footprints created in Layout are stored in a library, which will then be matched with the ones assigned in capture during netslisting. This allows Layout to club the geometrical shape of the components and the placing of the pins on them to the electrical connections specified in Capture. This process creates a mesh of footprints connected electrically alongside the circuit board.

Now, the components have to be placed on the circuit board. When the circuit topology doesn't not allow planar placement, double sided board design is explored. There are some considerations to make the the path length lesser as well as ones to decrease the number of vias on the board which will reduce the effect of errors in fabrication. The routing is done so as to have a wide gap between two adjacent tracks. The routing spreadsheet can be used to assign the track thickness and the via diameters. The top and bottom layers can then be selectively printed.

Certain other features of Layout have not been used here, like creating a bill of materials, and using the Layout features for design rules check, back annotation and design documentation.

Creation of PCB footprints

In layout's library manager, a library of footprints have to be created that will be used in the PCB layout. Footprints are a representation of the physical area that a part occupies on a PCB. Footprints are composed of one or more padstacks. These padstacks define how a pin on a part looks on each of the electrical and non-electrical layers. Each footprint will need at least one padstack defined. For example, if we make a padstack for a pin of one size, then it can be repeated with the right spacing to generate the footprint. There is a padstack spreadsheet on which the shape and size of the padstack can be specified. All the different kinds of padstacks required for the footprint are separately specified. As all the pins of ICs and the discrete components have the same diameter, and pin holes are the only component features that we desire on the circuit board, this is the only padstack needed.

Then, in the footprint spreadsheet, the geometrical orientation of the padstacks is specified. For example, in the case of the 555 timer, there are 8 padstacks of the same type which have to be arranged in 2 rows of 4 each. The dimensions of the ICs are sourced from drawings available at the manufacturer's website. Thus footprints are created for all the components that need to be placed on the circuit board and saved in one footprint library.

Finally, we need to place an outline around the footprint, called an obstacle which demarcates the projection of the region in space on the plane of the circuit board that the component will occupy. This is important while routing to avoid any overlapping between the parts. One final point is that of keeping to the right system of units while indicating the dimensions. Layout works by default with the English system of units, which has to be converted to metric if so desired. It is important that the board design as well as the footprint design is done in the same units, as otherwise two will obviously fail to match up.

Design of board template file

The dimensions of the board are first measured. The layout as created has to be transferred to this board on a 1:1 magnification. So, we need to fit the design on the size of the board in Layout. For this, the obstacle tool of layout is used to create an obstacle which maps the exact perimeter of the required board. Layout provides functionality with all the different layers of the board, of which can deactivate all except the top, bottom, power and ground layers. After netlisting the design, this board design will be clubbed with the Capture model and the footprints.

Netlisting and creation of board design

After the footprints are created, the next step is to assign the footprints to the parts in the schematic. Each part in the schematic in Capture has a property called PCB Footprint, which has to match one of the footprints created in the new library so as to import the design into layout. The properties spreadsheet of the design has a field for the footprint in which the name of the footprint is assigned. If there are a large number of parts, one can use the export properties function in Layout to assign all of them together. Now Layout knows how many times each footprint appears in the design, and how they are connected. To export the design to Layout, one must first create a netlist. A netlist is a file that has all the parts, footprints and nets for the design in a format that can be read by the layout program. When a netlist is imported from capture, the default widths and all other properties can be set later. In the schematic file in capture, select the Layout tab in the create netlist dialog box, as it is Layout that we are using as the companion software to Capture here.

Now, in layout, create a new board file. This attempt would prompt for the board template file and the netlist file. Then Layout combines the information in both to create the board file which is now ready for routing. Layout does not offer any help in the topology of routing, but its constraints of obstacles will help in the placement of parts. At this stage, all the component footprints, connected as it was in the schematic, is on the left side of the screen and ready to be placed on the board in the right side of the screen.

It might not always be possible (and it wasn't here) to arrange all the electrical connections in the same plane. But it is always possible to arrange them in only two planes, the two sides of the circuit board. Then the two sides are connected by means of VIAs or through-holes. At first, all the parts are placed on the board with regard to minimizing the track length between them. This might be a largely subjective judgment, and in some ways it is not worth the time to spend too many resources over optimizing part-placement as we don't have anything to lose as long as all the components are fit on the board. The unused copper is only washed away and lost.

After the parts are placed, the routing is carried out by first having the tracks move only horizontally or vertically, and all the horizontal parts being confined to one plane, and the vertical tracks to the other, which would mean having a VIA at every corner in the track. This method will work for even the most complicated design to be confined to two planes only. But, we have a plethora of VIAs in this conservative design which can be reduced by strategically rerouting some of the tracks.

It was possible to eliminate all the non-component VIAs in this design by this method. A slight concern here is the length of some of the tracks and the physical proximity of tracks carrying radically different signals. The former is a threat to circuit performance due to inter-connect capacitance, which is significant especially for the timer which is a wholly analog device. The latter will lead to inductance effects. But it is not expected to degrade the circuit a lot as the currents being carried are not too high between the devices.

Fig-4: The front and back sides of the Layout design

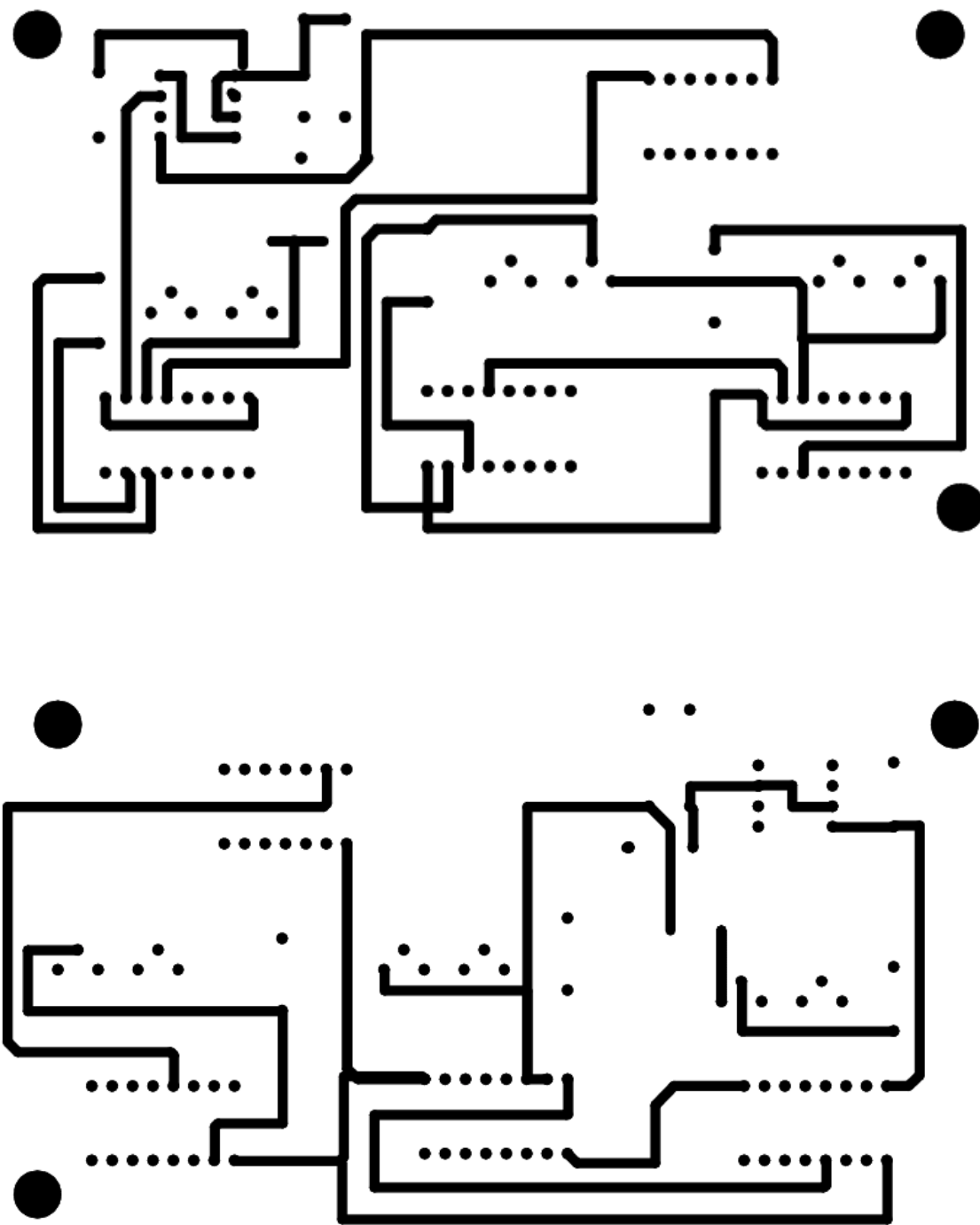
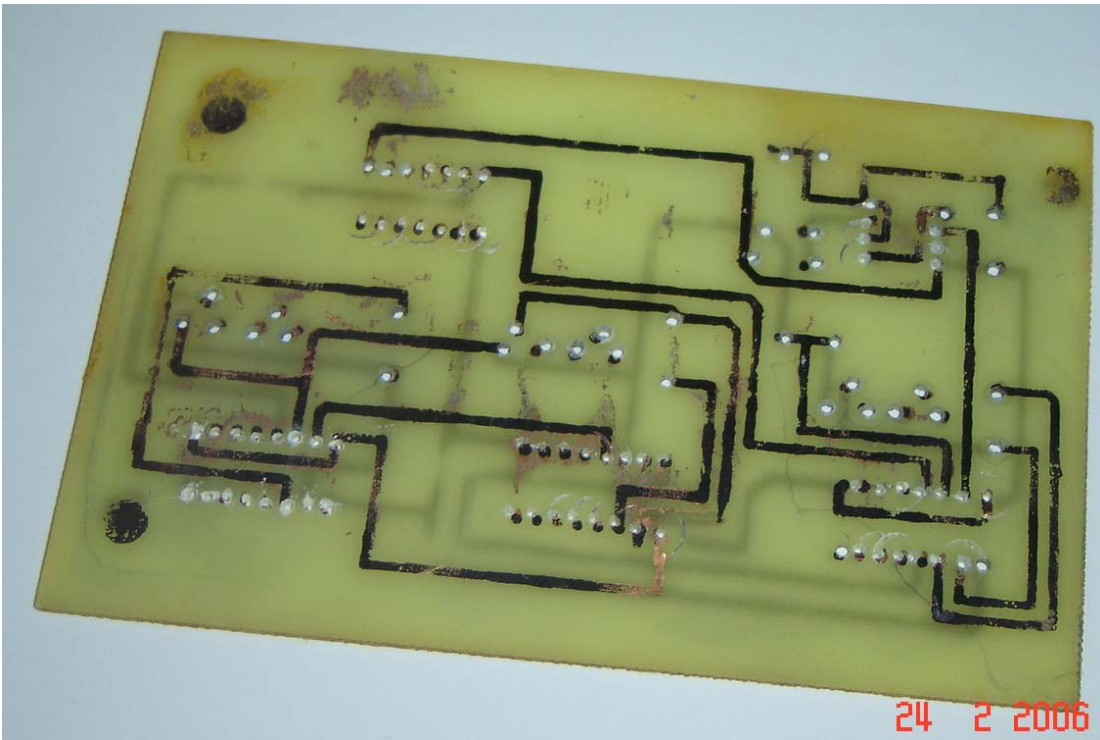
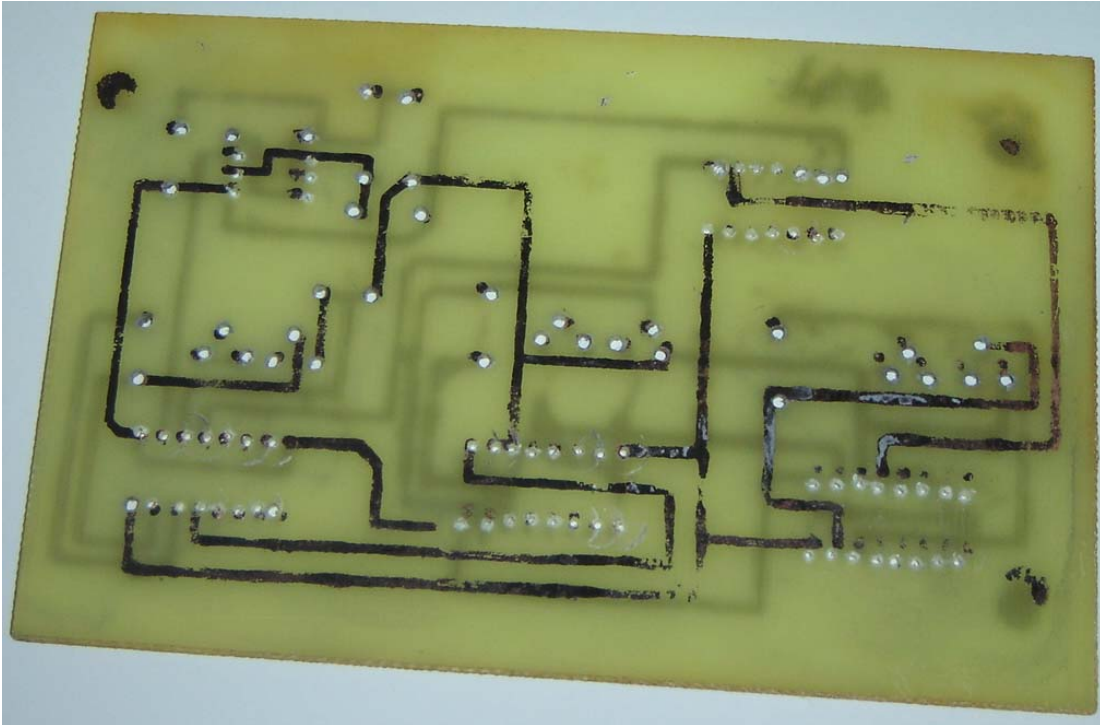


Fig-5: Camera Shots of the PCB.



Circuit board fabrication

An amateur PCB fabrication kit from Radio Shack is used to produce the circuit board from the Layout design. Both sides of the circuit board can be individually displayed on screen and printed. The images are printed onto photo-paper with black toner ink. The ink will separate from the paper at high temperatures, and we can use the separated ink as a mask while etching the circuit board.

The mask side of the photo-paper is brought into contact with one side of the board and maintained at a high temperature for about 15 minutes. This is accomplished by having an electric Iron impinging on the backside of the paper, which is hence sandwiched between the iron and the board.

After about 15 minutes, when the iron is taken off, the photo-paper would have stuck to the board. Then it is placed in a hot water bath for another 15 minutes which disintegrates the paper while leaving the toner sticking to the board. It is not desirable to manually force the process of liquidation of the paper, as this might cause the toner also to come off from the board. Now, the tracks have been marked on one side of the board by means of the black toner.

While setting up the print for the other side of the board, it is important to mirror the layout design for that side only, along a straight-line axis in the plane so that on placing this image, the back side of the board will coincide one on one with the front side. Another point of note is the effort to make sure that the latter side matches exactly with the former, when the print is placed on the board. Having markers in Layout which exist on both the sides will help, but I find that a better way is to also drill holes at a couple of the locations of the component pins that have been marked by the former side, and aligning the latter print so as to match these holes. Then the same process of combined heating and disintegration of the paper in a water bath is carried out for this side too.

Draw over the toner with resist pen, especially at places where it might not be very prominent. The toner and the resist ink will protect the copper on the board from the etchant. Now, the board is ready for etching. Place the board in about half the container of the PCB etchant solution. The solution is Ferric Chloride, which attacks the copper, removing it from the board and getting converted into cupric chloride in the process. Now, the board is almost ready. The black resist ink and the toner can be removed from the copper tracks by using the reducing solution provided with the kit. Although the copper underneath will still conduct, this is important particularly at places which have to be soldered, for maintaining conductivity.

Finally, the components of the circuit are mounted onto the board. The footprints of the various ICs fit well with the discrete components. Once all the circuit elements are connected, the board will be ready to use.

Conclusion

The circuit for the required application has been designed and a PCB fabricated for the same purpose. This course has exposed me to an actual design problem, although small, and sparked interest in device fabrication and design. Working in the lab was also a hands-on course on getting things done with the equipment available rather than seeking out the perfect answer to each problem, as I started out doing. I guess my biggest learning here has been that many of the most common tools can be used to great effect if one thinks and perseveres.