DESIGN TUTORIAL AND COMPARATIVE ANALYSIS OF PRINTED CIRCUIT BOARD PRODUCTION SOFTWARES FOR MICROCONTROLLER & FPGA-BASED SYSTEMS

by

Ryan Scott Hoover

An thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Fall 2009

DESIGN TUTORIAL AND COMPARATIVE ANALYSIS OF PRINTED CIRCUIT BOARD PRODUCTION SOFTWARES FOR MICROCONTROLLER & FPGA-BASED SYSTEMS

by

Ryan Scott Hoover

Approved: _____
Fouad Kiamilev, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Michael J. Chajes, Ph.D.
Dean of the College of Engineering

Approved: _____
Debra Hess Norris, M.S.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

To God, who saves me daily from my own demise and blesses me with the opportunities and people in my life who have allowed me to excel and achieve what would have not otherwise been possible.

To my advisor, Fouad Kiamilev, who is an invaluable resource, guide, mentor, and incredible person. I cannot thank him enough for his dedication to my educational and professional growth as well as all of the members of his research group. His attitude towards problem-solving and failure to accept impossibilities has inspired all who have ever worked with him.

To my parents, without whom my growth as an individual and the opportunities they have worked hard to give me would not have been possible.

To my fiance, Grace, who consistently lifts my spirit and encourages me throughout whatever obstacles I encounter.

To all of the members of CVORG, especially Nick Waite, Rodney McGee, Corey Lange, Rob Haislip, and Furkan Cayci. You have all harbored an exciting and unique environment to work and play that has been invaluable, not to mention awesome! You have all never hesitated to spend time sharing knowledge and time with me, and I could not have accomplished what I did at CVORG without it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Printed Circuit Board design and implementation is a research topic most any company or group manufacturing electronic devices must address. There are issues in every aspect of design, ranging from schematics to pad stacks to routing. An error in any one of these steps propagates through the remaining steps. There are many approaches and software packages for moving a design from schematics to a physical PCB. However, software packages can range from being free to costing thousands.

This thesis describes the design process from start to finish of a multi-purpose FPGA-based printed circuit board in two competing software packages. It will detail issues encountered along the way, and how these issues were resolved. Commercially available products will be used throughout the process, including the Xilinx Spartan 3E FPGA. The differing design processes in the software packages Cadence Allegro Design Entry HDL and Advanced Circuits PCB Artist will be detailed and explored in order to determine which characteristics of each package suites various types of hardware design.

The goal of this project is to create a "reference manual" for students to use as a guide in choosing a software package and working through the PCB design process at the University of Delaware's Electrical and Computer Engineering Department. These two software packages are available to all students in the department to learn and design printed circuit boards.

Chapter 1

INTRODUCTION

Printed circuit boards have been around since the 1940s. In fact, the first one was made in 1934, and they were used extensively for rugged radio designs in World War II during the 1940s. However, it wasn't until after the war that the US released the invention for commercial use. In fact, they did not become commonplace in electronic devices until the 1950s, after the US Army invented a much more streamlined method of assembly. Since then, printed circuit boards have become perhaps the most essential part to the devices that millions of people use every day. Since their invention, the process of designing a printed circuit board has become much less tedious, and there are many ways to make a finished product.

This thesis will describe two modern ways of doing PCB design with computers, and their advantages and drawbacks. It will describe the complete printed circuit board creation process, from component selection to finished product. This example will be a multipurpose field programmable gate array based board, meaning this will be the controlling component on this circuit board. A field programmable gate array, or FPGA, is an integrated circuit that is designed to include a vast array of reconfigurable hardware. This means that the user can apply different hardware configurations when necessary. While an application-specific integrated circuit is locked in once fabricated, an FPGA can be reprogrammed to implement whatever logical circuit the user wishes. This has obvious advantages, and allows for the rapid prototyping of new devices, as well as very flexible design processes.

Modern FPGAs are even capable of implementing microprocessors onboard, along with an application-specific circuit all on the same chip. This allows developers to run software alongside a hardware design created in a hardware description language such as VHDL or Verilog, and reconfigure the entire design on the fly. Tools such as the Xilinx Embedded Design Kit allow the designer to create systems with communication between hardware designs in a hardware description language and software written in a high-level computing language. This powerful combination can even be endowed with the power and functionality of an operating system, namely Linux, opening the door to a world of power that would not be otherwise possible. It is this powerful functionality that makes FPGAs perfect for this general-purpose board.

The structure of this thesis will begin with a description in Chapter 2 of the specific requirements for this particular printed circuit board. This includes walking through the component selection process, and noting design considerations and layout restrictions along the way. Next, in Chapter 3, the development process using Cadence Allegro 16.2 will be described in detail all the way to sending the PCB off for fabrication at our choice of foundry, Advanced Circuits. In the following chapter, the same process is described using another design package authored by the foundry, Advanced Circuits, called PCB Artist. Then in Chapter 5, the processes in the competing packages will be compared in detail, discussing application-specific advantages of each. The testing procedure will be described in Chapter 6, including the successes and failures of the various components. Finally, the conclusions that can be drawn from this work will be described, as well as possible future plans for this work.

Chapter 2

BOARD REQUIREMENTS

When beginning a printed circuit board project, the most important thing to do is to get a list of all the requirements, components, etc. that the final product will need to possess. The list below is the list of preliminary components that were required for this particular application, which is a custom controller for a custom testing platform.

Table 2.1: Basic Board Requirements

Required Component	Details
FPGA	approximately 150 inputs/outputs; Linux capable
Clock	50mhz speed
PROM	for program on power-up w/ JTAG interface
DDR-SDRAM	about 64MB of memory
Serial Port	DTE (DCE optional)
SD Card Slot	can use SPI interface (low volume of data)
Analog-Digital Conversion	atleast 2 analog inputs
Digital-Analog Conversion	atleast 2 analog outputs
FX-2 Connector	atleast 20 digital outputs
Switches/Buttons/LEDs	assorted I/O for physical interaction with user

2.1 Component Selection

Component selection is perhaps one of the most important steps in the printed circuit board creation process. If a mistake or error is made in this step, it can result in hours of lost work or worse yet, a useless fabricated board. Therefore, one must be sure to choose components carefully. The primary thing to look for in a part is

its ability to satisfy or exceed the requirements originally established. Obviously, if a part does not accomplish what it is required to, the board will be useless in the application it was intended for. The second thing to consider when selecting parts is its size. Small sized parts are great when it is necessary to fit many parts onto a small board. However, this may not be ideal when considering how the board is to be assembled. For example, the common 0402 sized part is very small, and can be soldered by a human. It is, however, difficult to solder by hand, and will take much longer than a larger sized part, such as an 0805 if the space is available. Another very important factor to consider when selecting components is the voltage and current they will require. It is desirable to use components throughout a design that have the same voltage, and to design a power supply system that is capable of providing not only the voltages required by the components, but also enough current for normal operation as well as instantaneous power operations.

Keeping these things in mind, the components were chosen. The most important component, the FPGA, is a Xilinx Spartan XC3S500E chip. This particular chip comes in several different sizes and packages. Some of these have more input/output pins than others, mostly because of size constraints. For the size and scale of this particular process, the PQ208 package was chosen. This chip is 28mm by 28mm, and has 208 pins with a 0.5mm pitch. These specifications are suitable for this project because they allow for an acceptable number of input/outputs while maintaining the ability to be soldered by hand. This particular model contains 500,000 gates, 158 input/outputs (65 differential), 1164 logic blocks, and a 2.7ns delay time. The FPGA itself requires a 1.2V and 2.5V supply. The device is also capable of input and output using the LVCMOS, LVTTL, HSTL, and SSTL single-ended signal standards. The input/ouputs on the FPGA are divided into banks, with 4 banks total. Each bank is powered individually, and must be configured to operate at the voltage it is powered at. The banks are organized physically as

shown in figure 2.1 below. It is important to note the number of inputs and outputs

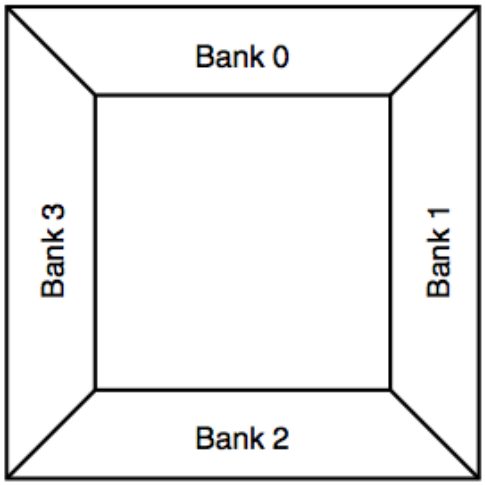

Figure 2.1: Xilinx XC3S500E PQ208 Bank Layout

available on each bank, and plan the connections to other components accordingly. If this is not done carefully, the designer may end up with usable inputs and outputs which are tied to an undesirable voltage. Furthermore, care must be given to physical arrangement of components such that locality is preserved between banks and their respective connections to components.

Another important requirement for this design is the PROM chip. Without one, the FPGA will need to be programmed every time the power is cycled. There are many different kinds of PROM chips. PROM stands for Programmable Read-Only Memory. Because this type of memory is meant to be read quickly on a regular basis, writes typically take much longer than reads. PROM chips exist for many different types of devices, not just FPGAs. Fortunately, Xilinx makes a PROM chip that is designed to work specifically with many members of the Xilinx FPGA line. For this application, the Xilinx XCF04S was chosen. The XCF04S features programability through the JTAG interface. This will allow us to arrange the PROM and the FPGA in a JTAG chain such that we will be able to program the devices independently through a single interface. Another important consideration is that

the XCF04S comes in a package which is easily hand-soldered, and takes a standard 3.3 volt supply. This particular device also happens to be of relatively low cost when compared to other available options. Another reason to use this particular device is that it is already used in combination with the XC3S500E on a commercially available board made by Digilent called the Spartan-3E starter kit board. Looking at devices which contain some of the components you wish to use is helpful because it is working proof that those components work well together. There are several other components on this board which meet the requirements listed above. The board is shown in figure 2.2 below.

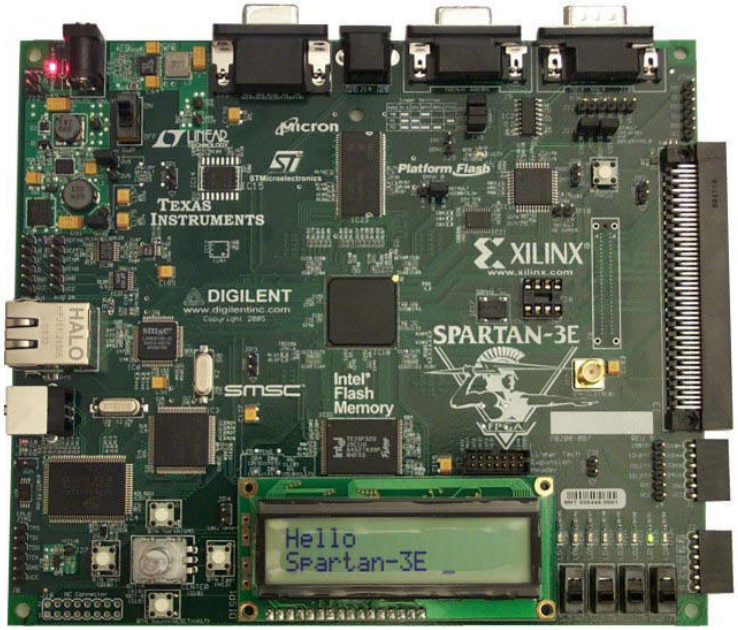

Figure 2.2: Digilent Spartan 3E Starter Kit Board

Choosing an external clock is important when working with FPGA-based devices. Without a clock, an FPGA is still very useful for simple hardware designs, but in any system that has communication or performs computation, a clock is essential. When choosing a clock speed, it is usually favorable to choose close to the fastest recommended clock speed that the FPGA is capable of. This is because

while the clock can be divided or slowed down to the desired speed, it cannot be made to change state faster. It is also important to note that our FPGA is able to handle modifications to the clock because it has a Digital Clock Manager (DCM). A designer must be sure to carefully check the recommended operating conditions for the DCM, and chose a clock accordingly. Fortunately, in this case, Digilent has selected a clock for the starter board that will work very nicely in this system. It provides a 50 Mhz clock signal, and uses a standard 3.3 volt supply. It also uses a simple footprint, and therefore is a suitable candidate for this application.

The next component that needs to be selected is a serial port. The choice of the physical serial connector is relatively easy, as there are very few choices. For this application, a standard female DB-9 connector will work just fine. These are easy to search for and find with many options, although most will have the same footprint, as it is a standard common part. However, the pins of the serial (RS-232) connector cannot be directly connected to the FPGA. The FPGA is incapable of reading and providing the correct type of signals for serial communications. Therefore, we must add another component to fulfill the requirement that this board have serial communication. Fortunately, there are many integrated circuits available to provide this functionality made specifically for serial communications. A quick search on a popular site like Digikey or Newark Electronics will yield a vast selection of components that would meet the requirements. However, since this type of component is also on the Spartan-3E starter board, it is safer and easier to use the same component as long as additional functionality is not required. In this case, that component is the ST3232CDR, a transceiver chip which can be powered with 3.3 volt supply and comes in a standard SOIC-16 package. Standard package sizes are favorable because the design footprints for these sizes are often included with printed circuit board design software, and can be reused for other IC chips. The ST3232CDR can run at data rates of up to 250 kbps and still keep RS-232 output levels, which is much

faster than is needed for this design.

Selecting the Secure Digital card socket is trivial, however deciding how it will interact with the FPGA is not. The Secure Digital standard provides operation in three modes, outlined in the table below: These modes of operation all provide

Table 2.2: Secure Digital Pinouts for 3 Modes of Operation

Pin	SD 4-Bit Mode	SD 1-Bit Mode	SPI Mode
1	DAT[3] - Data Line 3	N/C - Not Used	CS - Control Signal
2	CMD - Command Line	CMD - Command Line	DI - Data Input
3	VSS1 - Ground	VSS1 - Ground	VSS1 - Ground
4	VDD - Supply Voltage	VDD - Supply Voltage	VDD - Supply Voltage
5	CLK - Clock	CLK - Clock	SCLK - Clock
6	VSS2 - Ground	VSS2 - Ground	VSS2 - Ground
7	DAT[0] - Data Line 0	DATA - Data Line	DOUT - Data Output
8	DAT[1] - Data Line 1/Int	IRQ - Interrupt	IRQ - Interrupt
9	DAT[2] - Data Line 2/RW	RW - Read Wait	NC - Not Used

certain advantages and disadvantages. An important consideration to note is that many analog-to-digital/digital-to-analog converters also operate using the SPI standard. In fact, the analog-to-digital/digital-to-analog converters on the Spartan 3E starter kit board use the SPI standard. So, if we decided to use SPI mode for the SD card, the only signal that is unique to the SD card would be the CS, or control signal. This means less FPGA pins are used. In contrast, SPI is the slowest standard of the three, and is not suitable for transferring large amounts of data. Therefore, if the SD card is going to be used for sustained data transfer, SPI would not be a suitable operation mode. However, in this case, the requirements permit for the use of SPI mode, as there will be a low volume of data traffic to/from the SD card. In addition, this project demands as many digital input/outputs as possible, and the use of SPI mode allows for more allocation of general purpose input/output pins (GPIO).

Using SPI mode with more than one device, in our case the SD Card and

the analog-digital conversion ICs, can be configured one of two ways. In the first scenario, all SPI devices share a CLK signal, and SS (or CS - Control Signal), and Master In Slave Out (MISO) and Master Out Slave In (MOSI) are configured in a daisy-chain, as shown in the figure below. The second configuration also has a shared

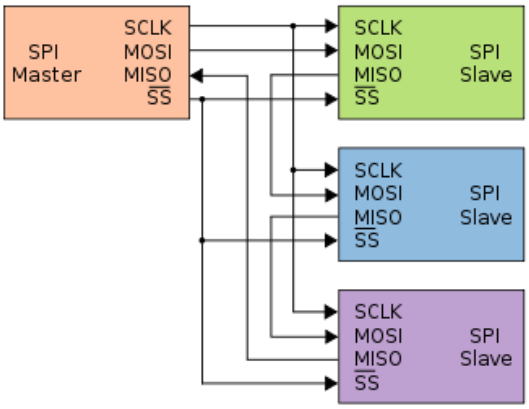

Figure 2.3: SPI Configuration with 3 Daisy-Chained Slaves

CLK signal. However, there is an independent SS (or CS - Control Signal) for each slave device. There is no daisy-chaining the MOSI and MISO signals. Instead, they are shared signals, as shown in the figure below. Since the ability to implement SPI

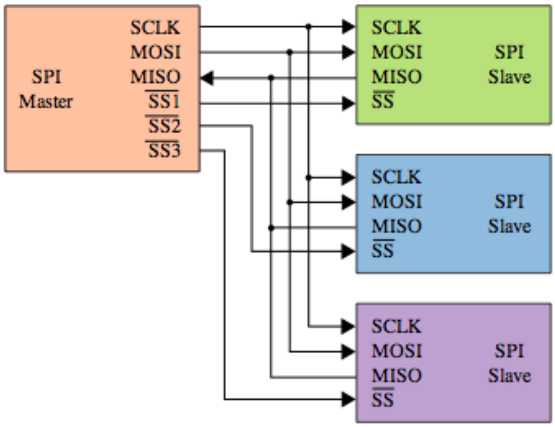

Figure 2.4: SPI Configuration with 3 Slaves

in the daisy-chained configuration requires device-compatibility, and also increases

the complexity of the system, the logical choice is to arrange the devices in a typical SPI configuration.

For the last several components that still need to be selected, it is prudent to draw knowledge again from the Spartan 3E starter board. The digital-analog converter on the starter board is a Linear LTC2624 chip, a 16-pin, 12-bit precision digital to analog converter. It can handle a 3.3V supply voltage, and can be operated at clock speeds of up to 50 megahertz. This chip will suit the requirements of this board easily, although it should be noted that there are other models of this chip with up to 16-bit precision. Likewise, the analog-digital converter used on the starter board is a Linear LTC1407 chip, a 10-pin, 12-bit precision analog to digital converter. It also handles a 3.3V supply voltage, and can sample input between -1.25V and 1.25V. For this reason, an amplifier IC is also needed to apply gain (negative or positive) to the incoming signals. The chip used on the starter board is another Linear chip, the LTC6912. This chip has 2 channels with independent gain control. It also uses an SPI interface, and takes a 3.3V supply voltage. These three chips provide four analog out channels and two analog input channels, which meet the specified requirements. The user interface components on the starter board are simple, and are large enough that a user can easily interact with them. The LEDs are bright, and about the same size as the other 0805 resisters and capacitors that will be chosen. Therefore, for this design, the same push buttons, and slider switches will be used. These are part numbers EVQ-PAE04M and 10SP001, respectively. This design will also use a similar FX-2 connector to connect to GPIO. While the starter board has a right-angled socket, this design will use a vertical socket to make it easier to connect to. The part number is FX2CA2-100S-1.27DSA, made by Hirose.

The last required component is a DDR SDRAM chip. There are many things to consider when selecting memory. Data bus width, instruction width, placement

requirements, power consumption, and size are only a few. Fortunately, the requirements for this project are simply that DDR SDRAM storage is present. Because the FPGA will be clocked at 50 megahertz, it will be under-clocking the DDR SDRAM to begin with. It is known that the DDR-SDRAM on the starter board works well, so this selection will also mirror the starter board. This chip is the MT46V32M16P-6T, a 66-pin package that has a data width of 16 bits (also capable of running with 8 bits), and is capable of running at speeds of up to 167 megahertz. There are many considerations and restrictions to consider when using DDR SDRAM that will be covered later in this thesis. As will be described in further detail later, Xilinx provides a tool called Memory Interface Generator (MIG) that will provide the correct pin configuration for the specific Xilinx FPGA that is being used. The tool will also provide information on compatibility between specific DDR SDRAM chips and the FPGA that is being used.

Once all of the major component requirements have been selected, it's important to look at the requirements of the components. Each component may require any of the additional components listed in the table below. It's important to care-

Table 2.3: Possible Component Requirements

Possible Requirement	Example
Specific Power Supply Voltage	2.5V, 3.3V
Maximum Current/Power Draw	0.7mA, 3.5μA
Bypass Capacitors	10μF, 47nF
Misc Capacitors	220nF, etc
Misc Resistors	1kΩ, 100Ω
Misc Inductors	6.8μH, 10μH
Differential Pair	+ and - signals

fully check the data sheets of the components that were selected to identify any additional parts that might be required. In the case of this project, there are many resistors, capacitors, and bypass capacitors. Most of the resistors and capacitors come in standard 0805 size. This is preferable because 0805 parts are easily hand

solderable, but are quite small and make good use of available space. Some of the required capacitors, however, do not come in 0805 packages, primarily because of size limitations. Fortunately, the remaining capacitor parts come in several other standard packages. For this project, it is most convenient to stick with surface mount parts, and tantalum capacitors offer that advantage. Therefore, the remaining capacitors are selected in 3528-21 standard size.

Now that all of the components have been selected, power and routing requirements must be taken into account. We have 4 separate voltages that are required, 1.2V, 1.25V, 2.5V, and 3.3V. There are several ways to provide these voltages. It should also be noted that it is recommended the DDR SDRAM have its own separate power supply, meaning separate 2.5V and 1.25V supplies. The chip used on the starter board to supply these voltages is a standard package and is produced by Linear, the same company that makes the ADC/DAC chips, so this chip will work nicely for this design. However, the chip used on the starter board to supply 1.2V, 2.5V, and 3.3V to the FPGA is a special Texas Instruments chip that must be baked on the board, and cannot be hand-soldered. Therefore, this makes it an undesirable choice for this project. There are not many chips on the market that will provide all three of these voltages in a single package. However, there are many regulators available that have adjustable output based on a resistor value. This lead the search to the PTH04070WAH, a Texas Instruments regulator that takes a 5V supply, and is capable of producing each of the three required voltages. A set resister is used to control the output voltage. This part comes in a 5 pin through-hole package, and will work nicely in this design. It is pictured below. Of course, since three voltages

Figure 2.5: Texas Instruments PTH04070WAH Power Regulator

are required, three voltage regulators must be used.

2.2 Design Considerations

Once all of the parts have been selected, there are factors which a designer should take into consideration before beginning the design process. The most important of these, especially when dealing with micro controllers or FPGAs is the pin mapping. This is especially important because once a PCB is fabricated, it is very difficult to re-route connections which are incompatible with the pins they are attached to. On FPGAs and micro controllers, pins have different capabilities and input/output standards, so its very important to make sure that when selecting pins for peripherals, they can support the needed input/output standards. It's also important to conserve the available resources. For instance, a standard practice is pin sharing when there aren't enough pins for all the peripherals. Of course this means that only one of the peripherals with shared pins can be effectively used at once without using multiplexing, and the pins of both peripherals must share a common input/output standard. Another way to save pin resources are to operate some peripherals with a smaller data bus. For instance, the SD card is capable of operating in both 4-bit and 1-bit data mode. If the SD card is not being used for sustained, high-volume data transfer, 1-bit mode is quite sufficient, and 3 more pins are available for other peripherals. For some devices, such as the Xilinx line of FPGAs, there is software available from the manufacturer to aid in the placement of peripheral pins. In Xilinx's PlanAhead software package, there is a pin planning tool that even has the ability to take a UCF file from their EDK, ISE tools to help in pin placement planning. This software tool is pictured below in Figure 2.6. The use of these hardware planning tools can help to eliminate errors and identify issues which may have been ordinarily overlooked.

In addition to pin planning, physical board constraints must be decided. Is this device going to be embedded into a system? How much space is available? How

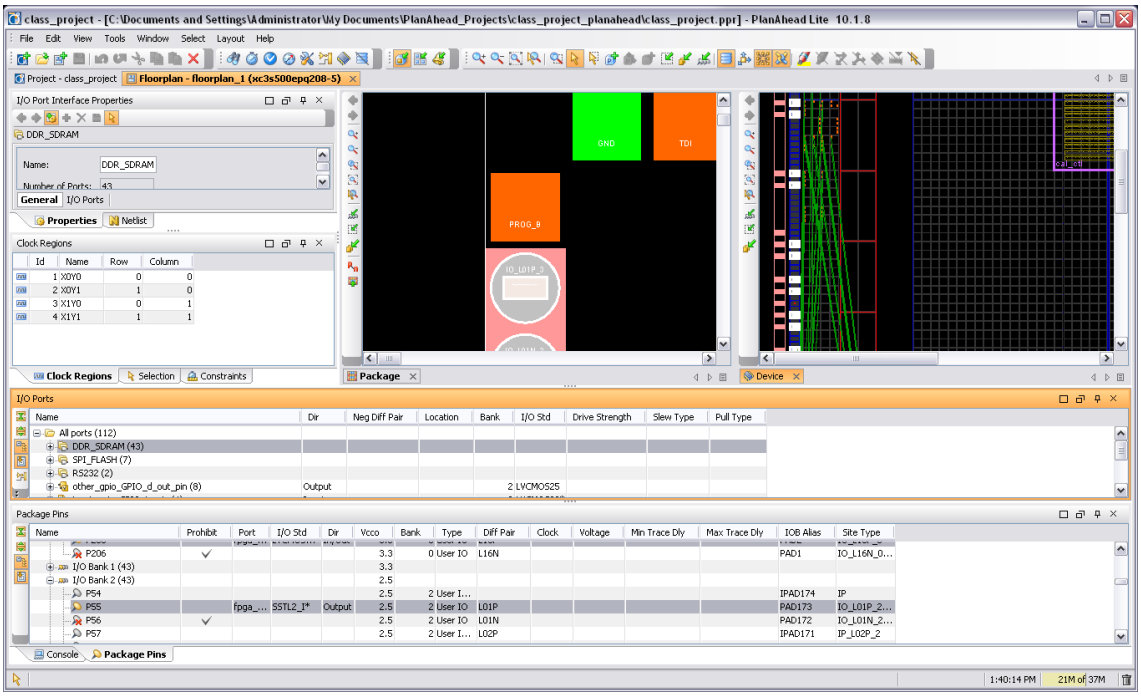

Figure 2.6: Xilinx PlanAhead 10.1.8 Pin Planning

many layers should the PCB contain? How much will it cost, and what can be done to make the cost per board as low as possible? These are all important questions to consider before beginning the design process. The further into the design process, the more difficult to make changes to constraints like PCB size/shape, and number of layers. It is also important to identify all design constraints such as differential pairs and line length limitations, such as those involved with DDR SDRAM which prevent skew and preserve signal integrity. Logging all of these constraints, and then checking them individually before sending out the design for fabrication can save tremendous amounts of time and money in the long run. For this particular design, it was decided that Advanced Circuits ($)66each.com process would be used. This requires that the design be limited to specifications listed below. It is for this reason that this board will be 6" by 5", and 4 layers. The two internal layers will be power planes. Since the ground signal is used all over the board, it is logical to make this

Table 2.4: www.66each.com PCB Specifications

Design Requirement
4-Layers, FR-4, 0.062", 1 oz cu Plate
Lead Free Solder Finish
Minimum 0.006" Line Spacing
Minimum 0.015" Hole Size
All Holes Plated
Green LPI (Liquid Photo-Imageable) Mask
White Silk Screen (1 or 2 sides)
Maximum Size of 30 Square Inches
No Slots (or overlapping drill hits)
No Internal Routing (cutouts)
No Scoring, Tab Rout, or Drilled Hole Board Separations

one of the internal layers. The three voltages are also used pretty commonly across the board, with 3.3V used the most, 2.5V the second most frequently, and 1.2 the least. Therefore, 3.3V will be the other power plane. The internal layers will be arranged as shown below in Figure 2.7.

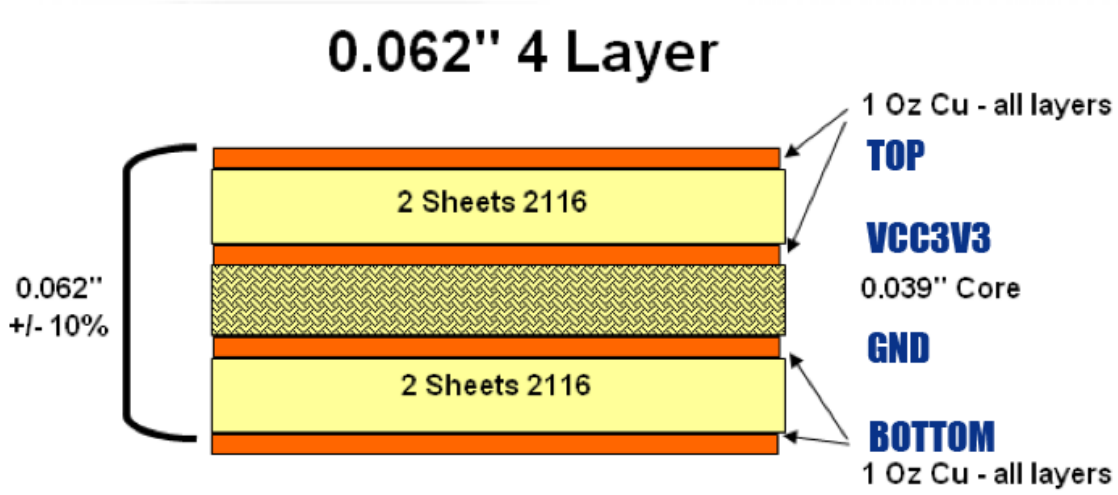

Figure 2.7: 0.062" 4-Layer PCB Stack-Up

2.3 Layout

When it comes time to start the layout of a PCB, it is important to take careful steps to insure that major changes do not have to be made at a later time. Simply thinking ahead about how different design decisions will affect future tasks, and planning ahead can save extraordinary amounts of time. For instance, if you know you are going to need 30 traces going through an area vertically, but only one horizontally, the 30 traces should get priority as to retain the original layer of routing. This is demonstrated in Figure 2.8 below. One can clearly see here that

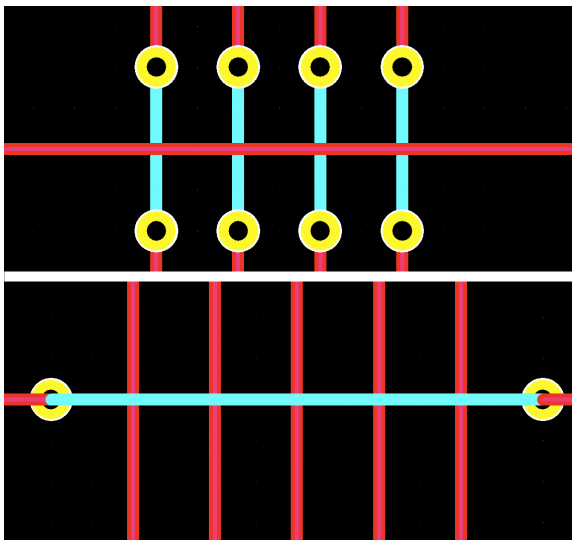

Figure 2.8: Simple Design Routing Option Comparison

the bottom routing option is better. However, if the horizontal line were routed first, the design choice on the bottom may not have been made. This is why it is important to have foresight about the way in which future traces may be arranged. In fact, the same is true for component placement. If the peripheral does not require human interaction (such as a port, button, LED, or other interface), it is generally best to place the component in a place as close to the controlling device as possible without interfering with the path of other peripherals.

Chapter 3

DESIGN PROCESS - CADENCE ALLEGRO 16.2

This section of this thesis will discuss and describe in detail the recommended design flow when using the Cadence Allegro Design Tools Version 16.2. The tutorial will cover the general process for each step from pad development to PCB fabrication.

3.1 Tutorial

In beginning to work with the Cadence Allegro Design Tools, the first step is to launch the project manager. Upon launch, a dialogue box will appear similar to the one shown below in Figure 3.1. Depending on the licenses that are available, there may more more or less selections. Unfortunately, the Allegro Suite was design

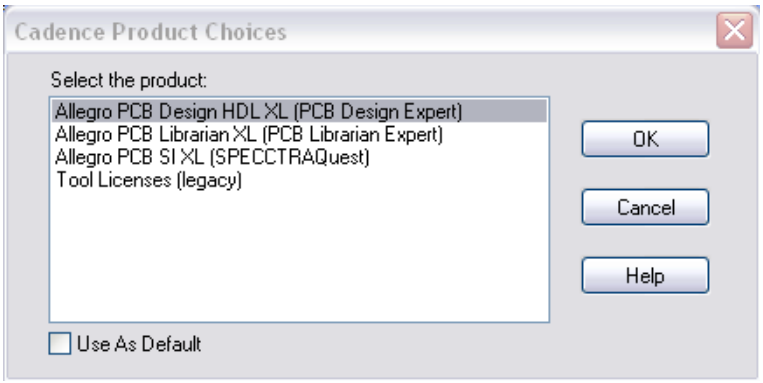

Figure 3.1: Cadence Allegro Product Choice Dialogue

for widespread use on a single project. Therefore, there are different product licenses for different functions. For instance, the Allegro PCB Librarian XL license provides

the most functionality for things like pad or part creation, while Allegro PCB Design HDL XL provides the most functionality for schematic editing and board layout. Allegro PCB SI XL is used mostly for signal integrity testing for high-speed devices. For creating a new project and getting started, choose PCB Librarian XL.

Now, the main project manager window is displayed. Here, select Create Design Project. Name the project something meaningful, and make sure that you pick a new unique folder on your hard drive that does not have any spaces in the file path. The window should now look similar to the one in Figure 3.2. Click next, and

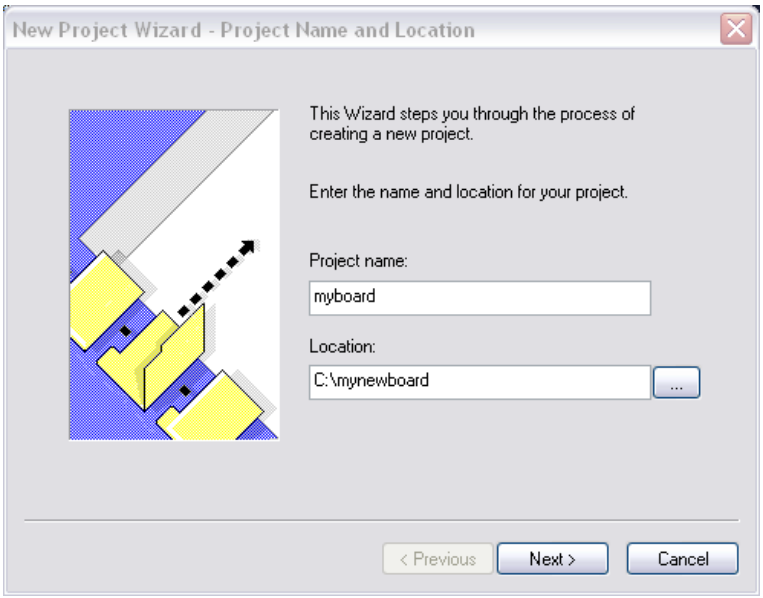

Figure 3.2: Cadence Allegro New Project Location Dialogue

move the standard library to the available libraries. The standard library contains useful items such as a schematic page layout, ground and voltage terminals, etc. In large company projects, there are often many libraries with pre-created parts. However, if it is not known who created the design/part or if it was tested, it is generally not a good idea to use it in a design. Note that there is a library automatically created based on the project name. This library will hold all of the parts unique to the new design, and in this case all of the parts used. The window

should now look similar to the one in Figure 3.3. Click next again, and now Project

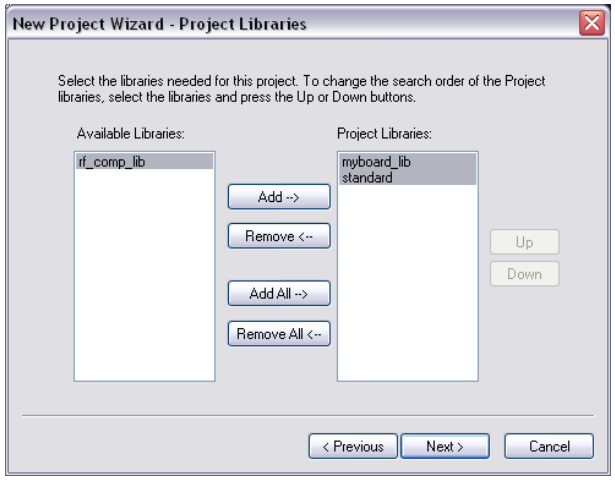

Figure 3.3: Cadence Allegro Project Library Dialogue

Manager will ask the user identify a cellname that will be the top level drawing in the design. Note that the new design library, in this case *myboard_lib*, has already been selected. Now choose a name for the top level drawing in the design. The window should now look similar to the one in Figure 3.4. Now, click next and a

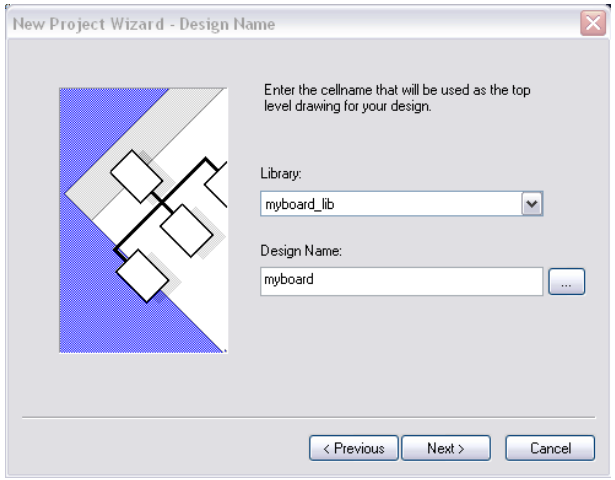

Figure 3.4: Cadence Allegro Top Level Drawing Dialogue

summary of the newly created project is presented. Look over the information and

select finish when everything looks alright. Once the project has been created, a workflow will be presented similar to the one on the right shown below in Figure 3.5. The flow on the right is the flow when the PCB Librarian XL license is being

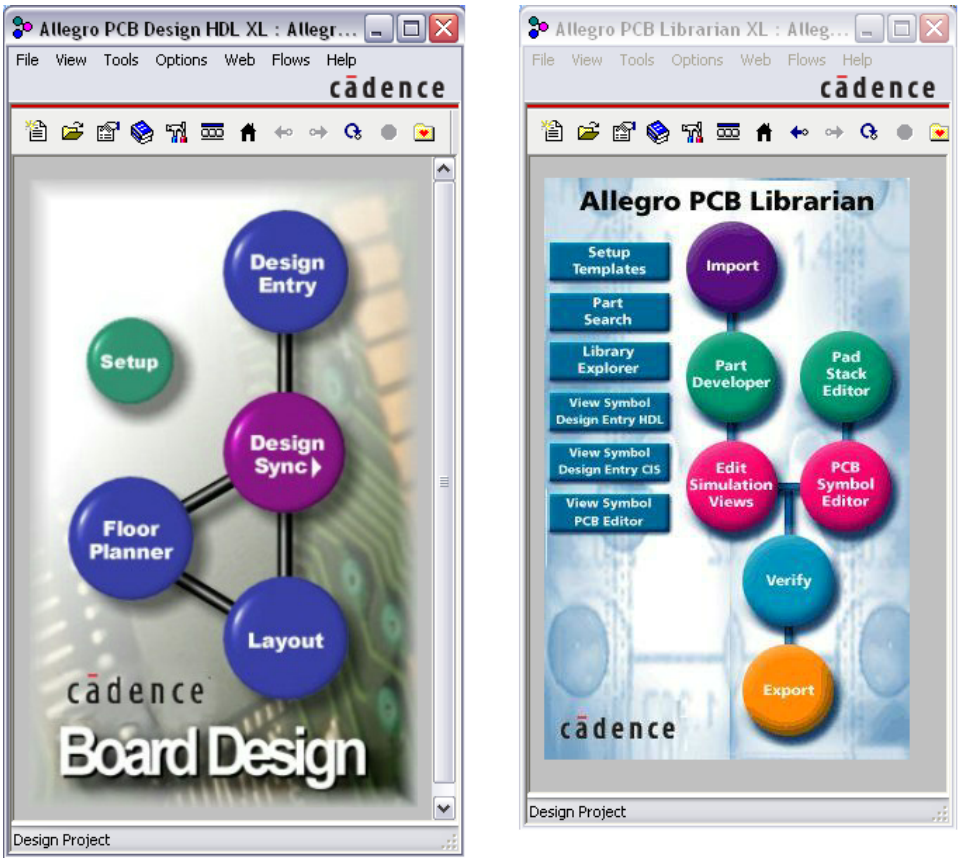

Figure 3.5: Cadence Allegro Project Manager Flow Window

used, while the flow on the left is when the PCB Design HDL XL license is being used. To switch between flows, simply click the Flows menu and select the flow you wish to see. Before getting started with part, symbol, and pad development, a few project settings need to be set. Cadence uses paths to specify where it will populate its list of available pads, symbols, and parts from. Enter the project settings by clicking Tools and then Setup. In the box that appears, click on the Tools tab. This should look similar to Figure 3.6. From here, open the PCB Editor Setup.

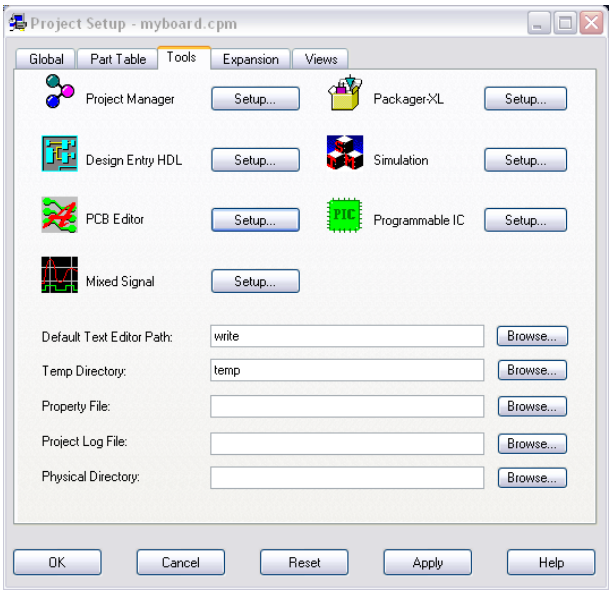

Figure 3.6: Cadence Allegro Setup Window Tools Tab

You will see here a list of categories on the left with many different things that are customizable about the software package. As mentioned before, it is necessary to modify the paths that Cadence uses to find its pads, symbols, and parts. Therefore, select the Paths category, and then the Library subcategory. In order to change where cadence looks for these files, we need to change the *psmpath* and the *padpath*. To change these default paths, click on the "..." button to the right of the title, and you will see a box similar to the one in Figure 3.7 below. Now create a folder inside the project folder and call it something like *syms&pads* to indicate that it will contain both pads and symbols. Now click the symbol to the left of the red X to create a new path, and enter the path of the previously created folder. Then click the new path and drag it to the top of the list, so that it is the first item in the directory list. This will make it the first place that Cadence checks for a referenced pad or symbol, and the one that takes precedence when there are duplicates. The same process will have to be followed for both the *psmpath* and the *padpath*. Once this is completed, simply click OK on both of the settings windows to return to the

project manager flow.

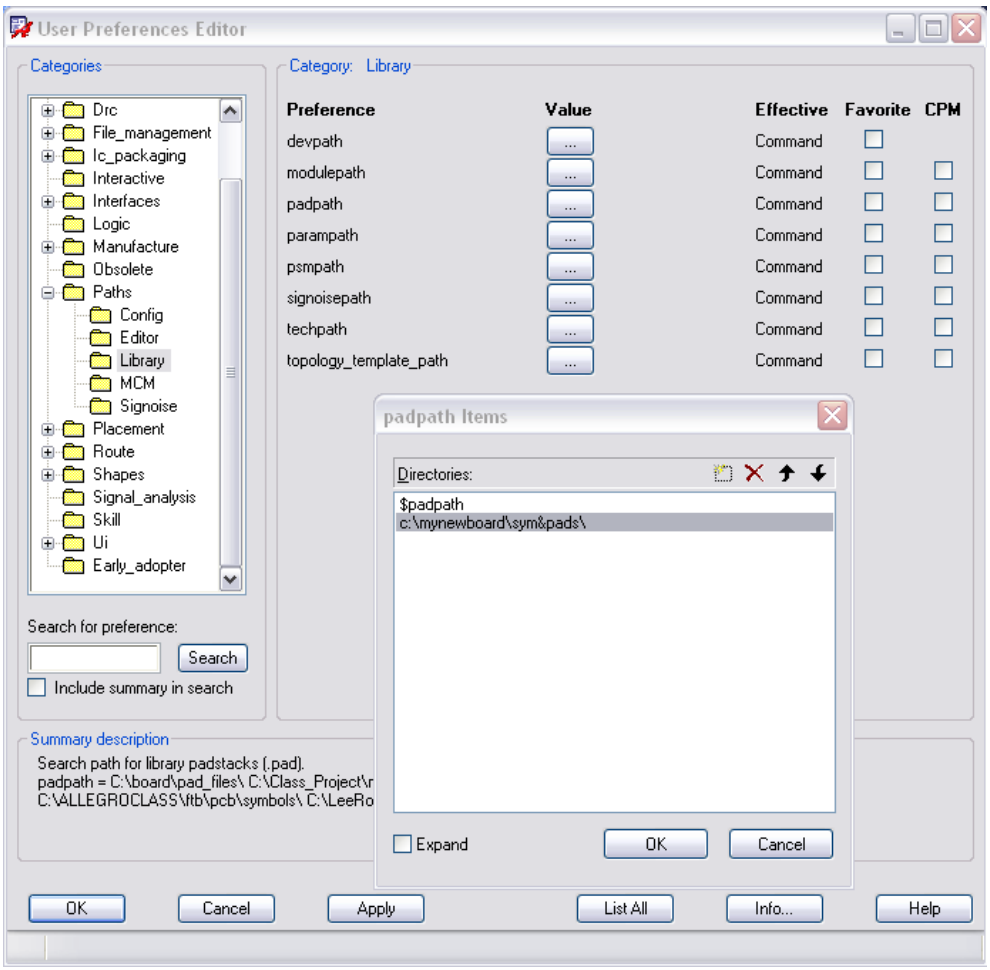

Figure 3.7: Cadence Allegro User Preferences Editor

3.1.1 Step 1 - Pad Development

The next step in the design process is to create files that describe the physical holes or pads that the components in the design will be soldered to. This process begins with the component's data sheet. This section will describe the pad file creation process for several types of solder points in an attempt to cover any type of surface mount pad or through hole that one might come across. The first example component will be the PJ102AH power connector socket. The data

sheet is as shown below for this component in Figure 3.8. The measurement key is *millimeters [inches]*. Some data sheets have more or less information. This particu-

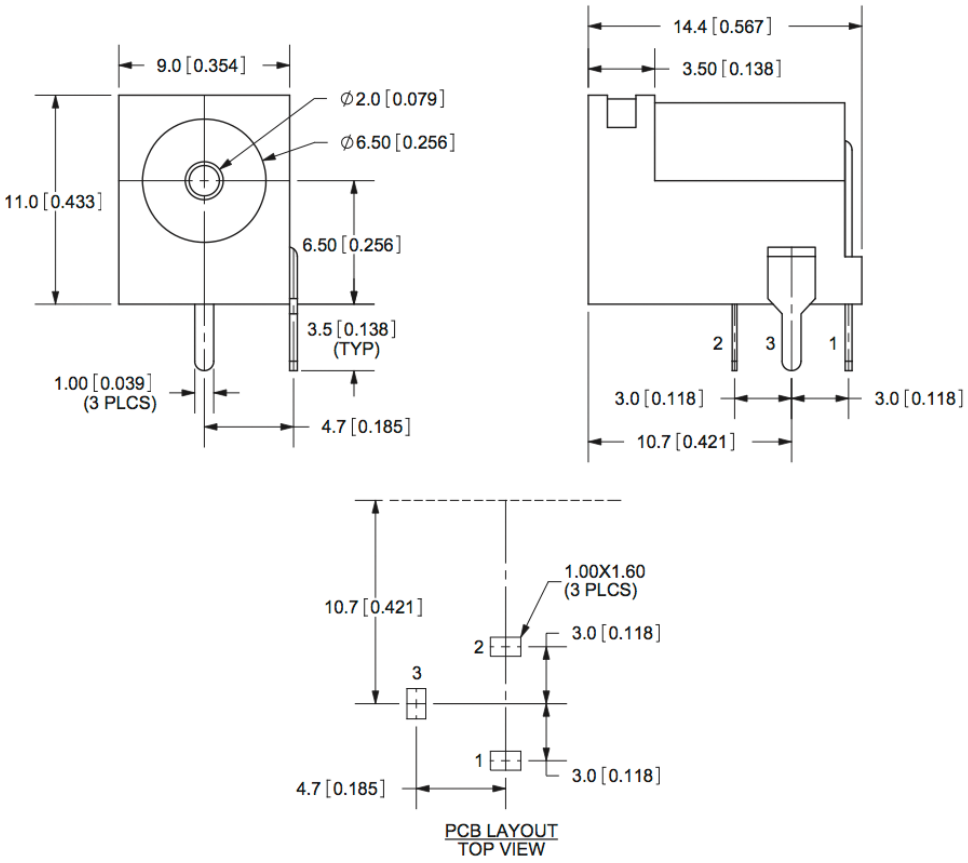

Figure 3.8: PJ102-AH Power Connector Socket Datasheet

lar data sheet contains precise measurements regarding pin placement and distance to the edge of the component. However, it is less than descriptive regarding the size of the holes. As one can see, the data sheet tells us that there are supposed to be three places where there is a 1mm X 1.6mm square hole. Unfortunately, there are not many foundries nor PCB design tools which provide for square holes, so it is best to avoid them if possible. To remedy this situation, simply use a circular hole, and make the size the largest dimension of any side of the proposed square hole. In this case, it is 1.6mm.

We begin by launching the Pad Stack Editor from the PCB Librarian flow. A prompt will appear asking for a name for the pad that will be created. Make sure that the name reflects information about the hole or pad, and information about what the pad/hole will be used for. First, set the units for this pad to millimeters to prevent having to convert from millimeters to mils. Then, If it is not already, set the hole type to 'Circle Drill', and the plating to 'Plated'. Then set the drill diameter to 1.6 millimeters. In the parameter box below, set the figure to 'Circle' and the width and height to 5 mils, or 0.127 millimeters larger than the drill diameter, totaling 1.727. Once this is completed, the window will look similar to the one in Figure 3.9.

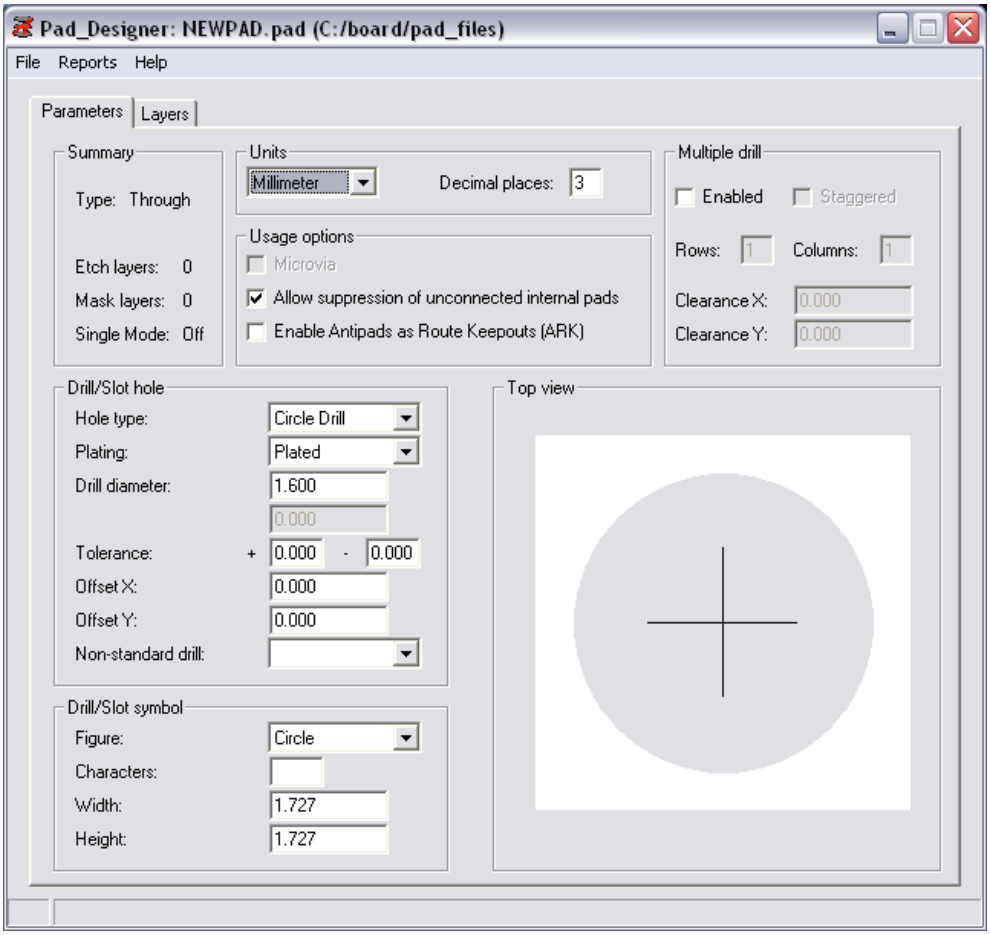

Figure 3.9: Pad Stack Editor (Parameters Tab) for PJ102-AH Pad

Next, click on the 'Layers' tab. This is where the hole's parameters will be specified for each individual layer in the PCB board. First, click on the 'Begin Layer'. Now, in the bottom half of the window, change the geometry for each type of pad from 'Null' to 'Circle'. The regular pad should be 20 mils, or 0.508 millimeters, larger than the drill hole. So the regular pad width and height are changed to 2.108 millimeters. The 'Thermal Pad' and 'Anti-Pad' should be 20 mils larger than the regular pad, a total of 40 mils, or 1.16 millimeters larger than the drill hole. So now the width and height of the 'Thermal Pad' and 'Anti-Pad' are set to 2.760 millimeters.

Now that we have specified the parameters for the 'Begin Layer', we can copy those parameters to other layers which should have the same parameters. In this case, the internal and end layers should have the same parameters. To copy and paste these parameters, simply right-click on the column to the left of the layer name, and select 'Copy'. Then, do the same on the row you wish to move the information to, and select 'Paste'. The last two layers that need to be defined are 'SOLDERMASK_TOP' and 'SOLDERMASK_BOTTOM'. These layers need to be about 25 mils, or 0.635 millimeters, larger than the drill hole, and only the 'Regular Pad' should have parameters defined for it. Now, change the Geometry for SOLDERMASK_TOP to 'Circle', and set the width and height to 2.235. Now, copy and paste this configuration to SOLDERMASK_BOTTOM in the same manner as the previous settings. The window should now look similar to the one in Figure 3.10.

A similar procedure will be used to create the pads for another component, the Fox $50mhz$ Oscillator. Unlike the power connector socket, the oscillator is a surface mount part, meaning that it has no through/drill holes. Instead, the part is soldered one of the surfaces of the board. This allows for easier routing, although is generally slightly more difficult to solder. First, it is necessary to take a look at the data sheet, as many surface mount parts will sometimes have multiple different

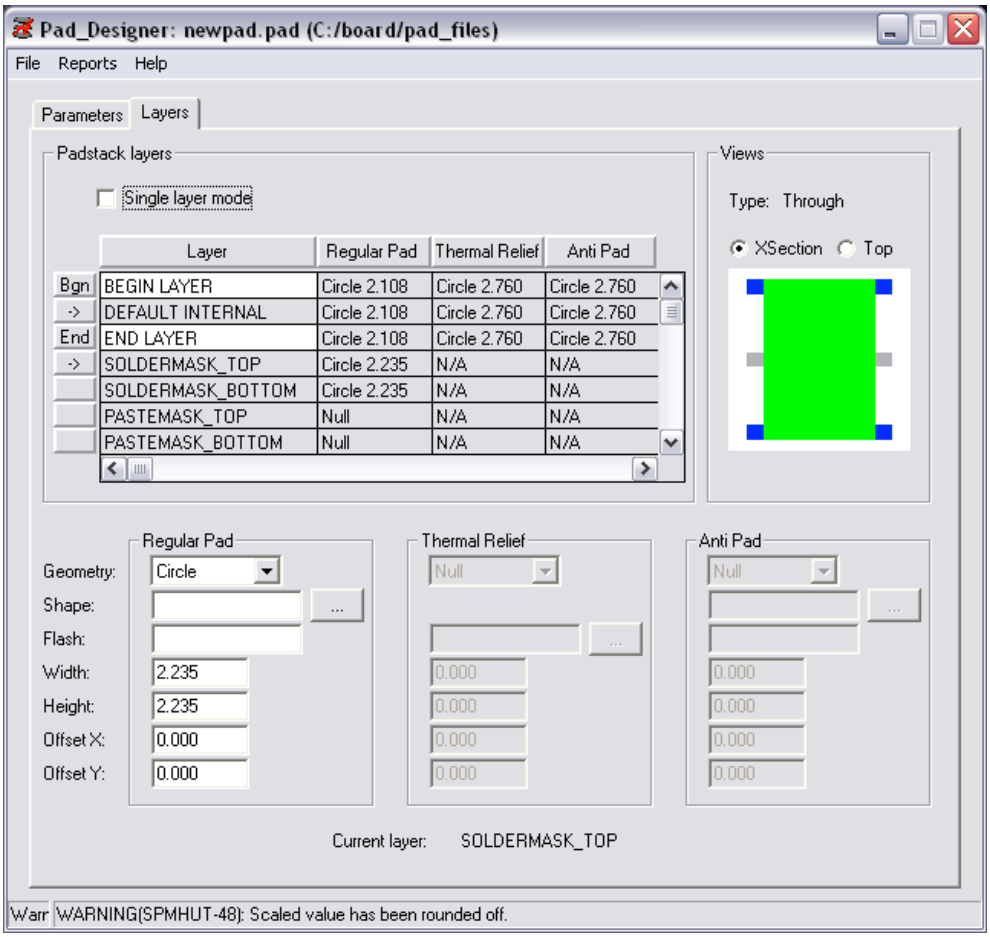

Figure 3.10: Pad Stack Editor (Layers Tab) for PJ102-AH Pad

sized pads. The data sheet for this part is shown in Figure 3.11. Fortunately, all of the pads in the recommended solder pad layout have the same size, 1.8 mm X 2.0 mm. The process begins in a similar manner as before, by starting Pad Stack Editor from the PCB Librarian flow. This time, however, only the units need be changed on the 'Parameters' tab in the Pad Stack Editor. Simply change the 'Units' from 'Mils' to 'Millimeter'. Now click on the the 'Layers' tab. Since we will only be dealing with a surface layer, click the check box to turn on 'Single layer mode'. This will change the layer options to those suitable for describing a surface mount pad. First, click on the 'Begin Layer', and change the 'Regular Pad' geometry from 'Null'

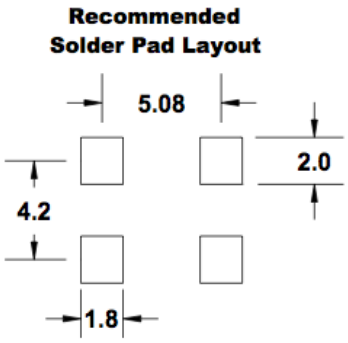

Figure 3.11: Fox 50mhz Oscillator Data Sheet

to 'Rectangle'. Then enter the recommended solder pad layout dimensions above into the width and height boxes. This defines the actual pad that will be soldered to.

Now, the 'PASTEMASK_TOP' field should have the same parameters as the 'Begin Layer'. Therefore, use the same method as above to copy and paste the parameters. The paste mask is a layer which can be used to create solder paste screens for use in hot re-flow soldering systems. The last layer that needs to be defined for the surface mount pad is the 'SOLDERMASK_TOP' layer. The solder mask layer is the protective coating found over everything except vias and pads. Defining a shape in this layer means there will be no protective coating placed over this region. For this reason, it is necessary to make this area slightly bigger than the pad itself. Generally this area is 10 mils, or 0.254 millimeters, larger than the pad. So, in the 'SOLDERMASK_TOP' layer, set the geometry to 'Rectangle' and the dimensions to 2.054 mm X 2.254 mm. When this is completed, save the pad file. The window should look similar to the one in Figure 3.12. There are many other types of pads and through holes that can be created, but what is described here are the most common types.

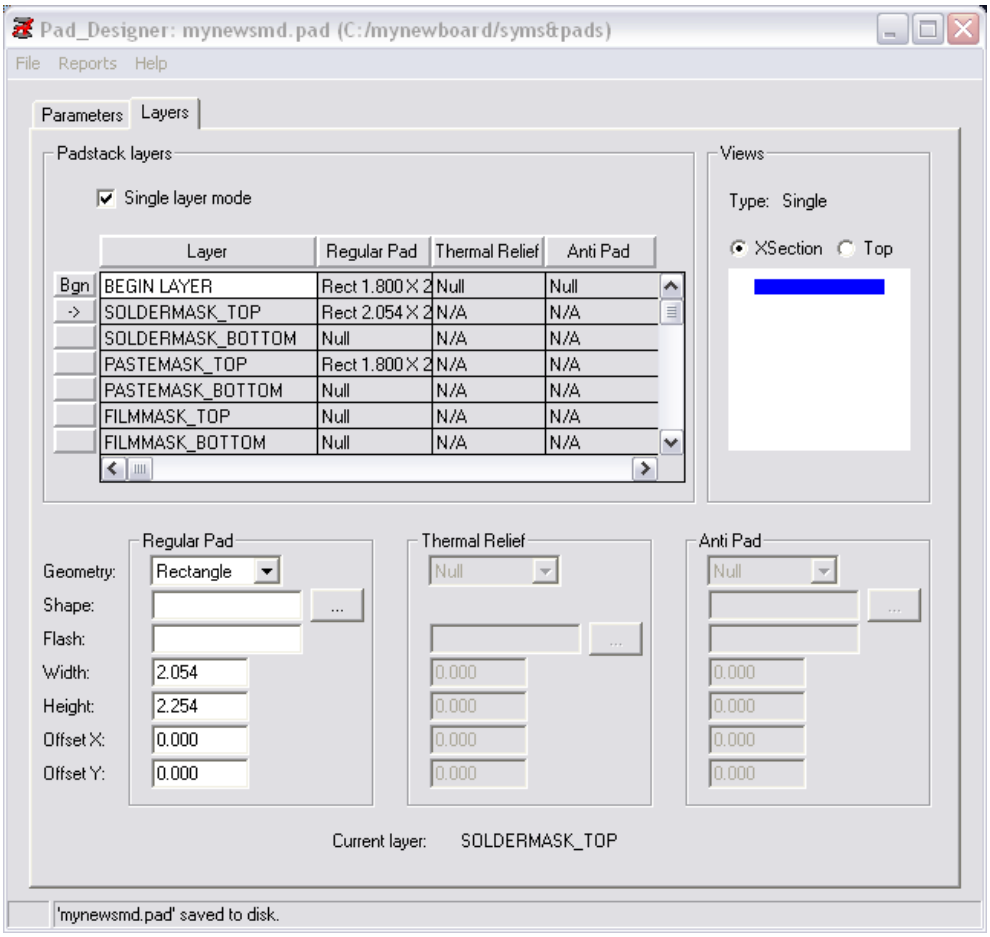

Figure 3.12: Pad Stack Editor (Layers Tab) for Fox 50mhz Oscillator

3.1.2 Step 2 - PCB Symbol Development

The next step in the PCB creation process is to take the pads we just created and arrange them in the patterns described in the data sheets. The first step to accomplishing this is to plot out the location of the centers of each pin/pad in relation to the either the origin or the bottom leftmost corner of the footprint. Note that if the part only has 2 pins/pads or is a connector with a uniform pin/pad layout, a wizard can be used with only a few standard dimension measurements to produce the desired pin/pad layout. Cadence uses a standard cartesian coordinate system for object placement, so plotting out all of the coordinates ahead of time is

quite useful. As an example, the PJ102-AH Power Socket will be used.

Now, start the PCB Symbol Editor from the PCB Librarian flow. Click *File - New* from the menu. This will bring up a new file menu. Select 'Package Symbol' as the drawing type, and type in a name for the new symbol. Click the OK button, and a blank new package symbol appears. Before beginning pin/pad placement, it is helpful to change a few settings. First, click *Setup - Design Parameters*. In the first tab, the 'Display' tab, there are several options that can be very beneficial when turned on. The grids, for example, are especially helpful when placing components by hand. To use grids, first check the 'Grids On' box, and then click the '...' button next to the Setup Grids label. This window allows you to change the distance between the gridlines, which the cursor will snap to when placing objects by hand. It is very useful to change the grids during the design process. Note that right now, the measurements are set to mils. Click OK to return to the 'Design Parameters' window. Now go to the 'Display' tab. Here, the units used in the design can be changed. Since the measurements in the data sheet are in millimeters, it will be most convenient to use millimeters in the symbol development process as well. In this tab, the extents of the working canvas can be changed, as well as the origin of the project. Note that values entered in the 'Move Origin' box denote the change in the X or Y direction, and not the new origin point. This window contains other tabs which allow for the adjustment of many default values that can help make the experience feel more streamlined. Adjust the settings as necessary, and click OK to return to the main window.

The next step to creating a PCB symbol is to place the pins. This is where the coordinates come in very handy. It is also important to place the pins in the order that they are numbered; for example, place pin one first, pin two second, etcetera. This will save time in having to go back and change the pin numbers later. To instantiate a pin, click *Layout - Pins*, and a sidebar menu should appear on the

righthand side of the window that contains options for pin placement. Click the '...' button to the left of the 'Padstack' field and select the pad you want to place. Below this, there are many useful options for placing a large number of pins spaced equidistantly from one another, rotating a pad, numbering those pads, and offsetting them from each other. This set of tools proves invaluable in instances where a large number of pins or pads are to be placed. However, there is no need to change any of these settings in this case. Now it is time to place pin 1. This can be done simply using the mouse, but this is highly inaccurate unless measurements are in all whole numbers, something that rarely occurs. It is best to enter the coordinates manually. Simply click at the bottom of the tool in the command window and type 'x x-coord y-coord'. For example, if I wished to place the first pin at 'X=1000 Y=1000', i would type 'x 1000 1000', as shown in Figure 3.13. Simply repeat this process until

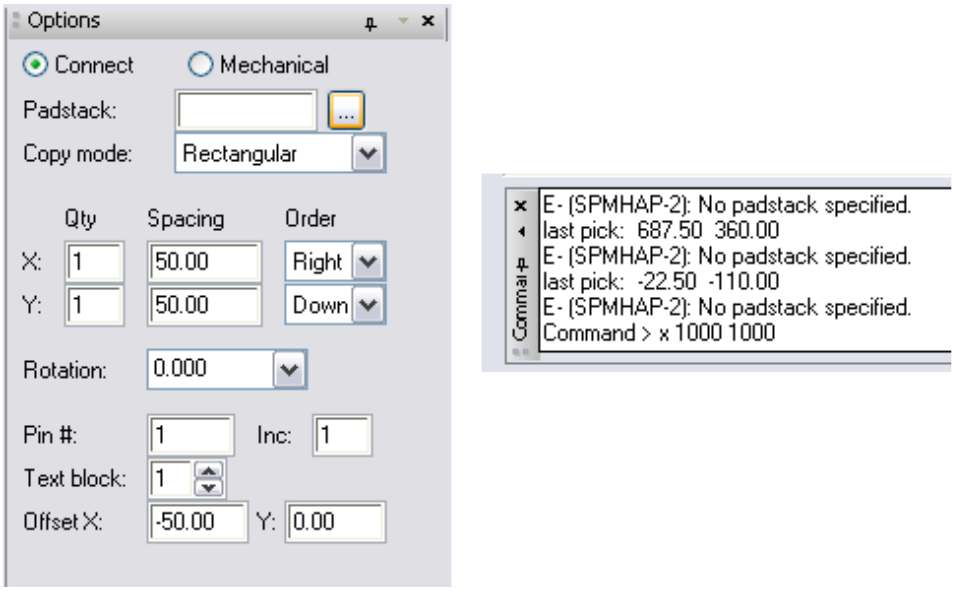

Figure 3.13: Allegro PCB Editor (Package Symbol) Pin Placement Procedure

all pins/pads are placed where they are supposed to be. When pin placement is complete, right-click an unpopulated area on the canvas and select 'Done'. When

the tool is an a "mode" such as 'Pin Placement Mode', it will stay in that mode until 'Done' is selected.

At this point, the pins are in place, and this symbol could technically be used on a PCB. However, there is no indication at this point in time of where the physical part will extend to and cover. Without some kind of guideline, a designer could easily place other components, such as surface mount resistors, in an area which will interfere with the component. To prevent this, a package geometry line will now be created. To do this, click *Add - Line*. This will bring up a specific options menu on the righthand side of the window. In this menu, select 'Package Geometry' as the active class, and 'Assembly_Top' as the subclass. The line width and font should already be set to 0 and solid, respectively. The line lock is useful for drawing lines with the mouse. This feature allows the possible angles to be locked such that lines and arcs can only be drawn using the specified angle. This is not necessary in this case, as the component is square, and we will be specifying the vertices using the coordinate specifier. The options window should look similar to Figure 3.14. Now, enter the coordinates one by one, ending on the starting

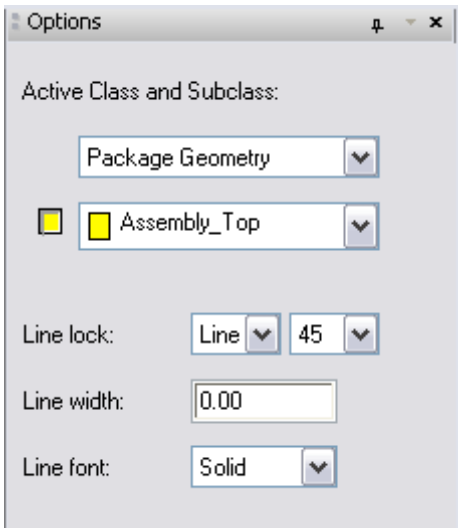

Figure 3.14: Allegro PCB Editor (Package Symbol) Line Options

coordinates. Right-click on an unpopulated area of the canvas and select 'Done' when finished.

Some designers also like to create a silkscreen outline of the part, however some do not. This next step, therefore, is not necessary. If a silkscreen outline is desired, begin bringing up the line options menu by following the same procedure as for the package geometry line. This time, however, select 'Silkscreen_Top' as the subclass, keeping 'Package Geometry' as the active class. Now, enter the coordinates or use the mouse to draw another box around the component. The box can be the exact dimensions and coordinates as the package geometry line. Now that there a visual indicator of the package size, as well as an optional silkscreen indicator, it is also necessary to provide Allegro with a package boundary. A package boundary defines the area in which no other components may be placed. This is useful in parts such as the PJ102-AH power connector, because some of the pins extend out past the actual package geometry, and it is preferable to keep neighboring components a safe distance away. To define this boundary, click *Setup - Areas - Package Boundary*. As before, a special options window appears. In this case, however, it is not necessary to set the options, as they are already correctly configured. Simply draw a boundary box using the mouse, or by entering coordinates.

The next step is to place labels around and inside the symbol. The first label to place is the reference designator. This label will go inside of the package geometry, and it is for the reference of the designer only; it will not appear on the fabricated board. Place this label by clicking *Layout - Labels - RefDes*. As before, an panel will appear on the right side of the window containing various text options. It can be useful to change some of these options before clicking the point at which to place the text, such as center justification. When options have been set, click where the start point of the text should go. If the text is not placed correctly the first time, it can be changed later. This label should designate the type of component or device,

as well as the instance number. Because this is a power connector, 'P' will be used to precede the instance number. In Allegro, the wildcard '∗' is used to denote instance number, and will be replaced by the tool later. Therefore, type 'P∗' as the text for this label. When finished it should look similar to the one shown in Figure 3.15. Another label that is helpful to the designer is the device type label. To set this label, click *Layout - Labels - Device*, and follow the same procedure as above to place the label inside the package geometry. This label should contain something about the package type, if it is a standard package type, such as the 0805 or SSOP16. In this case, the type is unique to the device, so something such as 'DC_PWR_CONN' will do nicely. The last label will be visible on the end product, and is also a reference designator. This label will prove very useful when placing components, as often there are many different parts that need to be soldered to the same footprint type. To place this final label, click *Layout - Labels - RefDes*. In the options panel that appears on the right side of the window, select 'SILKSCREEN_TOP' as the subclass. Now place a label outside of the device, and give it the same designation as the first reference designator, 'P∗'. Rotate the text as necessary and when finished, exit label placement mode as previously described. The completed labels should look similar to those below in Figure 3.15.

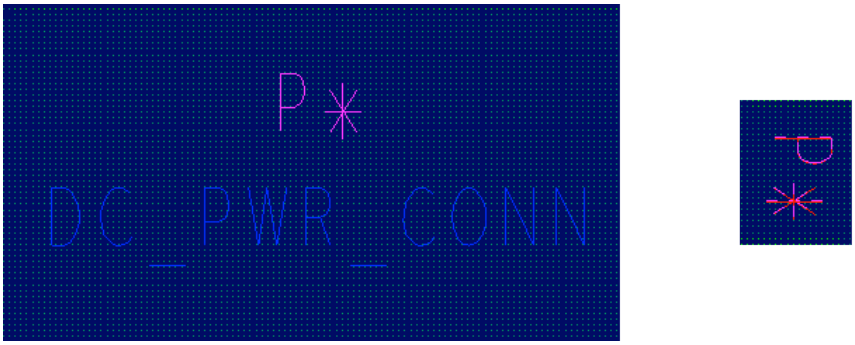

Figure 3.15: Allegro PCB Editor (Package Symbol) PJ102-AH Completed Labels

The completed symbol should look similar to that of Figure 3.16. Note that

the colors can be changed to the designer's liking by clicking *Display - Colors/Visibility*. Simply save the design (ensure that it is in the project's work library OR your symbols and pads folder so that Allegro will search there) and continue on to part development.

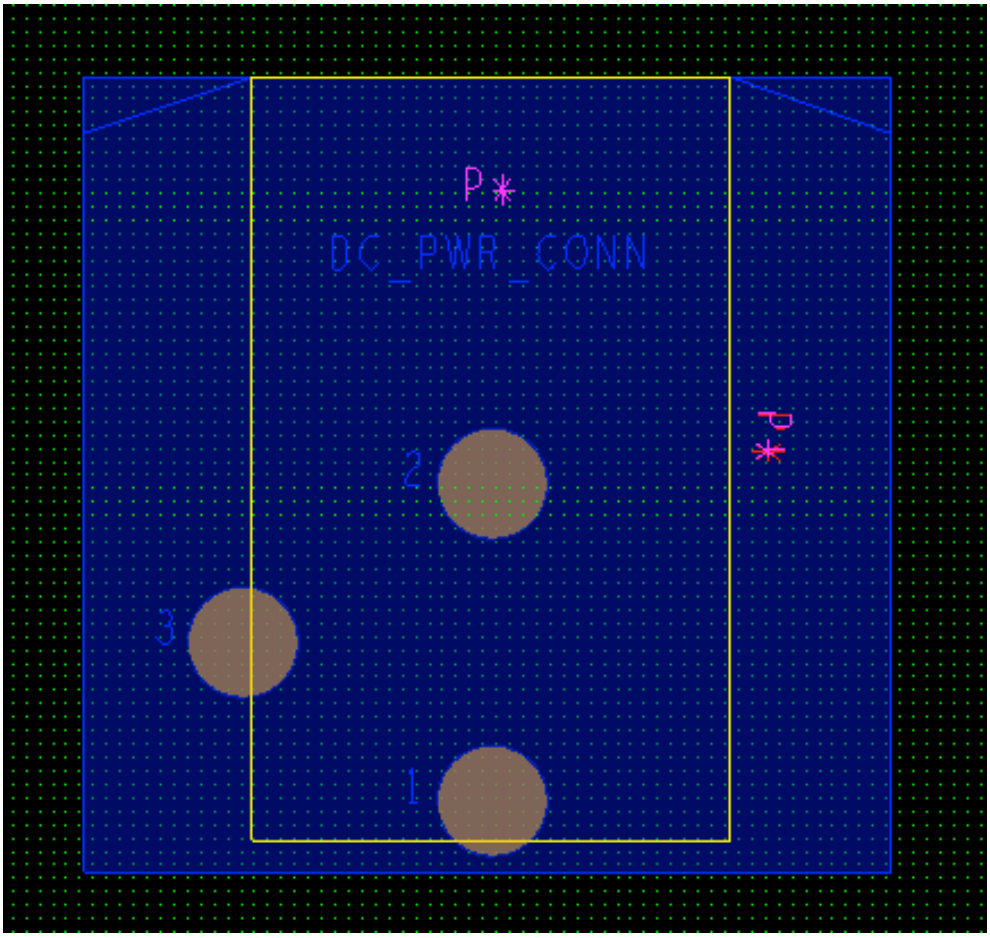

Figure 3.16: Allegro PCB Editor (Package Symbol) PJ102-AH Completed Symbol

3.1.3 Step 3 - Part Development

Now that the symbol for the part has been created, the next step is to create a complete part. When completed this part entity will contain all of the pin names that describe what each pin on the device is used for, as well as a symbol that

can be instantiated in schematic editor in order to describe the connections to other components that will be routed in the final design. Begin by starting Part Developer from the PCB Librarian XL flow. This will bring up the utility with an empty screen. To begin creating a part, select *File - Import and Export*. A dialogue will pop up, asking you to select an action to perform. Select 'Import Cadence Footprint'. In the flow, as well as in the previous flow, the 'Footprint' to which Allegro is referring was called a 'Package Symbol'. They are the same thing, and most often in other programs and documentation, it is referred to as a 'Footprint'. For the purposes of this tutorial, the PJ102-AH footprint created in the previous step, as well as the footprint for the Fox 50mhz oscillator, are used as a demonstration. Once the footpring has been selected, click the next button, noting that Allegro is capable of importing from a great number of sources to create a part file that can be used in the tools. Select the symbol, or footprint, that was created in the previous step. An example of this is shown in Figure 3.17. Click Next, and enter the name of the

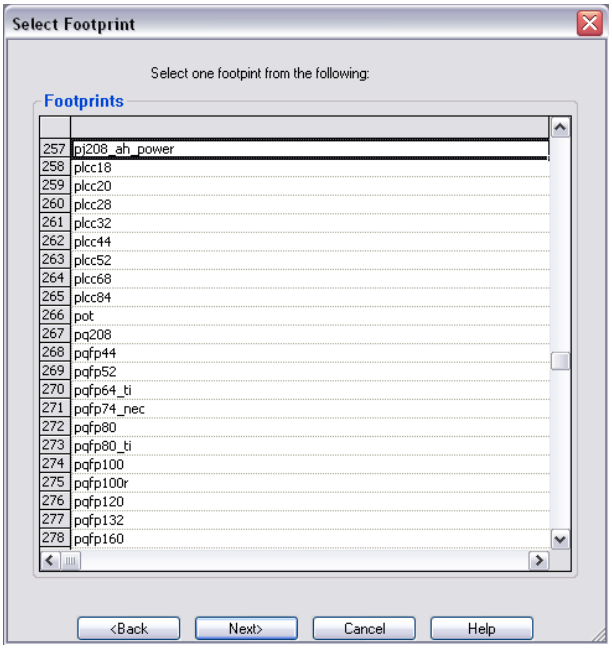

Figure 3.17: Allegro Part Developer - Importer Footprint Selection

part, or use the same name as the footprint. The project library should be set by default as the destination library. Click next again, and there is a preview of the pins that will be imported. It is recommended, but not required, that the type of each pin as well as the name be entered at this time. There is also the opportunity now to specify global pins. Global pins are those which do not need to be specified explicitly in the schematic phase, but are recognized as nets and will be eventually routed as such. The common use for global pins is for pins which will be connected to power planes or power nets, as well as ground nets. This can be done in this step of the import wizard. Simply right click inside the global pins box, and choose 'Insert Row After' one time for each signal that needs to be global. Specify the net name, type, and the pins which are to be connected to that net. The original entries for those pins must be deleted from the 'Logical Pins' section so that the pins are not mapped twice. Because the PJ102-AH is a power socket, all of its pins are global. This window for the Fox 50mhz oscillator is also shown below in Figure 3.18 for reference. Once this is completed, the window for either of these two parts should

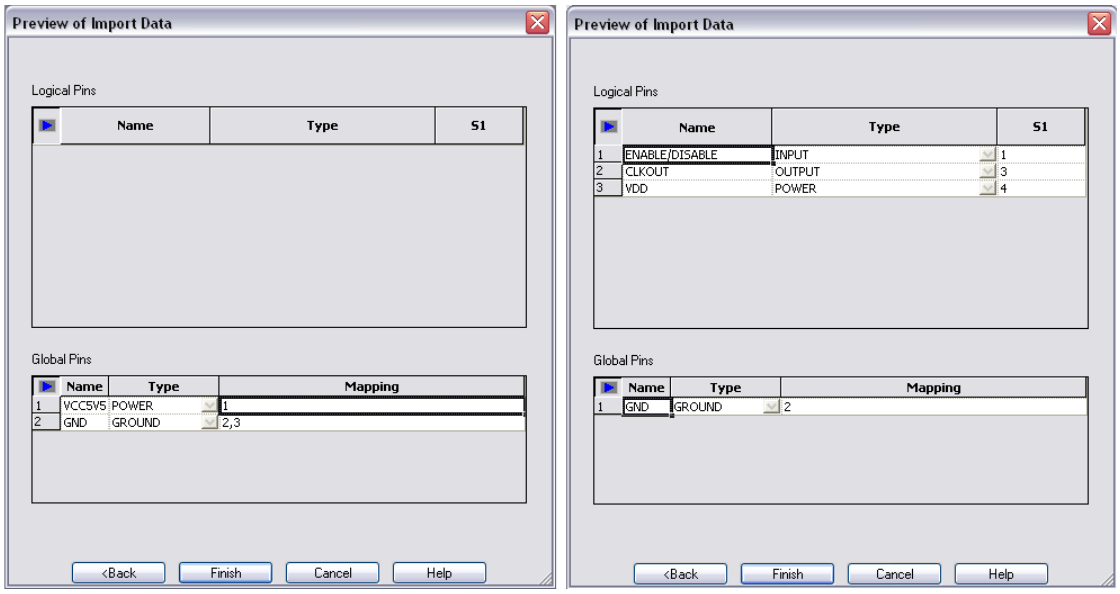

Figure 3.18: Allegro Part Developer - Importer Pin Assignment

look similar to Figure 3.18. Not that the window for the Fox 50mhz oscillator has 'VDD' in the 'Logical Pins' area. This is because while the pin will be connected to the 'VCC3V3' net, it has a bypass capacitor connected to it. If in the schematic stage the bypass capacitor is connected directly to this net on the symbol, Allegro will make this connection explicit, and will display a rat to help during component placement. Now, click the 'Finish' button to commit these settings.

The screen for the new part should appear. The part developer tool provides many opportunities to enter design-specific constraints that can aid invaluably in the prevention of errors throughout the design process. It is now necessary to generate schematic symbols for the part we have just created. In larger designs,

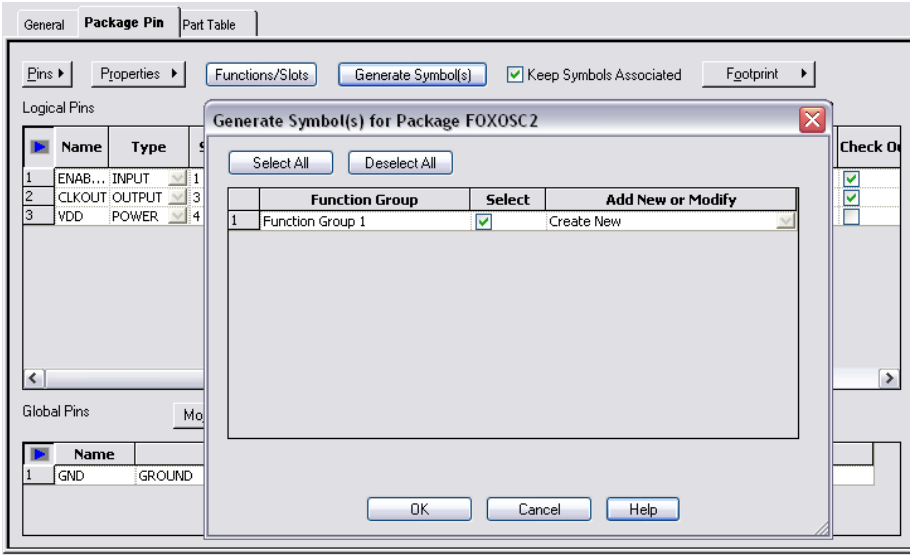

Figure 3.19: Allegro Part Developer - Generate Symbol(s) Dialogue

where there are hundreds of pins, it is desirable to split the pins across multiple schematic symbols. This can be accomplished by clicking the 'Functions/Slots' button. A new window will appear that has a list of the function groups that exist currently. To add function groups, simply click the add button as many times as is necessary. When the desired amount of function groups appears in the window, click the 'Distribute Pins' button. This will bring up yet another window that

allows the distribution of the pins for a particular part across the function groups that were just specified. To move several pins at once to a particular function group, highlight the rows, and then right click and hit 'Move to...'. Once the pins have been distributed, click OK to return to the original window. In this case distributing the pins is not necessary, as both of the parts have less than five pins. To generate a symbol, click the 'Generate Symbol(s)' button. This should produce a dialogue as shown in Figure 3.19. This produces a basic schematic symbol from the data provided during the import wizard. To view the symbol generated, click 'sym_1' in the hierarchy on the left. The symbol should look something like the one shown in Figure 3.20, very disorganized, confusing an rudimentary. Depending on the configuration, a generated symbol may look acceptable. However in many cases, such as this one, the generated symbol needs modification before it is used. Using the provided interface, modify the symbol so that all of the pin labels are

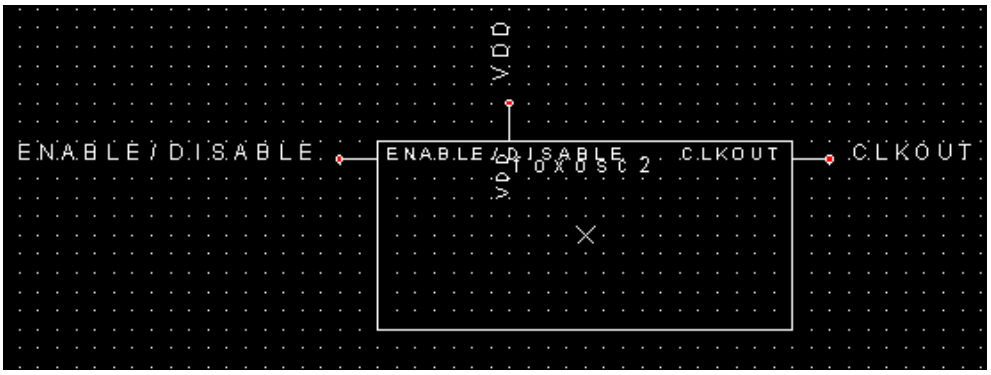

Figure 3.20: Allegro Part Developer - Initial Generated Symbol

readable, and the symbol description is not obstructing the pin labels. This can be accomplished by changing the pin positions, label positions, text size, pin locations, and the symbol outline. When this has been completed, the symbol should look clean, and the pins easy to identify and logically ordered. An example of a cleaned up rendition of the symbol shown above is in Figure 3.21.

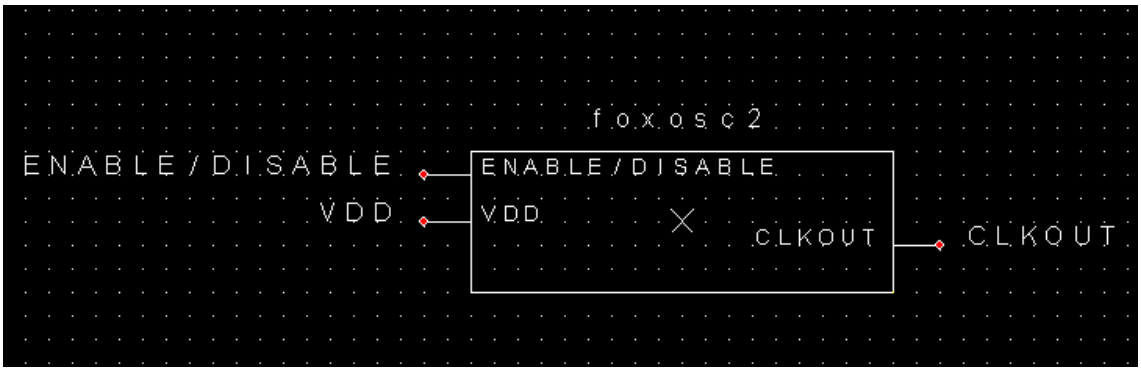

Figure 3.21: Allegro Part Developer - Modified Generated Symbol

For this particular part, the part creation process is finished. Simply click *File - Save* to write the file to the project's working folder. However for parts such as a resistor or capacitor, where there are many values for the same part, it is necessary to create a part table. To do this, right click on 'Part Table Files' and select 'New'. This will create a new part table, and launch the PTF Editor. Once the PTF Editor has fully loaded, click the '+' symbol next to the name of the part, in this case 'RESISTOR805', then click on Header and fill in the data as shown below in Figure 3.22. Of course, these values can be modified to meet the requirements of a particular part, such as a capacitor or LED. Fields can be added for other parameters if desired by the designer. This can be very useful when dealing with parts that have very subtle differences. Next, click on the Part Rows on the left. Now click add for each different set of values until all of the required parts have been entered. This should look similar to the one below in Figure 3.22. Now click *File - Save* or *File - Exit* and select 'Yes' when asked if the file should be saved or not. Part Developer will then reload the part file, and when save, the part now contains a part table. It is now appropriate to continue to schematic development.

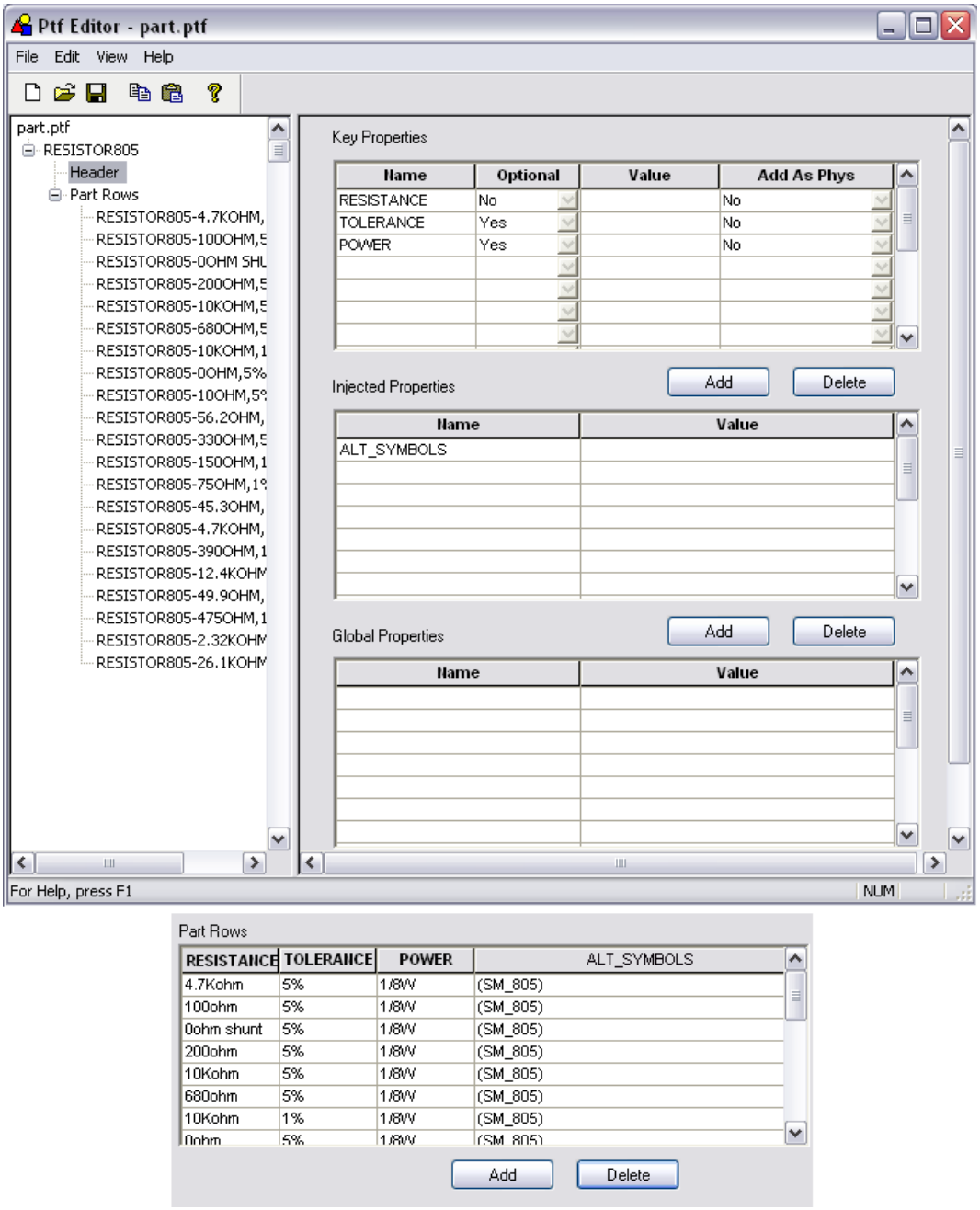

Figure 3.22: Allegro Part Developer -Part Table File Example for 0805 Resistor

3.1.4 Step 4 - Schematic Development

Once all of the parts necessary for the design have been saved into the design library, it is time to begin entering design schematics. In the 'Project Manager', it is necessary to change the product license by clicking *File - Change Product....* The 'Allegro PCB Design HDL XL' provides the optimal set of options for entering schematics. Select this license and hit OK. The click *Flows - Board Design*, which will bring up the board design flow. Now startup the schematic editing tool Design Entry HDL by clicking the 'Design Entry' button. When Design Entry HDL has finished loading, there are a few things to make the experience more user-friendly. Right click in the workspace, and select 'Add Component'. In the component browser that appears, click *Configuration - Setup.* In the left pane, click 'Details' and check the box next to 'Minimize on Add', and click OK. Now click *View - Single Detail Tab.* In the 'Browse Libraries' screen, select the 'standard' library. This library contains special symbols commonly used for creating schematics, including 'VCC' and 'GND'. In order to keep the schematics organized and contained, a symbol in the standard library will be used. Click the cell 'b size page' and then click 'Add'. use the cursor to place this frame anywhere on the page. This object is exactly the size of an 11x17 sheet of paper. Therefore, designs can be easily printed when the time comes with nicely formatted labels that state the nature of the work, and its designer(s). To place these labels, first use the zoom tools (the 'Zoom Points' tool is especially useful in the case) to center around the label section in the bottom righthand corner of the object. Then bring up the text placement window by clicking *Text - Note....* This should produce a window similar to the one shown below in Figure 3.23. In the text placement window, type the page title, designer name, date, and page number into the text box, separated by carriage returns as shown in the figure. Now use the mouse to place the labels inside the boxes they belong in. Allegro seems to lock these labels into a default

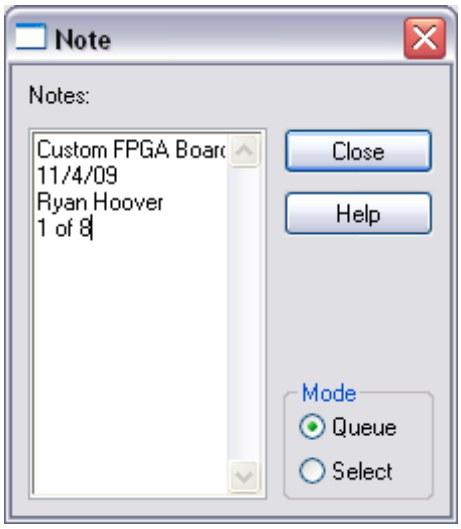

Figure 3.23: Allegro Design Entry HDL - Note Entry Window

position, and when finished, the labels will look similar to those below in Figure 3.24. When the labels are in place, it is time to begin entering schematics. To

Figure 3.24: Allegro Design Entry HDL - Page Labels

add components to the page, follow the same procedure as used before to add the page layout. This time, however, select the library associated with the project (for example *myboard_lib*) and choose the component you wish to add. Then place it in the schematic in a logical location. When dealing with components that have multiple symbols, or components that have multiple part values, click on the part in the table in the bottom half of the screen. This will bring up the part details tab for this part. In this tab, the different symbols available can be viewed via the dropdown menu, and when the correct symbol has been found it can be added to

the workspace in the same fashion as the page layout. The part details tab can be seen in Figure 3.25 below.

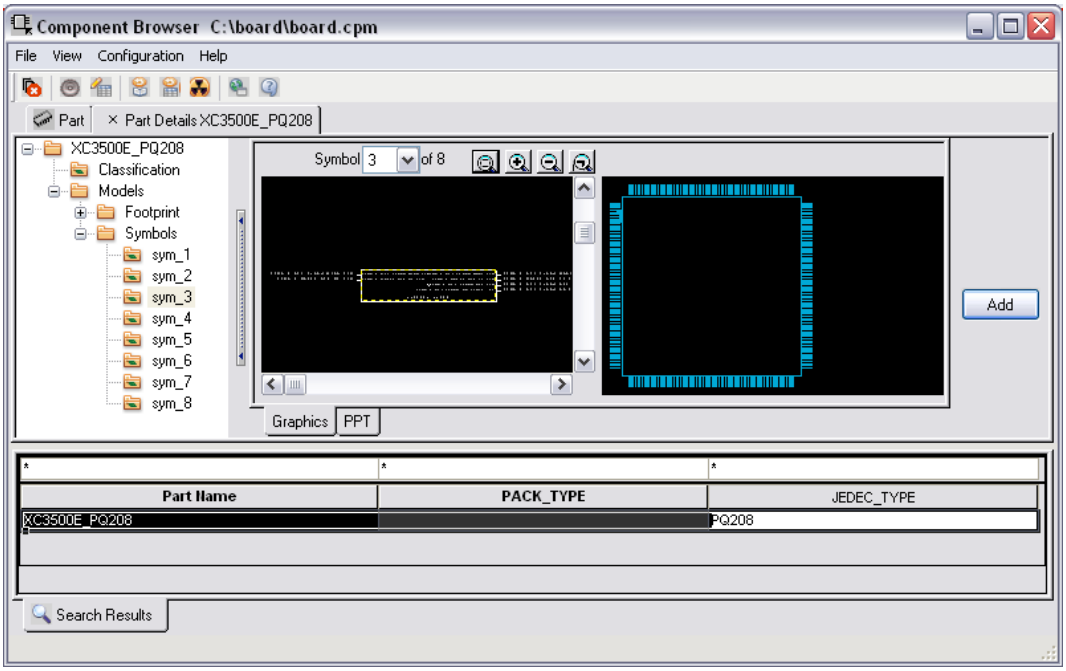

Figure 3.25: Allegro Design Entry HDL - Component Browser Part Details Tab

Draw connections between components by selecting the 'Draw Wire' tool from the toolbar, and then clicking to form right angled paths between two points. To connect to a pin on a component, click directly on the terminal of that pin. To tap off of another wire, simple click on the wire at the point you wish to tap from. This types of connections will look similar to the ones below in Figure 3.26. For bus wires (denoted by thicker lines), Allegro makes it easy to connect a bus to another bus of the same size. However, if the source or destination does not have a bus input, then it may be necessary to tap the bus to bring out the individual components. This is very simple to do with Design Entry HDL. Simply bring the bus along the area in which you wish to break out the bus. Usually, when a bus is tapped, the wires are tapped in numerical order. The tool makes it very easy to tap in this fashion. To add a bus tap, simply click *Wire - Bus Tap...*, and then click on

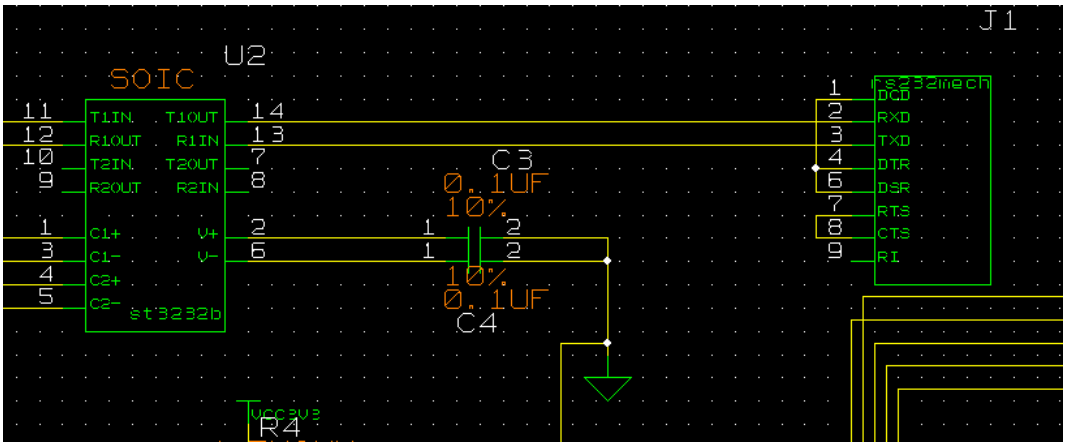

Figure 3.26: Allegro Design Entry HDL - Example of Simple Schematic Connections

the point on the bus closest to the destination pin/point, and then the destination pin/port. Do this until the correct number of taps have been created. Note that at this point, there are only '?' next to the taps. This is because the particular wire within the bus that is being tapped has not been assigned yet. To assign the wires, click *Wire - Bus Tap Values...* Select the most significant bit, and the least significant bit (start and end points), and the increment between consecutive taps. Then, click outside of the starting bus tap on the side away from the other bus taps to be labeled. Now, click outside of the last bus tap on the side away from the other bus taps. All of the taps will now be automatically labeled. An example of a completed tapped bus is shown below in Figure 3.27. To name any net in the design, simply click *Wire - Signal Name*. The procedure for placing the signal names is similar to that of labels. After typing in the signal name, click on the wire to be named.

To create additional schematic pages, click on the last page in the 'Hierarchy Viewer' on the left of the Design Entry HDL window. Then click in the menu bar *File - Edit Page/Symbol - Add New Page*. Then make sure to save the page, so it shows up in the 'Hierarchy Viewer'. Follow the same procedure as before

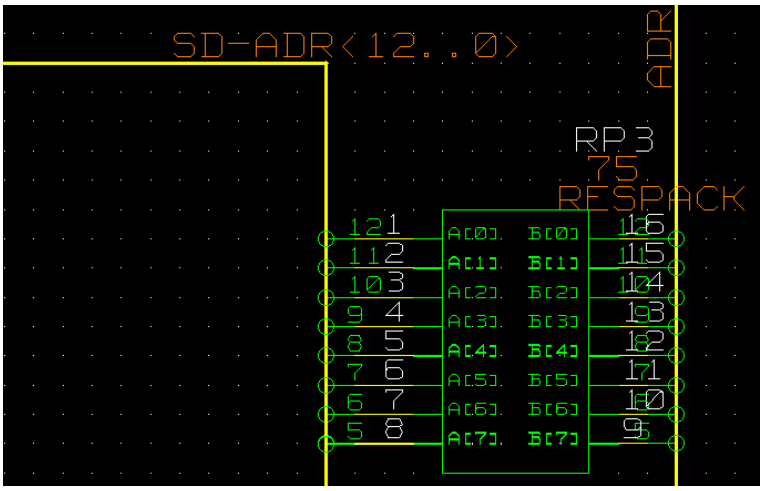

Figure 3.27: Allegro Design Entry HDL - Example of Bus Tapping

to instantiate a page layout and draw schematics for the design. When design schematics are complete, ensure that all schematic pages are saved, and then close the Design Entry HDL, and move on to the PCB Layout Stage.

3.1.5 Step 5 - PCB Layout

At this point the schematics, including all of the parts and their footprints and pads, are completed, and it is time to begin laying out the physical board. Before beginning the process, there is one thing that needs to be done. One required constraint during the layout process is the default via padstack. Vias are the through holes used to bring a signal from one layer in a board to another. Because the minimum hole size for the foundry we are using is 15 mils, create a padstack using the Padstack Editor that matches this specification, and use this padstack when it is asked for in the Board Wizard. Begin by opening PCB Editor from the Project Manager. This can be done by clicking *Tools - PCB Editor*. The most recent document that was open will automatically be re-opened. To get started creating the PCB board file, click *File - New...*. A dialogue box should appear asking for the drawing name and type. Enter a logical drawing name for the board. In this

case, 'myboard' will be used. Then, choose 'Board (wizard)' as the drawing type, and click 'OK'. Depending on the foundry that is being used to tape out the board, there may be configuration data that can be imported in the first few steps of the board wizard. This will ensure that the board meets certain design specifications imposed by the particular foundry. In this case, the foundry does not have these files available, so these steps will be bypassed. After the import steps, the wizard will enter the parameters stage. The first page of parameters contains general settings and preferences. Here, the units to be used in the board file are to be selected. Since most of the measurements from this particular foundry are in mils, it will be easier for this design to work with mils. Note that all the previous parts whose measurements were entered in millimeters will be automatically converted to mils when placed in the PCB design. The drawing size for the board drawing deals with what standard size paper the design will fit inside of if printed at actual size. Size B was used in the schematic stage, and the board that is being produced will easily fit within the specified space, so 'B' is chosen. The last setting is a preference for where the origin of the design will be, at the center of the drawing, or the lower left corner. This setting is purely a personal preference, and matters only when dealing with the placement of parts in relation to the cartesian coordinate system. Figure 3.28 shows the completed 'General Parameters' stage for the example design. Click next when ready to proceed.

The next page of settings deals with grid spacing, etch layer count, and artwork films. The grid spacing can be changed later if necessary, so leave it at its default setting. The number of etch layers is the number of routable layers (including power planes) to be included, in this case four. Select 'Generate default artwork films', although some of these settings will be changed later. Click next to reach the 'Etch Cross-section details' options. This is where the etch layer names will be specified, as well as the layer types. The top and bottom layer are by default set to

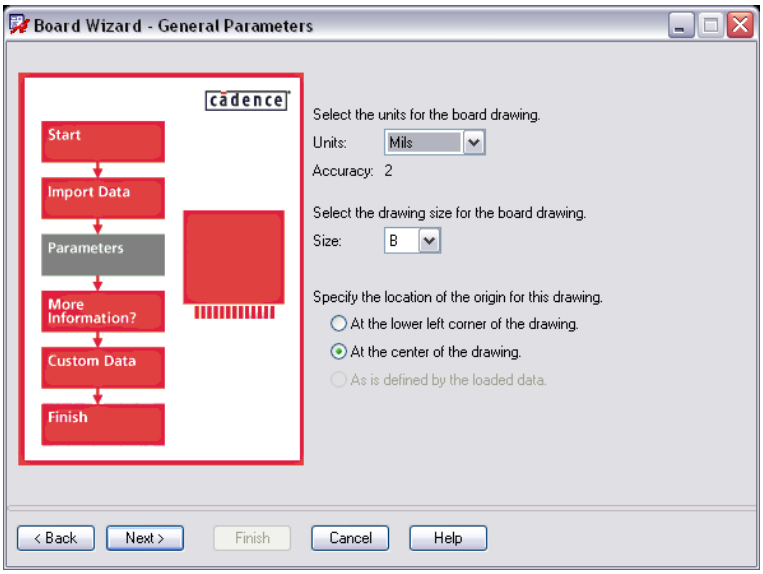

Figure 3.28: Allegro PCB Editor - Board Wizard General Parameters

routing layers, so leave them alone. Layer 2 and layer 3, in this case, will be used as power planes, so set the 'Layer type' to 'Power plane' on both of these layers. Note that these layers can be used for routing if so desired. For this project, VCC3V3 and GND will represent to the middle power planes, so change these layer names accordingly. In addition, make sure that 'Generate negative layers for Power planes' is checked. This should look, when completed, as pictured in Figure 3.29. When finished, click 'Next'. The next step in the wizard is foundry-specific, dealing with 'Spacing Constraints'. To obtain this information, check the website or other source of information from the foundry that will be used to fabricate the design. In the case of Advanced Circuits, the foundry that will be used to fabricate this specific design, the minimum line spacing is 6 mils, while the minimum hole size is 15 mils. This information is found in Table 2.4. Therefore, fill in 6 mils for the 'Minimum Line width'. Hit the 'Tab' key on the keyboard, and 6 mils is automatically filled in for 'Minimum Line to Line spacing', 'Minimum Line to Pad spacing', and 'Minimum Pad to Pad spacing'. These spacing constraints will ensure that when the design is

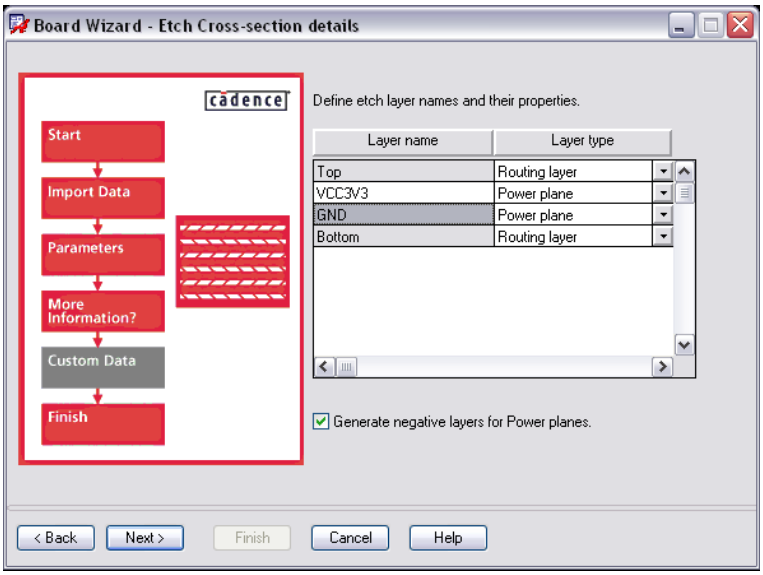

Figure 3.29: Allegro PCB Editor - Board Wizard Etch Cross-Section Details

finished, the foundry is able to reproduce the design. Now click the '...' to the right of the 'Default via padstack' field, and choose the padstack that was created in the beginning of this step. When finished, click 'Next'.

The next step in the wizard asks what kind of board outline will be used, circular or rectangular. For this project, select rectangular. Note, however, that a board is not restricted to any particular geometric shape, despite there only being circular and rectangular in the wizard. Click 'Next' again. Now the wizard prompts for 'Rectangular Board Parameters', including the board width and height. Remember that in Table 2.4, the www.66each.com board specifications state a maximum size of 30 square inches was stated. The design choice for this project was to create a board 6 inches by 5 inches, or 6000 mils by 5000 mils. Fill the width and height values into the board wizard. There is an option to have corner cutoffs of a user-specified length. These are not necessary for this design, so the default values do not need to be modified. The 'Route keepin distance' refers to how far from the edge of the board traces are allowed to get. Though a recommended value is not

stated by the foundry, a value of at least double the 'Minimum Line-Line Spacing' is generally a good practice. Since the 'Minimum Line-Line Spacing' is 6 mils, set the 'Route keepin distance' to 12 mils. The 'Package keepin distance' refers to how far from the edge of the board a component is allowed to get. This is another designer preference, and is not particularly necessary as long as the designer is checking component placement carefully. For this design, 5 mils will be used as the 'Package keepin distance'. When finished, the board wizard window should look similar to the one in Figure 3.30.

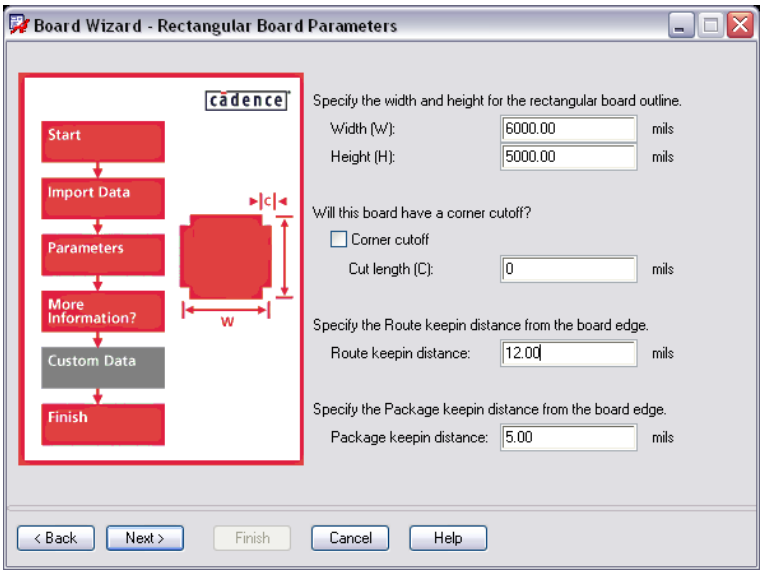

Figure 3.30: Allegro PCB Editor - Board Wizard Rectangular Board Parameters

Click 'Next', and then finish to allow the wizard to generate the board file. Click *File - Save* and then close PCB Editor. In the Project Manager, click the large 'Design Sync' button and choose 'Export Physical'. This utility will synchronize the schematic design that was created with a specified PCB '.brd' file. In this case, choose 'myboard.brd' for both the 'Input Board File' and the 'Output Board File'. Make sure 'Package Design' is checked with the 'Preserve' option, and 'Update PCB Editor Board (Netrev)' is also checked. The 'Export Physical Window should

look similar to the one shown below in Figure 3.31. Next, click the large 'Layout'

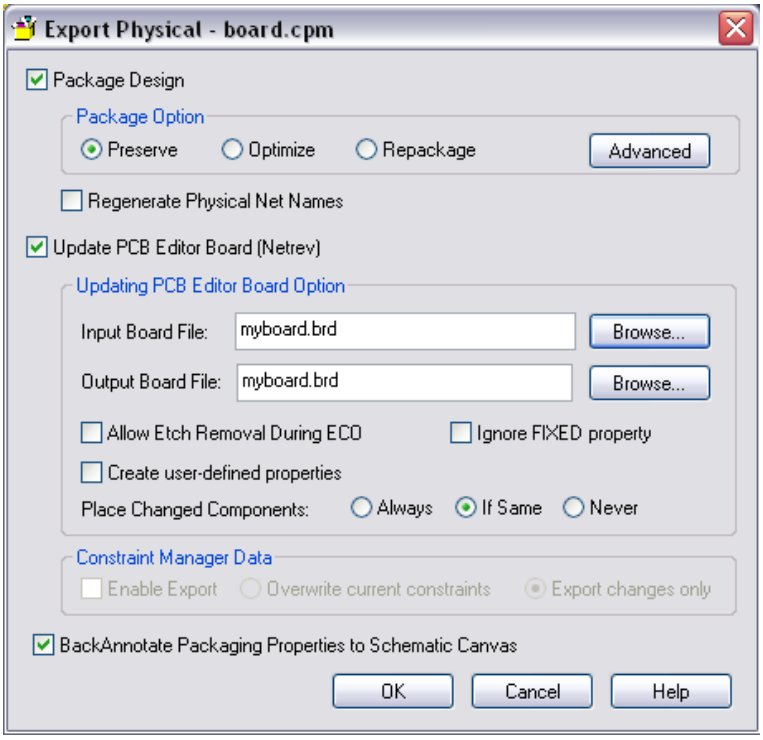

Figure 3.31: Allegro Design Sync

button in Project Manager to reopen PCB Editor. Now its time to place all of the components for the board. To open the component placement window, click *Place - Manually*. The tool has a 'Quickplace' option, however this option is only useful on boards which have a very small number of components. The manual placement window is very useful in that it gives the user several ways to filter the list of parts. This is of course also a personal preference. However, the 'Schematic page number' filter is particularly useful because schematic pages are usually a set of related components, and therefore are generally located within close proximity to one another. For example, all the power supply components are in page three of this particular design, so the filter for schematic page number three will be turned on. Now, only the components on page three are shown in the list. To place a component, simply click on the reference designator of the component that needs to

be placed, and click on the design to place it. This is shown below in Figure 3.32. The tool has many options and setting that can be tweaked to increase productivity.

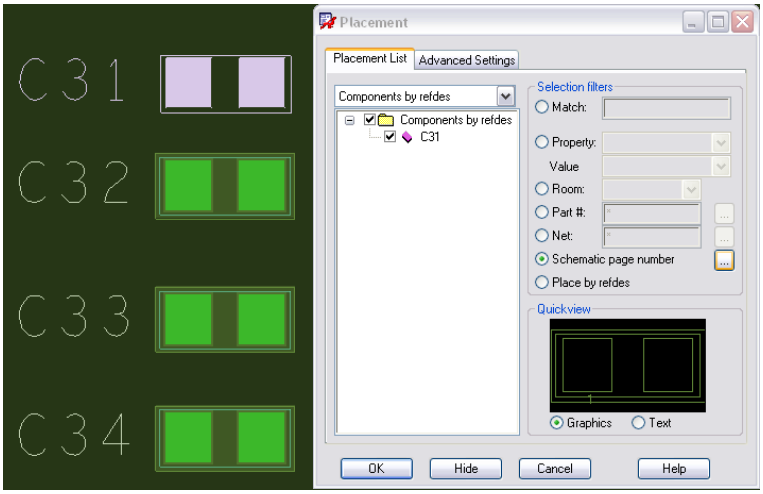

Figure 3.32: Allegro PCB Editor - Component Placement

It is useful for a designer to find what works best for them. Use the lines that indicate a connection between two parts, or rats, to help aid in finding the optimal placement for a particular component. To show only particular rats or to adjust the way in which rats are shown, click *Display - (Show Rats or Blank Rats)* and use the offered controls to customize the experience. Once all of the components have been placed, it is time to move on to the PCB routing step.

3.1.6 Step 6 - PCB Routing

At this point, all of the components used in the design are placed as they should be. Make sure to save a backup copy of the design at this point, as it may be important to go back to this point in the future. There are many steps to accomplish before beginning the actual routing of the PCB. First, identify any differential pairs used in your design. These must be identified to the tool, as they will require special routing rules. Identify differential pairs by clicking *Logic - Assign Differential Pair...*, which will bring up a window that lists the differential pairs currently identified, as

well as a net selection section. To add a differential pair, find and click on the two signals which comprise the differential pair. In this design, those signals are named 'SD-CK_P' and 'SD-CK_N'. Clicking on a signal adds it to the differential pair information located at the bottom of the window. When the proper pair of signals are both displayed in the 'Net 1' and 'Net 2' boxes, respectively, click the 'Add' button to add this pair to the differential pair list. When all pairs have been added, click OK. The window may look similar to the one below in Figure 3.33.

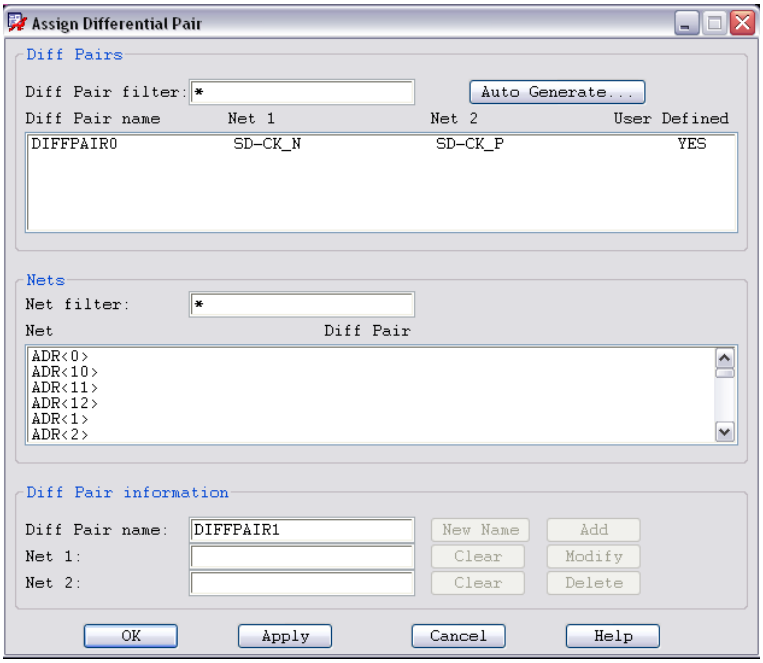

Figure 3.33: Allegro PCB Editor - Differential Pair Identification

The next step is to identify the nets which will be the power planes. Do this by clicking *Logic - Identify DC Nets....* For this particular design, locate the 'GND' net, and click on it. Set the voltage to '0V'. Now locate the 'VCC3V3' net, and click on it. Set the voltage to '3.3V'. Look through the rest of the list, and ensure that the voltage for all other nets says 'None'. When finished, click 'OK'. In order to ensure that the power traces can handle the power that will be flowing through them, it is also necessary to check/modify the design constraints. Open the physical

design constraints by clicking *Setup* - *Constraints* - *Physical...*, which will bring up 'Allegro Constraints Manager'. Under the 'Physical Constraints' folder, select 'All Layers'. Note that there is already a 'DEFAULT' constraint set that contains the spacing rules that were entered in the board wizard, with a line width of 6 mils. However, a 6 mil trace will not be sufficient for a 5V, 2.5mA power line. In fact, 40 mils is a more reasonable line width for the main 5V DC line. In addition, that power is being distributed, after being converted, to many components throughout the board. These lines also require thicker traces, around 12 mils. Therefore, it is necessary to create another two constraint sets. This is done by right clicking on the preexisting constraint set 'DEFAULT', and selecting *Create* - *Physical CSet...*. A dialogue box will appear asking for a name for the new set, which will be named PWR. Follow the same procedure to create a second named PWR40. Now, for each of these two new physical constraint sets, change every value that is currently 6 mils in the 'DEFAULT' constraint set, to the new value for each respective constraint set. In addition, since the default via is 15 mils, for power lines larger than 15 mils (40 mils in this case), a new larger via should be created and used if signals of these type will be moving between layers. When this is finished, the window should look similar to Figure 3.34.

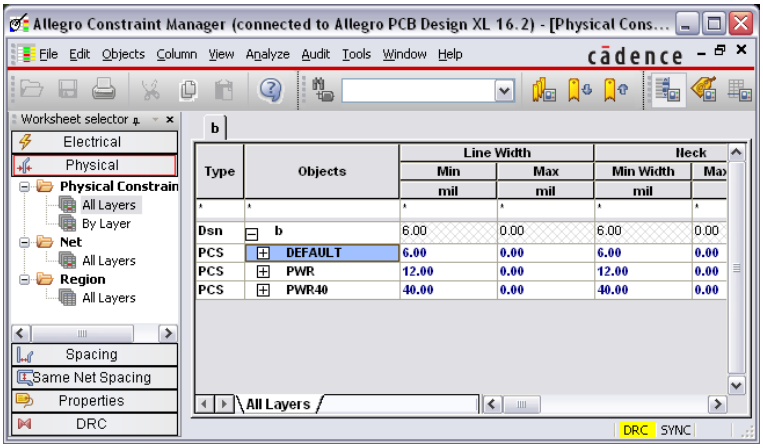

Figure 3.34: Allegro PCB Editor - Constraint Manager

Now that these constraint sets exist, they must be applied to specific nets, so that Allegro knows that all traces in those nets need to follow the new set of constraints. To do this click 'All Layers' underneath the 'Net' Folder to bring up a list of all the nets in the design. Now simply find all the nets to which the new physical constraints should apply, and click where it currently says 'DEFAULT' and use the drop down box to change it to the new constraint set. Once this is completed, explore the Allegro Constraint Manager to make sure all of the constraints meet foundry specifications, and will provide for solid signal integrity. When finished, close Allegro Constraint Manager. The last step before beginning the actual routing is to define dynamic copper shapes. Currently, Allegro knows which nets are to be routed as power planes. However, it does not know what locations on the board it is to fill in copper and route to the plane. These locations need to be defined. In this case, that location can be the entire surface of the board. So to define these dynamic copper regions, zoom out so that the whole board is visible on the screen. Then click *Shape* - *Rectangular*, and open the options panel on the righthand side. Choose Etch as the class, and (Gnd or Vcc3v3) as the Subclass. Note that 'Dynamic Copper' is selected as the shape fill. Then, under 'Assign net name', select the corresponding net (Gnd or Vcc3v3). The options panel should look similar to the one below in Figure 3.35. Then draw a rectangle around the entire board, and repeat this for the remaining power planes. The board is now ready for routing.

There are several methodologies to follow in order to achieve successful routing. The first, and oldest, is to manually route each connection in the design. To do this, simply click the 'Add Connect' button, and route each connection, inserting vias as necessary, until there are no more rats, and all the board is finished routing. The second method is to use an auto router. Allegro has several 'routing' methods. Which are available depends on which type of license is available. However, PCB Editor usually has its built-in router available. To use the automatic router, click

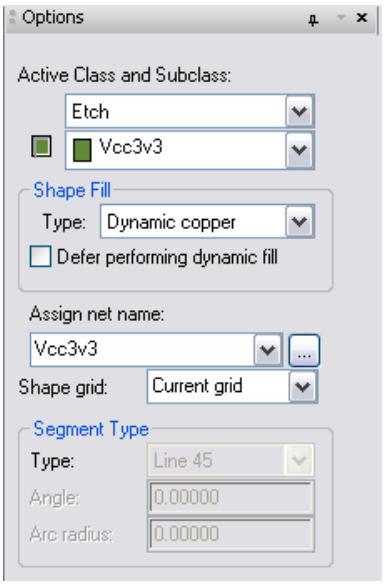

Figure 3.35: Allegro PCB Editor - Dynamic Copper Rectangle Options Panel

Route - *Route Automatic...*, and the 'Automatic Router' window will appear. There is an abundance of options available to custom configure the routing procedure. For this particular project, there were several settings that tended to yield better results that others. In the 'Strategy' section, set the strategy to 'Use smart router'. Now click on the 'Smart Router' tab. Check the box next to 'Miter after Route', 'Minimum via grid', and 'Minimum wire grid'. The default values in these boxes are sufficient. Before pressing the 'Route' button, click on the 'Selections' tab, and not how it is possible to route only specific nets or portions of the design at a time. This is useful if there are several nets that need to be routed manually, but the rest of the nets can be routed automatically. Now click the 'Route' button. The auto router will then make a number of passes, and route as many connections as it is able on the design. If the components were placed with routing in mind, all the connections should route OK. When the router is finished, it returns to the 'Automatic Router' window. Here, you can either 'Undo' if the results of the route are unsatisfactory, or 'Close' the window to accept the results of the route. An example of a routed

board is shown below in Figure 3.36. Before continuing, it is important to ensure

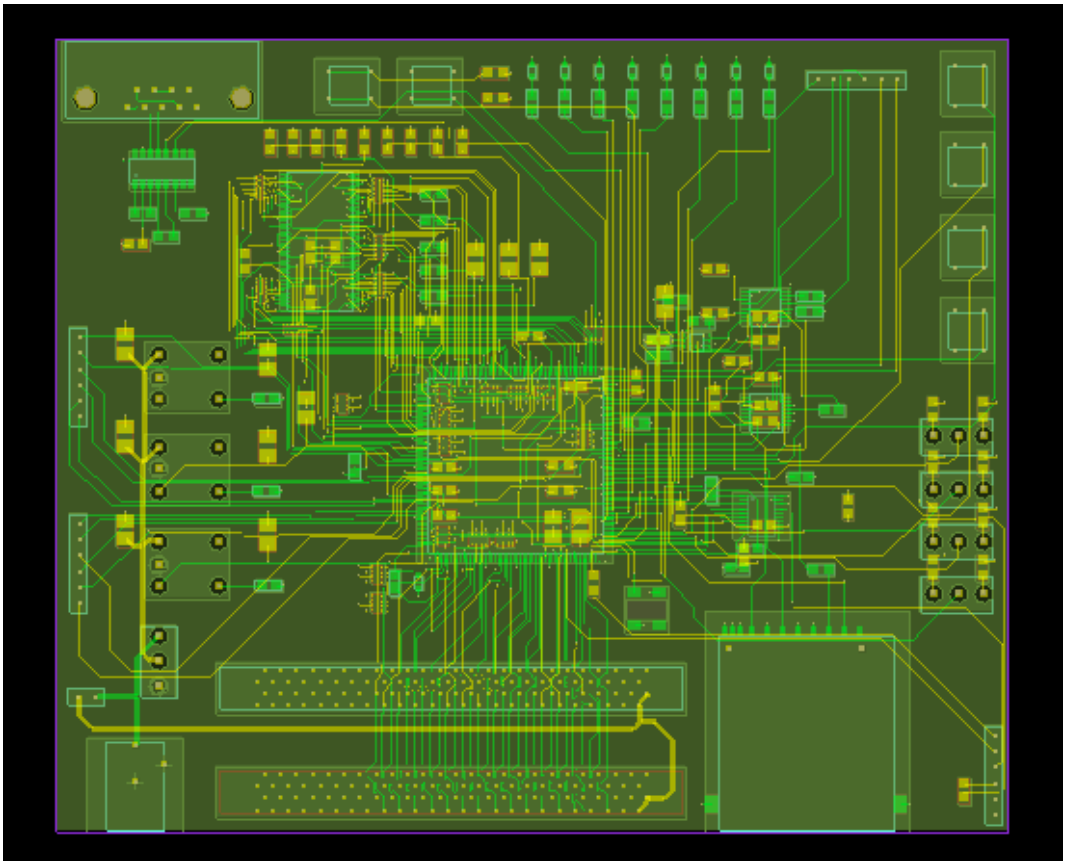

Figure 3.36: Allegro PCB Editor - Automatically Routed Board Example

there are no DRC or other errors, and that all lines have been routed. To do this, click *Display* - *Status*. If all the squares are green, that means everything is as it should be. If not, it may be time to do some investigating.

At this point, the last step before preparing to submit the board for fabrication is to create a silkscreen layer. In order to see and better interpret the silkscreen labels, it is best to change the display colors of the silkscreen layers. To do this, open the color dialogue by clicking *Display* - *Color/Visibility...*, and then click on the Manufacturing folder. Now, choose two bright distinctly different colors. The

closer to white these colors are, the easier they will be to read. Now change the colors of the 'Autosilk_Top' and 'Autosilk_Bottom' layers. When finished, click 'OK'. It may also be helpful to enable and disable certain layers to make the pads and silkscreen easier to see. Now it is time to begin by generating the auto silk layers. First, click *Manufacture* - *Silkscreen...*, and configure the options available to tailor to the design as best as possible. For this design, it was decided that only text labels for the reference designators was necessary in order to aid in the soldering process. The other important labels would have to be done manually. Therefore, the 'Auto Silkscreen' options were set much like those below in Figure 3.37. When

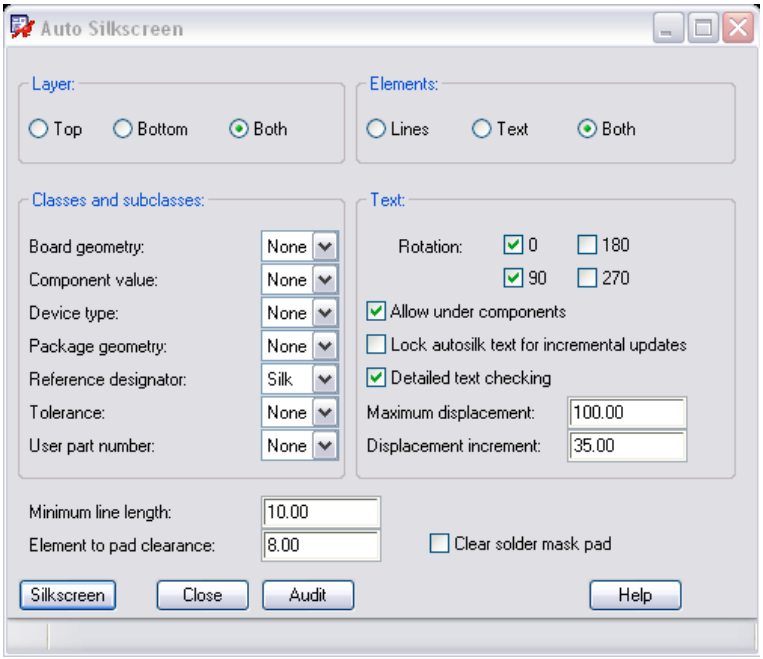

Figure 3.37: Allegro PCB Editor - Auto Silkscreen Generator

ready, click the 'Silkscreen' button. The auto silkscreen generator is helpful, but will never get everything quite right. Therefore, it's important to go through, and manually move the labels around as needed. It may also be necessary to add extra labels. To add labels, click the 'Add Text' button or click *Add* - *Text*. Then, in the

options panel on the right, change the active class to 'Manufacturing' and the subclass to 'Autosilk_Top' or 'Autosilk_Bottom'. Then configure the text parameters and click on the design where the text should be added. When finished typing, right click outside the text, and select 'Next' or 'Done'. When finished added/modifying labels, the board is ready to be sent off for fabrication.

3.1.7 Step 7 - Sending PCB for Fabrication

At this point, the board is a finished product, and the only thing left to do is send off the design to be produced by the foundry. There are many manufacturing settings that need attention before the actual board files can be generated. Begin by clicking *Shape - Global Dynamic Params...*, and entering the 'Void controls' tab. Change the 'Artwork format' to 'Gerber RS274X'. It is not necessary to change any of the other settings. They should look like those below in Figure 3.38. Now,

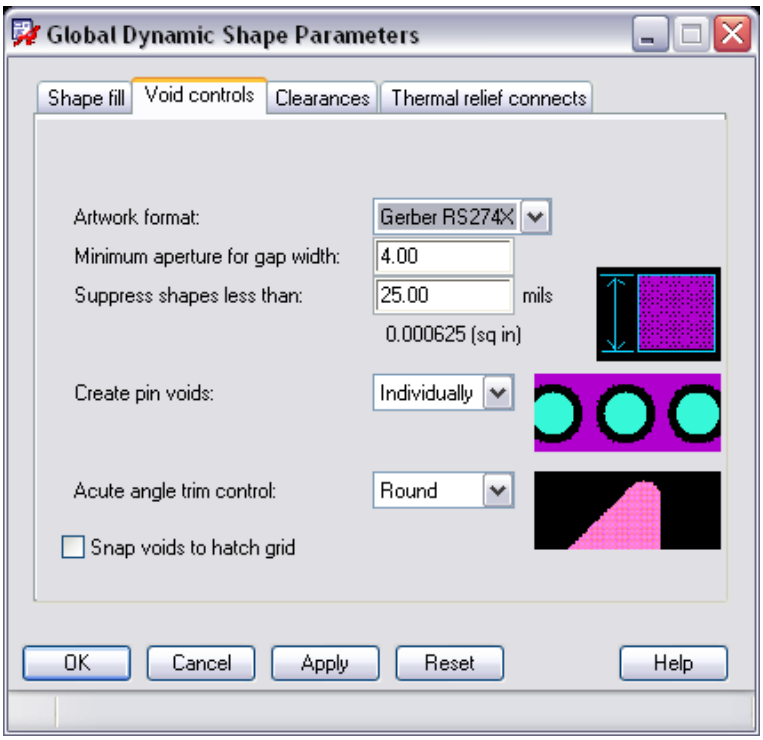

Figure 3.38: Allegro PCB Editor - Global Dynamic Shape Parameters

it is necessary to configure the layers that will exported to gerber format. These output layers are referred to as 'Artwork'. Click *Manufacture - Artwork...* to bring up the Artwork Control form. Now click the 'General Parameters' tab. Set the 'Device Type' to 'Gerber RS274X', and in the 'Format' section, set the 'Integer places' to 3, and the 'Decimal Places' to 5. The 'General Parameters' tab should now look similar to Figure 3.39. Now click on the 'Film Control' tab. In the

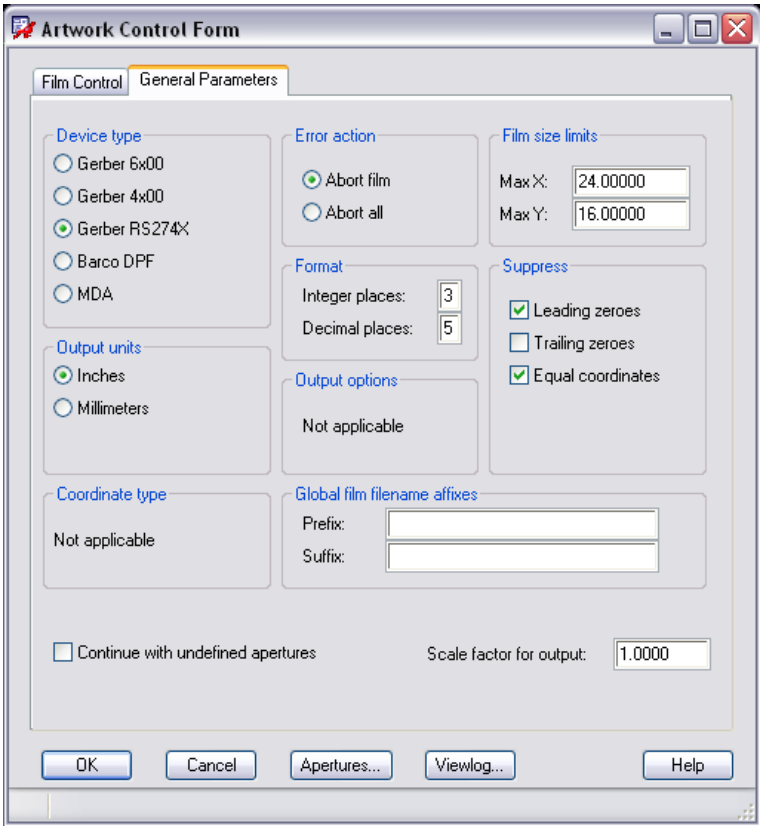

Figure 3.39: Allegro PCB Editor - Artwork General Parameters

'Film options' section, there is an 'Undefined line width' control. This dictates the width of a line whose width is zero, or has not been defined. This is important because it's easy to overlook an undefined line width, especially with silkscreen text. For this reason, set the 'Undefined line width' in all 'Available films' to 10. Now, it is time to add new films. The way that Allegro determines which layer

will be represented when a new film is added is simply by what layer is visible at the time. So, bring up the 'Color dialog' by clicking the Colors/Visibility button, or clicking *Display - Color/Visibility...* from the menu bar. First, click the 'Off' button next to the 'Global Visibility' label. This turns off all layers. Now, click on the 'Manufacturing' folder, and turn on 'Autosilk_Top'. Click 'Apply'. In the 'Artwork Control Form', right click on any of the available films, and select 'Add'. Now type a name for the new film, something such as 'SILKSCREEN_TOP' and click 'OK'. Make sure to set the 'Undefined line width' for this new film to 10 as well. Then, perform this same operation for the 'Autosilk_Bottom' layer, being sure to turn off the 'Autosilk_Top' layer. The next film to add is for the bottom and top solder masks. First, make sure to make all layers invisible again. Then, in the 'Stack-Up' folder and the 'Non-Conductor' subfolder, turn on the 'Pin' and 'Via' subclasses in the 'Soldermask_Top' layer. Now click 'Apply' and right click on one of the available films in the 'Artwork Control Form', and select 'Add'. Type a name for the new film, such as 'SOLDER_TOP', and click 'OK'. Now, do the same for the 'Soldermask_Bottom' layer, being sure to turn off the 'Soldermask_Top' layer. Now would be a good time to save a new copy of the board. This way, if needed, the previous state can be returned to.

The artwork can now be generated. To create the artwork, enter the 'Artwork Control Form' again by clicking *Manufacture - Artwork...*, and then click the 'Select all' button. This will ensure that all the available films are set to be created. Then, click 'Create Artwork'. Once the process is finished, click 'OK' in the 'Artwork Control Form'. Now, the drill pattern must be configured in generated. Bring up the 'Drill Customization' window by clicking *Manufacture - NC - Drill Customization...*, and a list of all the drill holes for this design should be listed. This file will be a map of all the drill hole locations and sizes. Each different type of drill hole is represented by a different symbol. To automatically generate these symbols, click

the 'Auto generate symbols' button. Depending on the design, there may or may not be duplicates of different drill hole sizes. If there are duplicates, set them to the same 'Symbol Figure', and when finished click the 'Merge' button to consolidate the list. This will save time during the fabrication process. The window should now look similar to Figure 3.40 below. Now click 'OK'. A prompt may appear asking

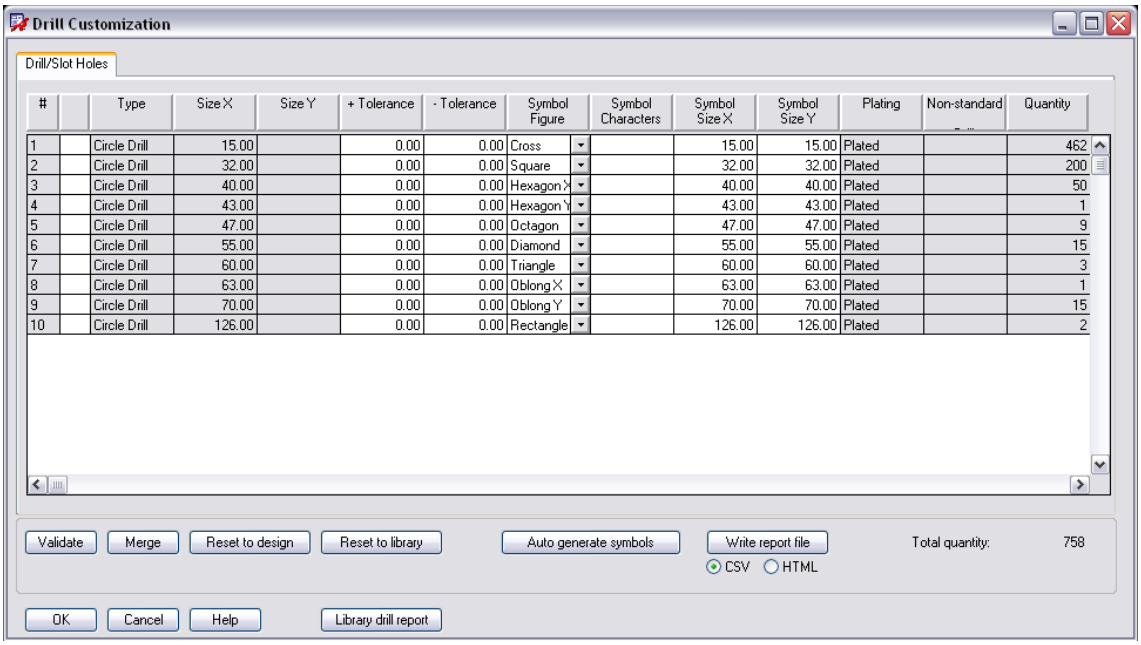

Figure 3.40: Allegro PCB Editor - Drill Customization

if it should make the changes to the pad stacks in the project. This is fine, as there were not any real changes, only the elimination of redundant duplicate pad stacks. It is now time to generate the actual drill file. To begin this process, click *Manufacture - NC - NC Drill...* and click the 'NC Parameters...' button. In the window that appears, make sure that the 'Code' is set to 'ASCII', and the 'Format' to 2.5. Then click the 'Close' button to return to the 'NC Drill' window. Click the 'Drill' button to generate the drill file. Once the process is complete, close the 'NC Drill' window, and save and close the PCB Editor. It is now time to collect the files to send off for fabrication.

3.1.8 Step 8 - PCB Fabrication

At this point, all of the files necessary for fabrication have been generated. Now, the files must be gathered, and prepared to send to the foundry of choice. When these files are generated in Allegro, they are placed in the physical folder within the design directory. For example, if the board folder were 'C:\board\', the files would be located in 'C:\board\worklib\board\physical'. Navigate to this folder specific to the design. Now, create a folder somewhere else to move the design files to. Each of the design files should be named similarly to those shown below in Table 3.1. Once the files have been located and successfully moved to a new folder, all the

Table 3.1: Design Files Needed for Successful Fabrication

File Name
TOP.art
VCC3V3.art or ◁(Inner Layer Name)▷.art
GND.art or ◁(Inner Layer Name)▷.art
BOTTOM.art
SILK_TOP.art
SILK_BOT.art
SOLDER_TOP.art
SOLDER_BOT.art
myboard.drl or ◁(Board Name)▷.drl

files needed for fabrication are ready. Simply submit these files in whatever form is required to the foundry. In the case of Advanced Circuits, they require these files to be in a zip archive. Note that the uploader for this particular foundry did not work properly with some web browsers. Once the design is successfully submitted to the foundry, it is time for the designer to wait in anxious anticipation for the board to arrive in the mail.

3.2 Remarks on Cadence Allegro 16.2 Software

Overall, the Cadence Allegro suite is a very capable tool for creating PCBs. It has many advanced options that are not required for most PCBs. It also uses a

multiple tool structure, so there is alot of moving back and forth between different tools when attempting to make corrections and changes. This is very time consuming, and is a bit of an annoyance. Furthermore, the lack of consistent keyboard shortcuts and general functionality is blatant in the midst of today's applications. The way you are able to perform a particular task in schematic editor, such as moving around the workspace, is completely different in PCB Editor. It's advanced feature set makes in unavoidable in the industry for many tasks, especially those which require high-speed signal integrity testing. The selection of automatic routing tools is unparalleled, and can save a great deal of time on a large design. The compartmental nature of the tool also is suitable for an environment where many people are working on one project, but a hassle for single-party projects.

Chapter 4

DESIGN PROCESS - ADVANCED CIRCUITS PCB ARTIST (FREE)

This section of the thesis will discuss and describe in detail the recommended design process when using the Advanced Circuits PCB Artist software. The tutorial will cover the general process for each step from pad development to PCB fabrication.

4.1 Tutorial

When beginning to work with PCB Artist, it is important to make sure the latest version of the software is installed. PCB Artist makes this easy, as it checks to make sure it is the most recent version upon launch. Once the program has completely launched, click *File - New*, and select 'New Project'. Click 'Browse...' to specify a location for the new project. Then click 'OK'. The project is now ready to begin.

4.1.1 Step 1 - Part Search

PCB Artist is different from other PCB creation tools in that there is a library of pre-made parts available to use, as well as a multitude of built-in parts included in the software package. To check if any of the required parts are in the database, first check in the included database. Do this by clicking on the 'Libraries' button, or click *File - Libraries*. This will bring up the 'Library Manager'. Inside the 'Library

Manager', the 'Folders' tab allows the management of the libraries and folders that have libraries within them. Click on the 'Components' tab. The dropdown menu on the top allows the selection of the particular library. The libraries are organized by manufacturer or component type. Be sure to check any libraries which might contain the desired part. To check the online library components database, click the 'Download Online Library Components' button. This will launch the system's web browser, and links to the online part database with a search system as shown below in Figure 4.1. As is evident from the figure, a versatile search system is available,

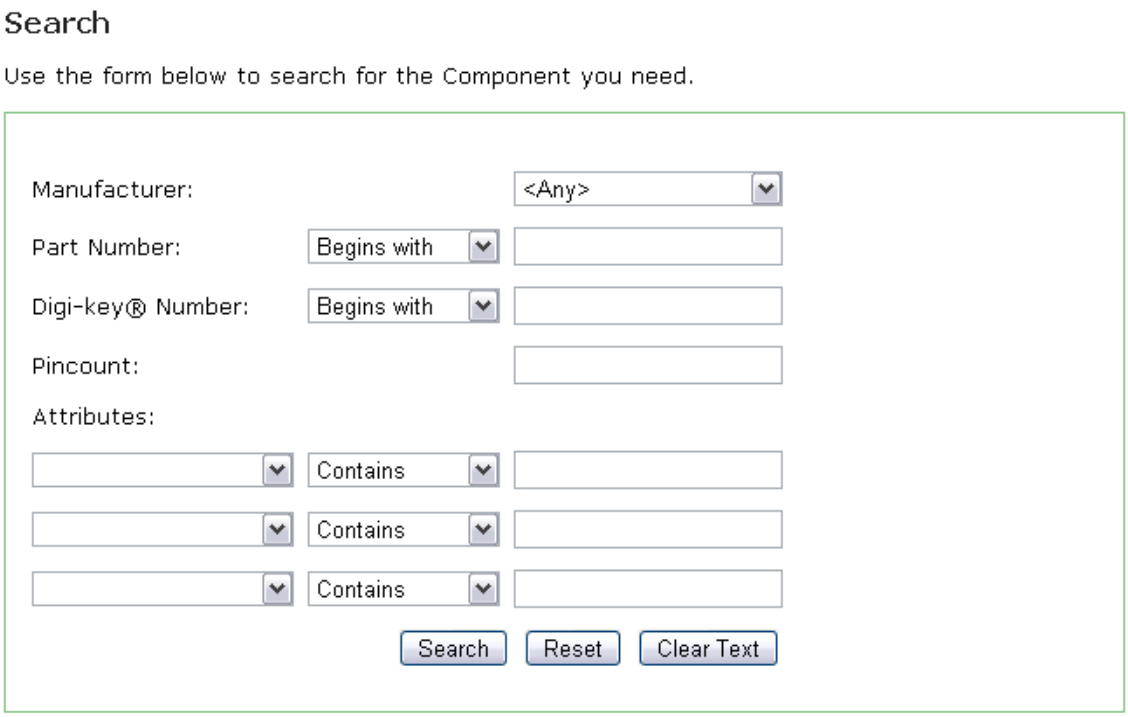

Figure 4.1: PCB Artist - Online Component Library Search Form

allowing the designer to quickly and easily find parts for which there already exists a mechanical symbol and a schematic symbol. Once the desired part is found and the details have confirmed that it is indeed the correct part, click the 'Download' button, and store the file in the desired location. To load the part into PCB Artist, simply drag the part file from it's file folder on top of the PCB Artist workspace.

This will load the part into the 'Downloaded' library. Part files can also be loaded by clicking 'Add File' in the 'Library Manager', and change the 'Files of type' dropbox to 'All Files'. Then, find the downloaded file, select it, and click 'Open'. Once all of the available parts have been downloaded or located in the libraries included with PCB Artist, it is time to move on the part development stage.

4.1.2 Step 2 - Part Development

At this point, all components that PCB Artist has included in its database, as well as the online database, have been located. Now it is time to create component entries for all of those parts for which a suitable preexisting part could not be located. First, create a library to store the new parts in by clicking 'New Lib...' in the 'Library Manager'.As an example, two components will be created, the PJ102-AH and the RS3232 controller chip. To begin creating the mechanical part, it is necessary to look at the datasheet for the first part, the PJ102-AH, shown in Figure 3.8. Bring up the 'Library Manager' and bring up the 'PCB Symbols' tab. Click on the 'New' button. An empty canvas should appear. First, click *Settings - Units...* and select the desired units for the part. Before placing pads, the pad style must be defined. To do this, bring up the 'Design Technology' window by clicking on the 'Abc' button or slicking *Settings - Styles...* in the menu bar. In the 'Design Technology' window that appears, click 'Add Style...' to bring up the 'Pad Style' dialogue window. Since this pad style is for the PJ102AH, call the style 'PJ102AH_PD'. Since the actual pin, at its largest, is 1 millimeter, set the drill hole size to 1.5mm. The 'Width' field refers to the plated part of the pin or pad outside of the hole. This needs to be atleast 20 mils larger, or around 0.3mm, so set the 'Width' field to 1.8, and hit 'OK' to save the new style. Figure 4.2 shows the completed 'Pad Style' window. If there are multiple pad styles in the datasheet, add the others in a similar fashion now. In this case, there is only one, so placement can be started. Enter pad placement mode by clicking the graphic of the pin, pressing F4, or by clicking *Add - Pad*. To set the pad

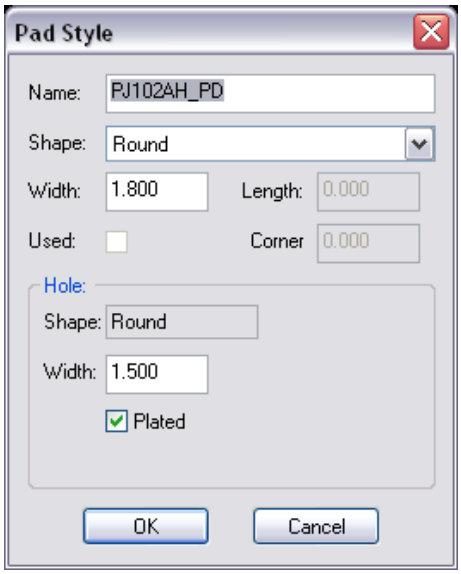

Figure 4.2: PCB Artist - Pad Style Dialogue Box for PJ102-AH Pad

style, right click inside the canvas, and select change style (or press the 'S' key on the keyboard). Select the recently added 'PJ102AH_PD' style, and click 'OK'.. Now, calculate the distance in the X and Y direction that the pin 2 is from the origin of pin 1. Click *Settings* - *Grids*, or click the 'Grids' button to bring up the grid settings. In this case, from pin 1, pin 2 is exactly 6 millimeters in the Y direction, and pin 3 is 3 millimeters in the Y direction and -4.7 millimeters in the X direction. So, in the grid settings, set the X grid to 3 millimeters, and the Y grid to 4.7. Make sure that the 'Snap Mode' is set to 'Grid'. This means that when placing objects, the cursor will lock the possible locations to increments of the grid. The Grid settings window should look like the one in Figure 4.3. Then click inside the canvas to place pin 1. It does not matter where, as this will be set as the symbol origin automatically. Now click to place pin 2 two increments in the Y direction above, and place pin 3 halfway in between one increment in the -X direction. To verify that these pins have been placed correctly. Use the measurement tool by clicking the 'Measure Gap' (picture of ruler) on the toolbar, or by clicking *Tools* - *Measure* to check the

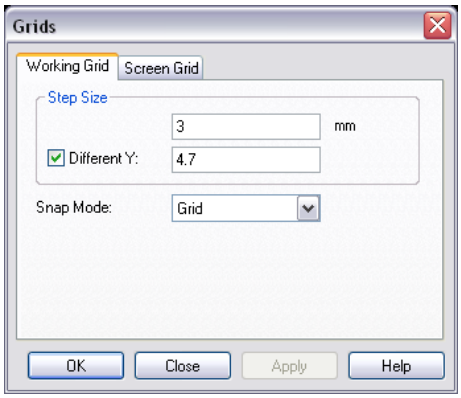

Figure 4.3: PCB Artist - Pad Grid Settings Window

distance between two pins. Click on one pin, then the other, and the measure tool will display the distance between them, as well as the angle of measurement. Note that pads can also be placed by exact XY coordinates by first placing the pin in a random location, then press *Alt* - *Enter* or right click and select 'Properties...', and changing the X and Y coordinates in the 'Position' section. Once the pins have been placed, technically speaking the symbol is finished. However, it is very useful to place indicators for the size of the physical part. To do this, use the rectangle tool, and draw a rectangle as close to the estimated size as possible. Then, click on the sides, and use press the '=' key, or right click and select 'Type Coordinate' to change the location of the side. Once this is finished, save the symbol by clicking *File* - *Save*, pressing *CTRL* - *S*, or clicking the disk icon in the toolbar. A prompt will appear asking for the library and name for the symbol. Store the symbol in the newly created library, and name the symbol something meaningful, such as the the package type if the symbol has one. A prompt may appear asking whether to save the default technology file. This is because there were new styles added. Be sure to click 'Yes' to allow PCB Artist to save the changes to the technology file.

The symbol creation process is different for parts that have a standard package, such as the RS3232. This part in particular is an SOIC-16 package. The

datasheet for this part is shown in Figure 4.4.

Because this is a standard part, the symbol wizard can be utilized. From the 'Library Manager' in the 'PCB Symbol' tab, click the 'Wizard...' button. When the first prompt asks which technology file to use, leave it set to the default file unless the desired units are not select. In that case, uncheck use technology file, and select the proper units, and click 'Next'. The next prompt asks which type of footprint is to be created. Since this is an SOIC type part, click 'SOIC' and click 'Next'. Fill out the next prompt according to the information in the datasheet. The H and T sizes are used to make the pads larger than the pins to ease the difficult of soldering. Be sure to enter a name for the created pad style. This will ensure that the pads can be easily changed later if necessary. When finished, click next. The next prompt deals with silkscreen options for the symbol. Set them as necessary, and click 'Next'. This prompt asks whether or not a placement outline is desired. After this is decided, and the necessary options selected, click 'Next'. Here, enter the name for the new footprint, and select the library for it to be saved to. At this point, the symbol should look similar to Figure 4.5. Click 'Finish' to generate the symbol. It will automatically be opened for editing. If the pads are not the correct size, simply change the dimensions in the 'Design Technology' window. Save if needed, and close the symbol. It is now time to create the schematic symbol.

Open the 'Library Manager' and click the 'Schematic Symbols' tab. Unlike the PCB symbols, the wizard can be used for just about any schematic symbol that will have a standard box shaped symbol. Click the 'Wizard' button, and click the 'Next' button to bring up the first prompt. Again, the default technology is sufficient for schematic symbols, so click 'Next' again to bring up the symbol type prompt. Choose the appropriate type, which in this case is the 'Rectangle' type symbol. Click 'Next' again to go to the style selection prompt. Set the pin and line styles as desired for the component. When finished, click 'Next' again. It is

SO-16 mechanical data

Dim.	mm.			inch.		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			1.75			0.068
a1	0.1		0.25	0.004		0.010
a2			1.64			0.063
b	0.35		0.46	0.013		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.019	
c1	45° (typ.)					
D	9.8		10	0.385		0.393
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F	3.8		4.0	0.149		0.157
G	4.6		5.3	0.181		0.208
L	0.5		1.27	0.019		0.050
M			0.62			0.024
S	8° (max.)					

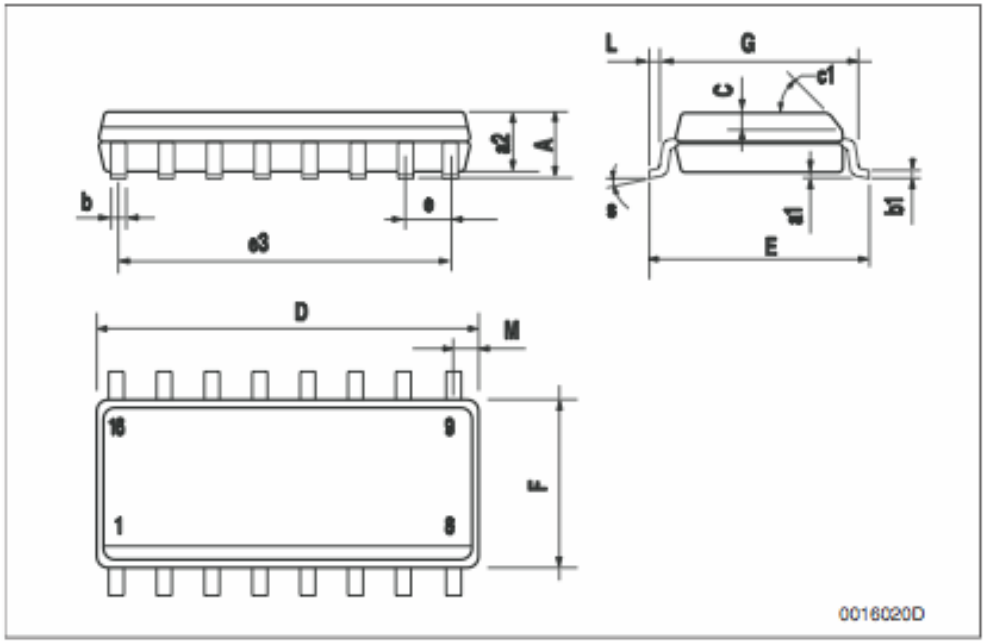

Figure 4.4: STMicroelectronics ST3232CDR Package Dimensions

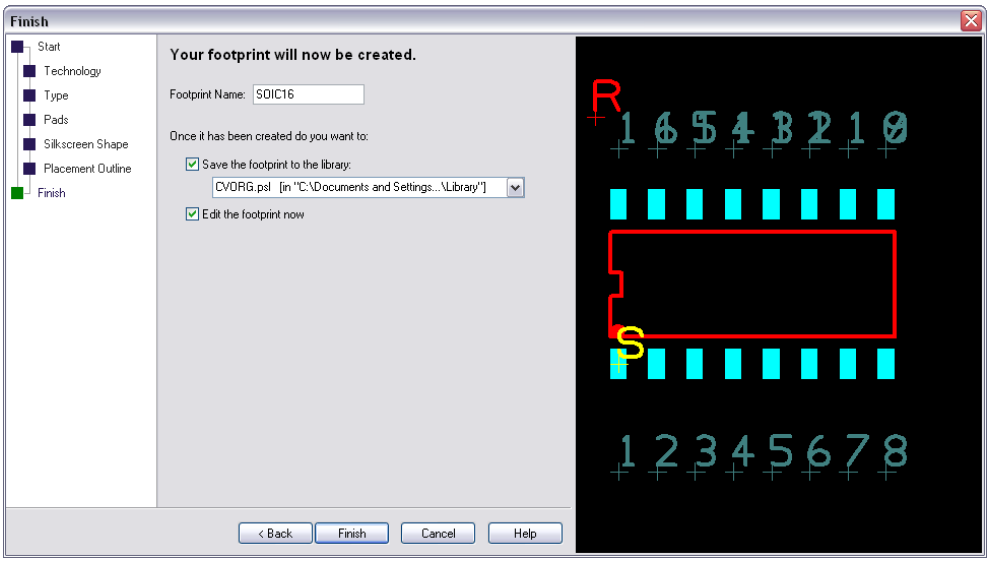

Figure 4.5: PCB Artist - Symbol Wizard Finish Prompt

at this stage that the pin information is listed. Depending on the type of part, a decision will have to be made as to how many pins will reside on the left, right, top, and bottom of the symbol. Generally, power pins go on the top and bottom of a symbol (however, PCB Arist does not allow this), while the inputs go on the left, and the outputs go on the right. This is not always true, and the designer may choose what is best for the particular design. In the case of the power connecter, one designer may choose to place the ground and the 'VCC5V' on one side, or since there are no other pins, they may be placed on the left and right. Note that there are two ground pins on the component. However, those two pins can be mapped to one schematic pin for simplicity sake. When finished, click the 'Next' button. Now, choose a name for the new schematic symbol. If it is a generic and frequently used schematic symbol, name it as such. However, most schematic symbols will be suited for their specific component. Click 'Finish' and if necessary, make minor adjustments, then close the window.

The next step is to connect the PCB symbol with the schematic symbol to make a component. To do this, open the 'Library Manager' and click on the

'Component' tab. From experience, the component wizard is lacking alot, and it is much easier to simply create a new component the normal way. Begin by clicking the 'New Item...' and the new component form will appear. Type in a component name, in this case 'PJ102-AH', and choose a package type. If the correct package type is not in the dropdown menu, simply type it in, or choose the 'USER' package type. Select the default reference, which is basically the reference designator. Typically, 'CONN' is used for connectors, 'R' for resistors, 'C' for capacitors, 'U' for IC chips, etc. Next, choose the library and name of the schematic and PCB symbols. If the PCB symbol will use more than one schematic symbol, select the first one. The rest will be added later. When finished, click the 'OK' button. An example of the new component form is shown in Figure 4.6. In the Terminal Name column, enter the

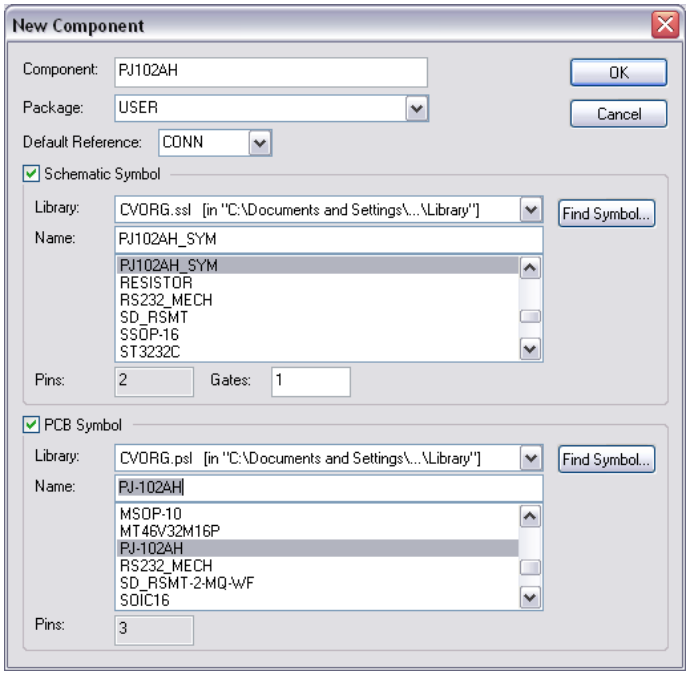

Figure 4.6: PCB Artist - New Component Wizard

names that will go on the schematic symbol for the pins. To map these symbol pins to the PCB symbol pins, enter the corresponding pin on the PCB symbol in the 'PCB Symbol Pad Number' column. If multiple pins need to be mapped, separate

them with a comma, as shown in Figure 4.7 below. If another schematic symbol (PCB Artist refers to them as 'gates' in this context) is required, simply right click on the first gate and select 'Edit Gates...' or click *Add - Gate...* and then select the schematic symbol (gate) to add. Repeat this as many times as necessary. When finished, simply save the part and close the window. Once all of the parts needed

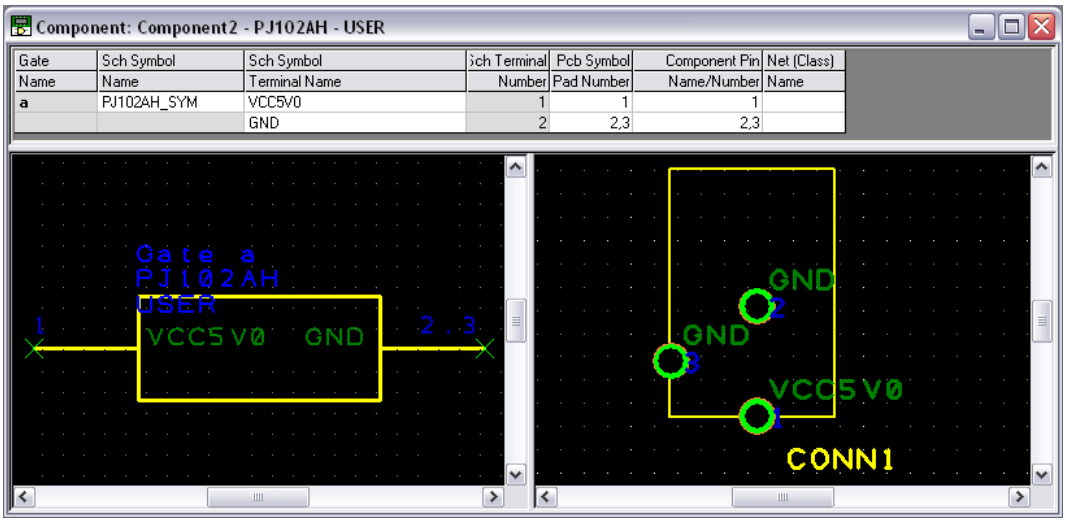

Figure 4.7: PCB Artist - Complete Part - PJ102-AH

for the project have been created and stored in a library, it is time to continue to the schematic development stage.

4.1.3 Step 3 - Schematic Development

At this point, all of the components necessary for development have been created and stored in the necessary library. To begin schematic development, first open the project if it is not already open. Click the 'New' button or select *File - New...*. Select the radio button for 'New Schematic Design'. Use the 'Browse...' button below to select a location and name for the new schematic design. Click 'Save' when finished, and make sure that 'Add to Open Project' is checked before clicking 'OK'. There should now be an empty schematic canvas. If desired, a page layout can be placed in the canvas of a standard paper size such that printing the schematic design

will be easier and look more professional. To do this, or to begin adding components, click the 'Add Component' button, select the 'Add Component' tab in the panel on the right, or click *Add - Component...* and select the library of choice, and then the component. To add the page layout component, select the 'Schema' library and then select one of the ANSI components and place it in the canvas. Note that the 'Schema' library also contains standard symbols for Vcc, Vdd, GND, and other useful components. Once the desired component has been placed in the canvas, press the 'Esc' key or right-click and select 'Cancel' to exit component placement mode. Repeat this process to add any other needed components. To draw connections between the components, click the 'Add Schematic Connections' button or click *Add - Connection*. Then, simply draw the connection from one terminal or component to the next. To split the schematic drawing into many different schematic files, simply create a new schematic file with the project open, following the same procedure as above. PCB Artist will link the schematics together as part of the project. Note that one gate of a particular component can be in one schematic, while other gates can be in different schematics, even though they are representing the same part. If one gate has been added to a particular schematic, but not the rest, the other gates will show up in the component bin. To add then to another schematic simply drag them from the component bin to the other schematic. A completed schematic looks similar to the one in Figure 4.8. Once all necessary schematics have been completed, it is time to continue to the PCB layout stage.

4.1.4 Step 4 - PCB Layout

At this point, all of the schematics have been entered for the design. Simply click the 'New' button or select *File - New...*. Select the radio button for 'New PCB Design', and make sure that 'Add to Open Project' is checked. Then click OK, and select a location for the new PCB board file, and click 'Save'. Click Next to bring up the 'Board Size' prompt. Here the units for which the board will be measured

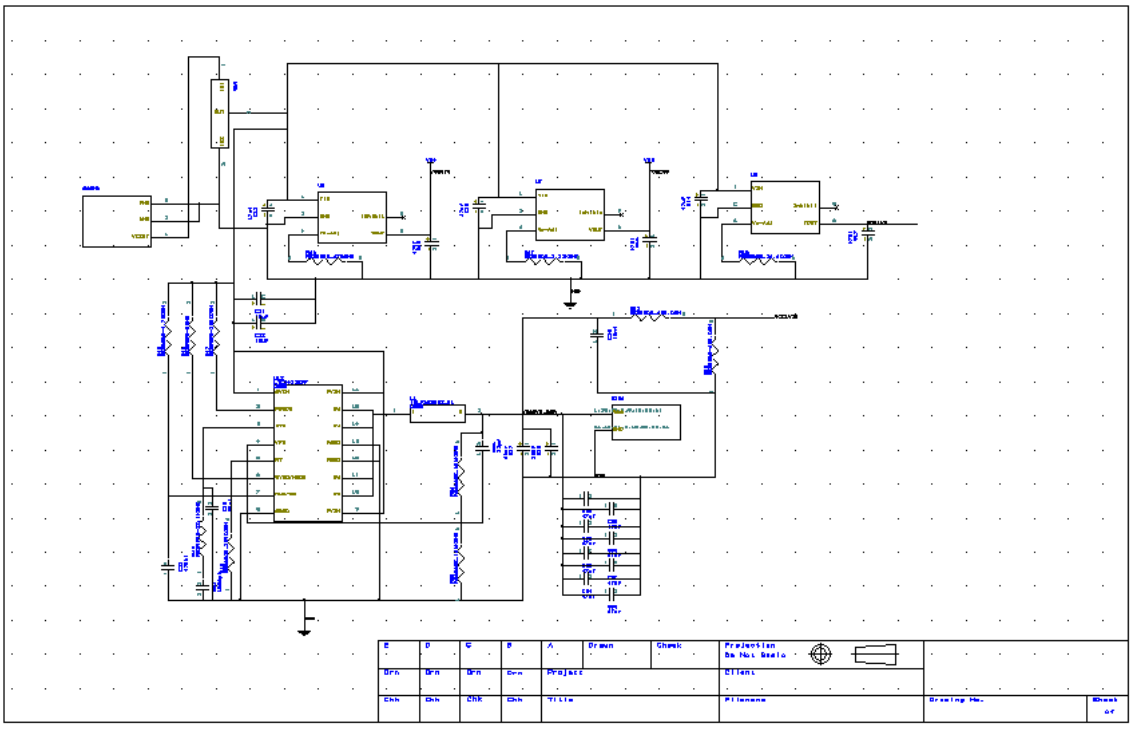

Figure 4.8: PCB Artist - Complete Schematic Example

can be selected, and the size specified. In this case, the size is 6 inches by 5 inches, a total of 30 square inches, meeting the Advanced Circuits 66each standard. Click 'Next' to move on to the 'Design Requirement' prompt. In this case, the design requirement for the 66each standard is '4-Layer Standard'. Click 'Next' to go to the 'Layers' prompt. In the 'Powerplane Layers' section, specify which layers will be power planes, and the nets that will be attached to this power plan. In this case layer 2 and layer 3 will be the 'VCC3V3' and the 'GND' nets, respectively. In the 'Solder Mask' section, the option is given to cover the vias in the design with soldermask. This means that the vias cannot be used as test points, so this is generally not a good practice until the final production version of the board. However, in final versions, covering the vias can help prevent adjacent electronics from accidentally touching a via. In the 'Silkscreen' section, be sure to check the 'Bottom Side' box so that the bottom silkscreen is enabled. Then click the 'Next' button again to move on to

the 'Board Parameters' prompt. Here, most of the options cannot be changed due to the 'Standard 4-Layer Standard' specifications. However, the material thickness can be changed to 0.031 or 0.093 inches. In this case, though, 0.062 inches will work nicely, so this setting need not be modified. The minimum track width, on the other hand, is by default set to 0.010 inches, and this is far to large a constraint for this particular application. Set the minimum track width to 0.007 inches. Note that while 66each standards allow for a minimum track width of 0.006 inches, PCB Artist does not allow this option using this specification. When finished, click the 'Next' button. This will bring up the 'Additional Requirements' prompt where electrical testing can be added to the final order, though for a 4-layer board, this is not really necessary. Click 'Next' again to move to the 'Production' prompt. At this point, a board part number and revision number must be specified. The rest of the options can be changed later by clicking *Settings - PCB Configuration....* When finished, click 'Next' and enter a design name for the new PCB board and click 'Finish'.

A board outline will appear on the canvas. There are no components anywhere on the canvas. This is because the design schematics have not been forwarded to the PCB design yet. To do this, click *Tools - Schematic <–>PCB - Forward Design Changes...* and PCB Artist will sync the design components and nets. Most likely, the tool will place all of the components at the bottom of the canvas. It make take some zooming and scrolling to locate all of the components. Now the components must be arranged logically, and with routing in mind within the board outline. Move components around by simply dragging them file folder style to the destination location. Press the 'F' key to flip a component's side (i.e. top or bottom). Below in Figure 4.9 is an example of a board with all of its components placed, but no routing has been done. Note how the placement of the components minimizes the number of overlapping rats (connection lines), and thereby increasing the chance of an uninterrupted connection. In the example, the silkscreen has been

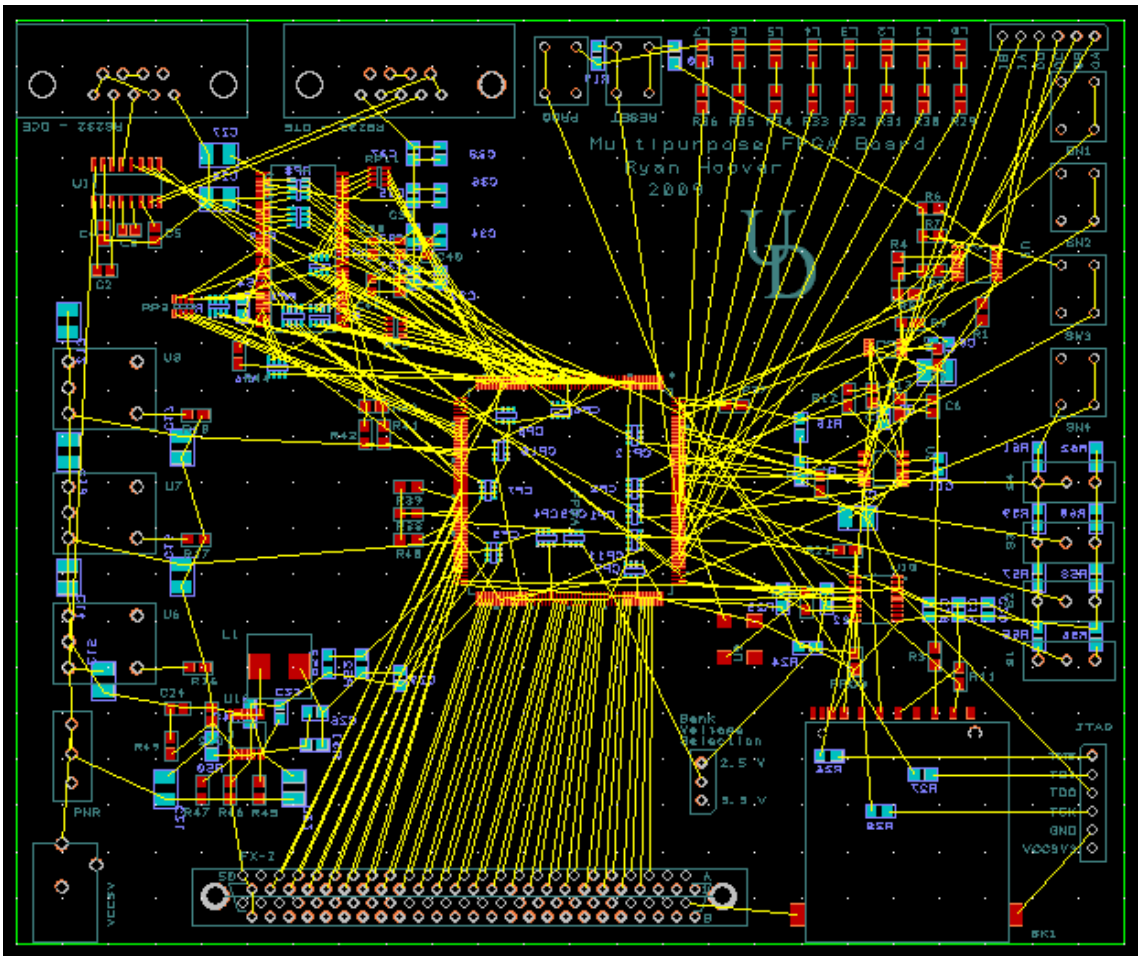

Figure 4.9: PCB Artist - Completely Placed Board Example

polished, however this shouldn't normally be done at this point. At this point, it is time to move on to PCB routing.

4.1.5 Step 5 - PCB Routing

At this point, all of the components necessary for the design should be placed logically for the purpose of the board, and in the way that will reduce the number of vias needed to route the design. Unobstructed rats are best. Like many PCB design programs, PCB Artist also has an automatic routing feature. However, the

automatic routing feature is only useful for designs with a limited number of components, and larger sized pads. Therefore, it is necessary when using PCB Artist to design boards such as the one in thesis to route all the connections manually. The downside to this is that it is time consuming. However the routing, when done correctly, is generally better than an automatic router is capable of accomplishing. To manually route connections, select the 'Add Track' button, or click *Add - Track* and click the pin or pad to start the connection. Then, either click to make vertices and route the connection to its destination. If a via is necessary, right click and select 'End on Via'. Then, place the via, and click again to begin routing in the opposite layer from the newly created via. In certain situations, it is useful to follow a vertical-horizontal top-bottom layer schema, as shown below in Figure 4.10. To

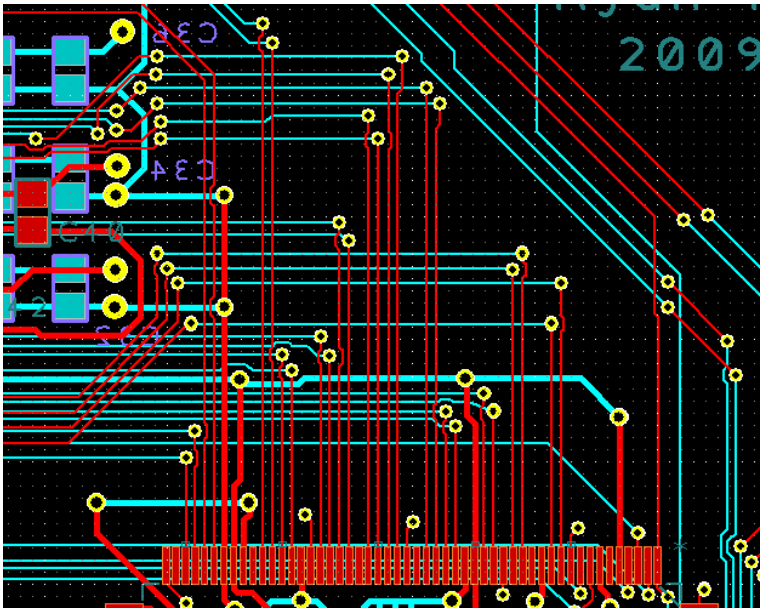

Figure 4.10: PCB Artist - Horizontal/Vertical Routing Example

change the layer of a track (or trace), click the start point of the track, and then right click in the canvas, and select 'Track Layer...' or press the 'L' key. Right clicking in many modes presents many other options to the designer such as the ability to highlight all instances of a particular net, change track or via style, or type a

coordinate point to place a vertex at a particular point. The keyboard shortcuts are mostly the same as those used when editing schematics.

It's also very important to do DRC, or Design Rules Check, throughout the routing process. This will help to identify any violations of the constraints, as well as bad connections and places where the schematic does not match the board. Unfortunately, PCB Artist does not yet automatically DRC in real-time, so make sure that throughout the routing process, a DRC is run. If DRC is ignored until the end, it will be much more difficult to correct the errors because of all the other traces around the one for which there is a DRC error. When routing is finished. It is necessary to make sure all silkscreen labels are in their optimal position. These can be moved independent of the components which they represent. To add new labels, simply click on the 'Add Text' button, click *Add - Text...*, or press the 'T' key. To change the font or other settings of the text, either right click on the text, and select 'Properties...' or highlight the text and press '*Alt - Enter*' to print up the properties window for the object, and customize it. When all reference designator labels have been adjusted, and custom labels added, the board should be ready to be sent off for fabrication. Below in Figure 4.11 is an example of a routed and labeled board ready to be sent off for fabrication. At this point, it is time to send the board to the foundry, Advanced Circuits, to be fabricated and shipped so that it can be assembled.

4.1.6 Step 6 - Sending PCB for Fabrication

At this point, the board has been completely routed and DRC checked, and all finished touches have been completed. Before sending the board off for fabrication, it is helpful to run one last DRC, and check the design status by clicking *Output* in the menu bar, and selecting the three available reports in the middle section. The design status, design rule check, and net completion report will provide a good idea of the status of the board and if it is ready to be send off for fabrication. When ready,

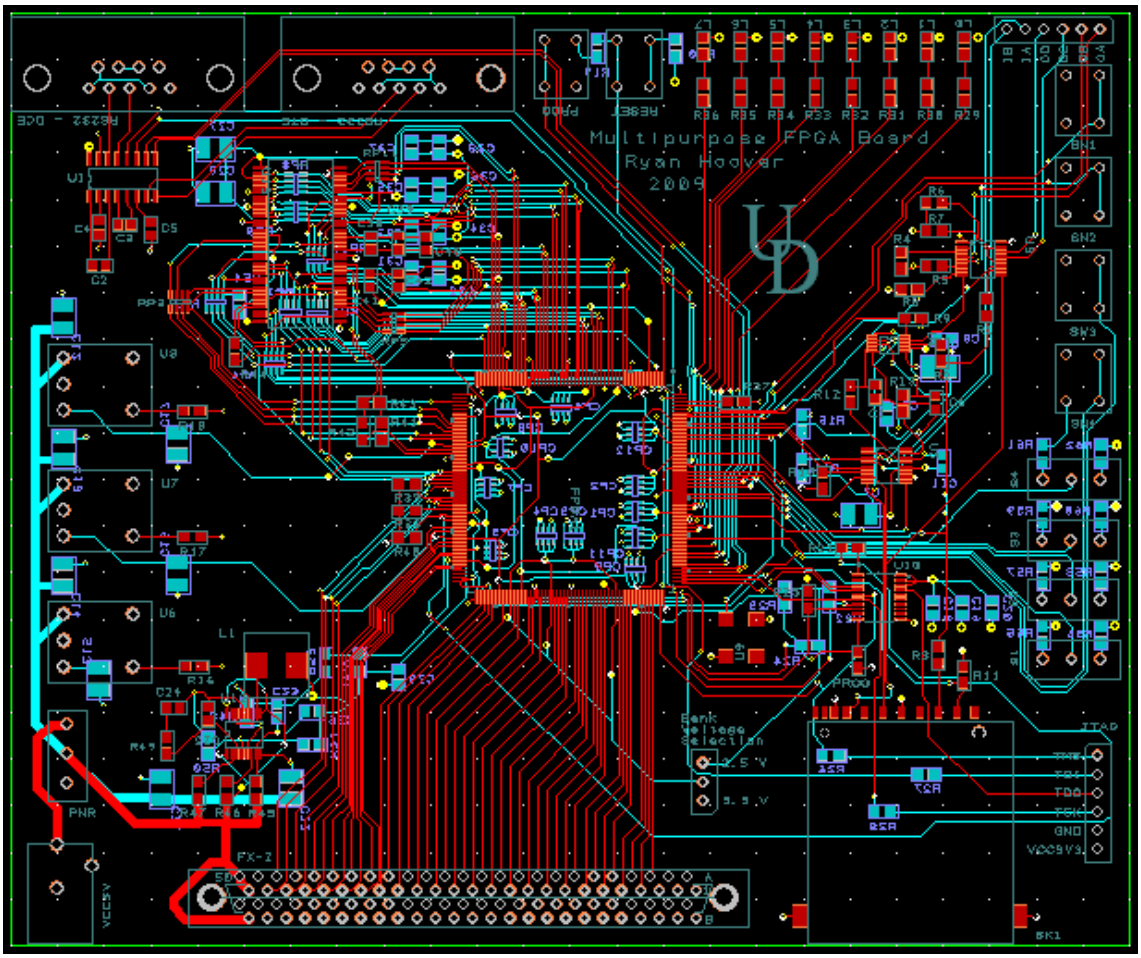

Figure 4.11: PCB Artist - Complete Routed Board Example

click *Output - Submit Order*. This will run several checks on the board, and finally bring up an order submission window that looks similar tot Figure 4.12. Depending on the board, direct submission using the software may or may not be possible. In this case, the board requires special attention due to the use of a promotion code '66each'. For this board, the 'Email Order' button will be used, which will package the files necessary and e-mail them to the appropriate representative, who will make contact with the designer to finalize submission. Once the an order for a particular board has been placed with Advanced Circuits, they will provide the 'Gerber' files necessary for submitting the design to another foundry, if necessary.

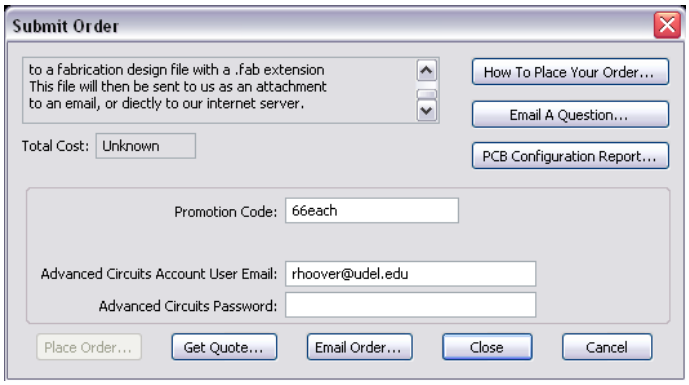

Figure 4.12: PCB Artist - Order Submission Form

4.2 Fabricated Design Notes

The limitations with fabrication using PCB Artist are mainly that the tool uses a proprietary type of file format that cannot be sent to any foundry but Advanced Circuits. However, the company will give the designer the files necessary to send to another foundry once the board has been fabricated at their facilities a certain number of times. The fabrication process at Advanced Circuits is detailed on their website, but no information is available as to the translation of the proprietary board files to the equipment which replicates the physical PCB.

4.3 Remarks on Advanced Circuits PCB Artist

Overall, Advanced Circuits PCB Artist is rather robust for free software, and provides a rather pleasant interface to work with. The tool is lacking in many advanced features, and has a few user interface quirks. For a beginning designer or an advanced designer creating a board of this type, the tool is great. Advanced Circuits also continues to add features, and provides excellent customer suppport considering the software is free.

Chapter 5

COMPARATIVE ANALYSIS

This section of the thesis will give a background of each software packages, and the advantages and disadvantages of each. Then, the conclusions which can be drawn from this information will be discussed, as well as which software package is the best choice for a particular project or situation.

5.1 Cadence Allegro 16.2

Cadence Design Systems was founded in 1988, and since then, the company has acquired a vast number of smaller companies that focused on specific issues in the design automation industry. As such, this software is a conglomeration of many utilities originally written and designed by other companies. In addition, there are many components that enhance specific portions of the experience if the user so desires, such as special routing method tools, and signal integrity analysis tools. The suite was also built with the development group, rather than a single user, in mind so it is easily adaptable to corporate environments. The package also includes various tools for reusing schematics and components, and also for streamlining the processes with large designs. The product consists of the following core components, shown in Table 5.1.

5.1.1 Key Advantages

The main advantage that the Cadence Allegro tools has over PCB Artist is the sheer number of features and options it possesses. Cadence Allegro contains

Table 5.1: Cadence Allegro Design Suite 16.2 Components

Component
Allegro Design Entry HDL
Allegro AMS Simulator
Allegro PCB Designer
Allegro PCB SI
Allegro PCB Librarian XL

vastly superior automatic routing tools, signal analysis tools, library management tools, planning tools, and reusability tools. The suite feels robust and well seasoned, aside from the occasional glitch. There is simply not many things that a designer could dream of doing in PCB creation that cannot be designed using Cadence.

5.1.2 Key Disadvantages

This robust functionality does not come without a price. The sheer number of tools in the suite create confusion and sometimes cause unnecessary time to be spent switching between tools to do what seems like a rather simple task. Furthermore, each tool feels very much like a part of a different software package. The keyboard shortcuts and menus are not consistent, and unnecessary complexity is added to the design process by having another utility to do design sync, requiring the user to close both applications, to perform synchronization. In addition, nothing about the design process is very intuitive, and good material documenting the process is not readily available without a price. Tasks like generating the final design files without a guide are nearly impossible to achieve, with no way of setting all options in one command to standard settings for standards like 'Gerber RS274X'.

5.2 Advanced Circuits PCB Artist

Though Advanced Circuits was founded in 1989, their PCB Artist software is relatively new, having only been released in August of 2007. Still, PCB Artist introduced many features which no other free PCB software can claim even presently.

PCB Artist was the first to provide free rudimentary automatic routing and forwards and backwards design synchronization. The software has been given extensive updates, fixes, and feature additions and enhancements since its launch, and has grown into a very capable PCB design package with a very big library of pre-designed components available.

5.2.1 Key Advantages

The greatest advantage that PCB Artist has over Cadence Allegro is the pure simplicity and low learning curve of the tool. A designer familiar with computer design software could easily work their way through the creation of a PCB board in the tool with no instruction or guidance. This is largely due to the single-application form factor of the application. In addition, the price barrier can keep many small businesses or companies from adopting the pricey Cadence Allegro suite.

5.2.2 Key Disadvantages

The limitations with regard to automatic routing and signal integrity analysis, as well as hardware description language integration and planning tools are the largest disadvantages of this tool. The other obvious issue with this tool is the lack of support for 'Gerber' file generation, though Advanced Circuits will provide these free after the first order of a particular design. This is primarily because the tool was created as a revenue generator for the company.

5.3 Conclusions

What has been learned in this process is that depending on the application, and constraints of the project, one tool can clearly outperform the other. For example, if working on a company project where signal integrity is mission critical, it would be more important, regardless of the initial cost, to use the more expensive Cadence Allegro software. Companies like NVIDIA, who develop boards in which

signal integrity is mission critical benefit tremendously from the offerings of a tool like Cadence. However, a smaller company or research group creating an FPGA or microcontroller-based board would most likely find that a simpler tool like PCB Artist will result in higher overall productivity, and reduce the time spent learning a tool and switching between advanced applications. Having used both tools to create the same design, it can be safely said that neither of the two software design tools are better than the other. Each offer their own advantages and disadvantages, and both have ample room to grow and develop.

Chapter 6

TESTING

When a PCB comes back from the foundry, it must be checked extensively to ensure that all of the components and connections are working within normal parameters. There are many facets to testing, from the moment the board arrives till every component is on the board, and it is time to power it on. This section of the thesis will describe the testing process, and the issues that came up during this process for this particular design.

6.1 Implementation

When the board comes back from the foundry, it look similar to the one below in Figure 6.1. Before beginning the process of soldering on parts, it is important to check the electrical continuity in the power planes, as well as discontinuity in any potentially troublesome signals (for example signals that are routed very close to one another, or close to a pin or pad). Though these signals should not be problematic as long as they passed the constraints test during DRC, there are sometimes mistakes that occur during fabrication that can cause these signals to have an electrical connection between them. Once this testing is complete, it is time to begin soldering on the components, starting from smallest and shortest to biggest and tallest. While soldering components such as 0805 resistors or capacitors or any other small part, it is very important to ensure discontinuity between the two pads of the component. It is very easy to accidentally bridge solder between the pads underneath the component, such that it will not be visible unless the component is removed.

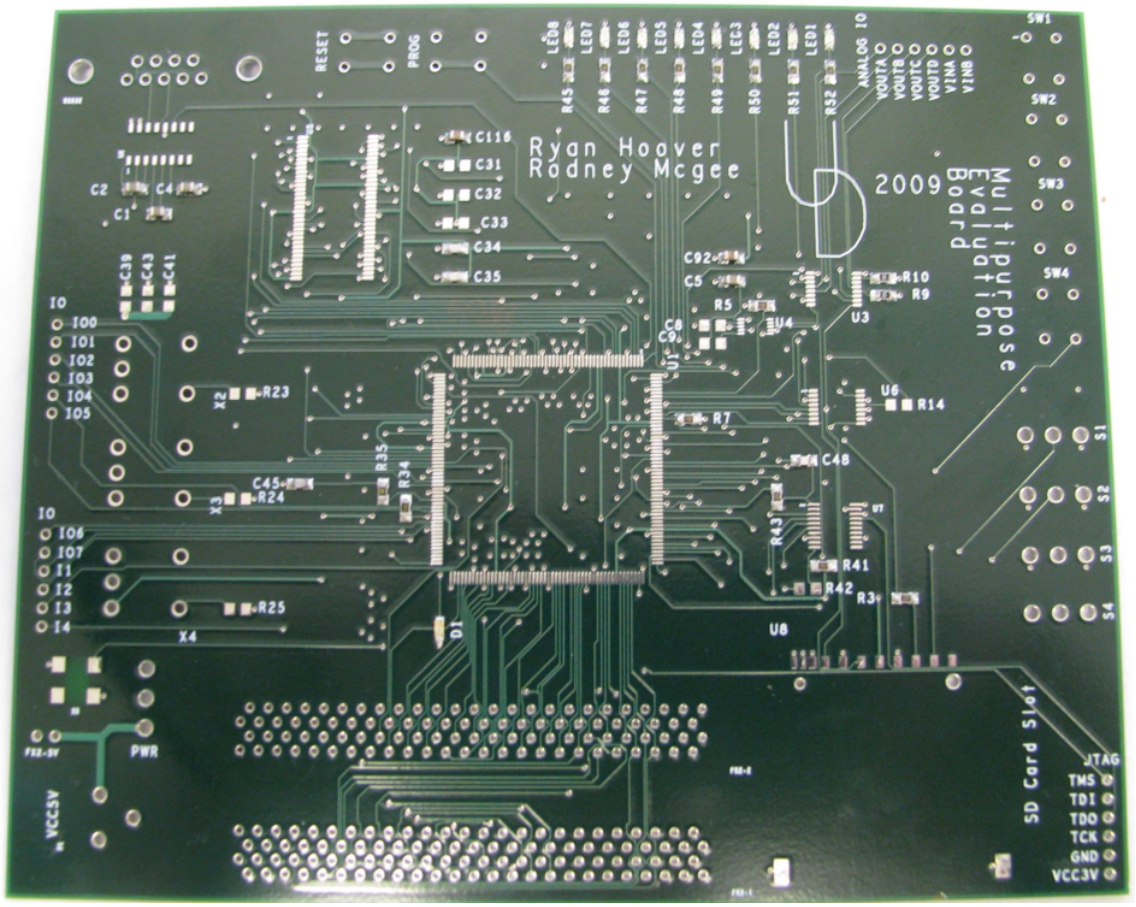

Figure 6.1: Cadence Early Design With Only Small SMT Parts Soldered

If this condition is not identified immediately, it can become more difficult as more components are soldered onto the board. If possible, in keeping with the smallest to largest part soldering order, solder on components that can be tested before soldering on other components. This way, if the dysfunctional condition of one group of components will prevent others from being tested, there is no reason to solder on any additional parts until the problem can be isolated and remedied. Finally, create simple test cases first, and work toward the more complicated functionality of the component. For example, write and read a one byte to a memory device before trying to work with a file system. This will save time, and help to solve potential problems quicker. When soldering on chips with small pins, be sure to carefully

check for discontinuity between pins, as it is also easy to bridge solder between two pins. When all components have been soldered, and the board is assembled, it will look similar to that below in Figure 6.2.

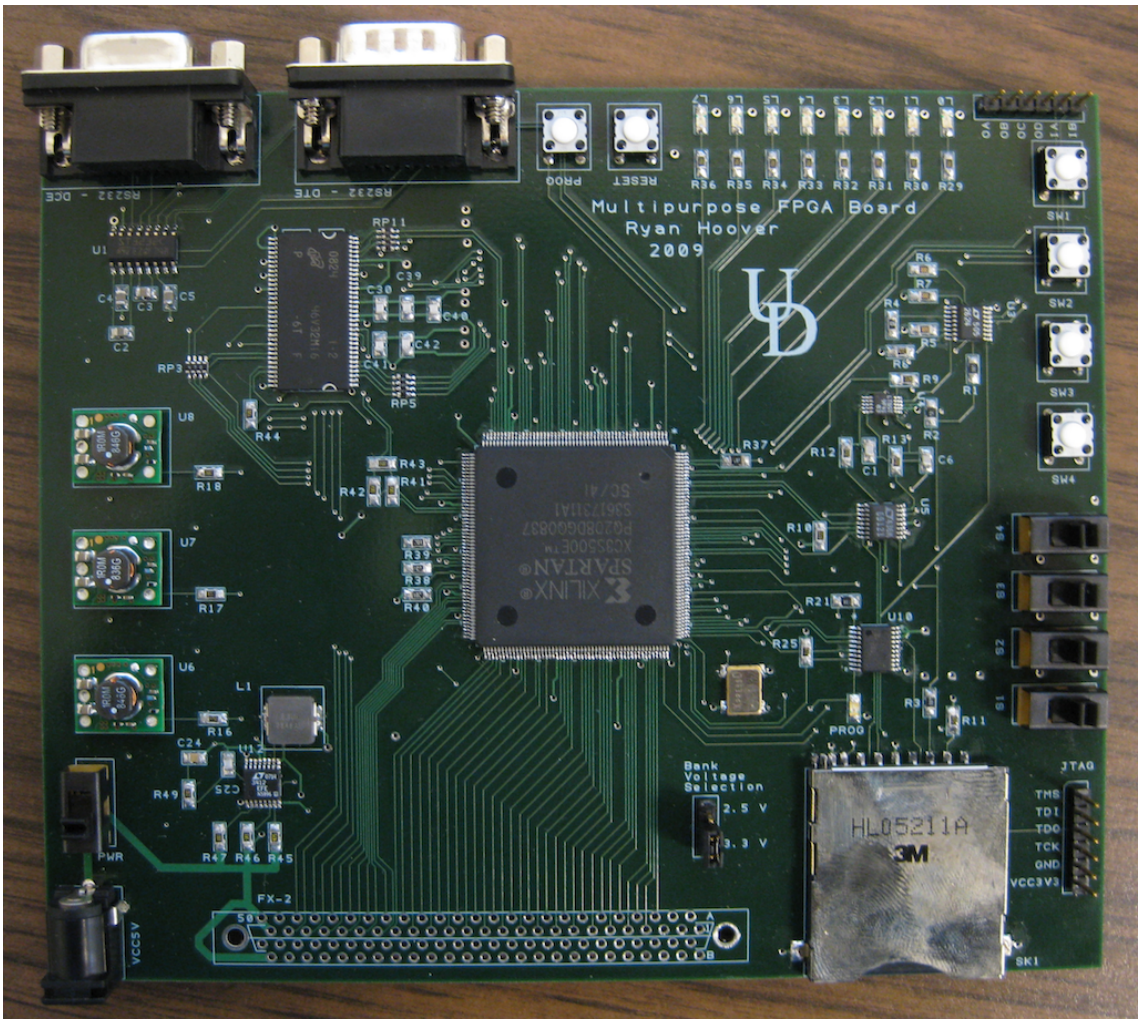

Figure 6.2: PCB Artist Board Mostly Assembled

At this point, it is important to test crucial components, such as the power supplies/converters to ensure that they are operating correctly, and at the right voltages. After successful power supply testing has been accomplished, the next logical step is to test communication with the microprocessor or FPGA through whatever communications method the board was designed to implement. If that

communication is not initially successful, check continuity between the correct pins on the device, and the pins to the interface. If the connections are OK, next connect contact points for those nets to an oscilloscope or logic analyzer to view the signals. Often the problem can be identified in this phase. Issues such as noisy connections, signal interference, or lack of proper supply to one of the devices in the communication chain can be easily identified. In most cases, a workaround such as cutting traces and running bus wire, or fixing a soldering issue, can fix the communications problem. Once successful communications have been established, it is best to begin with a simple test program or hardware design, such as one that controls the LED lights using the switches or buttons on the board. If this is successful, continue to test more complicated functionality component by component until each component achieves normal operating parameters.

6.2 Issues

When an issue arrises during testing, it is important to take several steps of action in order to diagnose and correct the cause of the issue. The first thing to check when working with a component is the power supply to the component. If the component was created in software by a third-party, or even by the designer, double-check the pin mappings, as well as the orientation in which the device was soldered to the PCB. Once this has been verified as correct, check the voltages at each of the power pins, including ground. If all power pins register at their correct values, move on to checking the lines of communication. Depending on the device, there may be a varying number of input signals. For the RS232 level shifter, as an example, the number of input signals is two per interface, with a maximum of four. Test the connection between the microprocessor or FPGA and the component, ensuring that connectivity is present. If the connections check out OK, next test the output signals and their destinations. Depending on the device, there may not be any output signals. At this point, if there still has not been an identified problem,

take a look at the signals with an oscilloscope/logic analyzer. Look specifically for signals that are not present or are duplicates of another signal, or for noisy signals. If none of these problems can be identified, draw out a timeline of how the signals should look, and ensure that the signals match in actuality. If everything still checks out, the component could very well be defective (assuming that there are no errors in source code or understanding of the communications protocol, and no design requirements which were not met such as routing length, etc.). During this process, it is important to never discount a possibility without first testing it. Mistakes happen, and often something as simple as a bad solder connection or a part soldered the wrong way will be the source of the issue.

Some of the issues encountered during development of this specific board were related to the communal design using Cadence. Several people contributed parts to the design, and when an error was made, it was not discovered until it was too late. One such error was the naming of global pins. In the part, one of the power pins was assigned to a net named 'GROUND' while all other parts and connections in the design were using a net named 'GND'. Therefore, this one pin was overlooked during routing (because an automatic router was used) and the pin was never connected because there was no other net in the design named 'GROUND'. This resulted in an ungrounded part, and manifested itself as some very noisy signals. To remedy this problem, the pin was attached, via bus wire, to the nearest ground pin. Another issue that was encountered with the Cadence design was a JTAG signal that was routed 6 mils apart from another line. The first board that was tested, because of tolerances or manufacturing defects, had significant electrical conductivity between two signals. However, another of identical design did not have this same conductivity. To remedy this situation, the traces had to be cut in several places, and the signal rerouted around the trouble region. This solved the JTAG communication issue. On the PCB Artist rendition of this design, an issue was encountered where the output signals

that were expected were not present at the labeled pins. Upon further investigation during a continuity check, it was discovered that the silkscreen labels were simply on the wrong side of the header, and there was never an issue with communication or the part. This was due to a last minute design change were a header was flipped in order to provide better access to the pins for a nearby component. However, the silkscreen labels for the header were never flipped to match this design change.

6.2.1 DDR RAM

The issues faced in attaining correct operation of the DDR RAM in this design were extensive. Several versions and revisions of the board were made before the revelation that certain design constraints specific to the particular package of the FPGA were not being met. Specifically, the FPGA's internal components had to be connected to specific pins in a specific order in a single bank to the DDR RAM in order for certain timing constraints to be met. Because of the decision to mirror the design used in the Spartan 3E board, these constraints were overlooked. Because the FPGA used in the Spartan 3E was a BGA or Ball Grid Array package, it contained more available input/output pins per bank, and therefore was capable of supporting the DDR-RAM that used a 16-bit data bus. However, because there are less input/output pins available per bank on the PQ208 package, it is simply not possible to support a 16-bit bus whilst satisfying the timing constraints associated with the internal components of the FPGA. Therefore, significant adjustments to the design had to be made for the last revision thus far.

6.2.2 Small Parts

In the original design, created using Cadence Allegro, 8-array resistor packs were used to reduce the number and size required for parts where there was to be a high volume of resistors in a confined space. However, the resistor packs that were chosen first were of a very small pitch, shown below in Figure 6.3 as compared to the

4-array resistor packs used in the most recent revision of the design. It is obvious

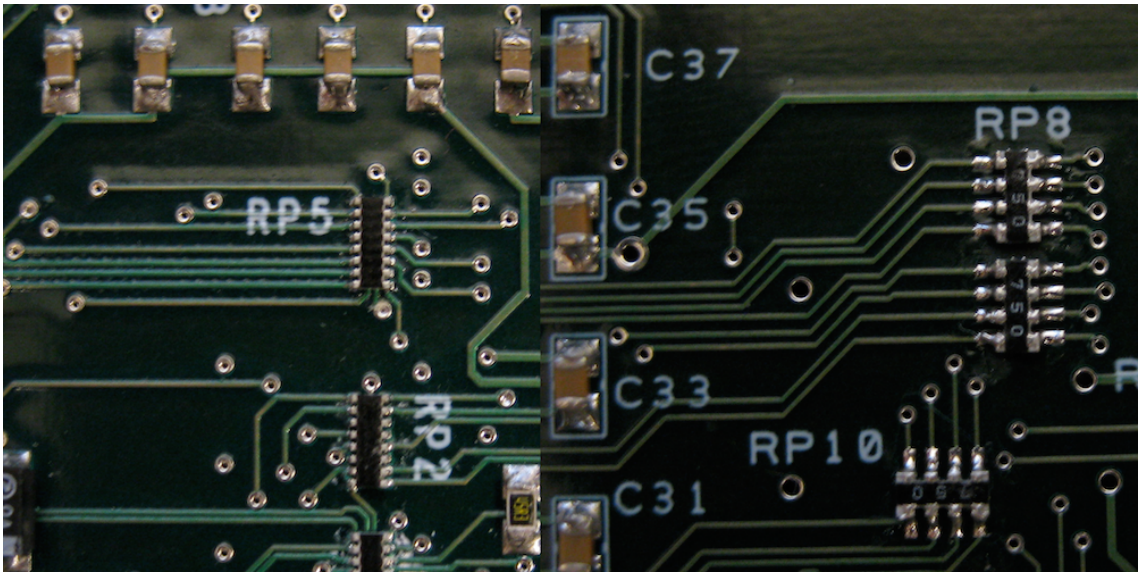

Figure 6.3: 8-Array (Left) VS 4-Array (Right) Size Comparison

how much smaller the resistor packs are in general when compared to the adjacent 0805 components. The 8-array resistor packs on the left presented much difficulty, and in several cases, bridges formed underneath the arrays, and the array had to be removed to eliminate the bridge. During the removal, some of the pads were pulled up as well, making it even more difficult than ever to re-solder. In addition, it was very easy to accidentally use too much solder paste, and form bridges between the tiny pins.

Chapter 7

CONCLUSIONS AND FUTURE WORK

Throughout the design of this board, there were many unanticipated challenges which one design package handled better than the other. With Cadence Allegro, it seems that the drive to create a comprehensive software package eradicated the ease of use and simplicity of the software. While with PCB Artist, it seems that while many tasks and operations are easy to use and simple, there are many features that are lacking, preventing the tool from being taken seriously by those that require such features. It can only be concluded that both tools have ample room for improvement before they can be considered powerful enough for extremely complex board design, yet simple enough for a beginner to learn quickly. It is these traits that software engineers in all markets strive to accomplish, yet rarely are able to.

The board designed in this project was built for testing a specific IC also designed at the University of Delaware. This board will be used in future testing, and most likely modified to suite other purposes as the project develops. The board is also suitable for a variety of other purposes, and will perhaps be used as a beginning template for other needs as well.

BIBLIOGRAPHY

[1] Wikipedia, C., *Printed Circuit Board*, Retrieved September 2009, from Wikipedia: `http://en.wikipedia.org/wiki/Printed_circuit_board`

[2] Wikipedia, C., *Cadence Design Systems*, Retrieved September 2009, from Wikipedia: `http://en.wikipedia.org/wiki/Cadence_Design_Systems`

[3] *Advanced Circuits Introduces Free PCB Layout Software*, Business Wire, 2007, Retrieved October, 2009 from accessmylibrary: `http://www.accessmylibrary.com/coms2/summary_0286-32736590_ITM`

[4] WestDev, Ltd, *PCB Artist Tutorial*, Advanced Circuits, Retrieved October 2009 from `http://www.4pcb.com/downloads/pcbartist/PCBArtistTutorial.pdf`

[5] *Allegro© Design Entry HDL Front-to-Back Flow*, Cadence Design Systems, February 14, 2008, Retrieved March 2009 from Cadence Design Systems