PCB DESIGN AND SIMULATION USING CADENCE ALLEGRO 15.5

BY

SAMUEL P. KUO

B.S., Rutgers University, 2004

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois

To Father and Mother

ACKNOWLEDGMENTS

I am indebted to Dr. José Schutt-Ainé for advising me in this project, for being patient with my progress, and for understanding the challenges this endeavor has entailed. It has been a pleasure to work in his group. Many thanks to my colleagues and (more so) friends, Pavle Milošević, Yidnek Mekkonen, Dr. Zhi-Chao Deng, and Dr. Ki-Hyuk Kim, for their quick assistance in tackling the various aspects of writing this thesis. The 371 corner of Everitt Lab will be missed.

I would like to thank the Department of Electrical and Computer Engineering of the University of Illinois at Urbana-Champaign for awarding me a teaching assistantship all four of my semesters. The financial support is received with much gratitude. Likewise, I would like to thank the EWS staff for helping with the Cadence licensing issues.

Finally, I would like to thank my parents, for being living examples of where diligence and persistence can take a person. And thank you for your encouragements along this journey—they have been timely.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Background

In modern high-speed circuits, printed circuit board (PCB) design is hardly mentioned without consideration of signal integrity (SI) analysis. As data rates increase and miniaturization improves in each generation of circuits, signal integrity becomes an integral part of the design process. An otherwise working topology may be required to undergo continuous iterations—traces rerouted, vias displaced, conductor widths adjusted, stub lengths tweaked, and dielectrics modified—in an effort to meet the required specifications.

Fortunately, high-speed circuit board designers have a variety of tools to simulate and analyze how signals propagate through a given circuit board. This aids the designer in making well-informed adjustments in lieu of signal integrity compromises. Therefore, learning these electronic design automation (EDA) tools are essential to producing operational, high-speed, PCB designs.

1.2 Purpose and Motivation

One such EDA simulation tool is the Cadence Silicon-Package-Board (SPB or Allegro) software suite. This comprehensive software package spans virtually the entire circuit design process, save IC design—from schematic entry to package design to board layout. With this EDA tool as its focus, this thesis serves as an educational and learning tutorial on some of the most commonly used programs included in Cadence Allegro SPB 15.5. It should be noted that the following contents were written primarily to serve our UIUC research group in signal integrity as a starting point and stepping stone in their future work in designing high-speed mixed-signal devices like analog-to-digital converters (ADC).

Therefore, an example flash-type ADC circuit will be used throughout this thesis. The goal of future work is to eventually acquire chip-package codesign expertise. On top of ADC architecture and measurement research, the group requires adequate software design tool support—at least working knowledge of the current generation of tools like Cadence Allegro 15.5—in order to build future cutting-edge ADCs.

1.3 Contents

While other Cadence tools such as Virtuoso are commonly taught and used in undergraduate VLSI courses, Cadence Allegro frequents academia to a much lesser degree. Therefore the learning summarized in this thesis was learned nearly from scratch.

This thesis is not a know-all, end-all manual in learning these EDA tools, nor will it attempt to be. In truth, there are thousands of pages of user's guides and software manuals one could read that are already included with the software suite. However, this tutorial does seek to give a neophyte a certain competency in operating this software—at the very least, in performing basic and common tasks needed in design, simulation, and analysis. Common procedures will be addressed and walked through while building a working 3-bit flash ADC. As a disclaimer, at the time of writing I still only have limited knowledge of these tools and do not presume to be able to fully educate anyone in them. However, this thesis is humbly presented as a starting point.

The three main sections of this tutorial expound through the three major steps of PCB design—schematic entry, PCB layout, and SI analysis.

Schematic entry refers to placement of electronic devices at the circuit level such as transistors, and passive and active components. This may also include available digital logic symbols such as decoders, multiplexers, etc. Verification of the logic and behavioral portion of the circuit is finalized at this level. Footprints of the component packages will also have to be assigned in order to have proper net listing and translation into PCB layout tools. This part of the thesis will focus on using Design Entry CIS 15.5 as the schematic entry software and Allegro AMS Simulator as the simulator (formerly known as OrCad PSpice), both of which are part of the Cadence Allegro SPB 15.5 software suite.

PCB layout involves defining the substrate, metallization, number of layers, and the physical dimensions of the board. Placement of packages, and the routing of traces and interconnects constitute the rest of PCB layout. While this portion can be done with many tools, this tutorial will focus on Cadence Allegro PCB Editor to complete this design step. Whenever routing with Allegro PCB Editor, one will also be commonly interfacing with two other programs. Constraint Manager is used to specify timing and electrical constraints on each of the nets to be routed. SigXplorer is a topological extraction and modification tool. Both of these complementary software programs will also be touched upon.

The SI analysis portion of this thesis will also involve using Cadence Allegro PCB SI for EM simulation. There is a brief overview on SI modeling. SigXplorer is again frequently used here, and will be revisited. Eye diagram simulations of PCB nets and other SI simulation reports are discussed. SigWave is used as the waveform viewer.

1.4 Cadence Allegro 15.5 Installation

Installation of Cadence Allegro 15.5 should be straightforward. If one does not have the CDs, CD image files for installation can be found at http://download.cadence.com. If the user attends the University of Illinois at Urbana-Champaign, to receive the school login to this website please contact the ECE help desk or network administrator. Upon logging in to this download site, choose the operating system or platform of your computer. Then select and download the package named “SPB1551.” SPB, again, stands for silicon-package-board. Thus, the Cadence Allegro 15.5.1 software suite is also known as Cadence SPB 15.5.1. In this thesis, this software version (15.5.1) will be used interchangeably with version 15.5. Version 15.5.1 adds additional eye diagram features that make plotting eye diagrams easier. This is covered in Chapter 5.

During installation a valid license server will be requested of the user. This license server information can again be obtained from the ECE help desk or network administrator, if one is a UIUC ECE student.

It should be noted that in the Windows® operating system installation, it will only require and ask for the first three CDs. The fourth CD has additional libraries that can be installed on its own if one so desires. It is recommended that this fourth CD be installed, especially for Design Entry HDL libraries.

It will be assumed for the remainder of the thesis that the reader uses the Windows® operating system installation of Cadence Allegro. Screenshots and step-by-step directions follow from this version.

2 OVERVIEW OF PCB DESIGN FLOW

2.1 Integrated Circuit to Printed Circuit Board Flow

Though the specific tools used in this thesis have been previously stated in Section 1.4, it is worthwhile to take a step back and view the entire Cadence flow from IC to package to board. That way, the user can understand the functionality of Cadence Allegro amidst this flow.

In essence, this thesis focuses on PCB design (the green portion of Figure 1). However, Cadence Allegro SPB 15.5 is also capable of package design (the orange portion of Figure 1). Typically in the complete design flow, an organization would use Cadence Virtuoso to lay out and verify all of the chip-level designs and then migrate that design along the package and PCB flow.

2.2 Cadence Allegro SPB 15.5 Program Suite Description

Figure 2 displays the individual programs of Cadence Allegro SPB 15.5 and how they accomplish both package and PCB design. Additional comments were also added. When starting out, this should give the user a high level description of what each program does.

2.3 Flash Analog to Digital Converter Overview

In order to sufficiently exercise the Cadence Allegro tools, it would be best to build a simple yet useful circuit and complete a full design cycle. Since this work is related to high-speed ADCs, it is fitting to design one. Therefore, a 3-bit flash ADC was selected to be designed. The architecture is also very convenient for a board design, since the resistors, comparators, and encoder are easily obtained in packaged chips.

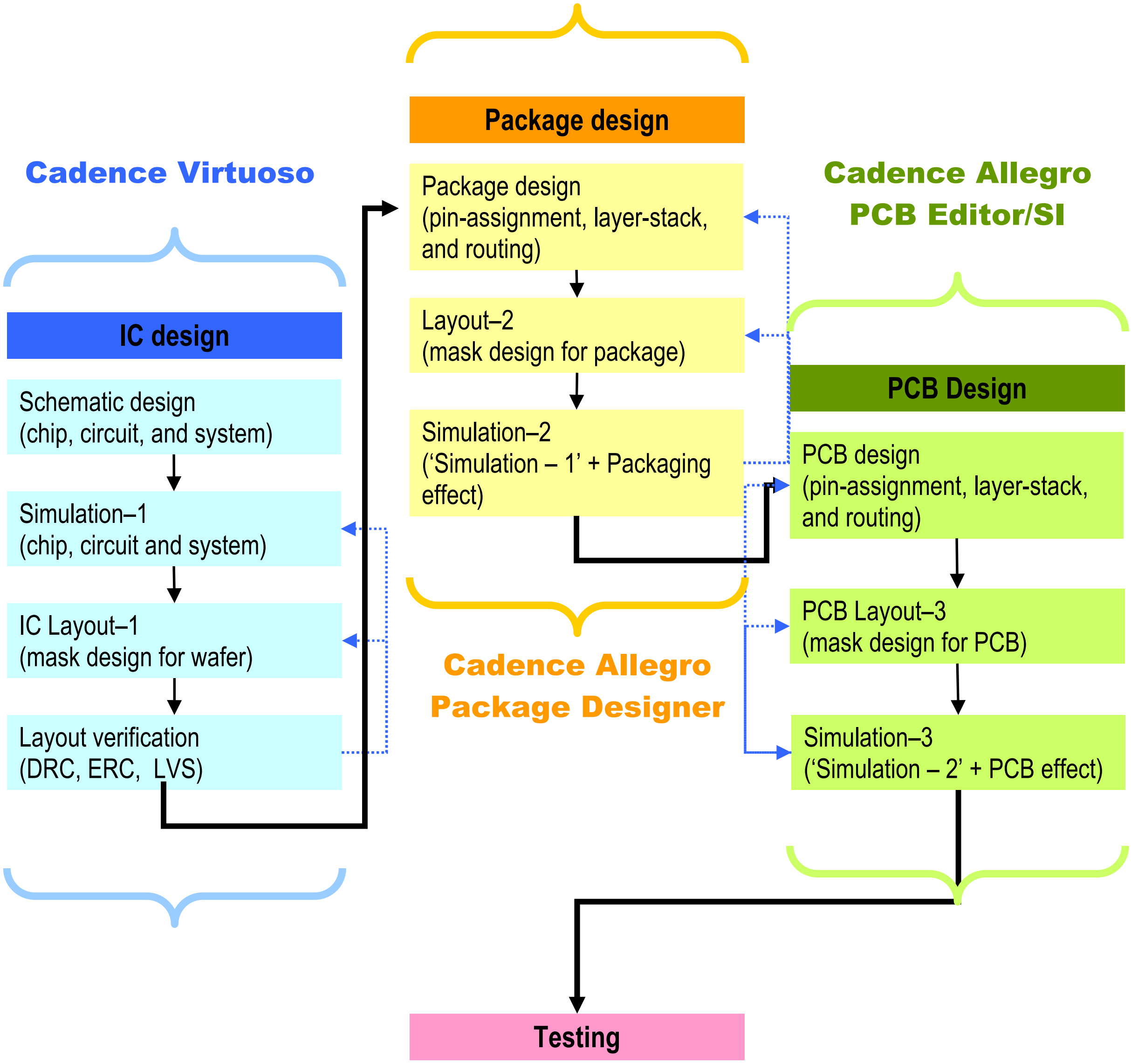


Figure 1 - Cadence IC to Package to PCB Flow [1]

The architecture is fairly simple to understand. An N-bit flash ADC requires 2^N reference voltages and 2^{N-1} voltage comparators, which are then fed into a priority line encoder to encode the digital bits [2]. The digital reading that serves as an input to the encoder is often known as a "digital thermometer code" since the digital "1"s rise as the input voltage rises [3]. A three-bit topology is illustrated in Figure 3.

The priority line encoder encodes the digital output giving priority to the most significant input bit with a "1." For example, a "00001111" would encode as "100," and even a glitch like "00101111" would encode as "110." This is something called a "sparkle code" where a spurious "0" produces a discontinuity within an expected string of "1"s. This is usually resolved with a track-and-hold amplifer at the input of the ADC [3]. However, we will take the simplest case and produce a schematic and PCB resembling Figure 3.

The rest of the thesis comprises the "how-to" of building this 3-bit ADC converter in Cadence Allegro.

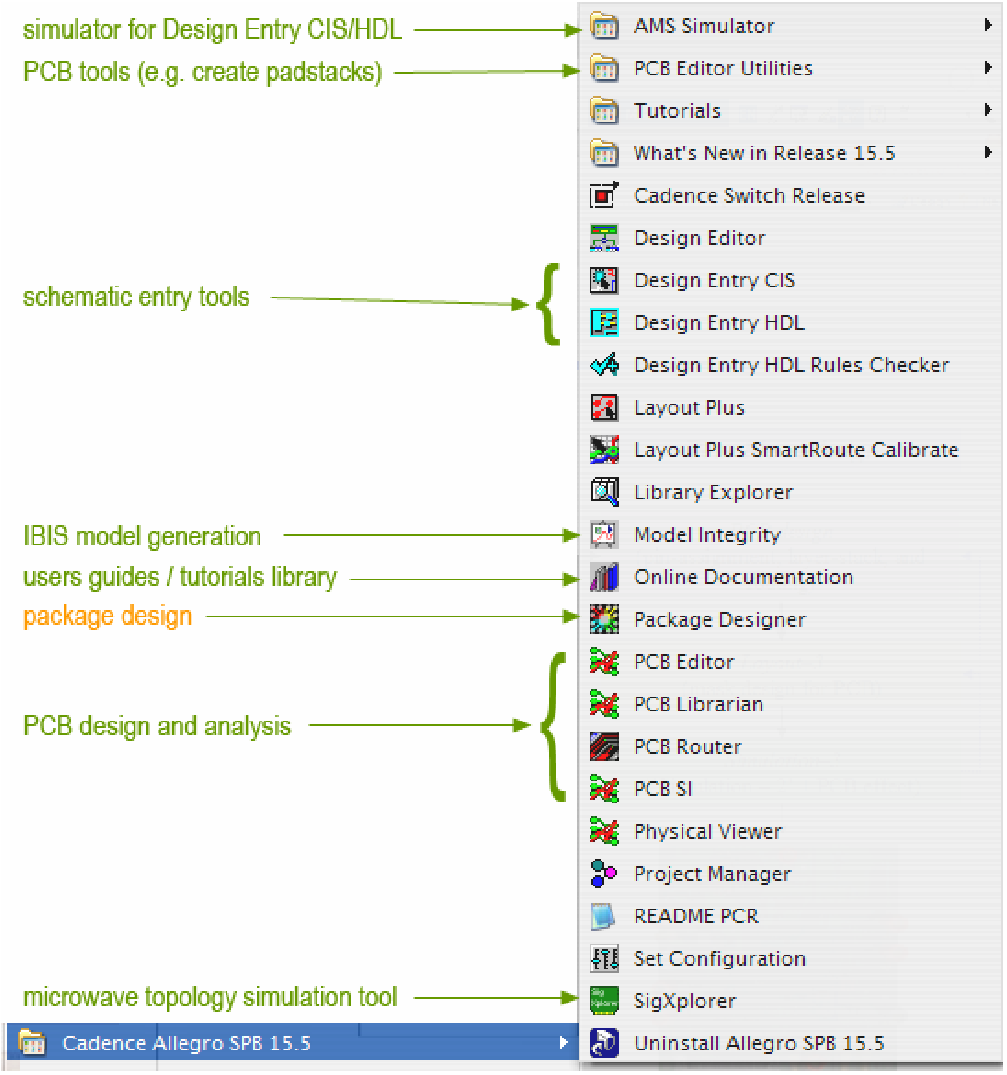


Figure 2 - Cadence Allegro SPB 15.5 Components

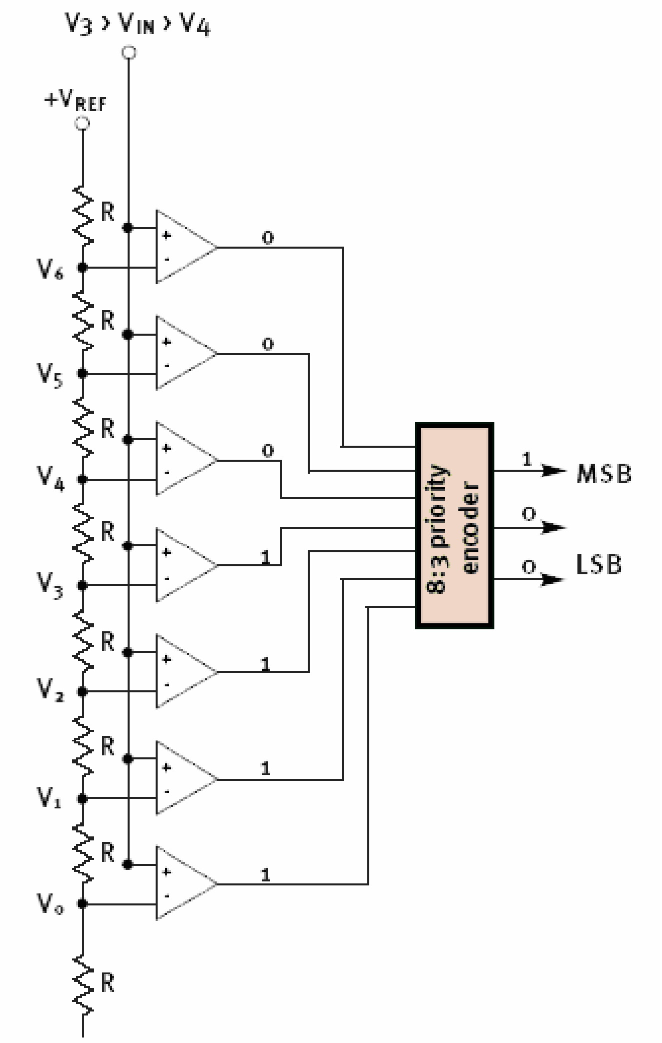


Figure 3 - Three-Bit Parallel Flash ADC Architecture [2]

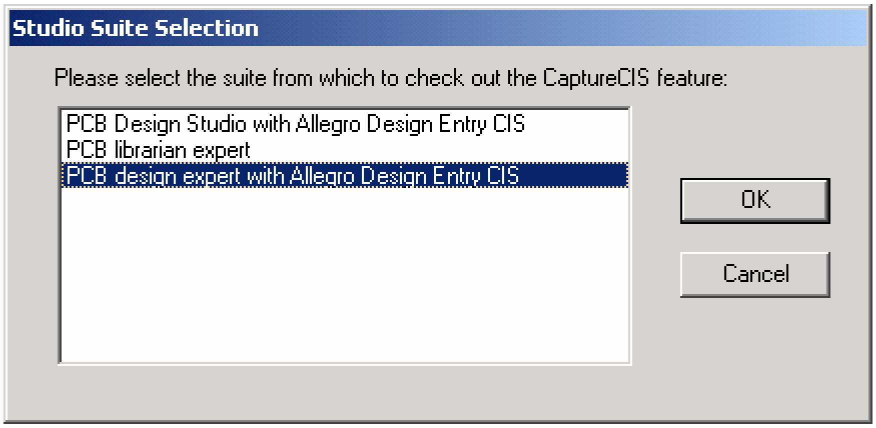
3 SCHEMATIC GENERATION USING DESIGN ENTRY CIS

3.1 Overview

This section focuses on realizing the flash ADC topology in a schematic using Allegro Design Entry CIS. A fully working 3-bit ADC circuit will be built with available parts (off the shelf), and then simulated at 10 kHz and 300 kHz for verification and to stress the circuit. Although in the current state of the art, most ADCs are built fully *on-chip* in an integrated circuit (which would require IC layout tools like Cadence Virtuoso), we will build an ADC *on-board* from separate chips/packages in order to exercise the PCB layout and simulation tools (Allegro PCB Editor and Allegro PCB SI, covered in Chapters 4 and 5).

3.2 Creating the Schematic

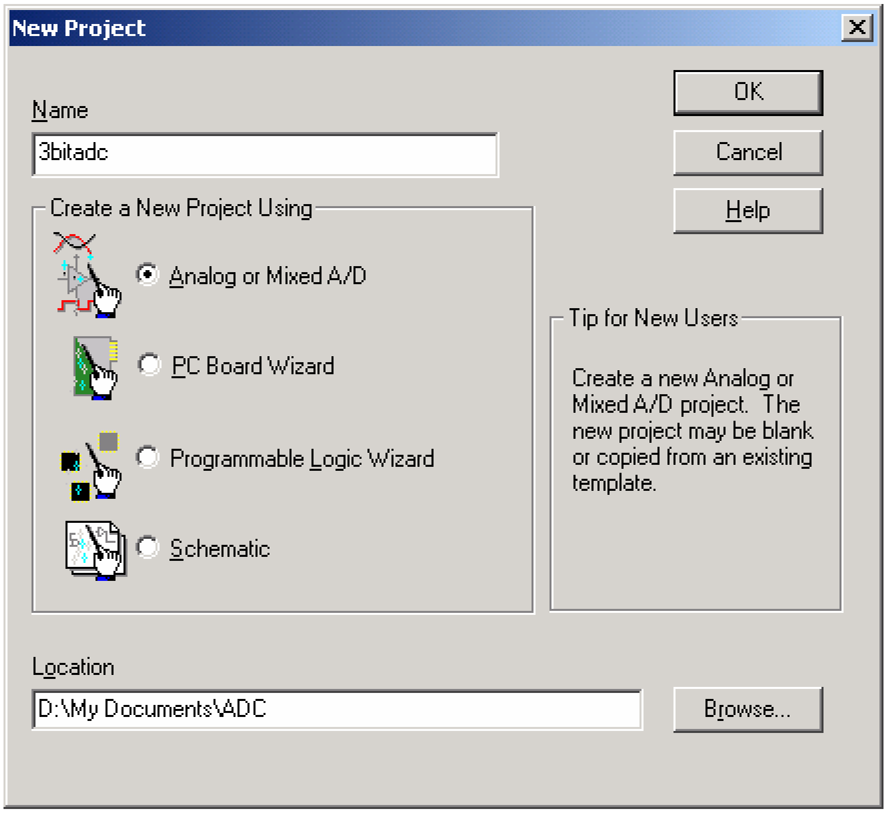
1. First we need to execute the Design Entry CIS program. This is usually found by clicking *Start* → *All Programs* → *Allegro SPB 15.5* → *Design Entry CIS*. If there are no licensing problems, there should be a pop-up that allows you to select one of the many versions that we are licensed to use. Select "PCB design expert with Allegro Design Entry CIS" – the most comprehensive package (Figure 4).
2. The first thing we would like to do is create a new project. Be sure to select "Analog or Mixed A/D" project so that any future schematics can be simulated (Figure 5). Additionally, in the "Location" field, select a working directory of your choice. Note that this Design Entry CIS design file (.dsn) and its future Allegro board file (.brd) will be found in this directory or subdirectories.



Please select the suite from which to check out the CaptureCIS feature:

PCB Design Studio with Allegro Design Entry CIS
PCB librarian expert
PCB design expert with Allegro Design Entry CIS

Figure 4 - Design Entry CIS Studio Suite Selection



Name: 3bitadc

Create a New Project Using
● Analog or Mixed A/D
○ PC Board Wizard
○ Programmable Logic Wizard
○ Schematic

Tip for New Users

Create a new Analog or Mixed A/D project. The new project may be blank or copied from an existing template.

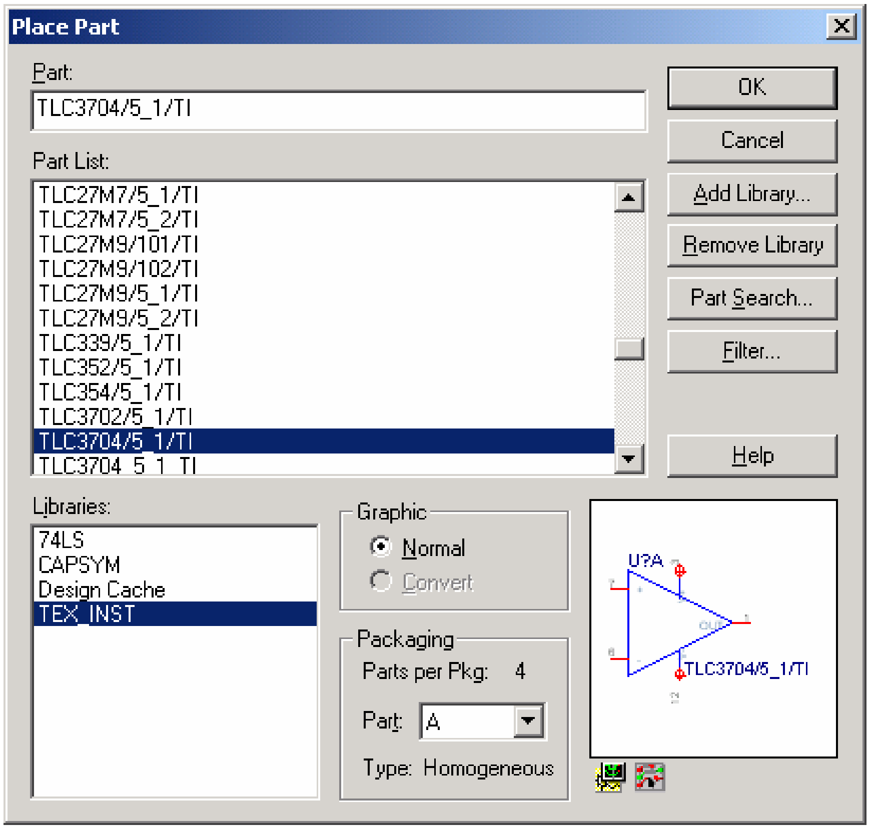
Location: D:\My Documents\ADC

Figure 5 - New Project Pop-Up

3. Since our ADC schematic would fit better on one page if it were portrait oriented rather than landscape oriented, we need to change the dimensions of the page. To do this we select the menus *Options* → *Schematic Page Properties* and then change the

dimensions manually. If desired, customize the colors and user interface by choosing *Options* → *Preferences*.

4. In order to place off-the-shelf available parts, we need to add the appropriate libraries. Docked on the right side of the window are tools to modify the schematic. Find the one named *Place Parts* → *Add Library*, and then select the libraries found in "pspice/source.olb" "pspice/TEXT_INST.olb" and "pspice/74ls.olb." This will enable us to place and simulate Texas Instrument devices, the common 74LS transistor-transistor logic chips, as well as common voltage sources for simulation (Figure 6).
5. The core of this flash ADC architecture are the parallel voltage comparisons made between the input signal and several reference voltages [1]. This correlates to the "digital thermometer" mentioned in Chapter 2 [1]. In our topology this is done by eight voltage comparators eventually resulting in three digital bits. We will use the Texas Instruments TLC3704 low power CMOS analog comparator chip for this function. Upon selecting this part, notice in Figure 6 how it is stated that there are four parts per package, meaning according to the pin-out, there are four comparators in each chip. Therefore, we need two total packages for all the comparators. Each comparator in the package is uniquely identified by a letter A through D.
6. In the 74LS library the chip for an 8 to 3 priority line encoder called 74LS148 can be found. Using this 74LS chip, the TI comparators, additional 1-kΩ resistors, and voltage sources, we can build the ADC schematic shown in Figures 7 and 8). For the input source, a 5-V_{pp} sinusoidal source with 2.5-V DC offset was used. This was chosen to make the circuit single supplied. V_{CC} was set as 5 V.



Part: TLC3704/5_1/TI

Part List:
- TLC27M7/5_1/TI
- TLC27M7/5_2/TI
- TLC27M9/101/TI
- TLC27M9/102/TI
- TLC27M9/5_1/TI
- TLC27M9/5_2/TI
- TLC339/5_1/TI
- TLC352/5_1/TI
- TLC354/5_1/TI
- TLC3702/5_1/TI
- TLC3704/5_1/TI
- TLC3704_5_1_TI

Libraries:
- 74LS
- CAPSYM
- Design Cache
- TEX_INST

Graphic: ● Normal ○ Convert

Packaging
Parts per Pkg: 4
Part: A
Type: Homogeneous

Figure 6 - Placing a Schematic Part after Library Additions

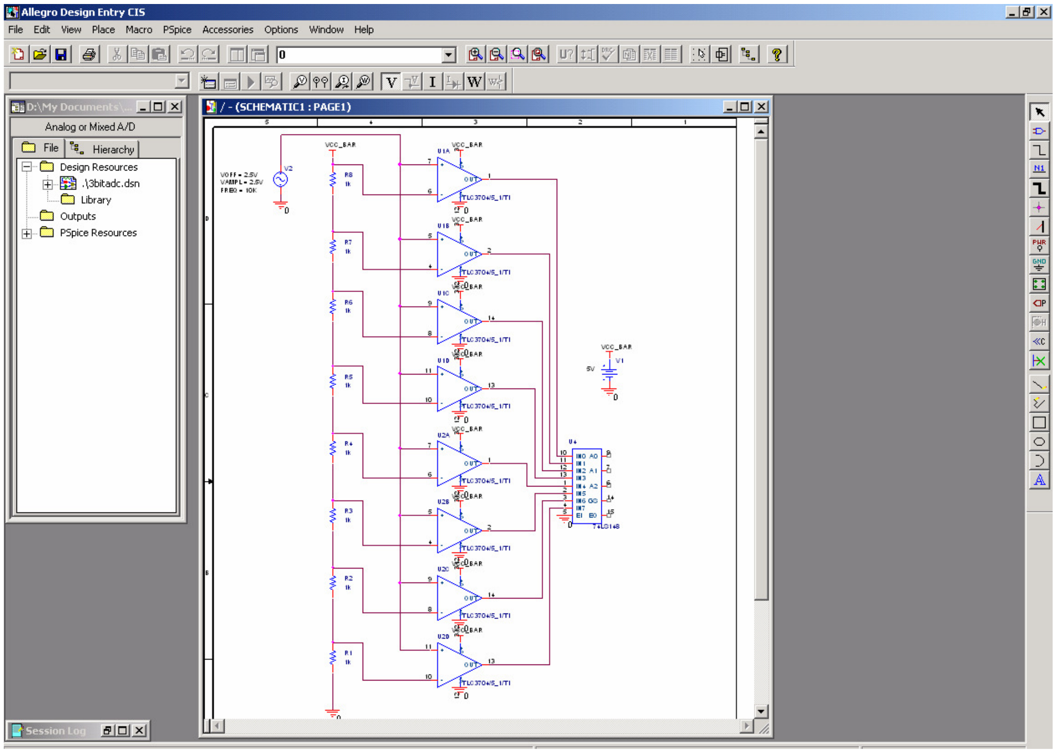


Figure 7 - ADC Schematic, Zoomed Out

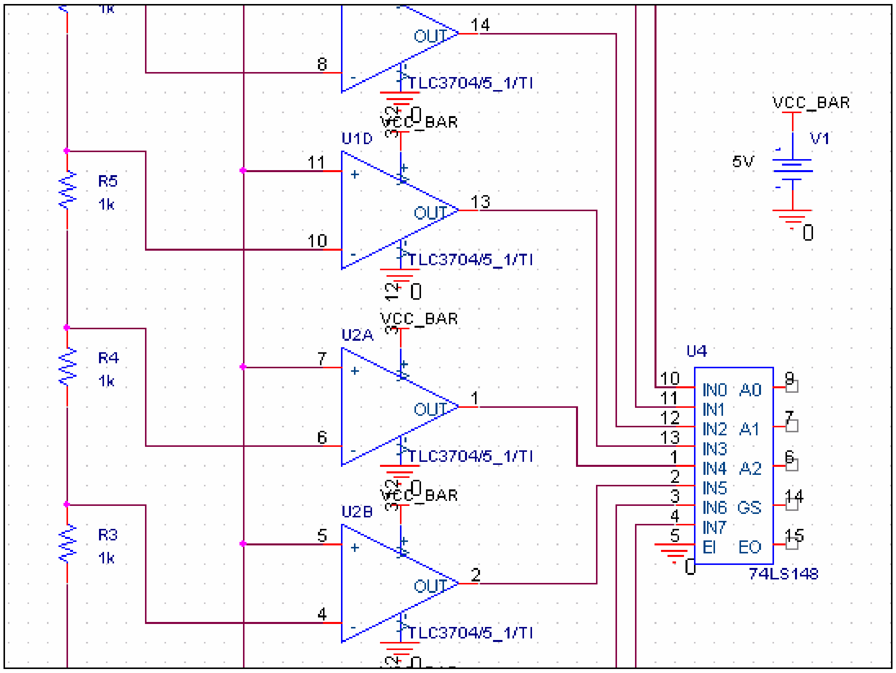


Figure 8 - ADC Schematic, Close-Up

7. After the schematic is ready, we need to verify the ADC's functionality by simulating the circuit. Create a new simulation profile by clicking the "New Simulation Profile" icon (Figure 9). Since the input voltage source being quantized was set to 10 kHz, the run-to time was set to 200 µs in order to see two full cycles of transient simulation (Figures 9 and 10). Use the "Voltage/Level Marker" tool to measure any relevant voltages necessary to verify the design. In this case, voltage probes are placed on the input voltage source, the outputs of all eight comparators, and the three output bits of the encoder.

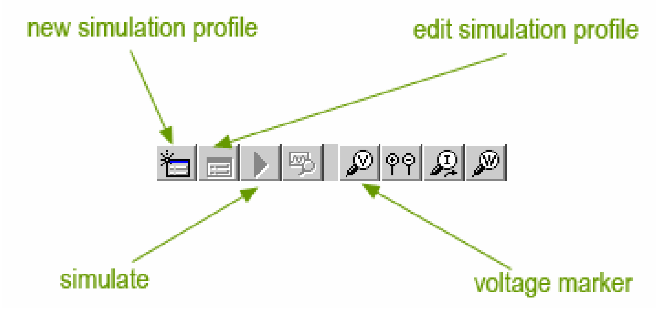
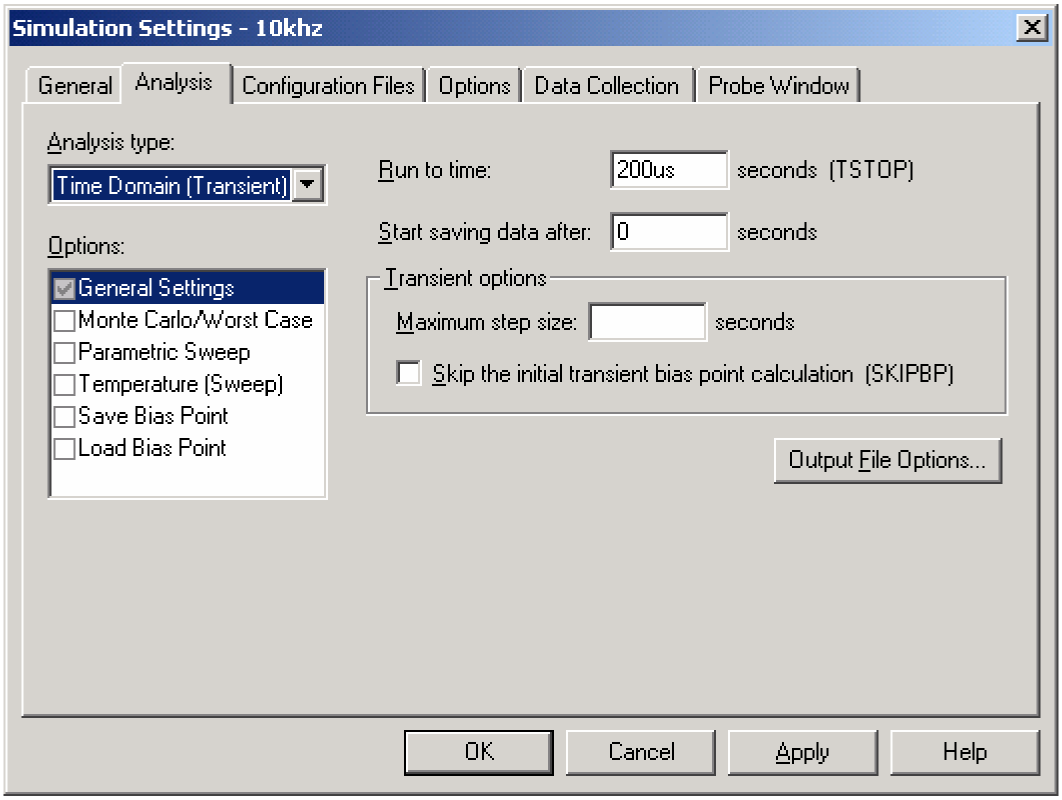


Figure 9 - Simulation Icons



Simulation Settings - 10khz

Analysis type: Time Domain (Transient)

Options:

☑ General Settings
□ Monte Carlo/Worst Case
□ Parametric Sweep
□ Temperature (Sweep)
□ Save Bias Point
□ Load Bias Point

Run to time: 200us seconds (TSTOP)

Start saving data after: 0 seconds

Transient options

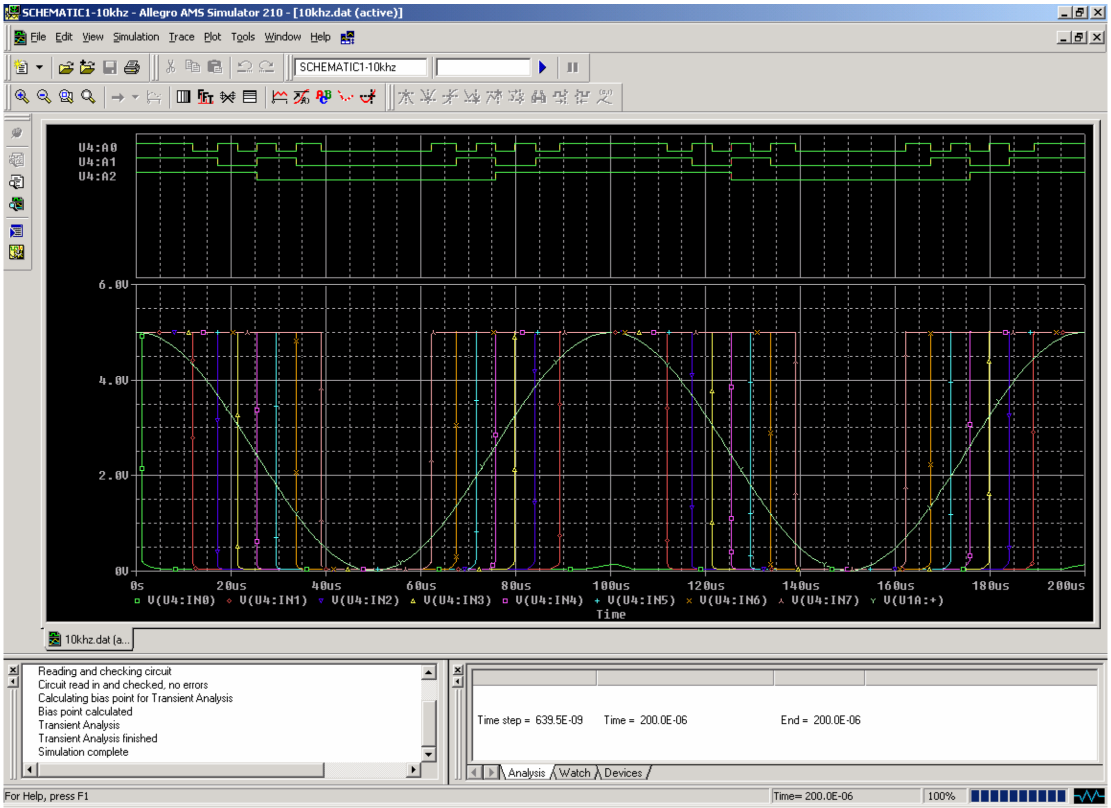
Maximum step size: seconds

□ Skip the initial transient bias point calculation (SKIPBP)

Figure 10 - Modifying the Simulation Settings

8. After setting up the simulation and voltage probes, click the “Run Pspice” icon (Figure 9) to simulate the design according the simulation profile. At this point in time, the program will attempt to netlist the schematic and, if successful, will simulate using a bundled program called Allegro AMS Simulator. If there are netlist errors, correct them and proceed accordingly. The errors reported in the “Session Log” window should be descriptive enough to correct if necessary.

Note how the simulation displays the analog input voltage and comparator output voltages in a separate graph from the output digital bits (Figure 11). After some inspection we see that the schematic works as designed.


SCHEMATIC1-10khz - Allegro AMS Simulator 210 - [10khz.dat (active)]
File Edit View Simulation Trace Plot Tools Window Help
SCHEMATIC1-10khz
U4:A0
U4:A1
U4:A2
6.0V
4.0V
2.0V
0V
0s
20us
40us
60us
80us
100us
120us
140us
160us
180us
200us
□ V(U4:IN0) ◇ V(U4:IN1) ▽ V(U4:IN2) △ V(U4:IN3) □ V(U4:IN4) + V(U4:IN5) × V(U4:IN6) ⅄ V(U4:IN7) Y V(U1A:+)
Time
10khz.dat (a...
Analysis
Watch
Devices
For Help, press F1
Time= 200.0E-06
100%

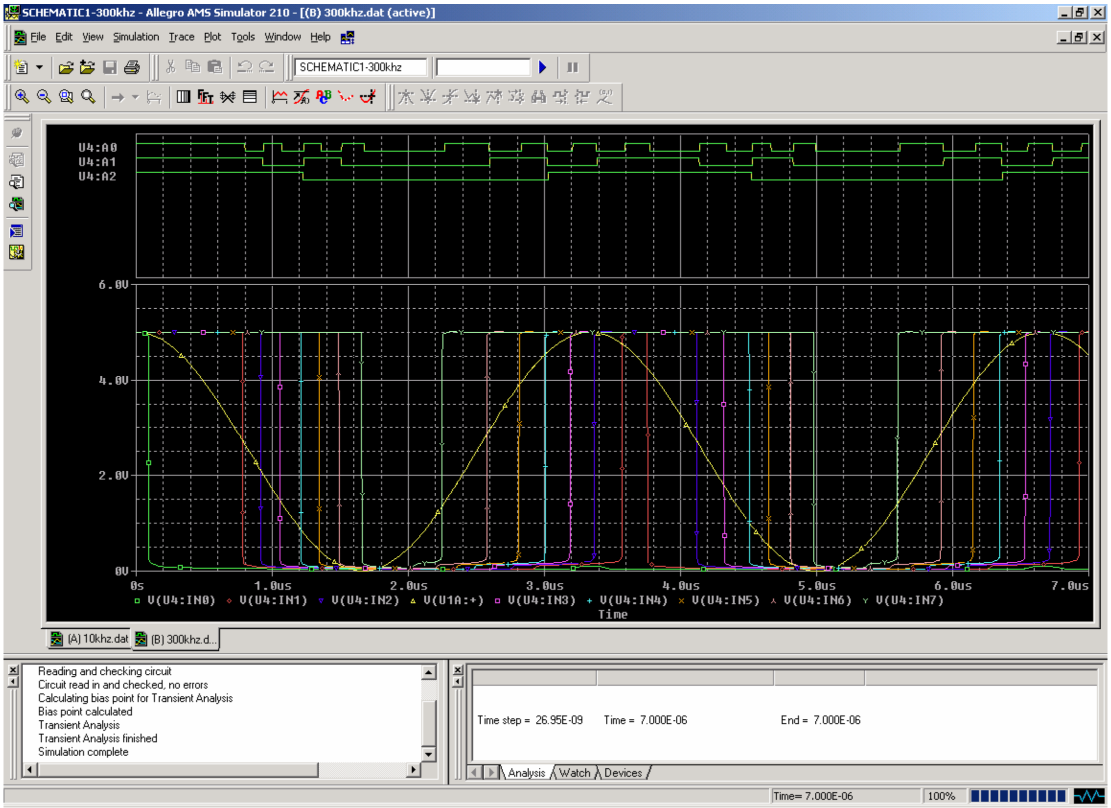
```
Reading and checking circuit
Circuit read in and checked, no errors
Calculating bias point for Transient Analysis
Bias point calculated
Transient Analysis
Transient Analysis finished
Simulation complete
```

```
Time step =  639.5E-09      Time =  200.0E-06                End =  200.0E-06
```

Figure 11 - ADC Simulation at 10 kHz

9. In order to stress this design and learn at what maximum analog frequency it can accurately convert into digital bits, we replace the original 10-kHz source with a new 300-kHz sinusoidal source. Additionally, a new simulation profile needs to be created to accommodate this. Namely, the simulation time was changed from 200 µs to 7 µs which would also display two full cycles of the sine wave.

The results are shown in Figure 12. Observe that the comparator delays are more pronounced in this faster simulation. The comparator output voltages are no longer centered around the sinusoidal signal. In addition, we see that the highest digital output of "111" from the ADC barely makes its appearance. Therefore, we can conclude that 300 kHz is approximately the maximum frequency of conversion.


SCHEMATIC1-300khz - Allegro AMS Simulator 210 - [(B) 300khz.dat (active)]
File Edit View Simulation Trace Plot Tools Window Help
SCHEMATIC1-300khz
U4:A0
U4:A1
U4:A2
6.0V
4.0V
2.0V
0V
0s
1.0us
2.0us
3.0us
4.0us
5.0us
6.0us
7.0us
V(U4:IN0) V(U4:IN1) V(U4:IN2) V(U1A:+) V(U4:IN3) V(U4:IN4) V(U4:IN5) V(U4:IN6) V(U4:IN7)
Time
(A) 10khz.dat
(B) 300khz.d...
Analysis
Watch
Devices
100%

```
Reading and checking circuit
Circuit read in and checked, no errors
Calculating bias point for Transient Analysis
Bias point calculated
Transient Analysis
Transient Analysis finished
Simulation complete
```

```
Time step = 26.95E-09    Time = 7.000E-06    End = 7.000E-06
```

```
Time= 7.000E-06
```

Figure 12 - ADC Simulation at 300 kHz

3.3 Going Further

It is valuable to note that Design Entry HDL is capable of performing the same functionality of schematic entry as Design Entry CIS. Since Design Entry HDL is native to the Allegro flow—whereas Design Entry CIS was later adopted into Allegro from Orcad—it may be worthwhile for the user to exercise Design Entry HDL as well.

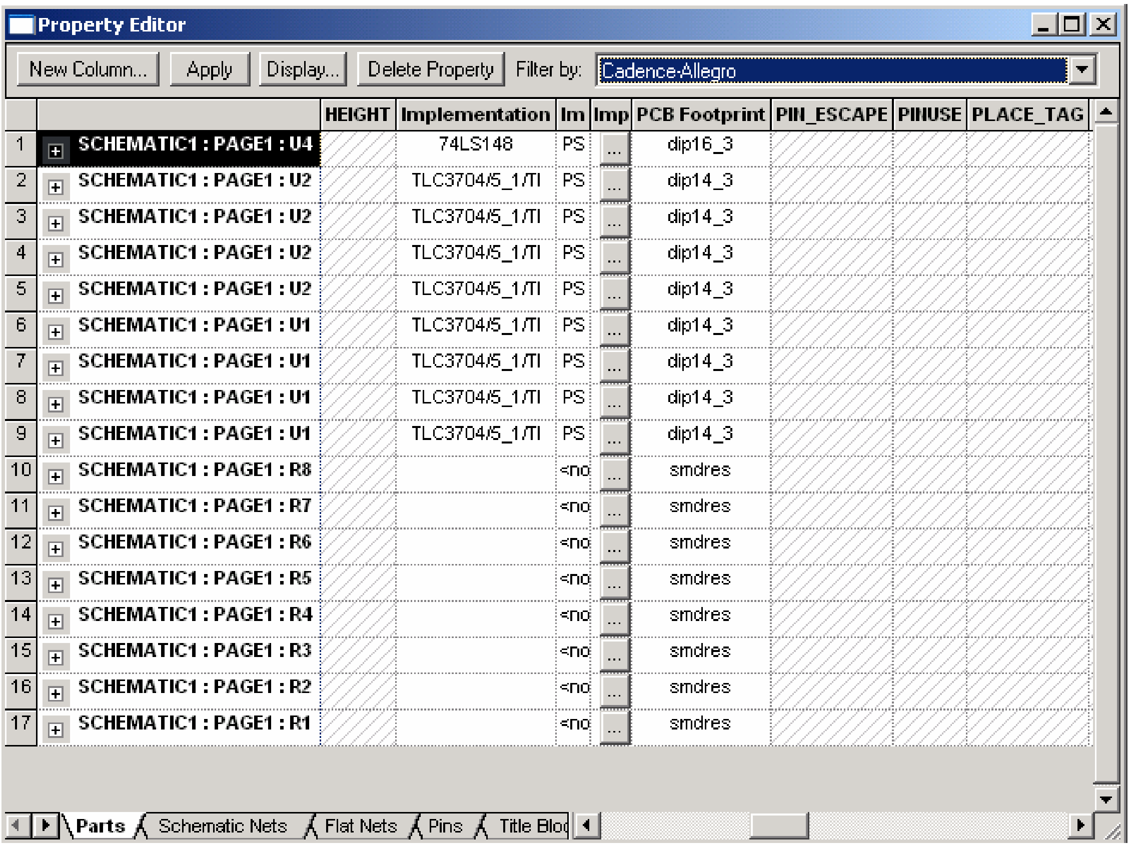
4 PCB LAYOUT USING ALLEGRO PCB EDITOR

4.1 Overview

In the previous section we have already verified the functionality of our ADC through simulation in Allegro AMS Simulator. In this chapter, we seek to translate our schematic into a routed printed circuit board using the appropriate footprints of each package. This section continues where Chapter 3 left off in Design Entry CIS, and eventually migrates into using Allegro PCB Editor. In this section we complete the full routing of a PCB.

4.2 Defining PCB Footprints

1. The first step in a successful migration into PCB routing is to define appropriate footprints for each of the components in our schematic. Using the *Ctrl* key and mouse to select multiple components, and then *Right Click → Edit Properties*, we can give each part the appropriate package. The three footprints we will be using are “dip16_3” for the 8 to 3 priority line encoder, “dip14_3” for the voltage comparators, and “smdres” for the 1-kΩ resistors. They represent 16-pin dual in-line packages, 14-pin in-line packages, and surface-mount resistive packages, respectively. These footprint IDs should be entered in the “Cadence Allegro” property which can be selected in the drop-down window towards the top of the “Property Editor” window (Figure 13). Other standard Allegro footprints IDs can be found later when using Allegro PCB Editor and placing items manually. This will be shown later in step 6.



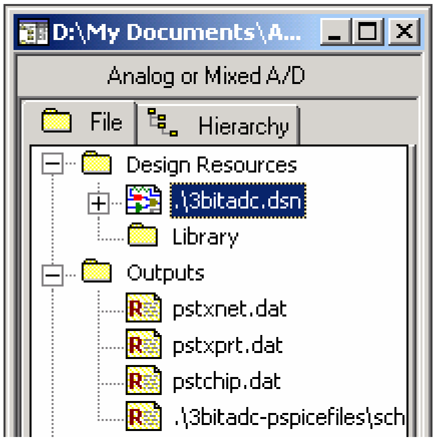
Property Editor

New Column... | Apply | Display... | Delete Property | Filter by: Cadence-Allegro

		HEIGHT	Implementation	Im	Imp	PCB Footprint	PIN_ESCAPE	PINUSE	PLACE_TAG
1	SCHEMATIC1 : PAGE1 : U4		74LS148	PS	...	dip16_3			
2	SCHEMATIC1 : PAGE1 : U2		TLC3704/5_1/TI	PS	...	dip14_3			
3	SCHEMATIC1 : PAGE1 : U2		TLC3704/5_1/TI	PS	...	dip14_3			
4	SCHEMATIC1 : PAGE1 : U2		TLC3704/5_1/TI	PS	...	dip14_3			
5	SCHEMATIC1 : PAGE1 : U2		TLC3704/5_1/TI	PS	...	dip14_3			
6	SCHEMATIC1 : PAGE1 : U1		TLC3704/5_1/TI	PS	...	dip14_3			
7	SCHEMATIC1 : PAGE1 : U1		TLC3704/5_1/TI	PS	...	dip14_3			
8	SCHEMATIC1 : PAGE1 : U1		TLC3704/5_1/TI	PS	...	dip14_3			
9	SCHEMATIC1 : PAGE1 : U1		TLC3704/5_1/TI	PS	...	dip14_3			
10	SCHEMATIC1 : PAGE1 : R8			<no	...	smdres			
11	SCHEMATIC1 : PAGE1 : R7			<no	...	smdres			
12	SCHEMATIC1 : PAGE1 : R6			<no	...	smdres			
13	SCHEMATIC1 : PAGE1 : R5			<no	...	smdres			
14	SCHEMATIC1 : PAGE1 : R4			<no	...	smdres			
15	SCHEMATIC1 : PAGE1 : R3			<no	...	smdres			
16	SCHEMATIC1 : PAGE1 : R2			<no	...	smdres			
17	SCHEMATIC1 : PAGE1 : R1			<no	...	smdres			

Figure 13 - Entering Footprint Definitions

2. Now we are ready to netlist our schematic into Allegro for routing. In order to netlist, we need to select our schematic to netlist. Highlight 3bitadc.dsn icon in the project window (which deghosts many options) and go to *Tools* → *Create Netlist* → then select the *Allegro* tab. Select the options as shown (Figures 14 and 15).

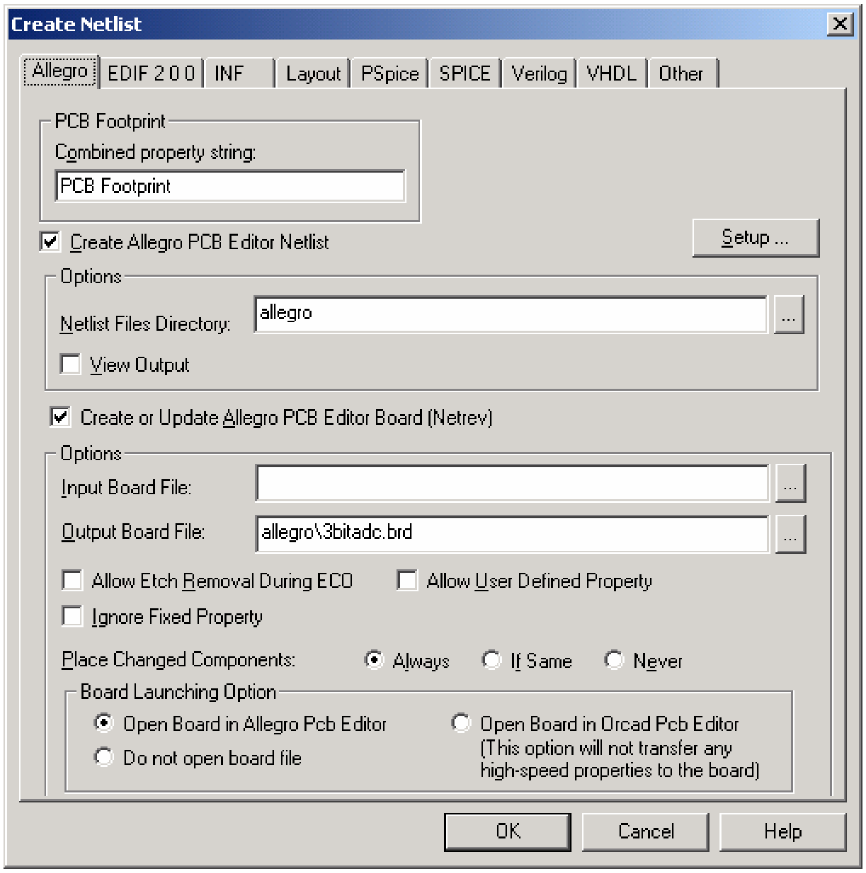


D:\My Documents\A...

Analog or Mixed A/D

Design Resources
.\3bitadc.dsn
Library
Outputs
pstxnet.dat
pstxprt.dat
pstchip.dat
.\3bitadc-pspicefiles\sch

Figure 14 - Highlight "3bitadc.dsn" for Netlisting



PCB Footprint

Combined property string: PCB Footprint

☑ Create Allegro PCB Editor Netlist

Options

Netlist Files Directory: allegro

☐ View Output

☑ Create or Update Allegro PCB Editor Board (Netrev)

Options

Input Board File:

Output Board File: allegro\3bitadc.brd

☐ Allow Etch Removal During ECO ☐ Allow User Defined Property

☐ Ignore Fixed Property

Place Changed Components: ● Always ○ If Same ○ Never

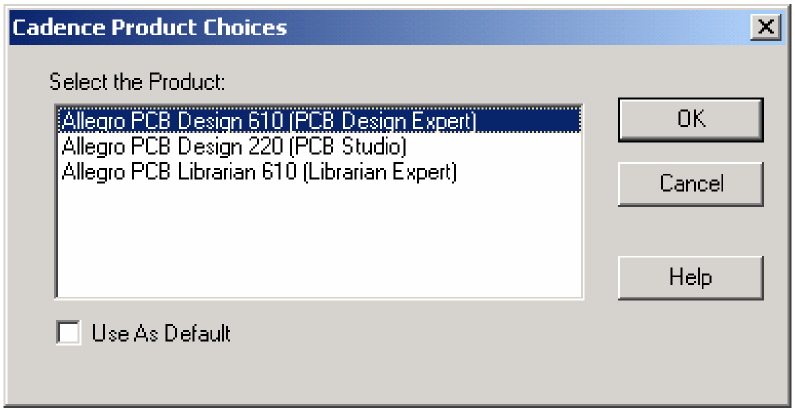
Board Launching Option

● Open Board in Allegro Pcb Editor ○ Open Board in Orcad Pcb Editor (This option will not transfer any high-speed properties to the board)

○ Do not open board file

Figure 15 - Create Allegro Netlist

3. After pressing the OK button, Design Entry CIS should begin netlisting and show its progress in a pop-up window. When finished, it should automatically open Allegro PCB Editor 15.5, and a dialog box as seen in Figure 16 will be shown. Select the "Allegro PCB Design 610 (PCB Design Expert)" product.



Select the Product:

Allegro PCB Design 610 (PCB Design Expert)
Allegro PCB Design 220 (PCB Studio)
Allegro PCB Librarian 610 (Librarian Expert)

☐ Use As Default

Figure 16 - Allegro PCB Editor Product Choices

4.3 Routing the PCB

4. When the program finishes loading, you should see a new graphical user interface with a large empty black area. Note that this program can be loaded directly to create a new board file or open an existing board (.brd) file. Go to *Start* → *All Programs* → *Allegro SPB 15.5* → *PCB Editor*. In our case Allegro PCB Editor was automatically opened after netlisting in Design Entry CIS. Throughout the rest of this chapter, we are essentially building a printed circuit board from scratch, with a reference schematic design we created in Design Entry CIS in Chapter 3.

Before any layout, there are a number of items to set up. This may include, but is not limited to defining the geometry, the board materials, and the electrical constraints of our printed circuit board routing. Hence, this "background setup" portion of PCB design can be very involved. The following only highlights some of the common tools where these prelayout functions can be done. Most of them can be found under the *Setup* menu in PCB Editor. Some of them can also be found among the tool icons (Figure 17).

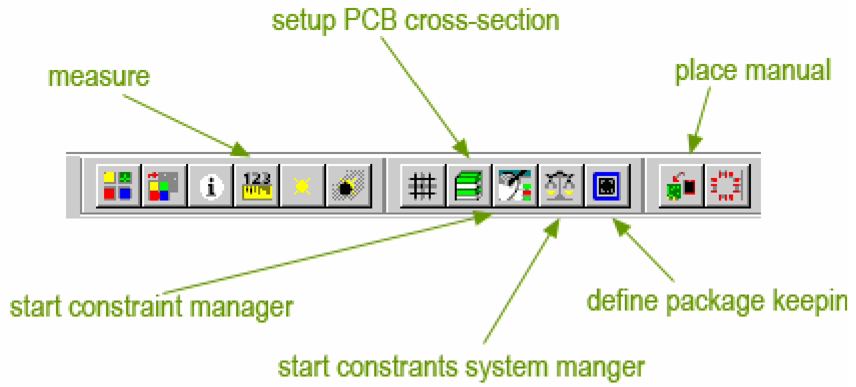
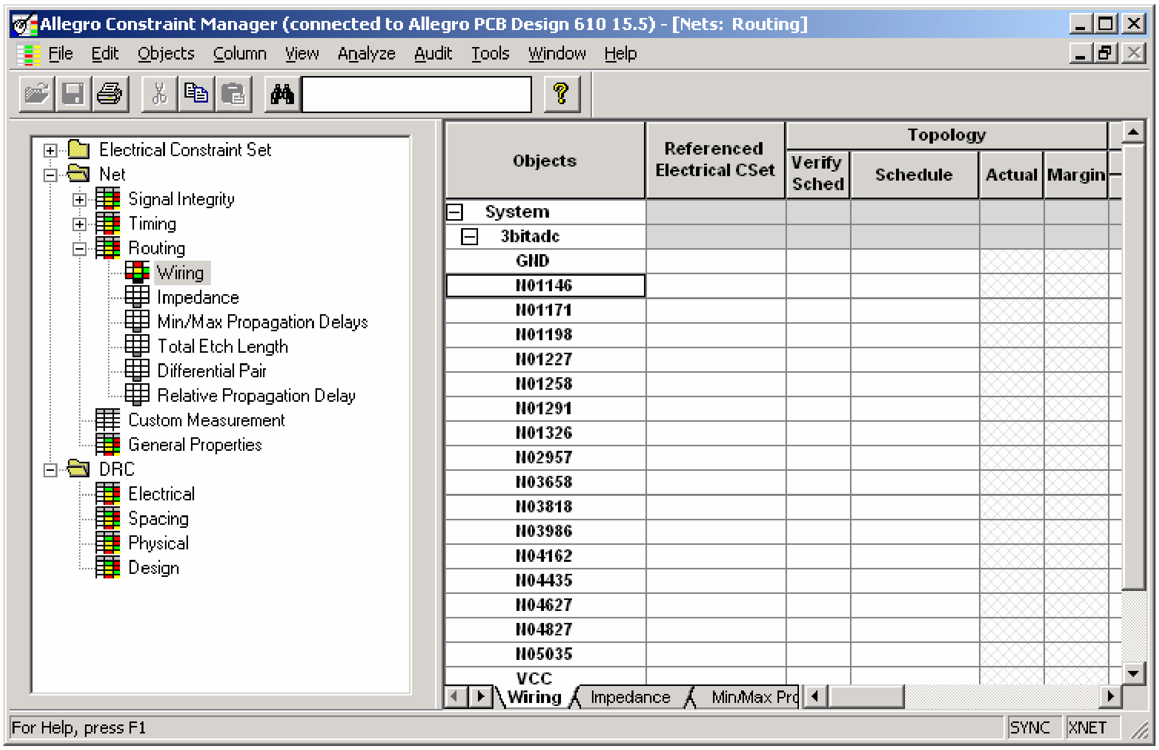


Figure 17 - Allegro PCB Editor Setup Toolbar

a. *Setup* → *Electrical Constraints Spreadsheet* basically opens up an Allegro backend software called Constraint Manager. This software allows the user to enter signal integrity, timing, and routing constraints on all the nets in a board. In addition, the user can schedule nets. When right-clicking on a net, it is possible to select that net on the PCB editor display, or analyze if it meets the constraints. Furthermore, buses and differential pairs can be defined in Constraint Manager (Figures 18 and 19). To learn more about this backend software, there are five demonstrations in an Allegro Constraint Manager Tutorial (Back-End) Product Version 15.5 which is included in the software suite [4].


Allegro Constraint Manager (connected to Allegro PCB Design 610 15.5) - [Nets: Routing]
File Edit Objects Column View Analyze Audit Tools Window Help
Electrical Constraint Set
Net
Signal Integrity
Timing
Routing
Wiring
Impedance
Min/Max Propagation Delays
Total Etch Length
Differential Pair
Relative Propagation Delay
Custom Measurement
General Properties
DRC
Electrical
Spacing
Physical
Design
Wiring
Impedance
Min/Max Prc
For Help, press F1
SYNC
XNET

Objects	Referenced Electrical CSet	Topology			
		Verify Sched	Schedule	Actual	Margin
⊟ System					
⊟ 3bitadc					
GND					
N01146					
N01171					
N01198					
N01227					
N01258					
N01291					
N01326					
N02957					
N03658					
N03818					
N03986					
N04162					
N04435					
N04627					
N04827					
N05035					
VCC					

Figure 18 - Allegro Constraint Manager Backend

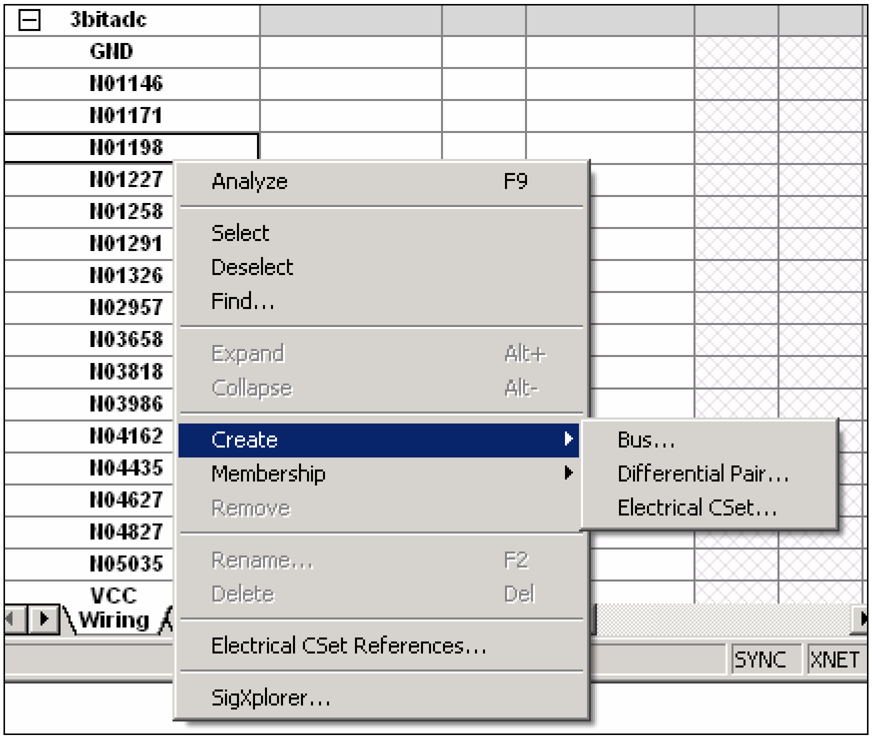
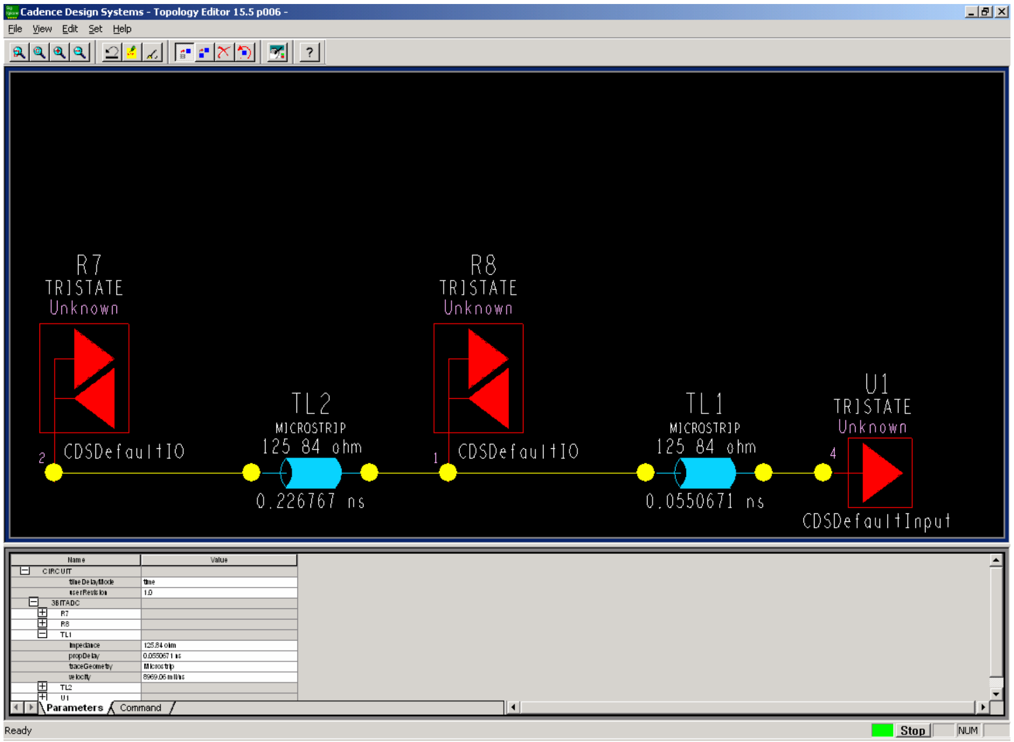


Figure 19 - Wiring Right-Click Menu

b. When using Constraint Manger, often it is useful to extract a net or bus topology to SigXplorer, rearrange connections, and re-update Constrant Manager with the new information. Although this is covered in detail in [4], Figure 20 displays a typical net extraction in SigXplorer. Each element within a net is displayed in a block diagram with appropriate timing and impedances displayed. SigXplorer will be used more later, and is useful for signal integrity analysis too—where we can extract the topology of routed traces rather than prerouted traces at this current design step. In any case, SigXplorer operates as a microwave topology editor or simulation tool.

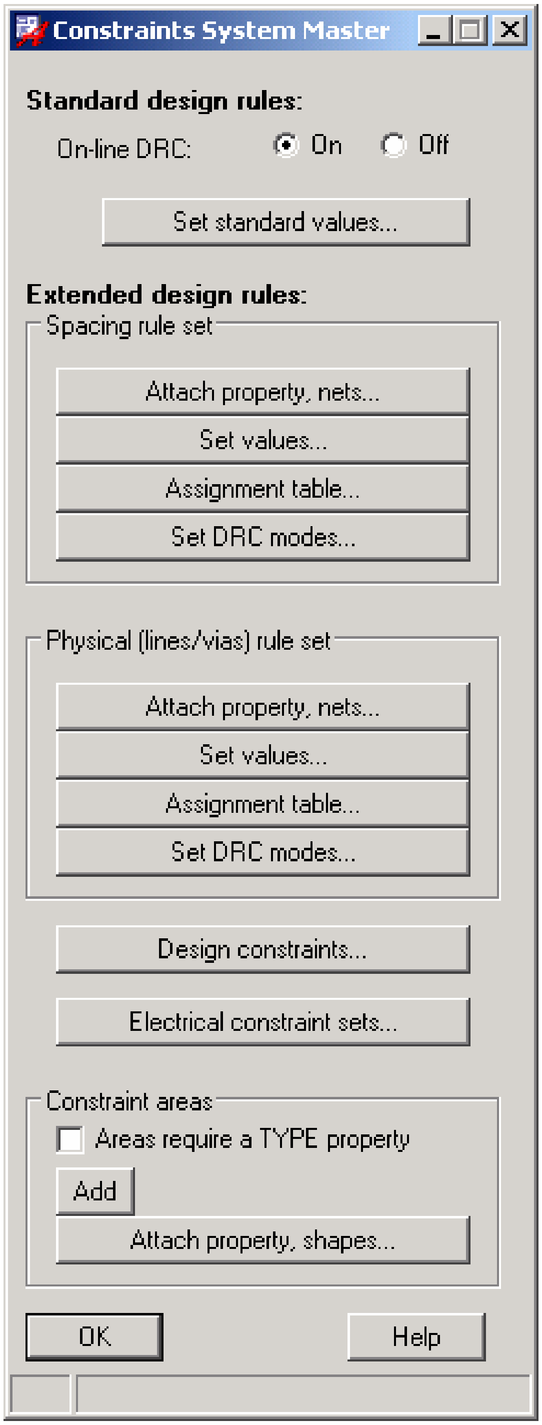


Name	Value
CIRCUIT	
timeDelayMode	time
userRevision	1.0
3BITADC	
R7	
R8	
TL1	
impedance	125.84 ohm
propDelay	0.0550671 ns
traceGeometry	Microstrip
velocity	8969.06 mil/ns
TL2	
U1	

Figure 20 - Extracted Topology in SigXplorer

c. *Setup* → *Constraints* loads up straightforward board-level user-defined constraints for design rule checking (DRC) (Figure 21). Since the options in this tool are fairly straightforward, it will not be covered in this tutorial. It is worth exploring on your own.

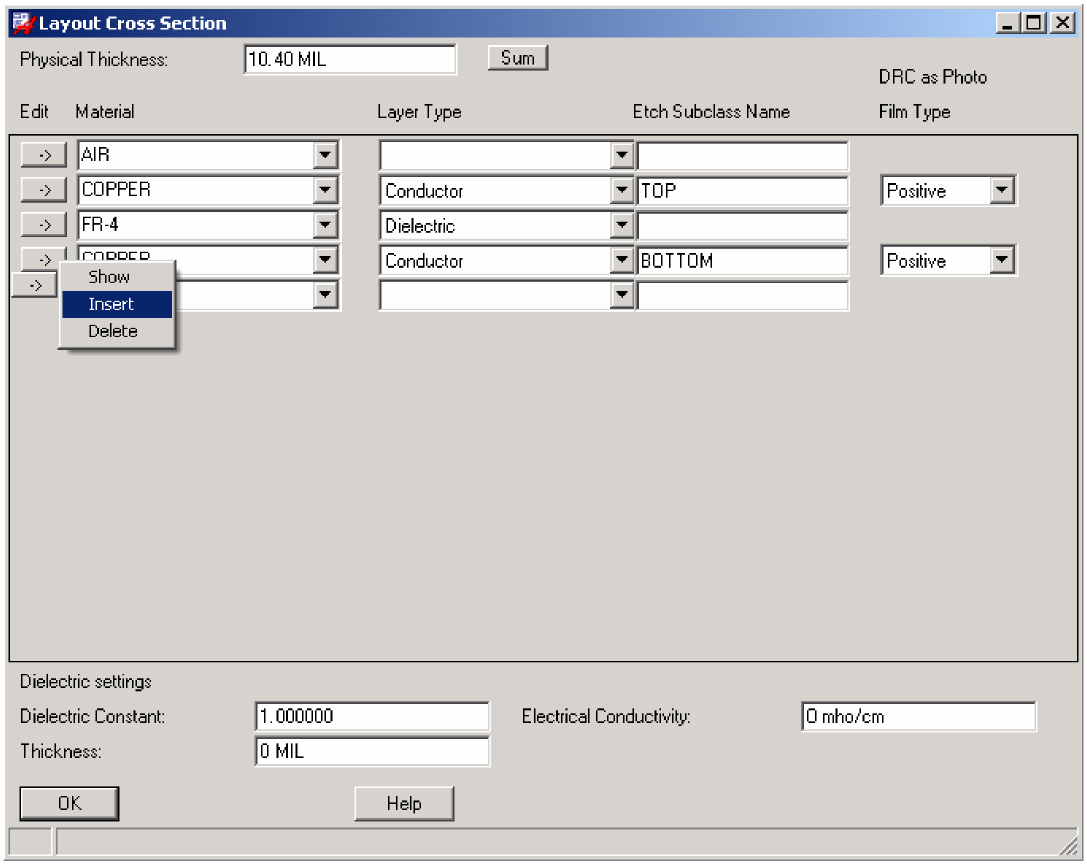
d. *Setup* → *Cross-section* enables the user to define conductor and dielectric materials used in the PCB. This is also known as adjusting the "stack-up" or "layer stack." In addition, this will affect the padstack. To add layers simply click on the appropriate arrow button (Figure 22). In our ADC board routing, two conductor layers in FR-4 dielectric should suffice.



On-line DRC: ● On ○ Off

□ Areas require a TYPE property

Figure 21 - Constraints Systems Manager



Layout Cross Section

Physical Thickness: 10.40 MIL

Edit	Material	Layer Type	Etch Subclass Name	DRC as Photo Film Type
->	AIR			
->	COPPER	Conductor	TOP	Positive
->	FR-4	Dielectric		
->	COPPER	Conductor	BOTTOM	Positive
->				

Dielectric settings

Dielectric Constant: 1.000000

Thickness: 0 MIL

Electrical Conductivity: 0 mho/cm

Figure 22 - Inserting a New Layer in the PCB Cross Section

e. *Setup* → *Areas* define some of the essential keepin and keepout areas for routing, such as *package* keepin/keepout, and *route* keepin/keepout (Figure 23). For example, when selecting the *package keepin* tool, the mouse cursor becomes a crosshatch, and a purple polygon is able to be defined with each mouse click. Any packages that do not fall within the keepin area are marked with a red square (Figure 24). Similarly, if there are any routed nets that do not fall within the *route* keepin area, they will be marked with a red square. The built-in autorouter will also adhere to these keepin/keepout restrictions.

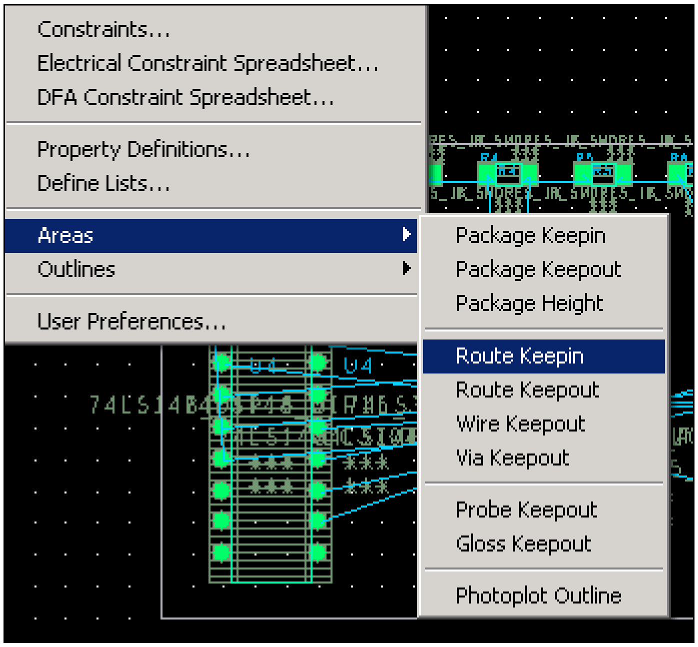


Figure 23 - Defining Keepin and Keepout Areas

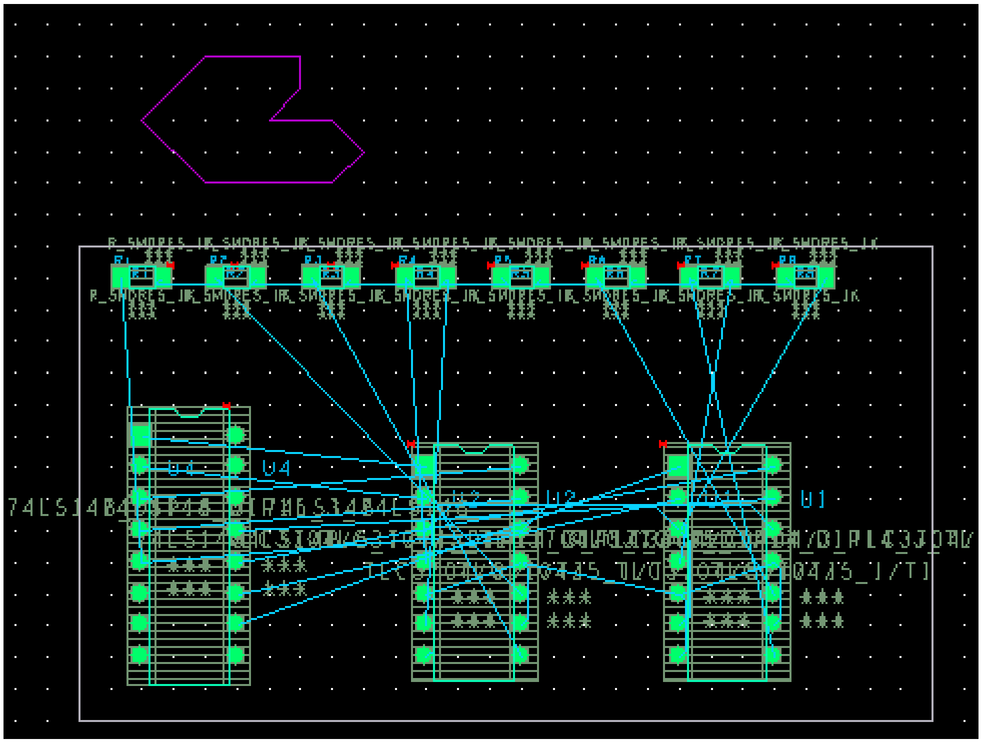
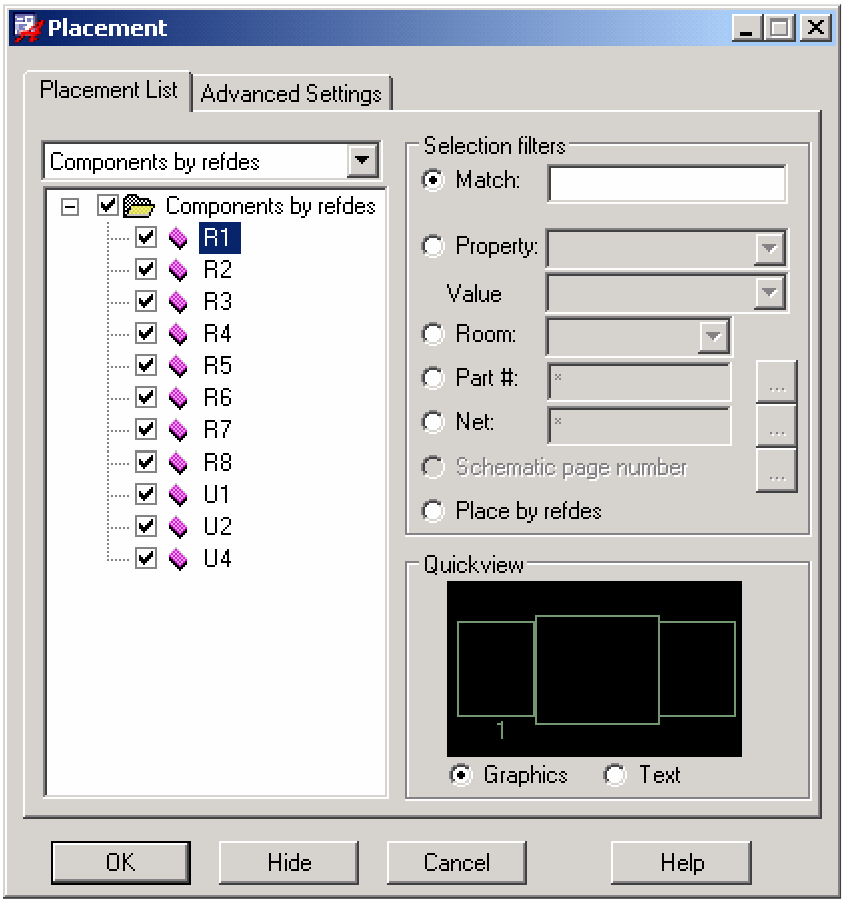


Figure 24 - Keepout Area with Packages in Violation

Those are the essential prelayout setups, although there are other tools to tweak other things including Design for Assembly (DFA) tables and user preferences.

5. After the necessary preroute setups and constraint definitions, we can start placing components on our PCB blank slate. Select *Place* → *Manually*. A “Placement” window should pop-up (Figure 25). Check the box next to the folder icon to select all the components. Now, with each click on the black drawing area, we will be placing each of these components in succession; from R1 down to U4. Randomly position all of these footprints on the drawing area, and then hit the OK button when done (Figure 26). The green shapes are the footprints, and the blue lines correspond to what is called a “ratsnest.” They roughly correspond to which pins are electrically connected to another pin.



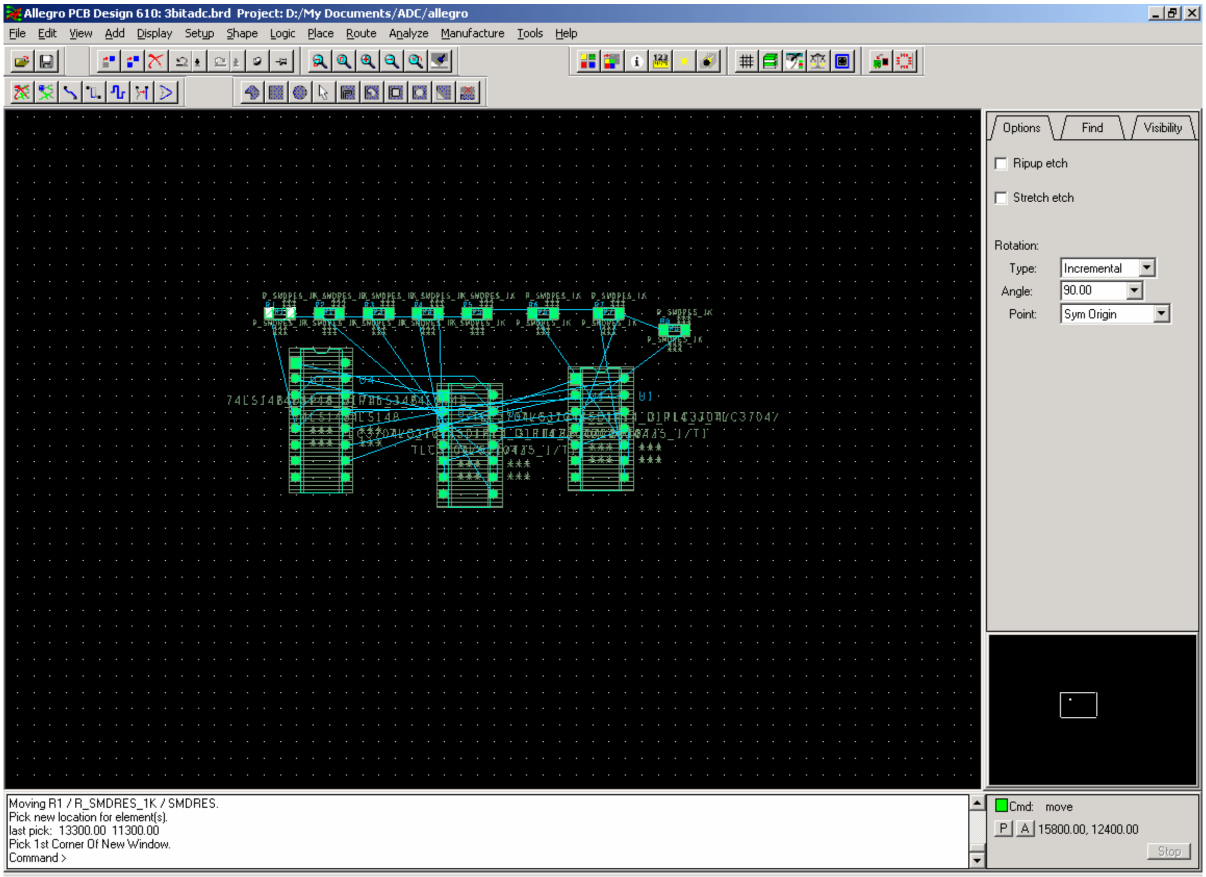
Components by refdes

☑ Components by refdes
- ☑ R1
- ☑ R2
- ☑ R3
- ☑ R4
- ☑ R5
- ☑ R6
- ☑ R7
- ☑ R8
- ☑ U1
- ☑ U2
- ☑ U4

Selection filters
- ● Match:
- ○ Property:
- Value
- ○ Room:
- ○ Part #: *
- ○ Net: *
- ○ Schematic page number
- ○ Place by refdes

Quickview
- ● Graphics ○ Text

Figure 25 - Manual Placement of Reference Design Footprints


Allegro PCB Design 610: 3bitadc.brd Project: D:/My Documents/ADC/allegro
File Edit View Add Display Setup Shape Logic Place Route Analyze Manufacture Tools Help
Options
Find
Visibility
Stop

□ Ripup etch

□ Stretch etch

Rotation:

Type: Incremental

Angle: 90.00

Point: Sym Origin

```
Moving R1 / R_SMDRES_1K / SMDRES.
Pick new location for element(s).
last pick:  13300.00  11300.00
Pick 1st Corner Of New Window.
Command >
```

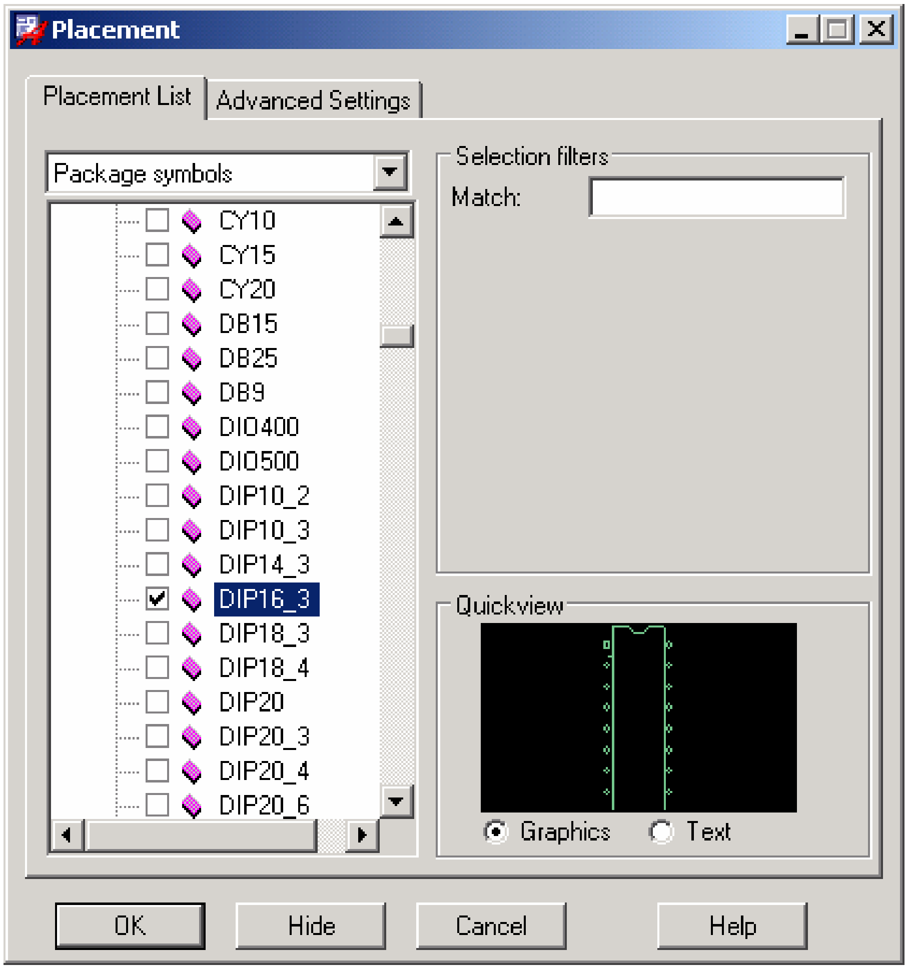
Cmd: move

P A 15800.00, 12400.00

Figure 26 - Arbitrary Manual Footprint Placement

6. Side Note: as mentioned towards the beginning of this chapter, other footprints can be found by clicking *Advanced Settings* tab → checking the *Library* box in the Placement pop-up. Return to the *Placement List* tab, and then select "Package symbols" from the drop-down menu (Figure 27). It should display over a hundred other standard PCB footprints that can be used in conjunction with property definitions in Design Entry CIS or Design Entry HDL.

7. At this time, we should define the physical board geometry, and the keepin areas, if we have not done so during setup. These are usually defined by specific applications and would be given as requirements to a PCB designer before any routing. For example, a PCI Express card for a personal computer would have standard pin locations and geometry for its slot.



Package symbols

- □ CY10
- □ CY15
- □ CY20
- □ DB15
- □ DB25
- □ DB9
- □ DIO400
- □ DIO500
- □ DIP10_2
- □ DIP10_3
- □ DIP14_3
- ☑ DIP16_3
- □ DIP18_3
- □ DIP18_4
- □ DIP20
- □ DIP20_3
- □ DIP20_4
- □ DIP20_6

Match:

● Graphics ○ Text

Figure 27 - Manual Placement of Library Footprints

To keep things simpler, for this design, I assumed and defined rectangular areas for the physical board and keepin areas. Typically, keepin areas are concentric with the board’s physical profile, and several mils inward. In Figure 28, I already defined the board geometry (outlined in gray), and also the package keepin area (outlined in purple). In the figure, I am in the middle of defining the route keepin area (outlined in gold) on top of the package keepin area. Notice how the footprints outside of the package keepin area are marked with red squares.

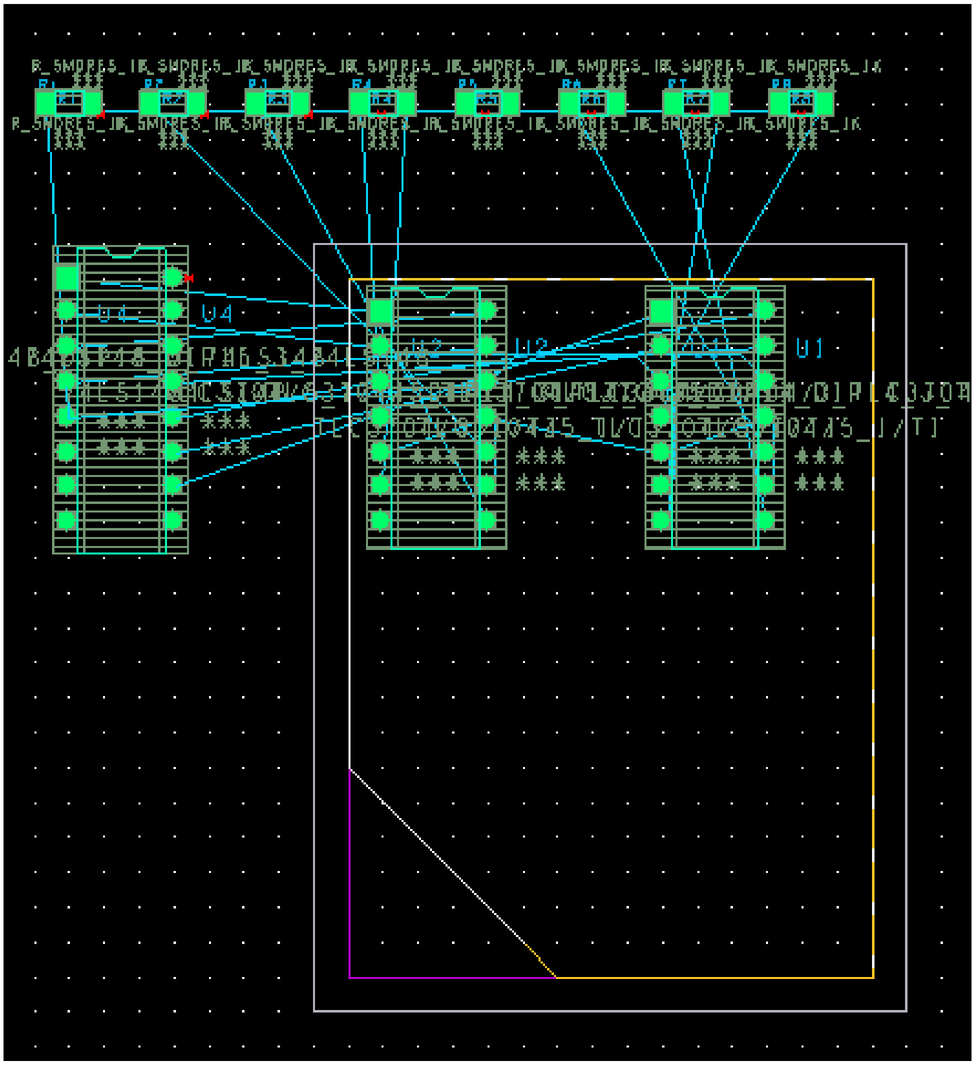


Figure 28 - Defining Route Keepin Area

8. Once we have defined the geometries of our board, we can arrange the packages as we like. To do this we should first familiarize ourselves with the common tools (Figure 29 and previous Figure 17). In brief, to rearrange packages, we use the *Move* tool; to route, we use the *Add Connect* tool; and to adjust routed nets, we use the *Slide* tool. To get a more detailed overview on the Allegro PCB Editor design interface, refer the tutorial included with the software suite [5].

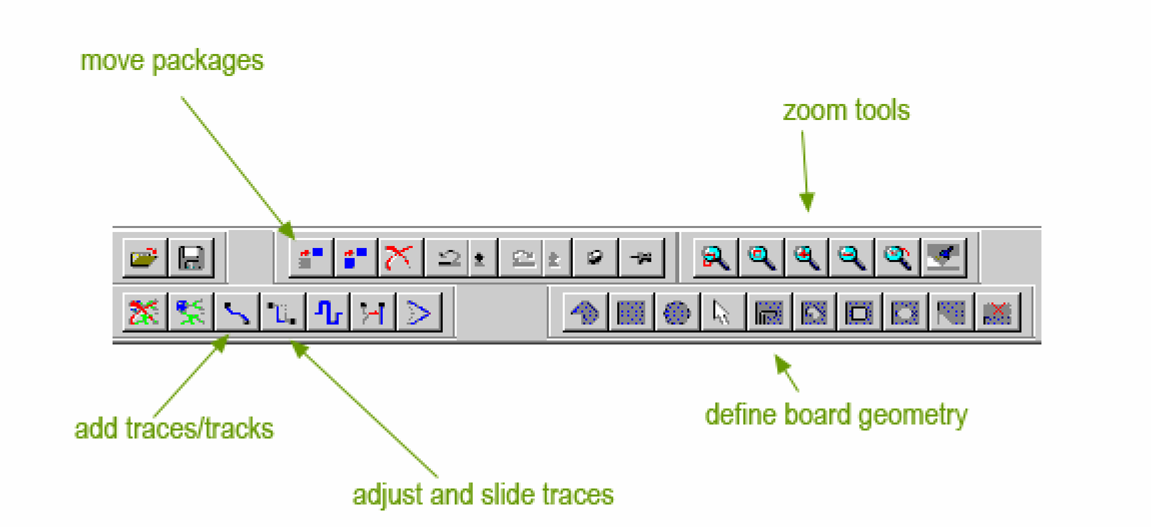


Figure 29 - Allegro PCB Editor Toolbar

9. As we move on to routing the board, take note that PCB routing or “tracking” is a skill in itself, requiring years of experience to achieve the best practices. Standard guidelines in PCB design and its associated issues can be found in the IPC-2221A standard [6]. Higher-level, but detailed overviews and rules of thumb in PCB design can be found in [7] and [8]. Some PCB design rules of thumb that affect SI issues can be found in [9] on pages 94, 97, and 128.

   Some common tips include [7]:

   a. Try to keep tracks as short as possible to avoid timing delays. Shorter nets reduce inductance, capacitance, and resistance.

   b. Tracks should have angles of 45 °. Avoid right angles. Never use turns more than 90º.

   c. Nets that connect to pads should always connect to the center of the pad, and not just touch the pad.

   d. Do not place vias under components.

10. In general, there are three methods to route a board. Most experienced PCB designers prefer to route the entire board full-custom [7]. However, autorouting is always possible, though not recommend, and so is a mixture of the two. In general, no matter how "smart" an autorouter is, it cannot replace a good PCB designer. On the other hand, experienced PCB designers will be able to leverage the added speed auto-routing affords and still produce the best results.

11. Should one desire to autoroute, the general procedure is to go to *Route* → *Automatic* or *Route* → *Custom*. Either will open up software titled SPECCTRA autorouter and will make multiple passes as it tries to route the PCB.

12. In any case, route the PCB. In this ADC example, I have routed the board as shown in Figure 30. Save your design in a .brd file and close Allegro PCB Editor. We will be using Allegro PCB SI for signal integrity analysis in the next chapter.

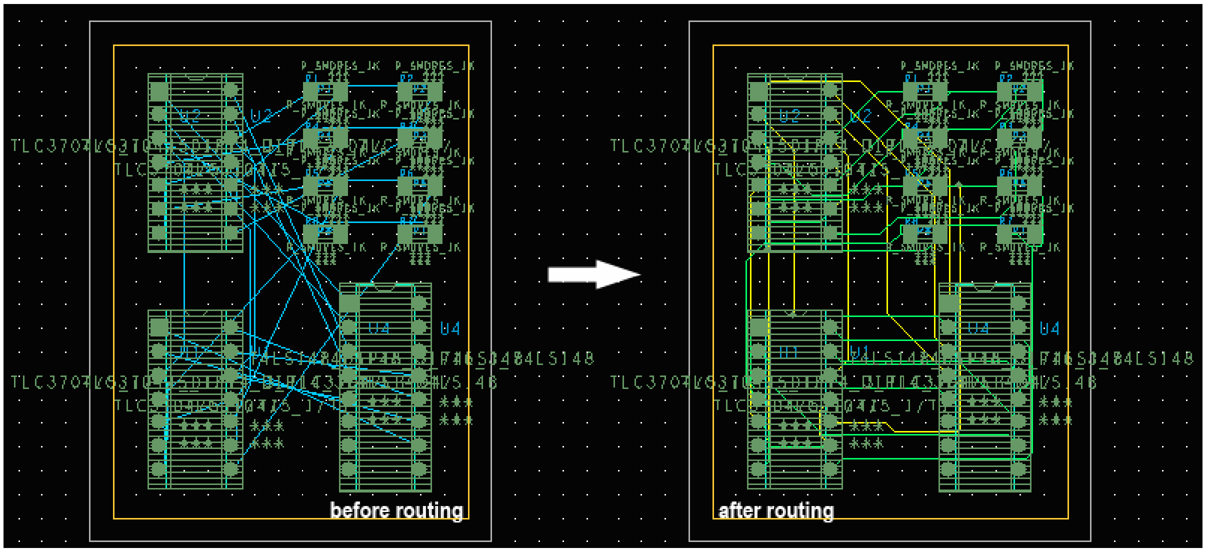


Figure 30 - Before and After Routing

4.4 Going Further

In this case, the PCB was a two-layer board. It could be a valuable exercise to route a board with more layers, including a ground plane, and check its effect on signal integrity. In addition, all the packages were placed on the top of the PCB. One can go further to compact the design by making a double-sided design with packages on both the top and bottom layers.

5 SIGNAL INTEGRITY ANALYSIS USING ALLEGRO PCB SI

5.1 Overview

In this final section, we analyze the signal integrity of our routed PCB. Typically, PCB SI simulation is not just a holistic chronological step done after routing is completed. Rather, SI simulation is often done during routing, especially when placing critical nets. While this thesis breaks down the processes as discrete steps for simplicity, the actual workflow is more like an involved interleaved solution, jumping between the two.

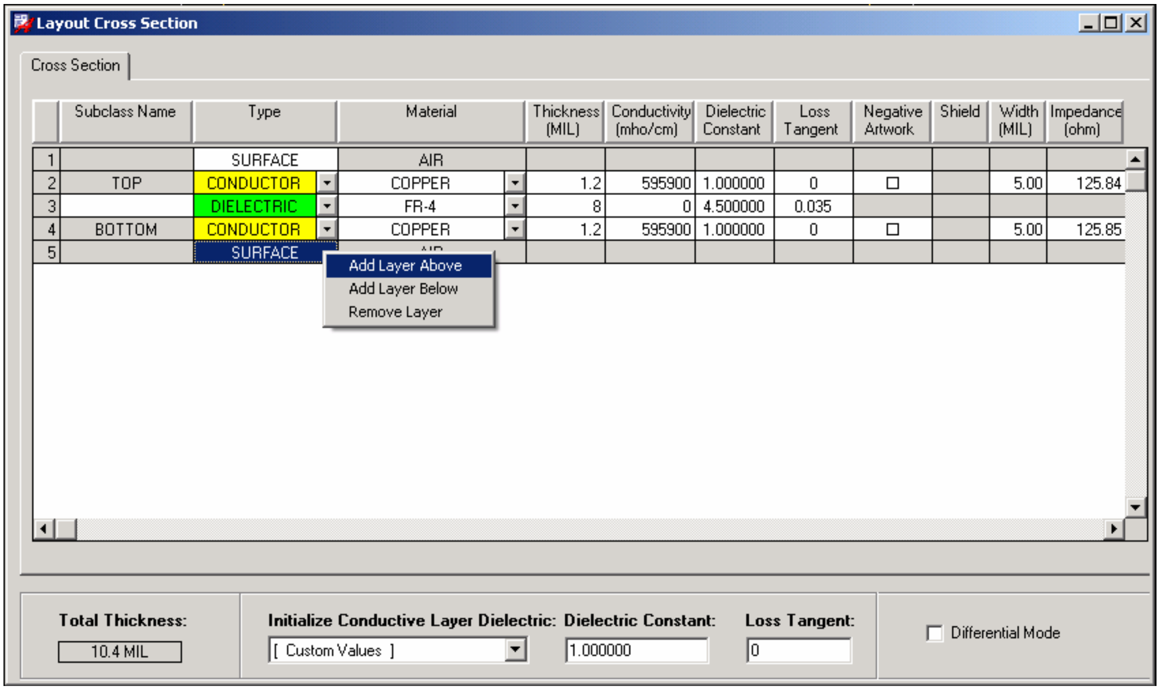
In fact, the included PCB SI user's guide specifies both a preroute analysis flow and a critical net (postroute) analysis flow [10].

This chapter as a whole will cover eye diagram simulation and other SI/EMI simulations available in Allegro PCB SI as described by the user guide's workflows.

5.2 SI Simulation Setup

1. Signal Integrity analysis uses Allegro PCB SI. This can be accessed by clicking *Start* → *All Programs* → *Allegro SPB 15.5* → *PCB SI.* The graphical user interface should be very similar to Allegro PCB Editor, except there are more tool icons and menus.
2. To get started with simulations we select *Tools* → *Setup Advisor.* This is a walk-through program that sets up the databases and models needed to simulate signal integrity of the devices on the board. The first page is an introduction. Click on the *Next* > button. This step allows us to edit the cross-section of our board and tune some of the parameters. Click on the Edit Cross-section button in the middle of the window. A new window should appear (Figure 31). This is very similar to the

preroute setup step seen in Chapter 4 item d. Therefore, there are not really any adjustments to make here. Close out the window and click on the *Next >* button.



Layout Cross Section

Cross Section

	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (MIL)	Impedance (ohm)
1		SURFACE	AIR								
2	TOP	CONDUCTOR	COPPER	1.2	595900	1.000000	0	□		5.00	125.84
3		DIELECTRIC	FR-4	8	0	4.500000	0.035				
4	BOTTOM	CONDUCTOR	COPPER	1.2	595900	1.000000	0	□		5.00	125.85
5		SURFACE	AIR								

Total Thickness: 10.4 MIL

Initialize Conductive Layer Dielectric: [Custom Values]

Dielectric Constant: 1.000000

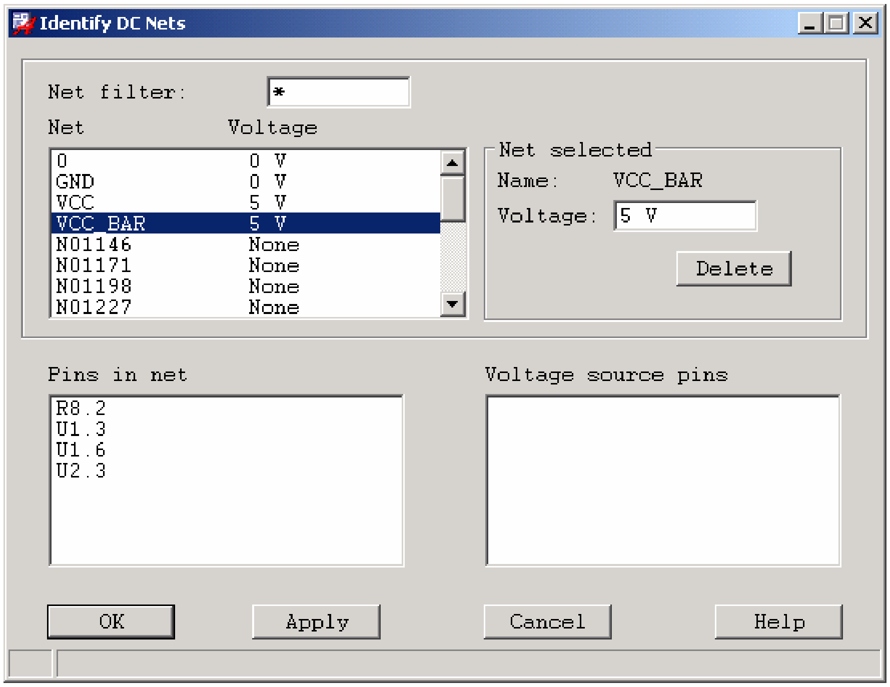
Loss Tangent: 0

☐ Differential Mode

Figure 31 - Edit Cross-Section Window

3. The next step allows us to identify the DC nets in our design. Click on the *Identify DC Nets* button. Correctly set the voltages for our ground and V$_{CC}$ pins by setting them to 0 V and 5 V, respectively. Then click *OK*, and *Next >* (Figure 32).

4. Clicking onto the *Device Setup* button allows us to specify specific resistances, capacitances, and inductances for component parts. Since in this case we only have eight surface mounted resistors, we only have to edit the Resistor field. We will change the default 1 Ω to 1000 Ω as specified in the original design (Figure 33). To set individual components to different values one can select the *Browse* button of the corresponding part. Hit *OK* and *Next >* to continue.



Net filter: *

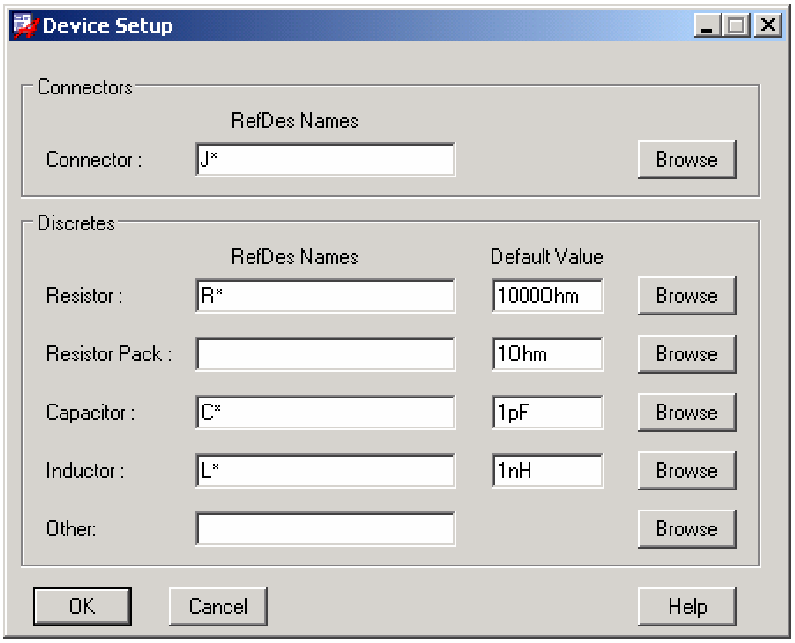
Net	Voltage
0	0 V
GND	0 V
VCC	5 V
VCC_BAR	5 V
N01146	None
N01171	None
N01198	None
N01227	None

Net selected
Name: VCC_BAR
Voltage: 5 V

Pins in net
R8.2
U1.3
U1.6
U2.3

Voltage source pins

Figure 32 - Identifying DC Nets



Connectors

	RefDes Names
Connector :	J*

Discretes

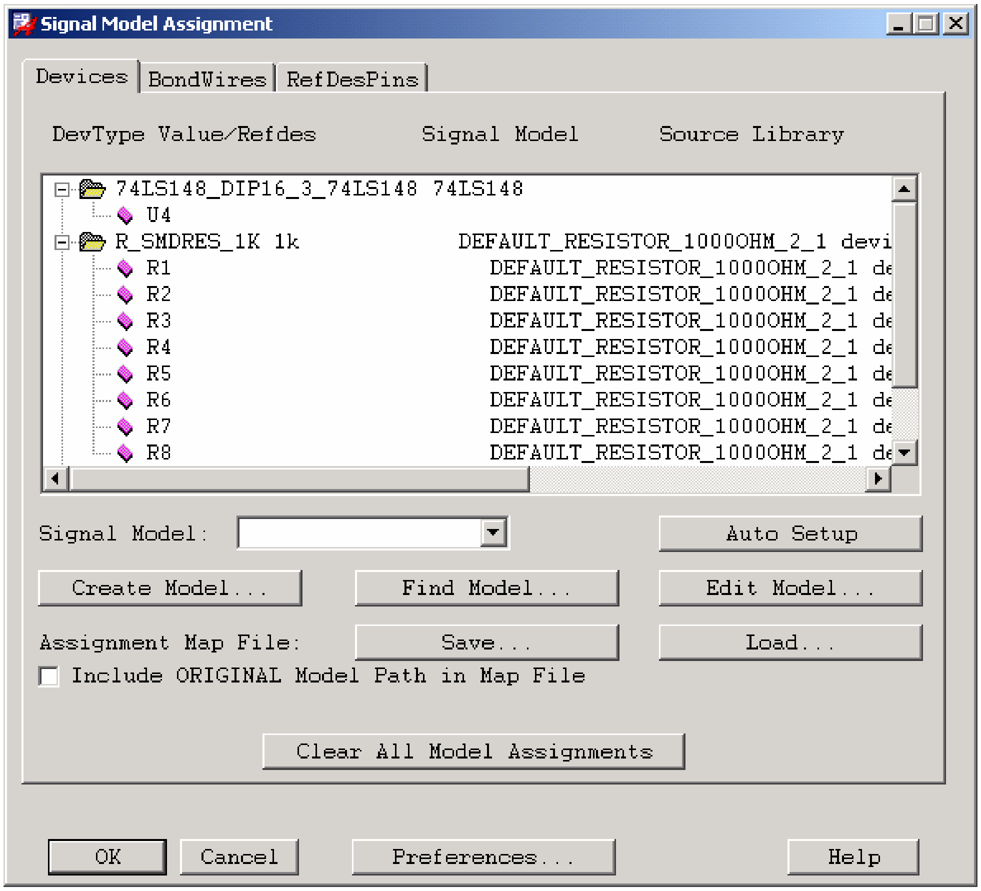
	RefDes Names	Default Value
Resistor :	R*	1000Ohm
Resistor Pack :		1Ohm
Capacitor :	C*	1pF
Inductor :	L*	1nH
Other:		

Figure 33 - Device Setup

5. Finally click on *SI Model Assignment* to complete the last setup procedure. This is where all the buffer and parasitic models for each device can be set up. In general, the vender of each of the chips would provide a model for the SI engineer. Models

can also be generated from measurement. Typically these models are either SPICE, IBIS, VHDL-AMS, or Verilog-A models, the most common being SPICE-based and IBIS models. A good comparison between these models can be found in [11]. Nevertheless, IBIS models I/O pins using V/I and V/T measure data and simulates faster than SPICE-based models in a signal integrity context, so we will use IBIS model.

Clicking on *Auto Setup* places models for all the resistors, as seen in Figure 34, but we will still have to create models for the three packages that contain the comparators and encoder.



Signal Model Assignment

```
DevType Value/Refdes          Signal Model        Source Library

74LS148_DIP16_3_74LS148 74LS148
    U4
R_SMDRES_1K 1k              DEFAULT_RESISTOR_1000OHM_2_1 devi
    R1                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R2                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R3                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R4                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R5                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R6                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R7                          DEFAULT_RESISTOR_1000OHM_2_1 de
    R8                          DEFAULT_RESISTOR_1000OHM_2_1 de
```

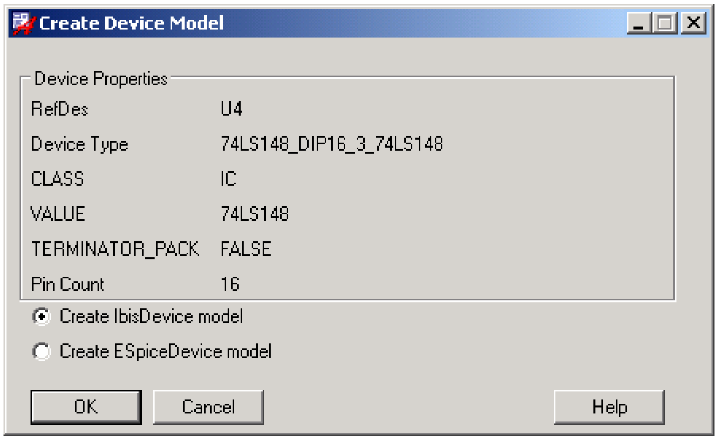
Signal Model:

Assignment Map File:

☐ Include ORIGINAL Model Path in Map File

Figure 34 - Signal Model Assignment Using Auto Setup

6. To create a model, select the 74LS148 device by highlighting the U4 icon. Then select the *Create Model…* button. A pop-up will ask which type of model to create. Select the radio button for "Create IbisDevice model." Then press *OK* (Figure 35).



Device Properties

RefDes	U4
Device Type	74LS148_DIP16_3_74LS148
CLASS	IC
VALUE	74LS148
TERMINATOR_PACK	FALSE
Pin Count	16

● Create IbisDevice model

○ Create ESpiceDevice model

Figure 35 - Create IBIS Device Model

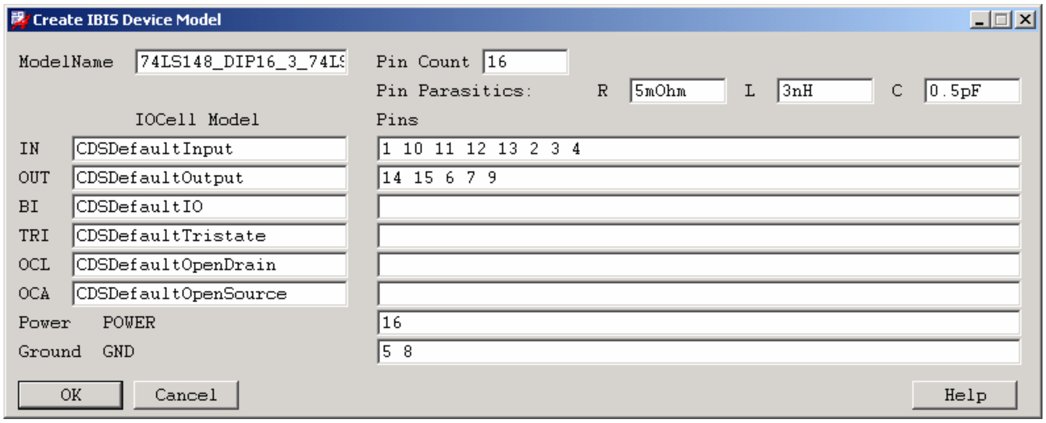
7. For illustration and the purposes of this thesis we will simply adjust the RLC values. Here I have arbitrarily chosen R = 5 mΩ, L = 3 nH, and C = 0.5 pF (Figure 36). Do the same with the U1 and U2 packages to complete all the modeling. Hit *OK* then *Finish*.

In reality, IBIS models or measurements would be supplied to the SI engineer by the vendor, or modeled in-house. More defined models and libraries can be created using *Model Integrity* (Figure 2 on page 7). A table of measurements would just be entered to define these IBIS models. IBIS 4.0 supports up to 1000 measurement points [11].

5.3 SI Simulation: Eye Diagrams

8. One common type of signal integrity simulation is the eye diagram simulation, because of its visual and simple analytical nature. "An eye diagram is created by laying individual periods of [digital] transitions on a net on top of each other [9]." By

analyzing an eye diagram one can determine “a first order approximation of signal-to-noise, clock timing jitter, and skew [12].” In addition, one can determine the best time to sample the signal—in the middle of the eye opening (Figure 37).



ModelName 74LS148_DIP16_3_74LS

Pin Count 16

Pin Parasitics: R 5mOhm L 3nH C 0.5pF

	IOCell Model	Pins
IN	CDSDefaultInput	1 10 11 12 13 2 3 4
OUT	CDSDefaultOutput	14 15 6 7 9
BI	CDSDefaultIO	
TRI	CDSDefaultTristate	
OCL	CDSDefaultOpenDrain	
OCA	CDSDefaultOpenSource	
Power	POWER	16
Ground	GND	5 8

Figure 36 - Create IBIS Devices Model: RLC Values

9. To simulate an eye diagram of a routed track, it is useful to extract the topology of the circuit into SigXplorer. To do this, we select *Analyze* → *SI/EMI Sim* → *Probe*. Click on a routed trace on the PCB layout, then select *View Toplogy* (Figure 38). In the extracted topology of this particular selected trace, SigXplorer displays the driving pin, the microstrip traces displaced by a via, then the load pin (Figure 39).

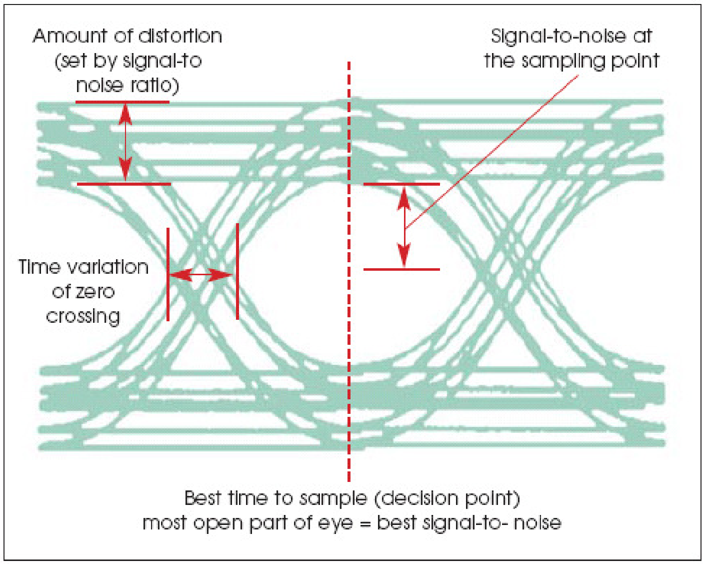
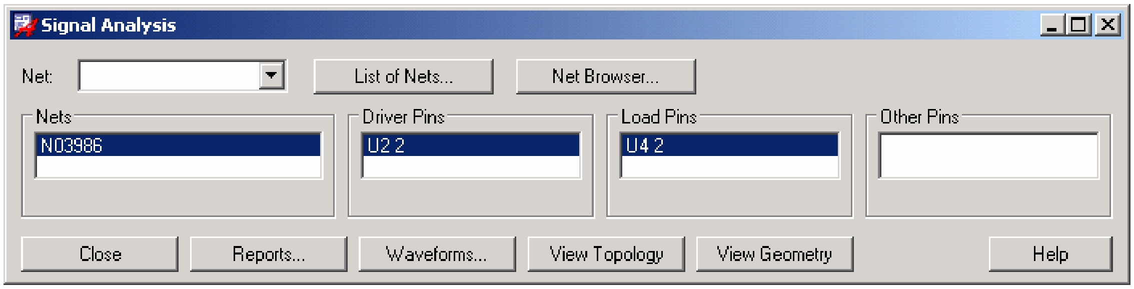


Figure 37 - Eye Diagram Analysis [12]



Signal Analysis

Net:

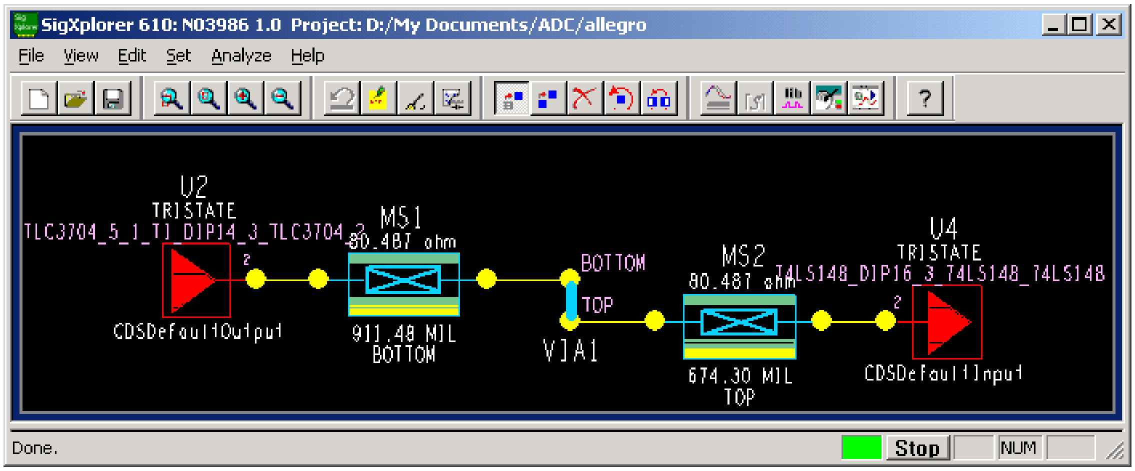
Nets: N03986

Driver Pins: U2 2

Load Pins: U4 2

Other Pins:

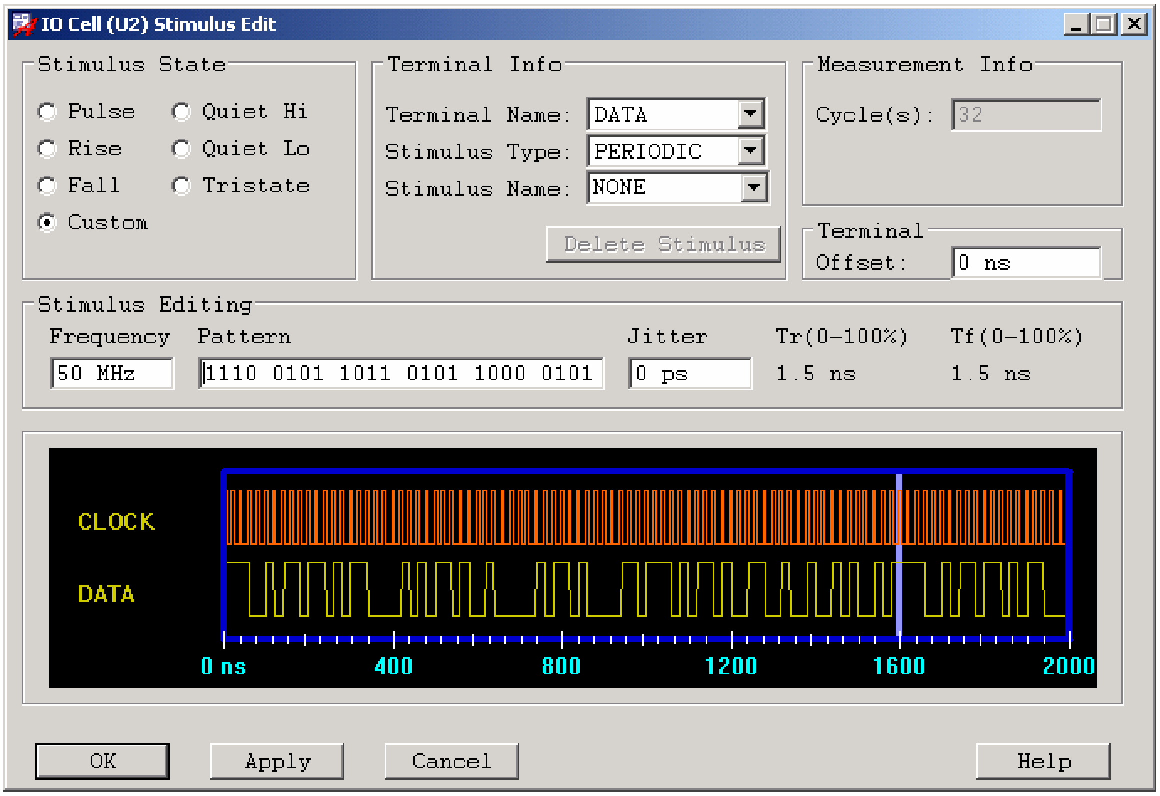
Figure 38 - Signal Analysis on a Routed Trace



SigXplorer 610: N03986 1.0 Project: D:/My Documents/ADC/allegro

Figure 39 - SigXplorer Extracted Trace

10. In order to accurately generate an eye diagram, a pseudorandom bit sequence (PRBS) driving the load would be ideal. To change the stimulus click on the "TRISTATE" label of the driving pin. A pop-up window allows you to change the stimulus. Select the "Custom" radio button. Enter a long stream of data bits randomly and at least longer than 32 bits. We will use a frequency of 50 MHz for this simulation. Then hit *OK* (Figure 40). It should be noted that the rise time (Tr) and fall time (Tf) of our signal cannot be manually adjusted and are automatically calculated by SigXplorer from the IBIS driver models [13]. This already presents a limit on the highest frequency we can simulate.



IO Cell (U2) Stimulus Edit

Stimulus State

○ Pulse ○ Quiet Hi
○ Rise ○ Quiet Lo
○ Fall ○ Tristate
● Custom

Terminal Info

Terminal Name: DATA
Stimulus Type: PERIODIC
Stimulus Name: NONE

Measurement Info

Cycle(s): 32

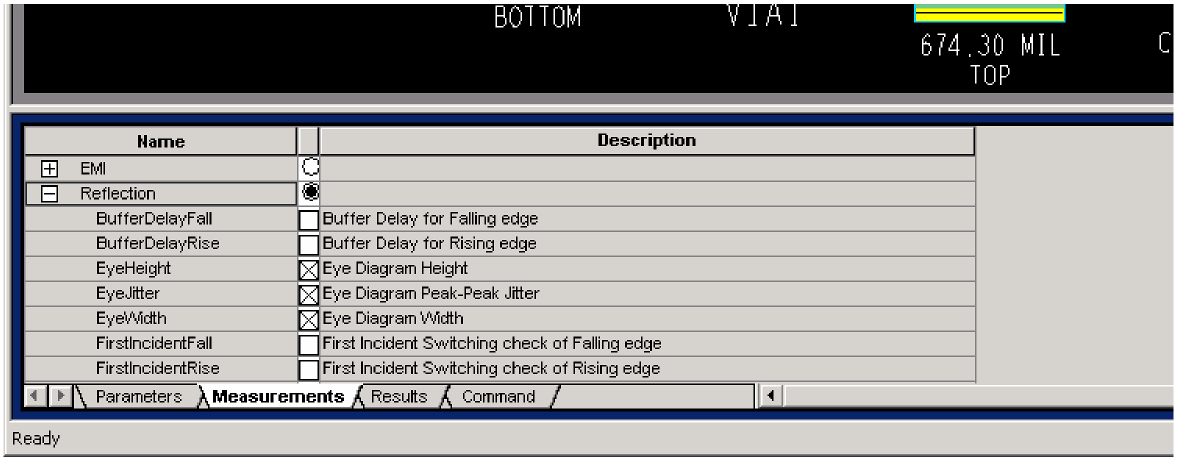
Terminal Offset: 0 ns

Stimulus Editing

Frequency	Pattern	Jitter	Tr(0-100%)	Tf(0-100%)
50 MHz	1110 0101 1011 0101 1000 0101	0 ps	1.5 ns	1.5 ns

Figure 40 - Editing Stimulus

11. After setting up the stimulus, enable eye diagram measurements such as eye height, eye jitter, and eye width. In the *Measurement* bottom tab expand the "Reflection" cross box, and check "EyeHeight," "EyeJitter," and "EyeWidth" (Figure 41). After this is setup click *Analyze* → *Simulate*.

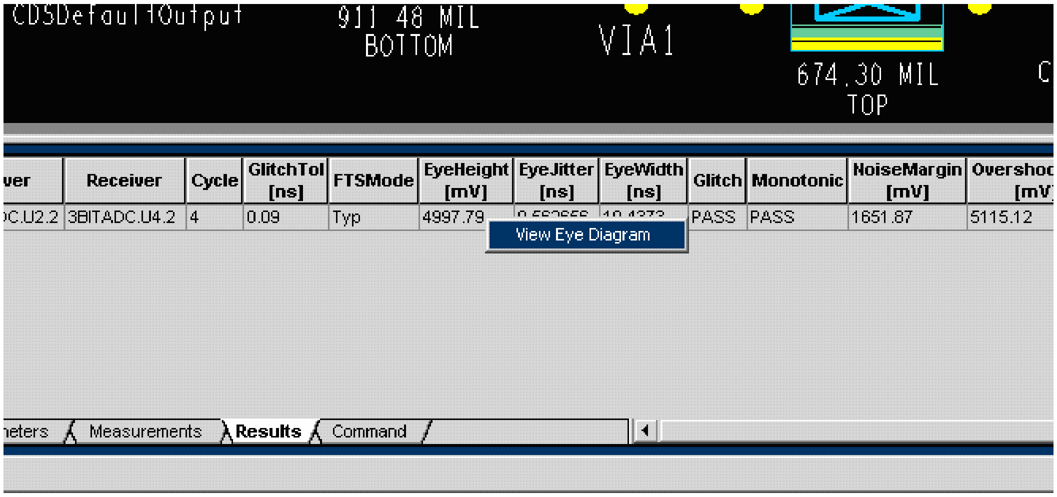


Name		Description
⊞ EMI	○	
⊟ Reflection	●	
BufferDelayFall	□	Buffer Delay for Falling edge
BufferDelayRise	□	Buffer Delay for Rising edge
EyeHeight	☒	Eye Diagram Height
EyeJitter	☒	Eye Diagram Peak-Peak Jitter
EyeWidth	☒	Eye Diagram Width
FirstIncidentFall	□	First Incident Switching check of Falling edge
FirstIncidentRise	□	First Incident Switching check of Rising edge

Figure 41 - Enabling Eye Diagram Measurements

12. When the simulation is complete, it should automatically open up a waveform viewer program called SigWave. For now, send that window to the background. Then select the *Results* bottom tab. It should list all the measurements set up before in the *Measurements* bottom tab. Among them are the three eye diagram measurements. To view the eye diagram easily just *Right Click* → *View Eye Diagram* (Figure 42). Note that the mouse cursor should be on one of the three eye diagram measurements, or else the user will just see *View Waveform*. Upon clicking View Eye Diagram, the background SigWave window should automatically update to an eye-diagram configuration.

13. Figure 43 displays our simulated 50-MHz driver eye diagram. Notice that there is a very good wide-open eye shape. This indicates that the signal integrity of this trace, operating at 50 MHz, is fairly good. We do see some overshoot problems, especially at the falling edge by about 1 V. Remember this simulation was based on the generic IBIS model, so it may not be entirely accurate. Within SigWave, in version 15.5.1, it is possible to change very easily how many eyes to display by going to *Graph* → *Eye Diagram Preferences* (Figure 44). Figure 45 displays three eyes at 50 MHz.



ver	Receiver	Cycle	GlitchTol [ns]	FTSMode	EyeHeight [mV]	EyeJitter [ns]	EyeWidth [ns]	Glitch	Monotonic	NoiseMargin [mV]	Overshoo [mV]
C.U2.2	3BITADC.U4.2	4	0.09	Typ	4997.79	[illegible]	[illegible]	PASS	PASS	1651.87	5115.12

Figure 42 - Postsimulation Results Tab

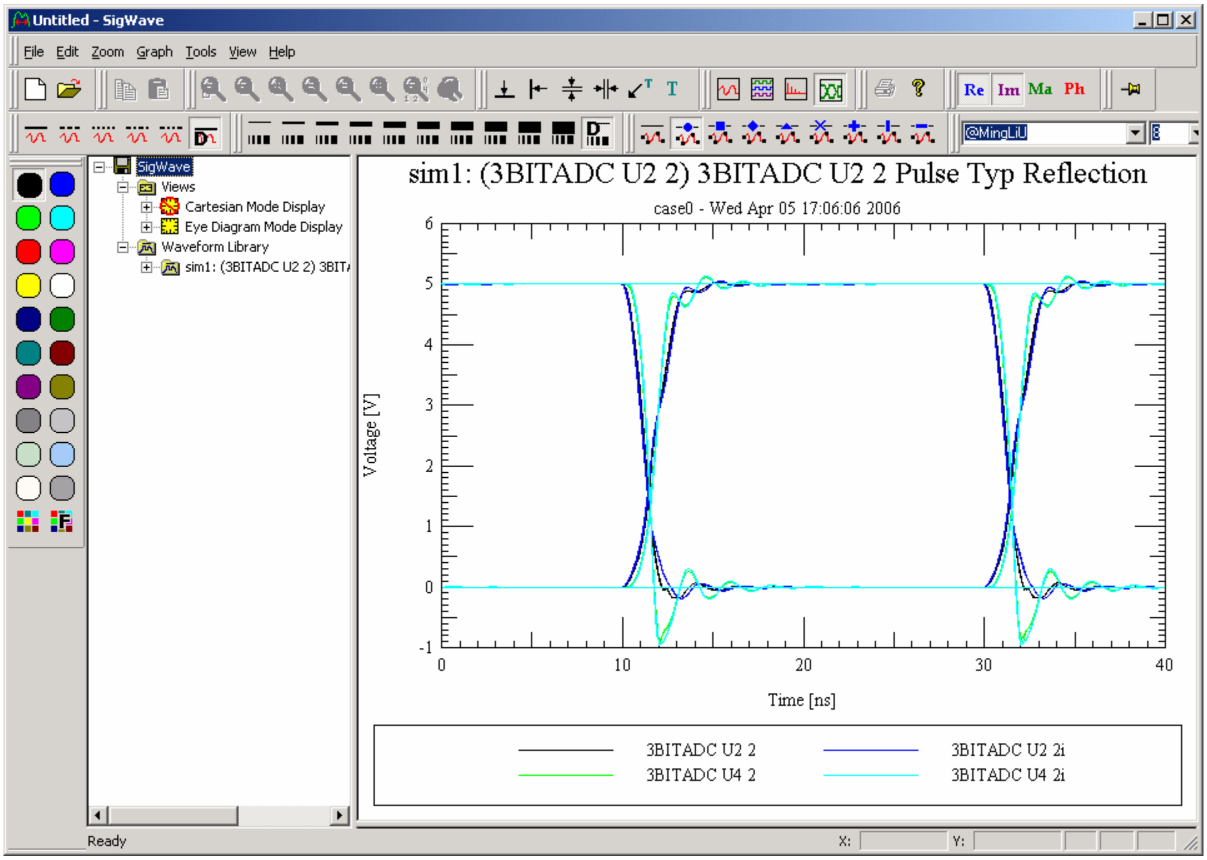
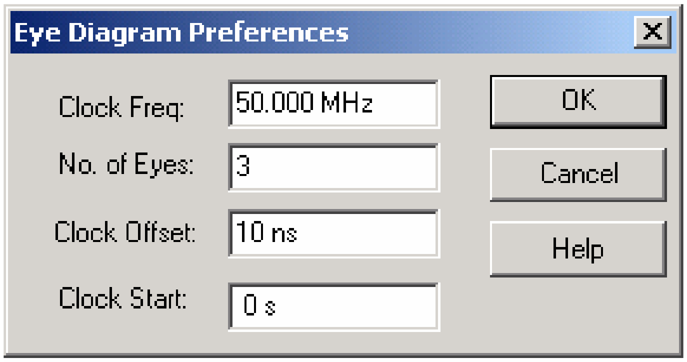


Figure 43 - A 50-MHz Eye Diagram Simulation



Eye Diagram Preferences

Clock Freq: 50.000 MHz

No. of Eyes: 3

Clock Offset: 10 ns

Clock Start: 0 s

Figure 44 - Eye Diagram Preferences Pop-Up

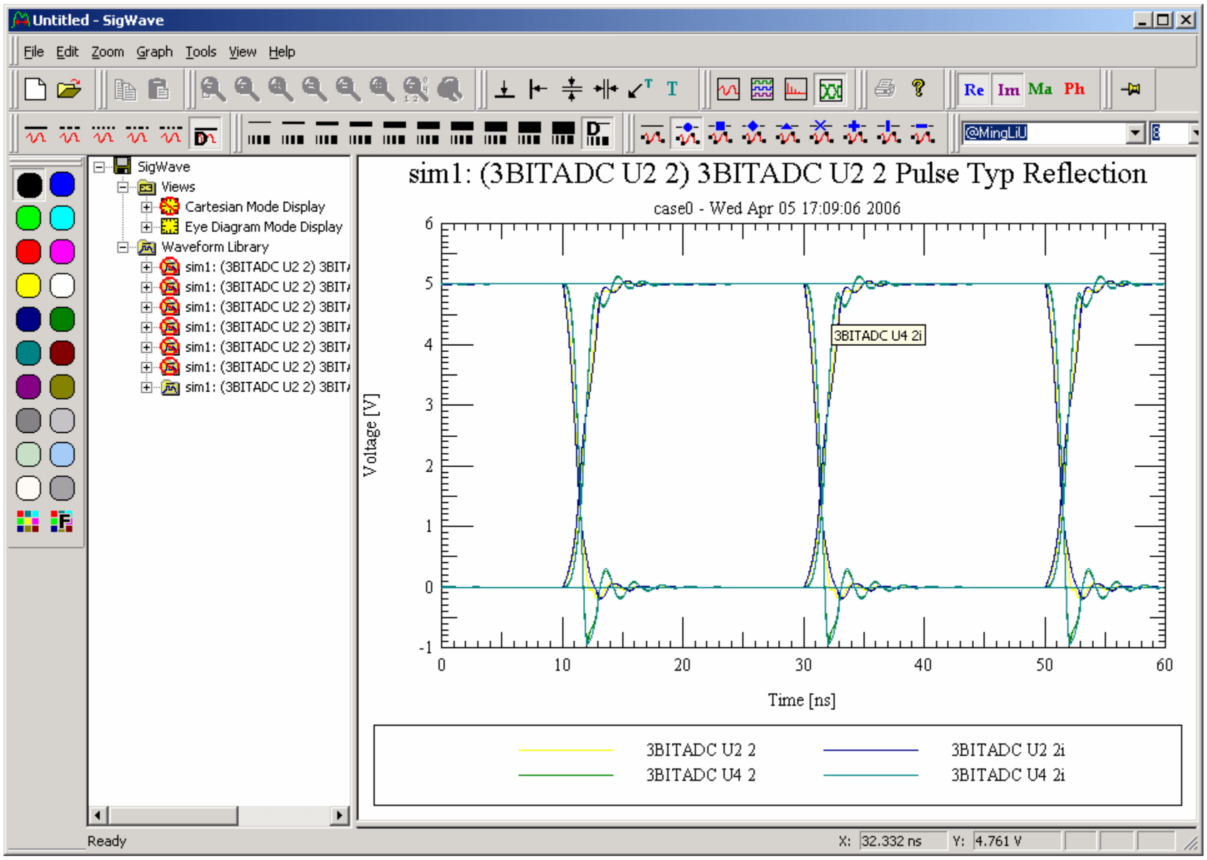

Untitled - SigWave
File Edit Zoom Graph Tools View Help
Re Im Ma Ph
@MingLiU
SigWave
Views
Cartesian Mode Display
Eye Diagram Mode Display
Waveform Library
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BITADC U2 2 Pulse Typ Reflection
case0 - Wed Apr 05 17:09:06 2006
Voltage [V]
6
5
4
3
2
1
0
-1
0
10
20
30
40
50
60
Time [ns]
3BITADC U4 2i
3BITADC U2 2
3BITADC U4 2
3BITADC U2 2i
3BITADC U4 2i
Ready
X: 32.332 ns
Y: 4.761 V

Figure 45 - Three Eyes in 50 MHz Simulation

14. To determine how well this PCB layout geometry will operate, let us simulate at 500 MHz and 1 GHz. These preferences can be changed by clicking on the “CUSTOM” text in the topology editor. (Previously it was titled “TRISTATE”.) The pop-up should look very similar to Figure 40 on page 41. Change the frequency to 500 MHz, simulate, graph, and then repeat for 1 GHz. The results are shown in Figures 46 and 47.

15. As one can see, the eye begins to close with increasing operating frequency, and appears nearly closed at 1 GHz. This is expected since parasitic elements, skin-effect, and other SI merits are more pronounced at high frequencies. From the 1-GHz diagram, we can conclude that this PCB, as is, would not be operational at 1 GHz.

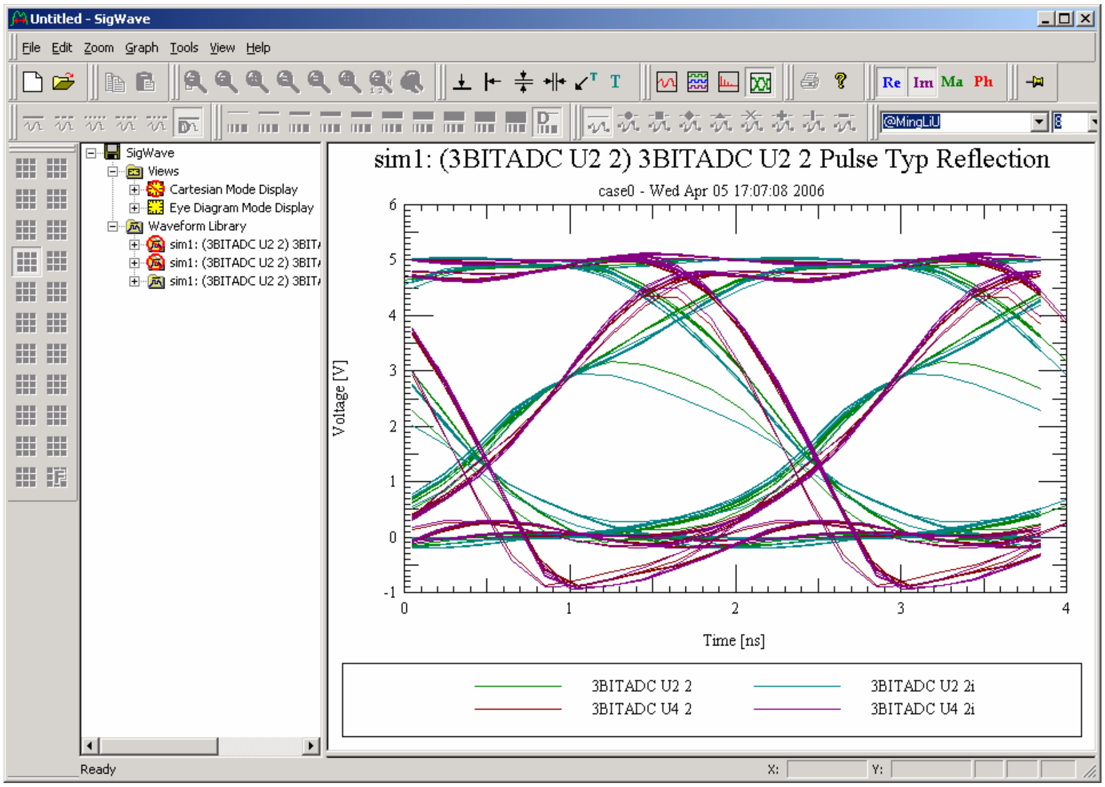
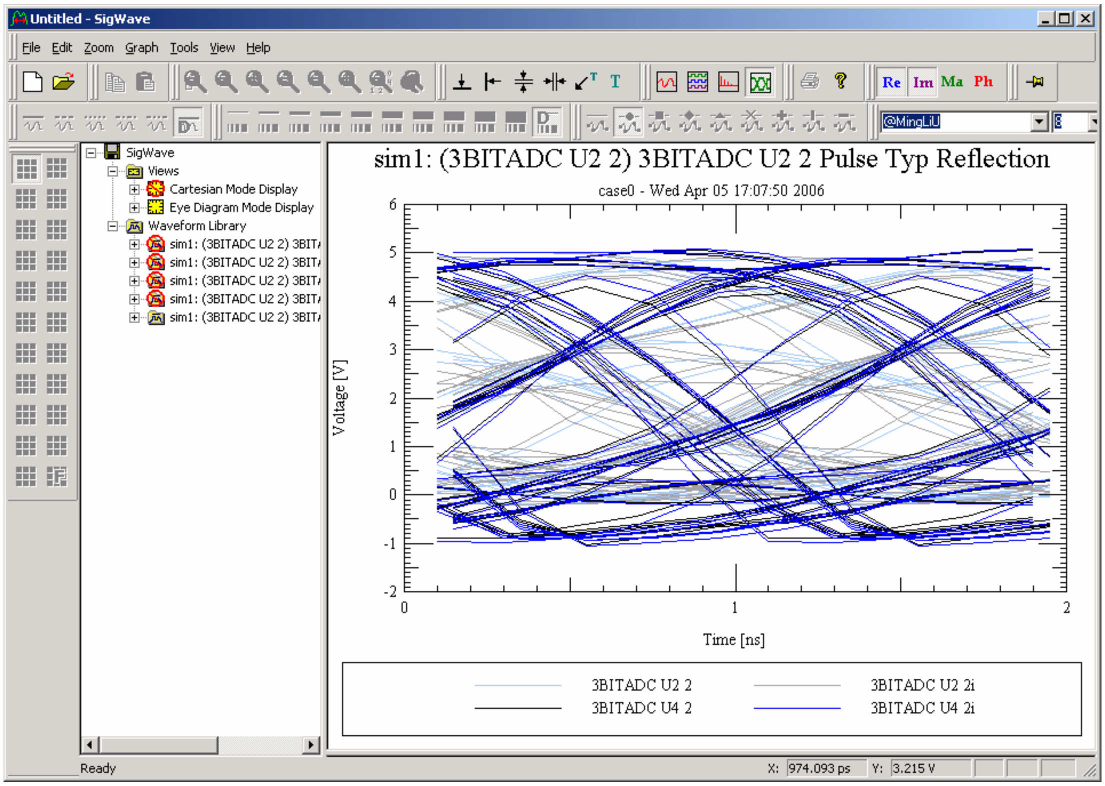


Figure 46 - A 500-MHz Eye Diagram Simulation


Untitled - SigWave
File Edit Zoom Graph Tools View Help
Re Im Ma Ph
@MingLiU
SigWave
Views
Cartesian Mode Display
Eye Diagram Mode Display
Waveform Library
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BIT/
sim1: (3BITADC U2 2) 3BITADC U2 2 Pulse Typ Reflection
case0 - Wed Apr 05 17:07:50 2006
Voltage [V]
6
5
4
3
2
1
0
-1
-2
0
1
2
Time [ns]
3BITADC U2 2
3BITADC U2 2i
3BITADC U4 2
3BITADC U4 2i
Ready

X: 974.093 ps Y: 3.215 V

Figure 47 - A 1-GHz Eye Diagram Simulation

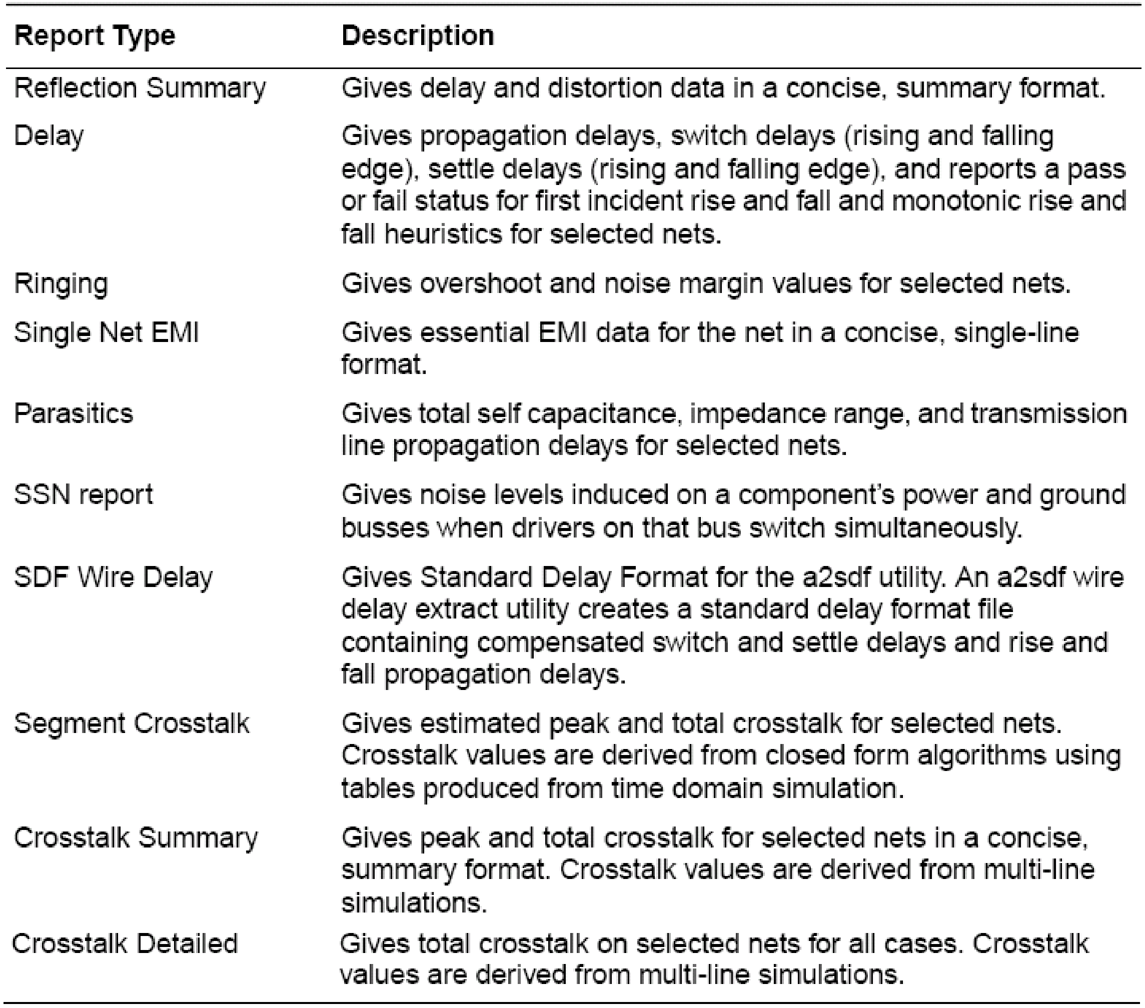
On the other hand, during the schematic entry portion of design, we had already determined that the maximum operating frequency was approximately 300 kHz (Figure 12 on page 16). Therefore, that was already the limiting factor of our design—SI issues do not come into play. In modern high-speed designs, especially beyond the 1-GHz threshold, SI issues *are* the limiting factor of operating frequency, so the PCB designer would have to iterate between routing and SI simulation to converge at a working solution. Different board materials and routing widths/lengths may also have to be considered during the process.

5.4 Other SI Simulations

16. In addition to eye diagram simulations, PCB SI has the following simulations: Reflection, Comprehensive, Crosstalk, Simultaneous Switching Noise (SSN), and Electro-Magnetic Interference (EMI) Single Net. These simulations can produce 10 standard types of analysis reports. Figure 48 summarizes these different reports.
17. In order to perform any of the aforementioned simulations, within the PCB SI interface, set up the appropriate models as needed then select *Analyze* → *EMI/SI Sim*→ *Probe*. Select the net desired again. For graphical simulations, select *Waveforms*, rather than *View Topology* as before (Figure 38 on page 40). A pop-up like Figure 49 should show up. The user can select a simulation case, which can be previously created by *Analyze* → *EMI/SI Sim* → *Initialize*. In addition, the user can specify the simulation preferences by clicking the preferences button.
18. Select appropriate tab for simulation, adjust any values or preferences, then hit the *Create Waveforms* button. After simulation, the user can highlight the waveform and

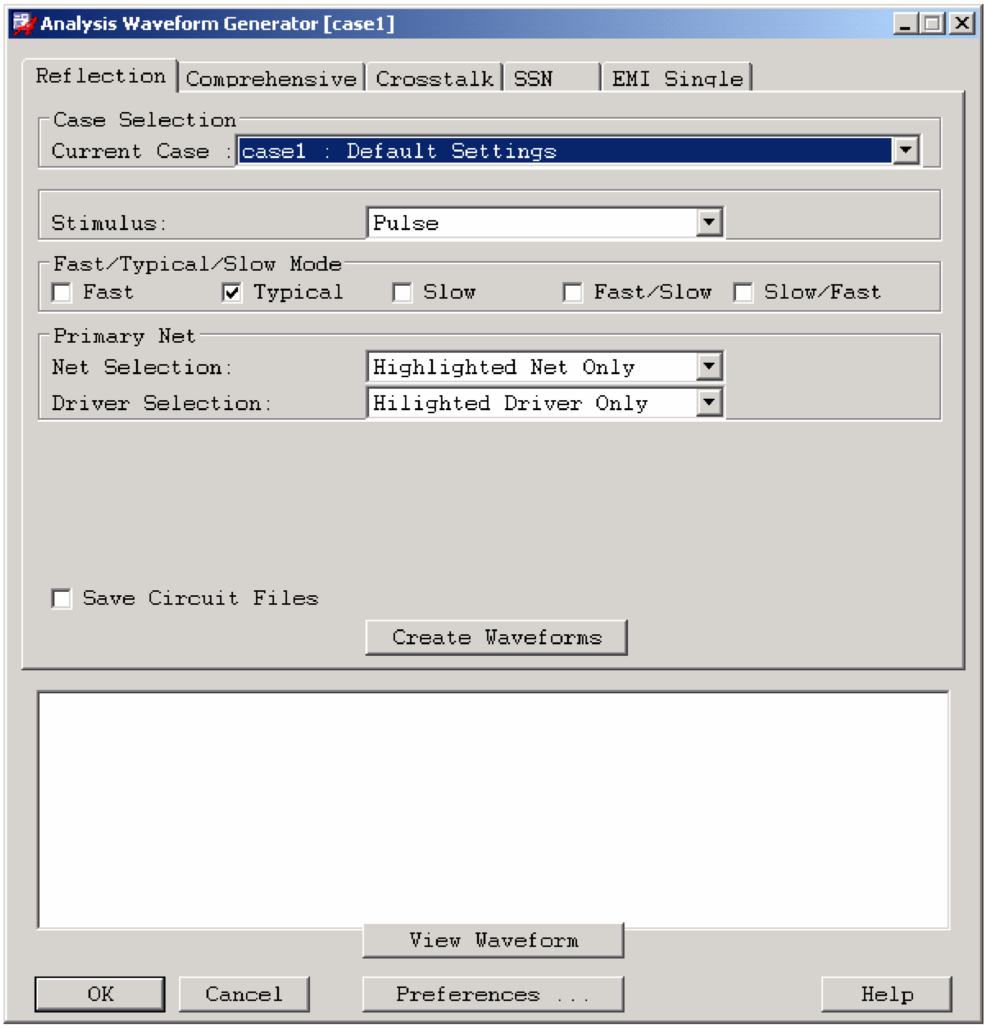
select the *View Waveform* button. SigWave will open up as in the eye diagram simulations.

19. Generating reports is similar. Instead of selecting *Waveforms* in Figure 38 on page 40, select *Reports*. A pop-up like Figure 50 will display.



Report Type	Description
Reflection Summary	Gives delay and distortion data in a concise, summary format.
Delay	Gives propagation delays, switch delays (rising and falling edge), settle delays (rising and falling edge), and reports a pass or fail status for first incident rise and fall and monotonic rise and fall heuristics for selected nets.
Ringing	Gives overshoot and noise margin values for selected nets.
Single Net EMI	Gives essential EMI data for the net in a concise, single-line format.
Parasitics	Gives total self capacitance, impedance range, and transmission line propagation delays for selected nets.
SSN report	Gives noise levels induced on a component's power and ground busses when drivers on that bus switch simultaneously.
SDF Wire Delay	Gives Standard Delay Format for the a2sdf utility. An a2sdf wire delay extract utility creates a standard delay format file containing compensated switch and settle delays and rise and fall propagation delays.
Segment Crosstalk	Gives estimated peak and total crosstalk for selected nets. Crosstalk values are derived from closed form algorithms using tables produced from time domain simulation.
Crosstalk Summary	Gives peak and total crosstalk for selected nets in a concise, summary format. Crosstalk values are derived from multi-line simulations.
Crosstalk Detailed	Gives total crosstalk on selected nets for all cases. Crosstalk values are derived from multi-line simulations.

Figure 48 - Standard Analysis Report Descriptions [10]



Analysis Waveform Generator [case1]

Case Selection
Current Case : case1 : Default Settings

Stimulus: Pulse

Fast/Typical/Slow Mode
□ Fast ☑ Typical □ Slow □ Fast/Slow □ Slow/Fast

Primary Net
Net Selection: Highlighted Net Only
Driver Selection: Hilighted Driver Only

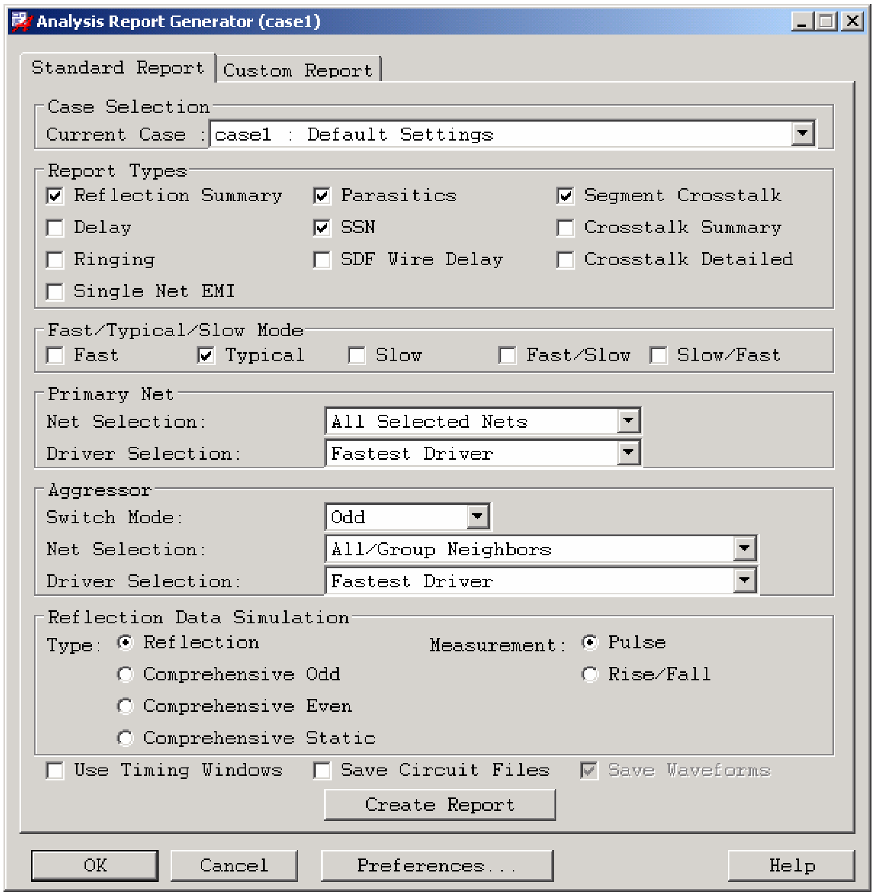
□ Save Circuit Files

Figure 49 - Five Different Simulation Types

20. After selecting which reports (four were chosen here) and configuring the preferences, just press *Create Report*. PCB SI will simulate, even using a 3D solver engine (for cross-talk reports), and then generate text-files that automatically display (Figure 51).

5.5 Going Further

PCB SI can also simulate multiple boards at once. By default, simulation is set as single-board mode. However, when initializing the setup, one can change it to multiboard mode. This becomes particularly interesting from an SI point of view, especially for high-speed backplane simulations.



Analysis Report Generator (case1)

Standard Report | Custom Report

Case Selection
Current Case : case1 : Default Settings

Report Types
☑ Reflection Summary ☑ Parasitics ☑ Segment Crosstalk
☐ Delay ☑ SSN ☐ Crosstalk Summary
☐ Ringing ☐ SDF Wire Delay ☐ Crosstalk Detailed
☐ Single Net EMI

Fast/Typical/Slow Mode
☐ Fast ☑ Typical ☐ Slow ☐ Fast/Slow ☐ Slow/Fast

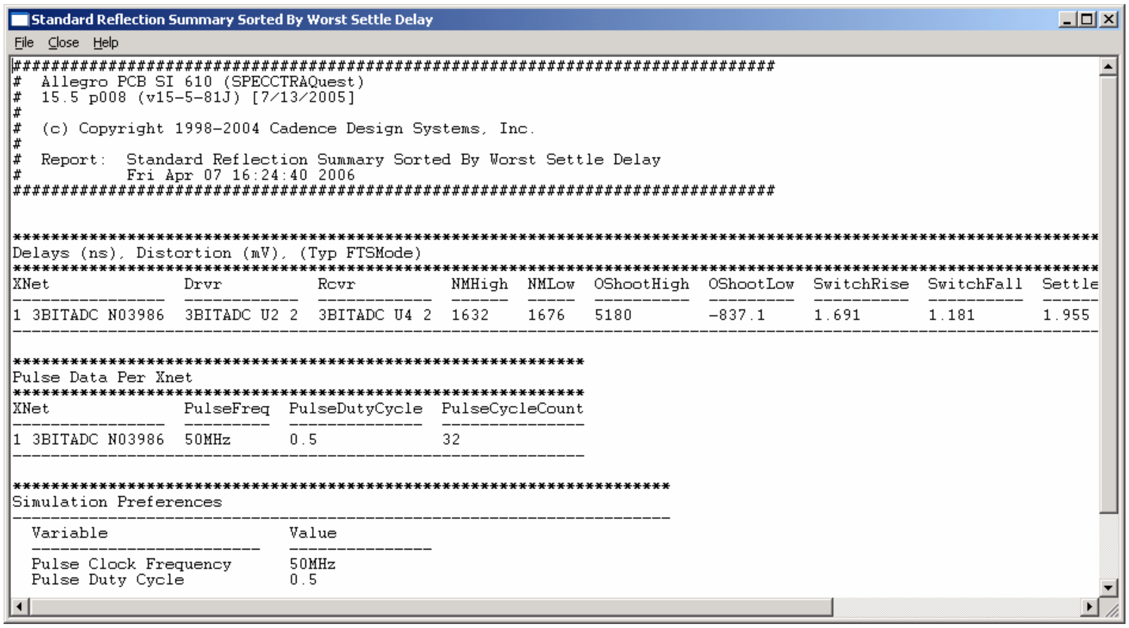
Primary Net
Net Selection: All Selected Nets
Driver Selection: Fastest Driver

Aggressor
Switch Mode: Odd
Net Selection: All/Group Neighbors
Driver Selection: Fastest Driver

Reflection Data Simulation
Type: ● Reflection
○ Comprehensive Odd
○ Comprehensive Even
○ Comprehensive Static
Measurement: ● Pulse
○ Rise/Fall

☐ Use Timing Windows ☐ Save Circuit Files ☑ Save Waveforms

Figure 50 - Ten Types of Standard Reports



Standard Reflection Summary Sorted By Worst Settle Delay

```
################################################################################
#  Allegro PCB SI 610 (SPECCTRAQuest)
#  15.5 p008 (v15-5-81J) [7/13/2005]
#
#  (c) Copyright 1998-2004 Cadence Design Systems, Inc.
#
#  Report:  Standard Reflection Summary Sorted By Worst Settle Delay
#           Fri Apr 07 16:24:40 2006
################################################################################


******************************************************************************************************************
Delays (ns), Distortion (mV), (Typ FTSMode)
******************************************************************************************************************
XNet              Drvr          Rcvr          NMHigh  NMLow  OShootHigh  OShootLow  SwitchRise  SwitchFall  Settle
----------------  ------------  ------------  ------  -----  ----------  ---------  ----------  ----------  ------
1 3BITADC N03986  3BITADC U2 2  3BITADC U4 2  1632    1676   5180        -837.1     1.691       1.181       1.955
------------------------------------------------------------------------------------------------------------------

************************************************************
Pulse Data Per Xnet
************************************************************
XNet              PulseFreq  PulseDutyCycle  PulseCycleCount
----------------  ---------  --------------  ---------------
1 3BITADC N03986  50MHz      0.5             32
------------------------------------------------------------

*********************************************************************
Simulation Preferences
---------------------------------------------------------------------
  Variable                   Value
  ------------------------   ---------------
  Pulse Clock Frequency      50MHz
  Pulse Duty Cycle           0.5
```

Figure 51 - Sample Reflection Report

6 CONCLUSIONS

In summary, three main PCB design steps were illustrated while building and verifying a simple, board-level, three-bit flash ADC. Schematic entry, performed by Design Entry CIS, completed and simulated a component level schematic. PCB routing, performed by Allegro PCB Editor, routed this ADC on a two-layer PCB using FR-4 as a dielectric. Signal Integrity simulations, performed by PCB SI, generated SI-related waveforms, and reports, including eye-diagrams at different frequencies. Each tool was illustrated step-by-step with any necessary mention to any related Allegro SPB 15.5 software such as AMS Simulator, Constraint Manager, SigXplorer, Model Integrity, and SigWave.

REFERENCES

[1] K. H. Kim, “AD converter: Custom IC design flow and the required CAD tools and Resources,” presented at EMLab, University of Illinois at Urbana-Champaign, February 13, 2006.

[2] Wolfgang Reis, “Analog-to-digital converters,” 2004, http://www.wbc-europe.com /media.php/WBC/Services/PIM_2_04_AD_Converters.pdf.

[3] Maxim Integrated Products, Appl. Note 810, pp. 1-2.

[4] Cadence Design Systems, *Allegro Constraint Manager Tutorial (Back-End) Product Version 15.5*, July 2005.

[5] Cadence Design Systems, *Allegro PCB Editor Tutorial Product Version 15.5*, July 2005.

[6] IPC 2221 Task Group, *IPC-2221A – Generic Standard on Printed Board Design*, IPC, May 2003.

[7] D. L. Jones, “PCB design tutorial,” Revision A - June 29, 2004, http://www.pcb123.com/tutorials/PDF%20Documents/PCBDesignTutorialRevA.pdf.

[8] R. Balog Jr., “Printed circuit board (PCB) overview and design issues,” 2004, http://power.ece.uiuc.edu/Balog/images/PCB%20Basics.pdf.

[9] S. H. Hall, G. W. Hall, and J. A. McCall, *High-Speed Digital System Design*. New York: John Wiley & Sons, Inc., 2000.

[10] Cadence Design Systems, *Allegro PCB SI User Guide Product Version 15.5*, July 2005.

[11] L. Greene, “Modeling and simulation for signal integrity,” Cadence Design Systems, San Jose, CA, Technical Paper, 2002.

[12] G. Breed, “Analyzing signals using the eye diagram,” *High Frequency Electronics*, vol. 4, pp. 50-53, November 2005.

[13] C. Brewster, “Re: Rise time of driver in Specctraquest,” Forum post to si-list.org, October 16, 2001.