TINA v14

The Complete Electronics Lab for Windows

USERS MANUAL

DesignSoft
www.designsoftware.com

TABLE
OF CONTENTS

DesignSoft

DesignSoft

DesignSoft

DesignSoft

6.   USING SUBCIRCUITS,

DesignSoft

DesignSoft

CHAPTER **1**

INTRODUCTION

1.1   What is TINA and TINA Design Suite?

TINA Design Suite is a powerful yet affordable software package for analyzing, designing and real time testing of circuits with analog, digital & microcontroller components and components defined in various Hardware Description Languages, VHDL, VHDL-AMS, Verilog, Verilog A, Verilog AMS, SystemVerilog, SystemC and for designing their PCB layouts. You can also analyze RF, communication, optoelectronic circuits and mechatronics applications with 3D interface.

Every year, electronic circuits become faster and more complex, and therefore require more and more computational power to analyze their operation. To meet this requirement DesignSoft engineers have included the ability to utilize the increasingly popular scalable multi-thread CPUs.

Since v12 TINA is available both in 64-bit and 32-bit editions. If you purchase TINA both versions are provided. The 64-bit version finally resolves the memory issues you may have experienced with large project.

Since 2013 TINACloud, the online version of TINA is also available. If you have licenses for both products, you can store your designs on the web and run anytime on any platforms without installation, including PCs, Macs, thin clients, tablets, smart phones, smart TVs and e-book readers. The program will run on our powerful web server with the same high speed whether you use a laptop, tablet or just a smartphone. You can then smoothly download your design from the web to your PC. Should you change something while you are on the road, continue the development off-line and upload your design again. Now TINACloud also includes a new online schematic editor with instant automatic saving of any changes you make and an integrated PCB designer.

In TINA 10 and later versions you can create and simulate multidisciplinary designs. Disciplines now include electronics, 3D mechanics and control engineering. This opens a rich new field of applications in automotive engineering, sensing and control, automation, robotics and more.

A unique feature of TINA permits you to bring your circuit to life with the optional USB controlled TINALab II and LabXplorer hardware turn your computer into a powerful, multifunction T&M instrument. With LabExplorer, you can carry out remote measurement which is great for distance education.

TINA is distributed in two major versions – TINA Standard and TINA Design Suite. TINA Standard includes circuit simulation only, while TINA Design Suite also includes the advanced PCB designer. This fully integrated layout module has all the features you need for advanced PCB design, including Multilayer flexible PCB's with split power planes, powerful autoplacement & autorouting, rip-up and reroute, manual "follow-me" trace placement, DRC, forward and back annotation, pin and gate swapping, keep-in and keep-out areas, thermal relief, fanout, plane layers, Gerber file output and much more. TINA and TINA Design Suite also have different editions tailored to customer needs. Some HDL languages and the Mechatronics extension are optional. Both the Standard and the Design Suite versions are provided in 64-bit and 32-bit editions.

TINA can also be used in the training environment. It includes unique tools for testing students' knowledge, monitoring progress and introducing troubleshooting techniques. With optional hardware it can be used to test real circuits for comparison with the results obtained from simulation. Of great importance to educators, the package includes all the tools needed to prepare educational materials.

Schematic Capture. Circuit diagrams are entered using an easy to use schematic editor. Component symbols chosen from the Component bar are positioned, moved, rotated and/or mirrored on the screen by the mouse. TINA's semiconductor catalog allows the user to select components from a user-extendible library. An advanced "rubber wire" tool is provided allowing easy modification of the schematic diagrams. You can open any number of circuit files or

subcircuits, cut, copy and paste circuit segments from one circuit into another, and, of course analyze any of the currently open circuits. TINA gives you tools to enhance your schematic by adding graphics elements such as lines, arcs, arrows, frames around the schematic, and title blocks. You can also draw non-orthogonal (diagonal) components such as bridges and 3-phase networks.

Live 3D Breadboard Tool. You can take your design for a solder-less breadboard (sometimes called a "whiteboard") and automatically build a life-like 3D picture of the breadboard. Now when you run TINA in interactive mode, virtual components such as switches, LEDs, instru- ments, etc. become "live" and will function with satisfying realism. Students will use the Live 3D Breadboard tool to prepare and docu- ment eye-catching lab experiments.

PCB Design. TINA Standard includes only circuit simulation, while TINA Design Suite includes TINA's advanced PCB designer. This fully integrated layout module has all the features you need for advanced PCB design, including Multilayer PCB's with split power planes, powerful autoplacement & autorouting, rip-up and reroute, manual and "follow-me" trace placement, DRC, forward and back annota- tion, pin and gate swapping, keep-in and keep-out areas, thermal relief, fanout, plane layers, bus & differential drawing tools, circuit block copying, 3D view from any angle, and much more. With TINA Design Suite you can prepare a PCB in at least two ways: using the G-Code control files to make in-house prototypes with milling machines using the G-Code control files provided by TINA; or send-ing Gerber files to PCB manufacturers.

Electrical Rules Check (ERC) will examine the circuit for questionable connections between components and display the results in the Electrical Rules Check window. ERC is invoked automatically, so missing connections will be brought to your attention before analysis begins.

Schematic Symbol Editor. In TINA, you can simplify a schematic by turning portions of it into a subcircuit. In addition, you can create new TINA components from any Spice subcircuit, whether created by yourself, downloaded from the Internet, or obtained from a manufacturer's CD. TINA automatically represents these subcircuits as a rectangular block on your schematic, but you can create any shape you like with TINA's Schematic Symbol Editor.

Library Manager. TINA has large libraries containing Spice- and S-parameter models provided by semiconductor manufacturers such as Analog Devices, Texas Instruments, National Semiconductor, and others. You can add more models to these libraries or create your own Spice- and S-parameter library using TINA's Library Manager (LM).

IBIS model Support. IBIS (Input/output Buffer Information Specification) is a method to provide modeling information about the input/output buffers of integrated circuits. The good thing about IBIS models that they are often available even for devices where complete device models are not available from manufacturers. One of the most popular uses of IBIS models is Signal Integrity Analysis, including impedance matching and more. TINA currently supports the most widely used IBIS 4.2 version.

In TINA you can convert IBIS models to Spice macros and then use them in any circuits in TINA. You can also complete simplified digital device models e.g. MCUs with IBIS models to better describe their analog behavior. The use of IBIS models in detail is described in the Advance Topics Manual of TINA.

Parameter Extractor. Using TINA's Parameter Extractor you can also create component models that more closely represent actual real world devices by converting measurement or catalog data into model parameters.

Text and Equation Editor. TINA includes a Text and Equation Editor for annotating schematics, calculations, includes graphic output, and measurement results. It is an invaluable aid to teachers preparing problems and examples. You can also create popup texts wich are displayed when the cursor is moved above their title.

The **Interpreter** is a powerful tool which extends TINA's usefulness. It allows the evaluation of mathematical expressions, solution of equations and systems of equations, calculation of derivatives and integrals, plotting of results, and much more. The Interpreter is available at many parts of TINA, including the text editor, excitation editor, post processor, design tool, educational training tool and more. A great feature of TINA that component names and values are automatically available in the Interpreter as variables, so it can easily be used for additional calculations, comparisons, post processing and more. Examples for the use of Interpreter are available in the

Examples\Interpreter folder. Interpreter in text editor of TINA can be activated from the Tools menu by clicking the Interpreter.

Python. In addition to the Interpreter, from v14 of TINA the Python programing language (www.python.com) is also available in TINA. Python is a modern high-level, general-purpose programming language with extensive libraries and lot of free programs available for various topics which can be use in TINA. You can find use examples in the Examples\Python and Examples\Design Tool folder of TINA.

The circuit diagrams and the calculated or measured results can be printed or saved to files in standard Windows BMP, JPG, WMF and CFG format. These output files can be processed by a number of well known software packages (Microsoft Word, Corel Draw etc.). Netlists can be exported and imported in Pspice format and also to drive popular PCB packages such as ORCAD, TANGO, PCAD, PROTEL, REDAC and other programs.

DC analysis calculates the DC operating point and the transfer characteristic of analog circuits. The user can display the calculated and/ or measured nodal voltages at any node by selecting the node with the cursor. For digital circuits, the program solves the logic state equation and displays the results at each node step-by-step.

Transient analysis. In the transient and mixed mode of TINA you can calculate the circuit response to the input waveforms that can be selected from several options (pulse, unit step, sinusoidal, triangular wave, square wave, general trapezoidal waveform, .WAV file, White noise and user-defined excitation) and parameterized as required. For digital circuits, programmable clocks and digital signal generators are available. Power dissipation end efficiency calculations are also included.

Piecewise Linear (PWL) Solver. From v14, in addition to Spice solver TINA also includes a PWL (Piecewise Linear) solver. A unique feature of TINA that is automatically creates the PWL models of semiconductors in PWL mode. This provides up to 10 times faster solution depending on the structure of the Spice models. You can find examples of PWL analysis with explanations in the Examples\PWL folder of TINA v14. The solver for PWL analysis can be selected in the Options dialog of the Analysis menu of TINA.

DesignSoft

PWL Analysis is also very useful at Multsine Analysis as it shortens the Transient analysis included here.

Auto convergence. Convergence that is obtaining solution is one of the most complicated task in circuit simulation as due to the strongly nonlinear nature of electronic circuits. Although TINA is one of the best converging simulator software on the market sometimes manual parameter settings might be needed to achieve convergence. There are several analysis parameter sets are available to be used in case of convergence problems. In TINA 12 and later versions In TINA v12 and later versions these parameter sets are automatically applied in case of need and the user can also add more settings.

Transient Noise Analysis. Noise effects are usually simulated with linear AC noise analysis which is also available in TINA. However when the noise influences the system behavior in a nonlinear way, linear noise analysis is no more satisfactory and transient noise analysis that is simulation in the time domain is necessary. A few examples:

- Analysis of systems with low signal-to-noise ratio
- Noise analysis of oscillator circuits
- Analysis of noise effects in digital circuits

The voltage and current generators of TINA now include a parameterizable white noise signal, and application circuits are available to generate other typical noise signals, which makes transient noise analysis possible.

Fourier analysis. In addition to the calculation and display of the response, the coefficients of the Fourier series, the harmonic distortion for periodic signals, and the Fourier spectrum of non-periodic signals can also be calculated.

Digital Simulation. TINA now includes a very fast and powerful simulator for digital circuits. You can trace circuit operation step-by-step, forward and backward, or view the complete time diagram in a special logic analyzer window. In addition to logic gates, there are ICs and other digital parts from TINA's large component library.

HDL simulation. TINA now includes all major analog, digital and mixed Hardware Description Languages: VHDL, VHDL-AMS, Verilog, Verilog A, Verilog AMS, SystemVerilog and SystemC to verify

DesignSoft

designs in analog, digital and mixed-signal analog-digital environments. Your circuits can contain editable HDL blocks from the libraries of TINA and Xilinx or other HDL components created by yourself or downloaded from the Internet. TINA compiles HDL into highly efficient machine code for speed optimization. You can freely combine HDL and Spice macros and the schematic components of TINA. Also you can edit the VHDL, Verilog, Verilog A&AMS source of HDL components then simulate and see the result instantly. With the built in HDL debugger you can execute VHDL, Verilog, Verilog A&AMS components step-by-step, add breakpoints, watchpoints, display variable information, etc. The SystemC source components you can edit and compile with MS Visual C and then add to TINA as high performance compiled components. Examples for use of different HDL languages are available under the Examples\HDL folder of TINA 14 and higher versions.

Microcontroller (MCU) simulation. TINA includes a wide range of microcontrollers (PIC, AVR, 8051, HCS, ARM, XMC) which you can test, debug and run interactively. The built in MCU assembler allows you to modify your assembler code and see the result promptly. You can also program and debug MCUs in C, using external C compilers including the MPLAB-XC compilers. In TINA v12 and later versions more than 1400 MCU models are are available in TINA for simulation and PCB design.

Flowchart Editor and Debugger. Writing MCU assembly code is often a hard and tedious task. You can simplify software development and gain more time to design the electronics hardware if, instead of manual coding, you use TINA's Flowchart editor and debugger to generate and debug the MCU code. This easy-to-use tool works with symbols and flow control lines with which you can represent the algorithm you want. TINA also supports other code generators, the free-of-charge XMC code generation platform DAVE from Infineon Technologies and the FLOWCODE graphical programming language from Matrix Technology Solutions Limited.

AC analysis calculates, complex voltage, current, impedance, and power can be calculated. In addition, Nyquist and Bode diagrams of the amplitude, phase and group delay characteristics of analog circuits can be plotted. You can also draw the complex phasor diagram. For

non-linear networks, the operating point linearization is done automatically.

Multisine analysis calculates the frequency response of circuits without linearization using Transient Analysis with a special excitation consisting of multiple sinusoidal voltages. This is especially useful in case of SMPS circuits where AC analysis is possible through special, so called average models only, which cannot be automatically created.

You can find examples of Multisine analysis with explanation in the Examples\Multisine folder of TINA v14.

Network analysis determines the two-port parameters of networks (S, Z, Y, H). This is especially useful if you work with RF circuits. Results can be displayed in Smith, Polar, or other diagrams. The network analysis is carried out with the help of TINA's network analyzer. The RF models of the circuit elements can be defined as SPICE subcircuits (SPICE macros) which contain parasitic components (inductors, capacitors) or as an S-parameter model defined by its S (frequency) function. S functions are normally provided by the component manufacturers (based on their measurements) and can be downloaded from the Internet and inserted into TINA either manually or by using TINA's library manager. In TINA v14 new S-parameter models such a Circulator, Directional Couplers and more have been included. Examples with RF components are included in the Examples\RF folder.

Analytic SMPS and other device modelling. Using the powerful featuers of TINA Interpreter and the integrated Python language lighting fast simulation and design of SMPS circuits became possible. This is especially important for semiconductor manufacturers and their customer for quick initial design and simulation of devices. In TINA Industrial v14 both Spice, HDL and Analytic modelling is possible. You can find examples in the Examples\Design Tool folder.

Linear AC Noise Analysis determines the noise spectrum with espect to either the input or the output. The noise power and the signal-to-noise ratio (SNR) can also be calculated.

Symbolic analysis produces the transfer function and the closed form expression of the response of analog linear networks in DC,

AC, and transient modes. The exact solution, calculated through the symbolic analysis, can also be plotted and compared to the numerically calculated or measured results. The built-in interpreter can evaluate and plot arbitrary functions.

Monte-Carlo and Worst-case analysis. Tolerances can be assigned to the circuit elements for use in Monte-Carlo and/or worst-case analyses. The results can be obtained statistically, and their expected means, standard deviations and yields can also be calculated.

Design Tool This powerful tool works with the design equations of your circuit to ensure that the specified inputs result in the specified output response. The tool offers you a solution engine that you can use to solve repetitively and accurately for various scenarios. The calculated component values are automatically set in place in the companion TINA schematic and you can check the result by simulation. This feature is also very useful for semiconductor and other electronics component manufacturers to provide application circuits along with the design procedure.

Optimization. TINA's enhanced optimization tool can tweak one or more unknown circuit parameters to achieve a predefined target response. The target circuit response (voltage, current, impedance, or power) must be "monitored" by meters. For example, you can specify several working point DC voltages or AC transfer function parameters and have TINA determine the values of the selected components.

Post-processor. Another great new tool of TIN A is its post-processor. With the post-processor, you can add new curves of virtually any node and component voltage or current to existing diagrams. In addition, you can post-process existing curves by adding or subtracting curves, or by applying mathematical functions to them. You can also draw trajectories; i.e., draw any voltage or current as a function of another voltage or current.

Presentation. With TINA you can make quality documents incorporating Bode plots, Nyquist, Phasor, Polar and Smith diagrams, transient responses, digital waveforms and other data using linear or logarithmic scales. Customize presentations easily using TINA's advanced drawing tools-you can print your plots directly from TINA, cut and paste them into your favorite word processing package, or export them to popular standard formats. Customization includes

complete control over texts, axes, and plot style; e.g., setting line width and color, fonts in all sizes and color, and automatic or manual scaling for each axis. In TINA v12 and later versions the cursor display is integrated into the diagram window and it is possible to display all curves under the cursor.

Interactive mode. When everything is in order, the ultimate test of your circuit is to try it in a "real life" situation using its interactive controls (such as keypads and switches) and watching its displays or other indicators. You can carry out such a test using TINA's interactive mode. You can not only play with the controls, but you can also change component values while the analysis is in progress. In addition, you can assign hotkeys to component values and switches to change them simply by pressing a key. You will immediately see the effect of the change. You can also test MCU applications in TINA's interactive mode. You can not only run and test them using the several lifelike interactive controls e.g., keyboards, but you can also debug them while the MCU executes ASM code step by step, And displays the register contents and TINA's outputs in each step. If necessary you can modify the ASM code on the fly and test your circuit again without using any other tool.

Virtual instruments. In addition to standard analysis presentations such as Bode and Nyquist plots, TINA can present its simulation results on a wide range of high-tech virtual instruments. For example, you can simulate the time response of your circuit using a virtual square wave generator and a virtual oscilloscope. Using TINA's virtual instruments is a good way to prepare for the use of real test and measurement equipment. Of course it is important to remember that the "measurement results" obtained with virtual instruments are still simulated. From v11 TINA also includes virtual instruments (to be found under the Meters component tab) for Efficiency, Average values and Frequency.

Real-time Test & Measurements. TINA can go beyond simulation when supplementary hardware is installed on the host computer. With this hardware, TINA's powerful tools can make real-time measurements on real circuits and display the results on its virtual instruments.

Training and Examination. TINA has special operating modes for training and for examination. In these modes, under TINA's control,

the students solve problems assigned by the teacher. The solution format depends on the types of problems: they can be selected from a list, calculated numerically, or given in symbolic form. The interpreter - providing a number of solution tools - can also be used for problem solving. If the student cannot solve the problem, he/she can turn to the multilevel Advisor. The package includes all the tools needed to produce educational materials. A collection of examples and problems worked out by teachers is also part of the package. Another special educational function of TINA is the software or hardware simulation of circuit faults to practice troubleshooting. Using TINA, you can transform existing PC classrooms into contemporary electronics training labs at low cost.

Mechatronics Extension. With this optional add-on package you can create and simulate multidisciplinary designs currently including electronics, 3D mechanics and control engineering. You can place light sources, light sensors, motors and actuators in TINA's mechanical window and connect with their counterparts in the analog, digital mixed electronic circuits. You can control the mechanics from the electronics part of TINA even with complex software written in C or assembly language, then compile and execute the code in the MCUs while running the electronic and 3D mechanical simulation simultaneously.

1.2   Available Program Versions

Different program versions, tailored to meet various needs, are available.
TINA is distributed in to major versions TINA and TINA Design Suite. TINA includes simulation only while TINA Design Suite includes our new advanced PCB designer too.

Both versions are available with the following features:

• **Industrial version:** Includes all of TINA's features and utilities.

• **Network version:** TINA can be used under most well known networks including Microsoft, Linux, Novell, Citrix and more.

This feature is especially recommended for corporate and educational use.

- **Educational version:** It has most features of the Industrial version but parameter stepping and optimizations are allowed for one parameter only, Stress Analysis and the Steady State Solver are not included.

- **Classic Edition:** It has the same features as the Educational version above, except that Network Analysis is not allowed, TINA's large S-parameter component library and the Parameter Extractor, Stress Analysis and the Steady State Solver are not included. IBIS model support, Dissipation and Efficiency calculation and the Autoconvergence tool are not included.

- **Student Version:** Has the same features as Classic Edition version except that the circuit size is limited to 100 nodes including internal Spice macro nodes. The number of pads on the PCB layout is also limited to 100. Global Parameters and HDL extensions are not allowed.

- **Basic version:** Has the same features as Classic Edition except that the circuit size is limited to 200 nodes including internal Spice macro nodes. The number of pads on the PCB layout is also limited to 200. Global Parameters and HDL extensions are not allowed.

- **Basic Plus version:** Has the same features as Classic Edition except that the circuit size is limited to 800 nodes including internal Spice macro nodes. The number of pads on the PCB layout is also limited to 800. Global Parameters and HDL extensions are not allowed.

OPTIONS:

- **HDL Extension:** Extends the default VHDL hardware description language in TINA with VHDL-AMS, Verilog, Verilog A and Verilog AMS, SystemVerilog and SystemC.

- **Mechatronics Extension** add-on package: Create and simulate multidisciplinary designs simultaneously including electronics, 3D mechanics and control engineering.

DesignSoft

1.3   Optional supplementary hardware

1.3.1   TINALab II High Speed Multifunction PC Instrument

With TINALab II you can turn your laptop or desktop computer into a powerful, multifunction test and measurement instrument. Whichever instrument you need; multimeter, oscilloscope, spectrum analyzer, logic analyzer, arbitrary waveform generator, or digital signal generator it is at your fingertips with a click of the mouse. In addition TINALab II can be used with the TINA circuit simulation program for comparison of simulation and measurements as a unique tool for circuit development, troubleshooting, and the study of analog and digital electronics.

TINALab II includes a DC to 50MHz bandwidth, 10/12 bit resolution, dual-channel **Digital Storage Oscilloscope**. Due to its advanced equivalent-time sampling technology, TINALab can acquire any repetitive signal with up to **4GS/s equivalent sampling rate**, while in single shot mode the sampling rate is 20 MS/s. The full scale input range is ±80V, with 5mV to 20V/div ranges.

The synthesized **Function Generator** provides sine, square, ramp, triangle and arbitrary waveforms from DC to 4MHz, with logarithmic and linear sweep, and modulation up to 10V peak to peak. Arbitrary waveforms can be programmed via the high level, easy to use language of TINA's Interpreter. Working automatically in conjunction with the Function Generator, the **Signal Analyzer** measures and displays Bode amplitude and phase diagrams, Nyquist diagrams, and also works as **Spectrum Analyzer.**

Digital I/O for the high-tech **Digital Signal Generator** and Logic Analyzer instruments allow fast 16-channel digital testing up to 40MHz.

The optional **Multimeter** for TINALab II allows DC/AC measurements in ranges from 1mV to 100V and 100 mA to 1A. It can also measure DC resistance in ranges from 1Ω to 1MΩ**.**

You can also plug **Experimenter Modules** into the slot on the front of TINALab II, allowing you to simulate, measure, and troubleshoot virtually the whole range of analog and digital electronics.

Using TINALab II with TINA gives you the unique capability to have circuit simulation and real time measurements in the same integrated environment. This provides an invaluable tool for troubleshooting and brings your designs to life by comparing simulated and measured results.

1.3.2 LabXplorer: Multifunction Instrument for Education and Training with Local and Remote Measurement capabilities

LabXplorer turns your desktop, laptop, tablet or smart phone into a powerful, multifunction test and measurement instrument for a wide range of applications. Instruments, whatever you need, are at your fingertips. LabXplorer provides multimeter, oscilloscope, spectrum analyzer, logic analyzer, programmable analog and digital signal generator, impedance analyzer and also measures characteristics of passive electronic components and semiconductor devices.

LabXplorer can be used with its virtual instruments both stand-alone or remotely through the Internet or LAN.

It also supports the TINA circuit simulation program and its cloud based version TINACloud for comparison of simulation and measurements as a unique tool for circuit development, troubleshooting, and the study of analog and digital electronics.

In remote mode Labexplorer's virtual instruments run on most OSs and computers, including PCs, Macs, thin clients, tablets--even on many smart phones, smart TVs and e-book readers. You can use LabXplorer remotely in the classroom, computer lab, at home, and, in fact, anywhere in the world that has internet access. LabXplorer comes with various, remotely programmable, plug-in analog, digital and mixed circuit experiment boards.

CHAPTER 2

NEW FEATURES IN TINA

This chapter describes the new features and changes in the latest TINA v14 and also in the previous v12, 11, 10, 9.x , 8.0 and 7 versions. Many of the new features were suggested by TINA users, while others were created by DesignSoft's team. We are sure you will share our excitement about these new features.

2.1   List of New features in TINA v14

- Multisine simulation
- PWL (Piecewise Linear) Solver
- Inverse Laplace Transform
- New RF components: Circulator, 3 and 4 port Directional Couplers
- User-defined Fast Analytic solver for SMPS and other devices both in Interpreter and Python
- 7 HDL languages: VHDL, VHDL-AMS, Verilog, Verilog-A, Verilog-AMS, SystemVerilog, SystemC.
- Faster solution algorithms
- Python support in the Design Tool and in a separate shell
- DC-DC Converter Application Circuit Search
- TINA and TINACloud e-book (440 pages) with Tutorial videos
- TINA v14 runs on Mac with one step installation including Wine
- New components and much more

New device models

- Infineon XMC4400 MCU

- Infineon Gate drivers: 2edf8275f, 2edf9275f, 2eds9265h, 2eds7165h

- Infineon/IR POL regulator ICs: IR3823A, IR3899A, IR3447A, IR3846A, TDA38820, TDA38840

- TI voltage reference models: REF7012, REF7025, REF7030, REF7033, REF7040, REF7050

- TI SAR ADC models: ADS79xx, ADS8860

- STMicroelectronics: ST1S40 DC step-down switching regulator IC

- Sensors: Ultrasonic, Light, Infrared, PIR

2.2   List of New features in TINA v12

- 32-bit and 64-bit versions
- Power dissipation and efficiency calculations
- IBIS model import and analysis
- Integrated cursor display in the diagram window
- Cursor displays for all curves in transient diagrams
- Syntax highlighting in Spice and HDL editors
- HDL library compilation and management
- Global parameter stepping
- Enhanced RF component insert dialog
- S-parameter wizard
- Spice control command editor (.AC, .DC, .NOISE, .TRAN)
- Enhanced support of PSpice format digital blocks
- Running Spice simulation from command line (external tool for netlist format)
- Running TINA from command line (.TSC format)
- Store last settings of virtual instruments
- Control for automatic separation of outputs in the diagram window

Convergence improvements

- • Autoconverge (Transient simulation with system or user defined different parameter sets)
- • Transient simulation with zero initial voltages if operating calculation fails

New device models

- • New Delta-Sigma ADC devices
- • BSIM3V3.2 model
- • GaAs devices
- • GaN devices

Advanced MCU support

- • More than 500 new MCUs, 1400 MCUs total
- • PIC16 microcontrollers (222 devices)
- • STM32 F4, F7 microcontroller support (227 devices)
- • PIC18, PIC32 CAN interface support
- • Texas Instruments Tiva C Series TM4C123x Cortex-M4 microcontroller support (51 devices)
- • MCP23S17 I/O expander support
- • Texas Instruments Sitara AM3358 processor support
- • Serial Monitor window for monitoring serial communication

Other new features

- • Interpreter functions for standard values for components (E series)
- • Set global parameter as input of DC transfer calculation
- • Add comments to design tool input parameters
- • Hotkey editor - define custom hotkeys
- • Show partial result of transient simulation if simulation is aborted
- • Advanced analysis control links

PCB Design

- • Bus connection
- • Transmission line design (Differential pair)
- • Block repetition (Macro-block copy)

2.3   List of New features in TINA v11

- 8,000+ new parts, including power electronics devices

- SystemC support

- Create Digital filters in SystemC and run in TINA

- Add MCUs in SystemC to TINA

- Infineon design folder with industrial designs

- Industrial Power, Lighting, Motor Control application circuits

- XMC microcontrollers and application circuits

- Support of the free-of-charge XMC code generation platform DAVE.

- Efficiency-, Average value-, and Frequency-meters

- Analysis control links

- Advanced Macro editing

- Advanced Macro editing

- SAR and Sigma-Delta ADCs

- DACs with SPI

- I2C , SPI bus simulation

- PM bus, SM bus simulation and monitors

- Transient Noise Analysis

- Transient Noise Generator

- Popup text

- DACs with SPIBSIM4 modelDACs with SPI

- Microchip XC8 compiler support

- FLOWCODE 7 support

- Arduino support

- Smart wire

- Enhanced I/O assignment

- Post processing of Fourier Spectrum

- Export of Diagrams in CSD format (Common Simulation Data File)
- Advanced "Remember diagram settings" option, saved with the circuit

PCB Design

- Importing in 3D Enclosure models in industry standard formats
- Visualization of PCB design with Enclosures in 3D
- Exporting PCB with Enclosure in industry standard formats
- 3D printer support
- Importing Footprints in 2D and 3D in industry standard formats

2.4 List of New features in TINA v10

- Windows 8 compatibility
- Open project files with preview of schematics and mechatronics
- Edif import
- Global Spice variables
- Integration with TINACloud, files upload and download
- Optional Mechatronics extension
- KLU - a faster solver on large-size circuits
- Enhanced and accelerated VHDL and Verilog simulation 10x times faster than in v9
- Xilinx simprim simulation in digital and in mixed mode
- Verilog A
- Verilog AMS
- MCU C compiling on 8051, AVR, PIC16, PIC18, PIC32, ARM
- Linux simulation on ARM MCU
- PSpice compatible AD-DA interface
- Oscilloscope settings are stored
- Oscilloscope works together with the interactive mode
- Frequency and waveform parameters are shown with cursors

DesignSoft

- Hotkeys to AC and Transient Analysis
- Cursor Max and Min functions added to the Process menu

2.5   List of New features in TINA v9

- Full Vista and Windows 7 compatibility
- Multi core support for dual, quad, i7 and higher multi-core processors
- Enhanced analysis speed up to 10 times higher speed on 1 core, 15x on dual core, 20x on quad core
- Improved convergence performance based on the latest state of the art algorithms
- Advanced powerful Spice-VHDL mixed mode analysis
- Extended catalog with ARM 7, ARM 9 and HCS08 microcontrollers, more SMPS ICs models, realistic ADC and DAC models, LCD display, Bi-color LEDs
- Extended MCU simulation, USB and other modules
- Programmable Design Tool where users can implement design procedures for calculating and setting circuit parameters so that the circuits can produce predefined target output values.
- Extended flowchart tool: PIC, AVR, 8051, code box, USART, external interrupt handling
- Open and save TINA designs, models and libraries directly from the Web
- Import Spice .CIR and .LIB files directly from the Web
- Import Libraries, Examples and Designs from any earlier versions of TINA v7 and later.
- Show analysis results on diagrams during analysis
- Fast diagram drawing and processing speed, even for very large circuits

PCB Design

- G-Code export of PCB designs for creating control files for milling machines
- Revised and largely extended PCB footprint catalog

2.6   List of New features in TINA v8

- Vista style installation and folder scheme
- Controlled sources
- Powerful Spice-VHDL co-simulation including MCUs
- Finite State Machine (FSM) editor with VHDL generation
- Flowchart editor an debubgger for controlling MCUs
- Any number of MCUs in one circuit
- Extended MCU catalog including PIC18, CAN and more
- Execution time measurement and statistics for Transient Analysis
- Hyperlinks can be added to schematics and to the diagram window
- Extended semiconductor catalog
- Application examples from Texas Instruments
- Labview based virtual instruments
- Interface to build LabVIEW based virtual instruments
- Wave (.wav) files can be used as input
- New Open Examples command in file menu to open built in examples
- Autosave. Save your current schematic or PCB design at adjustable time intervals.
- Parameter adding possibility to Spice subcircuits
- On line update possibility for libraries, program or both
- Post-processing formulas are stored with schematics, editable later
- Live 3D Breadboard (displaying and animating circuits with 3D parts on a virtual 3D breadboard)
- 3D virtual instruments to prepare and document lab experiments
- Integrated electronic design textbook with "live" circuits (optional)
- SMPS design templates from Christophe Basso (optional)
- Detection of components or nodes linked with convergence or irregular circuit problems.

PCB Design (Only in TINA Design Suite v8 Edition, additional to TINA v8)

- Creating "flex" PCBs including 3D display
- Creating PCBs of any shape including round edges
- Buried and blind vias
- Extended catalog
- Improved optimizing autorouter
- Display of complete 3D circuits including parts connected externally to the PCB

2.7   List of New features in TINA v7

- Much faster analog solver algorithm with improved convergent properties
- Integrated VHDL support
- User defined VHDL components with VHDL source code
- VHDL components containing VHDL source code can edited and executed instantly
- MCU support including wide range of PIC processors and more
- Built in debugger and assembler compiler for MCUs
- Assembler code of MCUs can be edited and executed instantly
- External simulator and debugger for VHDL
- 3D component view in the schematic editor to review if the design is PCB ready
- Passive and active filter design
- SMPS (Switching Mode Power Supply) design support (Steady State Solver)
- Control of interactive mode from the new Interactive menu
- Stress Analysis

- Advanced integrated PCB design
  - Multi-layer PCBs
  - Autoplacement
  - Autorouting
  - Rip-up and reroute
  - Follow-me trace placement
  - DRC
  - Forward and back annotation
  - Pin/Gate swapping
  - Keep-in/out areas
  - Thermal relief,
  - Fanout
  - Gerber file output
  - Copper pour
  - Split planes
  - Footprint editor with multi-pin footprint wizard
  - 3D view of PCB boards
- Support to design multi-pin schematic symbols
- More advanced Logic Design (simplification) tool
- IF statement allowed in Spice netlists
- More advanced file export (EMF, BMP, JPG) (File/Export)
- More advanced file import (EMF, WMF, BMP, JPG) (Insert/Graphics)
- Copy and Paste of any Windows dialogs (captured by Alt Prt Scr) into the Schematic Editor.
- Extended virtual instrument for real-time XY-recording, with average value, RMS calculation and recording vs. time

DesignSoft

INSTALLATION AND
START-UP

3.1   Installation Procedure

3.1.1   Minimum hardware and software requirements

- Intel Pentium or equivalent processor

- 1 GB of RAM

- 300 MB of available hard disk space

- CD-ROM (in case of CD ROM installation)

- Mouse or touchpad

- VGA adapter card and monitor

- Microsoft Windows XP, 7, 8, 10, 11

- Supported Networks (for Network versions): MS Windows 2000/ 2003/2008/2012/2016/2019/2022 Server or later, Linux Server, Novell Netware version 3.12 or later.

    If the program is copy protected by a hardware key, the minimum hardware configuration also includes a USB port.

3.1.2 Installation from CD-ROM or from the WEB

3.1.2.1 Installation from CD-ROM

To begin the installation simply insert the CD into your CD-ROM drive. The Setup Program will start automatically if the Auto-Run function of your CD-ROM has been enabled (Windows-Default). If not, Select Start/Run and type:

D:SETUP (Enter) (where D represents your CD-ROM drive).

The setup program will start.

3.1.2.2 Installation from the WEB

If you have purchased a downloadable version of TINA you need to download it using the download link sent by email after your successful purchase.

Windows usually saves downloaded files in the Downloads folder which is located under your user name in the **users** folder on the drive where Windows is installed (for example C:\users**your name**\downloads). Double-click the file to start the setup. If you purchased the Mechatronics add-on package, you should install it separately after installing TINA.

NOTE:

This software may come with copy protection. For further details see the Copy Protection and the Network Installation sections.
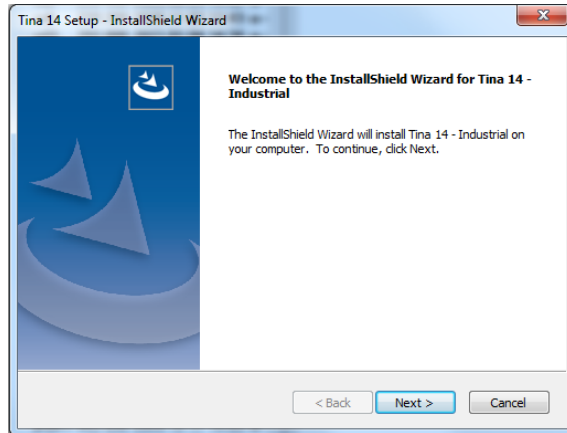
3.1.3   Following the Installation Steps

TINA's Setup Procedure follows the steps standard with most Windows Programs. There are several screen pages where you can enter or change important installation choices, such as Type of Installation, Destination Directory, etc. To continue installation, click on [Next>]. You can always step back, using the [<Back]-Button. If you do not want to continue installation for any reason, click on [Cancel]. If you elect to cancel installation, the program will ask you if you really want to exit. At this point you can either resume or exit Setup.

3.1.4   Welcome and Software License Agreement

To begin the Procedure click on Next on the Welcome Page. The first step is the Software License Agreement.
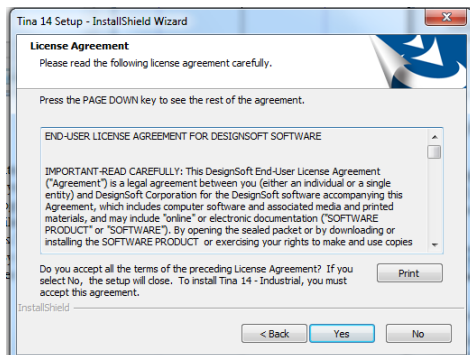
NOTE:

By clicking on "Yes" you are agreeing fully with DesignSoft's Terms and Conditions for using this software.

3.1.5   Entering User Information

This data is used to personalize your copy of the software. By default, the installation program picks up the data entered when you set up Windows. You accept these names as defaults by clicking on Next or you can change them.

Depending on your program version you might also need to enter a Serial Number located on your CD-ROM package or on your Quick Start Manual.

DesignSoft
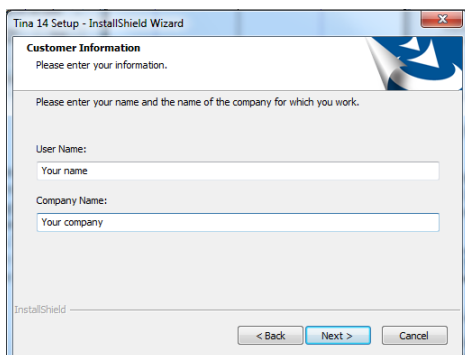
3.1.6   Platform Selection

The installer **includes both 32-bit and 64-bit TINA versions**. It will automatically choose the right version to install.
If you have 64-bit Windows and want to install 32-bit TINA version you can select the 32-bit version but not vice versa. Click on Next >

to continue.
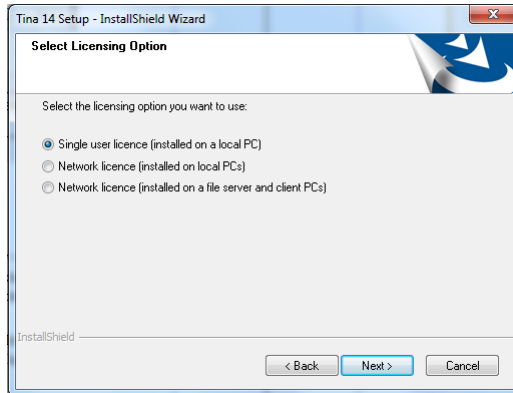
3.1.7   Single User License and Networking Options

3.1.7.1   Single user license (installed on a local PC)

Select this option if you have a single user license and want to use a single copy of TINA on a local PC.

However if you purchased a network license and want to use TINA in a network environment you can choose between 2 options.

DesignSoft
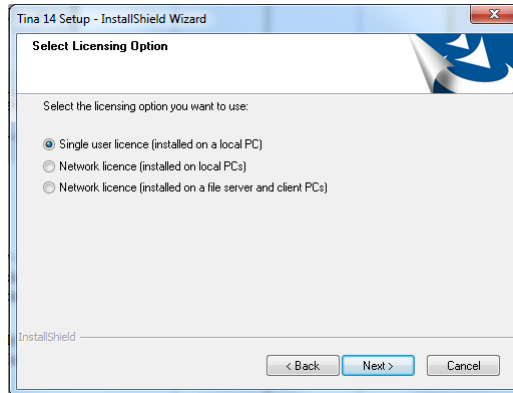
3.1.7.2   Network license installed on local PCs

Select this option if you want to use the server as a license server. In this case the server stores the license information and the software is installed on the workstations. After you've installed the package on the first workstation start TINA and select the location of the license information file on the file server then authorize the package. Finally install the software on every other workstation (client) where you want to use TINA. On these stations when you start TINA for the first time you have to locate the license file on the server.

No other authorization is required.

3.1.7.3   Network license installed on file server

Select this option if you want to use the server as a file server. In this case the server store the files of the software and the license information as well. You have to select a network share in the destination dialog of the setup and install the package there. Next you have to install the package on the first workstation and authorize it. Select the Run commands from the Windows Start menu, enter the command U:\Tina\NWSetup\setup, where U: represents your network drive, and follow the instructions. Note that TINA is the main program directory holding TINA on the server. Start TINA on the first workstation.

Note that when you authorize the first workstation full access is needed on the network share where TINA is located. Authorize TINA. Finally you must run the setup program on every other workstation (client) where you want to use TINA. No authorization is required on these stations.

3.1.8 Choose Destination Location

Here you can select an Installation Directory other than the one suggested as a default. The default is the Windows Standard Directory for Programs. To change the directory, click on Browse and select a different drive and/or directory from the Choose Folder Dialog.
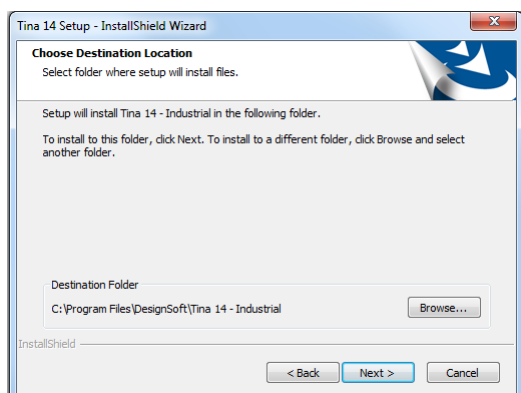
IMPORTANT NOTE:

> *If you are installing TINA for Windows to a hard disk that already has an earlier version of TINA, you must be sure to use a new directory name for TINA for Windows, such as the suggested directory, C:\Program Files\DesignSoft\Tina, or the working files you have already created will be overwritten and lost. If uncertain, exit setup, copy your TINA files safely to another hard disk directory or to floppy disks, then resume setup.*

3.1.9 Selecting a Setup Type

TINA offers you three different types of Setup. You can either run a Typical Setup (Default) a Compact Setup or a Custom Setup.
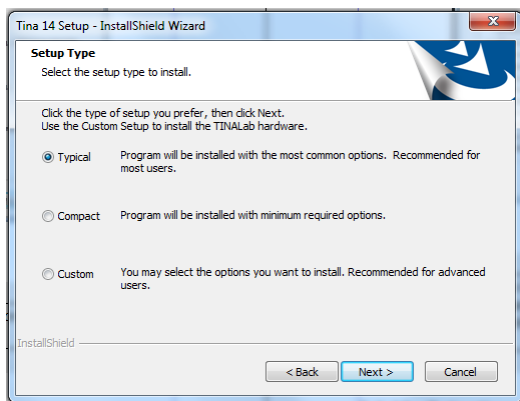
NOTE:

> The detailed settings for the Compact installation are made after you select Compact and click on Next.

3.1.9.1 Typical

The commonly used components are installed. This includes Program Files, Samples and Utilities (i.e. Exam Manager, Spice Library Manager).

3.1.9.2 Compact

Only the most important components are installed. This results in a usable TINA installation, but one without certain program components, such as Exam Manager.

DesignSoft

3.1.9.3 Custom

You will be able to decide which components are to be installed. The default settings are similar to those of the Typical installation. Deselect the unwanted components or select the missing ones.
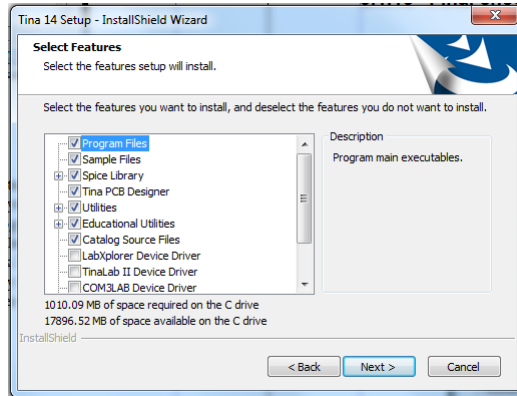
NOTE:

> *If you wish to install the TINALab Card, TINALab II or other third party supplementary hardware, you must select the Custom installation option at the time of installation and check the appropriate device driver on the list.*

3.1.10 Selecting the Program Folder

Here you can choose where in the Programs Section of your Windows Start Menu the Program Icons will appear. The default is a new sub menu called for example TINA 14 – Industrial. You can change this name or select an existing Program Folder from the list.

DesignSoft
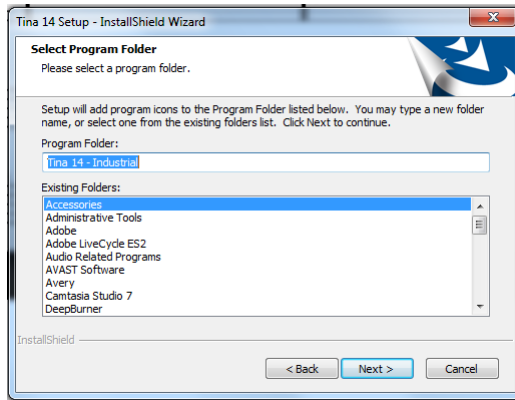
3.1.11   Select Environment Options

In TINA you can set up the settings, private/shared catalog and temporary folder. The Settings folder stores your personal settings. The private catalog folder will store your catalog files, while the shared catalog folder can be used to share catalog files with other users of the same PC or with other users in the network. The temporary folder stores the temporary files of the software. By default these folders are set to common Windows folders however you may change the folders by pressing the browse button.

DesignSoft
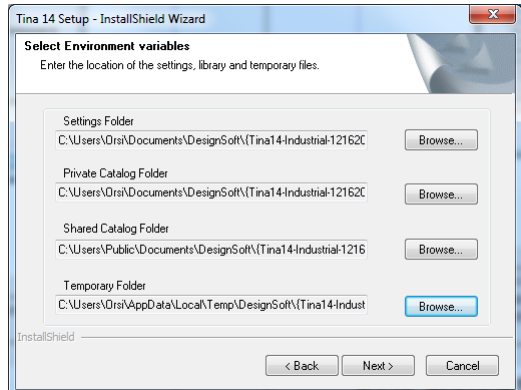
3.1.12 Selecting the Symbol Set

TINA can display its component schematic symbols according to the US (ANSI) or the European (DIN) conventions. Select the one appropriate for you.

DesignSoft
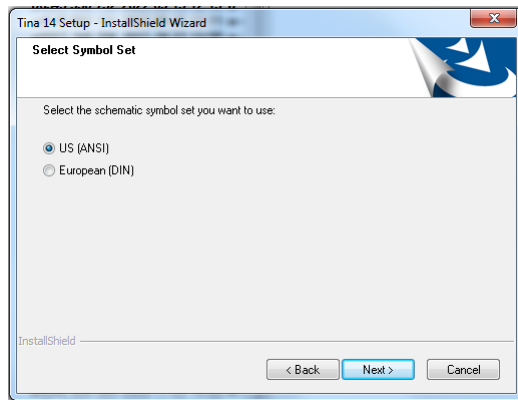
3.1.13 Final check and copying the files

This page lists the settings you have made, giving you an opportunity to check these settings and alter them and step back if changes are necessary. After you click on Next, the Setup Program starts copying the files automatically.
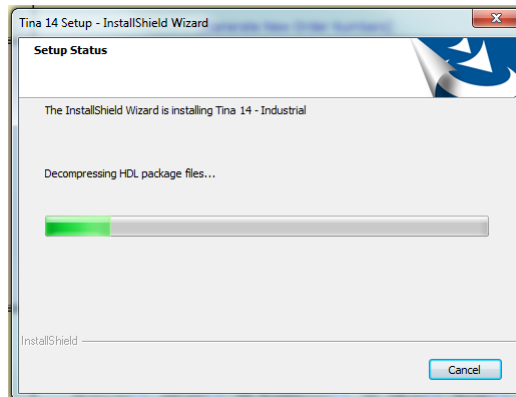
3.1.14 Completing the Setup

After all the selected files have been copied and the Start Menu entries created, you are asked if you want to place a Shortcut to the TINA program file on your Desktop. The last page indicates a successful installation and invites you to open and read a file with the latest information about TINA. We urge you to take a moment and review that file. Click on finish when you're ready.

DesignSoft
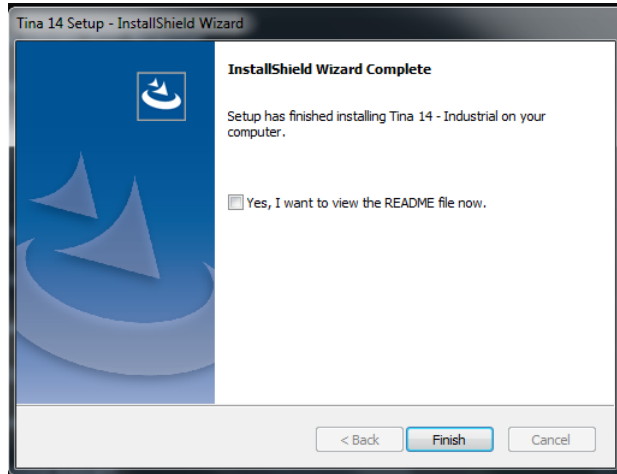
NOTE:

You can read the latest information in the file again at any time by selecting Read Me from the Tina Start Menu Entries. You can also get the latest information about changes or new features by visiting our Web Site, www.tina.com.

3.2 Uninstalling TINA

You can uninstall TINA at any time. Note that this will not delete files you have created.

1. To begin Uninstallation locate Uninstall Tina on the Start or Apps screen of Windows or in the Start Menu of earlier Windows versions.

2. Click on Uninstall Tina.

3. In the next dialog click on Yes if you are positive you want to uninstall TINA .

DesignSoft

INSTALLATION

3.3 **Maintaining or Repairing an Installation**

You can modify or repair an existing installation of TINA as follows: Select Settings and Control panel from the Start menu of Windows. Click on the Add or Remove programs icon. Locate your TINA installation on the list and press the Change button (Press the Remove button if you want to uninstall the software). The installer of TINA will start and you can either modify the existing installation by adding or removig components or repair the current installation or uninstall the pacakage.
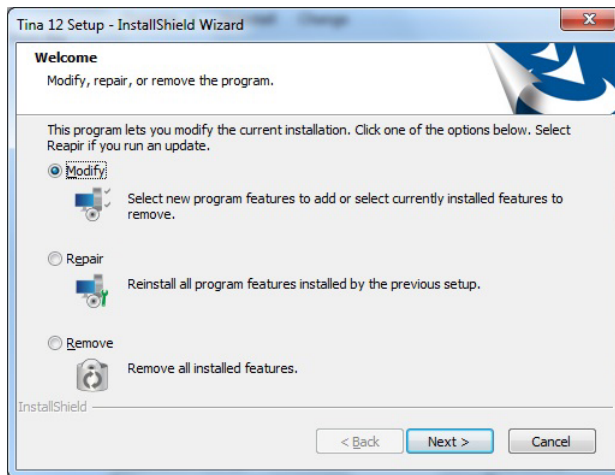
3.4 **Network Installation**

To install the Network version of TINA, you must have administrative privileges on the server machine and you have to share a disk volume or directory on your network, the place where TINA will be installed. With file sharing users can access TINA files. The share must be writeable by the administrator user during the installation process.

Therefore, carry out the following steps to make all files in the specified directory shared:

DesignSoft

Novell Netware 3.x: Logon to the server and perform the following commands:

```
FLAG *.* S SUB
```

Novell Netware 4.x and later versions:

```
FLAG *.* +SH /S
```

Linux server: You need Samba, a free software suite to provide file services for Windows clients. Logon as root and create a Samba share directory on your Linux system, adding the following section to your /etc/samba/smb.conf file, then restart Samba service. Example:

[TINA]

comment = TINA install folder

path = /TINA

writeable = yes

admin users = administrator root

valid users = TINAUserGroup

read list = TINAUserGroup

store dos attributes = yes

Later, you might have problem mapping Linux Samba shares to Windows Vista/7 client. Then check Vista LAN Manager authentication level: open the Run command and type "secpol.msc" and click OK. Go to Local Policies, Security Options and navigate "Network Security: LAN Manager authentication level" and open it. Change the settings from "Send NTLMv2 response only" to "Send LM & NTLM - use NTLMv2 session security if negotiated". Once you have done this, Windows Vista will be able to view network drives based on Samba servers.

Windows Server: Logon as Administrator and use the NET SHARE command, for example:

NET SHARE TINAFolder="C:\Program Files\DesignSoft\TINA"

Or, you can use the Windows Explorer:

1. Right-click the drive or folder, and then click Sharing and Security.
2. Select the "Share this folder" option and type a Share name.

DesignSoft

3. Click the Permissions button and ensure that the administrator have Full control permission and click OK twice.

Windows Client:

Next make sure that the clients have a mapped drive set to the network drive containing the TINA program folder.

To assign (map) a drive letter to a network computer or folder do the following:

1. Open Windows Explorer

2. On the Tools menu, click Map Network Drive.

3. In Drive, select a drive letter, e.g.: G:

4. In Path (Win9x/Me) or Folder (NT/2000/XP/Vista/7), select from the drop-down list or type in the network drive (server and share name: \\MyServer\Volume1) or folder name to which you want to assign (map) a Drive letter (\\MyServer\Volume1\Program Files\DesignSoft\TINA) . Note, that share name refers to a shared folder on the server. On Windows NT/2000/XP/Vista/7 you can use Browse to find the network computer, drive and folder.

5. Set the Reconnect at Logon checkbox, then press OK.

Then execute the installation procedure directed by section 3.1.1 on the mapped disk volume, that is accessible from the network.

After you have set everything up on the network disk according to the instructions above, you must run the setup program on each client where you want to run TINA. Start setup.exe (in some versions nsetup.exe) from the TINA\NWSETUP directory.

When you run setup.exe you must specify the working directory which should be located on a local drive of the workstation.

The working directory can be on the network; however in this case the path of this directory must be different on every work station. After you've specified the working directory, you may install the optional measurement hardware for TINA (e.g TINALab). After running setup.exe, you will be able to run TINA simultaneously on any number of workstations, just as though each workstation had a single user version.

Network versions are copy protected and need authorization. For the details regarding the special procedures required for authorizing networked computers, refer to section 3.5.

3.5 Copy Protection

Executing the authorization procedure described in this section, you will be able to run TINA. If you use a network version of the program, you will be able to do it simultaneously on the number of workstations the program licensed, just as though each workstation had a single user version.

3.5.1 Copy protection by Software

If your version of TINA is copy-protected by software you need to authorize it.

You have to be in Administrator mode when you authorize the program.

NOTE FOR VISTA AND WINDOWS 7, 8 AND 10:

> *Even though you are a user configured as an "Administrator", Vista and Windows 7 treats you as a standard user so the authorization may not succeed unless you make sure that User Account Control (UAC) is enabled in Vista. This is enabled by default, so if you didn't turn it off manually it should be OK. (You can find this setting in the Control Panel when you enter 'UAC' in the search field in the upper right corner of the Window.)*

Authorize the program with the following steps:

1. Run Authorization & Trial from the Start or Apps screens of Windows or from the TINA group of the Start menu.

2. The Authorization Manager dialog will appear.

In most cases you should just press OK to continue. The License Status dialog will appear showing your initial authorization status.

If the program starts but the License Status dialog does not appear, select the Authorization/ Authorize from the Help menu of the Schematic Editor. You will normally have 31 trial sessions to provide you enough time obtain the authorization.

-        Press the Authorize button on the License Status dialog which is displayed at program start or select Authorization/Authorize from the Help menu of TINA.

-         Enter your 16 digit Order number into the Order number field of the Authorize dialog appearing and press OK. For successful operation you must be connected to the Internet and your firewall should allow communication with our server.

- If the above is not possible for any reasons select the Other tab on the Authorize dialog.

- Email your Site code to DesignSoft using the link in the Authorize dialog or contact your dealer.

- We will email back a Site key, which should be copied into the Site key field of the Authorize dialog.

- Press OK to finish the authorization.

NOTE FOR NETWORK VERSION:

> *If you use a network version of Tina then it will run on workstations where the setup with the NWSETUP\SETUP.EXE program has already been done. It is sufficient to authorize the software at one workstation. This will allow to run the software simultaneously, as long as the total number of simultaneous users does not exceed the licensed number. In some cases the program comes with a Serial number, which needs to be entered during installation. When entering the serial number you do not need an active internet connection. For more information refer to the program's Authorization Help by pressing the Help button.*

INSALLATION

Authorization in secure environment

If you started the program with the "Authorization and Trial" command but you still get the "Please log on with administrator privileges!" message it is a sign that you have a secure environment and need a special installation. This may be the case in systems at larger companies.

In this case select Mode: "Authorization in secure environment" in the Authorization Manager dialog. However you need to consult with DesignSoft or with your dealer and ask for a special Order number before you continue.

DesignSoft

3.5.2 Copy Protection by Hardware (dongle)

3.5.2.1 Single user version

Make sure you are in Administrator mode.

If you have a USB dongle-protected version of TINA, install TINA first before connecting the dongle.

Next, connect the dongle to the USB port. The dongle driver installation will begin. If Windows looks for the dongle driver, select the recommended option, which is your hard disk.

If the dongle is not connected or not installed correctly the following error message will appear:

```
Hardware protection key is not present
                 (USB).
```

3.5.2.2 Site license with multi-user dongle (DSPROTKEY)

To avoid loss or damage of dongles you may have just one dongle for a whole site. In this case you need the dongle only at the first start of the program at each workstation.

1) Install the software on each computer as described in the manual for single computers, according to the number of licenses purchased for your site.

2) Start the program with the dongle plugged in, then close the program and remove the dongle. After you authorized all workstations this way keep the dongle at safe place. If on some workstations the license is lost for any reasons (e.g. disk crash), you can reinstall the program and authorize again with the same technique. Please do not use this possibility to install the software on more workstations than licensed, because this might eliminate the possibility to recover lost licenses.

3.5.2.3 Authorization with network dongle

If you have a network dongle do the following to authorize TINA on the server.

1.   After setting up the workstations (see section 3.4), login on one workstation as Administrator (with writing right to the volume where TINA resides).

DesignSoft

2.   Connect the dongle with the above workstation. The system must recognize the dongle and the LED on the dongle should light.

3.   Start TINA. Based on the information in the dongle TINA will be authorized for the number of users licensed, and a dialog box should appear to confirm this.

4.   Remove the dongle and keep it at a safe place, as you may need it to recover the license in case of a system crash.

5.   Now TINA should run on all workstations without the dongle.

3.6   Starting Up

After successfully installing TINA, you can start the program by simply double-clicking the TINA icon on your Desktop or by choosing Tina from the TINA Start Menu Entries.

3.7   Experimenting with Example Circuits, avoiding common problems

Start the program and click the *File* menu item in the top line of the screen to drop down the *File* menu. Select the *Open* command and the standard open file dialog box appears with **.TSC*, indicating that a file name with *.TSC* extension is sought. Select the *EXAMPLES* folder, and a list of files with *.TSC* extensions will appear. After selecting a file, the circuit schematic will appear.

Now you can execute an analysis, modify or expand the circuit, and evaluate the results. Keep in mind that every command may be aborted by pressing the *[Esc]* key or clicking on the *Cancel* button.

We recommend that you load the following circuits and follow the instructions on the screen for the circuit types listed below.

This will avoid some common problems.

Oscillator circuit	EXAMPLES*colpitts.tsc*
555 Oscillator	EXAMPLES*555_AST.tsc*
Rectifier circuit	EXAMPLES*Bridge Rectifier1.tsc*

INSALLATION

DesignSoft

GETTING STARTED

In this chapter, we present TINA's screen format and menu structure. A step by step introduction is given using examples.

4.1   Schematic Editing Using the Mouse

Here are some basic mouse techniques to help you edit schematics:

4.1.1   Using the right mouse button

If you press the right button of the mouse at any time, a popup menu appears. Using this menu you can:

- **Cancel Mode:** Exit from the last operation (e.g. moving a component, drawing a wire).

- **Last Component:** Return to the last component and reposition it.

- **Wire:** Switch to wire-drawing mode. In this mode, the cursor turns into a pen and you can draw a wire. For more details, see the **Wire** paragraph below.

- **Delete:** Delete selected component(s).

- **Rotate Left, Rotate Right, Mirror:** Rotate or mirror the component currently selected or being moved. You can also rotate a selected component by pressing the Ctrl-L or Ctrl-R keys.

- **Properties:** Use this command to edit the properties (value, label) of the component currently selected or being moved. From the Properties menu, you can set all parameters of a component (before it is placed). This lets you place multiple copies of the component, all with the properties just entered. While you are in the component property editor, the right mouse button has another function. When you are editing the field of any component parameter other than the label field, you can copy that field to the label field by pressing the right mouse button and then selecting the *Copy to Label* command. You can accomplish the same thing by pressing *[F9]*.

4.1.2   Using the left mouse button

In the descriptions below, 'clicking' always refers to the left mouse button.

- **Selection:** Clicking on an object will select the desired object and deselect all other objects.

- **Multiple selection:** Clicking while holding down the [Ctrl] key will add the object under the cursor to the group of currently selected objects. If the object under the cursor is already in the currently selected group, clicking will remove it from the group.

- **Block selection:** To select a block of objects all at once, first make sure there is no object under the cursor. Then press and hold down the left button while moving the mouse (dragging). This will create a rectangular block, and all objects within the block will be selected.

- **Selection of all objects:** Press Ctrl+A to select all objects.

- **Moving objects:** A single object can be moved by dragging it (Position the cursor on the object, press and hold the left button, and move the mouse.) Multiple objects can be moved by first selecting them (see above), then clicking the left button while the cursor is over one of the selected objects, holding the left button down, and dragging.

- **Parameter modification:** Double-clicking on an object will bring up its parameter menu so that you can modify its parameters (if it has any).

- **Crossing wires:** The crossing of two wires does not result in a connection at the crossing unless you deliberately choose there to be one. Use Edit.Hide/Reconnect to place or remove a connecting dot. However, it is better drafting practice to never make a connection at a wire crossing, as this avoids ambiguity about the presence or absence of a dot.

- **Block or symbol copying:** After a block or symbol has been selected, you may copy it by pressing Crtl+C. Then click outside the block or symbol to release it, and press Crtl+V. You will see a copy of the block which you can place as you wish. If the schematic window doesn't show enough room for the copy, press Alt - to zoom out. Once you've located the block, click the left mouse button once to anchor it and a second time to deselect the moved block.

4.2   Measurement Units

When setting parameters for electronic components or specifying numerical values, you may use standard electronic abbreviations. For example, you can enter 1k (ohm) for 1000 (ohm). The multiplier abbreviations should follow the numeric value, e.g., 2.7k, 3.0M, 1u, etc.

The following characters indicate multiplier factors:

$p = pico = 10^{-12}$	$T = tera = 10^{12}$
$n = nano = 10^{-9}$	$G = giga = 10^{9}$
$u = micro = 10^{-6}$	$M = mega = 10^{6}$
$m = milli = 10^{-3}$	$k = kilo = 10^{3}$

NOTE:

Upper and lower cases must be carefully distinguished (e.g., M = m), and the selected letter must follow the numeric characters without a space (e.g., 1k or 5.1G), or TINA will indicate an error.

GETTING STARTED

4.3 The Basic Screen Format
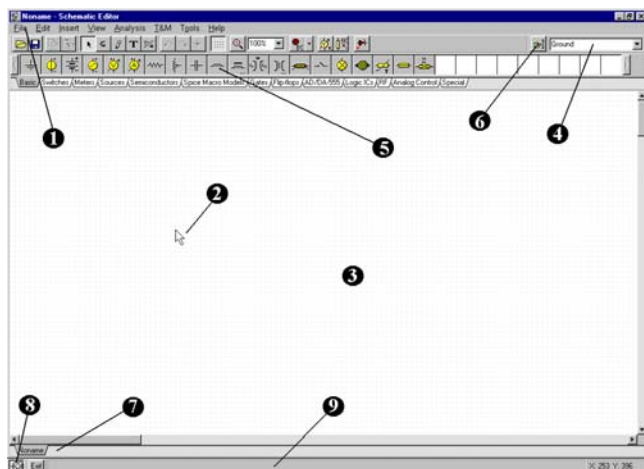
After start-up, the following screen appears on your monitor:

❶ **The Menu bar**

❷ **The Cursor or pointer:** This is used to select commands and to edit schematics. You can move the cursor only with the mouse.

Depending on the mode of operation, the cursor assumes one of the following forms:

An **arrow**, when a command selection is required in the edit window.

A **component symbol** (accompanied by an arrow and small box), when inserting that component onto the circuit in the schematic window. Until the position of the component on the schematic is chosen, its movement is controlled by the mouse.

A **pen**, when defining the endpoint of a wire.

An **elastic line**, when defining the endpoint of a wire or the second node of an input or output.

An **elastic box**, when defining a block after fixing its first corner.

A **dashed line box**, when positioning a component label or a text block.

A **magnifying glass**, when defining a zoom window.

DesignSoft

❸   **The Schematic window:** This shows the circuit schematic currently being edited or analyzed. The schematic window is actually a window onto a larger drawing area. You can move the screen window over the full drawing area using the scroll bars at the right and bottom of the screen. When selecting the New command on the File menu, the system automatically aligns the origin of the editor window with the center of the entire editor drawing area. The same is true when an existing circuit file is loaded, as this is the default window position.

You can think of *TINA's* schematic as existing on several "layers." In addition to the primary layer that holds components, wires, and text, there are two other drawing layers, which you can turn on or off individually. It is generally convenient to have these two layers on.

View|Pin Markers On/Off: Displays/hides component pin ends.

View|Grid On/Off: Displays/hides the grid.

A grid of closely spaced dots covering the entire drawing area may be made visible or invisible in the schematic window, depending on the current state of the *grid button* of the *Grid On/Off* switch on the *View* menu. At some schematic zoom levels, you will not see the dots of the grid; nevertheless, all component pins and connecting wires will be on the grid. These dots represent the only available interconnecting points. Component symbols are positioned on the drawing area horizontally and vertically. These symbols are rigid patterns with predefined pin positions and are handled as single units. This permits the software to unambiguously recognize the network nodes.

❹   **The Tool Bar:** You can select most of the editor commands (e.g., select, zoom, wire etc.,) from this tool bar. Let's summarize the most important commands on the Toolbar. You can find more detailed information in TINA's Help system. Note that most commands on the toolbar can also be found under the drop down menus, and can often be activated by Hotkeys. We show the menu name separated from the command name by a dot (Menu name.Command name).
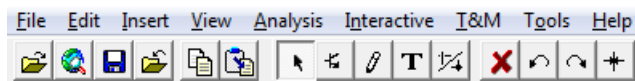
(File.Open) Opens a schematic circuit file (.TSC or .SCH), TINA macro files (.TSM) or Spice netlist files (.CIR).

.TSC extension is the current schematic file extension used in TINA v6 and above. The .SCH extension was also used for schematics in TINA v4 and v5.

.TSM is the extension of TINA macros, which can contain asubcircuit either as a TINA schematic, Spice netlist, or VHDL code.

.CIR files must be circuit files or subcircuits in Spice netlist format. The files will appear in the Netlist Editor, where you can execute most TINA analyses, and edit or complete the netlist.

(File.Open from the web) This command starts the built in web browser that allows you to navigate to any website and then directly open TINA files with TSC, SCH or CIR extension by simply clicking a link. It will also save TSM, LIB, and TLD files into their proper place in the user area. Finally, TINA will recognize the.ZIP file extension and will help you select, copy, and extract files.

By default the built in web browser in TINA opens the TINA circuits on Web page on the www.tina.com website (at the time of writing this manual www.tina.com/English/tina/circuits) where you will find interesting electronic circuit files which you can download or open directly from the web and then simulate with TINA.

(File.Save) Saves the actual circuit or subcircuit into its original storage location. It is wise to frequently save the circuit that you are working on to avoid loss of data in case of a computer crash.

(File.Close) Closes the actual circuit or an open subcircuit on the screen. It is very useful for closing an open subcircuit.

(Edit.Copy) Copy a selected part of the circuit or text to the clipboard.

(Edit.Paste) Paste Clipboard contents into the schematic editor. Note that the content may come from the schematic

editor itself, TINA's diagram window, or any other Windows program.

▶ *Selection mode.* If this button is pressed you can select and drag components with the cursor. To select a component (part), wire or text, just click on it with the cursor. You can also select several objects by holding down the Ctrl key and clicking on the objects one-by-one; or by clicking at one corner of the area, holding down the left mouse button, moving to the opposite corner and then releasing the mouse button. Selected objects will turn red. You can drag the selected objects by dragging one of them. Click and hold the left mouse button when the cursor is over one of the selected objects and move them with the mouse. You can unselect all selected object(s) by clicking on an empty area. One or more selected objects can be deleted while leaving the others still selected by holding the Ctrl key down and left-clicking the mouse.

You can select all the wire segments, connected by the same ID on Jumpers, by holding down the Shift Key and clicking one of the wire segments.

± *(Insert.Last component)* Retrieves the last component inserted, for a new insertion of another copy, with the same parameters as the previous insertion.

∂ *(Insert.Wire)* Use this icon for inserting (adding) wires to the schematic design.

T *(Insert.Text)* Add comments into schematics and analysis results. You can also create **popup texts** which are displayed when the cursor is moved above their title. To create a Popup text in the Text dialog click the "hand symbol" and enable Popup text.

⤢ *(Edit.Hide/Reconnect)* Use Hide/Reconnect to place or remove a connecting dot between crossing wire or wire-component connection.

↶ ↷ *(Edit.Rotate Left (Ctrl L), (Edit.Rotate Right (Ctrl R),* Rotates the selected component.

DesignSoft

✛ *(Edit.Mirror)* Mirrors the selected component.

Hotkeys Ctrl L, Ctrl H

▦ *Switches* On/Off the grid. I.e.makes the grid visible or invisible.

🔍 *Explicitly zoom* in on a selected portion of the current view. It will also zoom out a portion if you just click with the icon into the middle of the area you want to zoom out.
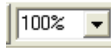
`100% ▼` *Select zoom ratio* from a list from 10% to 200%. You can also select Zoom All which will zoom the effective drawing to full screen.

Interactive modes, see also on the Interactive menu:

  🔵 DC   DC mode

  🔵 AC   AC mode

  🔵 TR   Continuous transient mode

  🔵 TR   Single shot transient mode, the time is the same as set at Analysis Transient

  🔵 DIG   Digital mode

  🔵 VHD   VHDL mode

▼ With this listbox you can select the analysis and the Interactive Mode Options dialog.

🔧 *(Analysis.Optimization Target)* Select Optimization Target to set up Optimization mode or to change settings.

⌷ₖᴷ *(Analysis.Control Object)* Select Control Object for Parameter Stepping or Optimization.

↦ *(Analysis.Faults enabled)* If this button is pressed it enables component Faults, set by Fault property of components. You can set component faults with the Property Editor by double-clicking on components.

🖉 *(View.3D view/2D view)* Hotkey F6. 2D/3D view. If this button is pressed, TINA's Schematic Editor displays circuit components as 3D pictures of the real component assigned to the schematic symbol. This is a simple but useful check before starting the PCB design.

⊣¶ *(Tools.PCB Design)* Invokes the dialog which initiates the PCB design module of TINA, if available.

🔖 Voltage Pin ▾ 🔖 *(Tools.Find component)* Find Component tool. Invokes a component searching and placement tool. This tool helps you find by name any component in the TINA catalog. The search string that you enter will be found wherever it occurs at the beginning, at the end, or anywhere within the component's name. This tool is useful when you don't know where a particular component is located, or if you want a list of all the components that match particular search criteria. A component found in a search can be placed immediately into the schematic by selecting it and pressing the Insert button of this tool.

Resistor ▾ *Component list.* With this tool you can select components from a list.

❺ **The Component bar:** Components are arranged in groups, named by the tabs on the Component bar. Once you have selected a group, the available component symbols appear above the tabs. When you click on the desired component (and release the button), the cursor changes to that component symbol and you can move it anywhere in the drawing area. You can also rotate the component by pressing the + or - keys (on your computer's numeric keypad) or mirror it by pressing the asterisk (*) key (also on your computer's numeric

DesignSoft

keypad). Once you have selected the part's position and orientation, press the left button of the mouse to lock the symbol in place.

❻ **Find component tool:** This tool helps you find by name any component in the TINA catalog. See more details above at the toolbar description.

❼ **Open files tab:** You can have several different circuit files or different parts (macros) of a circuit open in the schematic editor at the same time. Clicking on a tab brings that circuit page up in the editor.

❽ **The *TINA* Task bar:** *TINA's* Task bar appears at the bottom of the screen and provides speed buttons for the various tools or T&M instruments currently in use. Each tool or instrument operates in its own window and can be made active by clicking on its speed button (icon of the tool). Once the cursor is over the speed button, a brief hint appears. Note that the first button (furthest to the left), the Lock schematic button, has a special function. When the Lock schematic button is pressed, the schematic window is locked in place as a background behind other windows, so that it can never cover a diagram or virtual instruments. When the schematic window is not locked and it is currently selected, you will always see the entire schematic window with any other windows hidden behind.

❾ **The Help line:** The Help line, at the bottom of the screen, provides short explanations of items pointed to by the cursor.

4.4 Placing the Circuit Components

Components are selected from the Component bar and their symbols are moved by the mouse to the required position. When you click the left mouse button, the program locks the pins of the component symbol to the nearest grid dots.

Components can be positioned vertically or horizontally and rotated by 90-degree steps in a clockwise direction by pressing the [+] or [Ctrl-R] keys, or in a counterclockwise direction with the [-] or [Ctrl-L] key. In addition, some components (like transistors) can also be mirrored around their vertical axis by using the *[*]* key on the numeric keypad. You can also use the ⟳⟲⤧ buttons or the popup menu (right mouse button) to position components.

After a component symbol has been selected and positioned, you may double click on it to enable a dialog window where you can enter parameter values and a label. When entering numeric values, abbreviations of integral powers from 10^{-12} to 10^{12} can be used. For example, 1k is understood as 1,000.

NOTE:

> *Upper and lower cases must be carefully distinguished (e.g., M = m), and the selected letter must follow the numeric characters without a space (e.g., 1k or 5.1G), or TINA will indicate an error.*

TINA will automatically assign a label for each component you place on the schematic. It will also display the numerical value of the main component parameter (for example: R4 10k). Note that the value is shown only if the Values option of the View menu is checked. For files from the older versions of *TINA*, the Values option is turned off by default. The first part of the label, e.g., R4, is required for symbolic analysis modes. You can also display the units of the capacitors and inductors (for example: C1 3nF) if both the Values and the Units options of the View menu are checked.

4.4.1 Wire

A wire establishes a simple short (zero ohm connection) between two component pins.

To place a wire, move the cursor to the component terminal point where you want to begin. The cursor will change into a drawing pen. You can draw a wire in two different ways:

1) Select the starting point of the wire with a left mouse click, then move the pen with the mouse while TINA draws the wire along the path. While drawing the wire, you can move in any direction and the wire follows. At the end point of the wire, click the left button of the mouse again.

2) Hold down the left mouse button while positioning the pen; release it at the end point.

While drawing a wire, you can delete previous sections by moving backwards on the same track. By pressing the Ctrl key while drawing you can move the last horizontal or vertical section.

You can easily modify existing wires by selecting and dragging sections or edges.

For short wire sections, you may need to hold down the shift key while drawing.

You can also invoke the Wire-drawing tool by the Insert|Wire command (hotkey: [Space]). You can start drawing the wire at any place by clicking the left button of the mouse. When you have completed wiring, use the popup menu, press the right mouse button, or press the Esc key to terminate the wiring mode.

Be sure not to leave any component nodes unconnected. If there are unconnected components or terminals, TINA's Electric Rule Check tool (ERC) will issue a warning (unless you have disabled it).

Wire segments made by the Wire tool are always vertical or horizontal. However, you can add angled wire segments using the components made for bridges, Y and D circuits under the Special component toolbar.

4.4.2 Input and Output

Certain types of analysis (DC Transfer characteristic, Bode diagram, Nyquist diagram, Group delay, Transfer function) cannot be executed until both input and output have been selected. These establish where the excitation is applied and where the circuit response is taken. The output(s) chosen also determine which curve(s) will be displayed in the chosen analysis mode. Sources and generators can be configured as inputs, while meters can be configured as outputs. However, meters can also serve to determine the location of the input quantity that will be used when computing AC Transfer curves and functions. For even greater flexibility, inputs or outputs can be established at nearly any location by using the **Insert|Input** and **Insert|Output** commands. Note that you can define the input parameter for parameter sweeping only through the **Insert|Input** command.

To insert an input or output, select the Input or Output command from the Insert menu and move the input (**I+)** or output (**O+**)

DesignSoft

symbol attached to the cursor over the first schematic node that it should define. Click on that node, release the mouse button, and move the symbol to the second node, and click on that node. The program will draw a dashed rubber line between the two nodes while drawing, and will also place this line on the schematic when you click on the second node.

Since an input reference can be established in so many ways, it is important to remember that only one input at a time can be defined within a circuit.

Similarly, in some of *TINA's* analysis methods (e.g., Symbolic Analysis) only one output can be defined within a circuit.

4.5   Exercises

These exercises will help you build upon and integrate what you've learned from the manual so far.

4.5.1   Editing an RLC Circuit Schematic

Create the circuit diagram of a series RLC network as shown in the following figure.

DesignSoft
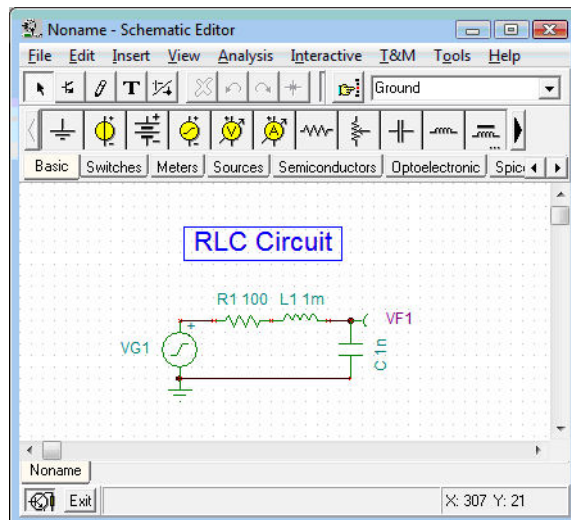
First clear the schematic window with the **File | New** command. The file name in the top line is set to Noname, indicating that a new circuit file is being edited.

Now start adding components. Click on the voltage generator icon, then release the mouse button. The cursor will change into the generator symbol. Position it using the mouse (or by pressing the [+]/[Ctrl-R] or [-]/[Ctrl-L] key for rotation] or the *[*]* key for mirroring) somewhere in the middle of the screen, then press the right mouse button; the schematic editor's popup menu will come up. Select *Properties*. The following dialog box will appear:
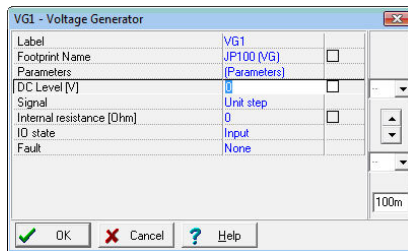
Leave the *DC level* and the *IO state* parameters unchanged. Note that by accepting *Input* for the *IO state* parameter you have selected the output of this generator to be the input for the Bode diagram.

Select the *Signal* menu line and then press the ⬚ button, a new dialog box with the graphics icons of available voltage generator signals will appear. When you select one of them (in this case, click on the Cosinusoidal button), the associated curve comes up with some default parameters. In the case of the Cosine signal, these are:

DesignSoft
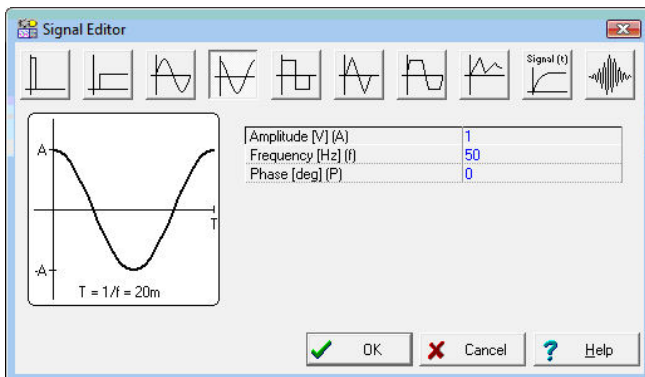
Change the frequency to 200k (200kHz). Click on *OK* and return to the previous dialog box and click on *OK* again. The program will automatically place the label near the component and you will be able to position and place the component and the label together. If the default label position is not satisfactory, you'll be able to drag the label to the desired position later on. When the component is where you want it, press the left mouse button to drop it. This completes the placement of the generator.

Now click on the *Basic* tab on the Component bar and choose the ***Resistor*** icon (your cursor will automatically change when you are over the tabs or the icons). After the symbol of a resistor has appeared in the schematic window, press the right button of the mouse and then select *Properties* from the popup menu.

When the dialog box appears, change the Resistance to 100.

After setting all parameters, click *OK*. Your cursor will turn into the resistor with the frame of the label. Position it as required and press the left mouse button to drop it.

Continue circuit entry with the **L** and **C** components as indicated in the figure above. Set the parameters to *L 1*= 1 m and *C1* = 1 n. Note the default values of the parallel resistive losses for the capacitor and the series resistive losses of the capacitor. Add the Voltage Pin (chosen from the Meters component group) on the upper pin of the capacitor (or you can add a volt- meter in parallel with the capacitor). Note that even though all the computed voltages, currents and signals are available after running an analysis (see below in this chapter and also in the *Post-processing analysis results* section), you still need to define at least one output. Place a ground below the generator and connect the generator and capacitor as shown in the figure. To do this, move the cursor over the appropriate pin node until the small drawing pen appears. When the pen appears, click the left button of the mouse, draw the wire, and left click again at its end point.

Finally, add the title to the schematic. Click the T button and the text editor will appear. Type in: RLC Circuit. Click on the icon and set size 14. The editor also lets you choose another font, style, color etc. Now click on ✔, then position and drop the text on the schematic editor Window.

Before going on, save the circuit with the *File|Save As* command. Name the circuit as **RLC_NEW.TSC** (the *.TSC* extension is added automatically).

Autosave. Note that TINA can automatically save your current schematic at adjustable time intervals. You can set the time interval at the Options dialog (under the View menu):

Autosave interval (0 - no autosave)   5 ⬍ [min]

The default value is 5 minutes. To disable Autosave, set the interval to 0. Note that you must give the circuit file a unique name–certainly not the default name Noname.TSC–in order that the autosave command will be completed.

If you wish, you can still change the circuit in many ways:

- Add new components.
- Delete, copy, or move selected objects using the Edit|Cut, Copy, Paste, and Delete commands.
- Move, rotate, or mirror groups of components. Select the components one by one, holding down the shift key as you click on them. You may also use window selection to identify the group. When you've selected the last component, release the left mouse button, then move the cursor over one of the selected components, press and hold down the left mouse button, and drag the selected parts with the mouse. While dragging, you can use the [+]/[Ctrl-L]/ [Ctrl-R], [-] and [*] keys to rotate and mirror components.
- Move any component label separately by clicking on it and dragging.
- Modify component parameter value(s) and labels of the component by double-clicking on it.

Of course, if you want to keep these changes you must save the circuit again.

4.6   Analyses

TINA has a variety of analysis modes and options:

The analysis method is analog when a circuit contains only analog components; then the components are modeled with their analog models.

The analysis method is mixed when a circuit contains both analog and digital components. TINA will analyze the analog parts in analog, the digital parts in digital, and will automatically create the interfaces among the components. This ensures synchronization and fast convergence.

The analysis method is digital when a circuit contains only digital components; then the components are modeled with their fast digital models.

Analysis Options

This dialog, with which you can set analysis parameters, is on the Analysis menu. Examine the screenshot that follows to see the parameters that you can adjust.

In this manual we will present only the most important options, those which you might want to change. The complete explanation of all the options is included in the on-line help, which you can display pressing the Help button on the dialog.

Note: If you change any settings in this dialog and close the dialog by pressing the OK button, the program will remember the changed settings.

Performance: TINA v9 and later support multi-core processors and run analysis in parallel threads. This results in a significant speed increase. By default the number of threads is equal to the number of cores in your CPU. However you can control this with the Number of threads parameter.

Number of threads: This parameter is set to Max by default and results in one thread per core. You can, however, set it to any number between 1 and Max, or to Dynamic. If you have a quad (4) core machine, it would be good to set this parameter to 3, to leave some computing capacity for other programs and processes. You can also use the Dynamic parameter which controls the number of threads automatically, depending on other processes running on your computer.

Matrix compilation: By default this parameter is enabled and results in compilation of an extremely fast code for some matrix operations. The only reasons to disable this parameter are to perform speed comparisons or to debug software. For maximum performance, it should normally be enabled.

Enable run-time statistics: If this option is set, Tina displays the simulation time of the transient analysis in the status bar and creates a file which contains detailed information on the last transient run. You can load this file by selecting the View.Transient statistics menu item.

Enable instant diagram drawing: If this option is selected TINA will draw the diagram during transient analysis, refreshing the diagram every 1-2 seconds. This is very useful for observing the progress of long calculations.

Save all analysis results: Check this box if you want to save the result of all nodal voltages, resistor, capacitor and inductor voltages and currents, to make the later post processing more convenient. However this option may slow down the analysis by 30-50%.

Disable warning for large site analysis results If you run a transient analysis, a warning will appear if the number of points is greater than 1,000,000 (1M). Select this checkbox if you want to disable this warning.

4.6.1 Analyzing an RLC circuit (DC, AC, Transient and Fourier analysis)

Now execute AC and transient analyses on the RLC circuit you have just entered.

First perform an AC nodal analysis. Select **Analysis|AC analysis|Calculate nodal voltages.** Your cursor will turn into a test probe which you can connect to any node. In a separate window the nodal voltages will be displayed. If you have placed any meters on the schematic, clicking on them with the probe will present detailed information from that instrument. Note that you can acquire DC nodal voltages in a similar fashion through DC Analysis.

Now select **AC Analysis|AC Transfer Characteristic...** from the main menu. The following dialog box appears:
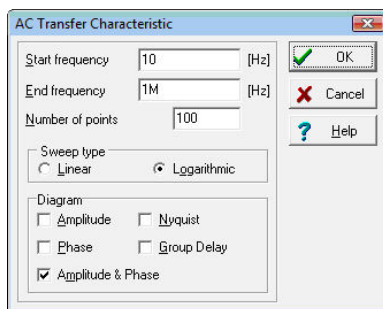
By default Amplitude & Phase will be calculated. Select **Amplitude** and **Nyquist** in addition. Modify the Start frequency to 10k and then press *OK*. A progress bar will appear while the program is calculating. After the calculations are finished, the Bode amplitude characteristic will appear in the Diagram Window. You can easily switch to Nyquist or Amplitude & Phase diagrams by using the Tabs at the bottom of the Diagram Window.

You can read exact input/output values by enabling one or more of the cursors. Note that in any representation you can get and place the formula of the transfer function using Symbolic Analysis and selecting AC Transfer or Semi Symbolic AC Transfer. The formula will appear in the Equation Editor Window and you can place it either on the Diagram or the Schematic window as described above.

GETTING STARTED

DesignSoft
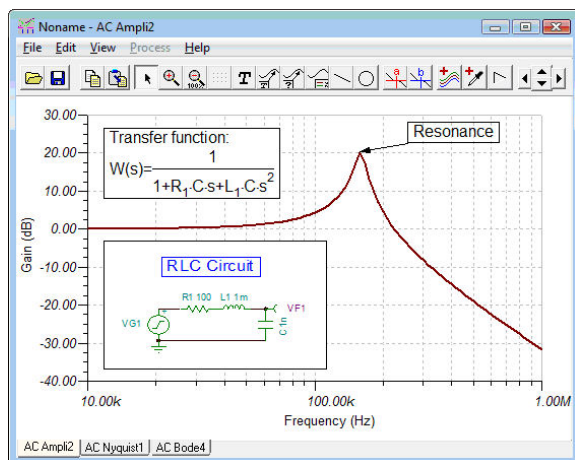
Using *TINA's* graphic facilities, you can add more useful information to your diagram. As an example, let's add markers, a special annotation, and the circuit schematic itself to the diagram.

To add markers to a curve move the cursor over the curve, find a position when the cursor changes into a + shape and click on the curve at this position. When the curve becomes selected, it turns red. Now you can either double-click on it or press the right button of the mouse and select Properties on the popup menu. A dialog box will appear and you can set the curve's parameters: Color, Line width, Marker. Select Marker Type: Square and click on *OK*.

To add some text, click on the ☐T☐ text icon. When the Text editor appears, type in **Resonance**. Note that using the ☐F☐ font icon of the editor you can select any font, style, size and color. Click on OK and place the text in the neighborhood of the resonance peak. Now click on the pointer icon, then on the text, and finally on the peak of the curve. Note that the cursor turns into a + when you are at the right position. You have just entered a line and arrow that will always point from the text to the curve, even if you drag the text into another position or make other changes.

Now place the schematic itself on your diagram. Click the schematic editor Window and select *Edit|Select All*. Copy this selection into the clipboard by selecting *Edit|Copy* or clicking on the Copy icon or using the Ctrl C hotkey. Click on the Diagram Window and use *Edit|Paste*, or click on the Paste icon, or use the Ctrl V hotkey. The frame of the circuit diagram will appear. Position and drop it at the left corner of your diagram. Now you can still modify this picture by dragging or double-clicking on it and changing its size, frame or background.

Now perform a transient analysis. First, make sure your cursor is the selection arrow, then double-click on the voltage generator and change the waveform to the default unit step. After selecting **Analysis | Transient Analysis**, the following dialog box appears:

Transient Analysis
Start display   0   [s]   ✔ OK
End display   1u   [s]   ✖ Cancel
● Calculate operating point   ? Help
○ Use initial conditions
○ Zero initial values
☑ Draw excitation
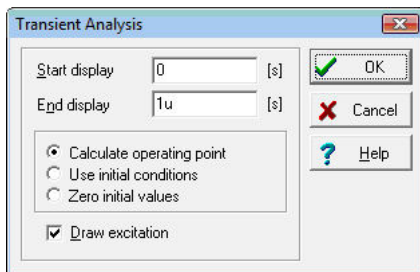
Change the *End Display* parameter to 30 u then press OK. In a separate window the transient response will appear.

As expected, the RLC circuit exhibits a response of damped oscillation. Exact input/output data pairs can be read by enabling the a and/or b graphic cursors.

Now select **Analysis|Symbolic** or **Analysis|Semi-symbolic Transient** from the menu. The closed form expression of the circuit response appears in the Equation Editor window. Click on the Copy icon of the Equation Editor Window, then switch to the Schematic window and select the Paste icon. The frame of the formula will appear. Move the frame to the desired location and press the left mouse button to place the formula. Note that you can reposition it by dragging to any position and you can edit it by double-clicking on it.

Now go back to the equation editor and click on the Interpreter icon (small calculator) on the toolbar. The expression shown in the equation editor is then transferred to the Interpreter window. The actual definition of the time function is at the top of the window, followed by drawing preferences and the 'Draw' command. Press the run button to draw the function in the diagram window on a new page. This curve can then be copied and pasted into the same transient function diagram, where all of the results can be seen simultaneously.
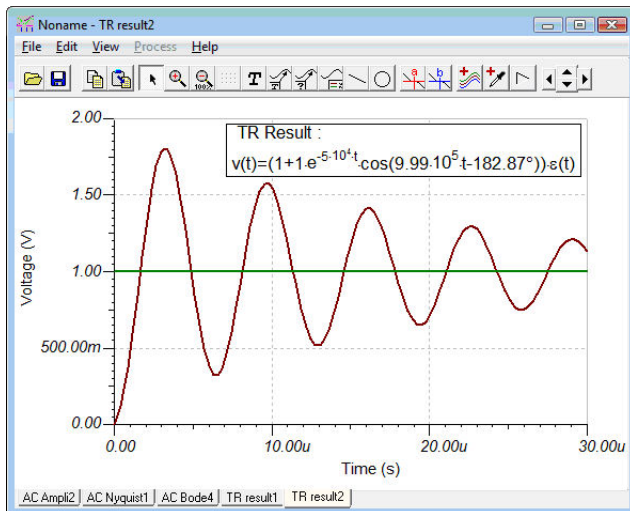
To make it easier to work with curves, you can switch them off or on individually. Use the Show/Hide curves… command under the View menu of the Diagram Window. Before demonstrating this feature, set the checkbox in the Analysis/Analysis Options dialog as shown in the partial screenshot below:
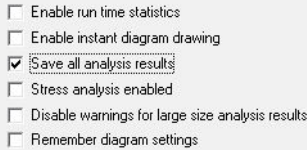
If the check box was not yet set, run the Transient Analysis again. Select the Show/Hide curves… command from the View menu of the Diagram Window. The following dialog box appears:
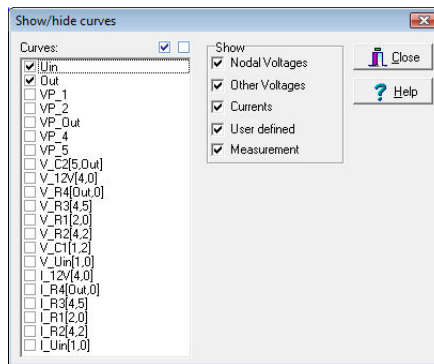
Now you can show or hide any voltages and currents in the circuit using the checkboxes. VP_1, VP_2, VP_3 are nodal voltages of the nodes denoted by numbered yellow boxes **1**, **2**, **3** on the schematic diagram in this working mode of the program.
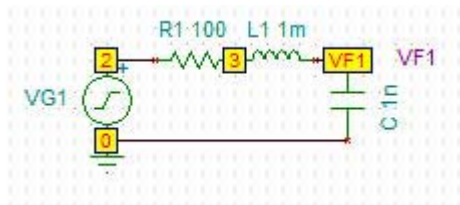
You can also select/deselect component or nodal voltages by moving the cursor and clicking above the selected component or node. Note that at least one of the original outputs (Out and Source) must remain switched on.

Another method to add more curves or process analysis results is to use TINA's Post Processor.

Note; in order to use this feature, you must check "Save all analysis results" in the Analysis.Options dialog.
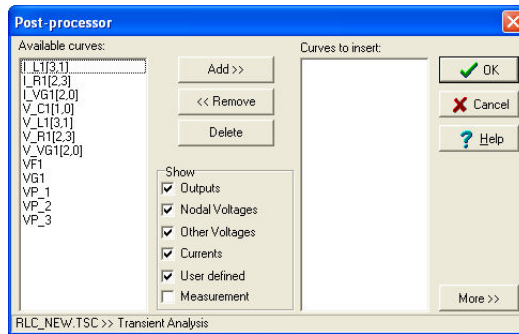
The available curves listbox on the left side shows all the curves that have already been calculated.

The symbols V_label[i,j] and I_label[i,j] denote the voltage and current respectively of the labeled components between the nodes *i* and *j*. The symbol VP_n denotes the nodal voltage of node *n*.

To add the voltage of the coil to the list of curves to insert, select V_L1[3,1] and press the Add>> button. Pressing OK inserts this curve into the current diagram page.
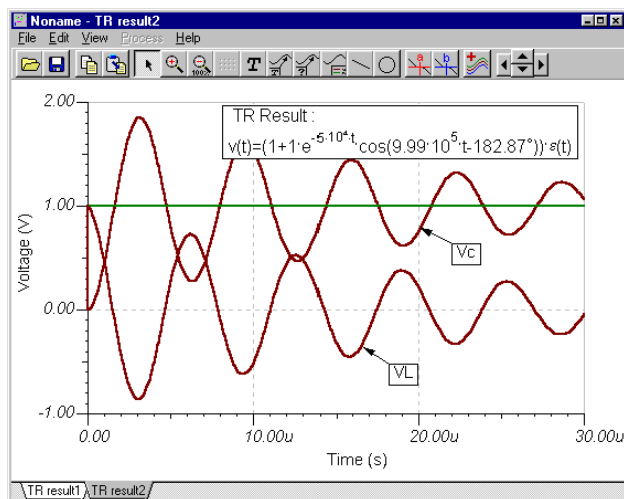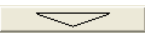
Although you could add a new curve to the diagram, if you run the Transient analysis again the voltage of the previously added coil would not be shown in the diagram. Next add another voltage to the diagram, in a way that the program retains that setting for future calculations and also see how to modify or delete these settings. As an example add the voltage of the RL-part of the circuit to the diagram.

First run the Transient analysis and then open the Post-processor by pressing the ⚡ Post-processor button on the toolbar. The Post-processor window appears. Press the More >> button on the right-bottom side of this window, the extended dialog box of the Post-processor appears.

DesignSoft
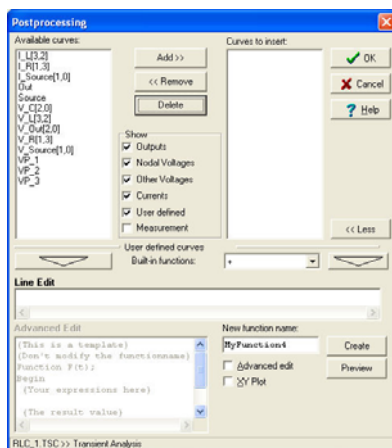
The best way to express the voltage of the RL part is, to express it as a difference of the generator voltage VG1 and the VF1 output voltage: `VG1-VF1`

To create the required voltage by the Post processor click the VG1 voltage on the Available curves list and then press the [⌄] „Add to edit area" button. VG1(t) appears in the Edit line of the post processor. Now enter the – operand and click the VF1 voltage on the curve list and press the [⌄] button again.

Note that although you could do the same by using the VP_1, VP_2, VP_3 these may change if you edit your circuit. Therefore it is better to use node outputs, voltage pins etc. as these are more independent of circuit changes.

The extended part of the postprocessor dialog looks like this.
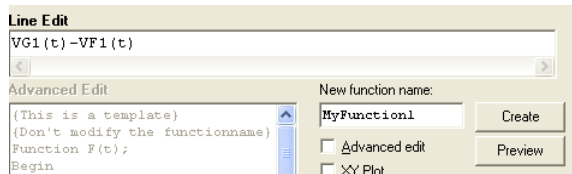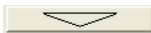
Change MyFunction1 to VRL then press the **Create** button. The VRL identifier will appear on the Curves to Insert list.
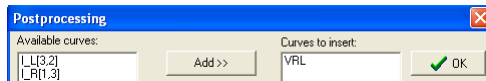
Press OK and the VRL voltage appears in the diagram Window.

DesignSoft
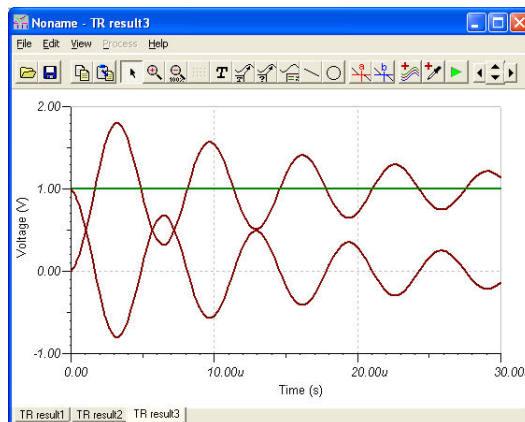
Now let's see how to modify the additional new curves. Press the ![icon] button again, the Post-processor window will appear, with the new VRL curve at the end of the 'Available curves" list. Click on VRL and press the [⌣] button, the definition of the curve will appear in the Line Edit field. However the New function name will be different, for example Myfunction3. Let's change this back to VRL. and change the VG1(t)-VF1(t) expression to something else, e.g. 2*(VG1(t)-VF1(t)). Press Create and OK and the new curve will appear in the diagram.

If you want to delete the new curve open the post processor again click on VRL and press the Delete button.

Note that if you save and reload the circuits with new curves added by the Post-processor, the new curve(s) will automatically be generated as long as they are present on the Post processor's "Available curves" list, and you can either edit these expressions or delete them in the Post-processor.
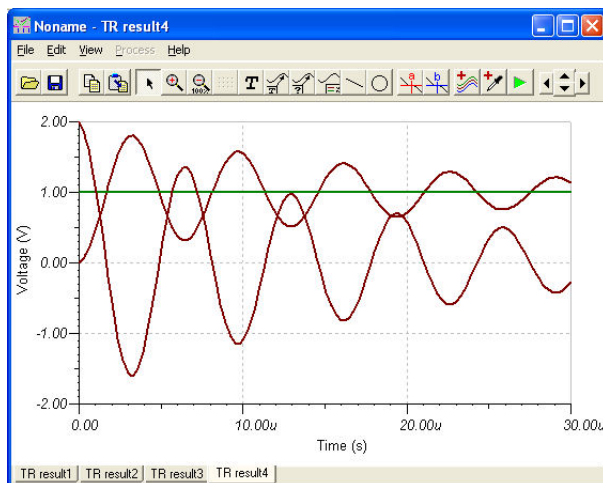
You can do a lot more with TINA's post-processing tool. For example, you can create curves of new functions created by adding or subtracting curves, or by applying mathematical functions on them. For a more detailed description, refer to the Post-processing analysis results section in the Advanced Topics chapter.

To demonstrate a more advanced feature of TINA, examine the **Fourier Spectrum** of the non-periodic transient response just obtained. First, in order to get a finer curve, select Analysis | Set Analysis Parameters… and change the "TR maximum time step" parameter to 10n. Next run the Transient Analysis for a longer 1ms time by changing the End display parameter in the Transient Analysis dialog to 1m.
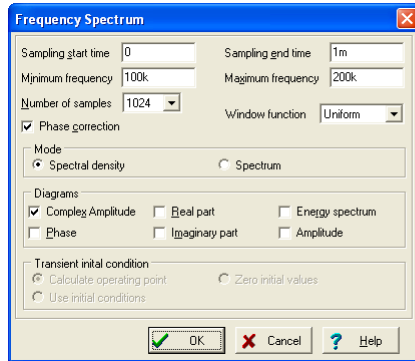
Select the damping output signal by moving the cursor over the curve and pressing the left mouse button when the cursor assumes a + form. The selected curve will change to red. Now press the right mouse button and select Fourier Spectrum from the popup menu. The Frequency Spectrum dialog box will appear. Set the Minimum frequency to 100k, the Maximum frequency to 200k and press OK.

The Fourier spectrum of the transient response will appear. The result is a continuous frequency spectrum shown in the figure below. As expected, the frequency where the Fourier Spectrum shows a maximum is the resonant frequency of the circuit.

The Fourier spectrum and the Fourier series dialog box can also be obtained directly from the Analysis.Fourier Analysis menu. This way you do not need to calculate the transient function manually TINA will automatically do it before generating the Fourier series or spectrum.
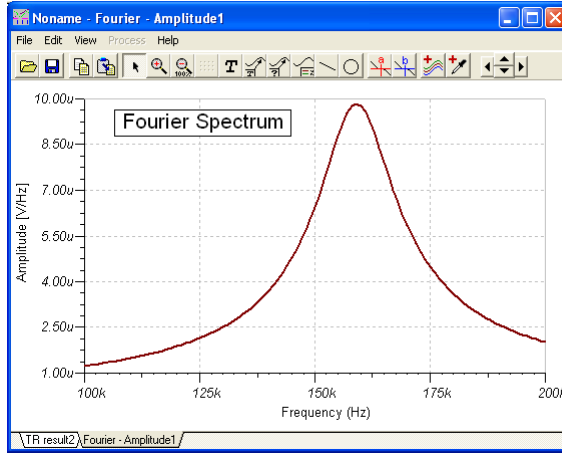
You might be surprised that the unit of the frequency spectrum is in V/Hz. That is because the continuous Fourier spectrum is a density function versus frequency. If you want to know the approximate amplitude in a narrow frequency band, you should multiply the average amplitude (given in V/Hz or Vs) with the bandwidth (given in Hz or 1/s).

You can also find the Amplitude in V directly, if you select Spectrum in the Mode field of the Fourier Analysis dialog. In this case the applied bandwidth is 1/DT, where DT is the length of the Transient analysis (End display - Start display). This feature is especially useful if your signal contains both non-periodic and periodic components. If your signal contains periodic components, you can display them in the diagram more accurately if you select a suitable *Window function* in the Frequency Spectrum dialog. For reading the amplitude from the diagram it is the best to use the Flattop window function.

Fortunately, Fourier analysis is not so complicated for clearly periodic signals. Periodic signals can be represented by Fourier Series or in other words as a sum of cosine and sine waves at the base (fundamental) frequency and integer multiples of the base frequency. To try out this kind of Fourier analysis in *TINA*, load **AMPLI.TSC** from the *EXAMPLES* folder.
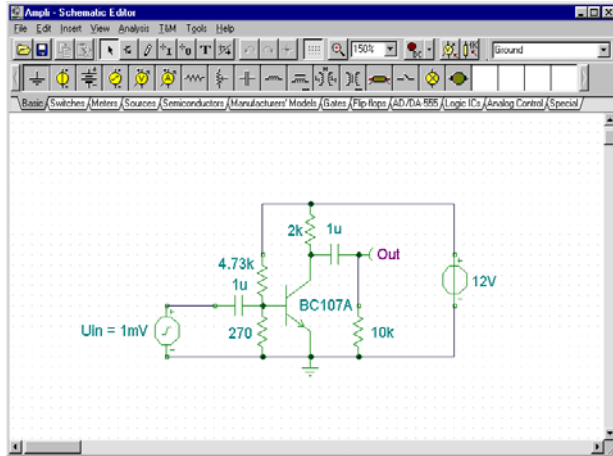
Run a transient analysis and then select the output curve with the largest amplitude. Press the mouse right button and select **Fourier Series** from the popup menu: the dialog box of the Fourier series will appear. Note that you can access this dialog directly from the Analysis. Fourier Analysis menu. Set Sampling start time to 1ms and the Number of samples to 2048. Note that for best accuracy, it is very important to set the starting time for the Fourier Series analysis to after the initial transient has died away. Now press *Calculate*. The list of Fourier components will appear.
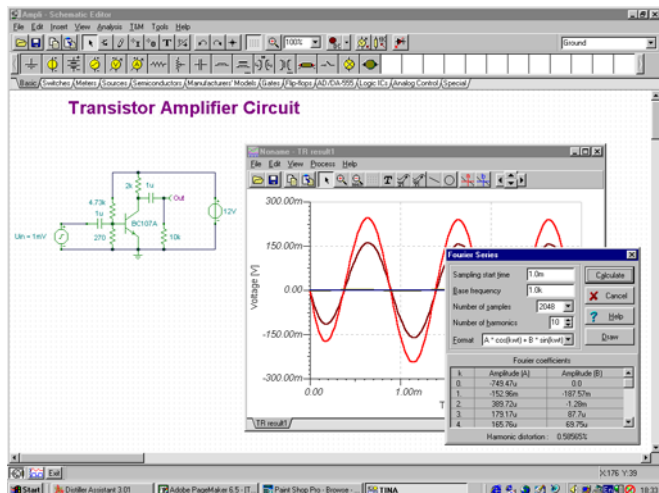
If you press *Draw* you can also draw a diagram showing the amplitudes in V (volts) at integer multiples of the base frequency.

4.6.2   Creating and analyzing an Op-Amp circuit

Create the circuit diagram using an OPA121E operational amplifier from Texas Instruments as shown in the following figure:
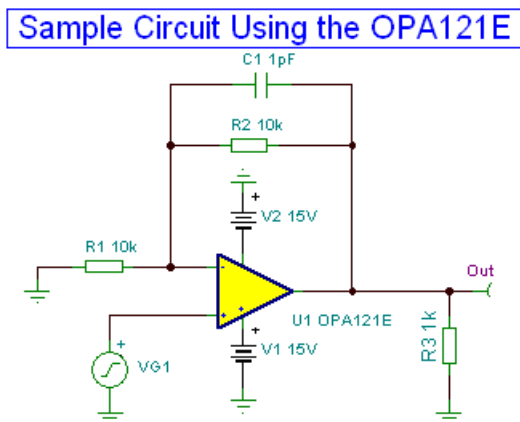
If you just opened TINA and wish to create your own circuit, you can start adding components right away. The circuit file name in the top line is set to Noname by default, indicating that a new circuit file Noname.TSC is being edited. If you already have a circuit loaded in the editor, for example, our previous RLC circuit, you can start a new one with the File|New command. You can switch between multiple circuits by clicking the Tabs at the bottom of the screen.

Now start adding components. Left click on the voltage generator icon then release the mouse button. The cursor will change into the generator symbol. Position it using the mouse (or by pressing the [+]/[Ctrl-R] or [-]/[Ctrl-L] key for rotation] or the [*] key for mirroring) somewhere in the middle of the screen, then press the left mouse button to drop the component into the schematic. We still need to set the properties of this generator. Double-click on the generator and the following dialog box will appear:
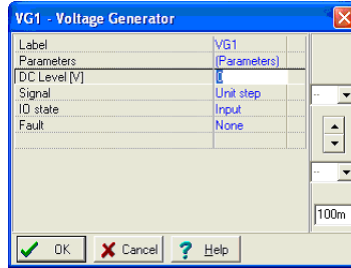
Leave the DC level, and the IO state parameters unchanged. Note that by accepting Input for the IO state parameter you have selected the output of this generator to be the input for this analysis (a Bode diagram in this example). Click on the Signal menu line. The dialog box will change as shown below:
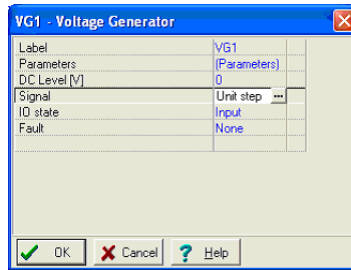
Press the button. A new dialog box with the graphic icons of available voltage generator signals will appear. When you select one of them (in our case, click on the Square Wave button), the associated curve comes up with some default parameters. In the case of the Square Wave signal, these are:

DesignSoft
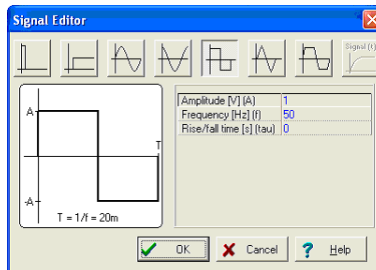
Change the Amplitude to 500m (this represents 500mV peak), the frequency to 100k (100kHz), and the Rise/Fall time to 1p (1ps). Click on OK and return to the previous dialog box and click on OK again. The program will automatically place the label (VG1) near the component and you will be able to position and place the component and the label together. If the default label position is not satisfactory, you'll be able to drag the label to the desired position later on.

Now click on the Spice Macros tab and press the left Operational Amplifiers button. The following dialog box will appear:
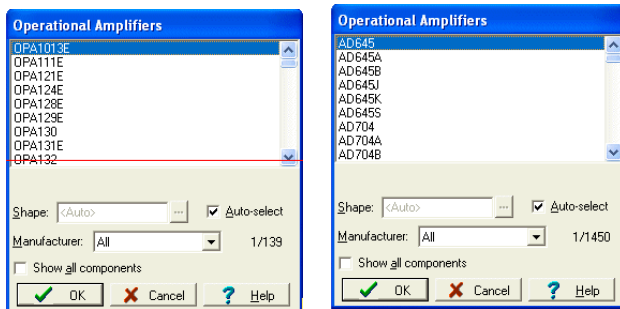
To find the IC we want, scroll down the list until you find OPA121E. You can narrow the list if you select the manufacturer (Texas Instrument in our example) from the Manufacturer listbox. You can also simply type OPA121E and the list will automatically jump to the IC (press the Delete button on the keyboard and try again if you make a typing error). Click on the line (OPA121E) and press the OK button. (Alternatively you can double-click on that line). The schematic symbol of this opamp will appear and be attached to the cursor. By moving the mouse, position the opamp as shown on the schematic at the beginning of the section and then press the left mouse button to place the opamp into your schematic.

You can also select a part using the Find Component tool at the top-right corner of the Schematic Editor. If you type the part number into the "Component to find" field and press the Search button, the list of available component(s) will appear. (You can enter just part of the name if you are not sure of the entire name).

Press the Insert button to place the component. With the List Component button you can create the list of all available components in a textfile.
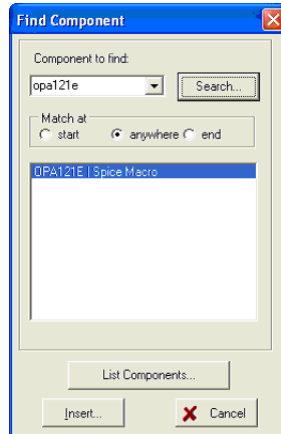
Note that other types of ICs are available under the buttons next to the Operational Amplifiers: Difference Amplifiers, Fully-Differential Amplifiers, Comparators, Voltage Regulators, Buffers, Current Shunt Monitors, and Other Components). You can bring all of these various components into the dialog box for any of the buttons if you set the Show All Components checkbox. In addition to selecting an IC on the list, you can also find it by clicking on any item on the list and then typing in the name of the IC.

Now click on the Basic tab on the Component bar and click the Resistor icon. The resistor symbol will be attached to the cursor. Move the resistor to the position of the R1 resistor on the sample schematic diagram at the beginning of this section and press the left mouse button to place this resistor into the schematic. Double-click on the resistor and the following dialog box will appear:
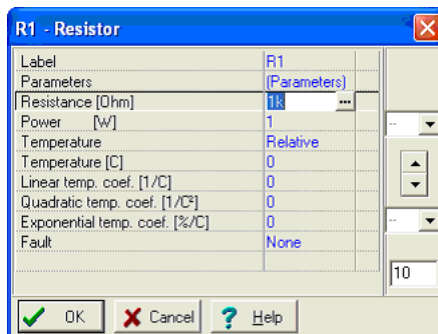
Change the value in the Resistance field to 10k, and press OK.

Note that you can set a component value before placement while you are moving it. To do this, press the right button of the mouse and select Properties on the popup menu. The dialog shown above will appear and you can set the properties of that component. After pressing OK, you can return to component placement.

Now let's place R2 at the top of the circuit. Click on the resistor symbol on the component toolbar, move and place the resistor. When you drop the resistor you will see that its value is already 10k since the program remembers the previous value.

Now let's place R3, which needs to be turned by 90 degrees. Click on the resistor symbol on the toolbar and turn the component by 90ş by clicking the ⟲ or ⟳ buttons or pushing the Ctrl L or Ctrl R keys. (+ and – on the numeric keyboard have the same effect). Place the component on the right side of the screen and set its value to 1k.

Continue circuit entry with the Capacitor, Battery, and Ground components as indicated in the figure above. Set the parameters to C =1 p, V1=15, and V2=15. Place a Voltage Pin Out (chosen from the Meters component group) at the right side of the new schematic. Pay attention to battery polarities and rotate the symbols if necessary.

NOTE:

> *If you have several opamps you may want to simplify their connection to the power supply. This can be done using the ⊣ Jumper component, which you can find at the first place on the Special toolbar.*

All jumpers with the same label are considered as electrically connected components in TINA. Therefore, if you connect a jumper called VCC with the positive power supply of the opamp, it is enough to connect jumpers with the same VCC label to the positive power supply pins of the opamps. As an example, you can load and study OPAMP2.TSC from TINA's EXAMPLES\PCB folder, also shown below.
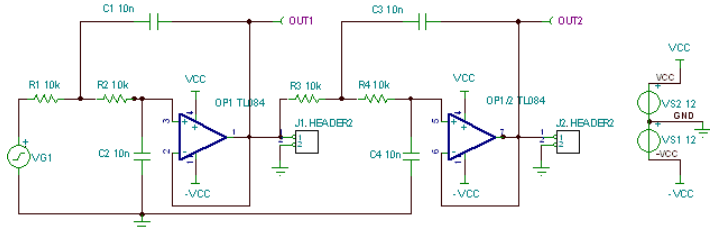
Note that even though all the computed voltages, currents and signals are available after running an analysis (see below in this chapter and also in the Post-processing analysis results section), you still need to define at least one output. We have placed the parts into the schematic but they are still unconnected. To connect devices, move the cursor over an appropriate pin node until a small drawing pen icon appears. When this pen appears, click the left button of the mouse, draw the wire, and left click again at its endpoint.

Finally, add the title to the schematic. using the $\mathbf{T}$ icon on the toolbar.

Let's check the circuit we have just built and run ERC from the Analysis menu. If everything is OK, the following dialog will appear:
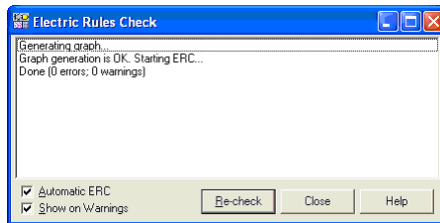
If there is a problem with the circuit, a list of warnings or error messages will appear in the dialog box. If you click on a warning or error message, the related part or wire will be highlighted in the circuit diagram.

4.6.2.1 Calculating DC Transfer characteristic

We have already seen several of TINA's analysis modes. But so far we have not used the DC analysis mode to calculate the DC transfer characteristic of this circuit. Select DC Analysis|DC Transfer Characteristic… from the Analysis menu. The following dialog box will appear:

DesignSoft
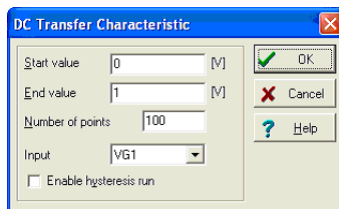
Set the Start value to –7.5, the End value to 7.5, and then press OK. After a short running time, a Diagram Window will appear as shown below. This displays the circuit's transfer curve-output voltage vs. input voltage.
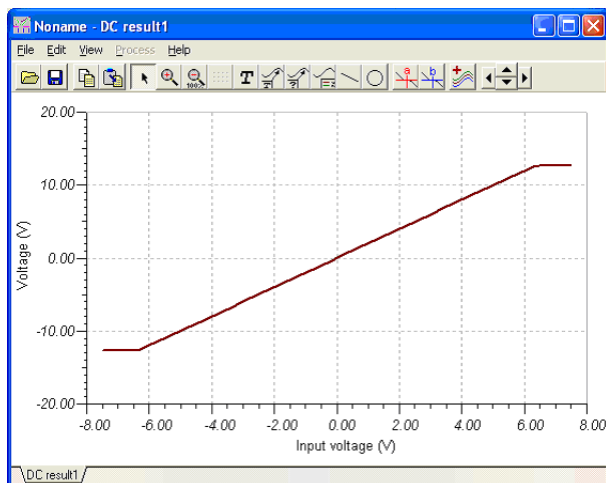
4.6.3   Analysis of SMPS circuits

SMPS or Switching-Mode Power Supply circuits are an important part of modern electronics. The heavy transient analysis needed to simulate such a circuit may take a lot of time and computer storage. In order to support the analysis of such circuits TINA provides powerful tools and analysis modes. In this Chapter we will demonstrate these through examples.

DesignSoft

Using the Steady State Solver

The most time consuming part of an analysis of an SMPS circuit is to reach its steady state, when the DC level of the output voltage does not change and the output waveform has only a small periodic ripple.

To find this state automatically, TINA has a Steady State Solver under the Analysis menu.

To demonstrate this tool, let's load the Startup Transient TPS61000.TSC Boost Converter circuit file from the EXAMPLES\SMPS\QS Manual Circuits folder.
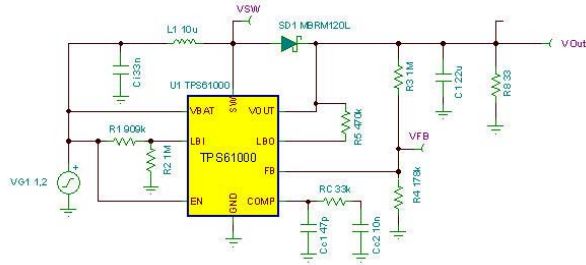
Select the Steady State Solver from the Analysis menu. The following dialog will appear:

DesignSoft
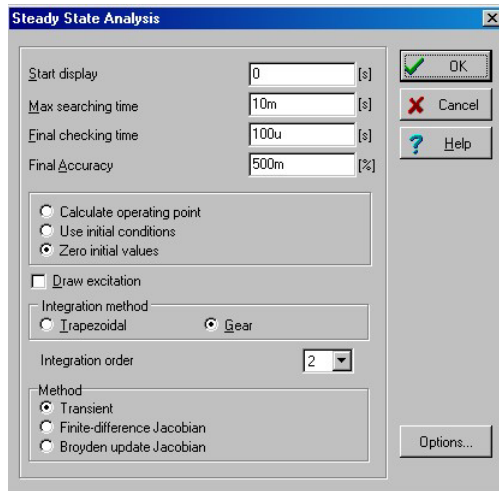
The new parameters compared to the Transient Analysis dialog box areas follows -

Max searching time: The solver will try to find the steady state solution for max 10ms. After this, the analysis will discontinued whether or not a solution was found..

Final checking time: After the steady state search is done, there is a final check for the length specified here. You should have a stationary waveform for this time interval.

Final accuracy: the maximum DC level change allowed. When the change is below this, the analysis will end. Note that the 500m in the example above means 0.1%

Method: You can select the method used for finding the steady state:

Transient: The steady state is searched by using transient analysis. *Finite-Difference Jacobian, Broyden update Jacobian:* The steady state is searched by the methods described in the paper, *Automated steady-state analysis of switching power converters* by Dragan Maksimovic.

Note that these last two methods may get to the steady state faster, but they do not go through the normal transient states, so the resulting waveform between the initial and final state does not reflect the real process (but rather the mathematical path of the methods to get there).

Now lets run the Solver. After a few minutes of running (approximately 2 minutes on a 2GHz Pentium computer) we'll get the following resulting waveform:
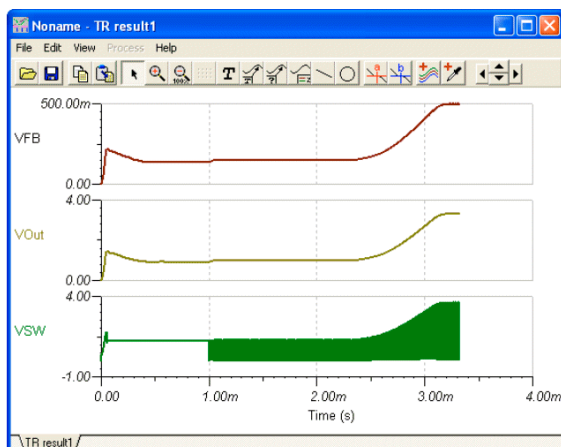
These waveforms show the detailed transient from switching on until reaching the steady output voltage. If you zoom out on the waveform, you can see that the period of the switching is around 500kHz and the time needed to arrive at the steady state is 4 milliseconds. Therefore, we need to calculate at least hundreds or sometimes thousands of periods if we want to see the whole transient waveform. This is why finding the steady state is such a time consuming process. The reason for this problem is the long start-up time of SMPS circuit compared with their switching frequency. The start-up time is basically determined by the filter capacitors on the output. The larger these capacitors are, the longer the start-up time.

NOTE:

In some cases you can accelerate the Steady State Search using the Finite-Difference Jacobian and Broyden update Jacobian methods, however they do not always converge and the intermediate waveforms provided by these methods do not reflect the real waveforms of the transient process.

Trigger

Use this to determine the starting and ending times of the switching period.

You can find this component on the Meters toolbar of TINA. You should connect it to the oscillator frequency control pin of the SMPS/PWM controller IC, but any node where the oscillator waveform of the IC is present will do.

If you double-click on the Trigger component you can set its parameters.

DesignSoft
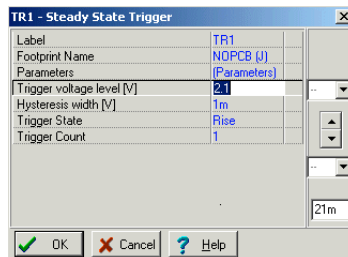
Trigger voltage level: the threshold voltage for the trigger event

Hysteresis width: hysteresis value for the trigger event. This value defines a region within which the trigger voltage is allowed to

oscillate without generating a trigger event.

Trigger State: Rise/Fall The direction of the voltage change required for a trigger event

Trigger Count: you can take several periods for the waveform analysis.

Once you have checked the initial transient and the steady state waveform and SMPS circuit the next thing you normally want to know is how it behaves when the input voltage or the load changes. This is realized by the **Input step** and the **Load step** analyses**.**

Sensor

The purpose of this component is to set the target voltage(s) to be watched during the steady state search. You can add more than one sensor to a circuit. By adding sensors, you can significantly accelerate the steady state search. You can make the search even faster if you can give the final voltage at a certain node.

Using the "Max. no. of saved TR. points" parameter in the Analysis/ Analysis parameters dialog, you can limit the maximum number of points placed in the Diagram. This is useful for large analyses to accelerate diagram drawing. By increasing this parameter, you can refine the diagrams but the drawing time will be greater.
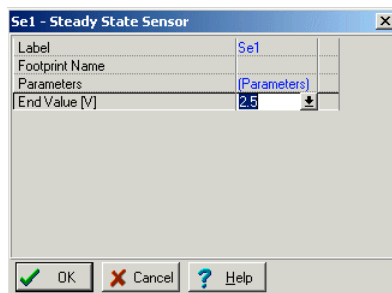
The only electric parameter of the Sensor component is -

End Value: Voltage|Not Used

Accelerating SMPS simulation using initial values

As we mentioned in the previous section, the long analysis time needed for reaching the steady state of SMPS circuits is mostly used for charging the output filter and some capacitors. If we start the analysis using initial values for larger capacitors and inductors, the analysis time can be significantly reduced. In TINA, the Steady State Solver will automatically place initial values into the model of larger capacitors and inductors and so the following Transient Analysis can be run significantly faster (assuming that we do not make changes which will need significantly different initial values). For example, if you want to study the effect of changing the output filter capacitor, it will not significantly change the output voltage DC level. Therefore, starting the new analysis with an initial value calculated by the steady state solver for another output capacitor, will result in a much faster analysis. You can accelerate the analysis of input and load changes in the same way.

To demonstrate this feature, let's run a transient analysis for our example. Selecting the Transient command from the Analysis menu, the following dialog box appears.
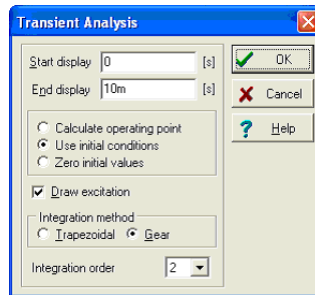
Note that Use initial conditions is set in the dialog. Press OK to start Transient analysis. You should see that analysis runs very fast compared to the previous steady state analysis. The output waveform is shown in the picture below.
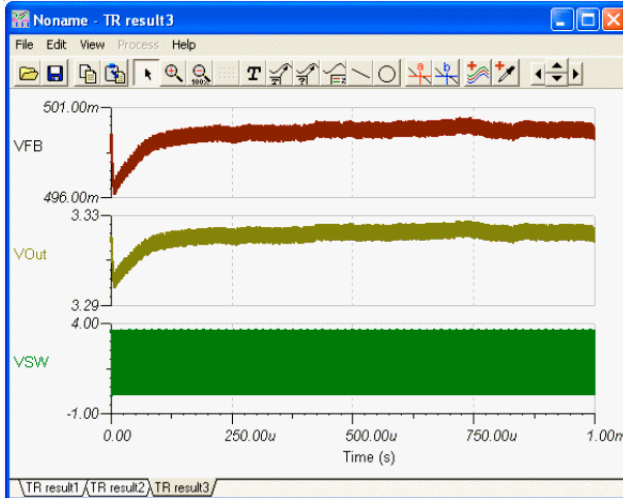
Why did the analysis run faster? Transient analysis was already preceded by the Steady State Analysis and the main capacitor's initial values (called Initial DC voltage in the Capacitor property box) was already set to the final DC voltage. For example, if you double-click on the C1 capacitor, you will see that the Initial DC voltage is already set to 3.31 V. Similarly, all the larger capacitor's initial values are set.

Once you have checked the initial transient, the steady state waveform, and SMPS circuit, the next thing you normally want to know is how it behaves when the input voltage or the load changes. This is realized by the **Input step** and the **Load step** analyses**.**

DesignSoft

NOTE:

GETTING STARTED
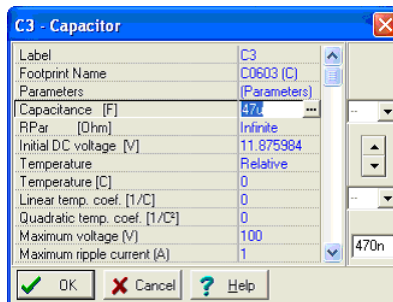
Input step analysis

One of the standard analyses for SMPS circuits is the calculation of the response to an input change to test the capability of the SMPS design to regulate the output with step changes in the input line. This can be accomplished by adding a pulse to the input voltage and checking the output and other voltages. Since the input change is relative to the steady state, we can start it from the steady state initial values calculated by TINA's steady state solver.

Load the Input Step Transient TPS61000.TSC Boost Converter circuit file from the EXAMPLES\SMPS\QS Manual Circuits folder. The schematic design is the same as above.

To see the input step waveform, double-click on the VG1 voltagegenerator on the left. The following dialog box will appear:

VG1 - Voltage Generator	
Label	VG1
Footprint Name	JP100 (VG)
Parameters	(Parameters)
DC Level [V]	5
Signal	waveform ...
Internal resistance [Ohm]	0
IO state	Input
Fault	None

✓ OK   ✗ Cancel   ? Help
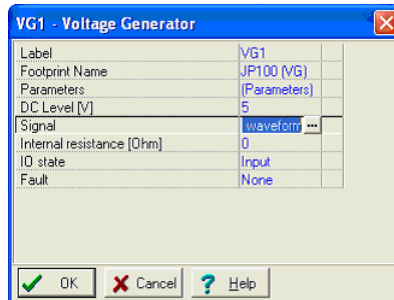
According this, the input voltage is 1.2V. This is converted by the SMPS circuit to 3.3V.

Now click on the Signal line of the above dialog and then the ... button. The following signal in the Signal Editor will appear:

DesignSoft
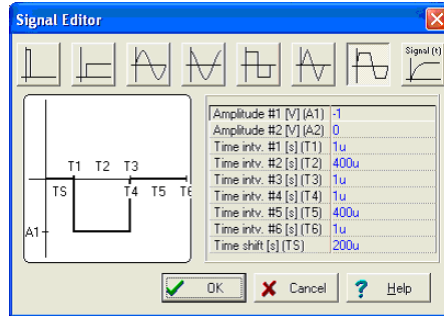
According to the waveforms, the input voltage will decrease from 1.2V to 0.8V for a T2=1ms time; and the starting edge (T1) and the ending edge (T3) of the pulse are 10us.

To see the response of the circuit, let's invoke and run the Transient analysis from the Analysis menu.
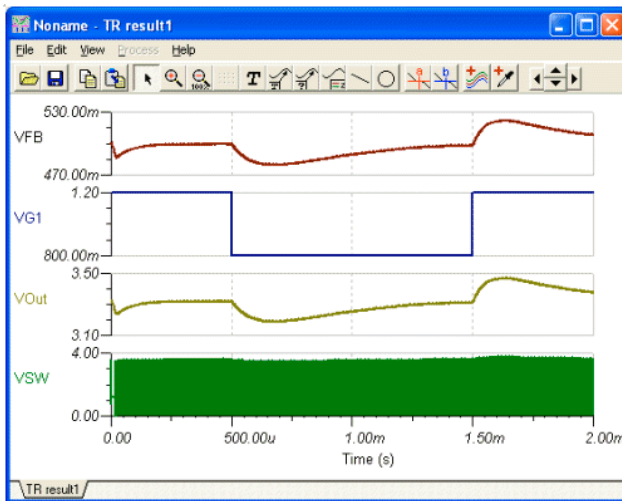
Load step analysis

Another standard analysis is to determine the SMPS response to a fast load change. Using simulation, the response to load changes is obtained by adding a current pulse to the load and analyzing the output and other voltages. Since the load change is relative to the steady state we can start it from the steady state initial values calculated by TINA's steady state solver.

Now load the example Load Step Transient TPS61000.TSC. The schematic design is the same as above , except for the ILoad Current generator on the output.

If you double-click on the ILoad generator and check its waveform, you will see that the DC part is
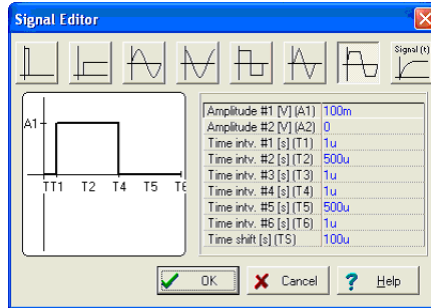
5mA while the pulse is 45mA in amplitude and 500us in width. Accordingly, the 5mA load current will rise to 50mA and then decrease to 5mA again.

Let's run Transient from the Analysis menu and see the result.
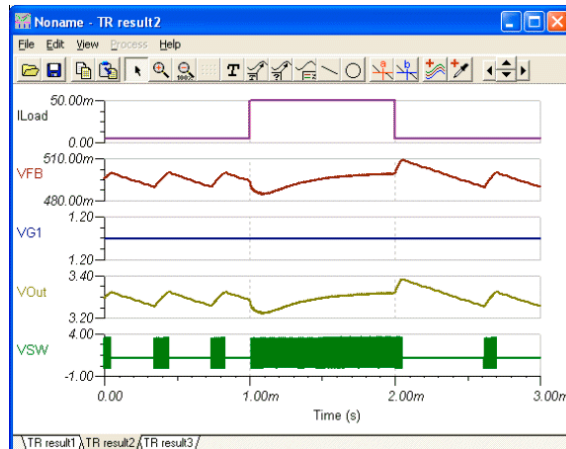
AC analysis

For AC analysis and stability analyses you can use the so called Average models provided in TINA. The average models represent a method, based on averaging the effects during the switching process. The resulting equations are linear therefore the method is extremely fast in order to draw Bode and Nyquist plots needed for stability analysis. Note that for using the AC analysis function of TINA you need an average model, the transient models are not applicable and will give improper results.

To demonstrate this tool, let's load the Average model TPS61000.TSC circuit file from the EXAMPLES\SMPS\QS Manual Circuits folder.
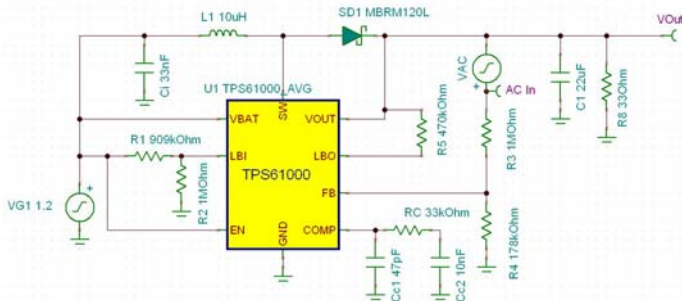
Note the VAC generator which is providing signal for the AC analysis., and the AC In Voltage pin which is the Input of the AC analysis (its IO state parameter is set to Input).

Let's run AC Analysis/AC Transfer Characteristic… from the Analysis menu and see the result.
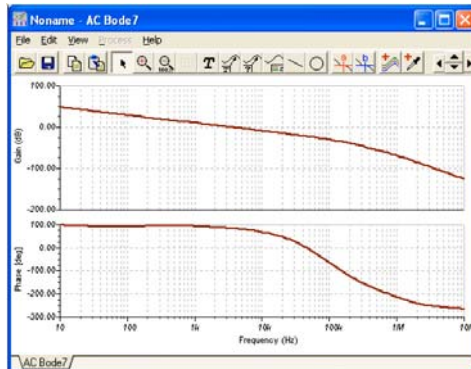
4.6.4 Power dissipation and efficiency calculations

Efficiency and dissipation are important factors in electronic circuit design, especially in the case of AC-DC chargers, AC-DC and DC-DC converters, power supplies, amplifiers, and other applications. TINA has an advanced tool to calculate input and output power, dissipated power by the components, and hence calculate the efficiency of the circuit and explain the reasons leading to current efficiency.

You can enable the power dissipation tool in the Analysis menu by clicking the "Power dissipation analysis enabled" line or the checkbox in the Analysis Options dialog box.

When the tool is enabled, after Transient analysis, a Power dissipation report table is displayed showing the Efficiency, Total input power (Source), Total power on the load (Sink), and dissipated power on other selected circuit components (Loss).

Before the Transient analysis, the appropriate power types should be assigned to the components participating in the calculation using the

 symbol on the command toolbar, shown below.

The last Pass/Fail column of the table shows if the power that dissipated on the components is below (Pass) or above (Fail) the maximum allowed power of the related component, specified in the component property window. If 0 or no power is provided for a component, then **No data** is displayed.

Power dissipation report

Efficiency: 85,68% Total input: 390,86m W Total load: 334,89m W

Component ▲	Power type	Power dissipation (W)	Percentage (%)	Pass/Fail
VG1	Source	390,86m	100	Pass
R8	Sink	334,89m	85,68	Pass
SD1	Loss	29,61m	7,58	Pass
R4	Loss	1,4u	0	Pass
R3	Loss	7,98u	0,00	Pass
R2	Loss	387,72n	0	Pass
R1	Loss	366,68n	0	Pass
RC	Loss	5,77p	0	Pass
L1	Loss	9,13m	2,34	Pass
R5	Loss	4,64n	0	Pass
U1	Loss	21,33m	5,46	Pass

☐ Show all/selected components   Export...

🔲 Close   ? Help

You can study the tool through the following example.

Open the TLV70218_TRANS.TSC circuit from the Examples\Texas Instruments\LDO\TLV70218 folder of TINA. The following circuit is displayed in the schematic editor of TINA.
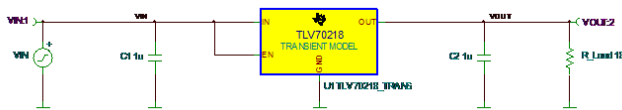
Run Transient . . . from the Analysis menu. The following diagram will appear:
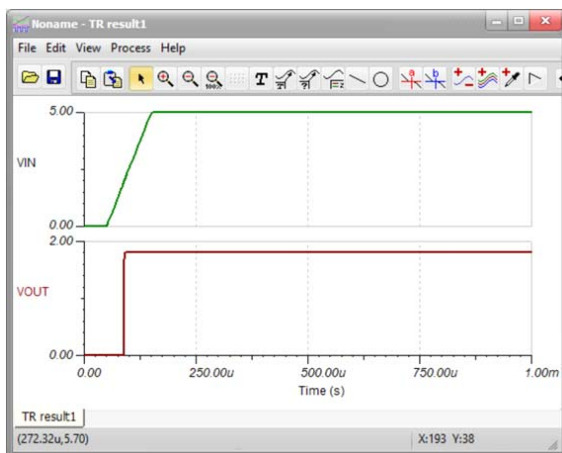
In this example, VIN linearly rises to 5 V. When it reaches approximately 2 V, the output voltage reaches 1.8 V and stays at that level.

Now set VN as Source, the TLV70218 IC as loss, and R_load as sink using the 🐞 button.
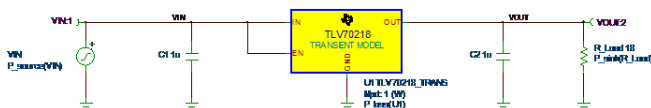
Enable the "Power dissipation analysis enabled" option in the Analysis menu.
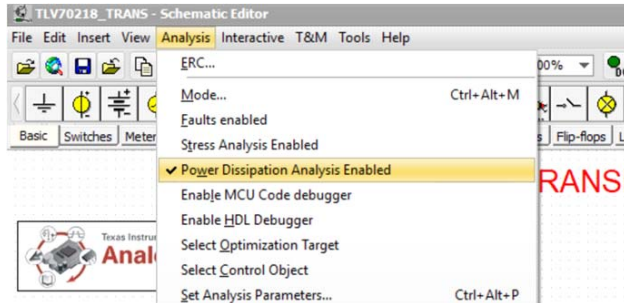
Now run Transient . . . from the Analysis menu setting. Set Start display to 500us to make power calculation after the output voltage reaches 1.8 V.
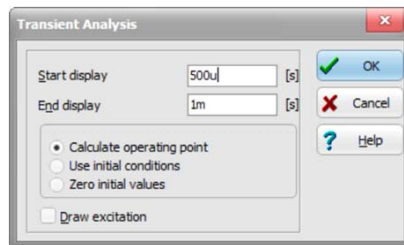
The following table will appear:

The efficiency of the circuit is not good; it is approximately 36% only. This is because the TLV70218 is a linear regulator, and approximately 64% of the total power is dissipated in the regulator IC.

GETTING STARTED

DesignSoft

4.6.5   Stress Analysis

Stress Analysis can check parts for stress conditions such as maximum power dissipation and maximum voltage and current limits. You can set these parameters in the property window of the parts or in the catalog. This kind of analysis is also called Smoke analysis, because overloaded parts often emit smoke.

You can enable Stress Analysis by setting the Stress Analysis Enabled checkbox in the Analysis | Option dialog or on the Analysis menu.

When running DC or Transient Analysis from the Analysis menu, a list of components will appear, along with the parameters exceeding maximum limits.

If you click a component in the list, the corresponding component on the schematic diagram will be selected and turned red.

The maximum values of the components can be set in the component property dialogs or in the component catalog parameter dialogs. Both can be entered by double-clicking on the components. Before running an analysis, check and set the maximum values of the components in your circuit.

As an example of Stress Analysis, open the file Stress Analysis.TSC from TINA's EXAMPLES folder and run DC.Calculate Nodal Voltages and Transient Analysis from the Analysis menu or the corresponding interactive modes. In the following figure, you can see the result of Stress Analysis in DV interactive mode.
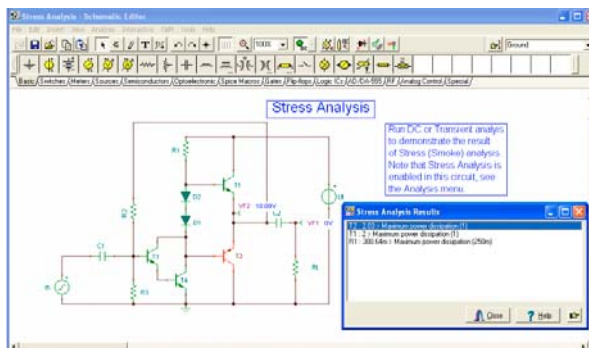
Apparently the power dissipation of T1, T2 and R1 exceed maximum limits allowed for these parts.

4.6.6   Network Analysis

TINA helps you perform network analysis and determine the two-port parameters of networks (S, Z, Y, H). This is especially useful if you work with RF circuits. Results can be displayed in Smith, Polar, or other diagrams. You can assign the two ports needed for Network Analysis with the Network Analyzer component to be found on the Meters component toolbar. As an example open the circuit EXAMPLES\RF\SPAR_TR.TSC.
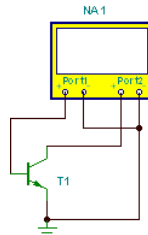
To analyze this circuit run Analysis/AC Analysis/Network Analysis. The amplitude diagram is as follows:
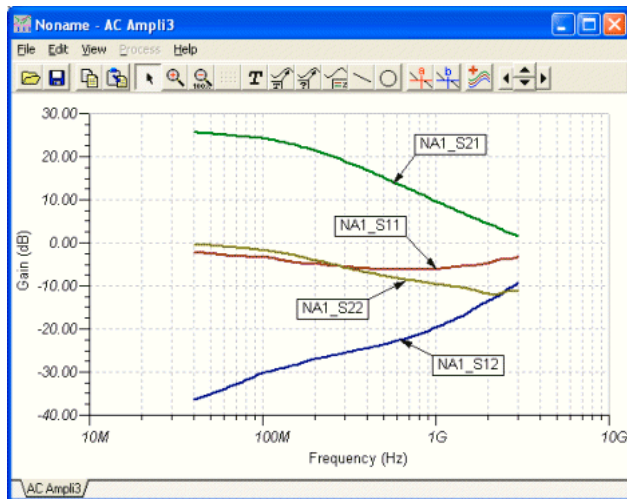
Note that we have added the labels to the curves using the Auto label tool of the diagram window. For more details on the Network Analysis see the "Network Analysisi and S-parameters" chapter of the advanced topics manual.

4.6.7 Analyzing a Digital Circuit with TINA's Digital Engine

Let's test a digital circuit. Open the file **HALF_ADD.TSC** from the EXAMPLES folder. Start the *Analysis|Digital Step-by-Step* command. A control panel will appear and you can examine the behavior of the circuit step-by-step by pressing the Step Forward button. Press the Play button for free-running mode. At each node a small box will indicate the logic level (Red for High, Blue for Low, Green for High Z, Black for undefined) as the circuit is clocked.

The picture below shows an intermediate state of the display.
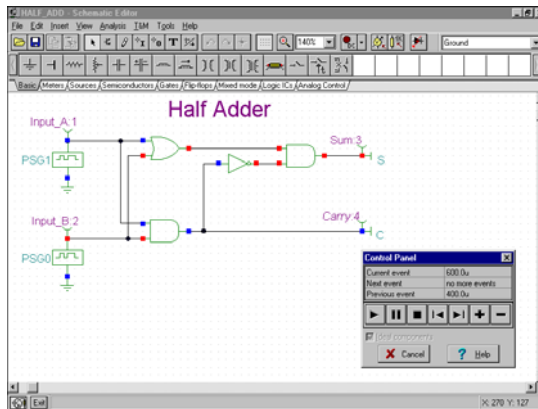
Now let's examine the transient behavior of the circuit. Selecting the *Analysis|Digital Timing Analysis* command, brings up this menu.
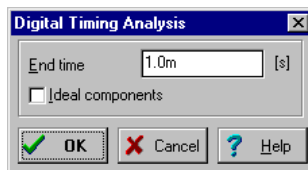
The result is shown on the timing diagram following.
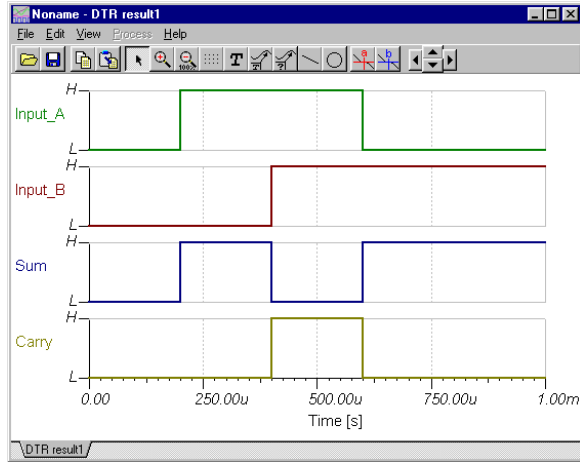
You could also select *Transient...* instead of *Digital Timing Analysis*, in which case the program would carry out an analog analysis, giving the detailed continuous waveforms and voltages instead of idealized logic levels. Note that circuits which contain only digital components can be analyzed by both digital and analog methods.

NOTE:

You can set the order of the curves by simply appending a colon (:) character and a number to the output name. This is particularly important when presenting the results of digital analysis, where each output is displayed as a separate diagram. For example, if you have outputs named OutA, OutB, Carry, and Sum, you can ensure that they will be displayed in the order given by using the labels OutA:1, OutB:2, Carry:3, and Sum:4.

The results of a purely analog analysis normally appear in one diagram: however, you can force TINA to display the results as separate diagrams, in the order you desire, by using the labeling method described above. You must use the View | Separate Curves command in the Diagrams window to separate the curves. If you don't use this labeling method, TINA presents the curves in alphabetical order.

DesignSoft

4.6.8   Analyzing Circuits using HDL models

Hardware Description Languages (HDL) are standard text-based modeling languages used by electronic designers to describe and simulate their chips and systems prior to fabrication.

TINA now includes the four most widely used Hardware Description Languages defined by IEEE standards: VHDL, Verilog, Verilog-A and Verilog-AMS.

VHDL and Verilog are used for modeling digital circuits. The two languages are comparable in modeling digital hardware. However the behavioral capabilities of VHDL are more powerful, while Verilog is easier to learn and understand. In TINA you can use and mix models of both languages.

Verilog-A is an easy to read high-level behavioral language for modeling analog electronic circuits and devices (e.g., bipolar and MOS transistors).

Verilog-AMS is an extension of Verilog for modeling analog and mixed signal circuits allowing both Verilog and Verilog-A instructions, connect modules, and rules.

A full presentation of HDLs in TINA is beyond the scope of this manual. We refer the interested reader to the detailed standards, manuals and information on the Internet: **www.vhdl.org** and **www.verilog.org** .

In the following sections we will demonstrate the use of these languages through examples.

4.6.8.1 Analyzing a Digital Circuit Using Digital VHDL Simulation

TINA includes a powerful digital VHDL simulation engine. Any digital circuit in TINA can be automatically converted into VHDL code and analyzed as a VHDL design. In addition you can analyze a wide range of hardware available in VHDL and define your own digital components and hardware in VHDL. The great advantage of VHDL is not only that it is an IEEE standard hardware description language, but also that it can be realized automatically in programmable logic devices such as FPGAs and CPLDs.

TINA can generate synthesizable VHDL code along with the corresponding UCF file (User Constraints File for pin assignment within the FPGA), if the Generate synthesizable code checkbox is set in the Analysis/Options menu. You can save the created VHD and UCF files with the "Create VHD & UCF File" command in the T&M menu. You can read the files with the free Xilinx Webpack and generate the bit stream file describing the implementation of the design and then upload it to Xilinx FPGA chips.
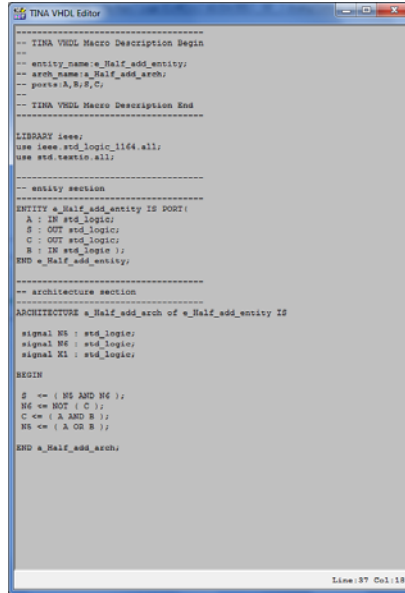
Before realizing a VHDL design, either with discrete components or with FPGA, verify it with simulation using TINA's Analysis|Digital Timing Analysis command. Let's examine some aspects of the VHDL simulation.

To do our first VHDL analysis, Open the FULL_ADD.TSC circuit from the EXAMPLES\VHDL folder. The following circuit will appear:
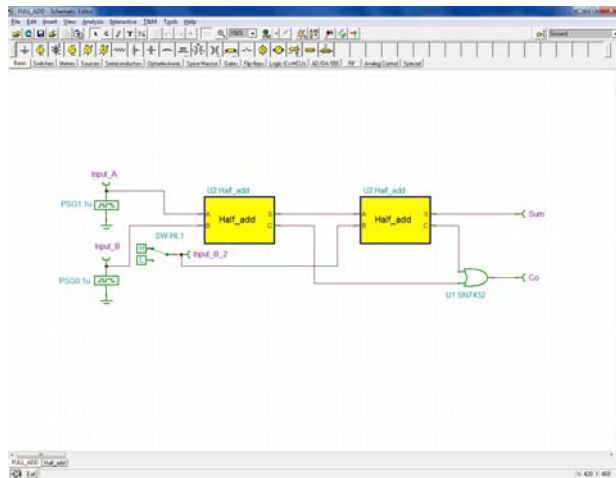
This circuit is a combination of two VHDL half adder blocks (macros) and a discrete OR gate.

If you double click on either of the HALF adder blocks and then press the Enter Macro button, the following window will appear:

DesignSoft

Note that the essential VHDL code of the half adder is at the bottom and it is only

```
S <= ( N5
AND N6 );
N6 <= NOT (
C );
C <= ( A
AND B );
N5 <= ( A
OR B );
```

At first glance, the code may look a bit strange, but it in fact is a machine translation of our half adder, assembled from gates in 4.6.1. Introducing the node names N5 and N6 as shown on the figure below, it is clear that

Half Adder
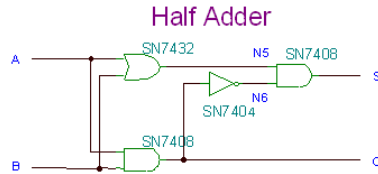

```
C <= ( A

AND B );

N6 <= NOT

( C );

N5 <= ( A OR B );
```

and therefore

```
S <= ( N5 AND N6 );
```

You might find it odd that in the VHDL code in the box, S appears to be calculated from N5 and N6 even before N5 and N6 have been calculated. This is valid, however, because VHDL is a concurrent language, and the order of the lines does not mean the order of execution.

The delays are taken from the given discrete values. But if the target hardware is an FPGA the synthesizer program will use the delay values of the FPGA data sheet.

Now select Digital VHDL Simulation from the Analysis menu and press OK. The following diagram will appear:

GETTING STARTED

DesignSoft
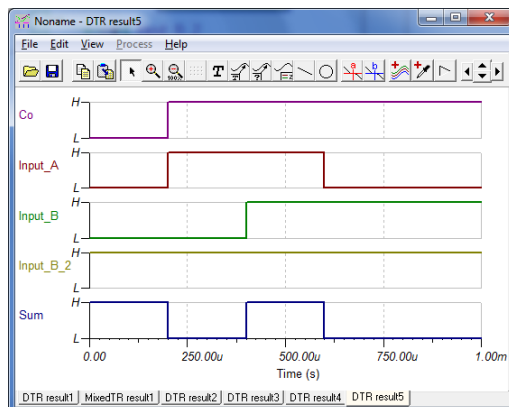
A great feature of TINA's VHDL is that you can not only view the VHDL code of each component, but you can edit and run them immediately. Let us replace the 4 line VHDL code -
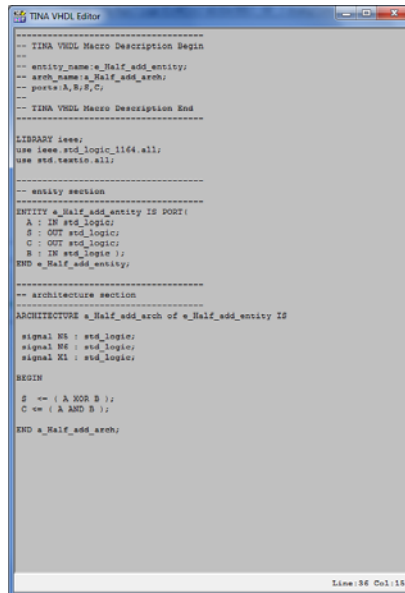
```
S <= ( N5 AND N6 ); N6 <= NOT ( C );

C <= ( A AND B ); N5 <= ( A OR B );
```

with this simpler 2 line code

```
S <= (A xor B); C <= (A and B);
```

This is easier to understand. In fact, if one of the A or B inputs is true, S is True. (A and B). We recognize this as an Xor function.

After editing the content of the VHDL blocks, they should look like this:

Now close the edit window by pressing 📂 on the Schematic Editor toolbar, select Digital VHDL Simulation from the Analysis menu, and press OK. The diagram that is drawn will be practically identical to the previous diagram.

NOTE:

> *In TINA of course you can make your own VHDL macros. This is decribed in chapter 5 under Making a VHDL macro from a .vhd file.*

4.6.8.2 The HDL Debugger: Debugging VHDL and Verilog codes

Debugging HDL programs is especially hard because of the concurrent processes in these languages.

A great feature in TINA is that the HDL debugger is now integrated. You can:

- Execute VHDL and Verilog codes statement-by-statement (Step)
- Execute subprograms as a single statement (Step Over)
- Add breakpoints (Toggle Breakpoint), running continuously (Start) and stopping at the breakpoints.
- Place variables, signals and other objects under the Watches tab and see their value during debugging.
- View all breakpoints and objects under the Breakpoints and Locals tabs at the bottom of the HDL debugger window.

To practice the use of the HDL debugger in TINA, open the vhdl_counter.TSC file from the EXAMPLES\VHDL folder with the Open command of the File menu. Next, click the Analysis menu and enable the debugger by clicking the Enable HDL Debugger line. Finally press, the DIG button on the toolbar at the top of the screen or click Start on the Interactive menu. The HDL Debugger will appear. Click the counter.vhd tab at the bottom of the code. You should see the following screen:

DesignSoft
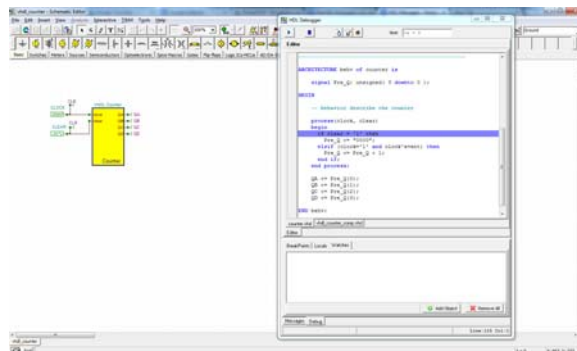
You will see two modules under the Tabs *counter.vhd* and *vhdl_counter_comp.vhd.* The first is the contents of the *VHDL Counter* macro, while the second file is the VHDL conversion of the whole circuit including the igitl sources.

This macro implements a counter. The counter entity consist of five processes, and all processes run in parallel. The first process is sensitive to the clock and the clear signal. So when one of these signals is changing, this process is triggering and executing. The other processes are sensitve to the Pre_Q signal. When Pre_Q(i) changes, the *i+1th* process is triggering and executing.

To follow some important steps in the program, let's add four breakpoints by clicking the lines of interest and pressng the Toggle Breakpoint button. The lines with the breakpoint will be marked by a red background:
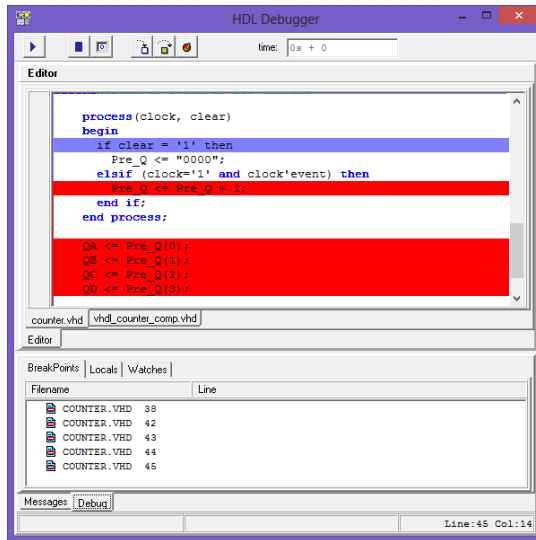
Note that under the Breakpoints tab you can see all the breakpoints and remove any of them with the Toggle Breakpoint button.

Now click the Watches tab, press the Add Object button and one-by-one add the Pre_Q signal and the QA, QB, QC, QD ports. These will be connected with the 4 outputs of the macro.

You will see the following in the HDL Debugger window:
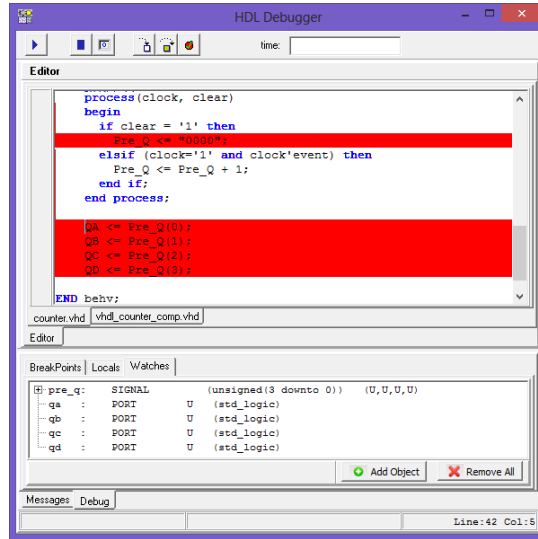
Now let's start debugging by pressing the Run button.

In the VHDL simulation each process runs once at the start of the simulation. Click the Run button several times until the debugger shows *500ns+1* in the *time* field. It means the simulator reaches 500ns and 1 delta cycle (Delta cycle is a special VHDL time period of infinitesimal duration). At this time clock='1', Pre_Q was initialized with '0'-s at *clear*. The line Pre_Q <= Pre_Q+1 will schedule a transaction on signal Pre_Q with value '0001' for time 500ns+2.

Press Run again and the debugger will stop at breakpoint QA <= Pre_Q(0). At this point time=500ns+2, because the nearest event was the previously described scheduled event. Now the simulator will schedule an event which will assign the value '1' for time=500ns+3 for the QA port.

Press Run again, now time=1.5us+1 and QA='1'. Note, the last three processes are now not triggered because there were no changes in their sensitivity list.

You can study updating the other ports in a similar way. There is a similar example in Verilog called verilog_counter.tsc in the Examples\Verilog folder.

4.6.8.3 **Analyzing a Digital Circuit Using Digital Verilog Simulation**

TINA also includes a powerful digital Verilog simulation engine. The advantage of Verilog compared to VHDL that it is easier to learn and understand, however there are more features in VHDL.

Verliog -similarly to VHDL- can also be realized automatically in programmable logic devices such as FPGAs and CPLDs.

TINA translates the Verilog models and the other digital components to synthesizable VHDL code along with the corresponding UCF file (User Constraints Fiile) for pin assignment within the FPGA), if the Generate synthesizable code checkbox is set in the Analysis/Options menu. You can save the created VHD and UCF files with the "Create VHD & UCF File" command in the T&M menu and, using the free Xilinx's Webpack software, generate the bit stream file describing the implementation of the design and then upload it to Xilinx FPGA chips.

Before realizing a Verilog or any other digital HDL design, either with discrete components or FPGA, you need to verify it with simulation with TINA's Analysis | Digital Timing Analysis command.

Let's run the previous VHDL circuit along with its Verilog model. the Full adder Verilog and VHDL comparison time diagram.TSC circuit from the EXAMPLES\Verilog folder. The following circuit will appear:
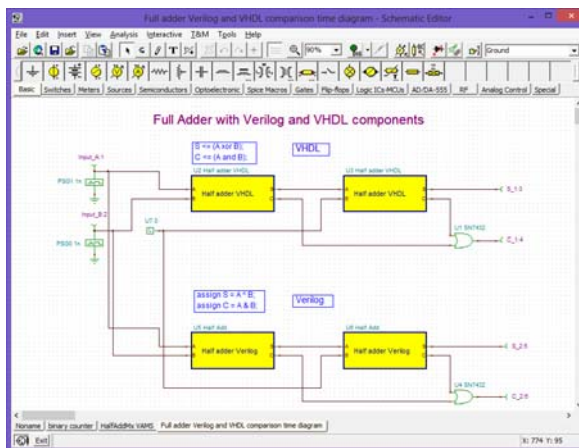
You can see realization of the half adder function in both languages, they are very similar. You can double click the VHDL or the Verilog macros and press Enter Macro to see all the details.

Now run the Digital Timing Analysis from the Analysis menu. The following diagram will appear:
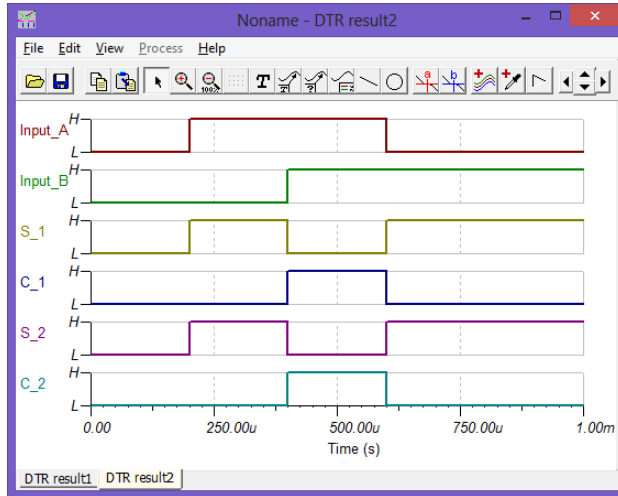
You can see that the output signals from both models are exactly the same.

4.6.8.4 Analyzing Circuits Using Verilog-A models

Today the most widely used language to describe electronics circuits and device models is the Spice netlist format (1973). However the Spice netlists are often hard to read and understand, and they lack a lot of the functionalities of programming languages which engineers would need while creating models and simulation.

The relatively new Verilog-A language (1995) provides an alternative method with an easy to read programming language style C like syntax. Thus Verilog-A is a suitable successor of the SPICE netlists for describing circuit topologies.

Most of the device libraries of TINA are in Spice netlist format. However you can already create and import models and place TINA macros in Verilog A format. You can find several language examples, device models, and circuits in the Examples\Verilog A folder of

TINA. Creating macros in Verilog A is described in Chapter 5 of this manual.

For a demonstration of Verilog-A in TINA, load the examples in the EXAMPLES\Verilog-A\Device Models folder.

We suggest that you start with the Opamp Model Comparison.TSC file where a simple opamp model is realized in three different ways: Verilog-A, Spice, and the schematic diagram.

You can also study nonlinear device models in Verilog-A and their characteristics in the other examples: diode.TSC, JFET.TSC etc.

4.6.8.5 Analyzing Circuits Using Verilog-AMS models

An even more sophisticated method of describing electronics circuit, containing both analog and digital components is the Verilog-AMS language. As we observed earlier, Verilog-AMS is a derivative of the purely digital Verilog extended with the purely analog Verilog A and an interface for the connection of the analog and digital parts.
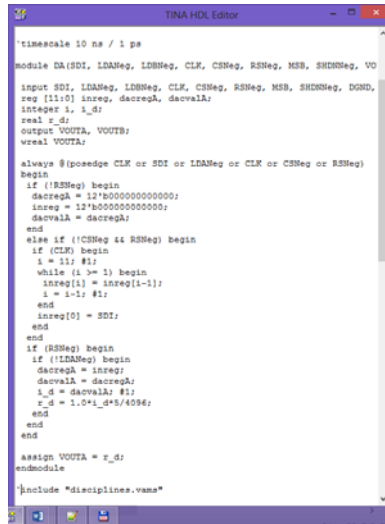
In TINA you can also create or import Verilog AMS macros for modelling mixed signal devices.

Let's see the structure of such a model. Open the DAC VAMS.TSC circuit from the EXAMPLES\Verilog AMS folder. The following circuit will appear.
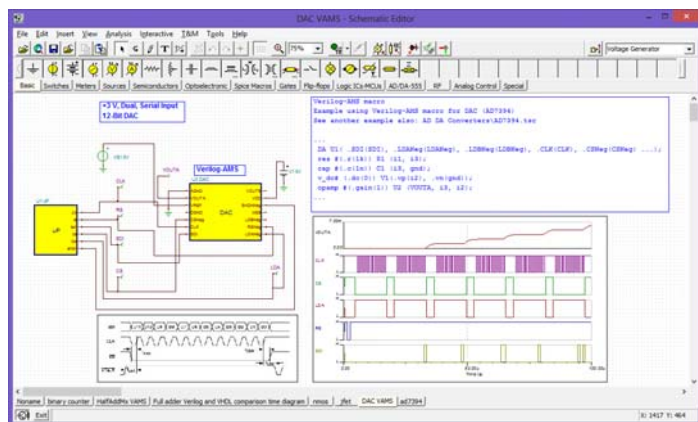
This circuit contains a Digital Analog Converter (DAC) macro with Serial Peripheral Interface (SPI) and a test bench macro, generating the digital SPI signal.The DAC model is defined in Verilog AMS. Interestingly, test bench on the left side is written in VHDL which is an example of mixing different HDLs but here we will concentrate on the Verilog AMS macro on the right.

To see the Verilog AMS code of the model double-click the DAC macro and press the Enter Macro button. The following window will appear.

We will not go into a detailed analysis of the code. We just want to show that in the first part shown above, the DA Verilog module converts the serial signal into an analog signal (VOUTA).

At the end of the macro shown below (you can scroll down there), the DA module is called and the signal is smoothed by a simple opamp and an RC filter using Verilog A instructions. You can also see the definition of the capacitor in the code fragment below.

DesignSoft

```
module cap(p,n);
 inout p,n;
 electrical p,n;
 parameter real c=1e-9 from [0:inf];

 analog
    I(p,n) <+ ddt(c*V(p,n));
endmodule

module opamp(vout, vin_p, vin_n);
     inout vin_p, vin_n;
     inout vout;
     electrical vout, vin_p, vin_n;
     parameter real gain = 1;

     analog
          V(vout) <+ gain*V(vin_p, vin_n);
endmodule

connectrules MyRules;
 connect a2d_TTL input electrical, output logic;
 connect d2a_TTL input logic, output electrical;
endconnectrules

module macro1(SDI, LDANeg, LDBNeg, CLK, CSNeg, RSNeg, MSB, SHDNN
 input SDI, LDANeg, LDBNeg, CLK, CSNeg, RSNeg, MSB, SHDNNeg, DGN
 inout VOUTA, VOUTB;
 wire i1;
 electrical VOUTA, i2, i3;
 ground gnd;

 DA U1( .SDI(SDI), .LDANeg(LDANeg), .LDBNeg(LDBNeg), .CLK(CLK),
 res #(.r(1k)) R1 (i1, i3);
 cap #(.c(1n)) C1 (i3, gnd);
 v_dc# (.dc(0)) V1(.vp(i2), .vn(gnd));
 opamp #(.gain(1)) U2 (VOUTA, i3, i2);
endmodule
```

Line:109 Col:10

4.6.8.6 Analyzing Circuits Using SystemC

SystemC is another great tool for modeling hardware. It includes all the features of C++, used all over the world, and a C++ class library specially designed for system design. SystemC has an open-source free implementation and you can compile it into a very efficient executable binary code with the also free Visual Studio Community C++ compiler of Microsoft. In SystemC you can model hardware at a higher abstraction level than in other HDLs and so for modeling some very complex hardware e.g. microcontrollers it is more easy and efficient to use than other HDLs like VHDL or Verilog. In v11 and later versions of TINA you can also create and use components modeled in SystemC both in TINA and TINACloud. The following are the requirements to use SystemC with TINA.

Compiler requirements

Use Microsoft Visual Studio to compile SystemC models. In our examples we are using Microsoft Visual Studio 2015.

SystemC distribution

Use the systemc-2.3.1 SystemC distribution.

Compiling the SystemC distribution (SystemC.lib)
Use the MSVC project file in the distribution (<sc_home>\msvc80\SystemC)
-       C++/Code generation: Multi-threaded Debug DLL (/MDd)
-       Extra cmd line option: /vmg
-       Remove this line from <sc_home>\src\systemc.h: using std::gets;
-       Build the project and the result will be in: <sc_home>\msvc80\SystemC\Debug\SystemC.lib

Compiling your model

Use the project template in <TINADir>\Examples\ SystemC\ systemc_model.zip (systemc\systemc_model.vcxproj) .

Open Visual Studio and open the project file. Open the property manager (View/Other windows/Property manager) and select the "Macros" entry, select Common properties/User macros. Change the SC_HOME macro where you extracted the systemc_model.zip file.

Compile the Debug configuration. If you'd like to test your model on TINACloud you have to compile the SystemC distribution and the systemc_model project with (/MTd) (Multi threaded Debug)

Creating a SystemC macro in TINA

Similarly to the other HDL and Spice components you should turn your SystemC model into a TINA Macro.
Here is how to do it.
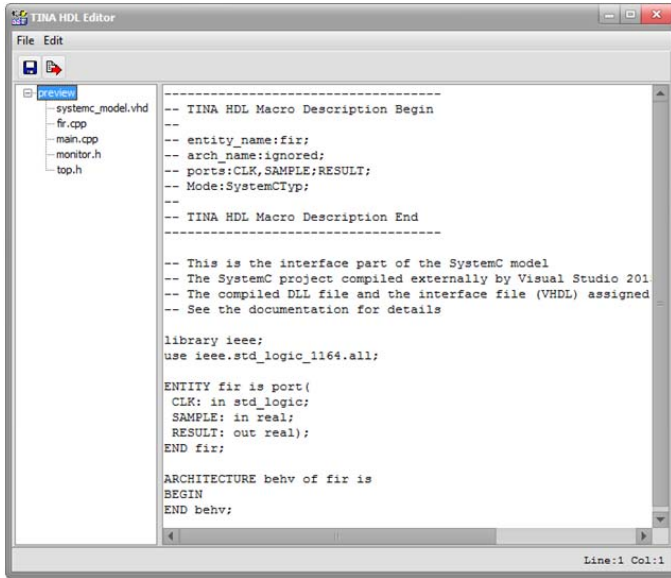
Modelling requirements

In sc_main use dynamic allocation for the top level module. Return with 0 after creating the top instance.

```
int sc_main(int argc, char* argv[])
{
    top* TOP = new top("TOP");

    return 0;
}
```

The SystemC macro consists of a VHDL file and a SystemC dll file. The VHDL file is an interface file only. The SystemC top level module must contain the same signals as in the VHDL interface module and in the same order.
Example of counter circuit in SytemC

Top level SystemC file:

```
SC_MODULE(top)
{
```

```
sc_signal<sc_logic> CLK,CLEAR,QA,QB,QC,QD;

  counter U1;

  SC_CTOR(top): U1("U1")
  {
      U1.CLK(CLK); U1.CLEAR(CLEAR); U1.QA(QA);
U1.QB(QB); U1.QC(QC); U1.QD(QD);
  }
};
```

The most important part of the counter SystemC model

```
#ifndef counterH
#define counterH

#include "monitor.h"

SC_MODULE(counter)
{
  sc_in<sc_logic> CLK, CLEAR;
  sc_out<sc_logic> QA, QB, QC, QD;
  int value;

  monitor MON;

  void proc()
  {
     double t = sc_get_curr_simcontext()-
>time_stamp().to_seconds();
   if (CLEAR.event() && CLEAR == SC_LOGIC_0 ) {
    value = 0;
   }
     else if (t > 0 && CLK.event() && CLK ==
SC_LOGIC_1) {
    value++;
    if (value == 10) value = 0;
   }

   sc_lv<4> sclv_value(value);
   QA = sclv_value[0];
   QB = sclv_value[1];
```

DesignSoft

```
 QC = sclv_value[2];
 QD = sclv_value[3];
}

SC_CTOR(counter): MON("MON")
{
    value = 0;
    SC_METHOD(proc);
    sensitive << CLK.value_changed() << CLEAR;
    MON << QA << QB << QC << QD;
}
};
#endif
```

VHDL interface file

```
library ieee;
use ieee.std_logic_1164.all;

ENTITY counter is port(
 CLK: in std_logic;
 CLEAR: in std_logic;
 QA, QB, QC, QD: out std_logic);
END counter;

ARCHITECTURE behv of counter is
BEGIN
END behv;
```

TINA has to pause the SystemC simulation in mixed mode after every top level output port change, to do this create a monitor instance and call our sc_pli_set_node_changed(), then call sc_pause().
The following is an example code for a counter.

```
#ifndef monitorH
#define monitorH

#include "systemc.h"
#include "C_SCPLI.h"


SC_MODULE(monitor)
{
```

```
  sc_in<sc_logic> QA,QB,QC,QD;

  SC_CTOR(monitor)
  {
    SC_METHOD(proc);
    sensitive << QA << QB << QC << QD;
  }

  void proc()
  {
    sc_pli_set_node_changed(true);
    sc_pause();
  }
};
#endif
```

Supported top level port types in mixed mode
`sc_bit, sc_logic, bool, double`

TINA Analysis setup

Select a proper time step in Analysis/Set Analysis Parameters/TR
maximum time step

Running the model

You have to install the "Visual C++ Redistributable for Visual
Studio 2015" if you don't have Visual Studio 2015 installed.

Examples

Counter example
Use the project template in <TINADir>\Examples\SystemC\
systemc_model.zip (systemc\systemc_model.vcxproj) .
Extract this zip file (<sc_model>)

Open Visual Studio and open the project file. Open the property
manager (View/Other windows/Property manager) and select the
"Macros" entry. Change the SC_HOME macro where you extracted
the systemc_model.zip file.

Copy the *<sc_model>\systemc_model\Examples\counter\top.h* to <sc_model>\systemc_model and *<sc_model>\systemc_model\Examples\counter\monitor.h* to <sc_model>\systemc_model. Rebuild the project.

Copy the <sc_model>\systemc_model\Examples\counter\systemc_model.vhd and <sc_model>\Debug\systemc_model.dll to a directory (for example d:\Temp).

In TINA Open the <TINADir>\Examples\SystemC\counter.tsc, delete the counter macro.
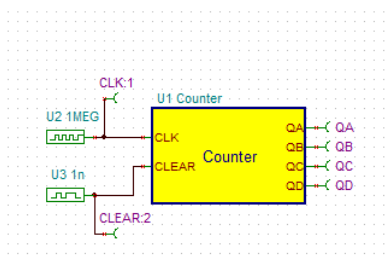
Now select Tools/New Macro Wizard… Type Counter in the Macro Name field, select From File, press the folder icon. In the dialog select Files of type "SystemC executable" and search for the previously copied systemc_model.dll.
Insert the new macro to previously place of the previously deleted macro. Run Transient. The result is the following.
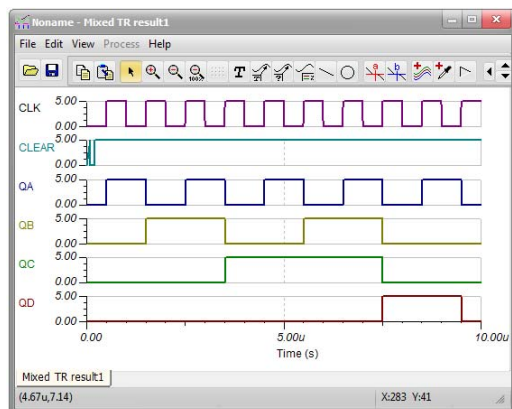
Microcontroller example

This examples assumes you have the Microchip XC8 compiler installed.

Now let's make a new microcontroller that is currently not in TINA (PIC16LF1906).
At first build the new model. Copy the <sc_model>\systemc_model\Examples\pic16lf1906\top.h to <sc_model>\systemc_model and <sc_model>\systemc_model\Examples\pic16lf1906\monitor.h to <sc_model>\systemc_model. Rebuild the project.

Copy the <sc_model>\systemc_model\Examples\pic16lf1906\systemc_model.vhd and <sc_model>\Debug\systemc_model.dll to a directory (for example d:\Temp).

Flasher example

In TINA Open the <TINADir>\Examples\SystemC\PIC16LF1906 sc flasher.TSC, delete the PIC macro.
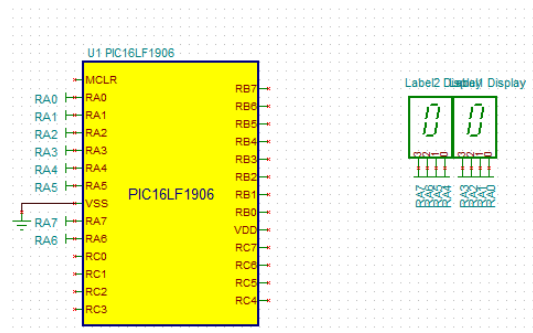
Now select Tools/New Macro Wizard… Type PIC16LF1906 in the Macro Name field, select From File, press the folder icon. In the dialog select Files of type "SystemC executable" and search for the previously copied systemc_model.dll. Select the "Options" button and check Enabled, check device type PIC16, enter "device name" PIC16LF1906, press OK.
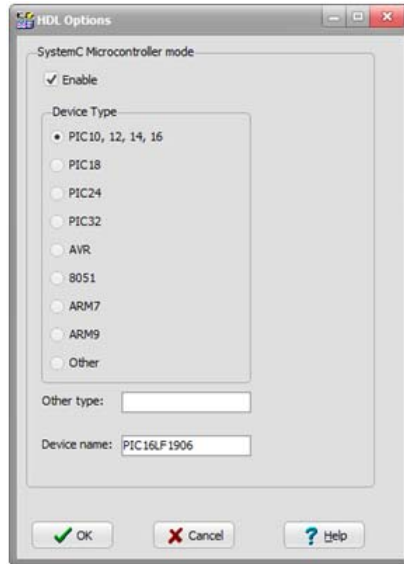
Insert the new macro to the place of the previously deleted macro.

Now assign a C code to the macro. The code implements a counter. The counter value will be displayed on PORTA.

Click on the PIC macro and press the "..." field in the ASM-Code field. Select C code and copy the content of the flasher.c (<sc_model>\systemc_model\Examples\pic16lf1906\flasher.c) as a new file in the editor, press Make, press OK.

Press the TR button to Run interactive transient.

The flasher C code. This C code will run on the SystemC microcontroller model.

```
#define _XTAL_FREQ 1000000

#include <xc.h>
#include <stdio.h>
#include <stdlib.h>
/*
 *
 */
```

```
int main(int argc, char** argv)
{
     unsigned char a, b, op, res;

     TRISA = 0x00;
     res = 0;

     while (1) {

     PORTA = res;
     res++;

     }

     return (EXIT_SUCCESS);
}
```

Calculator example

Now let's make a simple calculator.

In TINA Open the <TINADir>\Examples\SystemC\PIC16LF1906
sc calc.TSC, delete the PIC macro.

GETTING STARTED
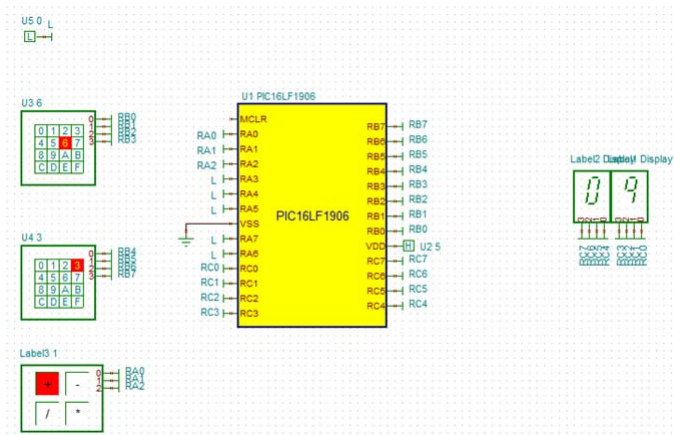
The steps are the same to the previous process except now we assign the calc.c code.
(alternatively you can copy paste the macro from the previous circuit and change the C code)

Press the TR button to Run interactive transient.
Press the numbers on the keypad and +,-,/,* signs. The hex displays will show the result.

The calculator C code. This C code will run on the SystemC microcontroller model.

```c
#define _XTAL_FREQ 1000000

#include <xc.h>
#include <stdio.h>
#include <stdlib.h>

/*
 *
 */
int main(int argc, char** argv)
{
      unsigned char a, b, op, res;

      TRISA = 0xFF;
      TRISB = 0xFF;
      TRISC = 0x00;

      while (1) {

      a = PORTB & 0x0F;
      b = (PORTB & 0xF0) >> 4;
      op = PORTA & 0x0F;

if (op == 1)
res = a+b;
else if (op == 2)
res = a-b;
else if (op == 3)
res = a/b;
else
res = a*b;
PORTC = res;
```

```
        }
        return (EXIT_SUCCESS);
}
```

Low-pass filter example

Open the <TINADir>\Examples\SystemC\sc lpf.TSC example.
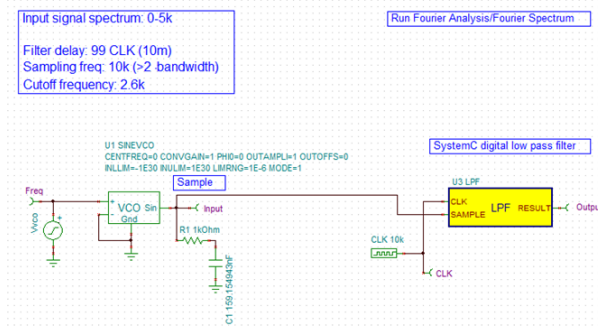
The filter characteristics and C-code is designed by the free tool at
http://t-filter.engineerjs.com/

The generated C-code was placed in the fir.cpp file.
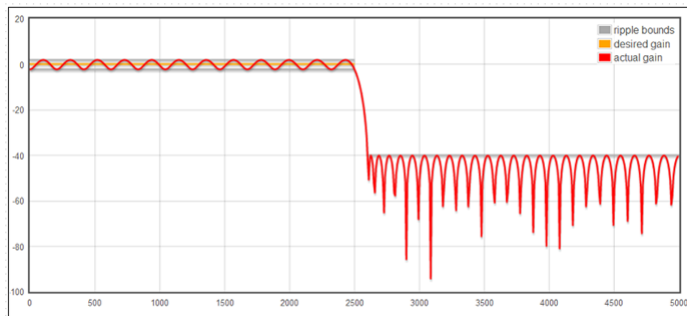
Run Analysis/Fourier Spectrum…

Press the 📈 button in the diagram window. The Post-processor will appear. Now draw the transfer function.

Press the *More* button. Type *Output(s)/Input(s)* in the Line Edit. Type *H* in the new function name. Press *Create*. Press *OK*. Select *View/Separate curves*.

The result is the following
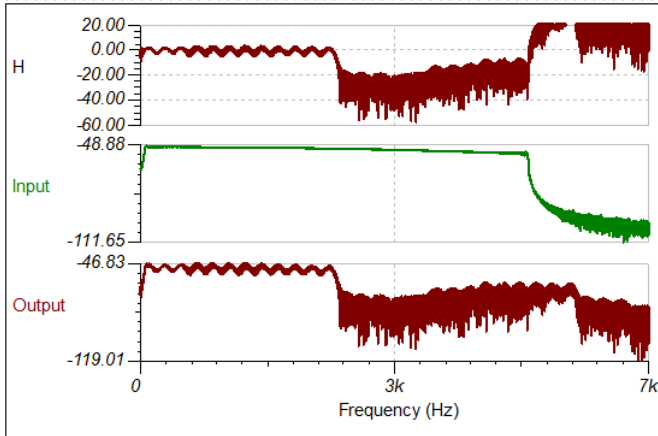
The most important part of the SystemC macro:

```
void fir::proc()
{
 double u, y;

 if (CLK.read() == SC_LOGIC_1) {
  sc_logic sc_val;

  // CALC
  x[0] = SAMPLE; y = 0;
  for (int k=0; k<M; k++) {
   y += b[k]*x[k];
  }

  // SHIFT
  for (int k=M-1; k>=1; k—)
   x[k] = x[k-1];

  // SET VALUE
  RESULT = y;
n++;

 }
}
```

You can find this example in the Examples/Fir folder of the
systemc_model.zip.

How to create your own microcontroller

You can create your own microcontroller model based on our code (<TINADir>\Examples\SystemC\systemc_model.zip). Our code implements the PIC16LF1906 PIC microcontroller SystemC model core (no peripherals).

Now let's make a new microcontroller model (PIC16LF1907 40 PIN PDIP). This new PIC model has two new port (PORTD, PORTE).

At first modify the VHDL interface file. Change the enity name to PIC16LF1907 and add ports *RD0-7, RE0-3*. You can find the modified file in *<sc_model>\systemc_model\Examples\pic16lf1907\ systemc_model.vhd.*

Next add the port handling to the SystemC model. Add the new ports and sensitive statements to monitor.h. Add the new top level signals in top.h. Add the new *PORTD, PORTE* defines to pic.h, add the new pin declarations, add the new sensitivity to *MON*. In PicSimulator.h add the new defines *TRISD, TRISE*. In *PicSimulator::SetDevice* modify the *SetPinLayout* call to reflects the new pins. In *pic::OnChangePinData* add the PORTC-PORTE cases, add pin assignments *RD0-7, RE0-3*, in *pic::IsTRISAddress* add new cases *TRISD, TRISE*
The changes you will find in <sc_model>\systemc_model\ Examples\pic16lf1907.

Now let's test the new microcontroller model.
Compile the model with Visual Studio. Copy the compiled model (systemc_model.dll) and the the interface file (systemc_model.vhd) to a directory. Create your new macro as described previously.

Now let's test the new port functionality. The flasher.c in *<sc_model>\systemc_model\Examples\pic16lf1907* is modified, it writes the data to the new port, PORTD. Assign the C code as described previously. Set a small time step (1u) in Analysis/Set Analysis Parameters/TR maximum time step. Test by pressing the TR button.

GETTING STARTED

DesignSoft
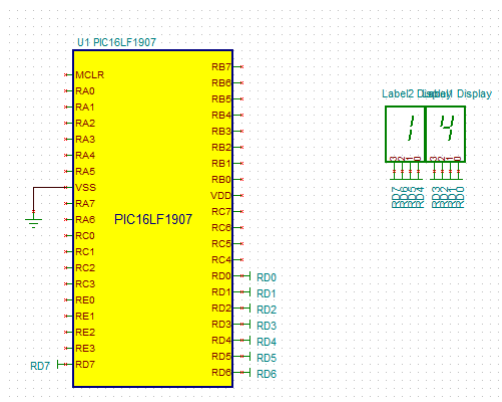
4.6.9 Mixed Mode Simulation (Spice - VHDL - MCU co-simulation)

TINA version 8 and above include a new powerful mixed mode simulation engine. It is based on the XSPICE mixed mode algorithm, extended with MCU and VHDL components. In your circuits you may freely mix any analog or digital components of TINA, including microcontrollers (MCUs) and macros with Spice or VHDL content.

You can modify these components on the fly along with the code in the MCUs. TINA will analyze the analog parts in analog, the digital parts in digital, and will automatically create the interfaces among the components. This ensures synchronization and fast convergence.

Let's explore some of the uses of this mode through a few examples.

4.6.9.1 Waveform generation with a VHDL and Spice subcircuits

The following circuit (EXAMPLES\VHDL\Mixed\Wave generator.TSC) generates an analog sine or sawtooth signal depending on the status of the left SW-MODE switch.

Digital wave generator   **DA converter**   **Low pass filter**
VHDL subcircuit   **SPICE subcircuit**
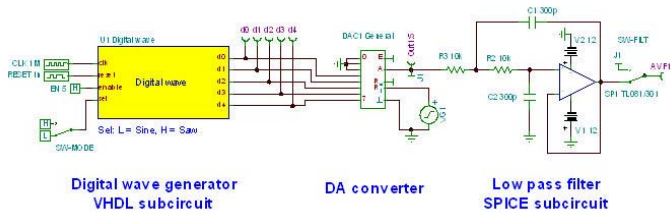
The Digital Wave box on the left of the circuit includes a VHDL code with a lookup table *Sine_LUT* for the sine wave and a counter for the sawtooth signal. The essential part of the VHDL code is:

```
process(Reset, Clk)
begin
      if (Reset = '1') then
      Wave <= (others => '0');
      LUT_index <= 0;
   elsif rising_edge(Clk) then
      if (Enable = '0') then
      Wave <= (others => '0');
   elsif (Sel = '0') then
      Wave <= Sine_LUT(LUT_index);
 else
      Wave <= conv_std_logic_vector(LUT_index,5);
 end if;
             if (LUT_index = LUT_index_max) then
             LUT_index <= 0;
      else
             LUT_index <= LUT_index + 1;
      end if;
      end if;
end process;
 d0 <= Wave(0);
 d1 <= Wave(1);
 d2 <= Wave(2);
 d3 <= Wave(3);
 d4 <= Wave(4);
```

You can see all the details of the code and modify it if necessary by double-clicking the Digital Wave box and pressing the Enter Macro button on its property dialog.

DesignSoft
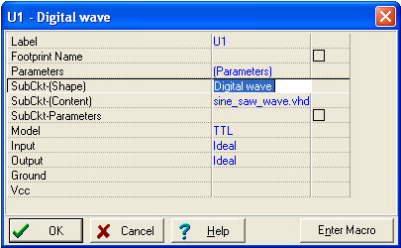
Note that the model is set to TTL in this dialog, but you may select from various other models (CMOS, LS, HC, HCT etc.).

The digital output of the counter is converted into an analog signal in the 5 bit DA converter of TINA shown in the middle of the circuit.

The DAC sine wave output needs to be cleaned up with a low pass filter. We will use a Spice opamp model of the TL081 in a Sallen and Key low pass filter configuration. Press the Enter Macro button on the property dialog and TINA will open the macro. You can review and, if necessary, modify the Spice code inside the macro.

The sawtooth signal from the counter output (on pin J1) does not need to be filtered, so we will connect it directly to one terminal of switch SW_FILT. The sine wave developed at the DAC output (DAEX) does, in fact, require filtering, so we will pass it through the low pass filter and connect the filtered Aout analog output to the other terminal of SW_FILT. A jumper (J1) connects the DAEX output to the switch. Although it's not obvious in the schematic, the switches SW_FILT and SW_MODE are synchronized as though they were a DPDT switch. We cause them to be synchronized by assigning both switches to be controlled by the Hotkey A. See the property dialog for SW_FILT where the Hotkey has been assigned to A:
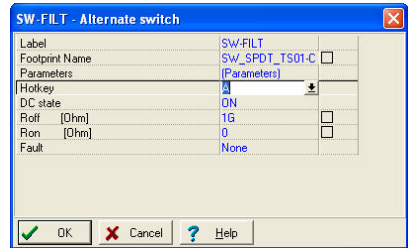
Here are the final waveforms of the full circuit, including the five counter output waveforms. SW_MODE is in the High state, selecting the sawtooth signal.
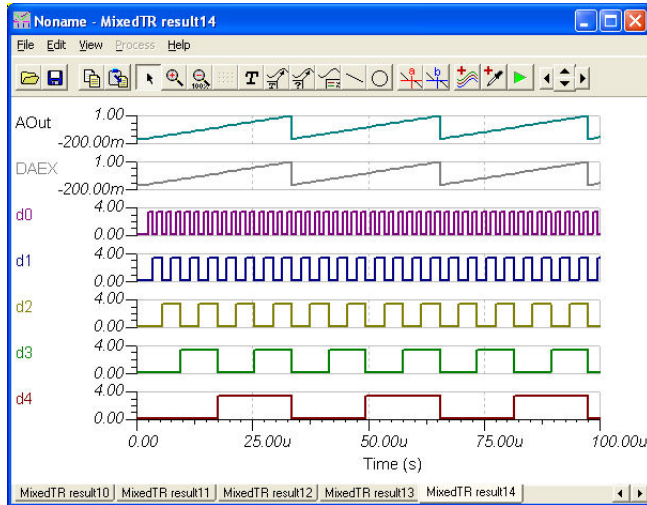
If we change the SW-MODE switch to Low and run Transient analysis again, the waveforms are:
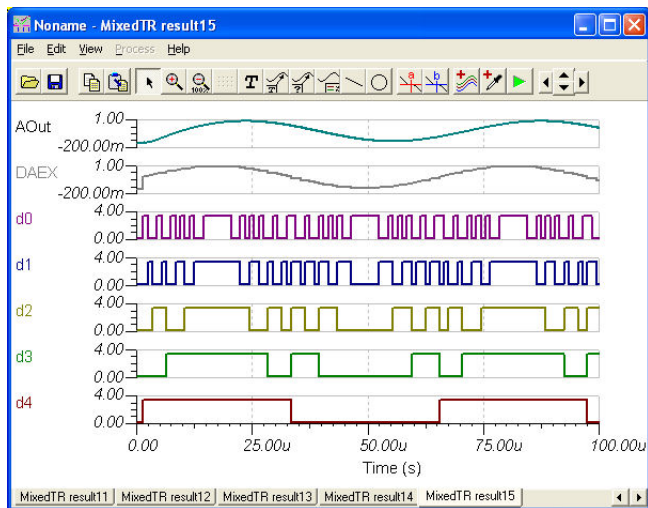
To see the effect of the analog filter, delete curves d0 to d4 from the diagram by clicking the curves and pressing the Del key. Alternatively, you can delete outputs d0 to d4 temporarily and run Transient Analysis again.
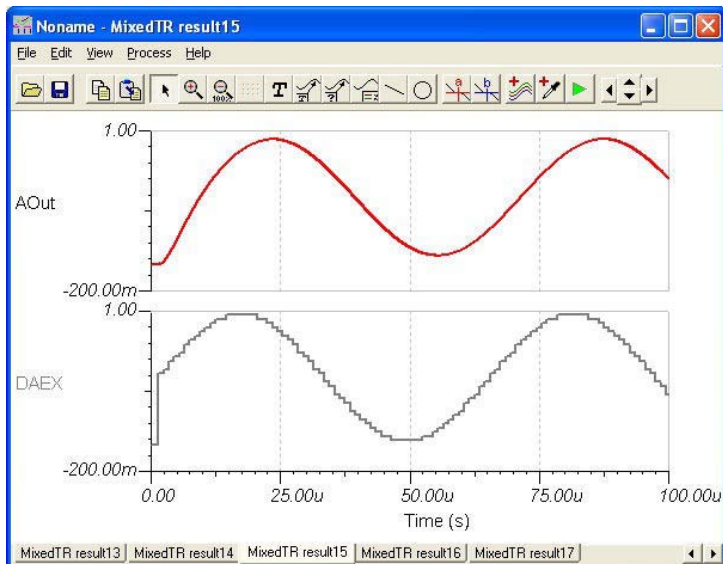
To demonstrate the flexibility of TINA's VHDL features, we'll modify the VHDL code to generate a square wave instead of the sawtooth waveform. Simply set Wave(0) to Wave(3) to zero in the VHDL code.

Double-click the Digital Wave macro and press the Enter Macro button. Locate the Wave <= conv_std_logic_vector(LUT_index,5) line and insert the following statements:

```
Wave(0) <= '0';
Wave(1) <= '0';
Wave(2) <= '0';
Wave(3) <= '0';
```

You can update the macro by simply closing the editor (click the x button in the top-right corner of the window). The following message will appear:

GETTING STARTED

DesignSoft

Press the Yes button to approve the changes.

Now you can run Transient or Mixed VHDL Simulation from the Analysis menu to get the following waveforms.
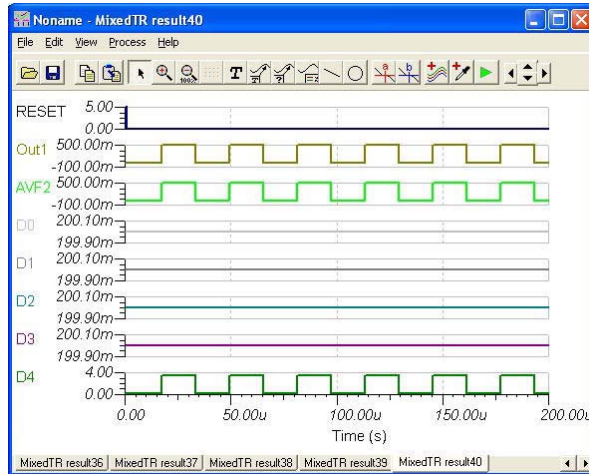
You can check out a more complex version of this circuit under EXAMPLES\VHDL\Mixed\Wave generator dipsw.TSC. There you can select all the three waveforms we discussed using a dip switch.

Note that you can download the VHDL portion of the code into an FPGA and use hardware form.

4.6.9.2 MCU controlled SMPS circuit

The mixed mode simulator of TINA not only allows MCUs, but also any linear or nonlinear parts in TINA's libraries. As an example, let's study the following circuit, which realizes a DC-DC converter, converting 5V DC to 13V DC, and operating in boost mode. You can find this circuit in TINA under EXAMPLES\Microcontrollers\Pic\ Boost_converter.TSC .

The PIC MCU in the circuit produces a PWM output at pin RB0 that controls the switching FET. The interrupt routine of the code in the PIC compares the feedback voltage at VFB (connected to pin RA1 of the PIC), with a built in threshold voltage. If the voltage is lower than the threshold defined in the code, the duty cycle of the PWM output waveform is increased. You can study the ASM code in the PIC by double-clicking on the PIC, clicking on the MCU line, pressing the ··· button (the little button with three dots in a row), and finally pressing the Edit ASM button. You can see and debug the code on the fly here. Click on the Enable MCU code debugger line under the Analysis menu, press the TR interactive transient analysis button (or select it on the Interactive menu), and, finally, click on Start.

The waveforms below demonstrate how the analog parts and the MCU interact in TINA.

4.6.10 Using IBIS Models in TINA

IBIS (Input/Output Buffer Information Specification) is a method to provide modeling information about the input/output buffers of integrated circuits. The good thing about IBIS models is that they are often available even for devices where complete device models are not available from manufacturers for any reason (e.g., complexity,proprietary information protection, etc.). One of *the most popular uses of IBIS models is Signal Integrity Analysis*, including impedance matching and more. TINA currently supports the most widely used IBIS 4.2 version.

In TINA, you can convert IBIS models to TINA Spice macros and then use them in any circuits in TINA. You can also complete simplified digital device models—e.g., MCUs with IBIS models—to better describe their analog behavior.

In the following, we will show the use of IBIS models through an example of fixing signal integrity between a Texas Instrument TMS320C6748DSP and an ADS1259 delta-sigma ADC.

Select *File/Import/IBIS File (*.ibs)*, select *c6748zce.ibs* from *<TINA directory>\Examples\IBIS*. The following dialog will be displayed. In this dialog, you can select the model to import.

DesignSoft

GETTING STARTED

Now select *SPI1CLK_GP213*signal, PBFZP18LL_X50_PI_3P3 model (cell operated at 3.3Vwithout pullup or pulldown),and *Typ* value set. Press OK. The IBIS model is automatically converted to a Spice macro.

SPI1CLK_GP213is the master configuration serial clock signal of TMS320C6748chip to drive SPI clock input of an AD converter, Texas Instruments ADS1259.

Select *File/Import/IBIS File (*.ibs)*, select *ads1259.ibs* from *<TINA directory>\Examples\IBIS*. The following dialog will be displayed. In this dialog, you can select the model to import.

Now select *SCK* input signal, DIN_PD_3 model and *Max*value (for 3.3V DVDD voltage range). Press OK. The IBIS model is automatically converted to a Spice macro.

U11 Input
C_comp=6.362p R_pkg=1.88m L_pkg=814.51p C_pkg=204.54f
dVr=0 dt_r=0 dVf=0 dt_f=0 VCC=3.3

Connect the DSP I/O buffer to the input of the ADC with a lossless transmission line. Add the power source and voltage generator to create clock signal of DSP side. Place voltage pins for the simulation onto the signal nodes.

We adjust the transmission line parameters to a few inches of microstrip trace routed on a four-layer PCB. This produces cc. 500ps delay and 90 Ohms characteristic impedance.

File from *<TINA directory>\Examples\IBIS\Impedance matching of TMS320C6748.TSC*is ready to be used.

Now, click Analysis, Transient. The DSP transmits the SPI clock signal where the impedance mismatch creates reflections. The result shows the reflections created by the impedance mismatches in this circuit simulation.

At the ADC side(pin EoTL),the voltage is beyond the ground and the supply voltage, which violates the absolute maximum rating of the digital input.

DesignSoft

To avoid under and overshoots at the line end is to match the output impedance of the driver to the trace impedance by inserting a resistor between the output and the trace. Let us place a 100 ? resistor in series now with the output.

Run the transient analysis again, and compare the results by copying the important curves with each other.

Now, we can see that using the IBIS model to understand and find the critical issues with the simulation helped to solve this problem.

DesignSoft

4.6.11 Testing your Circuit in Interactive mode

When everything is in order, the ultimate test of your circuit is to try it in a "real life" situation using its interactive controls (such as keypads and switches) and watching its displays or other indicators. You can carry out such a test using *TINA's* interactive mode. Not only can you play with the controls, but you can also change component values and even add or delete components while the analysis is in progress. The interactive mode is also very useful for educational and demonstration purposes, for tuning circuits interactively and for interactive circuits which you cannot test otherwise, e.g., circuits with switches, relays, or microcontrollers. First select the interactive mode required (DC, AC, TR, DIG or VHDL) with the ▾ button, then press the ▣ button. XX can be DC, AC, TR, VHD etc. depending on the mode, set by the ▾ button. You can also select the required interactive mode with the DC, AC, Transient, …VHDL commands of   the Interactive menu. You can start the interactive simulation with the Start command of the Interactive menu and stop it with the Stop command (The Start command will turn into Stop when the interactive simulation is started). Now the displays and indicators in your schematic will reflect whatever you do with the controls. In addition to displays, *TINA* has special multimedia components (light bulb, motor, LED, switch, etc.) which respond with light, motion and sound. Let's see a few examples.
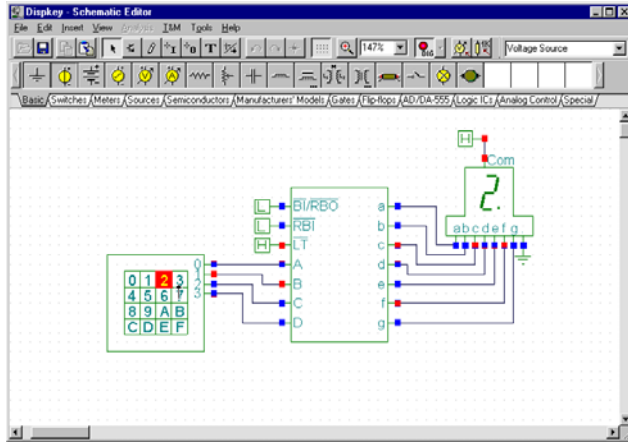
4.6.11.1 Digital Circuit with a Keypad

To try out the interactive mode, load the **DISPKEY.TSC** circuit from the *EXAMPLES\Multimedia* folder. The circuit is shown below. Select the *Digital* mode using the ▾ button, and then press the DIG button (the button will turn light green).

NOTE:

> *1) You can also select the Digital interactive mode with the Digital command of the Interactive menu. You can start the interactive simulation with the Start command of the Interactive menu and stop it with the Stop command.*

> *2) TINA can store the last Interactive mode in circuit files, so most likely the DIG mode is already set.*

Now you can play with the keypad and watch as the 7 segment display shows the setting of the keypad. If you have a soundcard on your PC, you will even hear the key clicks of the pad.

4.6.11.2 Light Switch with Thyristor

Open the Thyristor switch example, TSC circuit from the EXAMPLES folder and press the button button. You will see the following screen:
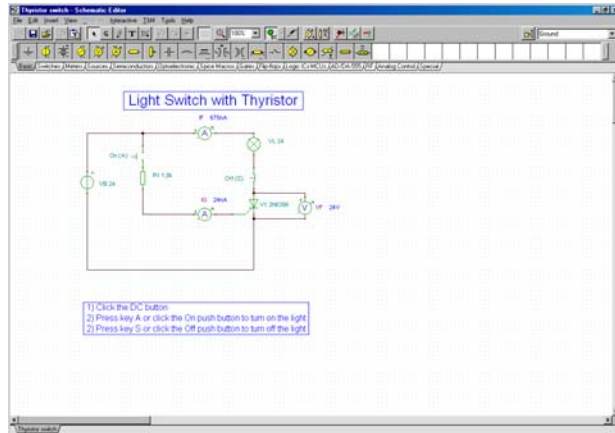
Press key A or click the On push button (Wait until the cursor turns into a vertical arrow) to turn on the light. The Thyristor will turn on and remain on even after the push button is released. So will the light. You can turn off both the Thyristor and the light bulb by pressing the key on the keyboard or clicking on the push button . In both states of the circuit, you will see the currents shown by the two ammeters.

4.6.11.3 Ladder Logic networks

Another version of a self holding circuit, this one based on ladder logic, can be found in the LADDERL.TSC circuit file in the EXAMPLES/Multimedia folder.

Initially the red LED will light. If you click on the START button (click when the cursor changes into a vertical arrow), OCR1 will close and stay closed (since the current flowing through OCR1 will keep magnetizing the relay coil CR). Now the green LED will light, OCR2 will open, and the red LED will turn off. If you now click on the STOP button, you will break the self holding circuit and the relay

DesignSoft

CR will release, the red LED will light again, and the green LED will turn off.

You can make it easier to operate switches if you "assign" them to "hotkeys" on the keyboard (your PC's keyboard). Double-click on a switch when the cursor has turned into a hand symbol. To assign a hotkey, select a letter or number on the list at the Hotkey field of the property dialog of the switch.
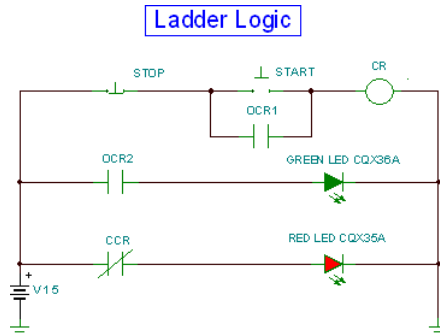
Ladder logic: Initial state or after clicking the STOP button.
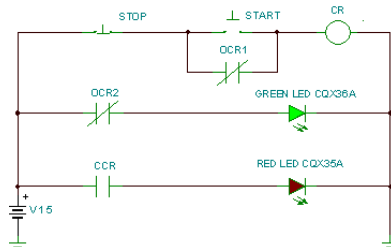
State after clicking the START button

4.6.11.4 HDL Circuits

A great feature of TINA is that you can not only test but also modify HDL circuits on the fly including, the HDL code itself. Let's see this with the example Calculator_ex_8.TSC in TINA's Examples\VHDL\Interactive folder.
This is a special calculator circuit controlled by the Opcode keypad.
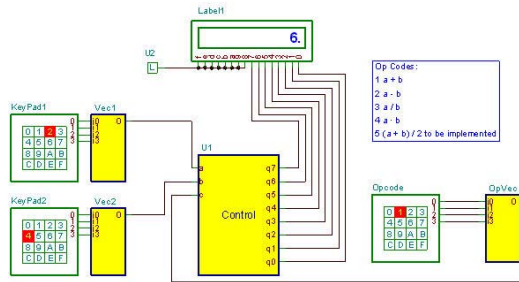
For the Operation codes 1, 2 , 3 and 4, it realizes a basic four func- tion calculator, complete with +, -, /, and * basic arithmetic opera- tions. Further operations can be added through modifying the VHDL code inside the Control unit. First press the button; as the Opcode is 1, you should see 4+2=6 on the LCD display. Try the other Opcodes with different settings on KeyPad1 and KeyPad2.

Now let's implement the operation to be assigned to Opcode 5. Double-click on the Control box and press Enter Macro. The VHDL code of the component will appear.

DesignSoft

GETTING STARTED
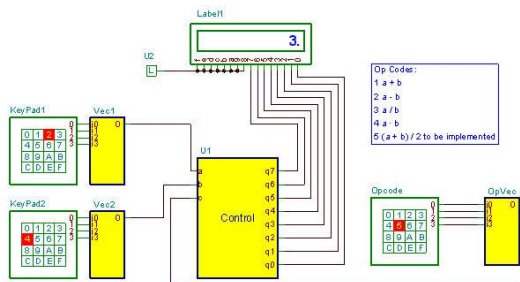
4.6.11.5 Microcontroller (MCU) Circuit

To test circuits with programmable devices requires special development software that permits a high degree of interactivity. This calls for debugging software that can test the code running in the device step-by-step.

You can see, modify, and debug the program running in any of the supported processors, and, of course, you can make and run your own code.

There are 4 ways of providing the program for microcontrollers in TINA.

You can:

1)   use the binary code and debug file made by any standard compiler (e.g., MPLAB for PICs),

2)   load your assembly code to run and debug directly in TINA using its built in assembler-debugger,

3)   write your MCU code in C, install a C compiler which generates the code for the MCU you want to simulate, (TINA will automatically integrate it into its C code debugger),

4)   or finally use the built in Flowchart editor in TINA to generate and debug the MCU code.

To load the code into the MCU, double-click on the schematic symbol.

One of the following dialogs will appear:
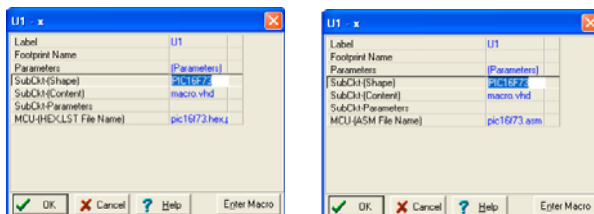
Click on the last MCU File name line and press the ⋯ to proceed. The following dialog will appear:
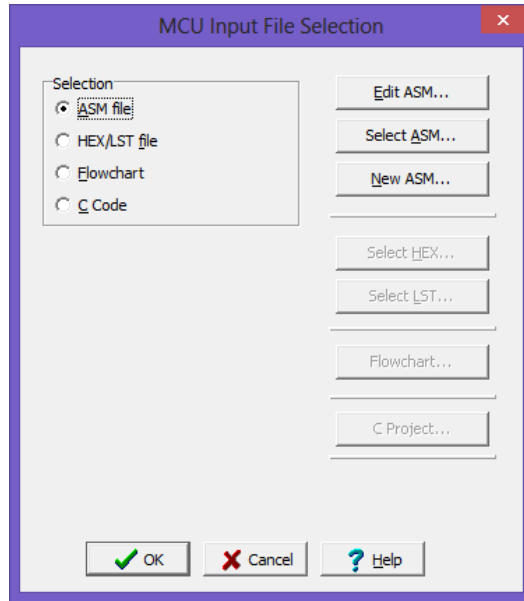
Here you can see and edit the ASM code in the MCU, select another ASM code file, or create a New ASM directly in the editor that will appear when you press the New ASM button.

If, however, you switch to the Use HEX/Lst file option, you can select the binary (HEX) file you want to run and the LST file to be used for debugging, as shown in the dialog below.

NOTE:

> *The HEX and LST files should be generated by an appropriate compiler (normally provided free by the MCU manufacturer. However, TINA has a built in compiler for all supported MCUs, so you can directly use your ASM source code.*

MCU Input File Selection

Selection
- ASM file
- HEX/LST file
- Flowchart
- C Code

Edit ASM...
Select ASM...
New ASM...

Select HEX...
Select LST...

Flowchart...

C Project...

✓ OK   ✗ Cancel   ? Help
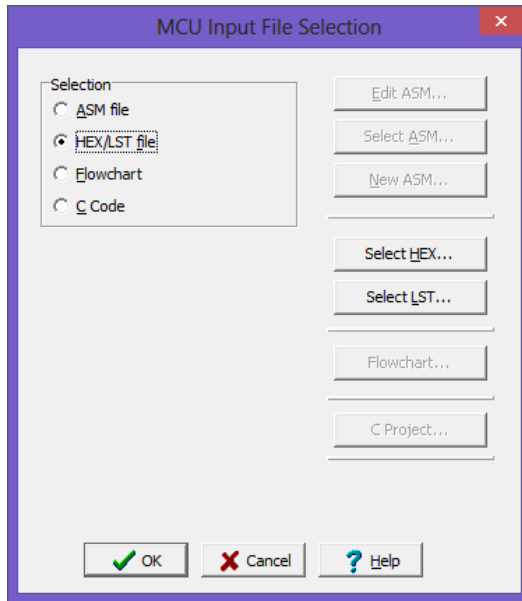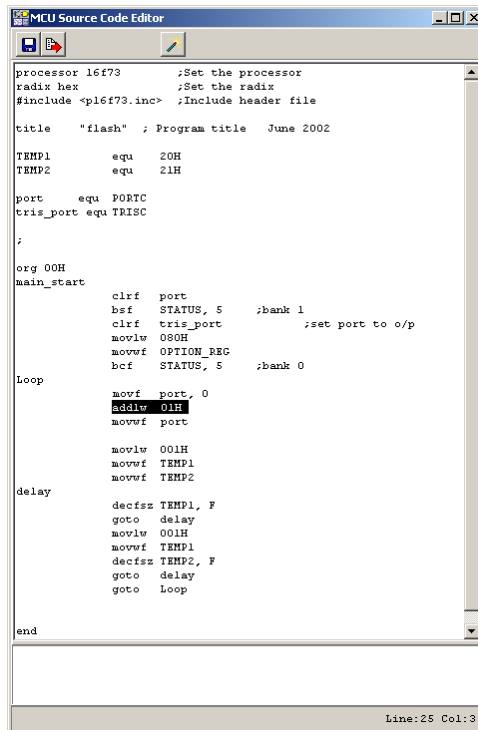
4.6.11.6 Using the ASM Debugger

Now let us run a microcontroller application and see how to test and modify its code. Load the PIC F lasher.TSC circuit from the Examples\Microcontrollers\PIC folder. The following schematic will appear with the 16F73 PIC microcontroller.

This circuit is simply counting forward one count at a time. Press the button to see how it works. The display should step forward one-by one.

Now let's release the button and modify the code to count by 2.

Double-click the MCU, and click on the button in the dialog below

DesignSoft
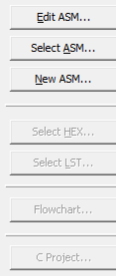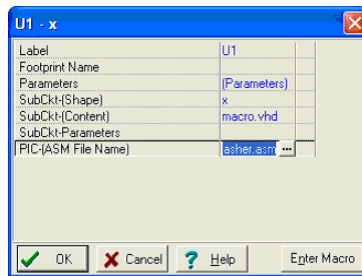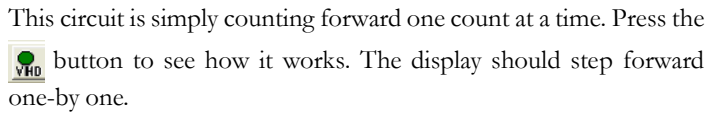
Press the Edit ASM button. The ASM code of the MCU will appear in the MCU Source code editor

Now lets make the following change in the code. Change the instruction (selected above) in line 25 (you can see the line number in the right bottom corner of the code editor window) from

```
        addlw 01H
```
to
```
        addlw 02H
```

Save the changed code by pressing the icon and close the open

MCU windows. If you press the button, now the increment will be 2!

Note that the changed code will be automatically saved in the TINA.TSC file.

DesignSoft

4.6.11.7 Example PIC Interrupt handling

Now let's see another application with some more interactivity. Load the PIC16F84_interrupt_rb4_rb7.TSC example from the Examples\Microcontrollers\PIC folder.
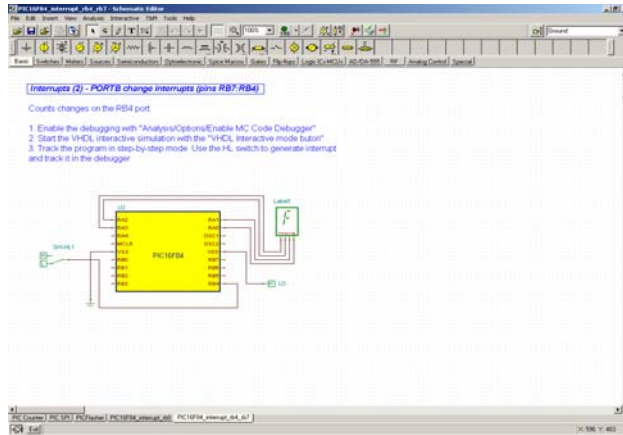
Press the button. At first glance, it appears that nothing is happening.

However, if you click on the SW-HL1 switch, the display will step forward by 1 each time that the switch changes from Low to High. This is realized with the interrupt handling capability of the PIC16F84. Now let's see the operation in more detail using the interactive ASM debugger.

To activate the debugger, select Option on the Analysis menu. Set the "Enable MCU Code debugger checkbox," as shown below in the Analysis Options dialog box.
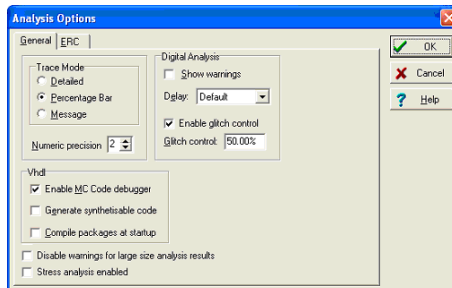
The MCU debugger will appear if you press the button:
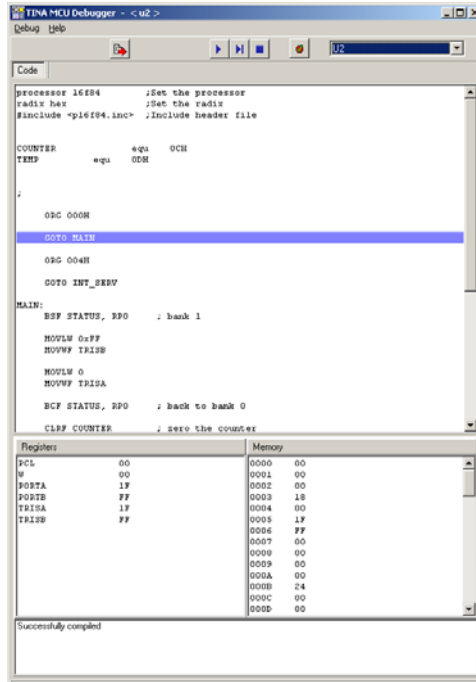
Here is a short description of the MCU debugger dialog.

On the top line there are the following controlling icons:

 Toggle breakpoint: Inserts or removes breakpoints in the selected line. Click on the line where you want to place or remove the breakpoint before clicking on the icon.

 Run the code in the debugger continuously. The lines being executed will be highlighted and the code is scrolled.

 Step forward. Step by step execution. Each time you press this button one command of the program is executed.

 Stop Halts program execution.

The Code window (below the control icons) displays the ASM code. The next actual command is highlighted with blue.

The actual content of the registers and memory locations of the MCU are shown in the lower part of the screen.

Let's follow the program execution step-by-step by pressing the Step forward button. After around 14 clicks, we get to the PT1: label,

where the program seems to be in an infinite loop.

```
PT1: INCF TEMP, F
GOTO PT1
```

Now click on the SW-HL1 switch and change it to High. (You should click when the cursor changes into an upward pointing arrow Î). Return to the Debugger and click the ▶ Step forward button twice. The program will recognize the interrupt and jump into the INT_SERV: label.

```
INT_SERV:
INCF COUNTER, F
MOVF COUNTER, 0
MOVWF PORTA
```

increment the COUNTER, and copy it to PORT A. The output will now be 1. After this, the program will return to the "infinite loop" at PT1.

4.6.11.8 Editing the ASM Code in the Debugger

Now let's see how to make a small change in the program using the debugger. Duplicate the INCF COUNTER, F statement using Copy and Paste like so:

```
INT_SERV:
INCF COUNTER, F
INCF COUNTER, F
MOVF COUNTER, 0
MOVWF PORTA
```

Now if you press the ▶ the program will ask:

DesignSoft
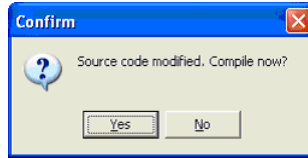
Press Yes and then press the ▶| button again. Now the increment will be 2 at each Low-High change of the switch.

You can also check the circuit in the Debugger's continuous Running mode by pressing the ▶ button. Even though the debugger will run fast, you can still see the "infinite cycle" and the jump to the Interrupt server routine (INT_SERV:) when you change the switch.

4.6.11.9 Making a Breakpoint in ASM

It is often essentially impossible to get to a certain place in the program since you'd have to single step a thousand times (if the program ever steps there in the first place). To get the program to run to a particular statement and halt there, you can tag the statement as a so called breakpoint. Now run the program in the Debugger's continuous mode using the ▶ Run command and the program will stop at the marked space before execution of the marked command. To demonstrate this, click on the increment statement in our interrupt service routine after the INT_SERV: label and press the

⬤ Toggle break button.

Now press the ▶ Run button. The program starts to run and falls into the "infinite loop."

Even though you have set a breakpoint, the code will not stop since it does not pass the breakpoint. However, when you change the switch from Low to High the program will stop at the

```
INT_SERV:
INCF COUNTER, F
```

statement. Now you can resume execution either step by step ▶| or

with the ▶ Run command again.

4.6.11.10 Programming MCUs using C

Writing assembly code is not easy, and the programmers of desktop computers have turned to a high level programming language. The high level languages are increasing in popularity, and C is perhaps the most used and most useful language for MCU programming. There are many C compilers available on the market, many of them are free or have a free version. You have to install a C compiler which generates the code for the MCU you want to simulate, and then TINA will automatically integrate it into its C code debugger. Here are the C compilers compatible with TINA:

(1) **For PIC:** Install Microchip PIC compilers (xc8, xc16, xc32) from http://www.microchip.com

(2) **For AVR:** Install Atmel Studio from https://www.microchip.com/mplab/avr-support/avr-and-sam-downloads-archive

(3) **For 8051:** Install SDCC and GPUTILS SDCC: http://sdcc.sourceforg e.net/, GPUTILS: http://gputils.sourceforge.net/

(4) **For ARM (non-cortex):** Install the yagarto-tools-*, yagarto-bu-* packages from http://www.yagarto.de

(5 **For ARM Cortex**: for XMC install DAVE, for TI Tiva install Code Composer Studio, for STM32 install System Workbench for STM32

Do not install the AVR, ARM toolchain in a directory which contains spaces like "C:/Program Files/"

You may have to restart your computer after installing these tools.

You can also debug the C-code and execute step-by-step for most MCUs in TINA. However, this does not work for the 8051.

Now let's see how to load and run a C code in an MCU in TINA. We assume that the Winavr compiler has already been installed on your computer.

Create a new circuit file in TINA with File/New

Search for and place the ATTiny26 MCU with the Find component tool at the top right corner of the screen. You can also do the same using the Logic ICs MCUs tab of the component toolbar and selecting MCU, then AVR as Manufacturer.

Double Click the MCU, and the property window of the MCU will appear. Click the "C code" line and then the …button. "The MCU input file selection" dialog will appear. Switch on the C Code radio button. You will see the following dialog.
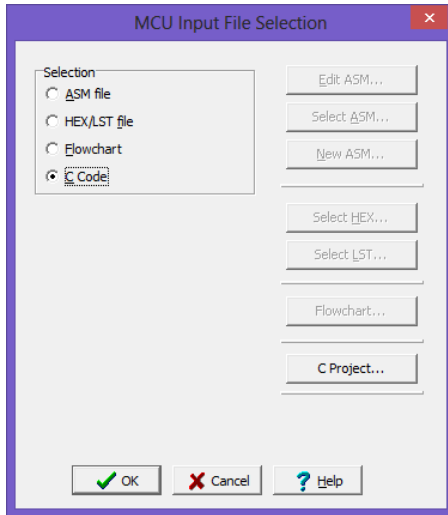
Press the C Project button. The following dialog will appear:
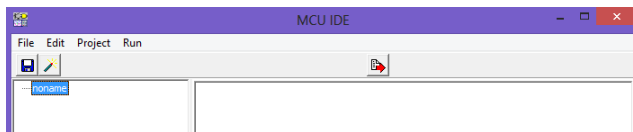
Right-click on the blue noname label on the top left corner of the screen.

A pop- up dialog will appear, click on "Add existing file."

An Open dialog will appear: navigate to the EXAMPLES\Microcontroller\C Compiler\AVR folder (or where your own file resides) and Open AVRflasher.C The following dialog will appear:
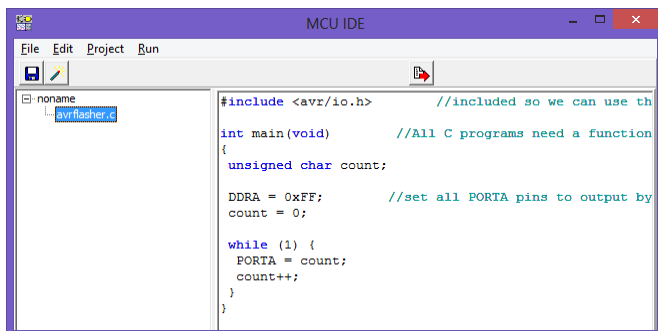
Click on *Project/Compiler Options* and set AVR Compiler to Atmel Studio.

Press the [icon] make project and then [icon] Save Project button, and exit by clicking the top right corner of the dialog.

Press OK in the MCU Input File Selection dialog and in the ATTINY26 property dialog.

Select the Digital on the Interactive menu or DIG with the narrow "half" button next to the Interactive mode On/Off button.

Now if you press the DIG button, your C code will start to run. You can see that from the changing red and blue logic states on the pins of the MCU. Note that the simulation of MCUs in TINA works even without a power supply, to simplify the schematics, but of course you should still connect the power supply and all necessary pins when you design a PCB.

Now add a power supply, voltage pin and an LED to the circuit as shown on the picture below.
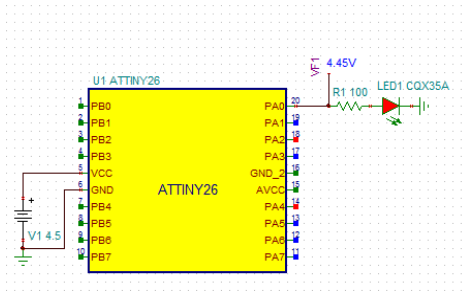
If you run the simulation, the LED will light when the logic level on PA0 is high and the voltage pin will show the analog voltage. Note that this voltage follows the voltage of the power supply. The other pins are handled with logic values according to the rules of Mixed Mode Simulation. This makes a big difference compared with external simulation of MCUs as in TINA. You can also see all the voltages and displays of your complete circuit during the simulation. You can do this even step-by-step with the help of the TINA C debugger.

ARM Cortex MCUs

For ARM Cortex MCUs (XMC, TI Tiva, Stm32) you can use Eclipse based development environment. For TI Tiva use TI Code Composer Studio, for XMC use DAVE, for Stm32 use System Workbench for STM32. Now let's see how to load and run a C code in an Stm32 MCU in TINA.

1.   Install System Workbench for STM32 (free software)
2.   Start System Workbench for STM32 and create your project
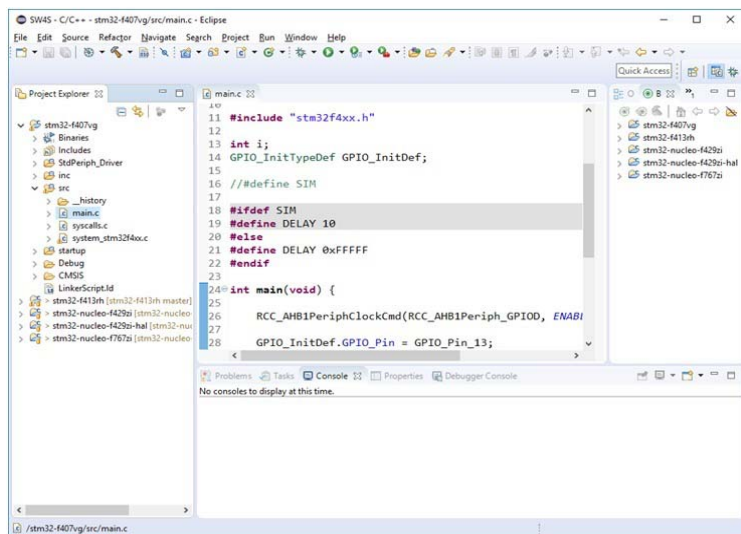
Change some settings in the project.

System Workbench for STM32 (SW4S)

C++ Build Settings
- Build steps/Post build steps/Command: add arm-none-eabi-objcopy -O ihex "${BuildArtifactFileBaseName}.elf" "${BuildArtifactFileBaseName}.hex";
- Tool settings/Preprocessor: define HSE_VALUE (for example: HSE_VALUE=8000000)

Build your project.

3.   In TINA select the MCU from the toolbar and add it to the schematic. Double click on the MCU, click on ASM-code/C-code (select C code).
The MCU IDE will appear. Select File/Refresh eclipse project.
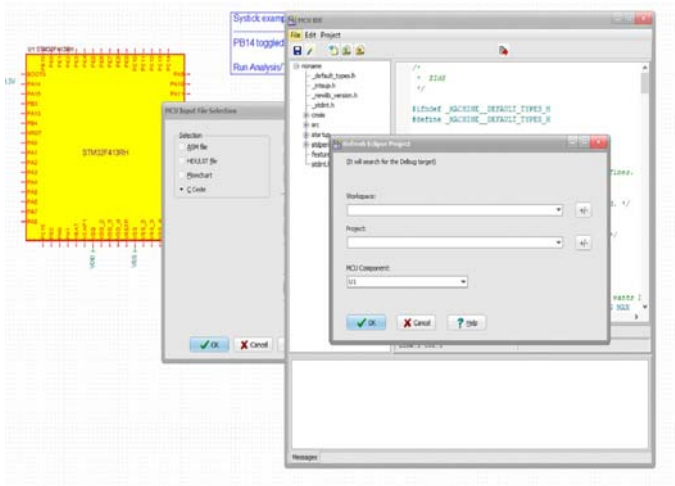
Fill the Workspace, Project fields in the Refresh Eclipse Project dialog (+/- button). The Workspace is the System Workbench workspace, for example *d:\Work\SW4S*. Project is the project name, for example *stm32-f413rh*.

Press OK. When the code assigned successfully to the MCU you will see the following dialog.

Press OK. Press *Save Project* in the MCU IDE. Close MCU IDE. Press OK, OK.

Now you can run a simulation.

4.6.11.11 Debugging C code in MCUs

Just as with ASM and HEX code, you can follow the execution of a C program and even watch the values of the required variables.

To demonstrate this, let's open the "check_prime.TSC" file in the Microcontrollers\C compiler\AVR folder. The following circuit will appear:
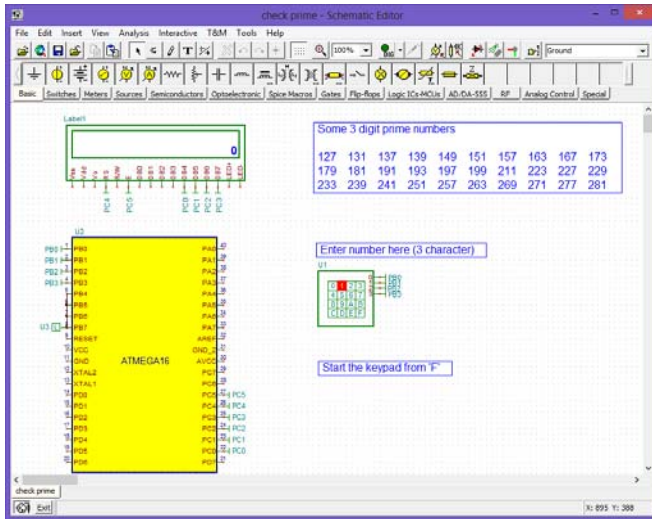
To test this circuit, press the DIG button and enter a 3 digit number (each digit must be different). The display will show "Prime number" or "Not prime."

Now to debug this C program, release the DIG button and then click the "Enable MCU Code debugger" in the Analysis menu and then press the DIG button again.

The C code debugger window (MCU IDE) will appear.

GETTING STARTED
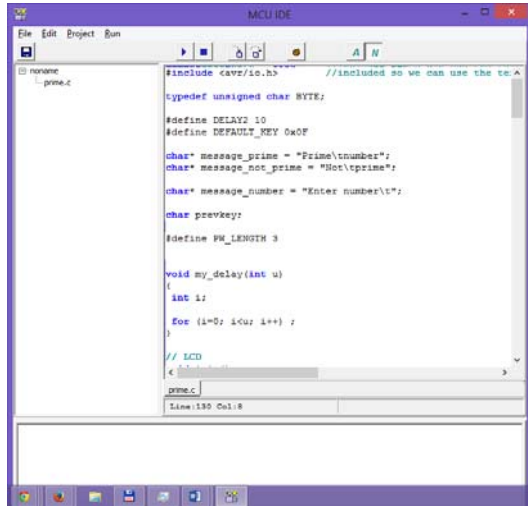
Scroll down the C code until the following function appears (line 128)

```
int check_prime(int a)
{
int c,mod;
for ( c = 2 ; c <= a - 1 ; c++ )
{
mod = a%c;
if (  mod == 0 )
return 0;
}
if ( c == a )
return 1;
}
```

Right click the variable "c" and click "Add Watch at cursor". Click the "Watch List" tab at the bottom of the MCU IDE window. The variable c should be on the list already. Add variables "a" and "mod" the same way.

Now click the if (mod == 0) line and press the ![icon] Toggle Breakpoint (Add Breakpoint) button at the top line of the MCU IDE window. The line with the breakpoint will turn red.
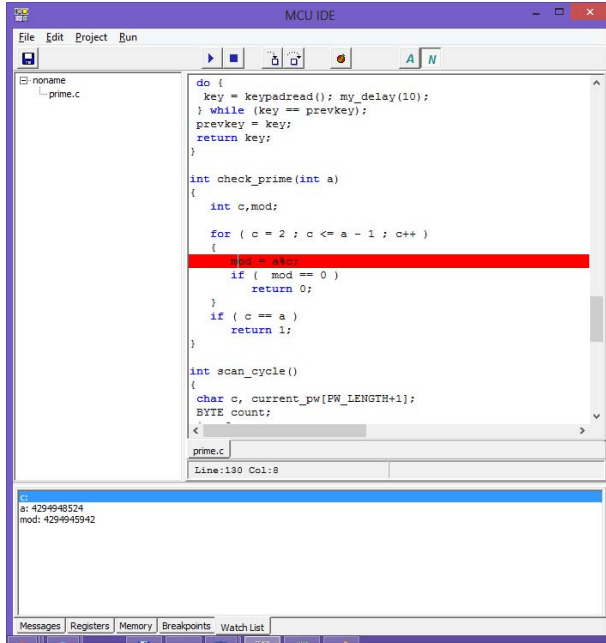
Press the ![icon] tart button. The progam will start to run. You will not see changes in the debugger window unless you also press the the „A" animate button at the top of the screen, which will show the active instructions and scroll the screen when necessary. However if you enter a 3 digit number on the small keyboard on the screen, the progam will stop at the if (mod == 0); line and you will see the a,c,mod variables at the breakpoint.

After the stop at the breakpoint, you can continue the execution step by step by pressing the ![icon] Step button for each step or run continuously by pressing the ![icon] Start button again.

DesignSoft

4.6.11.12 Serial Monitor

Serial monitor allows you to both send messages to the simulated MCU and receive messages from the simulated MCU. The serial monitor window communicates with the simulated MCU's using the UART peripheral. Baud rate, data bits, stop bit count, parity are automatically detected.

The Serial Monitor Window
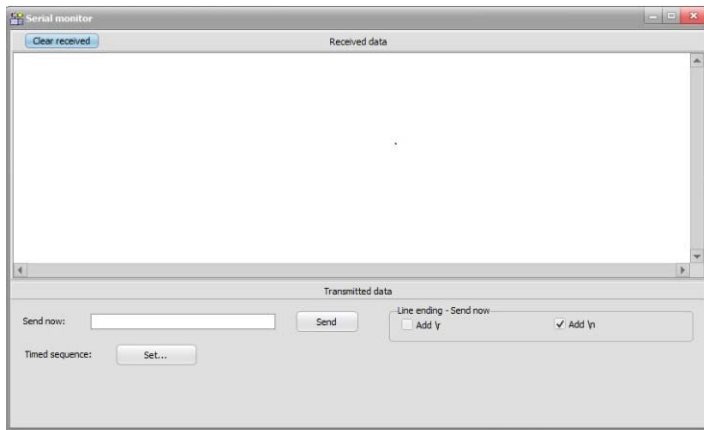
Clear received: you can clear the Serial Monitor window
Send now: when you press Enter or press the the Send button the text in the edit box will be transmitted to the MCU while the simulation is running
Line ending: you can specify here the line ending format. The line ending character will be added to the transmitted text when you use Send now.
Timed sequence: you can specify a timed sequence here. See the example in the Timed sequence window.

DesignSoft

Example 1: receiving messages

Let's open the tm4c uart terminal.tsc from the
Examples\Microcontrollers\TI Tiva folder.
Select Tools/Serial monitor and press the TR button. The MCU will
send some test messages. The received messages will appear in the
Serial Monitor window.
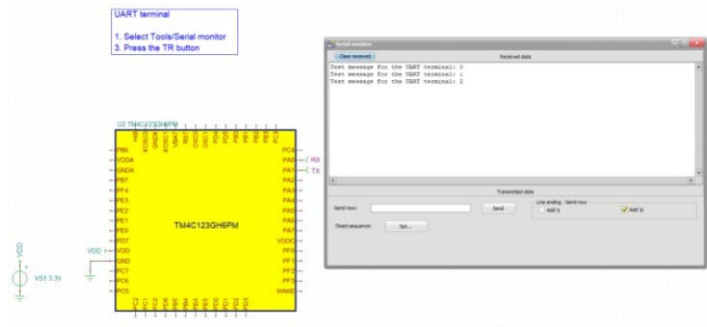
The UART is programmed using the TivaWare Peripheral Driver
Library.

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
#include "uartstdio.h"

int main(void)
{

    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_16MHZ);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

    GPIOPinConfigure(GPIO_PA0_U0RX);
    GPIOPinConfigure(GPIO_PA1_U0TX);
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

    UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,
        (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE));

    UARTprintf("Test message for the UART terminal: 0\r\n");
    UARTprintf("Test message for the UART terminal: 1\r\n");
    UARTprintf("Test message for the UART terminal: 2\r\n");

    while (1) ;

}
```

Example 2: sending commands to the MCU

Let's open the tm4c uart command.tsc from the Examples\Microcontrollers\TI Tiva folder
Select Tools/Serial monitor, check the status of the Line ending field: 'Add \r' must be turned on 'Add \n' must be turned off. Press the TR button. Wait for the 'Ready' message in the Serial monitor window.

Type LED0 in the Send now field: the LED0 led will be turned on. Type LED0 again the LED0 led will be turned off.
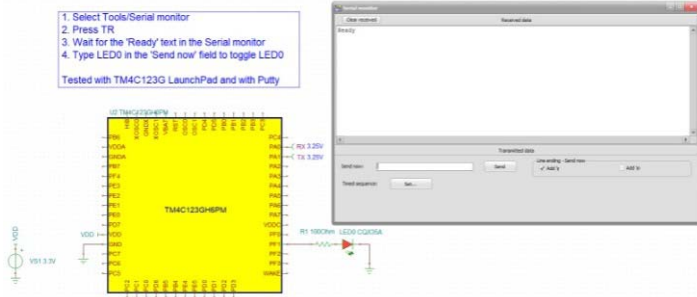
4.6.12 Using the Flowchart Editor and Debugger in TINA

Writing MCU assembly code is often a hard and tedious task. You can simplify software development and gain more time to design the electronics hardware if, instead of manual coding, you use TINA's Flowchart editor and debugger to generate and debug the MCU code. This easy-to-use tool works with symbols and flow control lines with which you can represent the algorithm you want. The Flowchart editor is opened via a MCU device as described below. You can find detailed descriptions of the flowchart symbols and their parameters under the Help menu of the Flowchart editor.

4.6.12.1 Flowchart Editor

In the following example, we will create a flowchart to control a MCU embedded in a simple circuit. The flowchart adds two numbers that are read from two ports of the PIC16F73 microcontroller. (You can find the complete circuit under EXAMPLES\ Microcontrollers\PIC*PIC Adder.TSC* in TINA).
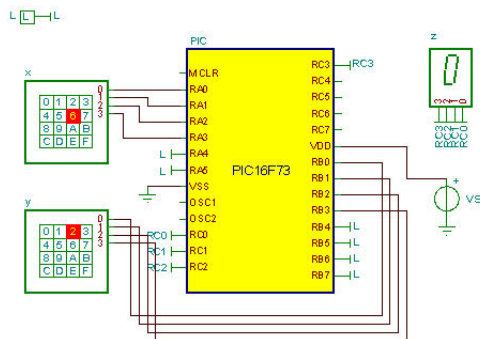
First, select the MCU from the component toolbar and insert it into the schematic editor. The MCU components are located under the *Logic IC-s-MCUs* tab. Click the MCU icon on the toolbar and select PIC16 in the Manufacturer line of the MCU dialog. The list of PIC16 MCUs will appear. Select PIC16F73 and click *OK*. The selected MCU will be placed in the schematic editor.

DesignSoft
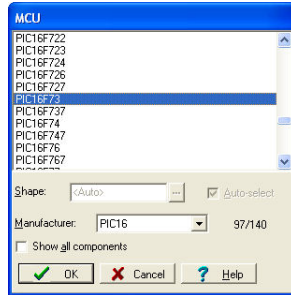
Now double-click on the PIC16 component in the editor, click the *MCU-(ASM File Name)* field, and then the ⋯ symbol. The *MCU Input File Selection* dialog appears.
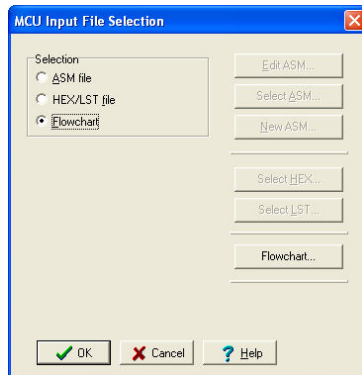
In this dialog, select the *Flowchart* mode under Selection on the left.

Click on the *Flowchart* button on the right, and the TINA Flowchart Editor opens.

With the *Flowchart* tab selected, the editor window is divided into two areas. On the left there is a toolbar of the symbols that you can place in the editor area on the right.

DesignSoft
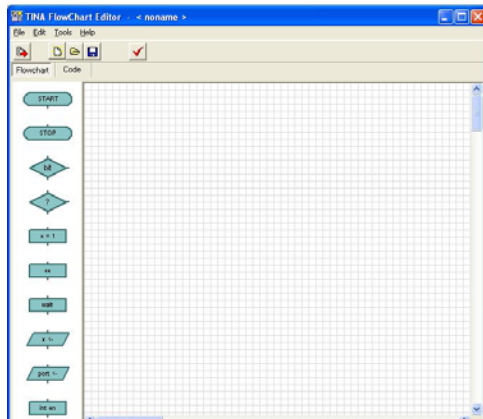
Select the *START* symbol from the toolbar by clicking on it. The START symbol will be attached to the cursor. Move it to the middle of the editor space and click with the mouse to place it.

Next we will read PORTA and PORTB ports into the variables x and y, and add them, write the result into PORTC. Flowchart execution will loop back to Start and continue running until stopped.

To read in the contents of PORTA port, click the symbol (*Read Input*) on the toolbar and move it into the editor space. Double click on this component to select the port you want to use for reading (*Source port*) and to set a variable name (*Target variable*). Select *PORTA* for *Source port* and enter *x* for the *Target variable*.

Now we must read in the next data by reading from another port, PORTB. This is very similar to the previous step. Insert another Read Input symbol, set *y* for *Target variable* and *PORTB* for *Source port*.
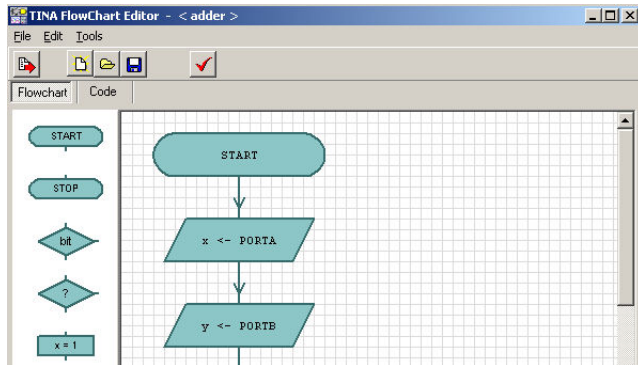
Next we will add the two numbers in *x* and *y* and write the result to PORTC. Here is how to do this.

First save the contents of *x* to a temporary variable, named *z*. To do this, select the *Set variable* component from the toolbar, set *Target variable* to *z*, and then *Value and variable* to *x*.

Next add *z* and *y*, by using the *Change variable* component Place the *Change variable* component below the previous *Set variable* component, set *Target variable* to *z*, *operator* to +, *Value or variable* to *y*.

To write the result to PORTC, add an *Output* component symbol, Select *Target port* to PORTC and *Value or variable* to *z*.

Finally, let's connect the flowchart symbols with flow lines, indicating the direction of the flow, by moving downwards until you reach the bottom PortC <-z *Output* symbol.

First, connect the START and Read Input symbols. To connect the two symbols, move the mouse above the end of the *connection line* out of the START symbol. The *connection point* is at the end of the connection line, and is marked by a small rectangle visible when the mouse is at the right position. When the small rectangle appears, press and hold the left mouse button, and drag the connection line until you reach the connection point of the other symbol.

Connect all other symbols in a similar manner.

When you reach the lowest symbol, connect the lower connection point of this symbol with the flow line between the START and x <. PORTA component.

If the symbol placement and connection are correct, the flowchart will look like this.
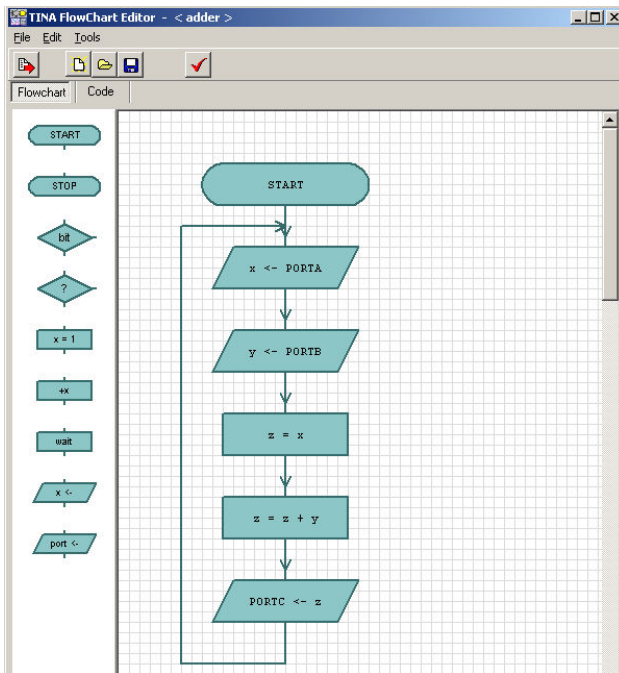
To make a formal check of the flowchart (e.g., see if all symbols are connected) press the ✔ button. To view the generated code, press the *Code* tab.
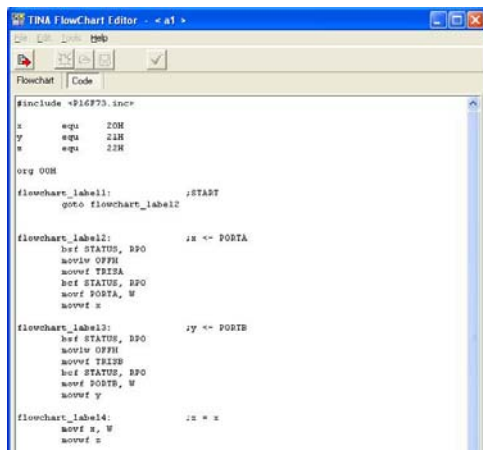
To save the flowchart in the MCU macro, press ![icon] on the toolbar, then press *OK* twice (on the *MCU Input file selection* and on the MCU property box dialogs) to go back to the schematic editor.

You can also save flowcharts in .tfc files with the Save and Save As… commands of the Flowchart editor. You can then Open them and associate them with other MCUs.

4.6.12.2 Flowchart Debugger

TINA automatically produces the assembly code required for the simulation from the flowchart.

Let's test and debug the previously completed flowchart. (You can open the complete circuit from (EXAMPLES\Microcontrollers\PIC*PIC Adder.TSC*)).

Set the *Analysis/Enable MCU code debugger* switch in the Analysis menu, then press the ![icon] button on the TINA toolbar. The MCU debugger appears.

GETTING STARTED

DesignSoft
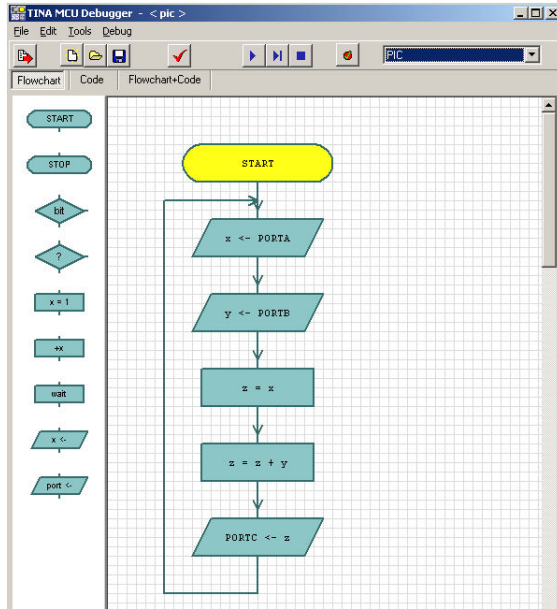
You can run the program continuously by pressing the ▶ Run button, step-by-step by pressing the ▶| Step Forward button, or stop the program by pressing the ■ Stop button. The debugger will show the active flowchart component by setting its background color to yellow.

There are three important tabs at the top-left corner of the debugger that establish the view of the source. If you select the *Flowchart* tab, you can see and debug via the Flowchart. If you select the *Flowchart+Code* tab, TINA will display both the flowchart and the assembly code. You can, in this view, place breakpoints both in then flowchart and in the assembly code. If you select the last mode, *Code,* you can debug using traditional assembly language debugging. See section 4.6.9.9 in this manual.

Note that in order to synchronize the flowchart and the assembly code, and to make the code more readable, TINA inserts additional labels and comments into the code; for example:

Flowchart label2: ;x <- PORTA

These labels do not alter the performance or the operational logic of the code.

GETTING STARTED

DesignSoft

Breakpoints are used to halt code execution at user-specified points, permitting the examination of registers and parameters. TINA offers several ways to insert and remove breakpoints.

Insert breakpoints into the flowchart by clicking on a flowchart symbol and pressing the ⬤ Toggle Breakpoint button.

Or you can place breakpoints into the code directly via the code window. Select an instruction line and press the ⬤ button .You can remove the breakpoint with the same button when the breakpoint is selected.

When breakpoints are set, the program will stop at breakpoints *before* executing the instruction under the breakpoint. You can execute the instruction and continue the program by clicking the ▶ *Run* or the ▶▏ *Step Forward* buttons.

4.6.13 Testing your Circuit with Simulated and Real Time Instruments

TINA lets you test and tune your circuit not only with the generators and analysis windows that you have used so far, but also with simulated and real-time measurements. Using the T&M menu you can place virtual instruments on screen which will automatically replace the generator and analysis windows. You can control the settings of these instruments and immediately see the results, just as you would in a real lab. *TINA* normally simulates measurements with its analysis engine, but if you have *TINA's* supplementary hardware, you can simply switch to the Real Measurement mode (using Option of the T&M menu). Now you can work with the same on-screen instruments and settings and you will be making real measurements on a real circuit.

GETTING STARTED
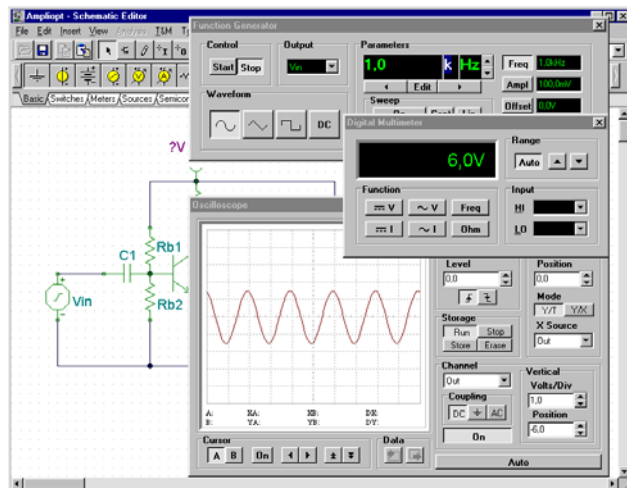
To start with virtual measurements, load the circuit **AMPLIOPT.TSC** from the examples directory. Select the T&M menu and place a Multimeter, Function Generator, and an Oscilloscope on screen. Press Run in the Oscilloscope panel. A distorted sine curve will appear on the oscilloscope screen. Click on the multimeter = button. The multimeter will show only about 0.7 V volts at the collector (Out) - this is the reason for the distortion. Now double-click on the Rb1 resistor. The dialog box of resistor properties will appear. Click on the Resistance field and then change the value with the arrows on the right hand side of the dialog box until the multimeter shows about 6V.

You can change the step size of the buttons by entering it into the edit box under the *down* arrow. You can also define a hotkey to the up and down arrow by selecting it from the upper and lower listboxes. Note that while the interactive mode is On, and once a hotkey is defined, the value of the resistor can be changed directly by pressing the appropriate key without opening the property dialog. You can assign hotkeys to most component values in TINA, including switches. To avoid accidental changes, the hotkeys for component values will only work while TINA is in the interactive mode. Switch positions, however, can be changed before activating the interactive mode in order to set their initial position. Once the collector voltage reaches 6V, close the property editor dialog box (if it is still open) and press Run on the oscilloscope.

GETTING STARTED

Set the vertical position to -6 V and use the horizontal and vertical settings to scale the curve for best appearance. The distortion will no longer be visible.

Press the *Ampl.*-button on the generator. The last amplitude value will appear in the large numeric display field of the generator. Use the vertical arrows beside the display to change the amplitude. You will see the sine wave become distorted again as you increase the amplitude, with the maximum input at about 500mV. Now change the waveform from sinusoidal to triangle and then to square wave. Vary the frequency of the function generator to explore the frequency domain over which the circuit performance is acceptable.

NOTE:

> *The virtual instruments under the T&M menu are not to be confused with the virtual instrument components on the Meters component toolbar. Some of the virtual instrument components are used in the interactive mode of the program, discussed in the next section. They are also used to assign outputs for the various analysis modes under the Analysis menu. The Oscilloscope and Signal Analyzer virtual instrument components have a small screen and their purpose main function is to be used with our 3D circuit analyzer program, EDISON.*

4.6.14 Using the Design Tool in TINA

TINA's Design Tool works with the design equations of your circuit to ensure that the specified inputs result in the specified output response. The tool requires of you a statement of inputs and outputs and the relationships among the component values. The tool offers you a solution engine that you can use to solve repetitively and accurately for various scenarios. The calculated component values are automatically set in place in the companion TINA schematic and you can check the result by simulation.

As an example, this tool can calculate feedback or other resistor and capacitor values of an amplifier in order to achieve a certain gain and bandwidth, and it can calculate component parameters of power supply circuits to meet output voltage and ripple requirements.

The TINA Design Tool promotes good documentation by storing the design procedure together with the circuit.

It is also very useful for semiconductor and other electronics component manufacturers to provide application circuits along with the design procedure.

Let's demonstrate the use of this tool through a simple operational amplifier example.

Open the Invert Gain OPA350 Test Circuit Design.TSC circuit from the Examples\Design Tool folder of TINA. In the TINA Schematic Editor the following circuit will appear:

With the Design Tool we will set Rf and Vref to achieve the specified Gain and DC output voltage.

Now invoke the Design Tool from the Tools menu of TINA. The following dialog will appear.

GETTING STARTED

Here you can specify the Gain (Vout/Vin), the DC output voltage (Vout_DC) and some other parameters. The simple design procedure will calculate Rg and Vref. The allowed minimum (Min) and maximum (Max) parameters are also shown. To enable or disable modification of Min and Max select Options from the Settings menu of the Design Tool.

Note that in the Design Tool dialog you can also refer to component parameter names. For example in the Vout_DC line the maximum value is set as V1-200m, telling that the DC output voltage must be at least by 200mV less than the V1 supply voltage of the IC.

If you just want to Run the design procedure press the ▶ button or the F9 key or use the Run command in the menu of the tool. If you run TINA in interactive mode you can immediately see the effect of the changes made by the Design Tool.

To see the design procedure itself, press the More button in the dialog. The code of the design procedure, written in TINA's Interpreter, will appear.

Note that the code part also shows the calculated parameter values according to the last stored calculation (Now Rf=[10.011k], Vref=[227.0455m]).

Now let's change the Gain input parameter to -1, Vout_DC to 3V and run the procedure by clicking Run in the menu or pressing the green ▶ button or F9 on the keyboard. In the code part we will see:

```
A:= 10^(Aol/20)
Rg:= Rscale
Rf:=-Gain*Rg*(1 + 1/A)/(1+Gain/A)
Rf=[1.0002k]
Vref:= Vout_DC/(1+Rf/Rg)
Vref=[1.4998]
```

and the new values will immediately appear in the schematic editor, drawn in brown color. Press the DC button to display the DC output voltage:

Now run an AC Transfer analysis, the Bode diagram will appear.
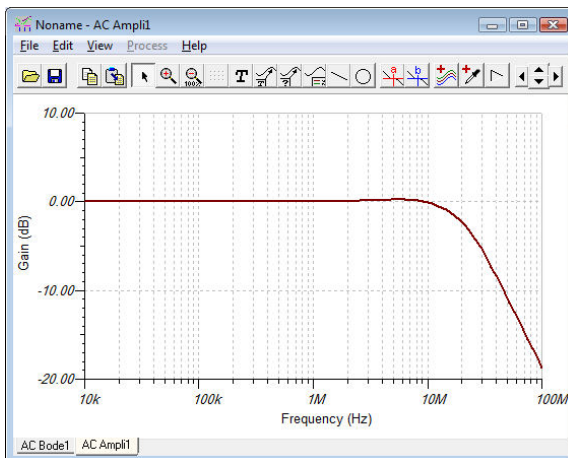
The small frequency Gain is 0dB which complies with the specified Vout/Vin=-1 value.

You can find more complex examples in the Design Tool folder of TINA.

You can make your own design procedure in any TINA circuits and save it together with the circuit itself.

For more information on the use and the controls see the on-line help in TINA by pressing the Help button on the tool and also the Advanced Topics Manual of TINA, available on-line at www.tina.com under Documentation.

4.6.15 Design Tool vs. Optimization in TINA

There are cases when writing a design procedure is not obvious or needs iteration or simply we do not have the time to implement it. In this case you may use the Optimization tool in TINA to determine the required parameter numerically in order to meet predefined circuit responses: voltage, current, power, gain, etc. You can learn more and find examples of Optimization in the Advanced Topics Manual of TINA available on-line at www.tina.com under Documentation.

NOTE:

> *Optimization for multiple parameters and target values is available in the Industrial version of TINA only.*

Generally speaking, although optimization is a very powerful tool, it is better to use a design procedure, if available because numerical optimization might need significant calculation time and it does not guaranty physically realistic results. But it is a very good tool to refine the results provided by a design procedure or tune already working circuits.

4.6.16 Live 3D Breadboard

Using the Live 3D Breadboard tool in TINA, you can *automatically* build a life-like 3D picture of a solderless breadboard (sometimes called a "whiteboard"). When you run TINA in interactive mode, components like switches, LEDs, instruments, etc. become "live" and will work on the virtual breadboard just as in reality. You can use this capability of TINA to prepare and document lab experiments. Beware, though, of the relatively high capacitance that exists between rows of adjacent pins. This comes about from the "parallel plates" that descend 3 to 5mm from the white board. For high frequency circuits (above 100kHz or so), this capacitance can lead to unexpected performance.

You can either assemble the circuit step-by-step or by generating the whole circuit on the breadboard. Pick up and move parts on the breadboard using the mouse, and TINA will automatically rearrange the wiring while retaining connectivity. In the same fashion, you can select and move wires for clearer appearance. Note that you cannot change the endpoints of a wire this way–wiring integrity is preserved.

The breadboard tool is mostly intended for educational purposes to prepare laboratory experiments in a safe 3D environment. You can also use this breadboard to guide you in actually wiring a physical breadboard for lab verification.

Let us demonstrate the use of Live 3D Breadboard though a few examples.

Boot TINA and bring in an unpopulated breadboard. Starting at the View menu, select Live 3D Breadboard and New. The 3D Viewer window will appear with an empty breadboard, in TINA's main window. This window is always on top so you can see both the schematic and the breadboard at the same time.

Place a Voltage Source, a Resistor, and a Voltmeter (found on the Basic toolbar), and an LED (from the Semiconductor toolbar), in the Schematic Editor window. The 3D version of these parts will be automatically appear in the 3D Viewer window.
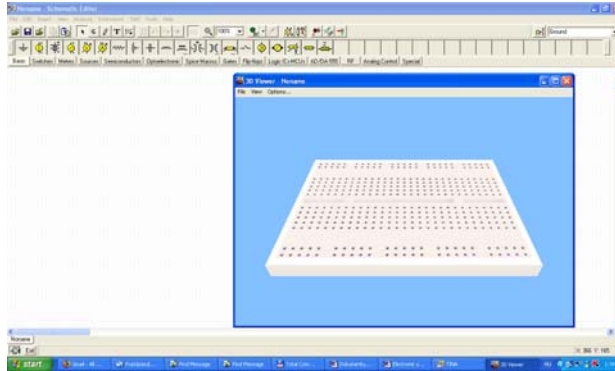
Connect the parts on the schematic with wires. TINA will place the corresponding wires on the breadboard automatically. Now save this circuit as LED Circuit.TSC using the File/Save command of the Schematic Editor.
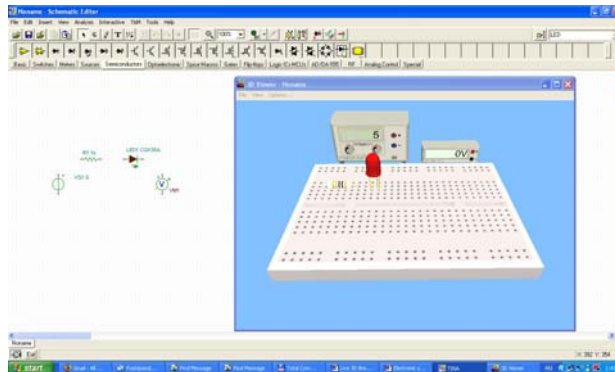
Press the DC Interactive analysis button or select Start from the Interactive menu. The LED on the breadboard will light and the Voltmeter will show the LED voltage. To get a better view, drag the Voltmeter to the foreground: click and hold down the left mouse button and move the mouse.

DesignSoft
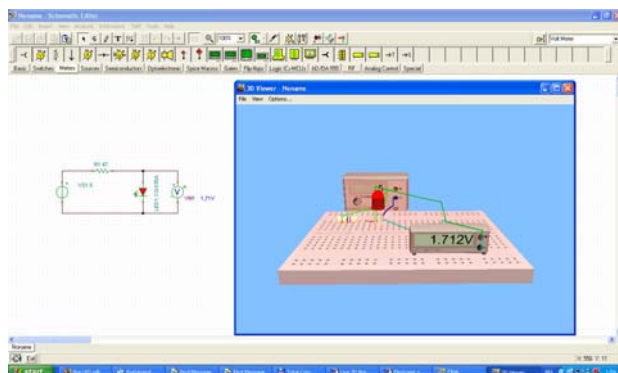
While in the interactive mode, you can modify several parameters and see the resulting changes on the virtual breadboard. For example, you can change the generator output voltage by moving the cursor above the knobs of the generator and turning the mouse scroll wheel. Or another technique is to hold down the left mouse button and move the cursor around the knobs. Yet another way to adjust a parameter is to move the cursor over the knob and right click with the mouse.

A parameter dialog opens for the left knob of the generator, and you enter a value in the data box. The dialog box looks like this:
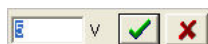
After you have entered a value, either press the ✔ button to confirm the change, or press the ✘ button to cancel.

When you are satisfied with the breadboard, save the 3D arrangement via the Save command from the File menu of the 3D View window. It will save the 3D view under the same name as the schematic, but with the .3DV extension.

Note: If you are dealing with a new circuit, we recommend that you name and save it right after choosing New file; otherwise the 3D view will be saved as Noname.3DV.

To open this example at a later time (with any changes you may have saved), load the file LED Circuit.TSC. Select 3D live breadboard from the View menu and press Continue.

You can find additional examples in the Breadboard folder EXAMPLES\3D Breadboard. Open the file 555_AST_BB.TSC and explore its more complex animation and interactivity. This screen will appear:

DesignSoft
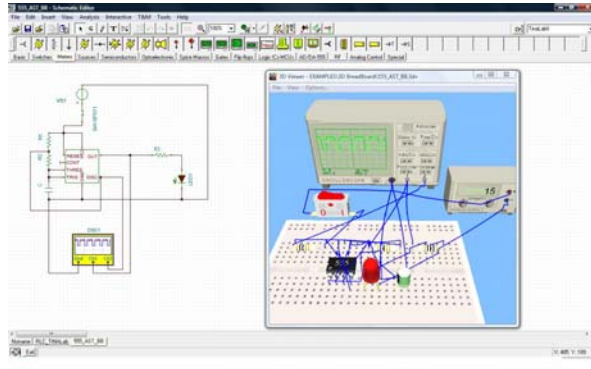
The file includes a screenshot of the circuit's 3D breadboard. To start the animation, open the working window, select Live 3D breadboard, and click on Continue (in the View menu).

Begin the live simulation (press the TR button on the toolbar or select Start from the Interactive menu). The waveform will appear on both the oscilloscope in the TINA schematic editor, and on the 3D oscilloscope in the 3D window. Once the 555 is active, the LED will start flashing. Note that, just as you could change a parameter of the generator in the previous example, here, too you can change the settings of the oscilloscope in the 3D window. Edit the settings using the mouse as described above.

To switch simulation on and off, click on either the switch on the 3D window or in the schematic editor.

Note: there is another way to create experiments in which virtual instruments are used; namely, you can connect virtual instruments to the 3D view of PCBs designed in TINA. Here's how you do this. With the same 555 timer file open, press the PCB Design button on the toolbar or select PCB Design from the Tools menu. The PCB design of the circuit will appear. Press the 3D View button on the PCB Designer's toolbar or select 3D View from the View menu. The 3D view of the experiment will appear. As with any breadboard, you can see the animation and control it with the switch.
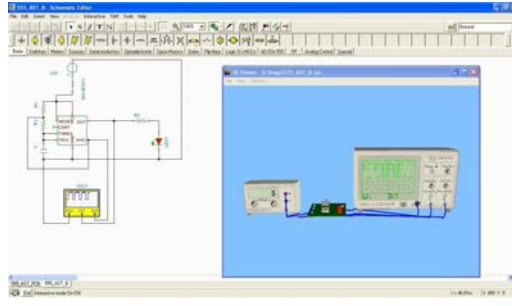
If you turn the PCB upside down (by holding down the left mouse button over an unused area and moving the mouse), you can see the connections of the wires from the PCB solder side to the oscilloscope.
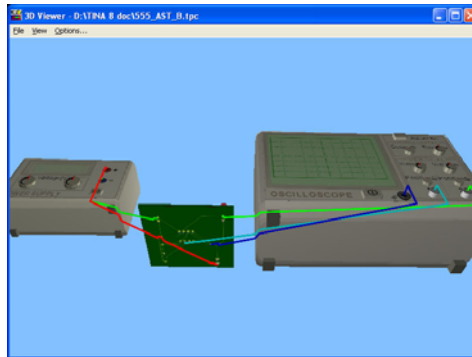
Note that you could use connectors instead of soldering wires directly to the PCB and connect the instruments through those. By adding Header2 and Header3 (they can be found under the Connectors button on the Switches component toolbar), you can make the interconnection shown on the picture below.
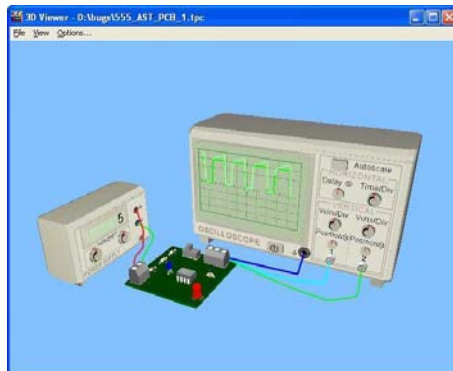
4.7   Mechatronics Extension

With the optional Mechatronics add-on package, you can create and simulate multidisciplinary designs, currently including electronics, 3D mechanics, and control engineering. You can place light sources, light sensors, motors and actuators in TINA's mechanical window and connect them to their counterparts in analog or analog-digital mixed electronic circuits. You can control the mechanics from the electronics part of TINA even with complex software written in C or assembly language. Then compile and execute the code in the MCUs while running the electronic and 3D mechanical simulations simultaneously.

In the following section, we will demonstrate the setup of the coupling between a circuit in the schematic editor and the mechanical window of TINA.

As an example, we will simulate a simple pendulum. Note that the mechanisms are created using the Newton software, which is now integrated into TINA (formerly it was only available as a standalone software package.) The detailed rules of using Newton are described in the Newton manual along with further examples.

We will build and simulate an electronically controlled "perpetual motion" pendulum (Novelty electric motor US Patent 3783550 A). The working principle described in the patent would allow us to construct many controlled devices; here, however, we will focus on the accelerated pendulum only.

A permanent magnet has been placed close beneath the pendulum. As the pendulum swings, it moves above the coil and induces an electric voltage in the coil. This voltage forces current into the bipolar transistor's base, which turns on the transistor and produces a much larger collector current. This current passes through the coil and acts as an electromagnet and accelerates the pendulum. As a result the amplitude of the pendulum's swing will not decrease (In fact, at first it will grow and then become stable), as long as energy is supplied to the controlling circuit

The controlling circuit and the pendulum are shown in the following screen.

Now let's create our first system.
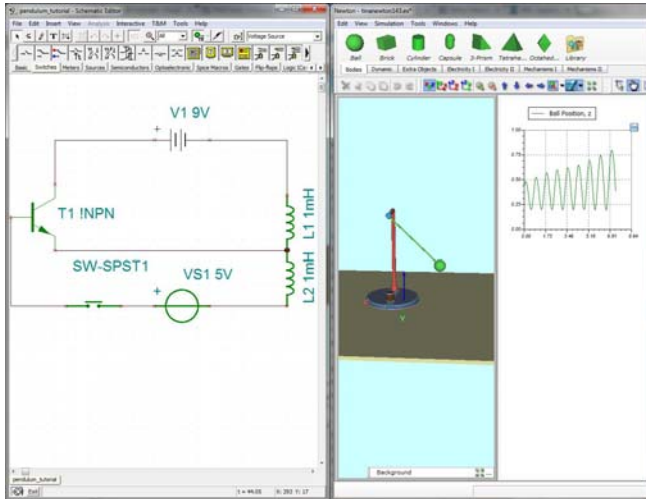
First we open a new mechatronics example.

Click on the *Analysis* menu then *Enable Mechatronics coupling*. Note that this menu item is available in TINA versions extended with the Mechatronics add-on package only. The Newton window will appear. On the top of the Newton window there is the Object Toolbar. It contains the icons of the commonly used bodies, constraints, and accessory objects that can be used to build mechanisms. The objects are grouped logically according to function, and each group has its own tab.

Next, we create a simple pendulum.

Click on the Extra Objects tab on the object toolbar, than place a

stand onto the 3D scene by clicking on the stand icon . Select

the object - click on it - then click the Dynamic tools and add a hinge

 to the stand.

Add a ball to the scene by clicking the appropriate icon in the

Bodies object toolbar. Move the ball under the hinge. Use the XY

planes to lift the ball up by double-clicking on it and setting the

Location z to 0.2 m in its Properties Window.

DesignSoft

Now link the ball to the hinge. If the hinge is selected, a stick joint will appear. Click and hold down the left mouse button on the stick, then move the cursor to the ball to be linked. When the left button of the mouse is released, the two objects are linked.

Now take a Coil from the *Electricity I* toolbar, and place it to under the ball.

Now let's create diagrams to represent the pendulum motion. First, switch the Description window to the Edit mode. Find the *Diagram*

icon [icon] and click on it to bring up the Diagram properties

window. Select Ball – Position – z, then click the OK button.

Create the circuit below in TINA's Schematic Editor.

In the next step, we will define the coupling between the pendulum and the circuit.
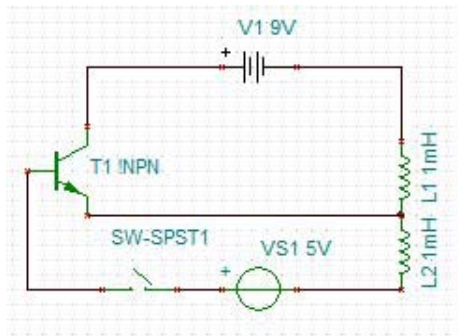
In the Newton window, select the *Tools* menu, then click *Couplings*. Click the *Add new coupling* button. From the drop-down menu select *Mechanical – Analog Converter*

Select the VS1 generator from the Tina components on the left side, then push the Editor button on the right. The Editor Window will appear. In this editor a Pascal-like language code can be added.

Enter the following code:

```
Begin
  result := 0.14/
dist(ball,coil)*sgn(ball.v[3]);
End.
```

GETTING STARTED

DesignSoft

This code will determine the voltage of the generator in TINA representing the voltage induced in the coil by the moving magnet.

The "dist" function calculates the distance between the center of masses of the ball and coil objects. This is a rough approximation for the sake of simplicity. If more precision is desired, one must use a more accurate formula.

Save the code under the name ma.cod, by pressing the Save button or using the Save command in the File menu. Close the Editor window.

Now you should assign the voltage calculated by the function to the VS1 generator in TINA. Scroll through the available codes using the control of the Code field and select ma.cod, then press the Save button. Close the Coupling Window.

Next we create a coupling between the coil object in the 3D window and the schematic coil symbol in TINA.

Press the Add new coupling button again. On the drop/down menu, click on COIL

From the Tina component list select L1, the coil where the Collector current of the transistor flows, and from the Newton component list select COIL

Now we need add a formula for calculating the "magnetic charge" (coil.charge) to be placed into the center of mass of the electromagnet (coil), which will be used for calculating the force between the electromagnet and the ball.

Press the *Editor* button and enter the following code:

```
var
N : integer; // number of turns
I : real; // current
A : real; // cross section area of core
L : real; // total length of magnetic field
C0 : real; // speed of light
m : real; // magnetic pole strength
Begin
N := 15000;
```

```
I := ABS(value);
A := 5.3E-4;
L := 0.2;
C0 := 299792458;
m := N*I*A/L;
coil.charge := m/C0;
End.
```

Save the code under coil.cod

NOTE:

> *In this simple approximation the electromagnet is substituted by a "magnetic charge" calculated from the current of the coil. This way the force between the electromagnet and the ball can be calculated simply by the Coulomb's law, included in Newton.*
> *(http://www.daviddarling.info/encyclopedia/C/*
> *Coulombs_law_for_magnets.html*
> *http://www.sciencedirect.com/science/article/pii/S1875389212000727)*

Now you should assign the magnetic charge calculated by the function to the electromagnet in the 3D window. Scroll down the available codes using the control of the Code field and select coli.cod, then press the Save button. Close the Coupling Window.

Finally set some parameters.

In the TINA window, set the Simulation time unit to 1s using the *Options* command under the Interactive menu in the Schematic Editor.

In the Newton window, double click on the ball and in the Property window click the Material icon and set the Charge to 4E-4.

Drag the ball into an initial position of approx. 30° as shown on the screenshot at the beginning of this section.

Now start the simulation in interactive transient mode by pressing the TR button in the Schematic Editor. If the circuit power switch is off, the pendulum will swing as a damped harmonic oscillation. But if the switch is turned on, the amplitude of the pendulum swing increases at first, then settles down to constant value.

You can find the complete example at:
EXAMPLES\Mechatronics\pendulum\Pendulum_tutorial.TSC

You can also find further examples under EXAMPLES\Mechatronics.

PRINTED CIRCUIT BOARD (PCB) DESIGN

Once you have completed your circuit diagram, you can design a printed circuit board to manufacture your design. This is easy to do in TINA 7 and later versions, since PCB design is an integral part of the program.

We'll learn about the PCB design process through a few examples. The files from the different phases of the design examples have been saved in TINA's Examples\PCB subdirectories using the following naming conventions:

PCB

* origin.tsc	original schematic file
*.tsc	schematic file backannotated (after pin/gate swapping and renumbering)
*placed.tpc	design parameters set, components placed pcb file
* routed.tpc	net properties set and routed pcb file
* finished.tpc	optionally pin/gate swapped and renumbered, routed, silkscreen adjusted, documentation layers finalized pcb file

5.1 Creating a Printed Circuit Board

To see the first example, open the opamp2.tsc project from TINA's Examples\PCB\OpAmp folder. The following schematic will appear:
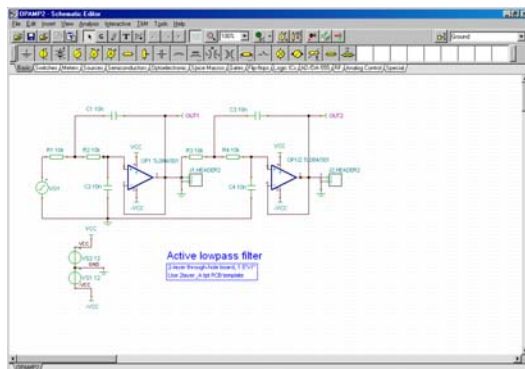
5.1.1 Setting and checking footprint names

The most important thing in PCB design is that every part in your schematic must have a physical representation with exact physical size. This is accomplished through so called footprints–drawings showing the outline and the pins of the parts.

TINA's footprint naming uses as a starting point the IPC-SM-782A (Surface Mount Design and Land Pattern Standard) and the JEDEC standard JESD30 (Descriptive Designation System for Semiconductor Device Packages).

In TINA, we have already assigned default footprint names to all parts which represent real components.

NOTE:

> *Some parts used for theoretical investigations (for example, controlled sources) do not represent real physical parts so you cannot place them on a PCB. If your design contains such components, you should replace them with real physical parts.*

PCB

DesignSoft

Of course there is no guar antee that the default physical representatives of the parts are the same as those needed by your design There are two ways to check this.

1) You can use TINA's "Footprint name editor" which you can invoke from the Tools menu. In this dialog you see all of TINA's components and the corresponding footprint names.

Clicking on the footprint name fields, you can select from the available footprint names. In the dialog, components that do not already have a footprint name association will be denoted by red characters and also by "???" in the footprint name field.

Note that although there are 2 op-amps in the schematic diagram, there is only 1 opamp part in the footprint list above: a DIP14 package denoted by OP1. This is because the TL084 IC we use contains 4 op-amps, so both of our op-amps are in the same package. This is shown by the labels of the op-amps in the schematic diagram: OP1 and OP1/2. If the parts are in the same package, the labels

must also be the same. Note that the second, third, and fourth op-amps have the extensions /2, /3, or /4 added to the label. You must not set the extensions /2, /3, or /4 manually, rather, use the "PCB information" dialog of the parts. An example: double-click on the parts to see the property dialog.
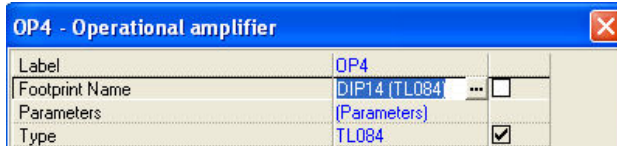
Now click the Footprint line and press the ⋯ button. The PCB information dialog appears and you can select the number of the part in the package at the Part field of the top-right corner of the dialog below.

2) Alternatively you can double-click on each part and check the Footprint Name of the component property dialog.

DesignSoft
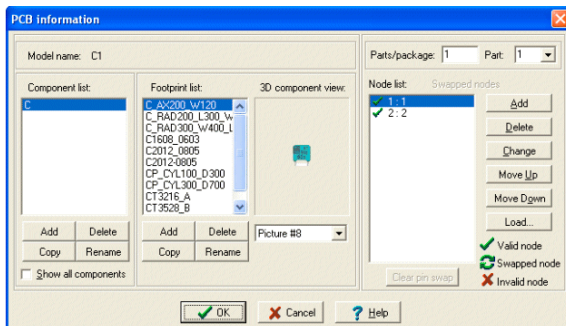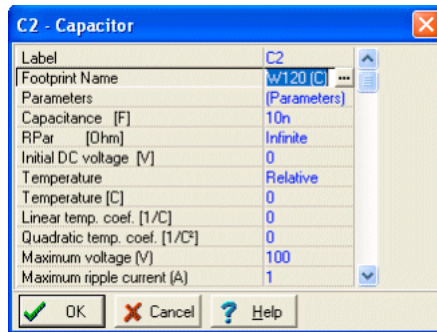
You can also click the ⚏ button in the Footprint Name line and see the "PCB information" dialog where you can select from the available footprint names. You can also see the 3D view of the different parts via the 3D package view field of the dialog.

If you find the footprint name you want on the list, click on it and press OK: you will be returned to the component property dialog with the selected footprint name in the Footprint name line. To confirm the change, press OK on the component property dialog again.

If you do not find the footprint name you want, you can add a new footprint using the Add buttons of the "PCB information" dialog. Press the Help button for more information.

When everything looks good, you can make a final check by clicking the ⚏ 2D/3D view button . The 3D view of those components for which a physical representation has already been added will appear.
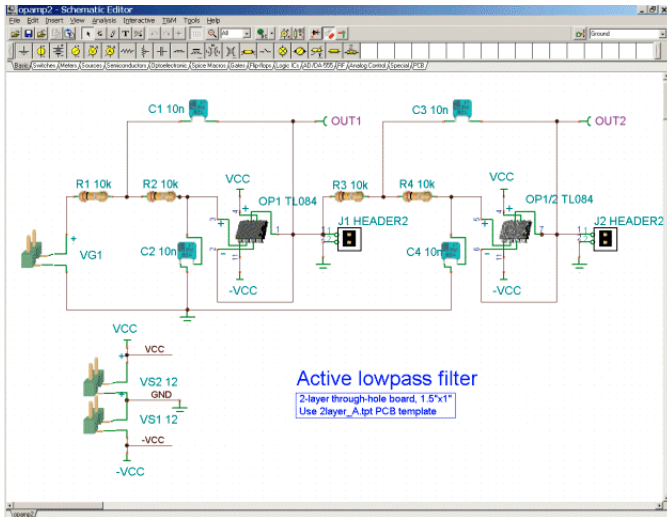
5.1.2   Invoking TINA PCB

Once each component has a satisfactory physical part association, we can proceed to PCB layout design. To do this, press the ⚏

button on TINA's toolbar (the last icon on the right) or select the "PCB Design" command on the Tools menu. Set the items in the PCB Design dialog as shown below.

Select "Start New Project", "Autoplacement" and "Use board template". With the Browse button find and select the 2layer_A.tpt template files from TINA's Template. The settings are appropriate for a double-sided PCB.

PCB
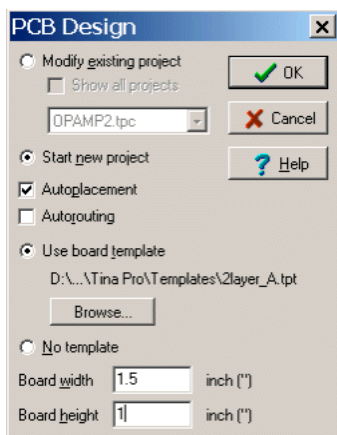
If you use a template, you should set the level of manufacturing complexity. The following three levels of manufacturing technology are defined by the IPC-2221 generic standard.

```
Level A : General Design Complexity
Level B : Moderate Design Complexity
Level C : High Design Complexity
```

The template file specifies the number of layers and their properties: system grid size, autorouter settings, spacing and track width. The following templates are included with PCB Designer:

	Level	Routing Layers	Plane Layers	Routing	Spacing	
1layer_A.tpt	A	1	-	25	12·1/2	Allows one track between standard
2layer_A.tpt	A	2	-	25	12·½	DIP·IC·pin
2layer_B.tpt	B	2	-	8·1/3	8·1/3	Use for SMT or mixed-technoly board
2layer_B_mm.tpt	B	2	-	0.1	0.2	
4layer_C_mm.tpt	C	2	2	0.1	0.15	For moderate and high density SMT boards

You can choose PCB template based on technology, density and package pitch.

Finally, you can set the size of the PCB board in inches or mm depending on the measurement unit settings in the View/ Options dialog of TINA.

When everything is set properly, press the OK button and the PCB layout design will appear with all the components automatically placed on the PCB board.
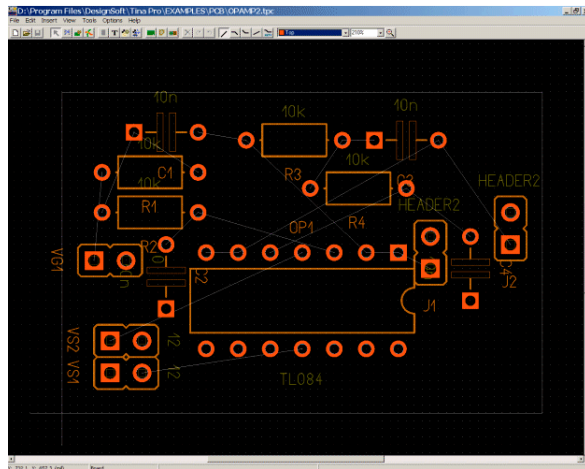
Now click and drag the parts to new positions as, shown on the figure below. (Find "opamp2 placed.tpc" to check your results.)
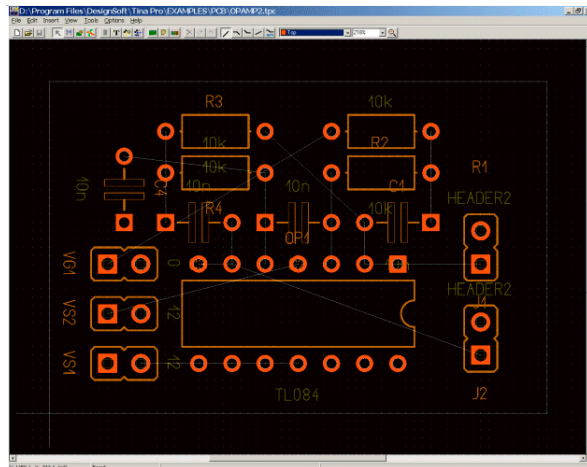
Press F4 to invoke the Net Editor and set nets routing width. First, click on "Modify all" and enter 12.5 into the "Track width" field. Then select power nets (Ground, VCC, -VCC) and set their widths to 25mil.

To automatically route the netlist, press the F5 button or select "Autoroute board" command from the Tools menu. The following screen will appear:
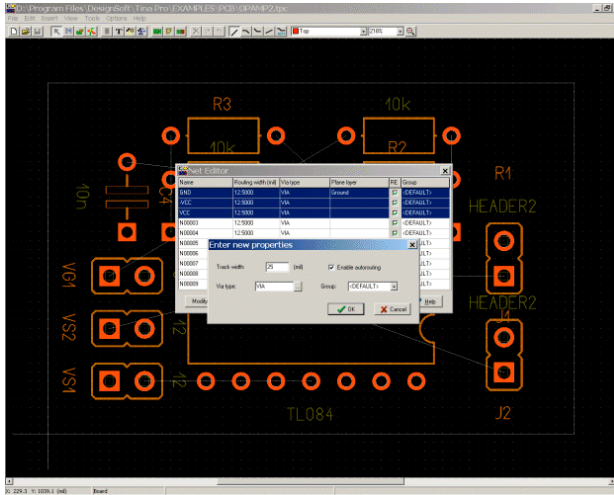
To see if everything is routed correctly, press F7 or select DRC (Design Rule Check) from the Tools menu. The following message will appear:

To finish our first simple design, let's add a text box to the silkscreen/ assembly layer. To do this, click the T button on the toolbar. The following message will appear:
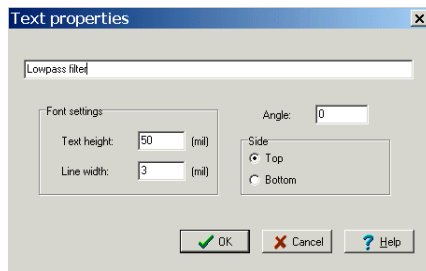
Enter the text into the empty upper field and press the OK button. The text will be attached to the cursor. Move it to the place shown on the picture below and press the left mouse button.

PCB

DesignSoft
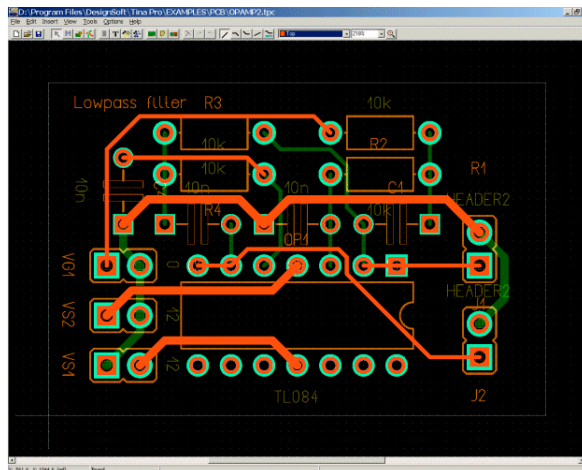
Finally, you can check your design in full 3D. To do this, press F3 or select 3D View from the View menu. After some calculation the following window will appear.
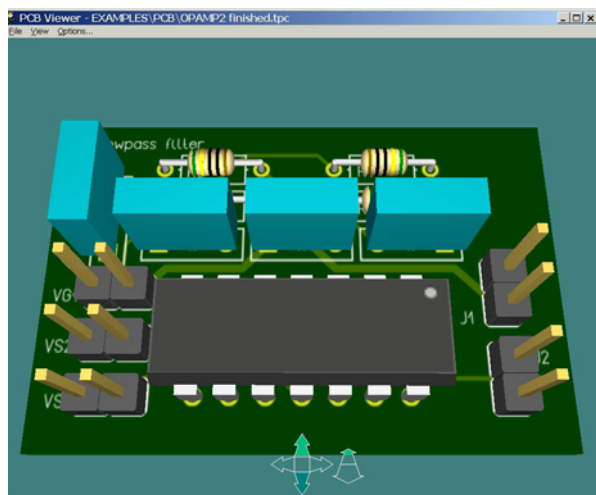
You can rotate the 3D model and zoom in and out with the control arrows at the bottom. You can display or hide these arrows through the Options menu with *"Use control arrows"* checkbox.

You can also rotate this 3D model by clicking with the mouse at any point, holding down the left button and moving the mouse. You can also move the camera forward or backward to see the whole design or just a part of it in more detail. To move the camera, hold down the right mouse button and move the mouse.

After this you can either print your design or create a Gerber file for a manufacturer.

To print use Print… from File menu.

To obtain Gerber files to direct a photoplotter, choose Export Gerber file from File menu. (Gerber option can be changed through Gerber output setting under Options menu.)

This example concludes the introduction to the use of TINA's PCB layout module.

We also suggest that you study the examples in TINA's EXAMPLES\PCB folder.

5.2   **Creating PCB components**

Now, let's see how to create our own PCB components for use in a PCB design. We place an LM78L05ACM (plastic SO-8 package) voltage regulator on the schematic to make the circuit +12V powerable. If we do not want to simulate the circuit, TINA gives specific tool to create PCB symbol. So, click PCB Component Wizard… in the Tools menu and enter the name of the component in the Name field.
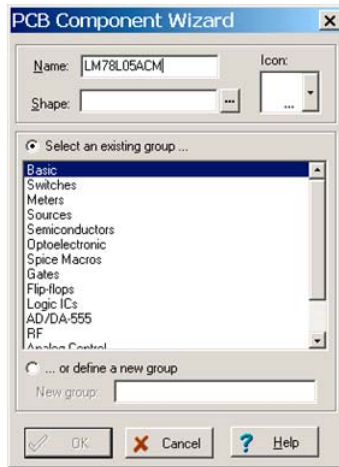
Click the ··· button to choose the component Shape from the library. Enter a few letters from macro shape name: <VoltageReg>, then click OK.

If you do not find a n appropriate macro shape for the new component, use the Design Suite's Schematic Symbol Editor.

Choose the REG icon from the Icon drop down list.

PCB

205

DesignSoft
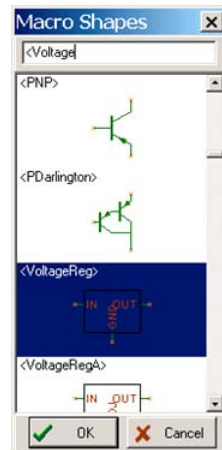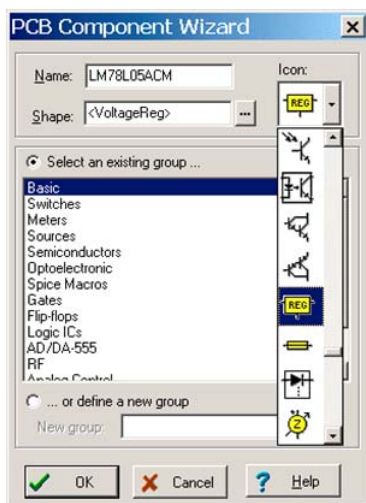
So that we will be able to access a new PCB component in TINA, we should add it to one of the component groups. This time, you can define a new group for our component, called PIC flasher, then press the OK button.
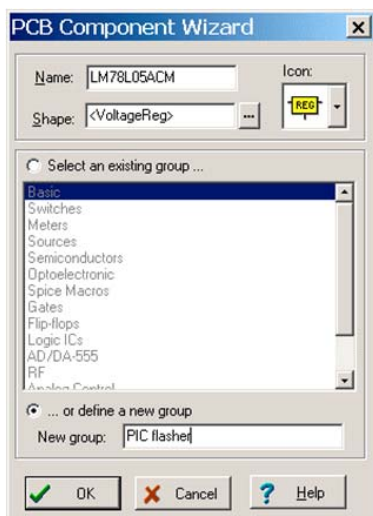
Finally, you may save the new PCB component macro into the Macrolib folder.

Replace the short circuit (on the schematic) between Vcc and +5V with the new component and double-click on the schematic symbol of the regulator. Enter the PCB information:

1.   Click in the Footprint Name edit line on the ⋯ button.

2.   Click the Add button on the Component list area and type LM78L05ACM. Then press OK.

3.   Click the Add button on the Footprint list area and choose SO8 from the SMDIC category.

4.   Build the part node list with the help of the component data sheet (http://www.ti.com/lit/ds/symlink/lm78l.pdf) : click Add, then select OUT from the Change Node List and press OK. Repeat these steps until all of the pins are defined and, at least, click OK.
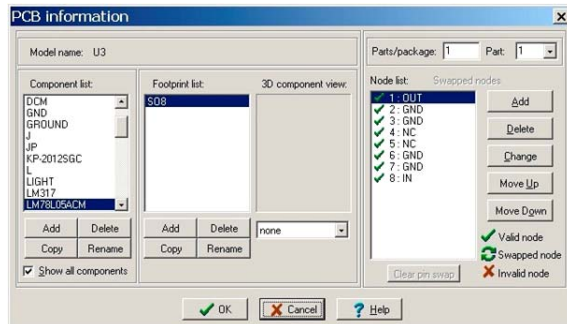
By this point, we have set every parameter and have a "PCB-ready" component. To verify the part correctness, you can open Examples\PCB\PIC Flasher rigid\PIC flasher reg.TSC and check it with ERC.

PCB

DesignSoft
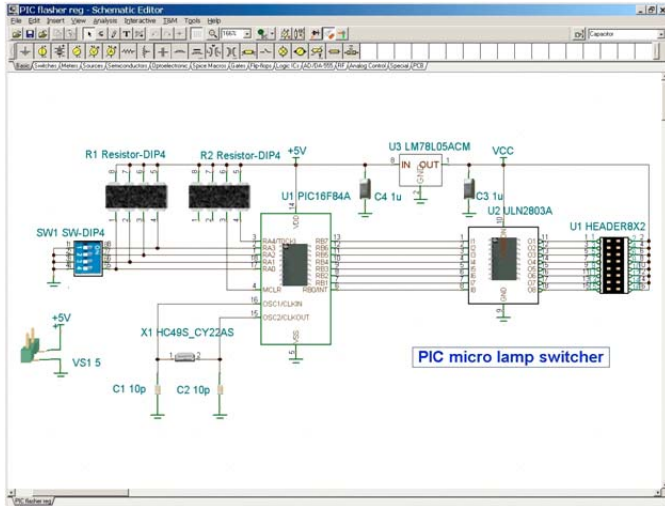
As the electric type of U3 pin 1 (OUT) is defined to be an output and is connected to the power pins of U1 and U2, you will encounter ERC errors when you call PCB Wizard or perform an ERC. But knowing that U2 is the power source for the ICs, we can safely disregard this error.

5.3   PCB Design Techniques

5.3.1   Multiple Units in the Same Package and their Power Supply

In order to simplify and reduce clutter on schematics, it is common practice to not show the power supply pins of Logic Gates and ICs on the schematic symbols. Likewise, several (two to eight, typically) gates or opamps are often manufactured in the same IC package. Although these simplifications do not have any effect on the result of circuit analysis, proper information is essential for the PCB design. In this chapter, we will show how to manage and solve these problems.

Let's consider the following logic circuit containing 6 NAND gates (EXAMPLES\PCB\Digital lock\Digilock.TSC):

DesignSoft
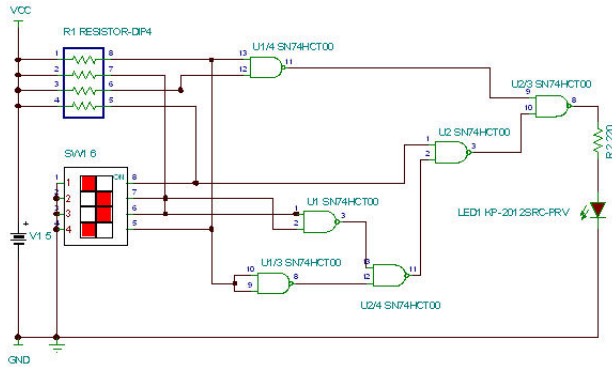
The SN74HCT00 package contains 4 NAND gates, but our design requires 6 gates. We will therefore use 2 packages labeled U1 and U2. In the U1 package we use all 4 of the gates, while in the U2 package we only will use 2 of them. The gates in the same package are labeled as U1, U1/2, U1/3, U1/4, respectively. The numbering of the gates is defined in the PCB information dialog of the parts. For example, let's see the assignment of the 3rd gate in the U1 package in the PCB Information dialog. Double-click on the U1/3 gate at the bottom of the schematic diagram. The property dialog of the component appears.
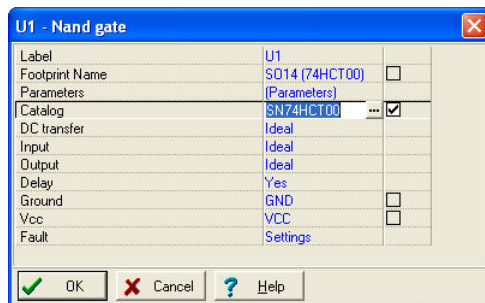
Click on the Footprint Name line and press the [...] button. The PCB Information dialog appears.

DesignSoft
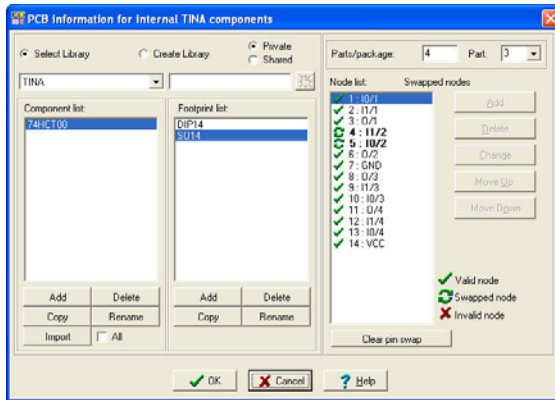
The Parts/Package assignment on the top-right part of the dialog shows that the package contains 4 gates and we are using the 3rd one.

Using the list box on the right, you can select any of the 4 instances of the NAND Gate. Be careful not to use the same gate more than once.

We will discuss in this section the assignment of power supply outputs to component power pins. In the quest for simpler, less cluttered schematics, the power pins are normally not shown, but can be seen when the part is opened up. The usual names are VCC for the power supply positive voltage, and GND for ground, the power supply negative voltage. And the usual VCC value is +5V. Recently, though, more and more ICs (like DSP chips, FPGAs, and processors) require 2 or even 3 different "VCCs." Study the chip data sheets or component properties and be sure to label the positive supply correctly. You may end up with a net for +5V VCC, another net for +3.3V VCC, and perhaps even a net for +1.8V. You could use the basic definition of VCC as +5V, a jumper labeled +3.3V, and another jumper labeled +1.8V. Jumpers in TINA are also useful for parts where the schematic symbols do not contain the power supply pins.

Let's study the Ground and VCC connections in the property dialog of the U1 package.

You will find (in the left side the schematic) the jumpers with the same GND and VCC names as in the property dialog above, establishing the connection with the package and the circuit nodes where the jumpers are connected.

Note that if the Ground and VCC fields of a component are left empty, TINA will connect them automatically to ground and +5V.

IMPORTANT: the Ground and VCC fields for all parts, even parts in the same package, must be filled out if their required voltage is different from the default (+5V).

Our completed PCB layout and its 3D view look like this:

DesignSoft

5.3.2   Differential pair routing

Differential pair routing in most cases is a layout design method to create a balanced transmission line to carry differential signals.
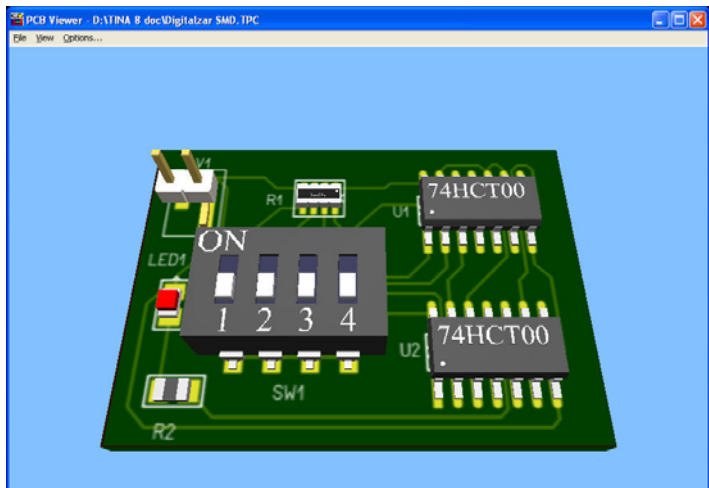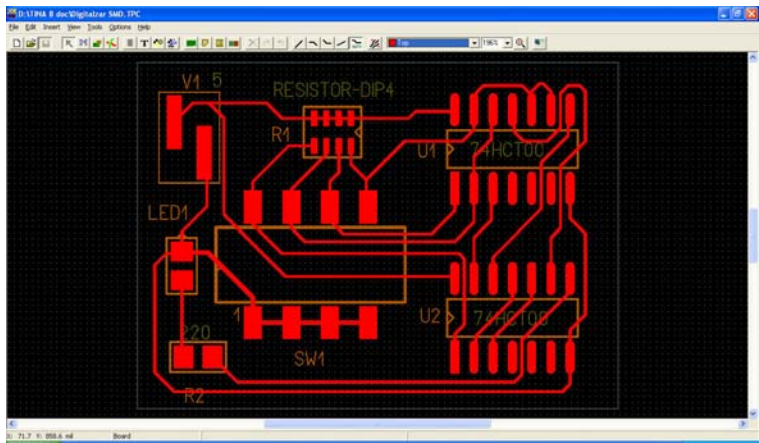
You can learn about differential pair routing in TINA through the example below. The example contains only the critical signals of a high-speed USB 2.0 connection.

Open and study the following circuit (high-speed USB signal connection.TSC) included in the Examples\PCB folder of TINA.

The circuit connects a Cypress CY7C68013A USB device controller to a USB connector.

The controlled differential impedance of the data line traces (D+, D-) is important in USB 2.0 PCB design. The differential impedance should be matched to the USB cable's differential impedance, which is 90Ω for the signal pair as per the standard.

The differential pair is defined on the schematic by naming the wire ID with the _N and _P postfixes of each wire of the pair.

Double-click the wire between the D- of J1 and DMINUS of U1. The following dialog will appear.
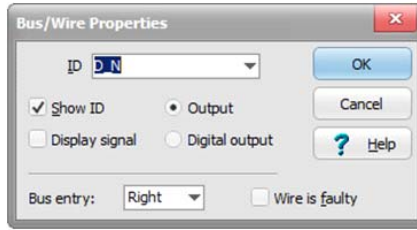
Do the same with the D+ wire connection to check the name of the wire. Note: The *Show ID* option must be switched on for the program to display the ID value.

It is important to know that the TINA PCB will be informed from the net names that the nets are members of a defined pair.

But before you enter the PCB design phase, you must calculate the geometric parameters of the edge-coupled microstrip, the physical representation of the differential pair on the layout side.

PCB
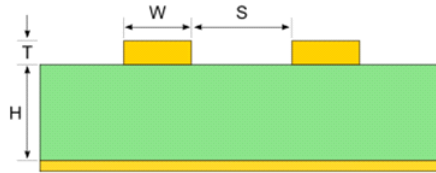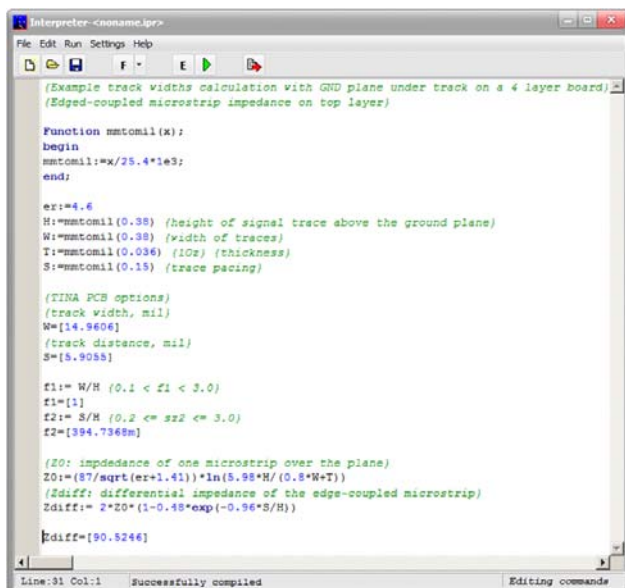
For the exact values of S and W, let's assume we are to route a four-layer printed circuit board with the following structure, which determines the value of H and T.

Layer	Thickness, μm	Material
SMT	20	Solder Mask
TOP	18	Copper Plating
TOP	18	Copper Foil
	380	Prepreg
GND	35	Copper
	760	Core
POWER	35	Copper
	380	Prepreg
BOT	18	Copper Foil
BOT	18	Copper Plating
SMB	20	Solder Mask

Now, with the help of the TINA Interpreter and some of the above constant values, you can estimate the differential impedance. Double-click the rectangle under the MCU symbol in the schematic, and the Interpreter window will come up.

The script includes formulas calculating differential impedance for selected W, S differential pair parameters. If you change any of the input parameters, press the *Run* button to recalculate the Zdiff value, which must remain within 10% of 90Ω.

Note: With 35μm thickness (1oz) on top, if you choose track spacing of 0.15mm (6mil) and edge width of 0.38mm (15mil), then calculation results in 90.52Ω differential impedance, which is acceptable.

Now, as you have double-checked the schematic design, press the PCB Designer button, accept the settings in the PCB Design dialog box, and press OK.

The following unrouted PCB design will appear.
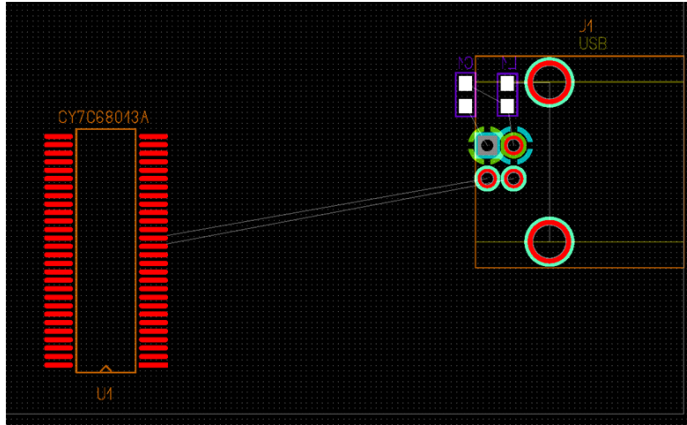
Check the netlist editor settings (*Tools/Footprint Editor*) and set 15mil routing width value for D_N and D_P differential pair nets. (Note: The design inherits the mil unit from the schematic editor. See *View/Editor Options*.)

Now, select Draw tracks mode and right-click on the DPLUS pin of the MCU. The following pop-up menu will appear.
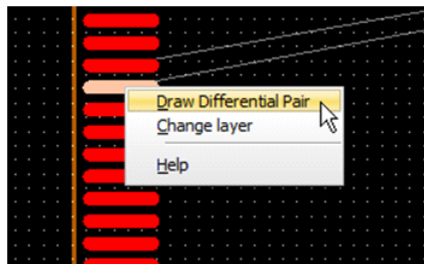
Select *Draw Differential Pair* and right-click again to choose *Set Track Distance*.
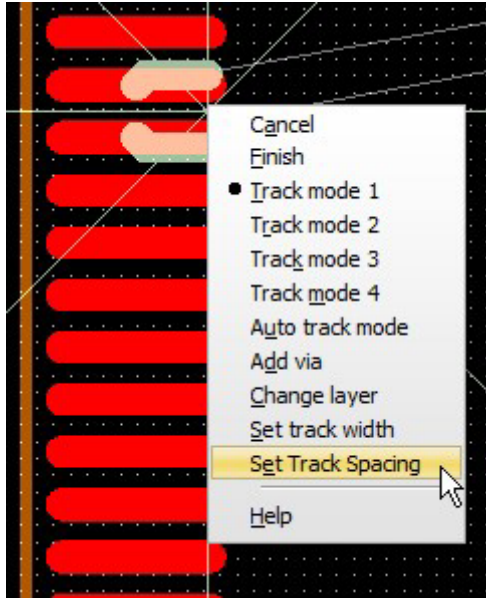
Set the spacing value to 6mil, and right-click again to choose Track mode 3.

Now, you can draw the differential pair tracks to the USB connector while the PCB Designer maintains the spacing and track properties.
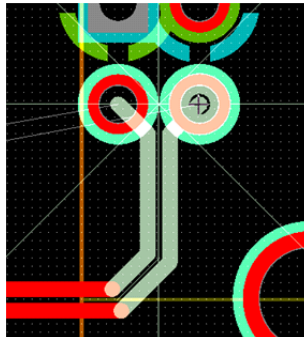
After this, you will get the following, or a similar layout to the one in the high-speed USB signal connection routed.TSC included in the Examples\PCB folder of TINA.

DesignSoft
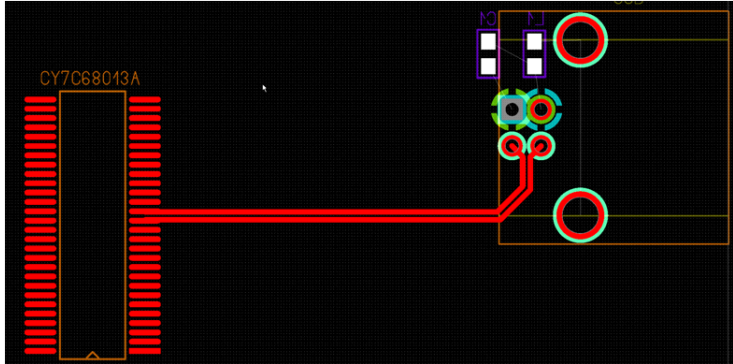
5.3.3 Creating Buses in the Schematic Editor and the PCB Designer of TINA

TINA PCB provides a tool to help the routing of buses—that is, data signals or other collections of signal tracks grouped together with the same distance and width.

Bus connections in the schematic editor of TINA are automatically translated into bus connection–type tracks in the PBC designer, so when you start drawing buses in the PCB design, all the tracks are drawn together. Bus routing should be done by manual design.

We will demonstrate creating buses through the example below.

Let's study the following circuit (*Ethernet cable tester transmitter.TSC*) included in the *Examples\PCB\Bus routing* folder of TINA.

PCB

DesignSoft
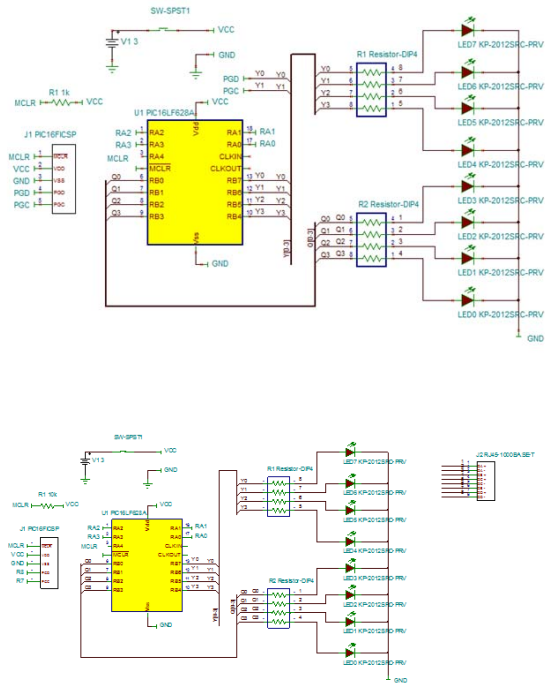
The circuit already includes two buses named Q and Y, drawn with a thick line.

Delete the Q bus running at the bottom of the schematic by selecting it with a click and pressing Del, and also delete the connecting wires.

Now, re-create this bus and its connections as follows.

Start drawing the Bus by selecting the Bus command from the Insert menu or with the Ctrl B hotkey. Draw the Bus the same way as drawing wires, by clicking the first point, moving the mouse, and then clicking at the endpoint.

Double-click the bus, and in the property dialog box, set its properties as Q[1-4]. This means that the name of the Bus is Q, and it contains 4 data lines. Press OK.

PCB

DesignSoft
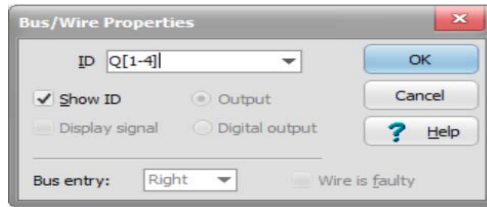
Now, connect the Bus with the circuit usingwires starting from the nodes of the Resistor network and ending at the Bus. Double-click the connecting wires and set their ID from the drop-down list, and press OK.
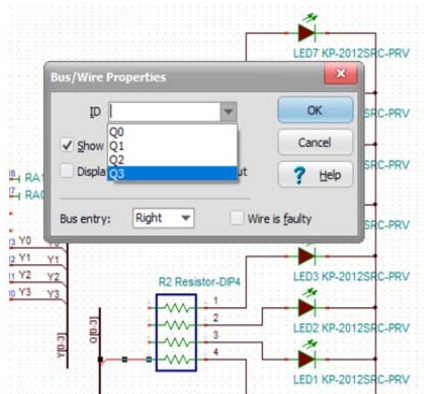
The ID of the wire will appear on the wire.
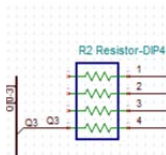
Do the same process for all 4 wires, both on the Resistor network and the PIC side.

PCB

DesignSoft

Now you have re-created the bus.Build the PCB layout.

Press the PCB button, accept the settings in the PCB Design dialog box, and press OK.

The following unrouted PCB design will appear.
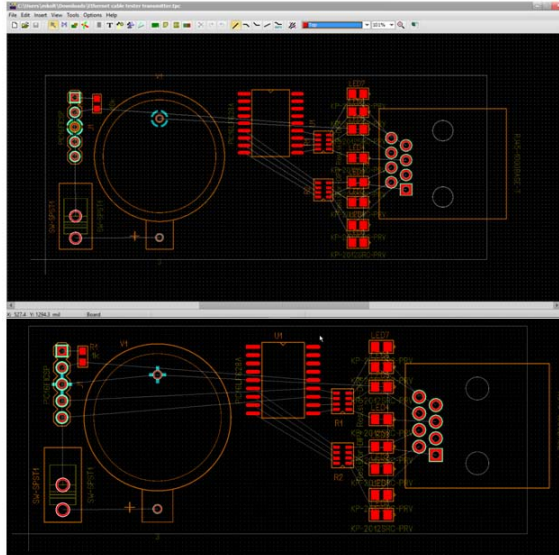
Right-click the RB7 pin of the MCU. The following pop-up menu will appear.
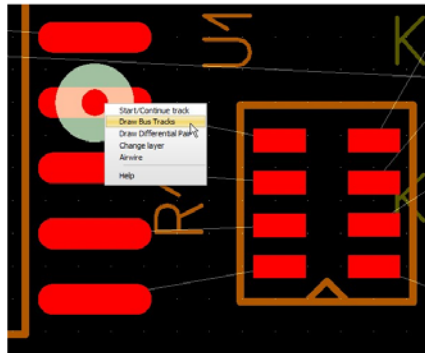
Select Draw Bus Track.Now you can draw the Y Bus tracks and connect with the Resistor network.
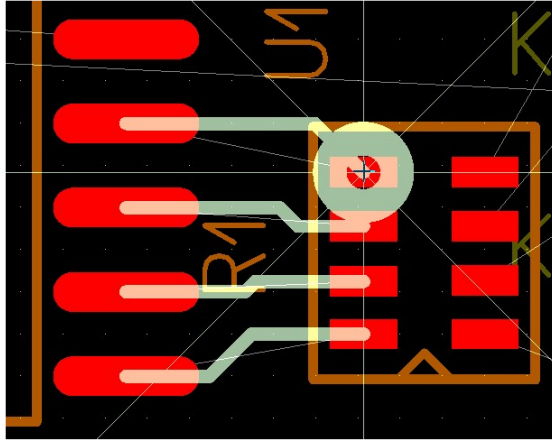
Note that in most cases, you still need to edit the tracks after the connections. In some cases, you may need to finish the Bus drawing before the connections and then make the connections manually.

In a similar way, you can also route the Q Bus.

After this, you will get the following, or a similar, layout.
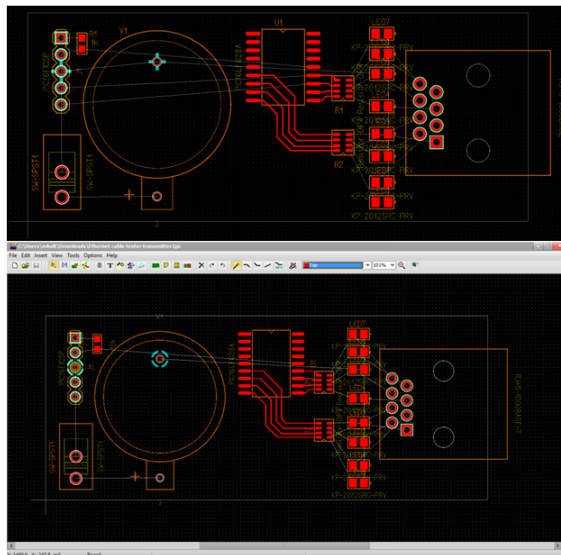
Note, that, while drawing the Bus, you can set the distance between the Bus tracks by right clicking and selecting Set Track Distance on the popup menu displayed. On the menu Default means that the distance is automatically defined by the pins where the Bus started, DRC( ) means the minimum distance allowed by the DRC, and Custom means that you can enter the distance manually.
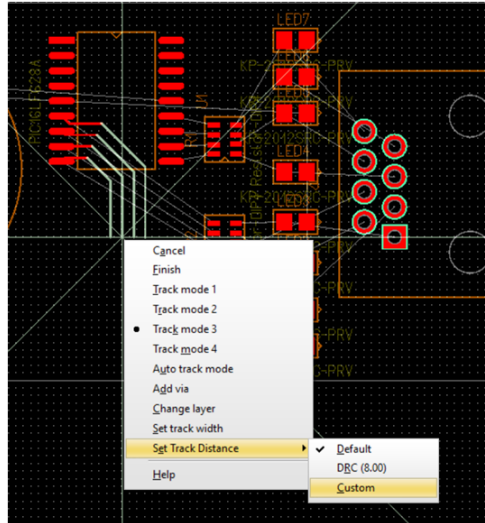
Finally, by pressing F5, you can do the rest of the routing with the Autorouting tool and some manual editing.
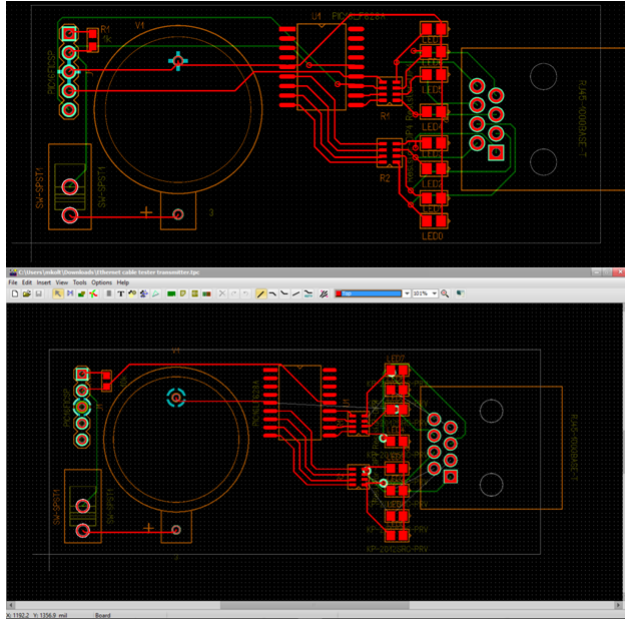
You can also see the PCB in 3D by pressing the ▦ (3D View) button.

PCB

DesignSoft
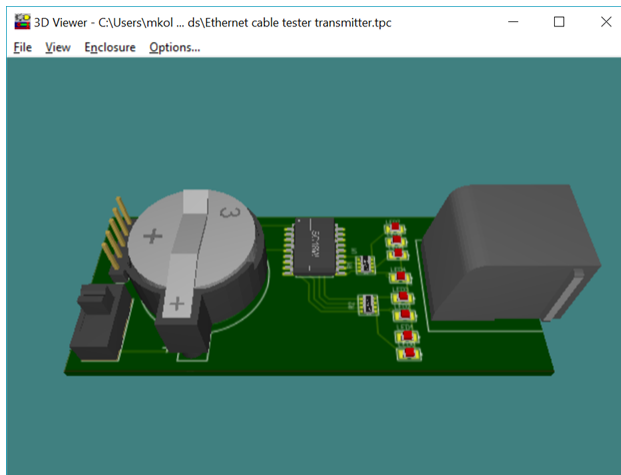
5.3.4 Repeating Circuit Blocks (Using the Copy Macro Function)

Sometimes you may need to reuse certain modules several times in the same circuit. Examples include multi-channel amplifiers, signal conditioners, redundant power supplies, memory modules, filter structures, etc.

You can use the Copy Macro function in the TINA PCB Designer to quickly route and place such modules on the PCB board.

To practice producing such a layout design, create a stereo microphone amplifier using the circuit in the *\Examples\PCB\PreAmp\microphone pre-amp.TSC* file as a template.
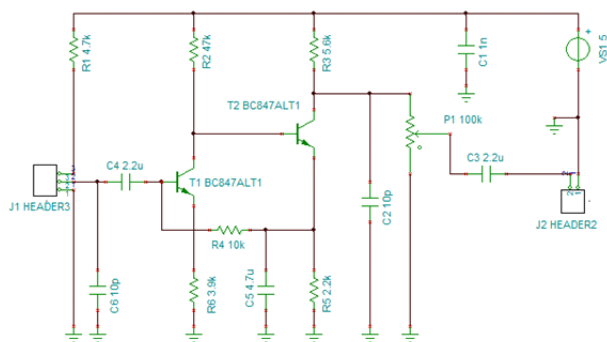
This circuit is a mono-microphone preamplifier with simplified external connections.

Turn this circuit into a schematic macro subcircuit in order to add one more channel to the amplifier.

Remove the connectors, as these will be placed later, outside the macro. Add macro pins.

PCB

DesignSoft
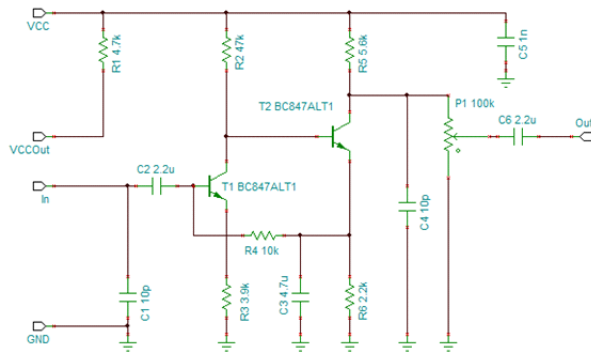
Start the *New Macro Wizard* from the *Tools* menu. Create a macro from the schematic and save it as Preamp.TSM.

Now, using the Macro... command on the Insert menu in TINA, add the macro into a new schematic. Select the macro and perform a copy-paste operation. After that, add power supplies and connectors for both channels. If you double-click the macros and press the Enter Macro button, you will see identical component labels in both macros, which is not acceptable in the layout design. Choose Renumber components from the Tools menu to fix that. This will automatically renumber all the parts in the schematic, as if you had a flat schematic design.

The ready-to-use schematic design (*Stereo microphone preamplifier.TSC*) can be opened from the \Examples\PCB\Copy Macro folder.

PCB

DesignSoft
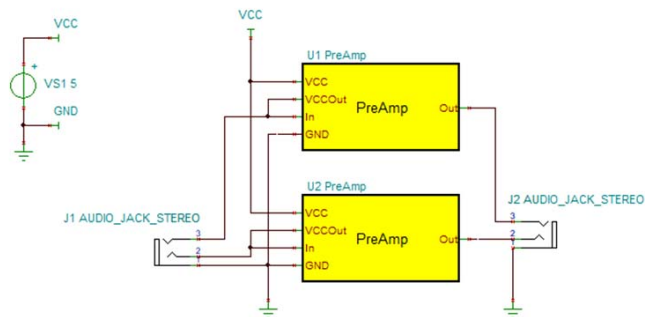
Now build the PCB layout. Press the PCB button and click *Start new project* in the PCB Design dialog box. Apply the settings shown below, and then press OK.
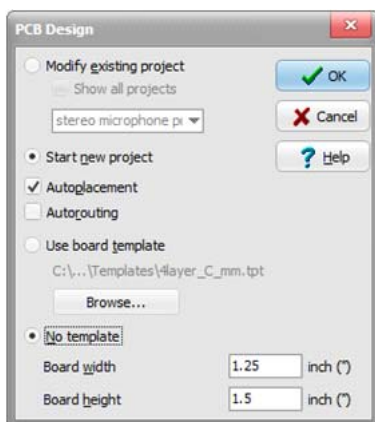
Now place and route the parts of the upper preamplifier channel to use it as a reference for the other one. The result should be similar to the PCB design in the *Stereo microphone preamp U1 placed.TPC* file, where the components on the upper half of the board are in their final positions. The routed version, shown below, can be found in the same folder as *Stereo microphone preamp U1 routed.TPC*.

PCB
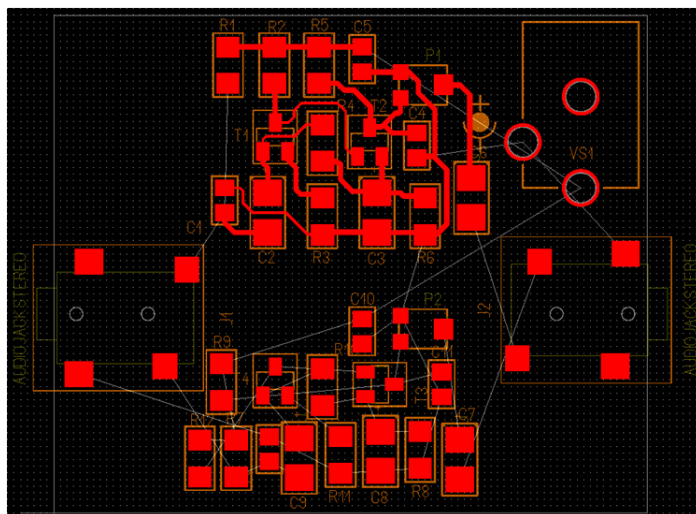
Start copying the U1 macro layout by right-clicking the R1 resistor. (The editor must be in Select mode.) Choose *Select Macro* from the menu.
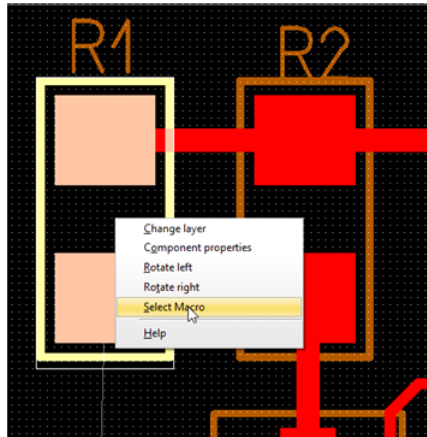
All the parts of the U1 macro and their connections will now be selected.
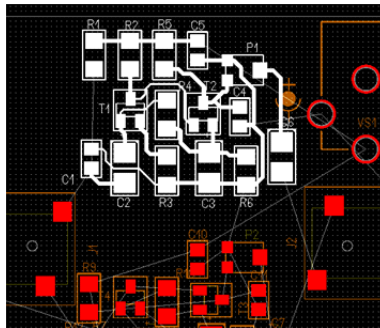
Right-click the corresponding R7 resistor in the U2 macro and choose *Copy Macro.*
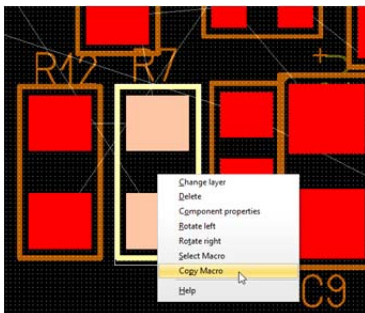
The program will automatically place and route the parts belonging to the U2 macro. The initial alignment is based on the position you selected for the resistor R1 in the copy-paste operation, but you can move this component group to a different place on the board if needed.
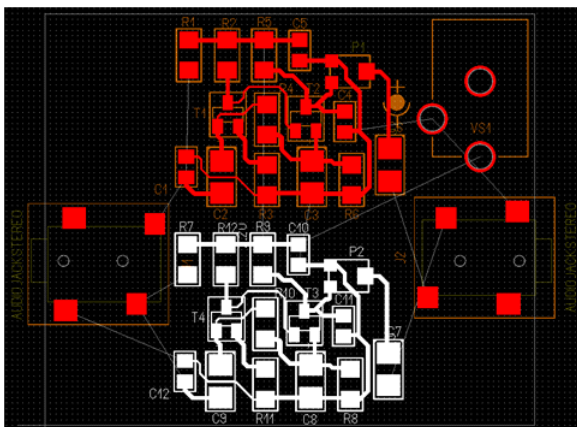
Press F5. You can now do the rest of the routing with the Autorouting tool and/or some manual editing. The result should be like that shown in the *Stereo microphone preamp routed.TPC* file.

Finally press the ⬛ button or select *3D View* from the Tools menu to display the result in 3D.

DesignSoft

PCB
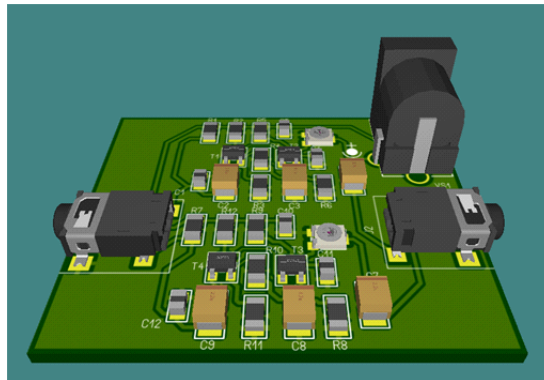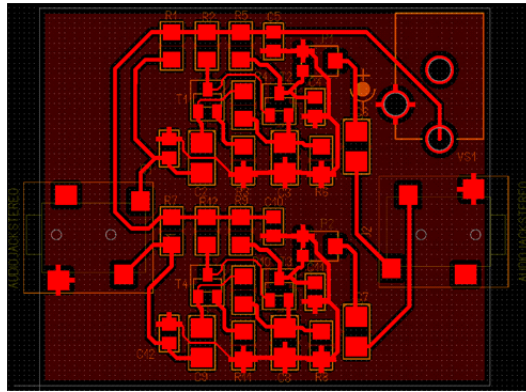
5.4 Creating a Two-Layer, Double-Sided, Surface -Mount Technology Board

To get into TINA in more detail, open the second example, the file *PIC flasher origin.TSC* project from TINA's *Examples\PCB\PIC Flasher rigid* folder. Press the 2D/3D View button in the toolbar and look at the schematic of the flasher.
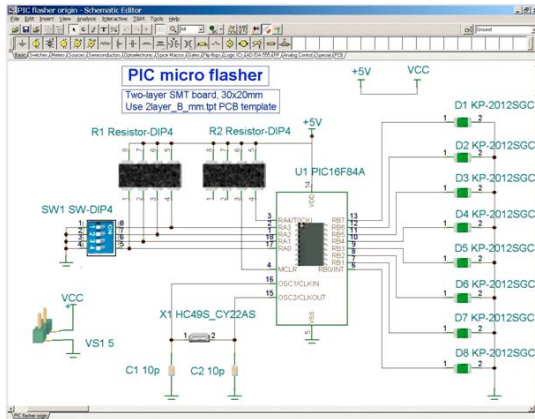
The schematic is PCB-ready—as you can verify by pressing the 2D/ 3D button, every part has a surface mount device (SMD) physical representation. Now, before we start the PCB wizard, click Tools/ Footprint Name Editor… to check the footprints list.

Since all the components appear to have a valid footprint name, we can start using the Tools/PCB Wizard. Set the "Start new project," check "Autoplacement" and "Use board template." Browse for the template file "2layer_B_mm.tpt."

Review actual physical parts, if possible. Be sure to allow for the area of all the components, mounting holes, and keep-away zones and make your best estimate of the values for Board width and Board height. Moreover, it is important to provide enough space between the components to allow for the placement of vias and tracks during routing. Enter 40mm length and 30mm width.

PCB

DesignSoft
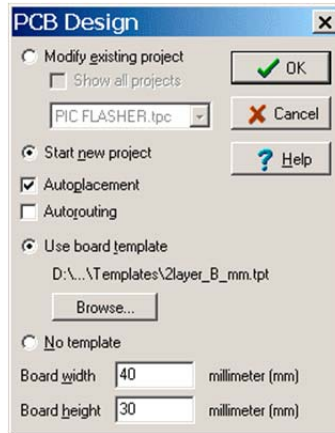
Press the OK button, ignore the Electric Rule Check warning, and save the board as PIC flasher.tpc.
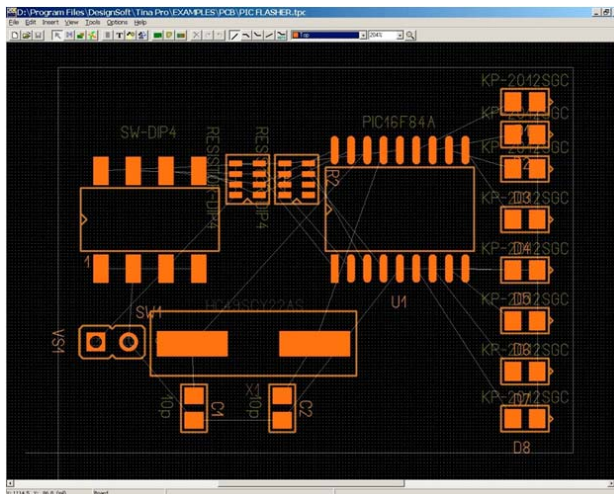
The components are placed in the close proximity to minimize the connection lengths, in a topology similar to that of the schematic, while still respecting the design rule settings. However convenient the result may be for autorouting, the designer usually has to make adjustments to the component layout to satisfy electrical, mechanical and other characteristics. Some of the considerations are -

• the ohmic effect of a long and/or thin power trace the length of a track from the signal source to the load in high-speed digital systems introduces reflections

• in analog situations, poor placement can lead to increased noise coupling

• allowance for automated parts placement clearance

• future serviceability of the PCB

• aesthetic values

These considerations influence the components' position and could be critical–not only in complex designs–but even in the simplest ones. For these reasons, one must still adjust parts placement manually.

Our circuit, although it is small and not very dense, has a f e w special requirements, namely to put the crystal closer to the microcontroller, t o position the power supply connector, and to adjust the LEDs along the board.

If you want to change the board size, click t h e "Draw/modify shapes" button 🔲 Let's try making the board smaller. You can double click on the middle of the board and enter the following values into the fields:

Shape properties

Shape type:	Board outline ▾

☑ Rectangular

Rectangle width: 30 (mm)

Rectangle height: 20 (mm)

Copper shape settings

Assigned net: <none> ▾

Plane layer insulation width: 0.254 (mm)

Plane layer priority: 0 ▾

✓ OK   ✗ Cancel   ? Help
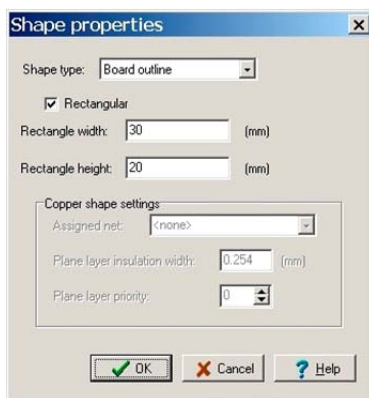
When you press OK, the board outline will shrink.

Now, before we begin routing, let's set the position of the components; place the power connector and the DIP switch on the left hand side and the LED bar on the right hand side. Place the capacitors and resistor networks on the bottom side by double clicking on the components and choosing Bottom Placement side on the dialog.
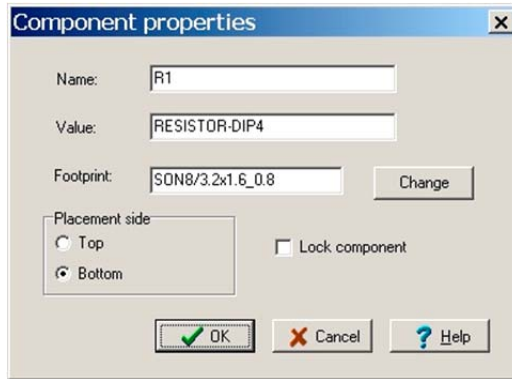
Compare your result to the file \Examples\PCB\PIC Flasher rigid\PIC flasher placed.tpc.

PCB
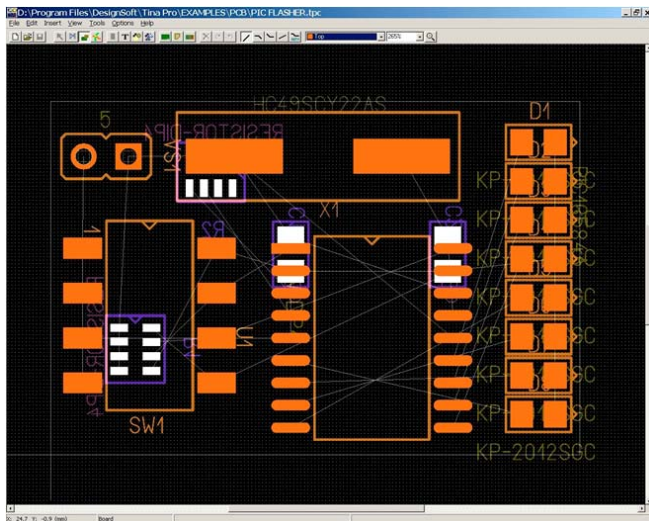
In order to decrease route length, swap R1 pins connected to SW1. Pin swapping is allowable if identically functioning pins are to be exchanged, such as pins of resistors, capacitors, etc. Click on the Pinswap button on the toolbar to pick up the tool, click on pad 1 of R1 (the upper right), and–finally–pad 4 of R1. The following window should come up:

DesignSoft

Press the Yes button, then do the same with R1 pad 2,3 and R2 pad 3,4.

Note that a pin swap changes the original connections, so we must update the original schematic later to maintain the correspondence between the schematic and the board taking into account the changes we made to the board while using PCB Designer. This process, called backannotation, has to be completed whenever a pin/gate swap and/or component renaming have been performed.

Signal planes (in our present example, the top and bottom layers) generally are given preferred trace directions up and down in one layer and left and right in the other layer. To s e t thes e tr ace dir ections , click o n Options/Autorouter settings. The Autorouter settings a r e set to our preferred direction by choosing an integer number from 1 to 9. A value of 1 forces the router to heavily emphasize horizontal lines, while a value of 9 forces it to use vertical lines. 5 tells the router not to care about horizontal or vertical preference. Choosing the extreme values (1 and 9 for a pair) is usually too strict, so we choose to enter 3 for the top layer and 7 for the bottom.
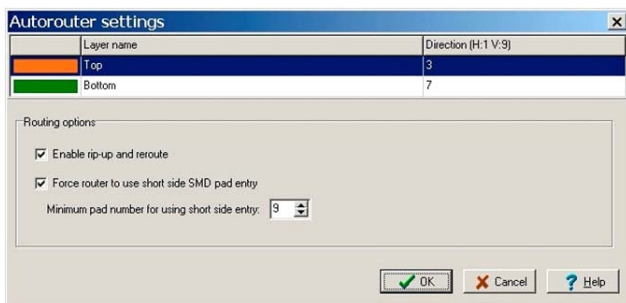
Also on this panel, check "Force router to use short side SMD pad entry" with pad number 9 constraining the router to connect SMD pads on their short side if the component has at least 9 pins. This option can be very used to preserve the gaps between SMD pads

PCB

DesignSoft

for track routing. Because of these choices, R1, R2 and SW1 are allowed to route freely, as half of the pins are connected together, while U1 pads connect on the short side.

After these moves, press Ctrl+F5 to route the PCB automatically. You can follow the autorouter a s it ma ke s the electrical connections among the components of the entire boards. First the power and ground nets are connected, then signals are routed. If necessary, the program will rip-up tracks and reroute the unconnected nets. The following result will appear:
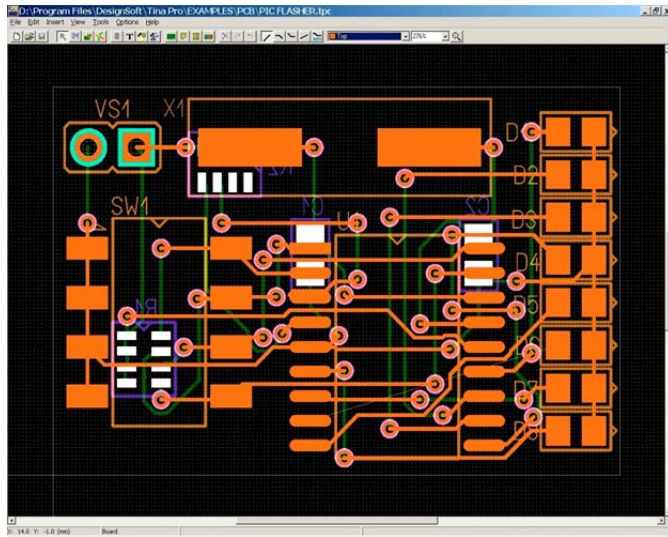
You can seee that there is a net left unconnected. (Note: this short-coming will be removed in later (relative to February 2006) versions of PCB Designer. If you are using a later version, there may very well not be any unconnected traces). Fortunately, the DLC utility will examinine your design and reveal any unconnected traces. Run DRC by pressing F7:

the DRC Results window will appear, highlighting the corresponding nets.
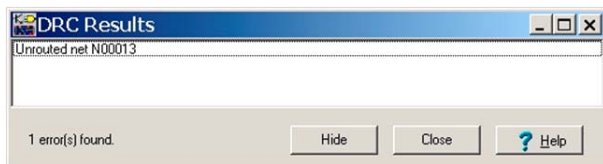
You should browse and correct any errors DRC reports. If you double click on the text 'Unrouted net N00013,' the program magnifies and highlights the selected net while centering the position of the selected objects.
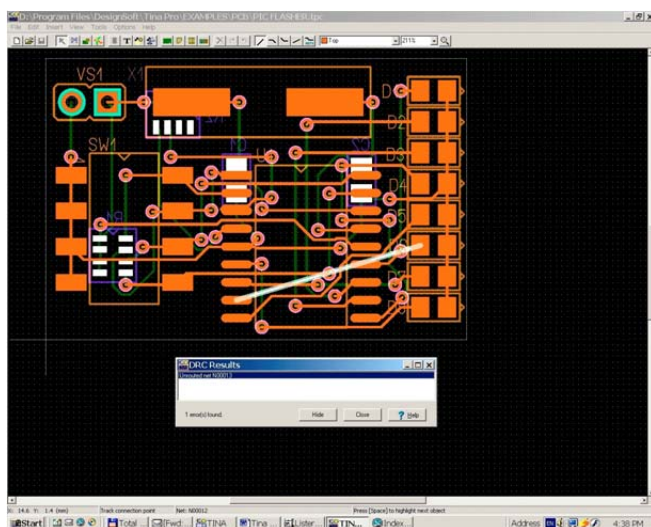
In order to fully route the board, we c a n use the manual route modes to make space for the unconnected tracks or delete crossing tracks and reroute them in a different order. Working in manual route, you can direct routing wherever necessary. You can move existing segments of tracks, remove segments, or create new segments. Proceeding step by step, we will use autorouting to route the last trace.

First, press the , the select button on the toolbar, then hold down t h e Shift key and click the track t h a t connects U1 pad 9 to D5 (see picture below), then press t h e Delete key to remove the track. This makes room for the unconneted net.

DesignSoft
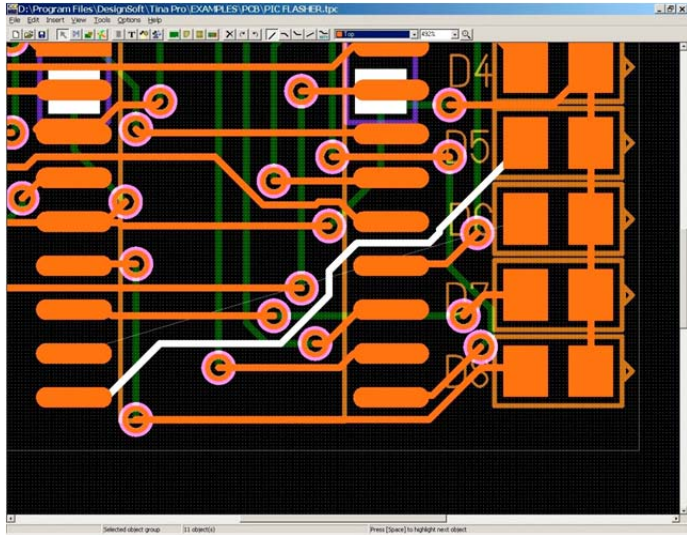
Now press F4 to invoke the net editor. Disable N00012, the previously removed connection. This will prevent the autorouter from reconnecting the net. Leave N00013 routing enabled.
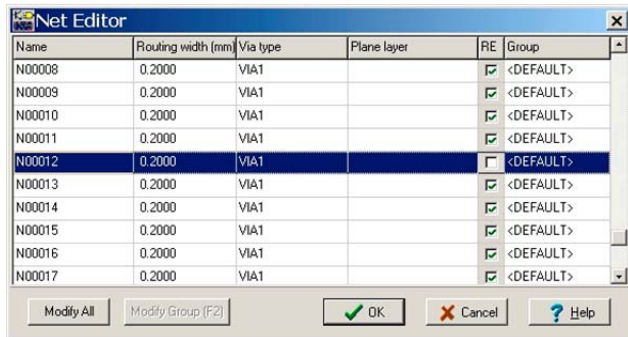
Click OK and press F5 and the autorouter wil connect all the enabled nets. Next, enable N00012 by pressing F4, checking RE, clicking OK and pressing F5 to autoroute the last net.
Let's see if our manual intervention has worked. Press F7 to perform DRC to verify that there are no errors.

PCB

DesignSoft
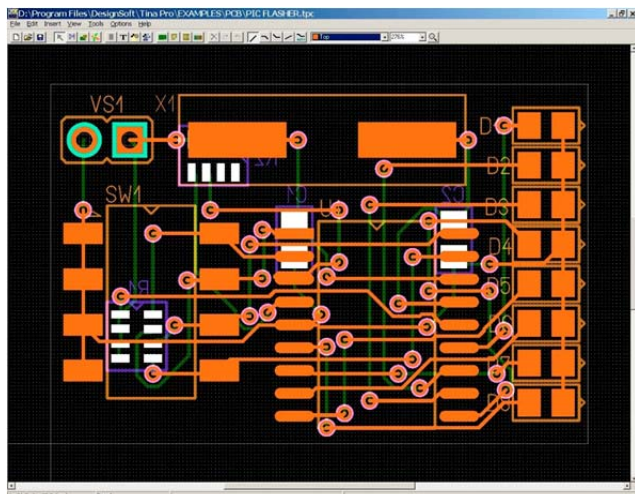
Now, let's synchronize the PCB and schematic files. First, choose Export Tina backannotation file from the File menu to save the changes into PIC flasher origin.ban. Start TINA Pro and click on Backannotate… under the Tools menu and select the same file you have just saved. Click OK and TINA will read and update the original schematic.

Take a close look at the resistor networks–the pin order reflects the result of the pinswap.
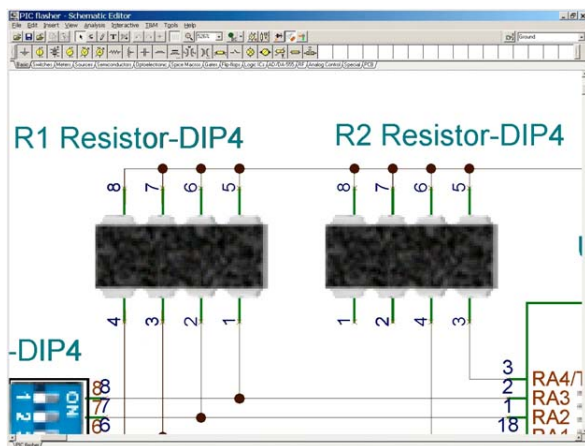
You can get to the *PCB* Information dialog by doubiing dicking on Rl and pressing **"..."** in the Footprint Name field. In the right hand window, TINA presents the swapped nocles in larger, bold type. Save this file as *PIC flasher.TSC.*
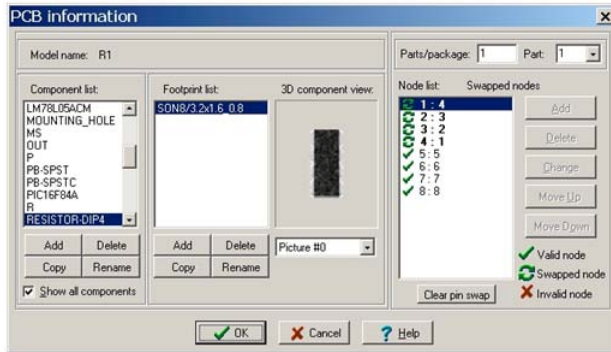
5.5 Creating a Flexible PCB Layout (Flex PCB)

Flex PCBs are PCBs whose lectronic devices are mounted on flexible plastic substrates. They are widely used in modern electronics where space is a critical factor e.g., cameras, mobile phones, etc. TINA supports Flex PCB design, which we will introduce by way of an example. Our example will consist of a conventional rigid PCB with two flexible extensions.

Let's start by loading the example file „PIC Flasher DIP4SW flex top.tpc" from the EXAMPLES\PCB\PIC Flasher flex folder.

Our design requires two flexible extensions or "paddles." One paddle mounts the DIP switch and the second paddle accepts eight LEDs. (For reference, check our final result „PIC Flasher DIP4SW flex top.tpc" from the EXAMPLES\PCB\PIC Flasher flex folder.)

PCB
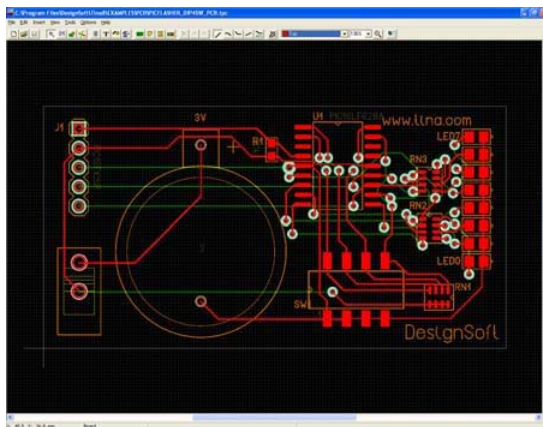
In our current example file, the LEDs and the switch are already routed. To do this tutorial, we'll remove their routing. You can do this manually by shift-clicking the appropriate tracks and pressing the DEL button. Alternatively, you can erase all tracks (Edit/Delete all tracks), then move the switch and the LEDs outside the board (so they will not be routed again) and run Tools/Autoroute board to re-route the other components.
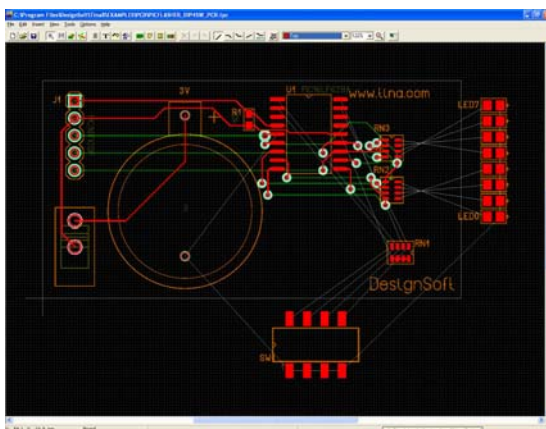
Now we are ready to set up the flexible board area. First, we have to decide whether to put the flexible part on the top or the bottom of the rigid PCB. In our case, all the SMD components are placed on the Top side, so let us also put the flexible parts of our design on the Top.

DesignSoft

Note that we could just as well have placed the flex PCB on the Bottom layer of the rigid board. In this case, the rigid shape should be assigned to the Top layer and the flex shape to the Bottom. SMD components that go to the flex paddles should be left on the Bottom side. Routing on the flex areas is also done only on the Bottom side. Otherwise the two cases are similar. (For reference, check the final result „PIC Flasher DIPSW4 flex bottom finished.tpc".)

Before creating the flexible board shape, click on the Draw/Modify shapes button to enter Shape mode and double-click on the rigid board to edit its properties. At the bottom there is a list of layers, each of which can be assigned to the board. Make sure the Bottom layer is assigned to the rigid board but the Top is NOT (the rectangle next to it is empty), because we want to assign that layer to the flexible board shape. As for the other bottom layers (e.g., Solder Mask Bottom), they can be left on.
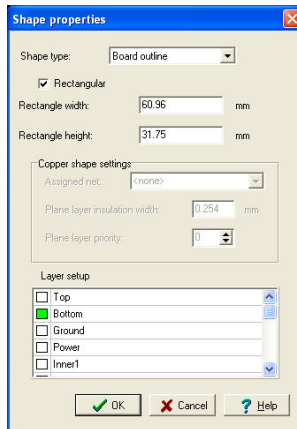
There is a more convenient way to draw the flexible board. Go to Options/System settings and set the system grid to 1 mm. (note: the rigid board was also drawn on a mm grid.)

Even though we want two paddles, we should create a shape as one piece, and that shape should overlap the rigid board because the layers will all be glued together in the manufacturing process. In our current example, the flexible shape should contain the whole rigid shape so we can use it for top side routing on the rigid part too.

To get started drawing a new board shape, select Insert/Board outline from the menu. Insert the first vertex by a single click. Make sure that you are not in the Rectangular shape drawing mode – right-click to see the pop-up menu and de-select the Rectangular mode. Then insert the other vertices of the shape.

We recommend that you give the flexible PCB bent edges. To create a 90 degree bend later, draw 45 degree cut-offs while drawing the shape.

Try to draw the paddles as in the image below. (This phase of the design is in the file "PIC Flasher DIP4SW flex top.TPC".)
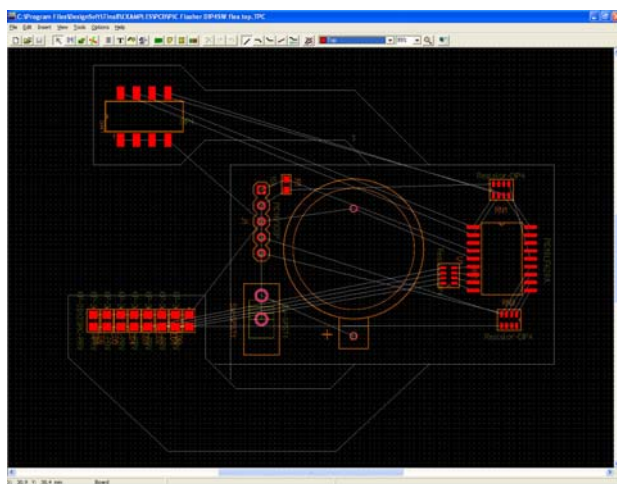
After inserting all the vertices, select Finish from the right-click pop-up menu. If   you spot an error, press ESC and start again. However, you can also edit shapes later (see program Help).

After defining the flexible layer's shape, right-click on the 45 degree edges and select Bend shape edge. Bend it by moving the mouse, then left-click to finish. You could also directly enter numerical data to specify the bend. Double-click the edge and enter the bend angle in degrees.

After placing the shape, double-click on it and edit its properties – switch the Top layer ON and the Bottom layer OFF.

Now that the flexible PCB shape with two paddles is completed, we can place the components on it. In the original example, all SMD

parts were placed on Top, and we should follow suit. Switch back to the first editor mode (Select/Move components/tracks), then double-click on the nine individual components to edit their properties. Move them to their final place.

Now you are ready for the final step, routing. But first, set a finer system grid (0.1 mm) in Options/System settings. Routing is done as in any other PCB design project. You can route manually (right-click on a pad, select Start/Continue track, then draw it section by section) or use the autorouter (Tools/Continue Autorouting, or press F5 button). The Auto track mode (right click while manually drawing a track) can also be useful. Auto track mode allows you to draw the tracks one by one in the desired order and still take advantage of the autorouter. When routing manually, make sure that you only draw tracks on the appropriate side of the flex region (in our case, the Top). Don't route traces on the flex PCB's Bottom side. Only the rigid portion of the PCB can accept routing on the Bottom side.

PCB designers know that traces with right angle (90 degrees) bends can be subject to over-etching with degraded performance. It is good PCB design practice to break right angles into 45 degree angle segments. This applies to traces on flex PCBs as well.

You can bend a track section by right-clicking on it and selecting Bend track. Alternatively, you can select multiple tracks (preferably only those sections that are placed on the flex parts) and use Tools/ Bend track edges on them. This tool takes a given length of the selected tracks starting from the track vertices and replaces them with bent sections. The maximum bend length should be entered after selecting the tool. Its optimal value can be determined by experimenting. If you do not like the results, use Undo (Ctrl+Z) and try again with a different parameter or a different selection of track sections.

Your final result should look like this:

PCB
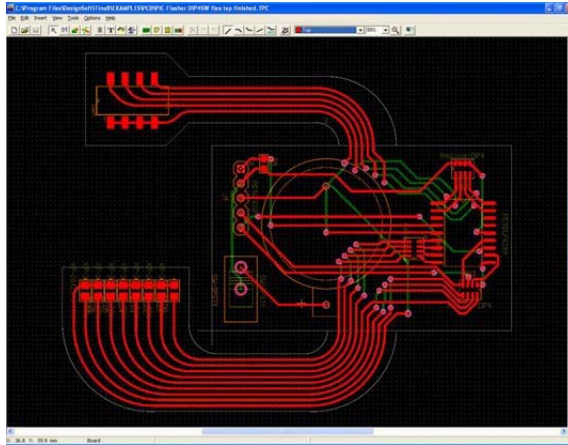
TINA can now present a 3D view of the circuit board. Press the rightmost button (3D View) in the TINA PCB Designer program see the PCB as presented in the next figure.
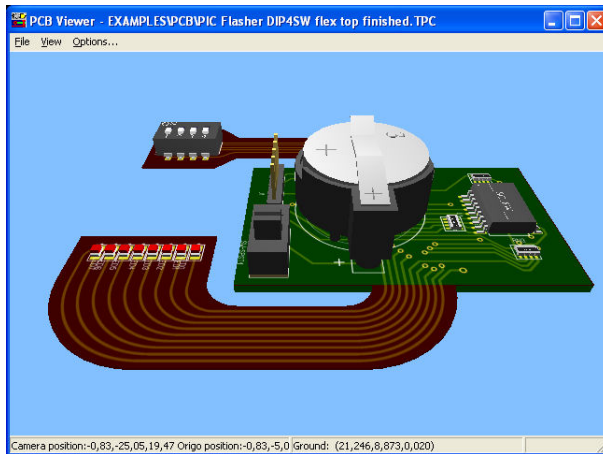
5.5.1 Adding 3D Enclosure to your PCB design

The Enclosure main menu in 3D Viewer contains several options for including enclosure models to the circuit board. As the first step under the Enclosure menu select the "Open 3D model" command to import the whole enclosure, or a part of it. In the next step you can check the dimensions of the model, set its orientation and position.

You can also drag the model to the right place with the mouse. The View menu offers several standard views (Top, Front, Left) for easier over view. Enclosure models can be shown semi-transparent (Enclosure/Transparent menu) to support precise placement.

You can open several models, like in the example below. You can find these examples in the PCB folder of TINA as *USB dongle.tsc*. Open the bottom part of the USB dongle casing, then the upper:
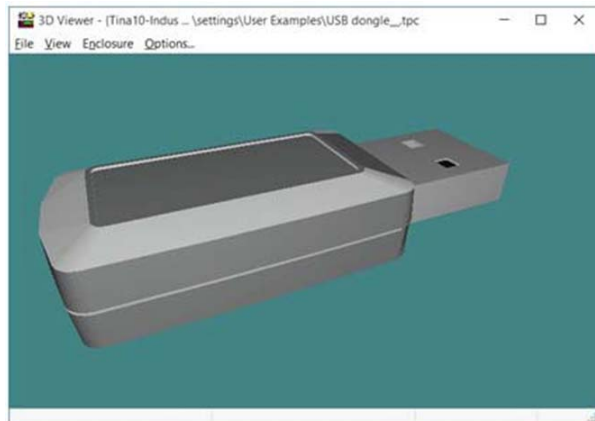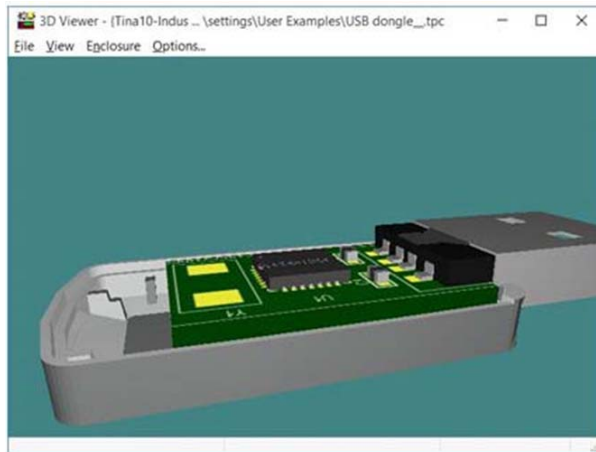
After creating a complete enclosure assembly, you can save it as an Enclosure configuration file (saved with .tenc extension). After exiting and launching ag ain the 3D Viewer the last saved configuration is automatically loaded back.

PCB
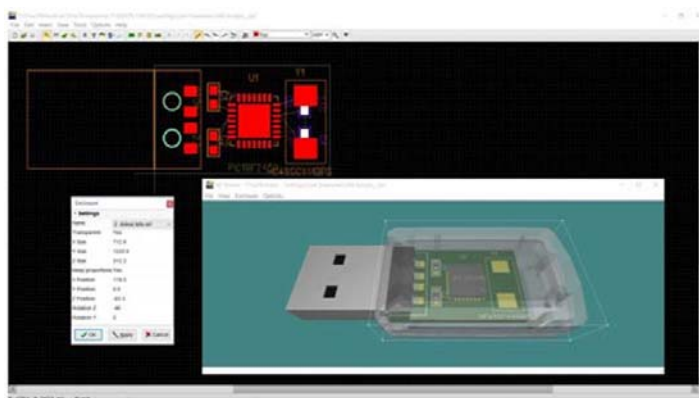
5.5.2 3D Export of your PCB Design

In TINA Design Suite v11 and later version in the PCB 3D Viewer of TINA you can export the 3D PCB model of your design, along with the enclosure if included, in STEP, STL, Google Sketchup (SKP) and X3D format in addition to the previously supported VRML format.

STEP is a widely used industrial format. Please note that it can only handle colors, no surface textures, so the board labels will not be displayed.

Sketchup is a popular format with a huge number of models, available online.

STL is mainly used for 3D printing, it is a plain color format. After exporting from 3D Viewer, your 3D printing application will probably correct some geometrical details to comply with 3D printing rules, after this you can have the board printed in 3D.

X3D is the improved version of VRML, an open source format which is supported by many applications.
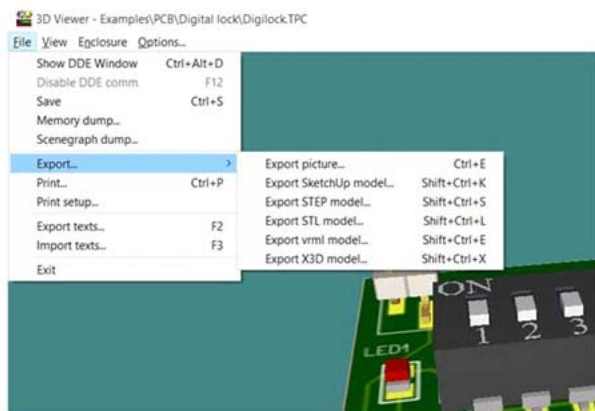
5.6 Creating 4 Layer PCB Layout

This chapter provides an introduction t o 4 layer PCB design. We will follow the entire process and design a medium size circuit PCB. This will demonstrate the major concepts and introduce the settings, techniques, and tools required. We will emphasize the concepts where they differ from the previous single and double layer designs. Many phases of the example can be found in the \Examples\PCB directory, such as:

The block diagram of the circuit we will design is shown in the

Schematicfiles	
FPGAorigin.tsc	original schematicfile
FPGA.tsc	schematicfile(backannotated,afterrenumbering)

PCBfiles	
FPGAplaced.tpc	designparametersset,componentsplaced
FPGAplacedsplit.tpc	asabove withpowerplanesplitting
FPGASpartanpowerrouted.tpc	netpropertiessetand SpartanFPGAchippowerrouted
FPGASpartanpowerroutedsplit.tpc	asabove withpowerplanesplitting
FPGAallpowerrouted.tpc	powerrouted
FPGAallpowerroutedsplit.tpc	asabove withpowerplanesplitting
FPGArouted.tpc	all connectionsrouted
FPGAfinished.tpc	optionallypin/gateswappedandrenumbered,routed, silkscreenadjusted,documentationlayersfinalizedpcb file

figure below. The main component is a field programmable gate array (FPGA) from Xilinx, Inc. In this circuit, we will use the FPGA with a few pushbutton switches as inputs and LEDs as outputs.

Many of the FPGA IO pins can be used freely for general purposes and are brought out on the connector J1.

The board contains a power connector for an external 5-volt supply, a programming interface for the FPGA, and some miscellaneous resistors and capacitors.
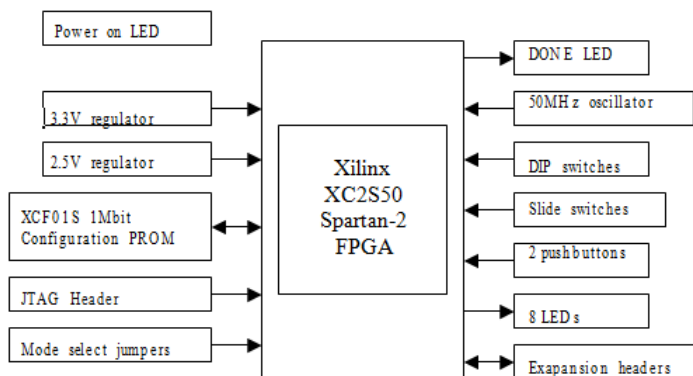
For the complete circuit diagram, open FPGA origin.TSC (the backannoteted version saved as FPGA.tsc.) All the footprints are named, the schematic is ready-to-use for layout design. Just click on the PCB Wizard… set "Start new project", check "Autoplacement", and use board template 4layer_C_mm.tpt (work in metric units because the components' physical dimensions are defined in the metric system). For the best routing performance, taking the smallest pitch size (0.5mm) as reference, we set the Grid size parameter to 0.1mm which will control the visible grid, the parts placement, and routing grids.

Enter the width (110mm) and height (60mm), then click OK.

After exporting the netlist from TINA, you should check a few parameters in t h e PCB editor. First, select Option/System setting. Make sure that PACKAGE.FPL is checked.

5.6.1 Placing parts

To make the screen less cluttered, let us turn off a few layers at this point. Select Option/Layer Settings, press Uncheck All, and check Assembly Drawing Top and Bottom layers.

You are now ready to place parts on your design. To get into parts placement mode, make sure that the Select/Move components/tracks tool is selected. A good starting task is to place the non-electrical components on our board, the mounting holes in each corner.

Every non-electrical component (mounting hole, rubber foot, enclosure…) symbol has to be placed on the TINA schematic before making a netlist. In the case when no footprint should be placed on the board (e.g. rubber foot) then the componet has NOPCB footprint name in TINA schematic.

Now, place the mounting holes at the corner.

Since we don't want to accidentally move the part, right click, select Component Properties and check the box labeled Lock component.
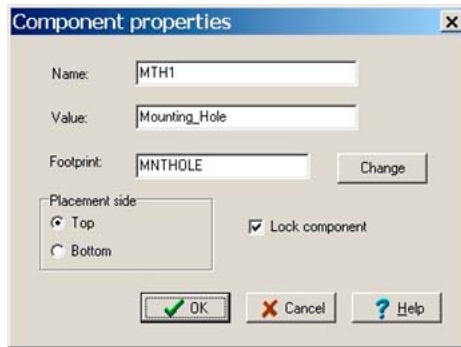
Now you can place the rest of your components. You will probably want to print out your schematics so that you can see where the components are supposed to go in relation to each other. When you pick up a component, the airwire for that component will appear to show you the connections to other parts.

When placing components, you may want to work on a coarser grid. Right now, the grid is set at 0.1 mm. You can change this by selecting Options/System Settings and then changing the Grid setting.

Start placing the components on your board. Begin by placing the power connector and voltage regulators on the right side of the board. Allow enough room for the power dissipating copper area. Place the FPGA in the left center with the 40 pin connector above, and LED and switches on the bottom. Try to keep components that belong together near each other.

Put the buffer and filter capacitors of the FPGA and flash memory on the bottom side of the board as close as possible to power pins of the chip, the smaller ones right under the leads, the bigger ones around the middle of the package.
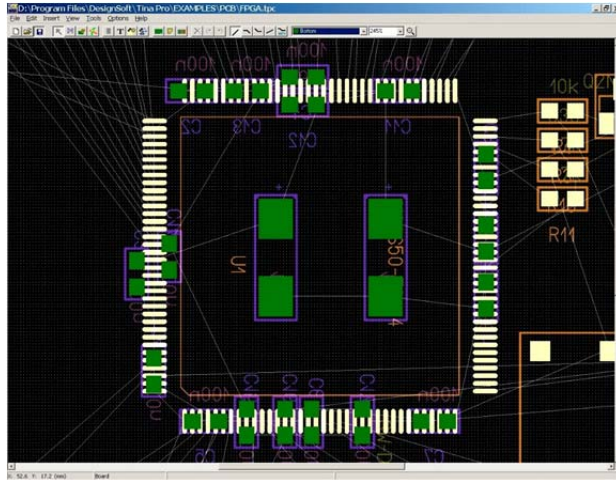
To place a part on the opposite layer, right click on it, choose Component properties and select bottom placement side.
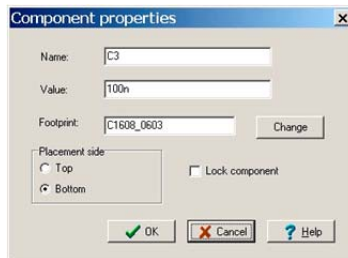
When you are done, your board should look something like this.
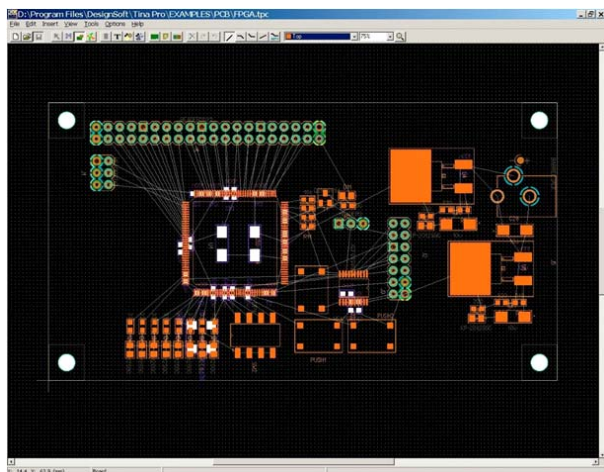
The silkscreen is a bit messy, but we will deal with that later. In fact, during routing the silkscreen can get in the way, so you may turn off the silkscreen and assembly layers now, just as you did earlier to suppress the two inner layers.

Now, you may save your design as FPGA placed.tpc.

5.6.2   Draw copper areas for voltage regulators

(Go on with your own created file or open FPGA placed.tpc in the \EXAMPLES\PCB directory.)

Copper areas can be used for noise suppression, shielding, to draw heat away from components that tend to get hot, to isolate signals, or to provide small voltage planes. Now, to enhance the maximum power dissipation of the voltage regulator circuits (U4,5), we draw a 15 by 15mm area of copper on the top layer to widen the regulator heatsink surface.

Choose "Add Cooper pour" [icon] icon from the toolbar then click on the upper left corner, drag the mouse towards the lower right hand corner, and finally click the left button.

PCB

DesignSoft
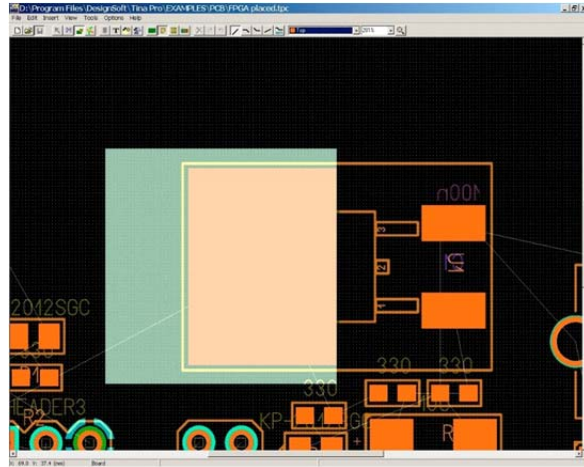
When you double click on the area, you can enter the parameters exactly. Do not forget to check the shape type and select Assigned net 2.5V. Normally, copper pour creates voids where there are tracks or pads except for the Assigned net. (See Options/Design Parameters). The other shape type is the copper fill area which is solid.

PCB
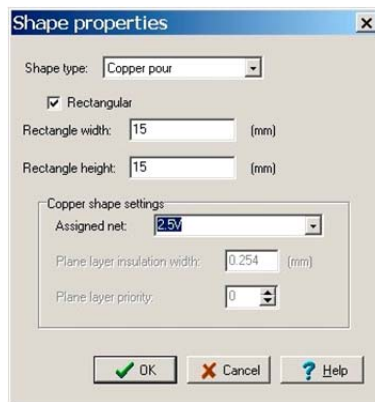
Now click on the copper pour, hold down the left mouse button and position the rectangle, expanding leftward to overlap the heatsink.

DesignSoft
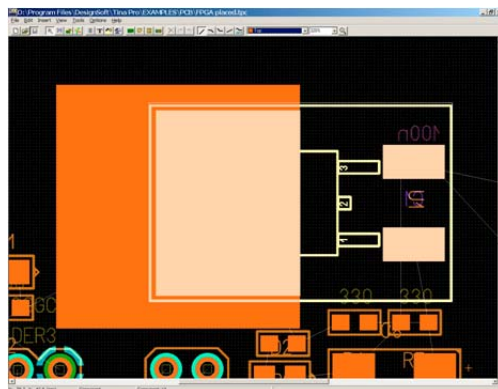
Finally, repeat these operations for the other voltage regulator and save your design. Note, you can find this state of the design if you open \Examples\PCB\FPGA placed.TSC.

5.6.3   Assigning and routing Ground and Power

In any design, it is usually wise to route all power and ground connections before doing anything else. On a thru-hole technology board, this is very easy because connections can be made to the solid plane as the pins pass through the board. On a SMT board, the power and ground pads need to be routed by vias to the appropriate plane using "thermal reliefs". Usually, the designer enables routing for the power and ground nets, while disabling routing of all the other signals. After routing power and ground nets, the designer will disable them and enable all the other signals to route the remaining signals.

Before routing, we need to set up our design so that the PCB editor knows that the two planes are associated with particular nets. (Interior plane layers are typically used for power and ground.) The most convenient way to establish the mapping of planes to netlists is to assign a label to a wire of the net (as we did) while in the TINA schematic editor. We have already placed such labels in FPGA.TSC at the output of the power regulators. (See FPGA origin.TSC at X:485, Y:880; X:440, Y:930) The GND label will inform TINA PCB that the solid plane on layer 2 (Ground) is asssigned to the net, while the PWR label designates layer 3 (Power).

You can handle other nets in the same fashion, by double clicking on the wire and modifying the ID in TINA schematic.

Note, only GND and PWR IDs have special effect on the plane layer allocation; other net names are just for clarity to help y ou recognize the source of the nets in the PCB editor, eg. 2.5V in this design. It is always advisable to name the significant nets for easier operation later when using the PCB editor.

Open the Net Editor and check the result.

Name	Routing width (mm)	Via type	Plane layer	RE	Group
GND	0.2000	VIA	Ground	☑	<DEFAULT>
PWR	0.2000	VIA	Power	☑	<DEFAULT>
2.5V	0.2000	VIA		☑	<DEFAULT>
5V	0.2000	VIA		☑	<DEFAULT>
N00003	0.2000	VIA		☑	<DEFAULT>

PCB
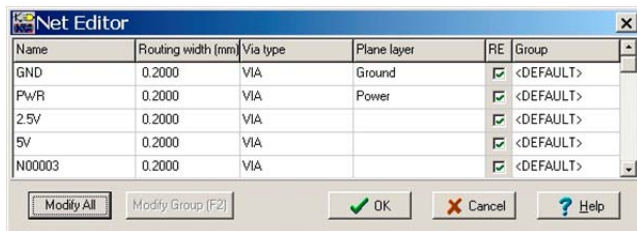
You can attach any nets to any plane layer in PCB editor also, but it is easier to do this in TINA schematic as described.

To view plane layers, click Cancel while in the Net Editor and choose Option/Layer settings, press Uncheck All and check Ground. The thru-holes are already connected by thermal reliefs.
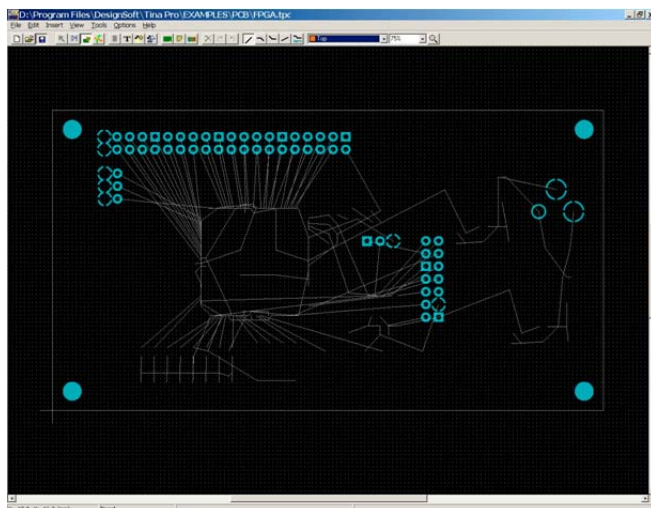
You can now see all of the connections to the ground plane. You can do the same thing for the power plane: Option/Layer settings, uncheck Ground and check Power.

At this point, we have only routed the thru-holes for power and ground connections. On a surface mount technology board, we should fanout[1] the board with only the power/ground net enabled. In the Net Editor (F4) press "Modify All" button to uncheck "Enable autorouting".
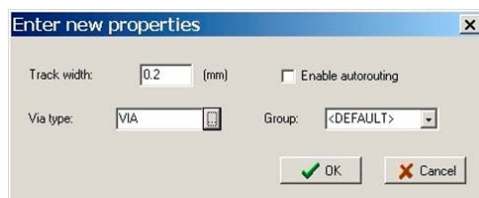
Select the ground and power nets and create the net group POWER; simply type P O W E R in the Group field. Enter 0.4mm to Track width and check Enable autorouting with via type PWRVIA.

---

[1] Fanout is the process of routing a SMD pad to a via so that the pad can be routed on other layers. For power and ground pads, the fanout is attached to a power or ground plane using a thermal relief.
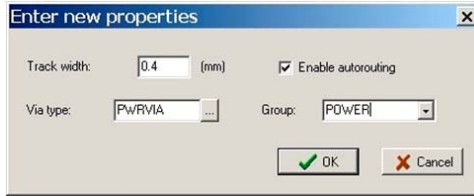
The Net Editor should look like this.
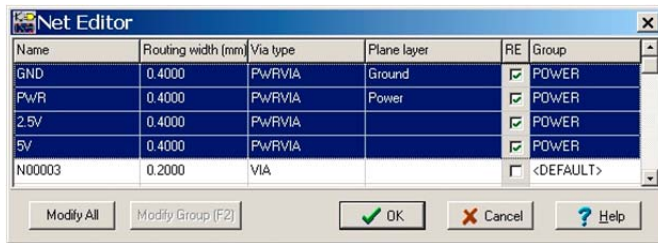
As we have a lre a dy use d two plane layers and still have one more voltage net to route –

the 2.5V regulator output going from U4 to the FPGA core– we need a dedicated trace for the 2.5V net. This net touches only a few pins, so we could route a trace to connect all the components. Click o n t h e Draw tracks tool and route 2.5V under the Xilinx chip package onto the bottom side.

PCB

257
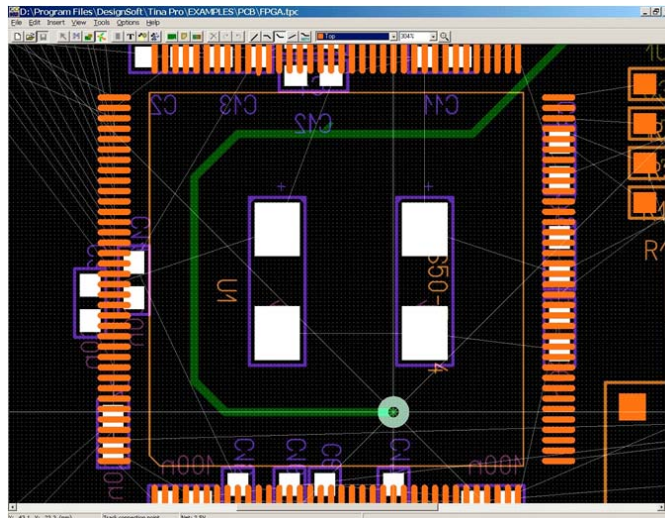
Connect all the power pins of the FPGA, then the pads of filter and decoupling capacitors; set the track width to 0.4mm. Now make the power track 2mm wide as shown below by double clicking on the track.
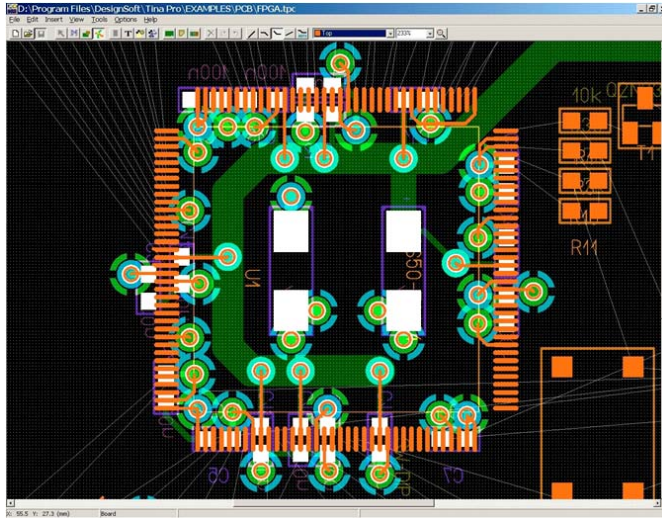
Next, route all the nets from the POWER net group. When you are manually routing, you can begin/end a new track on another track of the same net, which is known as T routing. After all that press F7 to run DRC. You will not see any unrouted nets belonging to this group (a good thing!).(See FPGA power routed.tpc as reference.)

5.6.4   Finishing the routing and post processing

Disable routing on the POWER net group and enable it on the other signals.

PCB

DesignSoft
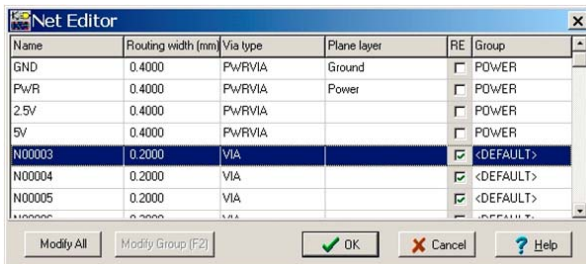
Set the autorouter in the Options/Autorouter settings as follows.
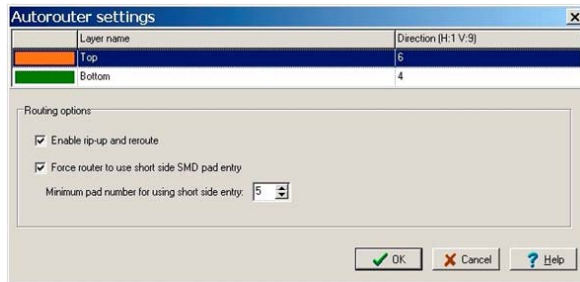
Check the settings under Options/Design Parameters.
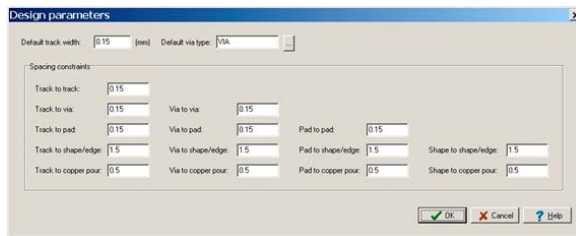
Press F5 to autoroute the whole board, then renumber components (Tools/Renumber components, F10). Renumbering begins at the upper left of the board, renames components in a sweep from left to right, then moves down and renames in successive sweeps.

After renumbering, backannotate the design, as you did in chapter 4.8, and save the new schematic. To check the result, refer to FPGA.TSC and compare.

5.7   Creating Split Plane Layers

Power layers are used to provide electrical power references and a stable ground throughout the board. In systems with multiple power supplies, the power planes, which are typically solid copper internal layers, are split. When you split the plane, you assign a part of a plane layer to a second net (e.g.: processor core voltage) by placing a copper pour onto a plane layer. You assign a primary net to the whole plane layer as we did by the GND and PWR IDs in TINA Pro, and a secondary net to the copper pour. Now, let's see

PCB

how it works within TINA PCB. Open the "FPGA placed.tpc" as a starting point to our work. Remember that the PWR net – the 3.3V primary supply voltage– has already been assigned to the plane layer Power. Now, you split a portion of that plane for the FPGA core voltage (2.5V) instead of routing a net from U4.

First, it is important to select the power layer, then click the Draw/ modify shapes ▗ and Add copper pour area 🄲 buttons. The cursor turns into a cross. You can begin to draw the outline of the copper pour. The shape must extend under the regulator heatsink and the middle of the Xilinx chip package to connect them through vias. Using the appropriate zoom factor (e.g.: 200-400%) during the drawing is usually helpful. When you are done splitting, your design may look something like the screen shot below. (Find this phase of the project in the file "/Examples/PCB/FPGA placed split.tpc".)
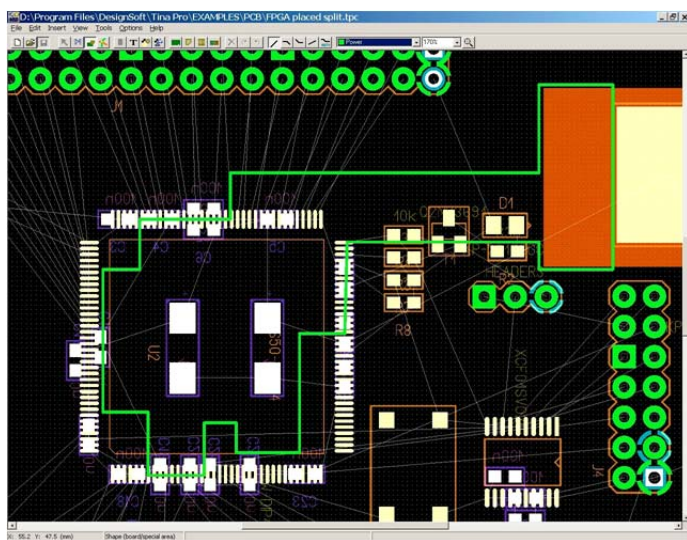
Plane layers are negative layers, so only the isolating „moats" can be seen on the screen. Before beginning other tasks, you should assign the 2.5V net to its split area. Double click on the shape, select the 2.5V net, and click OK.
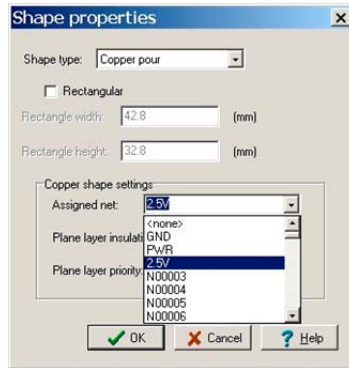
Note that if you create nested copper pours on plane layers which overlap themselves, you must specify the plane layer priority. The higher the priority number, the higher the copper pour sits above the lower ones owning the overlapping region.

If you now establish all the power connections, you will achieve a result similar to "FPGA all power routed split.tpc" shown below. The PCB Designer automatically places thermal reliefs around through-hole pins and vias whenever appropriate.
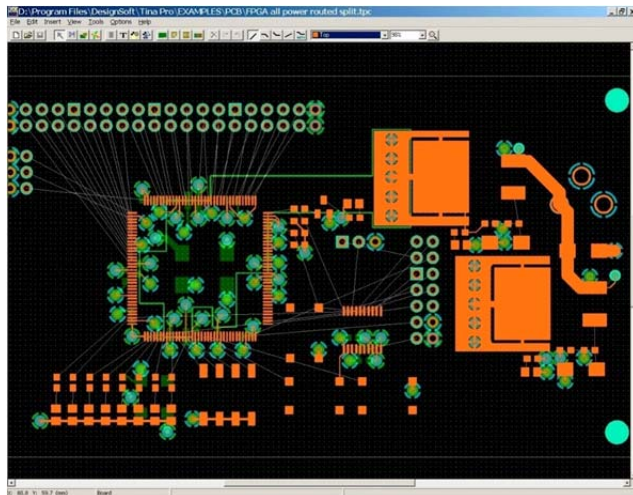
The rest of the process and the movements are similar to those which we applied in the example with the unpartitioned plane layers.

PCB

DesignSoft

USING SHEMATIC SUBCIRCUITS, SPICE AND HDL MACROS AND S-PARAMETER COMPONENTS

In *TINA*, you can simplify a schematic by turning portions of it into a subcircuit. In addition, you can create new *TINA* components from any Spice subcircuit hardware described by HDL or TouchStone format S-parameter file, whether created by yourself, downloaded from the Internet, or obtained from a manufacturer's CD. In this chapter, we show through text and examples how easy it is to do this in *TINA*.

6.1   Making a Macro from a schematic

Using *TINA*'s macro facility, you can simplify schematics and hide clutter by turning portions of the schematic into a subcircuit. *TINA* automatically represents these subcircuits as a rectangular block on your schematic, but you can create any shape you like with *TINA's Schematic Symbol Editor*.

You can convert any schematic diagram into a subcircuit - called a Macro in *TINA* - simply by adding the terminals and saving the new circuit in the special (*.tsm) format.

Now let's see how to create a macro in *TINA* through an example. Load the Half Adder example (*Half_add.tsc*) from the Examples folder of *TINA* and convert it into a macro.
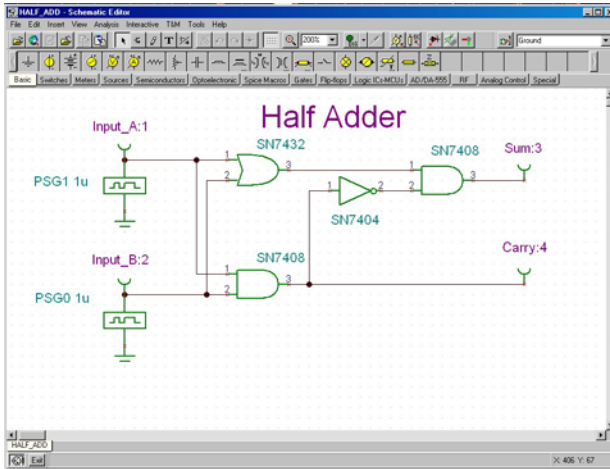
Delete the old terminals and replace them with subcircuit terminals, called Macro Pins in *TINA*. You can find and select the Macro Pins under the Special component toolbar.
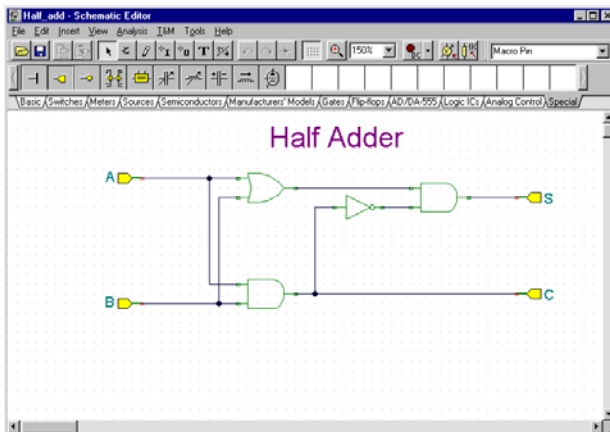
When you place Macro Pins, labels (such as Pin1, Pin2 etc.) are pre-filled in. Double click the Macro Pin and type in the new name in the label field. You can also drag the component with the mouse, or rotate it with the *[+]* and *[-]* keys or the ⟳⟲✚ buttons.

Next, create and save the new macro. Select the *New Macro Wizard* from the Tools menu. Set the Name to Half Adder (this will be displayed in the macro box that opens automatically), and set the Label to HA. This label will be displayed as the component label above the component. Note that you can leave this field blank if you don't want a component label.
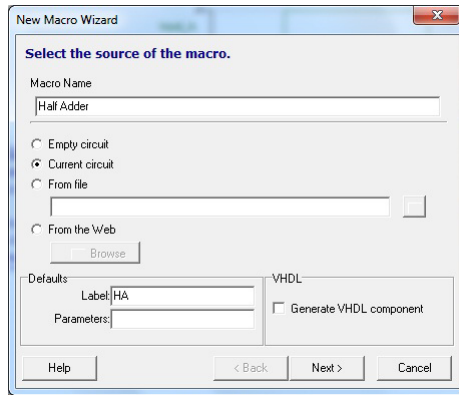
When done, press OK. A Save dialog box will appear. Set Half Adder as the File name and press Save. Note that there is already a macro with a similar name (Half_add.tsm). This has the same content as the one we just created, and is included for reference. You can also use it in the next section.
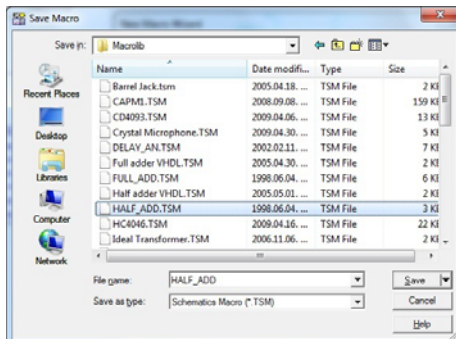
Now let's see how to insert a macro into a schematic and use it.

Clear the circuit with **File|New** or by restarting *TINA*. Select **Insert|Macro**, then our newly created *Half adder.tsm*, and click Open.
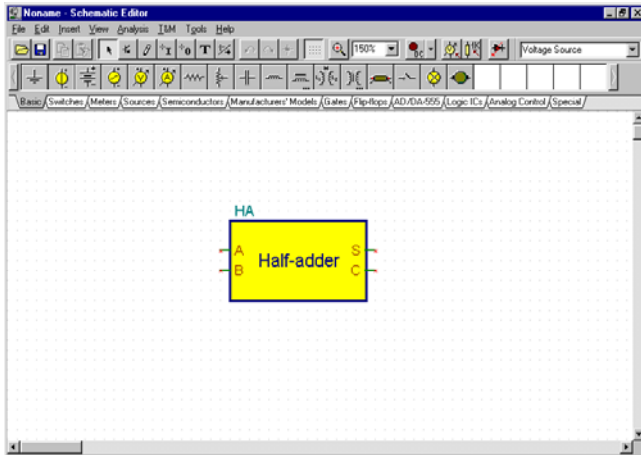
Our new macro will appear as a cursor. Move it to the center of the screen and click the left mouse button. The full symbol of the new macro will appear. Note that a rectangular schematic symbol has been automatically created, the macro name we specified is inside the rectangle, and the label name is above it.

Now you can add more components to the circuit, connecting them to the newly created macro, and start analysis as with any other circuit.

To check the content of the macro, double click on the symbol and *TINA* will display the model.

To return to the main circuit, select the Close command from the File menu.

TINA allows a hierarchical macro structure; that is, macros can contain other macros inside, and so on. Let us use our half adder macro to create a full adder macro containing two half adder macros.
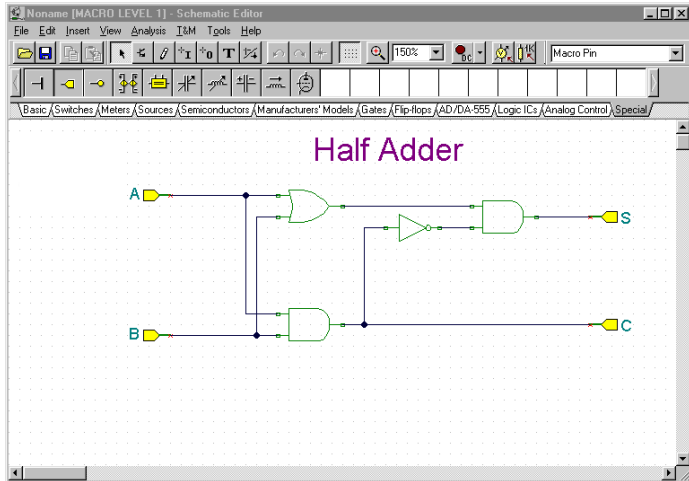
To do this, insert the newly created Half Adder twice into a new circuit and then add the additional components and wires as shown in the following picture.
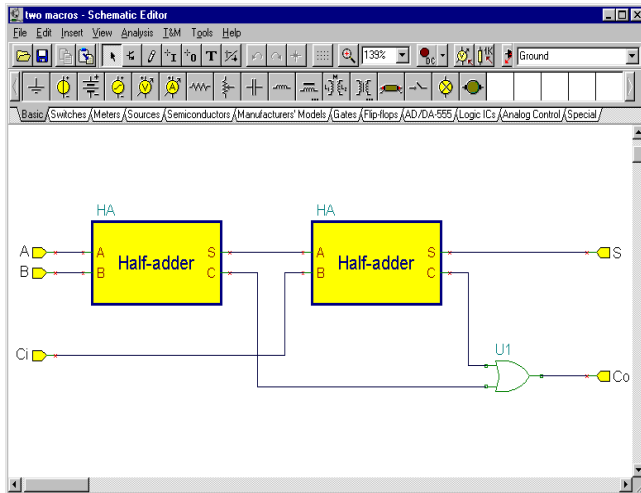
Now create and save the new macro with the *New Macro Wizard* from the *Tools* menu. At this point let us note that although the automatic symbol creation is very convenient, you can also create your own schematic symbols with *TINA*'s Schematic Symbol Editor and assign macros to them. Let us use this feature with an existing symbol. The creation of such a symbol will be described later in detail.

Set the Name to Full Adder and set the Label to FA (this will be displayed as the component label above the shape).
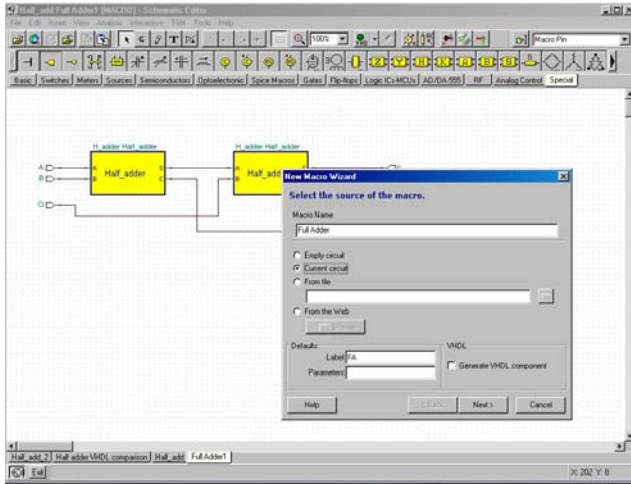
Press the Next button. The list of available symbols will appear in the Wizard as shown below.
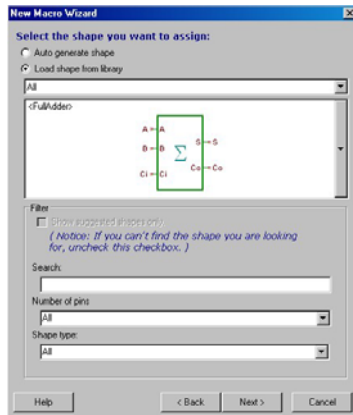
Note that in order to see the predefined symbols, the Macro Pin Label names must exactly match the names in the symbol. In our example, they must be (A, B, Ci, Co, S). If you do not see the symbol shown in the figure above, check the terminal names or try to recreate the symbol as shown later at "Making your own schematic symbols".
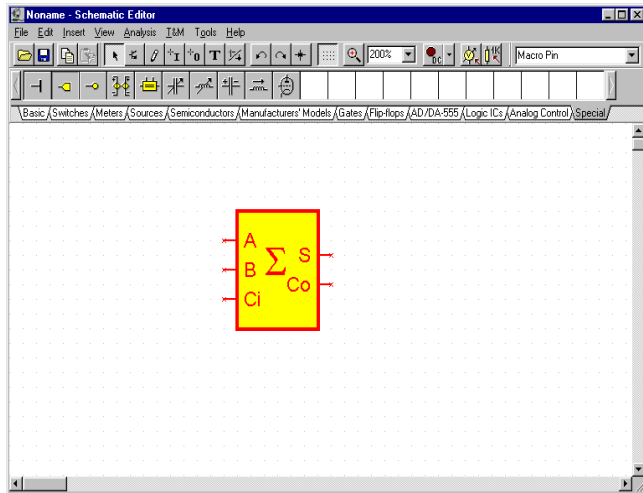
Click the schematic symbol with the large summation sign and press OK. The name of the schematic symbol will appear in the shape field of the New Macro Wizard dialog box. Finally, click OK and save the macro under the name of Full adder.tsm.

DesignSoft

6.2   Making a Macro from a Spice subcircuit

6.2.1   Creating Spice Macros in TINA

6.2.1.1   Creating macros from downloaded files

In *TINA*, you can create your own components from any Spice subcircuit that you have made or downloaded from the Internet. Note that there are already many Spice component models in the large and extensible manufacturers' model library provided with *TINA*. The extension of those libraries is described later.

Let's create a UA741 operational amplifier using a Spice subcircuit.

To do this, select the New Macro Wizard from the Tool menu. The following dialog box will appear:
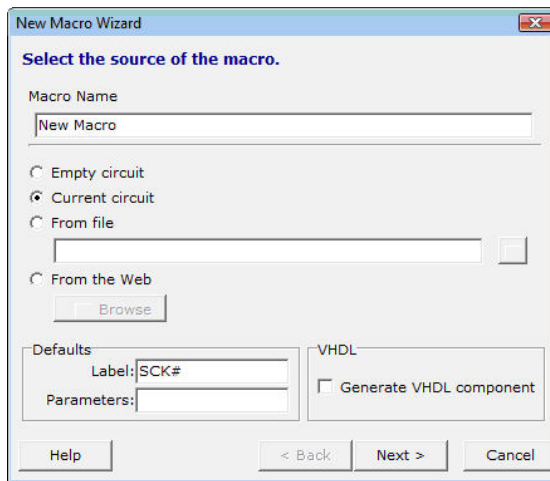
Change the settings from Current Circuit to From file and press the button. An Open dialog box will appear.
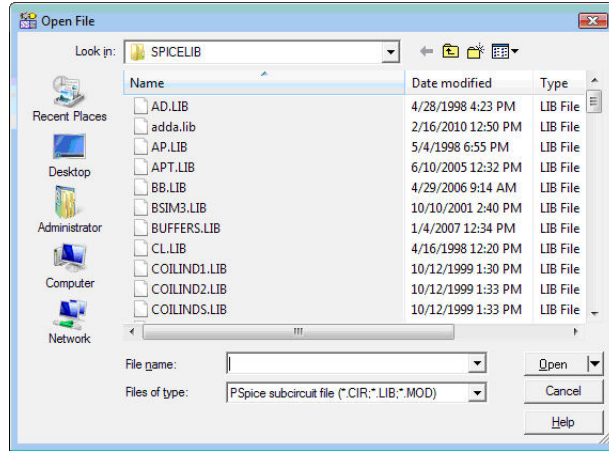
Now lets navigate to the EXAMPLES\SPICE folder of TINA using the small arrow next to the Open button.
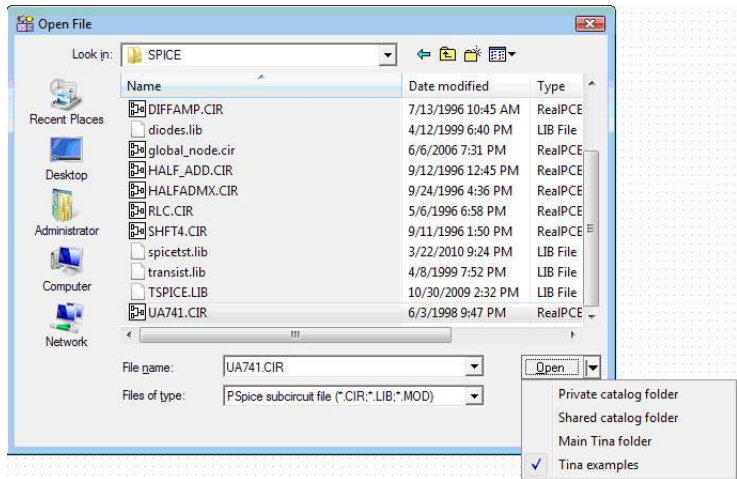
Select the UA741.CIR file and press the Open button. The New macro Wizard dialog will appar again with the path and name of the selected file. Now press the Next > button. The following dialog will appear:

DesignSoft
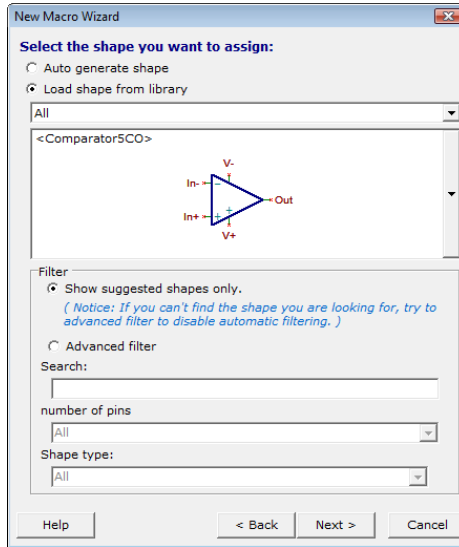
The wizard has already automatically selected the appropriate symbol. If you want something different, you can view and select a symbol by pressing the long vertical button on the right.

Press the Next button. The following dialog will appear:

DesignSoft
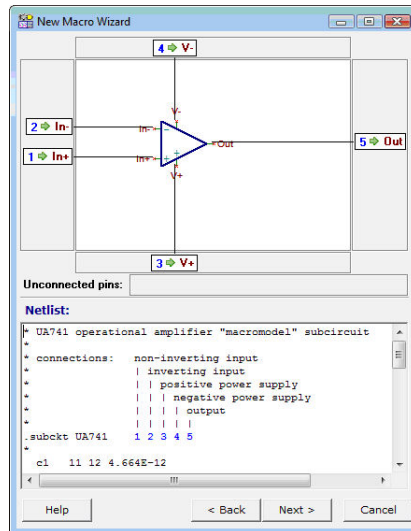
The dialog shows how the pin names on the graphic symbol are associated with the Spice node names in the macro. It also shows the text of the macro so you can check that the connections are properly made. If not, you can drag the macro node names to any terminal. However if all the pins are connected with a node number, then most likely the association is correct.

You've checked the connections; now press the Next button again. A Save dialog will appear and you can save the macro into the User macros area under **Documents\Designsoft\ TINA_Industrial_install date_id_number\Macrolib** or to the TINA macros area, under Program Files. For easier file selection, use the small arrow next to the Save button. Note that under Vista and Windows 7, you cannot normally write into the TINA program area. After saving your macro, you will see a dialog with which you can test the macro or close the wizard altogether.

Now we will see how to insert the new (or any other) subcircuit into a schematic and check its contents. Select the Macro command of the Insert menu.

NOTE:

You may need to select the User Macro or TINA Macro area using the selection arrow next to the Open button, or navigate there using the selection list on the top of the Open dialog.

Click the UA741.TSM file and then press Open. Now the new macro will be attached to the cursor. Position it on the screen and drop it by clicking the left mouse button. Double click the symbol and press the Enter Macro button to see its content. The netlist editor will appear, showing the macro in detail. Note that you can modify this netlist, and the modified netlist will be saved with your circuit. However this will have no effect on the original macro; it remains unchanged.

DesignSoft

6.2.1.2 Creating macros on-the-fly by browsing the web

A more convenient way to add new models to TINA is to browse manufacturer websites and add the Spice models of interest from the web site. Of course it also possible first to download the models and use the technique described in the previous section. Note that even if you prefer the latter you may find useful hints in the following section.

Now let's select the From the web option in the wizard and press the 📁 Browse **button.**
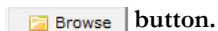
TINA's built-in internet browser will appear. Use this browser to find and select the THS4121 differential opamp from Texas Instruments. Enter www.ti.com and find the Spice macro on the TI web site using TI's Search option , or just enter following URL directly (use copy and paste if you like)

http://focus.ti.com/docs/prod/folders/print/ths4121.html (Note that the direct link above may change).

The THS4121 product page will appear. Scroll down the screen and find the link to the Spice model of this product shown below in red.
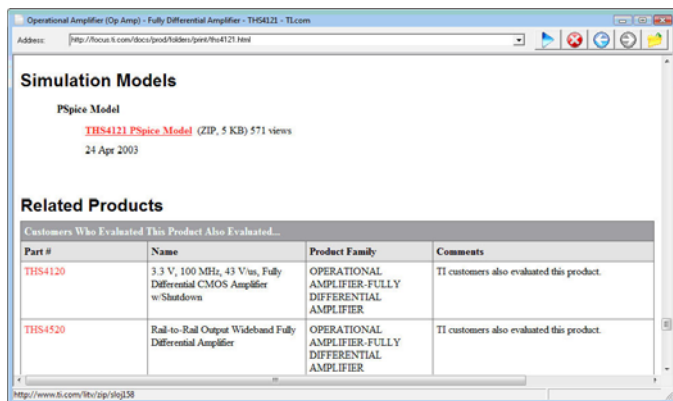
Click on the link. The following message appears in TINA:

DesignSoft
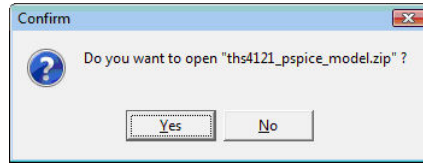
The file is compressed (or "zipped"), but TINA can open and download the files you need directly. Click the Yes button, and observe the next screen.
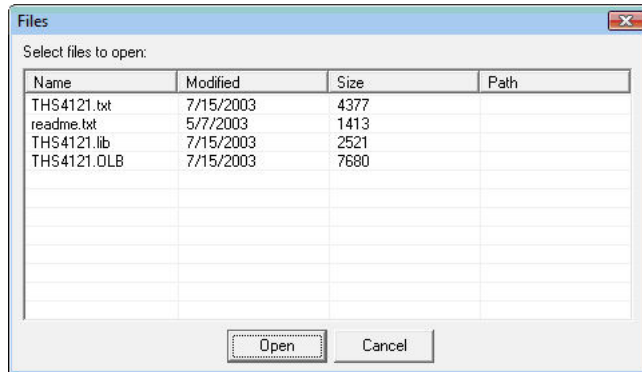
Click to select the THS4121.lib file
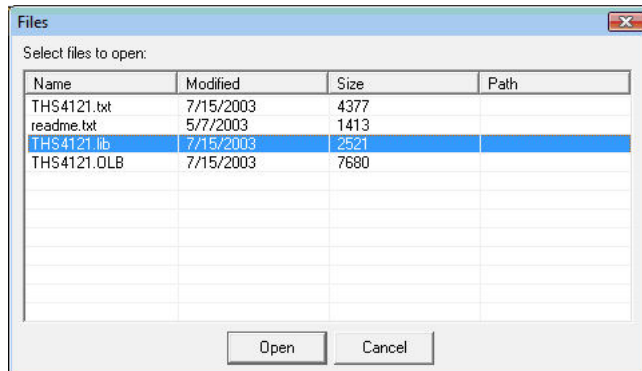
and press Open. The New Macro Wizard will appear again, with a (green) message (File downloaded press Next) confirming the successful download
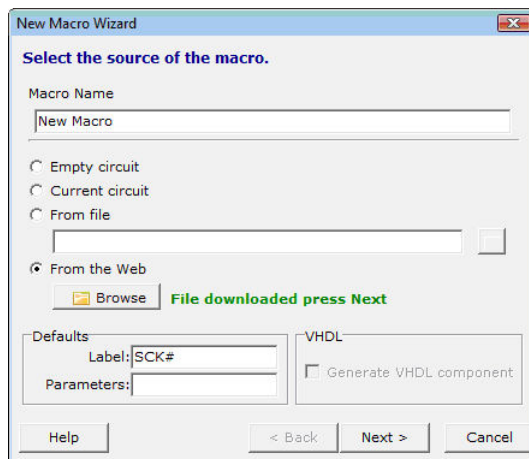
Press the Next button.
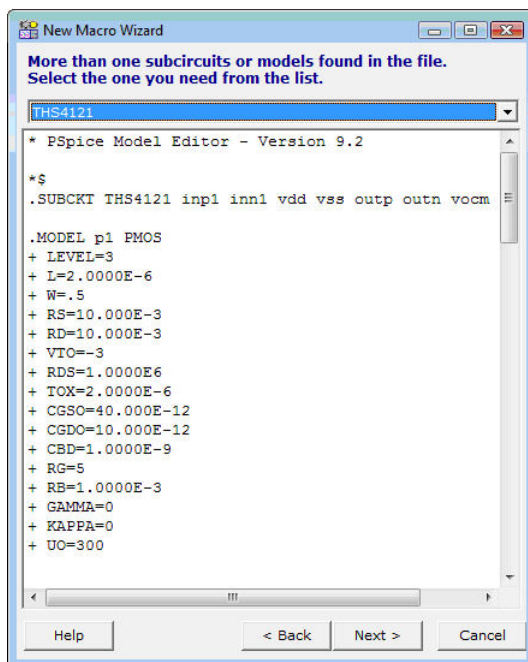
If there is more than one macro in the file, TINA presents them in a list.

DesignSoft

NOTE:

Some manufacturers place several device models in the same file.

You can bring these device models into TINA macros with this tool by selecting them one-by-one.

If there are a lot of models, you might want to use the Library Manager tool which allows you to add all the models into the TINA catalog in one step. Learn about it in the next section.

Now press the Next button. The wizard will show the schematic symbol (shape) suggested:

You can choose other shapes, if available, by pressing the long vertical button to the right of the symbol.

NOTE:
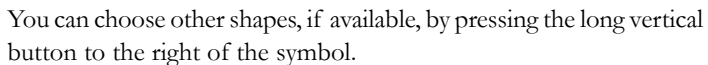
You may need to uncheck the "Show suggested shapes only" filter to see more shapes.

DesignSoft

In this mode, you can also search by name, by the number of pins, and by function (op amps, comparators etc.) You must select the Shape type option before searching by function.

In our example, TINA appears to have automatically selected an appropriate shape, so you may now press the Next button. The selected shape, pin connections, and Spice macro text will appear.
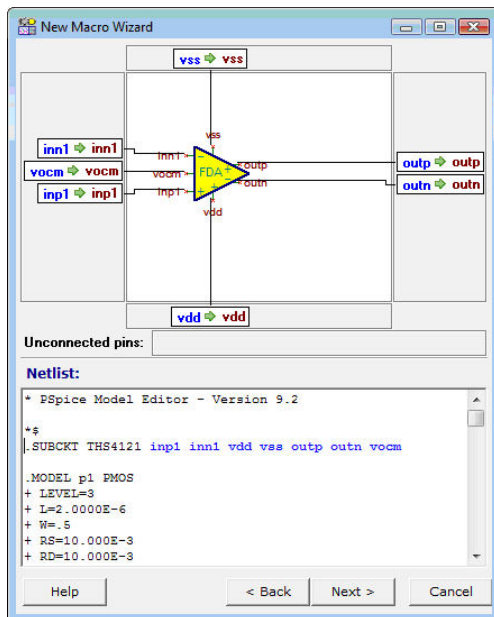
Take a moment to check the connections and if necessary correct them by dragging the connection labels.

If everything is correct, press the Next button again. The Save dialog will appear and you can save the macro into the User or TINA folder. Do so immediately, or do it later using the Insert menu.

Now let's insert a model from another manufacturer. Our choice is the LM111 comparator from National Semiconductors.

After invoking the internet browser in TINA find the Spice model page at http://www.national.com which is, at the time of writing this manual, http://www.national.com/analog/amplifiers/spice_models.

Scrolling down the page you will find the LM111 model under LM111.MOD name as shown below.

SPICE MACROS

DesignSoft
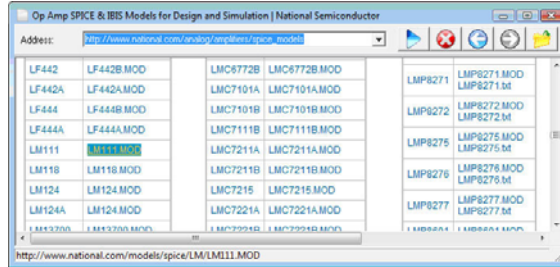
Click on the link and the text of the Spice model will appear in TINA.
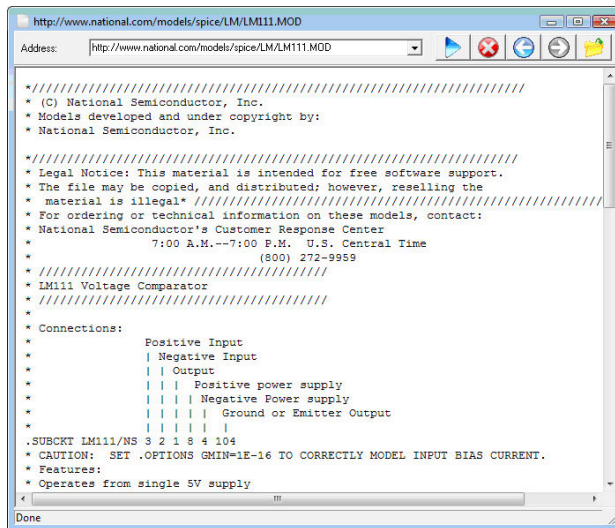
Now click on the Open icon at the top-right corner of the browser window, and the Macro Wizard will appear again confirming the successful load. Press Next. TINA will automatically present the selected symbol.

279
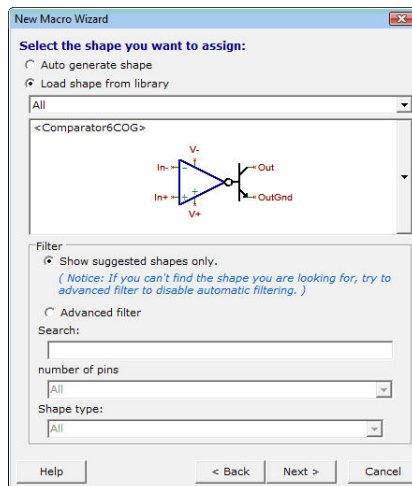
The remaining steps are the same as described in the previous section.

6.2.2   Adding Parameters to Spice Macros

TINA lets you add parameters to Spice subcircuits and set them from TINA. The parameters in the subcircuit are defined by the standard Spice syntax using the PARAMS keyword. For example look at the subcircuit in the circuit given in the MAC_RLC.TSC file in the EXAMPLES\SUBCIRC folder.
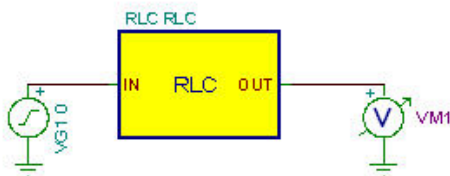
If you double-click on the RLC subcircuit and press the Enter Macro Button, the content of the subcircuit will appear:

```
.SUBCKT RLC In Out  PARAMS: R=100 L=1M C=1N
C1 Out 0  {C}
L1 1  Out {L}
R1 In  1  {R}
.ENDS
```

SPICE MACROS

The parameters are R, L and C. You can set the parameters in the property dialog of the subcircuit in TINA created as described in this chapter. In our example
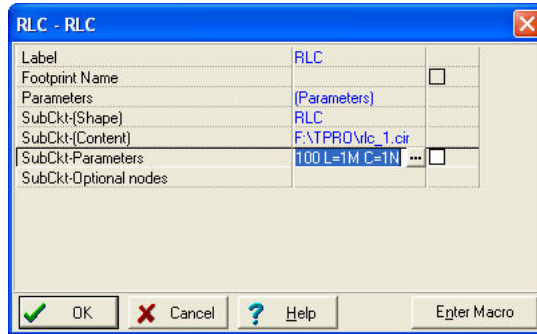
Edit the parameters either in the SubCkt-Parameters line or click the ▣ button and the Macro Parameter Editor dialog will appear. Enter or edit the parameters you want to change and press the OK
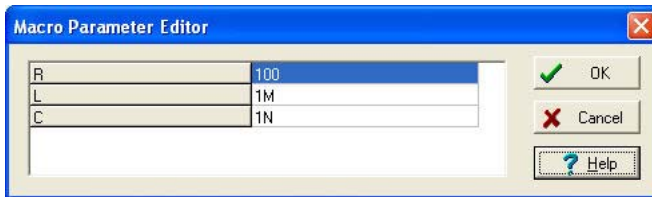
6.3 Using and extending Manufacturers' Spice model catalogs in TINA

In *TINA* you will find large catalogs of manufacturers' Spice models. In most cases, you will find that the required components are already in *TINA*'s manufacturers' model catalog. You can select components by function, manufacturer and part number.

You can also extend the library using *TINA*'s Library Manager program.

SPICE MACROS

DesignSoft

6.3.1   Using the Library Manager

TINA has large libraries containing Spice models provided by semiconductor manufacturers such as Analog Devices, Texas Instruments, National Semiconductor, and others. You can add more models to these libraries or create your own Spice Library using *TINA*'s Library Manager (LM).

Let's learn how to add a Spice model to *TINA*'s Spice libraries:

6.3.1.1   Introduction to Adding Spice macros to TINA Libraries

Start the Library Manager program. Use the Windows Start menu to locate the TINA folder and click on its icon. Select Collect subcircuits and models from the File menu. Find the EXAMPLES\SPICE folder (in the program folder where TINA is located) in the dialog box, click the SPICE folder where our example subcircuit -a ua741 amplifier model- has already been placed, and press Next.
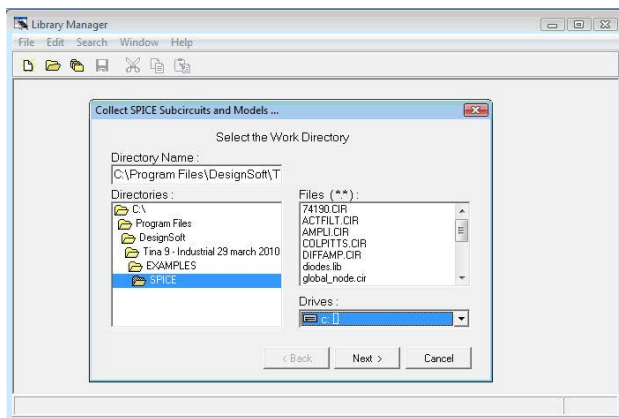
A new dialog box will appear with the list of available files on the left side. Note that the file you select must be a proper Spice subcircuit. Click UA741 and then press the > button. The UA741 model you have selected will appear on the list of selected files. In a similar way, you can select more files or even all the files by pressing the ⟩⟩ button.
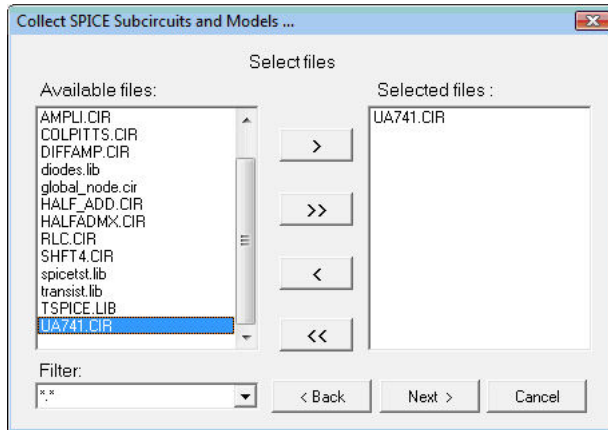
Press the Next button to continue. The following dialog box will appear.
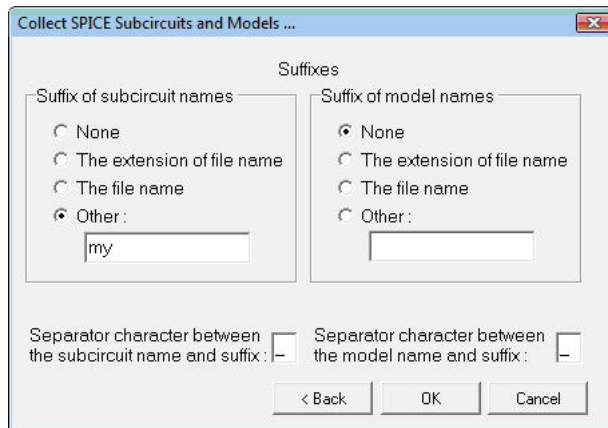
With this dialog box you can make changes in the subcircuit or model name. This might be necessary to avoid conflicts among different subcircuits or model versions with the same name. To differentiate the new model, you can add the file name as a suffix to the subcircuit name or add any text as a suffix to the name using the Other option. Let's add the suffix "my" to the subcircuit name and then press the Next button. The contents of the new library file will appear.

DesignSoft
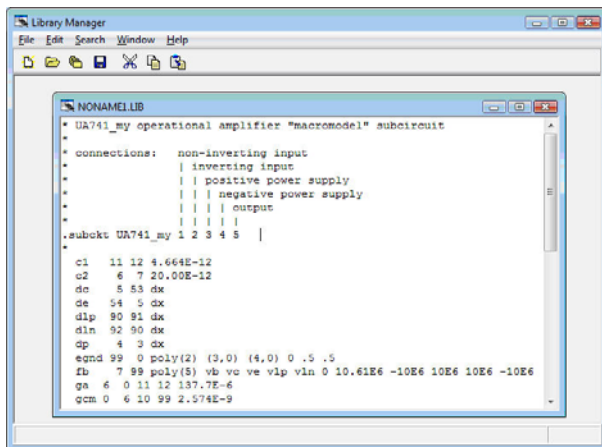
NOTE:

The new name of the subcircuit has the suffix "my": UA741_my.

Using File|Save As, save this library with the name my_741.lib in the Spicelib folder under the Windows Documents folder, for example: **Documents\Designsoft\TINA_Industrial_install date_id_number\Spicelib**

This folder is automatically created and set by TINA.

NOTE:

Under Vista and Windows 7 and later operating systems new libraries must be created in the user area under the Documents folder of Windows because the Program Files folder is normally write protected in these operating systems. This is also useful because the user data and the original libraries of TINA are separated this way.

SPICE MACROS

Now select *Create TINA Library Description* from the *File* menu and select select …for Spice models and subcircuits. The following dialog box will appear.
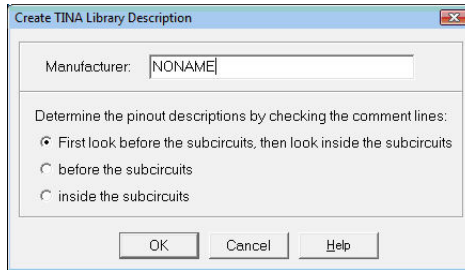
Here you give your new library a name– a name that will appear in the Manufacturer field of the Spice macro insertion tool. Let's change this to My Spice Library.
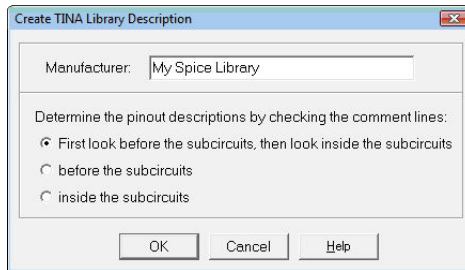
You can also specify some search options for determining the pinout description of the Spice model. The default setting is usually satisfactory. Press the Help button for more information.

The description (directory) of the new catalog will be displayed in a new window:

SPICE MACROS

DesignSoft
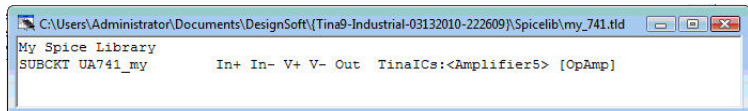
NOTE:

> *If everything went well, as it did above, you should not see any warnings (in red) such as "No processable statements" or "Autoshape" which would mean the Library Manager could not fully resolve the task fully automatically. If any of these warnings appear, read the next chapter for the resolution.*

Finally, save the library directory as my_741.tld in the same Spicelib folder described above. Note that the Save As command applies to the active (selected) windows only.
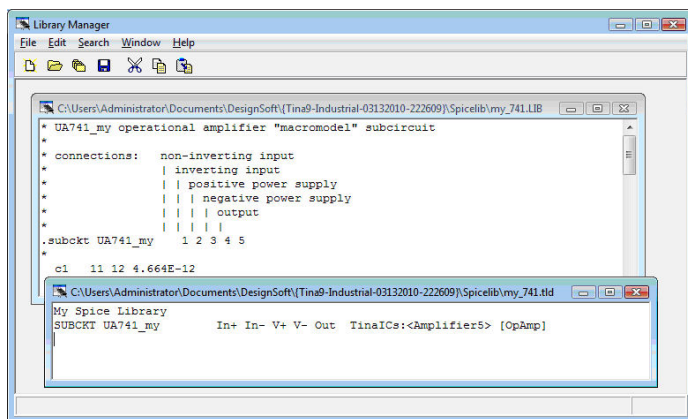
Next time you start *TINA*, this message will appear:
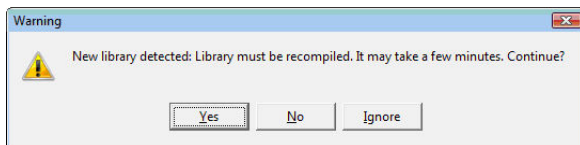
Press Yes to add your library to TINA's libraries.

NOTE:

> *If this message does not appear for any reason (for example due to file date format differences), re-compile the libraries manually using the command **Re-compile Library** from the Tools menu. You can also recompile the library already in the Library Manager. Select Create TINA Library and Compile Library in the Library Manager File menu. In this case the warning will not appear.*

Now click on the Spice Macros tab and on Operational Amplifiers on the component toolbar. Your new component library should appear here in the list of Manufacturers . To access your new subcircuit, select either "My Spice Library" or All. If you have selected All, simply press U to jump directly get to the U's where it will be easy to find the UA741MY on the list. If, on the other hand, you had selected My Spice Library, the list would, of course, contain only your new opamp.
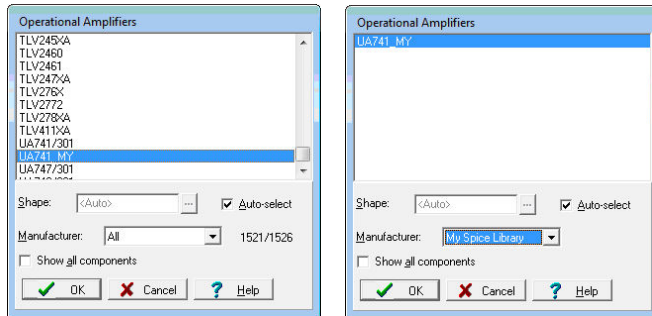
6.3.1.2 Problems and solutions while adding Spice macros to TINA

In many cases, adding models to TINA is as easy as described above, but in some cases it is impossible to find the connection between the Spice models and their graphic symbols automatically. Fortunately, TINA's latest Library Manager makes it easy to solve this problem.

Let's add   the   librar y   "SPICETST.LIB"   from   the EXAMPLES\SPICE folder to TINA.

NOTE:

You can navigate among TINA's folders using the small arrow next to the Open button.

DesignSoft

First, start the Library Manager as described above. Open the "SPICE TEST.LIB" file using the Open icon or the Open command from the File menu. Select the "Create TINA Library Description"command in the Spice Models and Subcircuits submenu and in the appearing Create TINA Library Description dialog window enter the name of your new library (spicetst) , the name that will appear in the Manufacturer field of the Spice macro insertion tool of TINA) as described above.

You will see the following window:
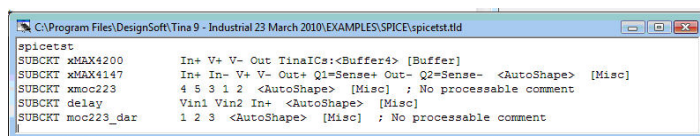
Looking at the lines of the "spice test.tld" file containing the library description, it seems that the first model, xMAX4200, was recognized automatically, since both the graphic symbol and the category were found.

For the second model, the xMAX4147, no graphic symbol was assigned and its category was not recognized; however, the program recognized its terminals.

For the third model, the xmoc223 Optocoupler, nothing was recognized. Note that even if the program does not recognize a model, it is placed in an automatically generated box (Autoshape) and still can be used.

NOTE:

There are also 4th and 5th models on the list. However, if you examine the Spice source in the spicetst.lib library, you can see that these are auxiliary subcircuits of the Optocoupler. We will address this issue later in this chapter.

DesignSoft

Now let's add appropriate graphic symbol to the models.

From the Edit menu, select the "TLD Editor for Subcircuits" command. The following dialog box will appear:
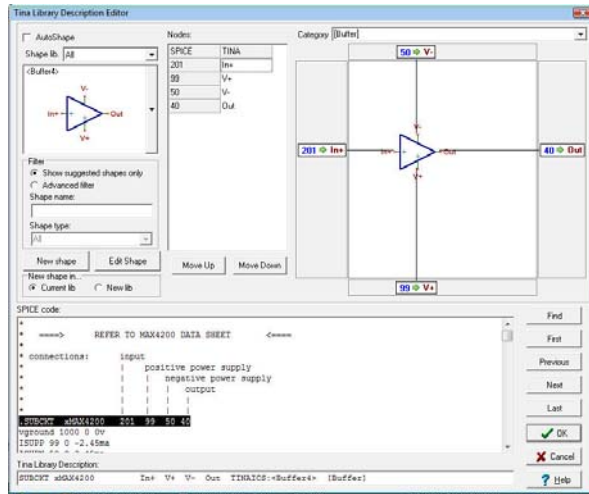

TINA presents the graphic symbol with the names of the terminals on the left. Next to the graphics symbol under Nodes: you can see the list of terminal nodes and the associated terminal names of the graphic symbol. You can move the graphic terminal names up or down by simply dragging them or by using the Move Up and Move Down buttons. You can also simply drag the terminal names to any terminals on the larger work area to right of the node list. If you drag a terminal name over another terminal and drop it by releasing the left mouse button, the two names will be interchanged. You can also shift the symbol and zoom in and out by holding down the left or right mouse button accordingly and moving the mouse or turning the mouse wheel. This is useful with larger and/or complicated symbols.

In the top right corner, TINA displays the category, which you can also change.

At the bottom of this dialog, the SPICE code of the selected component is shown, while below that, in the "TINA Library Description" field ,TINA displays the actual content of the line of the .TLD file. With the buttons First/Previous/Next/Last, you can move among the models in the library.

DesignSoft

Check that all entries for the first model are correct.

Now jump to the second model by pressing the Next button.
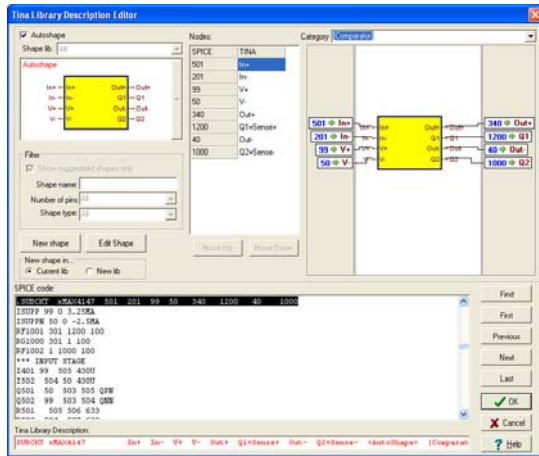
The first things you should notice are the square Autoshape graphic symbol and the red TLD line at the bottom. These indicate that the Library Manager could not fully understand the model.

Uncheck the Autoshape checkbox at the top-left corner of the dialog. The dialog will change as follows.

SPICE MACROS

DesignSoft
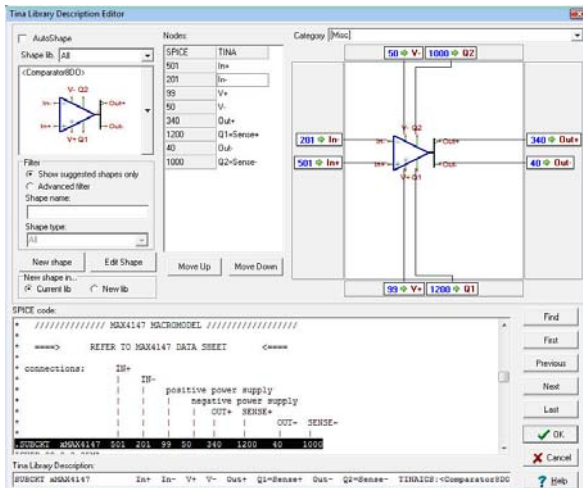
Strange, but it now appears that the Library Manager has found a correct symbol. What's happening here? What was the problem before? The reason was that there were several symbols available with the same number of pins.

Press the vertical button at the right side of the Shape window and review the list of available symbols that appears. You can choose another symbol by clicking on it. The Library Manager seems to have selected the best option, but in general it is not guaranteed.

Check the list showing the connection between the Shape nodes and the Spice terminals. It should be correct since the Library Manager did not give an error message for the Spice comments. If there had been a discrepancy, you would have seen an error message in the TLD line: "No processable comments".

Change the Category at the top right corner to <Comparator >.

Now click the Next button again to bring in the last model in this library. The following window will appear:
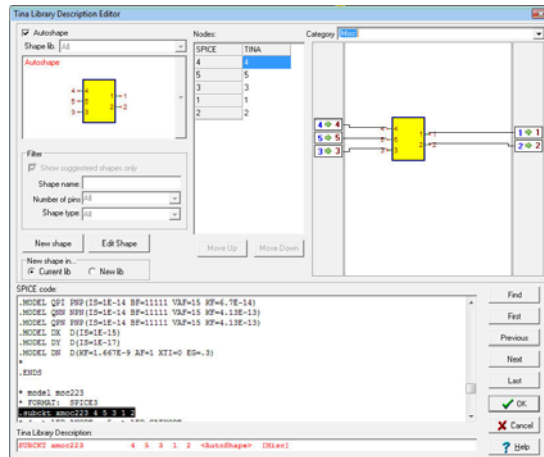
Note the red line at the bottom of the window. This means that the Library Manager could not identify the Spice terminals from the comments in the Spice model. If you look at the comments in the Spice code of the component, you can see they are not common in

the world of Spice macros (e.g. LED ANODE), this is why the Library Manager could not identify them. We will have to make the connection between the graphic shape and the Spice   terminals manually.

First uncheck the AutoShape checkbox. You will see the following window:
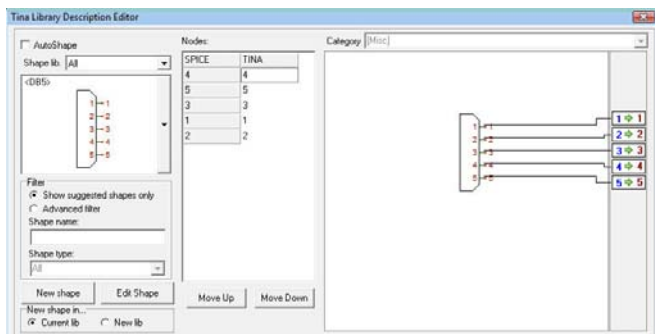
Obviously this is still not the right choice. To find the right symbol, uncheck the "Show suggested shapes only" filter and from the Shape type list select Optocouplers. Now the right symbol will appear:
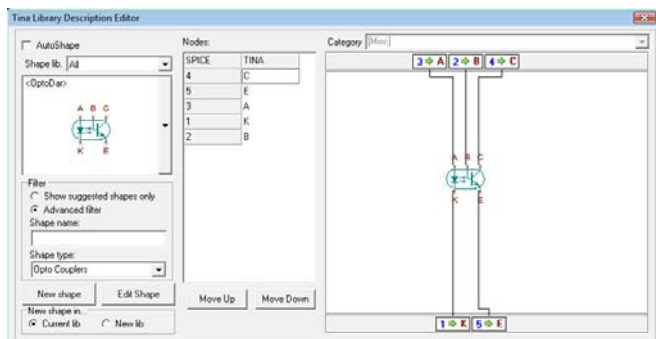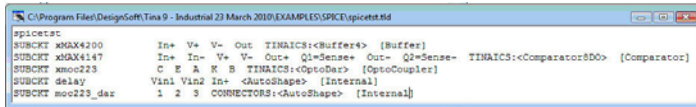

We're not done yet: the Spice terminal nodes and the Shape terminals are not properly mapped. For example the first node in the list, node 4, is associated with the Collector of the transistor, but according to the comments in the Spice code, it should be connected to A, the Anode of the input LED.

In the list under Nodes: click on A and drag A to the top of the list (to Node 4) , then drag K to Node 5, and finally drag E to node 3. Note that you have the option to make the same changes graphically by dragging the labels on the right side.

Check the remaining two nodes, which in this case must be correct. Change the Category at the top right corner to <Optocoupler >.

As mentioned earlier,, there are also 4th and 5th models on the list. However, if you examine the Spice source in the spicetst.lib library you will see that these are auxiliary subcircuits of the Optocoupler , so we do not need them separately in our schematics in TINA. To keep them from being listed in the TINA catalog, invoke them one after the other by pressing the Next button and set their Category to Internal.

This completes our editing of the various models. Press OK to close the TLD editor. The SPICE TST.TLD window should be updated and look like this:

Using the File|Save As command, save both the SPICE TST.TLD and the SPICE TST.LIB files in the Spicelib folder under the *Windows* Documents folder, for example: Documents\Designsoft\TINA_Industrial_install date_id_number\Spicelib

This folder is automatically created by TINA at installation. You can navigate to this folder by using small arrow next to the Save button and selecting Private catalog folder.

NOTE:

> *You can also use the Shared catalog folder which you can setup at installation of TINA. Using the latter you can allow all users to use the new libraries in case they have access normally through local network to the Shared catalog folder.*

DesignSoft

Finally, use the File|Create TINA Library command |Compile Library to register the changes for TINA. In this case when you restart TINA there will be no need to recompile the library.

Close the Library Manager.

When you restart TINA you can find these new models by looking in the appropriate category (Comparators, Buffers, and Optocouplers). These new parts will be at the end of the list, since the names of the new models start with X.
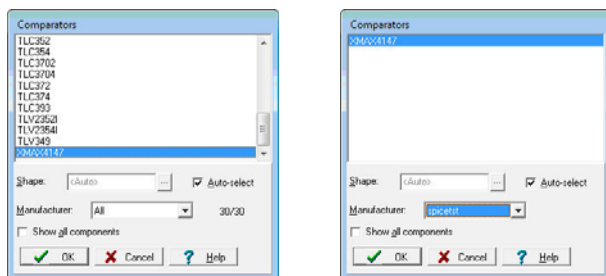

Of course, you can also set the Manufacturer in the appropriate category to "Spice test" to see only the newly added components.

6.3.1.3 Adding Spice models in .MODEL format to the library

In the previous example, you added a component described by a Spice subcircuit. You can also add diodes, transistors and other devices by simply using .MODEL instructions. These devices are normally placed in a file containing many .MODEL instructions. In TINA, there are two such sample libraries, called diodes.lib and transistors.lib

First, open the diodes.lib file from the EXAMPLES\SPICE folder using the File |Open File command or the corresponding icon on the toolbar. The following window listing the contents of the file appears:

DesignSoft
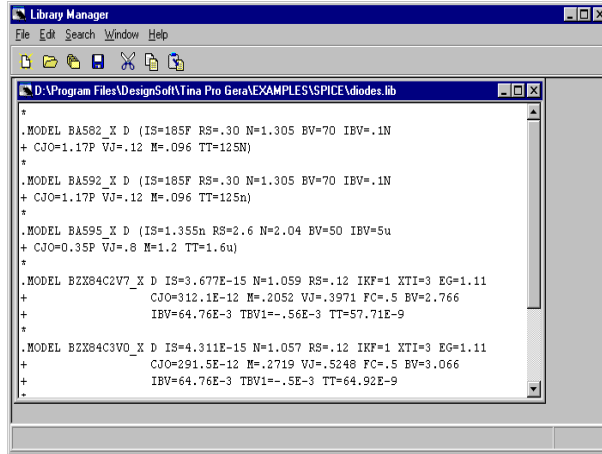
Execute the File/Create Tina Library Description/For SPICE models and subcircuits... command.
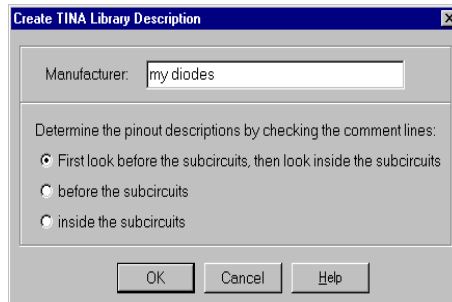
Do not change the other settings. Press OK. A description listing models in the new TINA library will appear:

DesignSoft
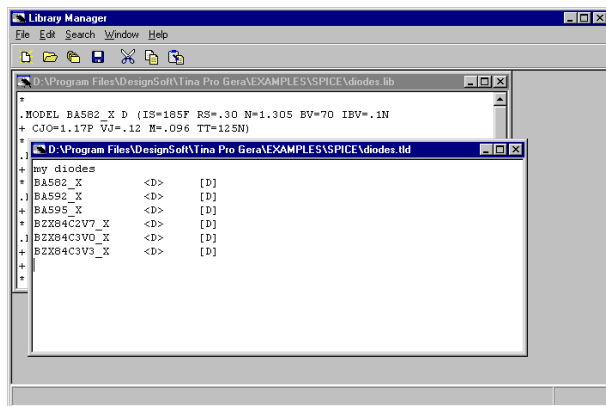
The file contains 3 normal and 3 Zener diodes. In the Spice language, there is no difference between normal, Zener, LED, Schottky, Varicap, and other diodes. However, in TINA, you can assign different schematic symbols to these types. To do this, select Categorize Components from the Edit menu. The following dialog box will appear:
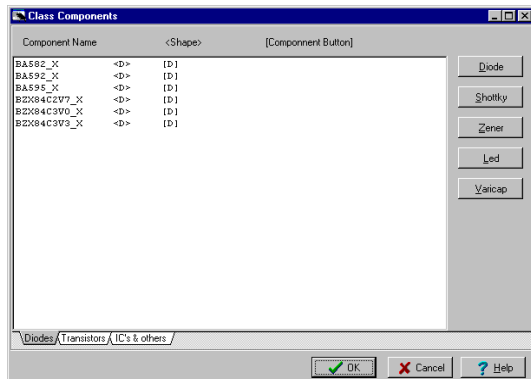
Select the Zener diodes (the last 3 items on the list) by clicking them one by one while holding the Ctrl key. Then press the Zener button.

<D> and [D] identifiers will change to <DZ> and [DZ] ensuring the use of the Zener diode symbols in TINA. Press OK and save both library files in the SPICELIB folder at

Documents\Designsoft\TINA_Industrial_install date_id_number\Spicelib

DesignSoft

To verify the new diodes, restart TINA, select diodes or Zener diodes from the toolbar, drop the diode onto the schematic, and double click on it. Press the ▣ button at the type line and select the "my diodes" library using the drop down menu of the Library field at the left top corner of the Catalog Editor dialog box.
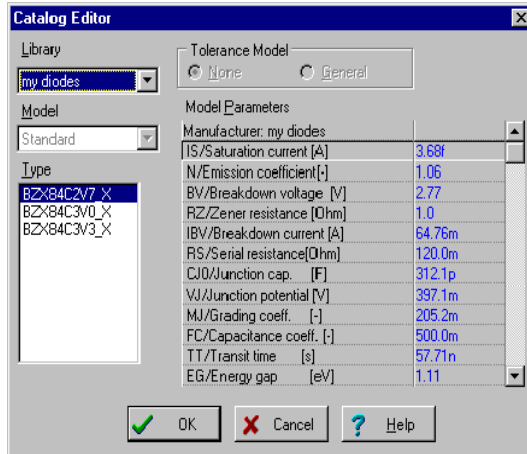
You will also find your new "normal" diodes under the Diodes category of the toolbar. Note that you can add new diodes to any existing manufacturer catalog if you select in the Library Manager you define a library name already in the drop down menu (of the Catalogy Editor of TINA.).

In a similar way, you can try adding transistors given by .MODEL Spice commands to TINA using the transist.lib library. There is no need to categorize these components since the NPN, PNP, NMOS, PMOS, etc. transistors have different notations in Spice.

6.4 Adding S-parameter models

Let's learn how to add an S-parameter model to TINA's libraries.

Start the Library Manager program. Use the Windows Start menu to locate the TINA folder and double click on the Library Manager icon.

Next, select Collect S parameter files... from the File menu.

Note: You should not use the File | Open… command to collect S parameter files.

Find the folder EXAMPLES\RF in the dialog box and double-click on the RF folder. This is where our example, an S-parameter transistor called s_bfp405.s2p, has already been placed. Press the Next button. Note: the S-parameter files must have either S1P or S2P extension (the manufacturers use the same convention). If the extension is S1P it means that the device is a "1-port" (described with 1 parameter) otherwise a "2-port" device (described with 4 S-parameters).

A new dialog box will appear with the list of available files on the left. Note that the file you select must be a proper S-parameter file. S-Parameter data files are in the TouchStone format. This is a typical data segment of a two-port file:

S-Parameter file description

\# MHz S RI R 50

0.30 0.02 -0.05 -0.03 -0.02 -0.03 -0.02 0.02 -0.05

0.31 0.03 -0.06 -0.02 -0.01 -0.02 -0.01 0.03 -0.06

0.33 0.04 -0.07 -0.01 -0.03 -0.01 -0.03 0.04 -0.07

....

The first line is a header that gives the frequency units, parameter, measurement format, and characteristic impedance of the measurement (here, 50 Ohms).

The first column is the frequency im Hz. The next columns are, in order, S11 Real, S11 Imaginary, S21 Real, S21 Imaginary, S12 Real, S12 Imaginary, S22 Real, S22 Imaginary. One-port data files are similar to the two-port files, except that there are no columns for the S21, S12 and S22 parameters.

Click on s_bfp405.s2p and then press the > button. The s_bfp405.s2p model you have selected will appear in the list of selected files. In a similar way, you can select more files or even all the files by pressing the >> button. In the next dialog box you can change the model name. This might be necessary to avoid conflicts among different modelversions with the same name. To differentiate the new model, you can create a model name from the file name or from one of the first 8 lines, or you can add a prefix or suffix to the model names. Let's just use the file name as a model name. Press the OK button and the contents of the new library file will appear.

Using File|Save As, save this library in the SPICELIB folder in your Private Catalog using the name myslib.lib. Now select Create TINA Library Description...|...for S parameter models from the File menu.

In the following dialog you specify a name for your new library, e.g., My S Parameter Library. You could specify the name of the manufacturer as a library name, but note that if there already is a library in TINA with the same name (e.g., Siemens), then your new model will be added to this library. The library descriptor file of the new catalog will be displayed in a new window. However, in the case of S parameter files, you must always categorize the models. To do this, select the Categorize Components from the Edit menu.

Press the IC's & other unrecognized components tab. Select one or more models from the list, then press a Move to page ... button (pick the button for the model type of the selected model). In our case, press Move to page Transistors, then click on the Transistors tab.

Select your new library by clicking the line at the top of the dialog.

Now select the appropriate category, which for this model is NPN.

Save the library descriptor file as myslib.tld in TINA's SPICELIB folder. (Both Spice and S parameter libraries are stored in this folder.) Note that the Save As command applies to the active (selected) windows only.

Finally, use the File|Create TINA Library command to register the changes for TINA.

Next time you start TINA, select RF components, and then NPN RF Bipolar Transistors and you will find the new component library in the list of Manufacturers. Your S parameter model will appear on the list invoked either by selecting "My S Parameter Library" or All.

6.5   Making a HDL macro from a file

You can create a HDL macro from any .vhd / .v / .va / .vams file that contains an entity (interface to the outside world) with its architecture (description of the hardware). Files with .vhd extension are VHDL files, with .v extension are Verilog files, with .va extension are Verilog-A files and with .vams extension are Verilog-AMS files. The ports declared in the interface part will automatically appear in the macro symbol (shape). By default, the input ports of the interface will appear on the left side of the generated macro shape and the output ports of the interface will appear on the right side, but by editing the generated macro you can change this arrangement.

For example (VHDL):

```
ENTITY e_Half_add_entity IS PORT(
  A : IN std_logic;
  S : OUT std_logic;
  C : OUT std_logic;
  B : IN std_logic );
END e_Half_add_entity;
```

In this case the A,B ports will appear on the left side and the S,C ports will appear on the right side of the macro shape.

Lets see how to do a macro from the following VHDL code (a half adder):

```
LIBRARY ieee, tina;
use ieee.std_logic_1164.all;
use std.textio.all;
USE tina.primitives.all;
```

```
— entity section
─────────────────
ENTITY e_Half_add_entity IS PORT(
  A : IN std_logic;
  S : OUT std_logic;
  C : OUT std_logic;
  B : IN std_logic );
END e_Half_add_entity;
─────────────────

— architecture section
─────────────────
ARCHITECTURE a_Half_add_arch of e_Half_add_entity
constant delay : time := 20 ns;
BEGIN
 S <= (A xor B) after delay;
 C <= (A and B) after delay;
END a_Half_add_arch;
```

1. Select Tools/New Macro Wizard...
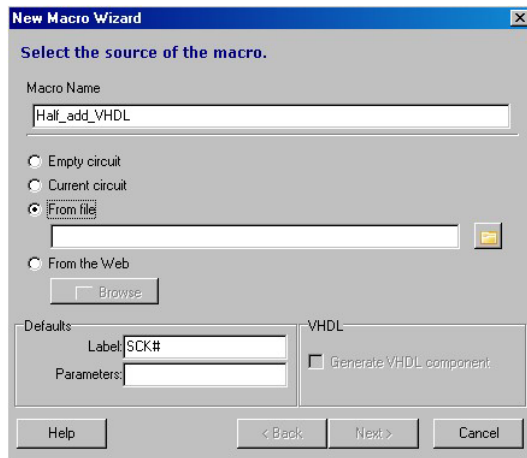
2. Type a name for the new macro.

3. Select the From file option then press the Open button and change the file type to VHDL, and navigate to EXAMPLES\VHDL in the TINA program folder. You should see the half_adder_VHDL.vhd file in the open dialog. Select this file and press Open.

DesignSoft

4. Press the Next button to save the macro, and save the macro into the default Macrolib folder.

6.5.1   Placing a HDL macro in the schematic editor

Now let's see how can we insert our previously saved VHDL macro into TINA's schematic editor.
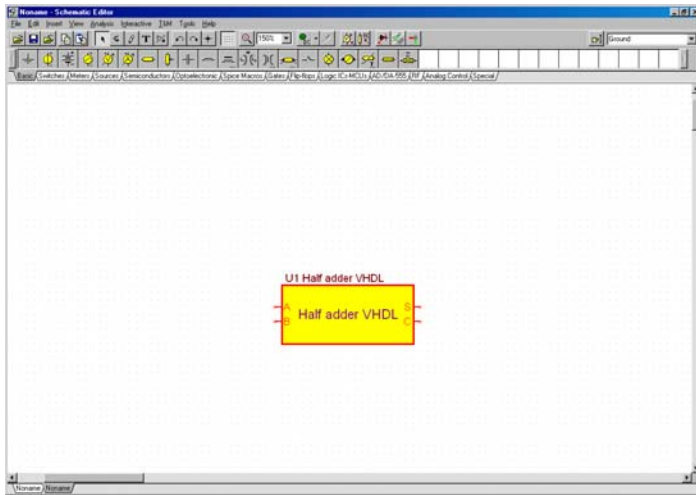
1. Select "Insert/Macro..." from the menu and select the previously saved macro half_add_VHDL.TSM from the MACROLIB folder of TINA's main program folder.

The screen will look like this:

To see the content of the macro double-click on it and press the Enter

Macro button on the property dialog that appears. The content of the macro will be displayed.

NOTE:

> *The command section starting with TINA VHDL Macro Description is generated automatically and should not be changed.*

6.5.2 Testing a HDL macro

Let's test our newly created macro in TINA's Digital interactive mode. To do this, place two High-Low digital switches from the Switches toolbar, one for each of the A,B inputs, and two logic indicators from

the Meters toolbar. Now select the DIG interactive mode with the ⏷

button or from the Interactive menu and press the 🟢DIG button. The

logic levels of the nodes will appear, Red for High. Blue for Low.

The logic indicators will also show the logic level of the outputs in a

Red square 🔴 for High, and empty square 🔷 for Low.

6.5.3   Changing the pin arrangement of a VHDL macro

To change the pin arrangement, you should add a special header to your VHDL macro.

The easiest way to do this is to open the automatically generated macro and edit its header.

For example the header in the previous example is

```
————————————————
— TINA VHDL Macro Description Begin
— entity_name:e_Half_add_entity;
— arch_name:a_Half_add_arch;
— ports:A,B;S,C;
— TINA VHDL Macro Description End
————————————————
```

The pin arrangement is determined by the:

```
ports:A,B;S,C;
```

line, the ports before the first semicolon (;) are placed on the left while the rest are placed on the right side of the macro box.

For example, if you change the ports line to

```
ports:A,B,S;C;
```

and add the whole changed header to the original VHDL file (which had no header) we get the following file (you can also load it from the EXAMPLES\VHDL\half_adder_VHDL.vhd.)

```
————————————————
— TINA VHDL Macro Description Begin
— entity_name:e_Half_add_entity;
— arch_name:a_Half_add_arch;
— ports:A,B,S;C;
— TINA VHDL Macro Description End
```

DesignSoft

```
─────────────────────

LIBRARY ieee, tina;
use ieee.std_logic_1164.all;
use std.textio.all;
USE tina.primitives.all;
─────────────────────

─ entity section
─────────────────────

ENTITY e_Half_add_entity IS PORT
  A : IN std_logic;
  S : OUT std_logic;
  C : OUT std_logic;
  B : IN std_logic );
END e_Half_add_entity;
─────────────────────

─ architecture section
```
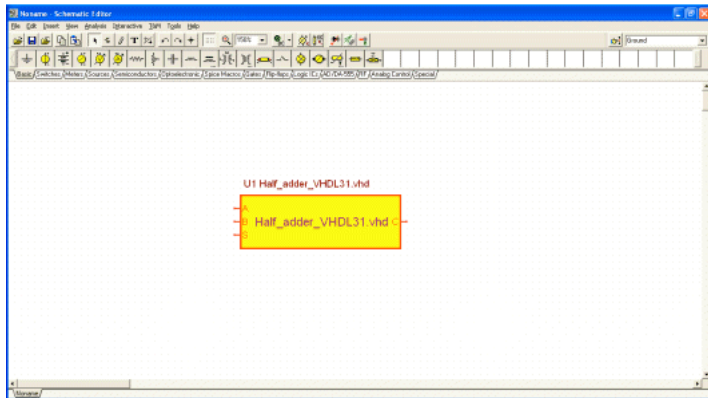


```
─────────────────────
ARCHITECTURE a_Half_add_arch of e_Half_add_entity
BEGIN
S <= (A xor B);
C<= (A and B);
END a_Half_add_arch;
```

Converting this into a new macro called Half_adder_VHDL.TSM
and then inserting it again we will see the revised pinout version.

MAKING YOUR OWN
SCHEMATIC SYMBOLS
AND FOOTPRINTS

7.1   Schematic Symbol Editor

Using *TINA*'s Schematic Symbol Editor, you can create new schematic symbols so that you can add your own circuit components to *TINA*.

To create new symbols, you place lines, arcs, rectangles, and arbitrary characters with any fonts, specifying line-width, color, and area color fills. After drawing the symbol, you add and define connections to it.

To start the Schematic Symbol Editor, use the Start or Apps screens of Windows 8 or the TINA group of the Start menu of Windows 7, Vista or XP.

To get acquainted with some features of the editor, read in the list of existing symbols. (Alternatively you may jump to paragraph "**Now let's create a new symbol"** below.)

Select *File|Open,* the little down arrow ▾ button on the right side of the Open button and select the Main TINA folder from the list. The folders and files in the main TINA folder will appear. Locate and then double click the *devices.ddb* file. On the right hand side of the Editor Window, the list of current schematic symbols will appear.
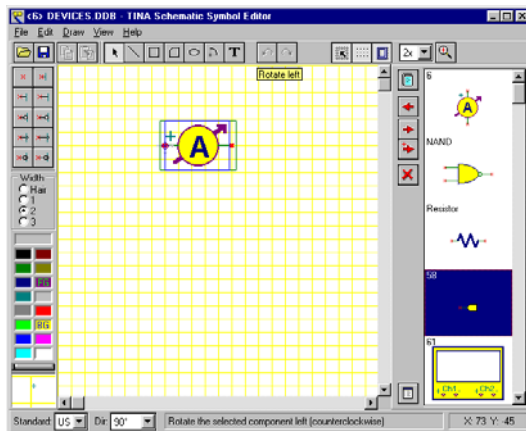
The first symbol on the list (an Ammeter) will appear in the editor window. Try the **Dir:** control at the bottom of the screen. Using this control, you can provide different shapes for symbols at each rotational orientation by designing each of them individually. Now click on the NAND symbol at the right side of the screen and press the ☐ button. The NAND gate symbol will appear in the editor window. Try the **Standard:** control to see the US and European versions of the part shape. You can design symbol versions for each standard, if necessary. If the symbols are identical in the two standards, you need create only one version.

Now let's create a new symbol for the full-adder circuit that was used above in our example of creating a half adder macro.
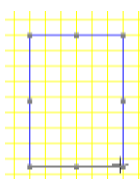
If this is your first own symbol press the File|New command to create a new library. If you already have your own symbol library open it with the Open command. First clear the editor window by pressing the ☐ New Device button on the left side of the Search field above the list of symbols on the right side of the screen. Now draw a rectangle as the body of the component. Press the ☐ button then click on any point in the drawing area, hold the mouse button, and move the mouse until the rectangle is properly sized.
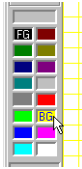
Fill the rectangle with a color by right-clicking on the palette at the lower left corner of the window. Note that a left-click will change the foreground color (FG), in our case the border of the rectangle.

Now add the terminals. Select the desired terminal type from the Terminal-Toolbar in the upper left corner of the window and move the cursor into the rectangle you just drew. Position it using the mouse or pressing the *[+]* or *[-]* key for rotation and click to locate the terminal. Be sure the small red x, indicating the pin end, is outside the body. Continue this process until every terminal is positioned.

Now add the terminals. Select the desired terminal type from the Terminal-Toolbar in the upper left corner of the window and move the cursor into the rectangle you just drew. Position it using the mouse or pressing the *[+]* or *[-]* key for rotation and click to locate the terminal. Be sure the small red x, indicating the pin end, is outside the body. Continue this process until every terminal is positioned.
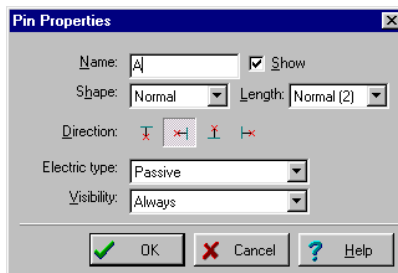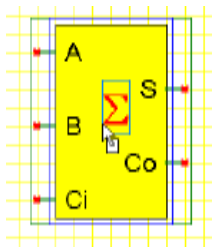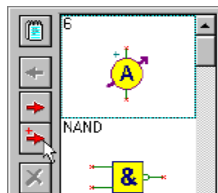
After you have positioned all the terminals, you can establish their properties by double clicking on each of them.

You should assign terminal names as shown in the picture below.

Next, add a large summation sign.

Click on the ☐ -Text-Editor button on the Toolbar, enter an S in the window, and select a font. To get the special Greek summation sign, select the Symbol-Font.

Press the ☐ Device Properties button, set the Name of the symbol to Full Adder, and press OK.

Finally, copy the new symbol into the symbol library with the ➡ button (it appears now at the end of the list), and use the File|Save or File|Save As… command to save the new or extended .ddb Library file.

Note, that if you create a new schematic symbol which must be placed in a new or existing library, it is strongly recommended to save it in a custom shape library (.ddb file) in your private catalog folder. To locate your **Private catalog folder** click on the little down

arrow ▾ button next to the Save button in the Save As... dialog.

Under Windows Vista, 7 and 8 or on a network you cannot save anything in the Main TINA folder, unless you have Administrator rights.

Another possibility to save your symbol(s) in the **Shared catalog**

folder, which you can select from the same list, using the ▾ button

where the Main TINA folder and Private catalog folders are listed. The Shared catalog folder can be specified at Installation and located in an area which all TINA users on a computer or network can access.

If you have a new schematic symbol, it can be assigned to a new component (macro) using the New Macro Wizard, as it is described

DesignSoft

in chapters 5.1, 5.2 and 5.5. - or using the Library Manager as it is described in chapter 5.3.

7.2   IC Wizard in the Schematic Symbol Editor

When you need to create the shape of an IC with a lot of pins, the IC Wizard can assist you. The IC Wizard can be activated from the Draw menu by selecting the IC Wizard command. The following dialog box appears:
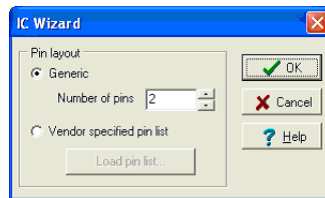
The wizard offers two options.

•   **Generic** If you select this option, the Wizard creates a rectangular-shaped IC with a DIP-style pin layout. The total number of pins must be specified. For example, if you enter 14 pins in this field, you get the following pin layout:
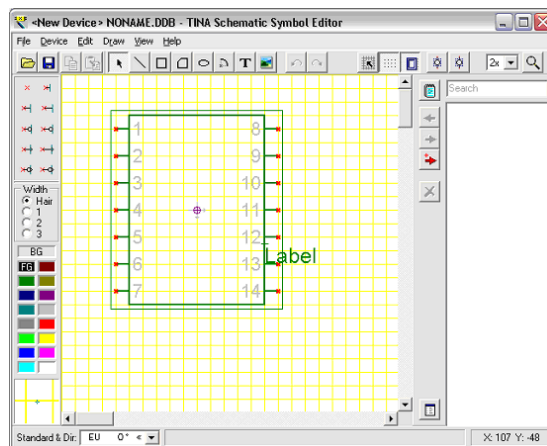
- **Vendor specified pin list** In this case the Wizard creates a shape based on a file where each line defines a terminal as **Pin number, Name, Electric type** separated by commas:

For example:.

1,RA2,INPUT

2,RA3,INPUT

3,RA4/T0CKI,INPUT

4,MCLR,INPUT

5,VSS,POWER

etc.

The electrical type can be INPUT, OUTPUT, INOUT, BUFFER and POWER. For example, if you read in the PIC16F84A.CSV file from TINA's EXAMPLES\PCB folder, the Wizard generates the next IC:
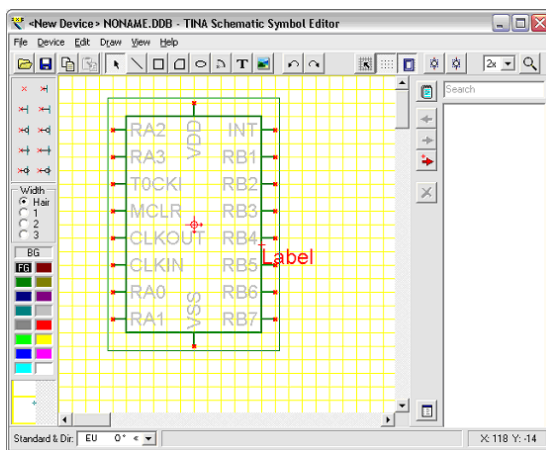

When the Wizard is finished, the shape can be further edited with the tools described above.

7.3 Footprint Editor

Using the Footprint Editor, you can create new footprint symbols that you can add to the footprint library. You can start the footprint editor from the Tools menu of TINA's PCB Designer by selecting the Footprint Editor command.

If you want to create a new footprint, you can build it by placing various primitive drawing elements and symbols, including lines, rectangles, arcs, text and pads. We'll recreate a simple resistor footprint already included in the system.

First clear the editor window by selecting the New Footprint command from the Footprint menu. Then set the position of the origin by double clicking on the cross symbol with the little arrows. Enter 1300, 1000 in the X and Y fields respectively. Check the Use Relative Coordinates checkbox, and press OK.
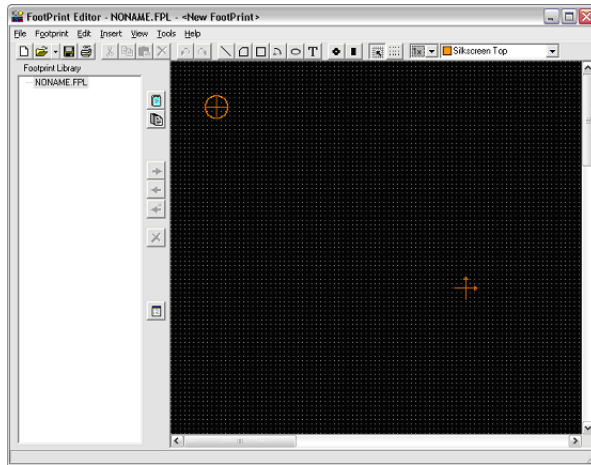
Now select the rectangle symbol on the toolbar and draw a rectangle around the origin. To do this, click on one corner, hold down the mouse left button, and drag the cursor to the opposite corner. Release the mouse button. If you create a footprint, you should be very careful with the dimensions. You must define the exact dimensions according to the manufacturer's data sheet, especially of the pads: otherwise the parts cannot fit on the board. To set the shape precisely, it is better to use coordinates rather than drawing with the mouse.

To set the size of our rectangle using coordinates, move the mouse over one of its edges and when the cursor changes into a hand symbol, double click at one edge of the rectangle. The Rectangle Property dialog will appear.
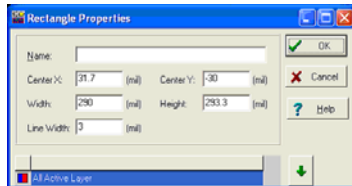
Now enter 0, 0 in the CenterX and CenterY fields; 840, 300 into Width and Height; and 5 into the Line Width fields.

In the Rectangle Property dialog of the shape you can change the layer settings, too. By default, a rectangle shape resides on the Silkscreen Top and Assembly Drawing Top layers.

Pressing the down arrow invokes the layer configuration editor. The layers can be turned on/off by double-clicking on the grey square next to the layer name. In our example, the default layer configuration is good, so don't change it. Close the property editor by pressing OK.
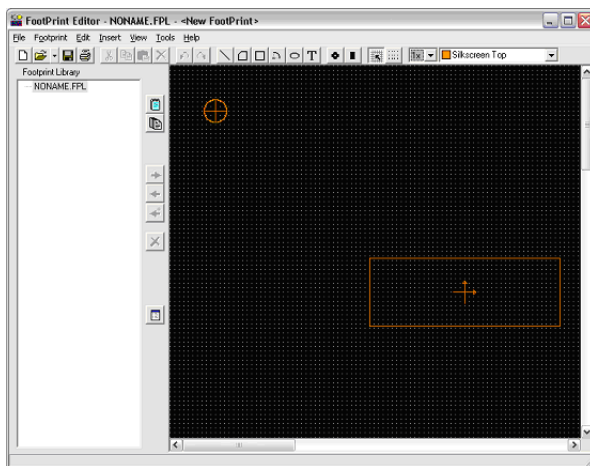

Now add 2 lines to our footprint. Select the line symbol and draw 2 horizontal lines next to the rectangle on both sides. Double click on the lines and modify the parameters as follows:

*Line1:* -460, 0, -420, 0, 5 (Point1 X, Point1 Y, Point2 X, Point2 Y and Line width)

Line2: 420, 0, 460, 0, 5 (Point1 X, Point1 Y, Point2 X, Point2 Y and Line width)

YOUR OWN SCHEMATIC SYMBOLS

Finally, add two through hole pads to the footprint symbol. Select the pad symbol from the toolbar. Move the pad next to Line1. Now activate the property editor of the pad, by moving the mouse over it and double-clicking when the mouse changes into a hand symbol. Enter - 500, 0 in the Center X and Center Y fields. The Drill parameter is 37. Now click on the down arrow. By default the pad resides on the Top, Bottom, Power, Ground, Solder Mask Top, Solder Mask Bottom, Drill Drawing and Drill Tape layers.

The default layer configuration could have been changed in a way similar to what we've seen in the rectangle example. Though the default layer configuration is good, we have to change the dimensions of the pad. Double click on the size field and enter 58 in the Diameter field on the Top, Bottom, Solder Mask Top and Solder Mask Bottom layers, enter 78 on the Power and Ground layers and 37 on the Drill Drawing and Drill Tape layers. It's important to enter the package pin number into the name field.
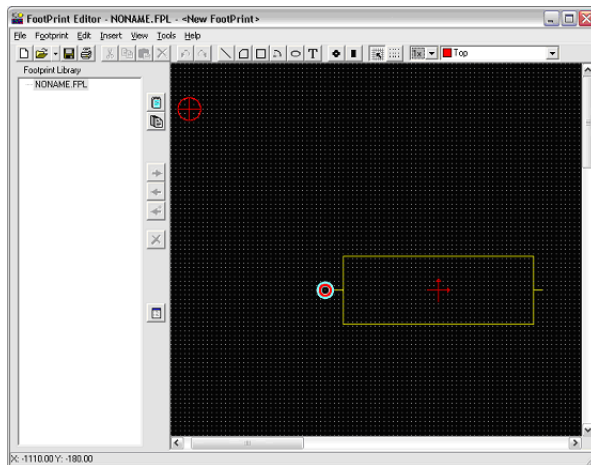
Now pick up the next pad and move to Line2. We have only one parameter to change, Center X, which should be 500.

DesignSoft
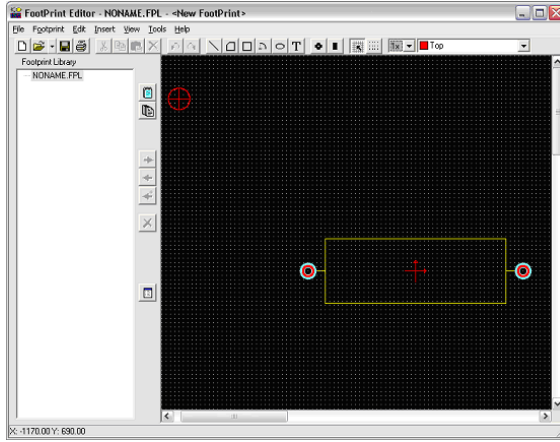
The footprint symbol is ready to save into a library.   Open the package.fpl file, select the resistor group (or define a new group) and press the Add footprint button.
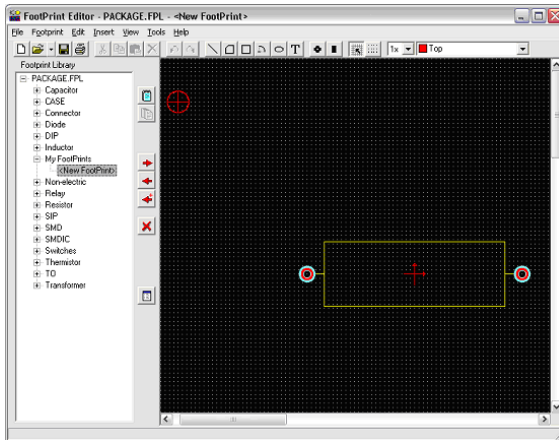
7.4   IC Wizard in the Footprint Editor

If you want to create the footprint of a more complex IC, e.g., an IC with a complex pin configuration, the IC Wizard can assist you. The IC Wizard can be activated from the Insert menu.

The wizard presents several properties of the IC which you can set.

YOUR OWN SCHEMATIC SYMBOLS

DesignSoft

In the **Technology group,** you can set the mounting mode and the package type of the IC. The mounting mode can be through hole or surface mounted. Depending on the mounting mode the following packages are available: DIP (Dual in line package), PGA (Pin grid array package), SPGA (Staggered pin grid array package), SOP (Small outline package), LCC (Leaded chip carrier package), QFP (Quad flat package), BGA (Ball grid array package), SBGA (Staggered ball grid array package), SIP (Single in line package) and ZIP (Zigzag in line package) respectively.

In the **Package dimension** group, the dimensions (length, width, 3D height) of the package can be set. Depending on the selected package, the 4th parameter is either notch, corner cutoff, or ignored.

The **Pad dimension** defines the shape and dimensions (length, width) of the pad. If the mounting mode is through hole, the shape of the drilled pad can be round, square or octagon. Moreover, the shape and dimensions of the drill diameter can be defined. However, if the mounting mode is surface mounted, the shape of the pad can be circular, rectangular or rounded corner and the appropriate dimensions can be also set.

In the **Pad position,** the number of pins and the distances between them can be set according to the package type.

Finally, in the **Pad numbering** group, the type and direction for pad numbering can be entered, depending on the package type.

An example:

Technology: Through hole

Package type: DIP

Package dimension/Length: 400

Package dimension/Width:270

Pad dimension/Shape: Round

Pad dimension/Drill hole: 20

Pad dimension/Diameter 40

Pad position/Number of horz. pins: 14

Pad position/Between pins: 50

Pad position/Between rows: 160

YOUR OWN SCHEMATIC SYMBOLS

DesignSoft
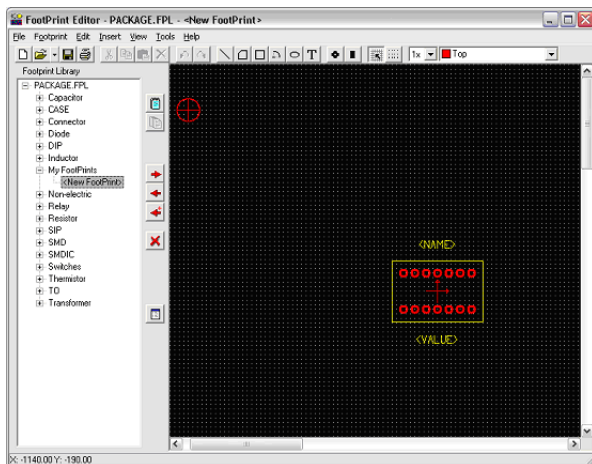
When the wizard is finished, the footprint can be further edited and saved in the library.

7.5   Adding Public PCB Footprints to TINA

TINA PCB can extend its footprint model library with footprints published by manufacturers or available from other sources. It can be achieved in a 2 step conversion process.

Footprint libraries are usually published in the format BXL which is used in a freely available software called Ultra Librarian (www.accelerated-designs.com/ultra-librarian/) In this software you have to open or download the component you wish to import into TINA. Select the export format "Ki-CAD" and press the "Export to selected tools" button. The Ultra Librarian software now creates new files in its own folder at the path: Library/Exported/KiCad/[name containing the date and time of processing] In this newly created folder you have to go to the folder "footprints.pretty". If you see one or more files with the extension "kicad_mod" the first step of the footprint conversion was successful.

DesignSoft

Now open Tina PCB's Footprint Editor. You can open a footprint package file or create a new one. Then create a footprint group with the appropriate button, then press "New footprint". Press the "Import footprint" button at the middle of the window and select the previously created kicad_mod file. On successful import the footprint is visible in the footprint editor. At this point you should check if the component's Origin is at the right place - check the menu View/ Supplementary and drag the Origin symbol which is displayed as 2 crossing arrows. You can also double-click on it and enter numerical coordinates. The Origin should be at the center of the component usually. Now you just have to save the component to the footprint library.

Some footprint models may also contain a reference to a 3D model file. If the 3D file is present, TINA PCB tries to import that automatically as well. See the next section on how 3D import works.

7.6   Adding Public 3D Footprint models to TINA

In addition to VRML format, in TINA Design Suite v11 and later versions the PCB component library can be extended with 3D models in industry standard STEP, STP, STL and the popular Google Sketchup (SKP) format.

In the Footprint Editor under the Footprint menu select Footprint Properties. In the Footprint Properties dialog select the "Import/Update 3D model" then press the Execute Command button (or double-click on the radio button). The Import 3D Model dialog appears where you can open models in STEP, STP, STL, SKP formats. The orientation of these model files can vary depending on the publisher, so you can rotate the model to the desired fit to the board with the top toolbar's buttons. You can also set the vertical position and the model size, using different measurement units. After the successful import procedure and saving to the library, the model is assigned to the footprint and it can be displayed in the PCB Viewer 3D as a component of the circuit board.
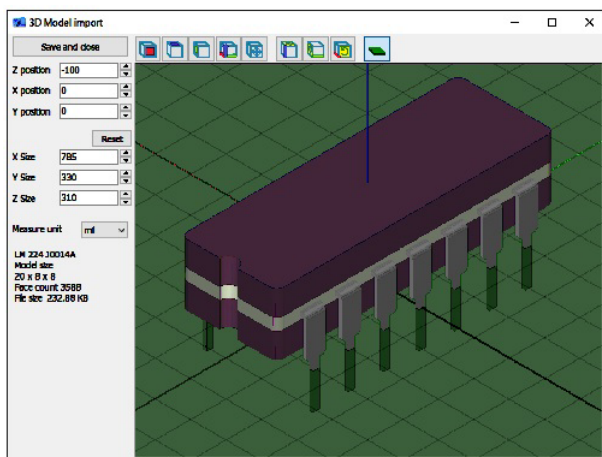
You can adjust the orientation, placement and size of the footprint model any time in the "Footprint properties" window with the "View/Adjust 3D model" function, so the 2D and 3D views will exactly match.

DesignSoft
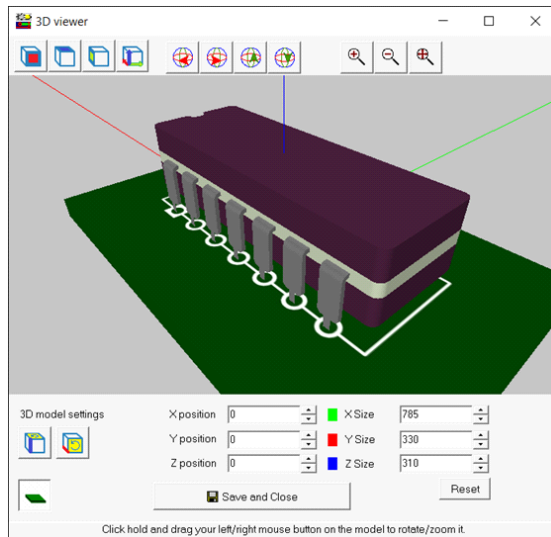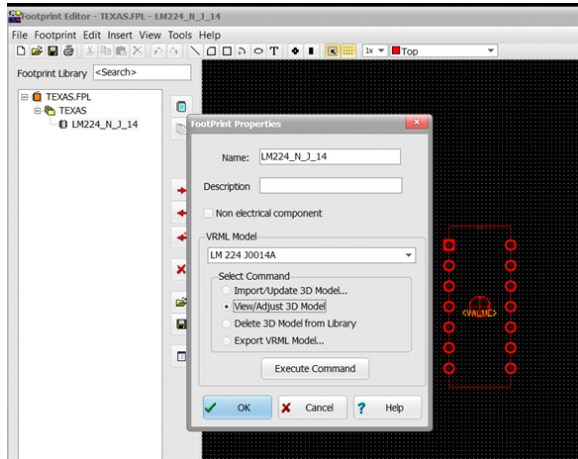
If you don't see a 2D footprint displayed on the board here, you should first save the footprint to the library.

USING THE PARAMETER EXTRACTOR

Using *TINA*'s Parameter Extractor you can create component models that more closely represent actual real world devices by converting measurement or catalog data into model parameters.


Use the Windows Start menu to locate the *TINA* folder. Start the Parameter Extractor by double-clicking its icon. To create a new transistor, which can be added to *TINA*'s transistor catalog later, select *File|New Library|NPN Transistor*.
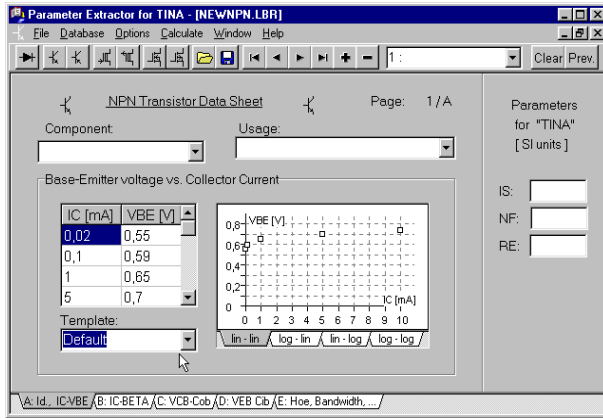
The dialog allows you to enter data from measurements, from manufacturers' catalog data, or from *TINA*'s default values (use the Template-ComboBox for this).

Click on each tab at the bottom of the screen and fill in all the transistor parameters. Select the default values or enter your own.
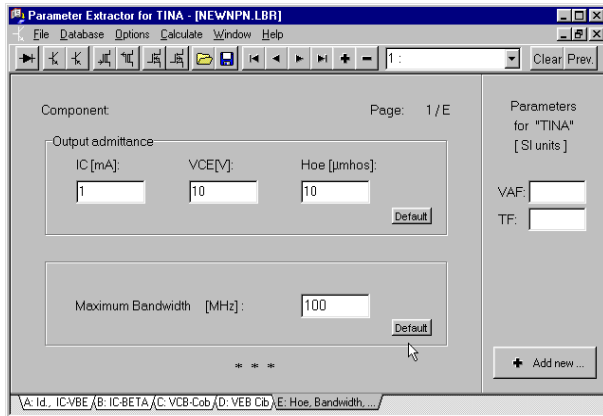

NOTE:

Be sure to fill in all data, since missing data may lead to incorrect results.

Next select *Calculate|Current component*. To check how well *TINA*'s transistor model matches the input data, you can walk through the tabs to see the calculated graphs and numeric values for every parameter.

Finally, let's insert the new transistor into the source file for *TINA*'s transistor catalog by selecting *File|Catalog Manager*. To be able to use the new catalog, you must recompile the modified source files and link them together into the CLCC.CAT catalog file.

Locate and open a component-file compatible with your component (e.g., if adding a bipolar transistor, choose a bipolar catalog, bipol_x.crc). Click on the Browse-Button and select the file from the File Open Dialog. All component-files delivered with *TINA* are placed in the CLCC subdirectory of the *TINA* directory (by default *C:\Program Files\Designsoft\TINA*).
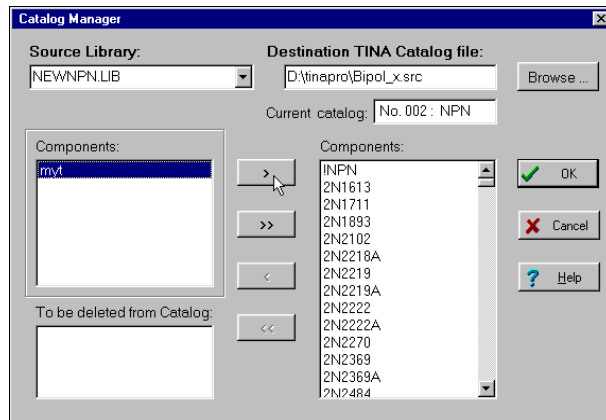
Move your component into the library by selecting it, clicking on the ⟩ button and then the OK button.

After Pressing OK, TINA will prompt you and ask if you want to recompile the catalog source files and create a new updated catalog. If you answer "Yes," TINA will create the new catalog and you can use it after restarting TINA. You can also recompile the catalog using the "Compile TINA Catalog" command in the File menu. This may be necessary if a previous attempt at compiling failed, e.g., due to insufficient hard disk space.

In a similar fashion you can calculate magnetic core parameters. You should enter the upper (A) and lower (B) curve of the hysteresis and the geometric parameters of the core. Run an example with the default parameters (load Default from the Template listbox) to see typical values.

ADVANCED TOPICS

9.1   Introduction

TINA offers many useful and advanced features for designing, testing, and teaching electronics. These include S-parameter models, network analysis, Fourier series and Fourier spectrum analysis, noise analysis, symbolic analysis, post-processing of analysis results, phasor diagrams, Nyquist diagrams, a built-in interpreter, multi-parameter optimization, and much more.

Detailed descriptions of all these features can be found in this chapter.

9.2   Parameter Stepping

In parameter stepping mode, TINA steps the associated component values a given number of times through a given range and runs the selected analysis for each value. This results in the calculation and plotting of a set of curves illustrating the sensitivity of the circuit to changes in the associated parameters.

Parameter Stepping is controlled by four parameters: Start value, End value, Number of cases, and Sweep type (Linear, Logarithmic or List).

The stepped component value parameters can be any numerical parameters of the TINA components–e.g., the resistance of resistors, the setting of a potentiometer, or parameters on transistor data sheets.

To set up parameter stepping for a component, click on the Analysis.Control Object menu item, or click on the 🖳 Select Control Object speed button.

A cursor that looks like a resistor will appear in the schematic editor window, waiting for you to identify the component for which you wish to assign a stepped parameter. Point at the desired component, right click, and all of its associated parameters will be displayed in the dialog window.

Should the desired parameter belong to a catalog component or be an excitation attribute, you must first select the semiconductor or waveform type so that you can access the appropriate data sheet.

In the case of a resistor, for example, the following dialog window will appear:
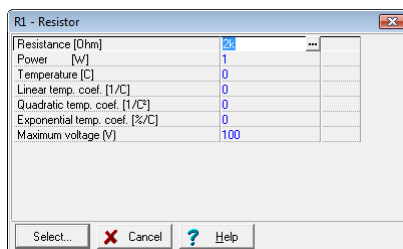

Select the line containing the parameter to be stepped and then click on the Select button. The following dialog will appear:
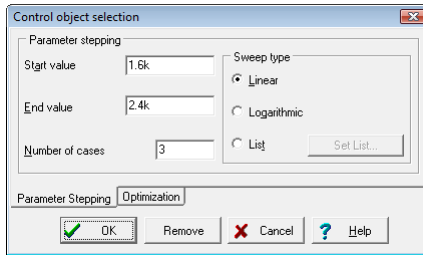
Set the appropriate stepping parameters. The number of cases includes the Start and End values.

By setting the number of steps to 1, you can perform a single analysis at the value defined by the Start value of the parameter. This gives you a way of examining the response with only a temporary change of parameter (without modifying the original value).

If you press the List button and then press the unlabeled button to the right, the List Editor for parameter stepping will appear:
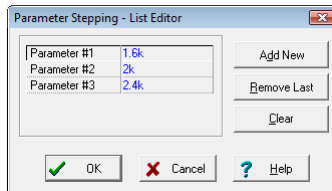

By default it will show three values calculated between the Start and End value, but you can enter any values you wish, remove values, or clear the list entirely.

Press OK to return to the previous dialog window which, in turn can be closed by its OK button. You can also turn off parameter stepping for this component using the Remove button.

You can repeat the above procedure for any number of parameters to be stepped.

Note that you do not have to use the same number of cases for each value being stepped. TINA will run an analysis for all possible combinations of values stepped. For example, if you ask to vary part A in three steps and part B in four steps, you'll get twelve curves.

The program will go into parameter stepping mode automatically whenever a component is set as control object according to the procedure above. The mode status can be checked via the Analysis Mode Selection dialog window.
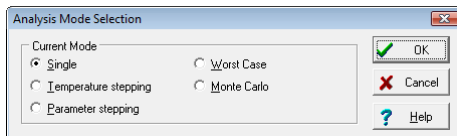
You can use this dialog to select single analysis or parameter stepping analysis, as well as Temperature Stepping, Worst Case and Monte Carlo analysis.
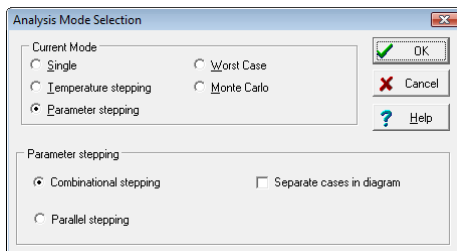
When stepping two or more parameters, TINA will use one or the other of two main modes of parameter stepping:

Combinatorial stepping: with combinatorial stepping, the stepped parameters can have different numbers of steps. TINA will run an analysis for all possible combinations of values stepped. For example, if you ask to vary part A in three steps and part B in four steps, the analysis will present twelve curves.

Parallel stepping: In this mode, the program will change all the values to be stepped simultaneously, yielding a quantity of analysis curves equal to the smallest number of values of any of the parameters being stepped. First TINA will compute the curves for the Start values for each stepped value. Next, the program will compute the curves between the Start and End values for each stepped value in isolation from each other and does not use the values from other cases in combination. We suggest when running a parallel stepping analysis that you give each parameter the same number of steps–it will keep things simpler.

For one stepped parameter the above two modes are equivalent.

An example of the two modes will clarify these modalities. If you ask to vary values of parts A and B, both in three steps you'll get only 3 curves with Parallel stepping and 9 curves with Combinatorial stepping.

EXAMPLE: Let the stepped values for R be 100, 200 & 300 Ohms, and for C be 500p, 1000p & 1500p Farads. TINA will run these cases with the following values:
**Combinatorial stepping (9 analyses)**
**Parallel stepping (3 analyses)**

R	C		R	C
100	500p		100	500p
200	500p		200	1000p
300	500p		300	1500p
100	1000p			
200	1000p			
300	1000p			
100	1500p			
200	1500p			
300	1500p			

Separate cases in diagrams. By default, all stepped curves are drawn in the same diagram. However, if the *Separate cases in diagrams* checkbox is checked, then each stepped curve is presented in its own diagram, with its own tab, in the Diagram Window.

Using **Separate cases in diagrams** and **Parallel stepping** makes it easy to create, run and document test benches (a suite of test circuits, each with a particular set of values that simulate devices under test). The following examples will make Parameter Stepping clearer. Load the file RLC_1.TSC from TINA's EXAMPLES folder. Note that this circuit is already in combinational parameter stepping mode for the resistor R, which is changed between 100 ? and 1.1k? in 3 steps. Click the ▯ button, select the R resistor with the new cursor, and press Select. You should see the following dialog window:
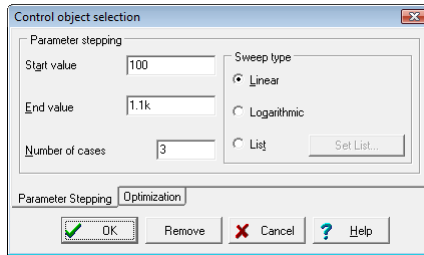
After closing this dialog and running Analysis/AC Analysis/ AC Transfer Characteristic, the following diagram will appear:
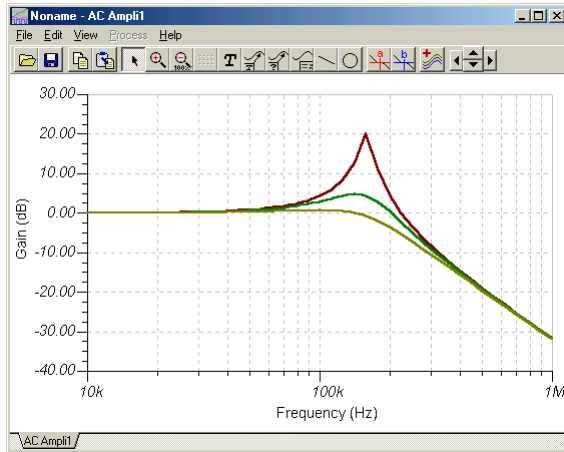

To determine which value of the stepped parameter is associated with a curve, move the cursor over the curve and read the corresponding value in the line at the bottom of the screen. You can also use the Legend and Auto label buttons in the diagram window to show the corresponding stepped values. Refer to the diagram below.
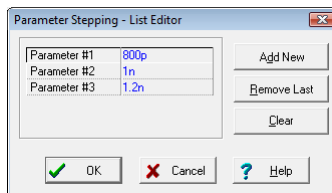
ADVANCED TOPICS

DesignSoft
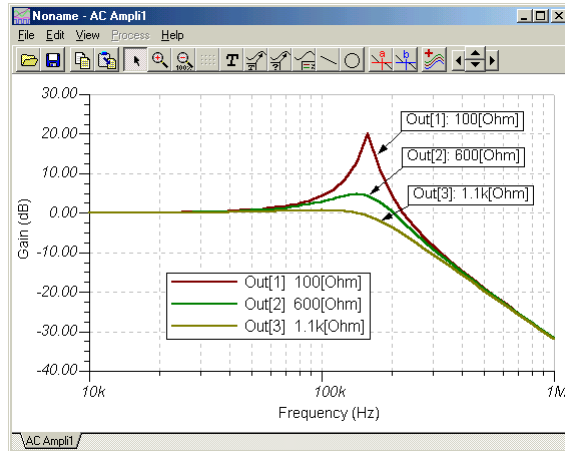
Now let's step the value of the capacitor using the List option. Let the values be 100pF and 1nF.

Click the ![button] button, select the capacitor C, and press Select. You should see the following window:
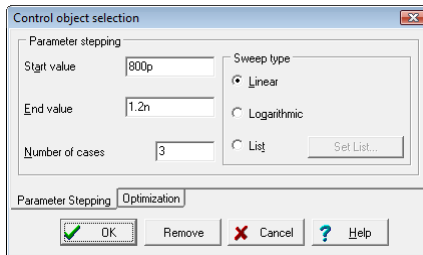
Select the List option and press the button to its right. The following dialog will appear:

ADVANCED TOPICS

DesignSoft

Press the Remove Last button and set the first parameter to 100p. Now close all the dialog windows by pressing their OK buttons and run Analysis/AC Analysis/AC Transfer Characteristic.
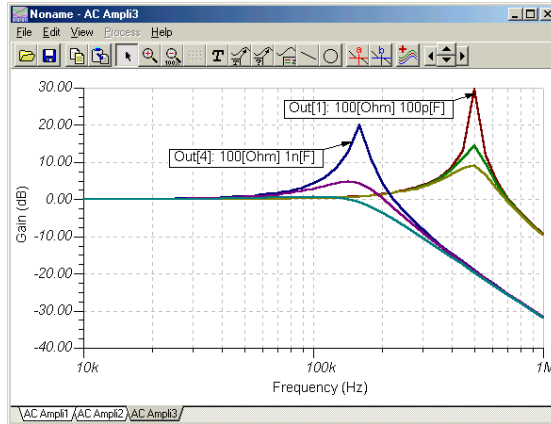

The new set of curves belonging to the cases where C=100pF can be clearly recognized on the right side of the diagram. We have also added two auto labels showing the typical values of the parameters. If you have multiple curves in the diagram window you can also separate them to independent vertical axes for clearer view. Just select the "Separate curves" in the View menu and you will get the following diagram:

DesignSoft
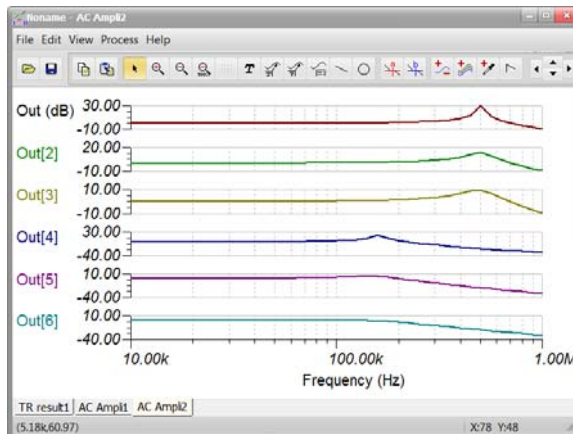
Now let's add another output — a voltage pin — to the circuit between the resistor and the inductor. We will call it Vrl.
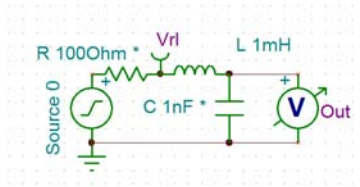
With this change AC transfer calculation will already generate 12 curves.
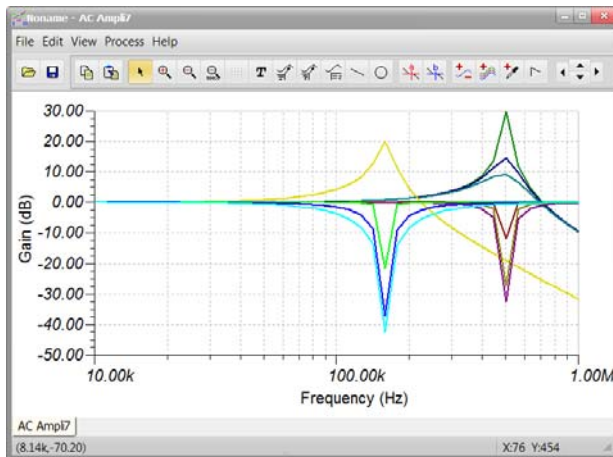
If you want to analyze the curves independently you can again use the "Separate curves" menu. However, in this case it would be much better to keep the curves belonging to the same output together. To reorganize the curves based on their output, first select "Collect curves" (if the curves are already separated) then click on "Separate Outputs" in the View menu.
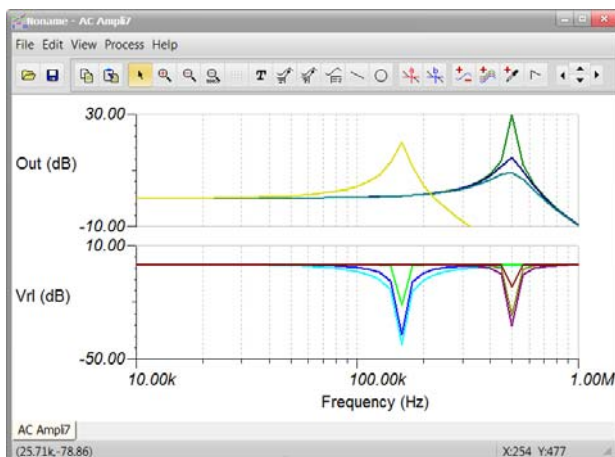
Curve separation can be also done automatically using the appropriate output labels. To automatically separate all the curves please add a colon (:) and an integer index to the output label names. E.g. renaming "Out" to "Out:1" will force the analysis to automatically separate the curves after calculation, and the index at the end of the label will be used to set the order of the curves from the top to the bottom of the diagram. If you want to separate only the outputs please use double colon (::) and an index number. E.g. "Out::1" Please also note, that you do not need to specify all the curve indexes, the curves will be separated already if one is specified. Unspecified curves will be put at the end of the curve's list on the bottom of the diagram.

To demonstrate **Parallel Stepping** open the RLC_1 Parallel Parameter Stepping.TSC circuit from the EXAMPLES\Parameter Stepping folder of TINA. For this circuit the Analysis Mode Selection settings:
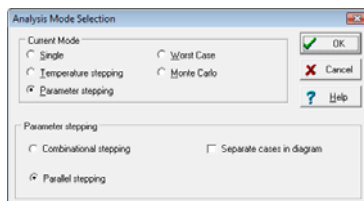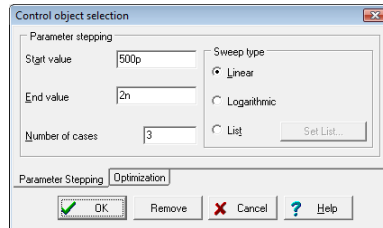
The parameter settings for R
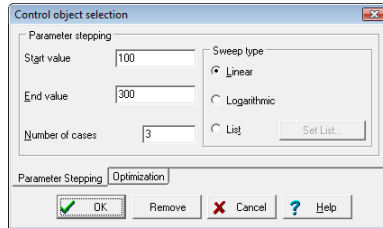
The stepped values will be: 100, 200 and 300 Ohms
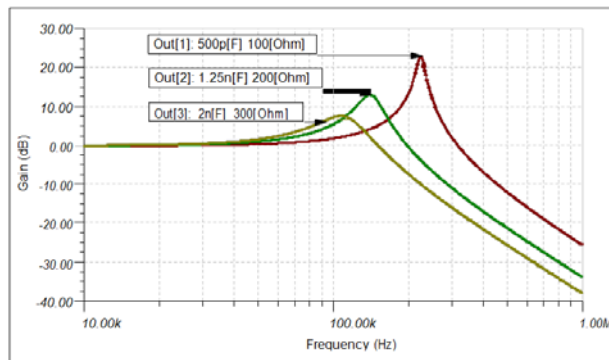The parameters setting for C:

The stepped values will be: 500pF, 1.25nF, 2nF.
The AC Transfer Characteristics:

Note that the labels on the curves were generated by using the Auto Label button of the Diagram Window.
Finally to learn about the effect of the *Separate cases in diagram* checkbox, enable this option in the Analysis Mode Selection dialog as shown below:
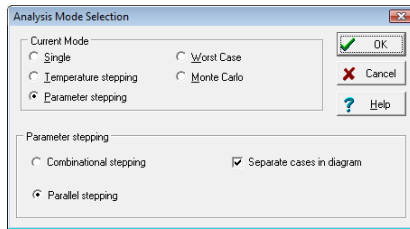
Now calculate AC Transfer Characteristics again. After adding the labels you will have 3 separate diagrams under 3 different Tabs of the diagram window.

For the first step under the tab AC Ampli1:
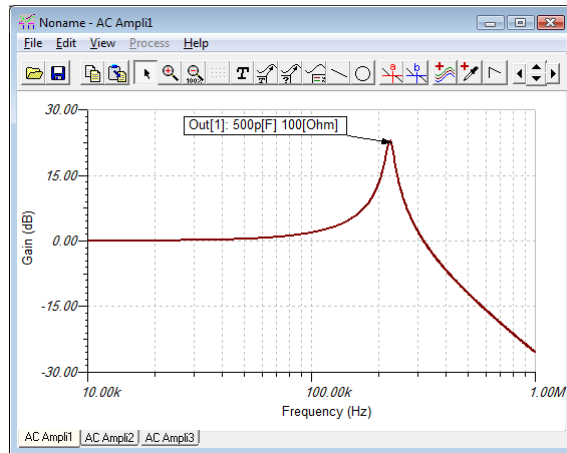
For the second step under tab AC Ampli2:

DesignSoft
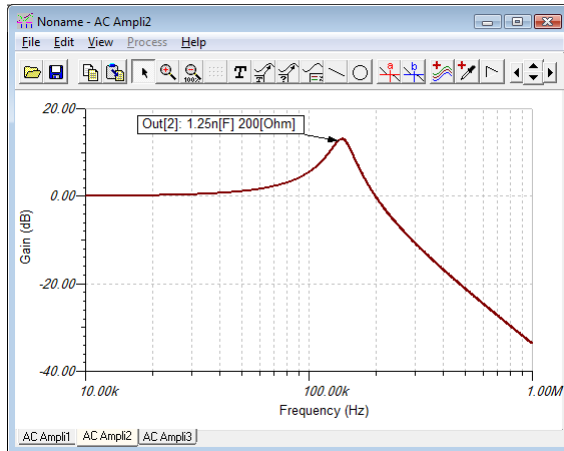
and finally for the third step:
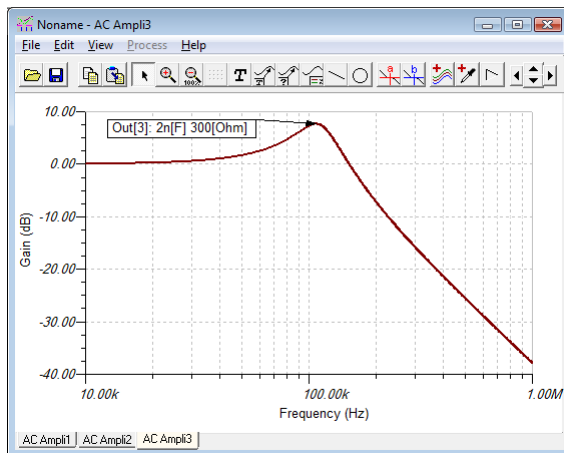
The curves are the same but shown in separately in different diagrams as the curves for the Parallel stepping without the *Separate cases in diagram* option. You can use these separate diagrams to document test benches, a suite of test circuits, each with a particular set of values that simulate devices under test.

9.3 DC Transfer Characteristic and parameter sweeping

To calculate a DC transfer characteristic, you should assign exactly **one** input (normally a voltage or current source or generator, or a resistor) and at least one output. Later, you can change the assigned input in the DC Transfer Characteristic dialog window. You can also add more outputs using the post- processor. Plotting a result as a function of a range of resistor values is called DC parameter sweeping.

There are several ways to assign a component to serve as the input.

1. In TINA Version 6, the simplest way is to select the part from the Input list box of the DC Transfer Characteristic dialog window. Once selected, TINA will remember it and it will be saved as the selected input in your circuit file.

2. Use the I/O state field of generators and sources to assign them as input.

3. You can also use the Insert/Input command to assign an input to a source/generator or to a resistor. First select the Insert/Input command and see a special $\mathbf{I}^+$ shaped cursor appear. Click the first node of the component to be selected and hold down the mouse button while you drag the input to

the other node. The Insert/Input command is not the easiest way to select a DC Transfer input: method one or two is preferred.

To assign an output, use any appropriate meter from the Meters Toolbar (Voltage Pin, Voltage Meter, Ampere Meter, Power Meter, etc.) or use the Insert/Output command. In most cases, a meter is the best choice; however, if your output is on nodes far from each other, the Output command might be useful.

After you have assigned at least one output, you can invoke the **Analyis/DC Analysis/DC Transfer Characteristic** command.

ADVANCED TOPICS

DesignSoft
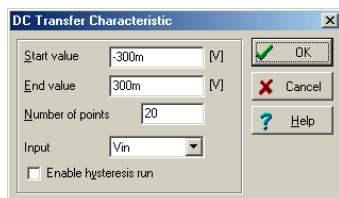
Explanation of the parameters:

Number of points: The program calculates the transfer characteristic at the number of points defined. The default value is 20.

Start Value and **End Value:** The program evenly divides the specified input range so that the number of points includes the start value and the end value. For each input

point, the program calculates the output value and uses it to build the transfer characteristic. The units indicated in the square bracket correspond to that of the input (e.g., V for a

voltage generator). Normally, the input values are generated in increasing order. However, if you prefer calculation in a decreasing order, assign the greater limit of the input range

to the Start value and the smaller limit to the End value. The order of calculation only matters with the DC characteristics of circuits with hysteresis (e.g., Schmitt trigger).

Input: Here you can check and change, if desired, the input. To change the input, click the arrow in this field and the available input quantities (label names of sources and resistors in the circuit) will appear. If a component has no label, TINA will automatically label it in the form V _no_label_1. Click on the name to select it as the input.

Enable hysteresis run: If this checkbox is checked, TINA will first run an analysis varying the Input quantity from the **Start value** to the **End value**, and then run an analysis from the **End value** to the **Start value**. You should use this option for components with hysteresis (where the characteristic depends on the direction of change).

Note that if you use a resistor as an input, TINA calculates the transfer characteristic multiple times, once for each step of the resistance (parameter sweeping).

When running an analysis on a nonlinear network, the DC analysis might not converge at certain points. These points will be skipped when plotting the response. If they are at the start and/or end values, the displayed (input) range will be truncated accordingly.

Let's look at two examples.

First, let's draw the characteristics of a bipolar transistor

(CHARBIPO.TSC)
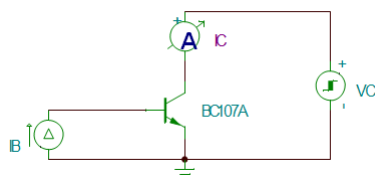
Select the current source and Analysis/Mode to see how the base current, IB, is stepped.
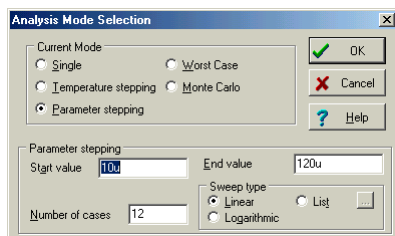
Then invoke the Analysis/DC Analysis/DC Transfer Characteristics command. See the following dialog:
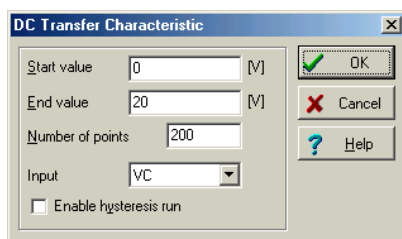
According to the dialog window, the input, VC, will be varied form the Start value (0V) to the End value (20)

The result is familiar family of curves that constitute the common emitter characteristic of a transistor.
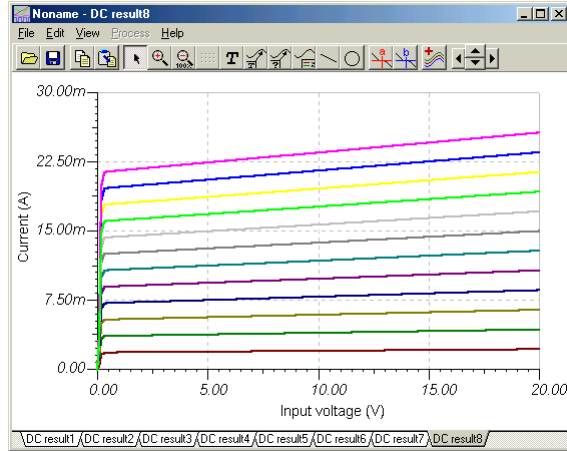
Now let's see an example of parameter sweeping (sweep.tsc)
The circuit:
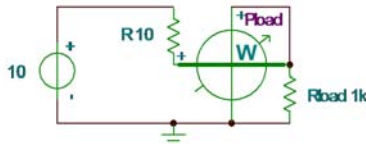
The output is the power meter, showing the dissipation on Rload.

If you invoke the Analysis/DC Analysis/DC Transfer Characteristics command, you'll be greeted by this dialog box:

Note that the resistor Rload is swept from 0 to 100 ohms.

Press OK to obtain the following diagram:
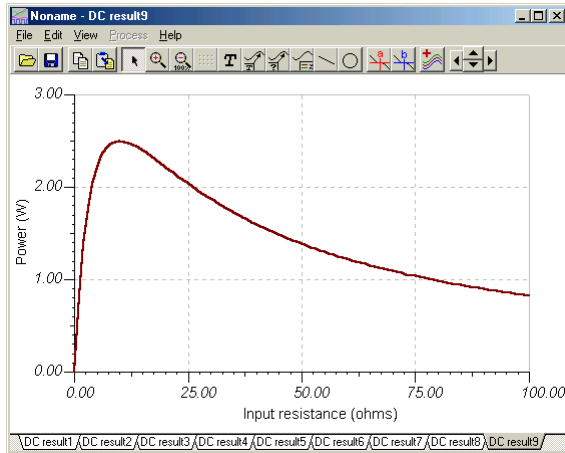
This is the well known curve of power dissipated on a load resistor as the load resistance is varied. Maximum power is transferred when the source resistance and load resistance are equal.

9.4   Phasor Diagram

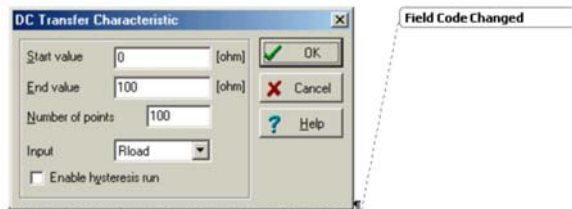The phasor diagram is a great tool for demonstrating and studying circuit operation via complex phasors (vectors). Phasor diagrams can be used only with AC analysis with linear circuits, or with nonlinear circuits in working point linearization.

Let's use TINA to draw the phasor diagram of an RC circuit. First open the circuit RC.tsc in the examples\phasor directory.

DesignSoft
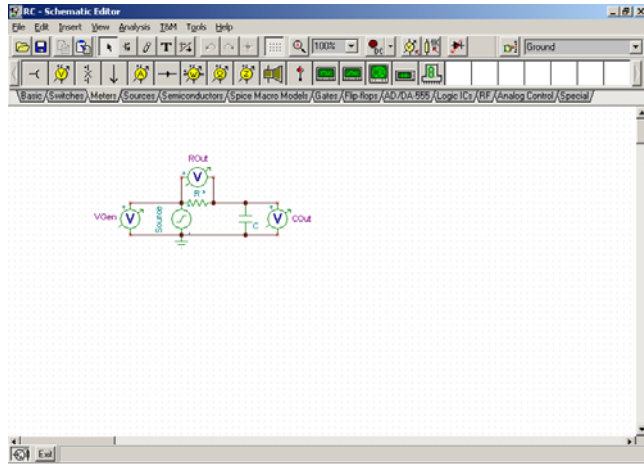
Run the *AC Analysis.Phasor diagram* command. TINA generates a new diagram containing three vectors in the diagram window . As you can see, Vgen is the vector sum of Rout and Cout, but to make this clearer you can change the position of the vectors.

By default all the vectors originate from the origin of the coordinate system; however, you can translate them by simply clicking and dragging them into another place. When you drag a vector close to another vector's starting or ending point, it will be automatically snapped to it. Try dragging the vector Rout to the end of Cout to form a triangle with Vgen. Notice that when you move the cursor above a vector, TINA presents its name, amplitude, and phase in the status- line.

ADVANCED TOPICS

345
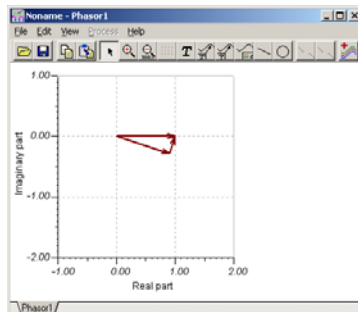
If you prefer a display without axes, turn the axes off using the *View.Display axes* menu.

If you double click on a vector (or right-click it and select Properties), the vector's configuration dialog box appears. In addition to the usual settings for the vector's color, width, and style, you can also set the vector head style.
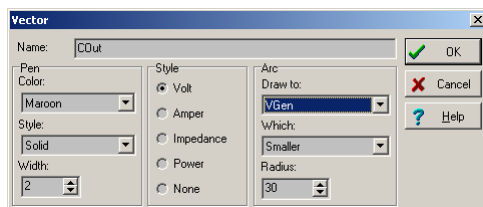
Using the Arc settings, you can draw an arc between Cout and Vgen by setting Vgen in the *Arc.Draw to* listbox of the Cout vector's configuration dialog box. You can also set the radius of the arc in pixels and choose whether the smaller (<=180°) or the bigger arc (>180°) should be displayed.

Next, you can add labels to all the vectors at once by pressing the legend button on the toolbar. You can also add legends individually: press the *Auto label* button, then select the vectors one-by-one. The actual label format depends on the selected *Vector label style* under the *View* menu. Finally, you can add lines, circles, or texts to finish your diagram.
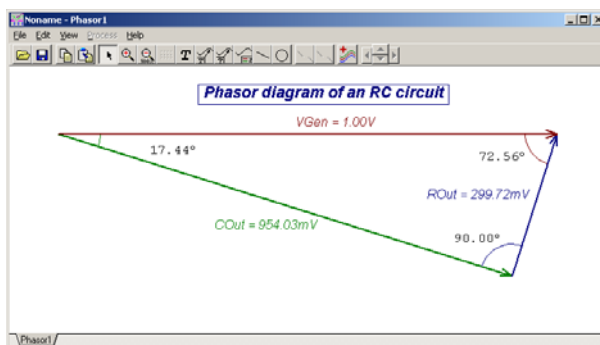
This diagram can then be copied to the clipboard and pasted into the schematic editor or any word processing program.

ADVANCED TOPICS

DesignSoft

9.5   Nyquist Diagram

TINA's AC transfer analysis can plot not only the Amplitude, Phase and Group delay of the transfer function of analog circuits, but the Nyquist diagram as well. To see how this works, first load the examples\rlc_1.sch circuit. Next, select the Analysis.AC Analysis.AC Transfer Characteristic menu item. Now the AC Transfer Characteristic dialog box appears on screen, where you can select the Frequency ranges of the analysis, the sweep type, and the diagram you want to plot (Amplitude, Phase, Nyquist, Group Delay).
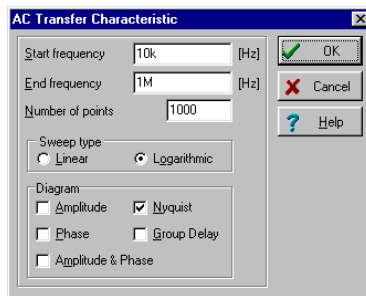
Set the frequency from 10k to 1M and set the diagram type to Nyquist. In order to get a smooth curve, increase the Number of point to 1000. Press the OK button and see the Nyquist diagram of the circuit calculated and plotted, as shown in the diagram window.

DesignSoft
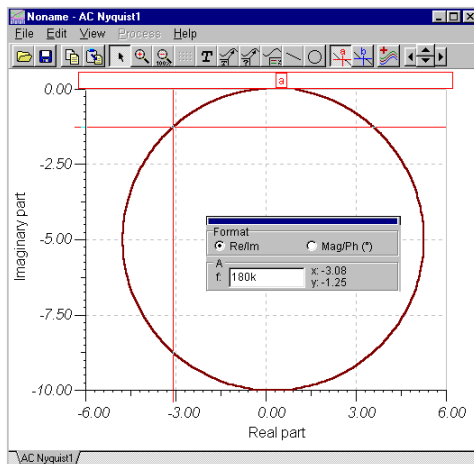
This diagram is different from a normal Transient or AC analysis diagram. A Nyquist diagram requires that horizontal and vertical scales always be proportional to each other. This keeps circular curves circular, even if you resize the window. Furthermore, a Nyquist curve presents the trace of a vector quantity as frequency is varied. When you activate a cursor and explore the trace, the cursor dialog shows the numerical values of the curves at a given frequency, in either real-imaginary or amplitude-phase format.

9.6   Noise Analysis

In TINA there are two models of noise analysis available:

- Small signal AC Noise Analysis
- Transient Noise Analysis

9.6.1   Small signal AC Noise Analysis

In this Analysis mode TINA calculates the noise contributions of circuit components and then presents them referred to the output in frequency domain. The random noise fluctuations are analyzed statistically, considering the time domain average to be zero, while the variance and sum of the squares are not zero. Noise analysis results can be displayed and printed over a frequency range similar to the Bode diagram of AC analysis.

TINA takes into account the noise sources inherent in resistors and semiconductors (transistors, diodes). Resistors are modeled as an ideal noiseless resistor in parallel with an equivalent thermal noise current. This equivalent noise is modelled by white noise, i.e., noise containing all frequency components with the same amplitude density. The contributions of all the noise generators in the circuit are calculated to determine each one's magnitude at the output. The 'output noise' is formed from the square root of the sum of the squares (rms value) of these individual contributions.

To correctly account for the noise contribution of operational amplifiers, you should use models made by manufacturers by choosing devices from the Spice Macro Models toolbar. Open the macro and check the noise model. Some manufacturer models have special purpose noise models, or no noise modelling whatsoever. TINA's ideal, standard and linear opamp models (on the Semiconductor toolbar) *do not include* a noise model.

Since the transfer functions of the circuit are known, the equivalent input noise can be calculated from the output noise. Expressing all the circuit noise in terms of an equivalent input noise makes it possible to compare the levels of real signals presented to the input to the circuit noise referred to the input. A circuit expected to amplify small input signals, signals approaching the magnitude of the equivalent input noise, will not be very useful.

Resistors generate resistive (thermal) noise which is computed as the following source current:

$$i^2 = \frac{4 \cdot k \cdot T}{R}$$

where

k = 1.38e-23 (Wsec/K)   Boltzmann constant

T = temperature in K   R = resistance in ohms

Semiconductors generate not only thermal noise (similar to resistors), but flicker noise (1/f noise) as well. The spectral density of flicker noise decreases in inverse proportion to the frequency, hence the name 1/f noise.

TINA calculates the Input and Output noise and expresses the noise as the noise voltage per $\sqrt{Hz}$ of bandwidth.

TINA can also present noise as a curve of Total noise, at the output, as a function of frequency. In computing this curve, TINA sums all the noise from the start frequency up to the specified maximum frequency value and presents it as total noise. Naturally, the curve rises monotonically. For example, if you set the start frequency to 20Hz and the end frequency to 20kHz, you will learn the total output noise over the typical audio bandwidth.

Another measure of noise, the "signal to noise ratio," takes the signal magnitude into account. This is the ratio of signal power to noise power, according to the following formula:

$$S/N = 20 * \log \frac{V_{Sig}}{V_{Tot}}$$

where V_{Tot} is the total input-referred noise voltage and V_{Sig} represents the expected signal amplitude. V_{Sig} is a user-defined value that you enter in the Noise Analysis dialog. The default value is 1mV.

ADVANCED TOPICS

DesignSoft

To get a better feeling for these noise concepts, let's study the following circuit, an equivalent circuit of a basic amplifier (noise.tsc).
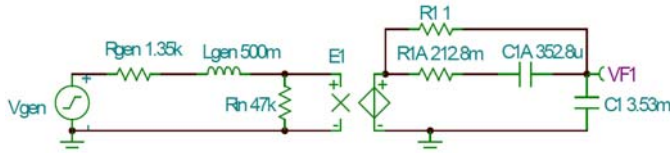
Select Analysis.Noise Analysis. The following dialog appears:
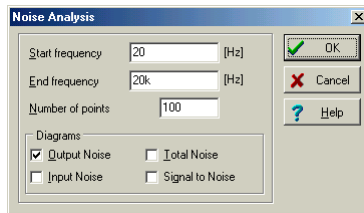
The dialog allows you to set the **Start frequency**, **End frequency**, the **Number of points** for which the noise is to be computed, and the **Signal amplitude**. You may also select the required analysis results– Output Noise, Total Noise, Input Noise, and Signal to Noise.

Output noise

DesignSoft
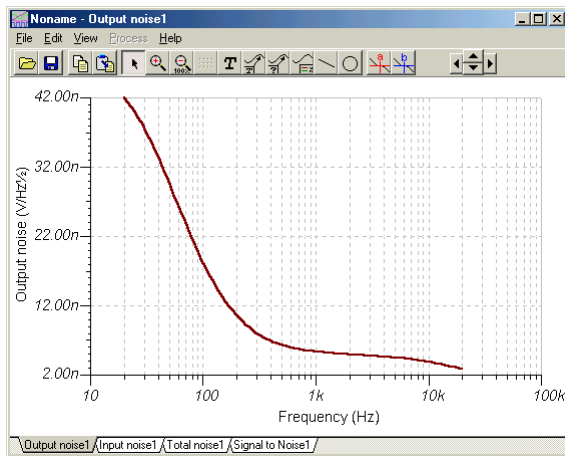
Input Noise

Total Noise

DesignSoft

9.6.2 Transient Noise Analysis

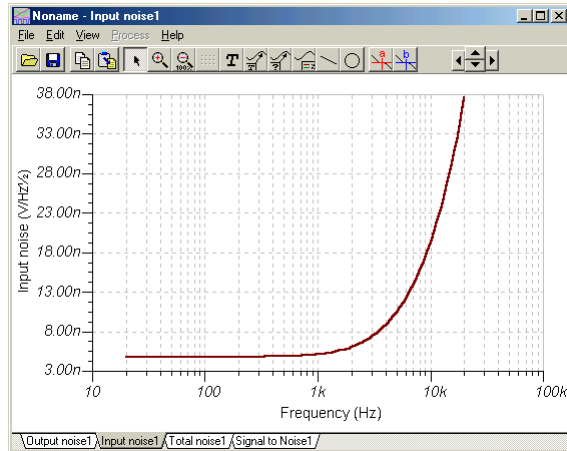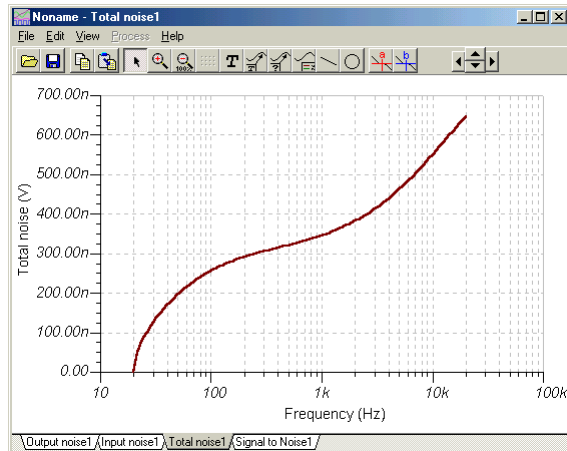Noise effects are usually simulated with linear AC noise analysis which is also available in TINA. However when the noise influences the system behavior in a nonlinear way, linear noise analysis is no more satisfactory and transient noise analysis, that is simulation in the time domain, is necessary. Here are a few examples:

• Noise analysis of oscillator circuits

• Analysis of noise effects in digital circuits

• Analysis of circuits with noisy input signals

• Analysis of systems with low signal-to-noise ratio

The voltage and current generators of TINA now include a parameterizable white noise signal, and in the Noise Analysis folder of TINA application circuits are available to generate other typical noise signals, which make transient noise analysis possible

Let us demonstrate transient noise analysis with an oscillator circuit where the linear AC noise analysis is not possible since oscillator circuits are strongly nonlinear systems. Our example is based on the following publication: A Study of Phase Noise in CMOS Oscillators by BehzadRazavi, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 31, NO. 3, MARCH 1996

We refer to the above paper for the theoretical details, here we just describe the main simulation settings.

Open the Relaxation Oscillator Noise.TSC circuit from the Noise Analysis folder of TINA.

DesignSoft
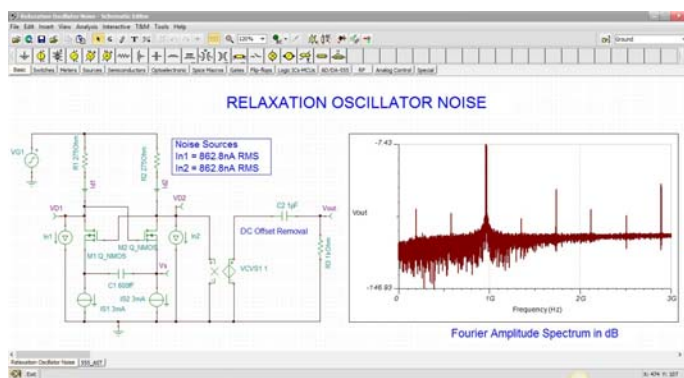
The noise is injected into the circuitthrough the In1 and In2 white noise generators with 862.8nA RMS both. Doubleclick on In1 and observe that the Signal type is set to Noise.
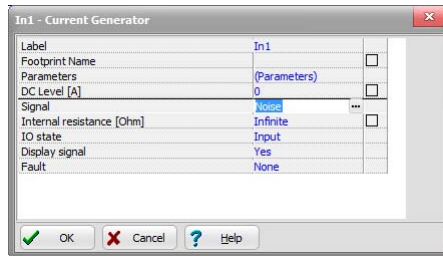
If you click the … button in the Signal line the settings of the white noise generator appears.
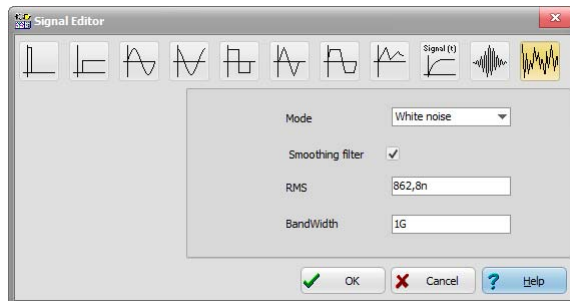
The result was obtained by Fourier Spectrum analysis. You can either run a Transient Analysis then click the waveform and from the Process menu select Fourier Spectrum or select Fourier Spectrum directly from the Analysis menu. In both cases the suggested Fourier Spectrum dialog setting are as shown below.
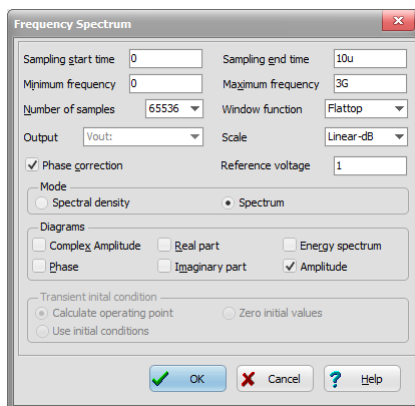
As usual you can place the diagram on the schematic editor using copy in the diagram then Paste in the main schematic window of TINA.

9.7   Network Analysis and S-parameters

TINA helps you perform network analysis and determine the two-port parameters of networks (S, Z, Y, H). This is especially useful if you work with RF circuits. Results can be displayed in Smith, Polar, or other diagrams.

The network analysis is carried out with the help of TINA's network analyzer. The RF models of the circuit elements can be defined as SPICE sub-circuits (SPICE macros) which contain parasitic components (inductors, capacitors) or as an S-parameter model defined by its S (frequency) function. S functions are normally provided by the component manufacturers (based on their measurements) and can be downloaded from the Internet and inserted into TINA either manually or by using TINA's library manager.

Let's get started with an example. Open the circuit examples\rf\spar_tr.sch.
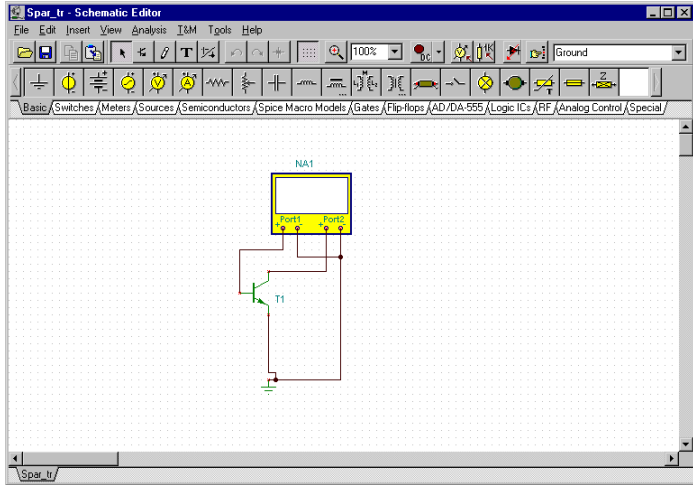
This circuit consists of a network analyzer connected to an RF transistor. The first port of the network analyzer measures the base- emitter characteristic while the second port measures the collector- emitter characteristic of the transistor. If you double-click on the transistor, you will see that the actual characteristics of this transistor are defined by its S11, S12, S21, S22 parameters, and R1 and R2 input/output impedances.

Each function is defined as a table of the S parameter and frequency. If you click on the … button beside the Complex label, you can see this table.

DesignSoft
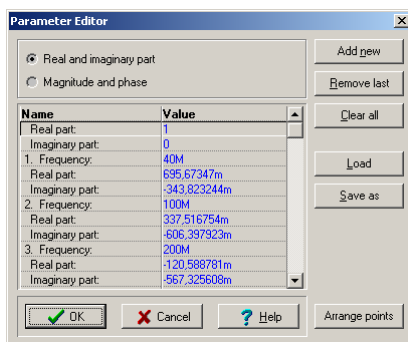
You can easily add more points to the function with the *Add new* button or load an entirely new function with the *Load* button. Similarly, the *Remove last* button removes the last point, and the *Clear all* removes all the points from the list. The *Arrange points* button sorts all the points in ascending order of frequency.

Note that TINA comes with many other models defined by S- parameters (under the RF toolbar).

Close all the dialog boxes and run Analysis.AC analysis.Network analysis. Among many other parameters in the dialog, you can specify the frequency range and the mode (i.e., what to draw) of the analysis.

DesignSoft
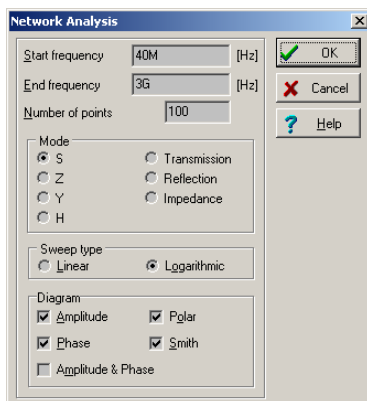
The diagrams available depend on the selected analysis mode. In S Analysis, the Amplitude, Phase, Polar, or Smith diagram cans be drawn. Let's select Analysis mode S, the Amplitude, Phase, Polar and Smith diagram types, and press OK. Four new pages are inserted into the diagram window. Each new page contains four curves, the S11, S12, S21, and S22 curves, displayed in the appropriate diagram style. You can obtain specific data values by placing a cursor on the desired points. When you place the cursor on a Smith curve, a special cursor diagram comes up, which can display the curve's values in the format set with the *Format* and *params* (S-params, Z-params, etc.) fields.
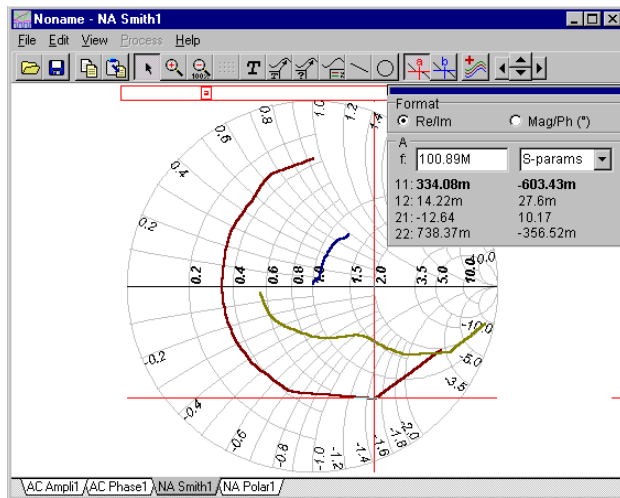
You can annotate these diagrams, as you can any other diagram types, by adding texts, circles, and lines. You can explore data values of particular points on the curves using the cursors, and zoom in and out. But for Smith and Polar diagrams, axis settings are handled differently. Even though these diagrams look like polar diagrams, the coordinate ranges are defined in the real-imaginary domain. The real part (i.e. real axis) can be set by double clicking on the horizontally aligned numbers (on the Polar diagram the amplitude values), and the imaginary part by double clicking on the circularly aligned numbers (on the Polar diagram the Phase values). You can set the format of the numbers and the horizontal or vertical range of the diagram using the dialog that appears.

ADVANCED TOPICS
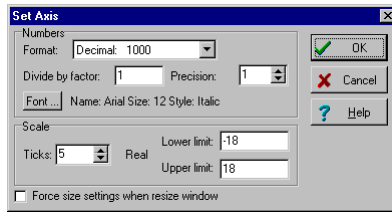
Models for Network analysis can also be defined using SPICE subcircuits. To demonstrate this, open the circuit examples\rf\spice_tr.sch in the schematic editor. This example contains a circuit similar to the one we've just looked at: a network analyzer connected to a transistor, and a current source and a capacitor.
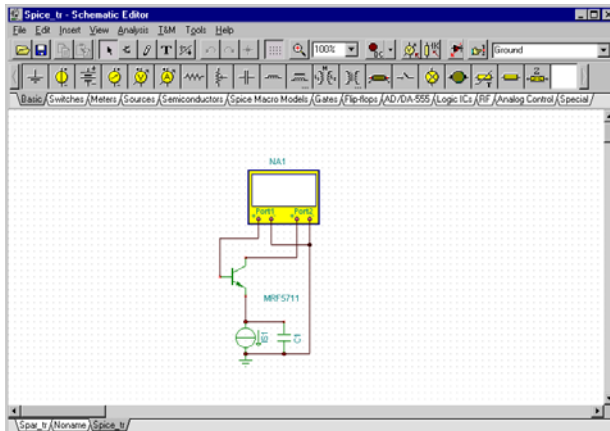
In this circuit, however, the transistor contains a SPICE subcircuit. Its netlist can be viewed by clicking *Enter macro* in its property dialog.
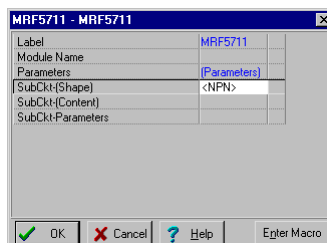
TINA contains many other RF models that can be found under the Spice Macro Models toolbar.

Now run Analysis.AC Analysis.Network Analysis again and note that the results are similar to those obtained with the S-parameter models.

Network analysis curves can also be generated using TINA's Network Analyzer under the T&M menu. To select the diagram you need, first set the Network Analyzer Mode, then the display type. After you press *Start*, the selected curves will be displayed on the instrument.
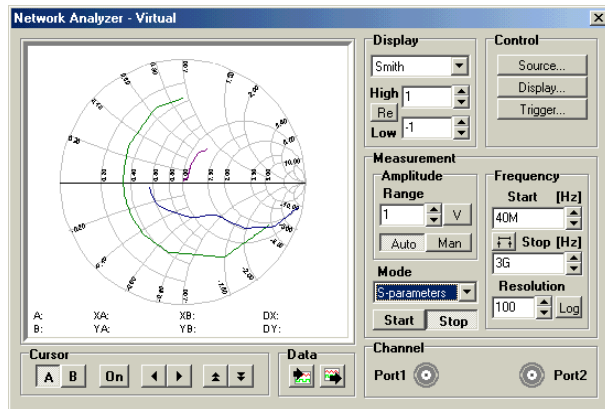
9.8   Symbolic Analysis

9.8.1   Symbolic Analysis

TINA's symbolic analysis produces closed form expressions of the transfer function, equivalent resistance or impedance, as well as the DC, AC, and transient response of linear circuits. These equations can then be inserted into the schematic editor or the diagram window. With the help of the built-in interpreter, they can be even further manipulated and displayed as a diagram in the diagram window. The formulas provided by symbolic analysis allow a unique in-depth understanding of circuit operation, invaluable both for design and education.

DesignSoft

Let's see a few examples. First load the circuit examples\symbolic\actfilt.sch. It is an active filter with an ideal operational amplifier.
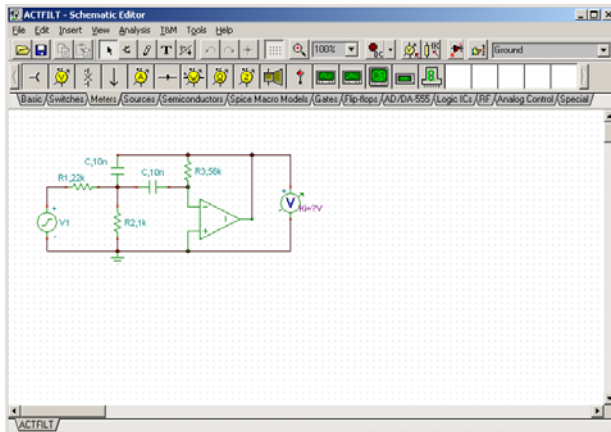
Now run Analysis.Symbolic Analysis.AC Transfer. The transfer function is generated and displayed in the Equation Editor.

All the variables (values of the capacitors, resistors, etc…) were treated as (unknown) symbols, so the expression contains their symbolic names. However, in case you want to calculate the numerical form of the transfer function using the circuit element's actual values, you need to click on Analysis.Symbolic Analysis.Semi-symbolic AC Transfer, and the following expression will appear.

DesignSoft

This expression can then be copied to the clipboard (by pressing the copy button on the toolbar) and pasted to the schematic editor or to any of the diagram pages.
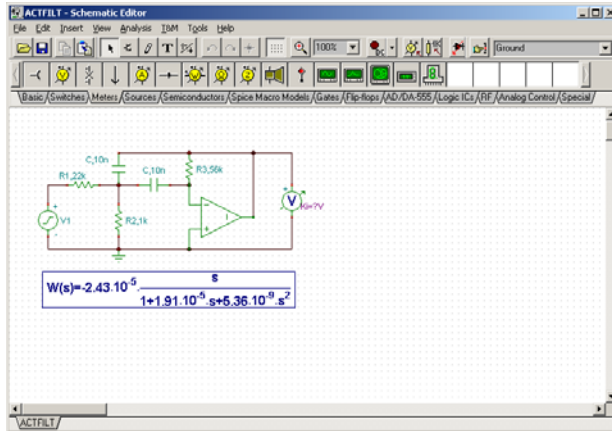
You can also generate a symbolic result with only a few symbolic variables, while the numerical values of the rest of the variables have been substituted. Symbolic analysis takes the symbols of the variable from the circuit element's *labels* in the schematic editor. When the label is empty or has been determined to not be a valid symbol name, symbolic analysis uses the part's numerical *value*. For example, if you change the label of the resistor R1 to "22k" or to "R1=22k", then the value of the part (22k) will be used in place of the symbol R1.
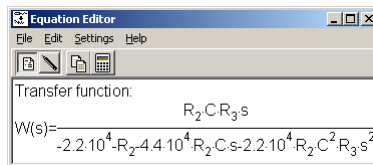
TINA goes beyond simply presenting the function as an equation. Any of these functions can also be interpreted and drawn in the diagram window. To demonstrate this, let's first generate the transient response of an RLC series resonant circuit. Load the examples\symbolic\myrlc1.sch circuit into the schematic editor.

DesignSoft
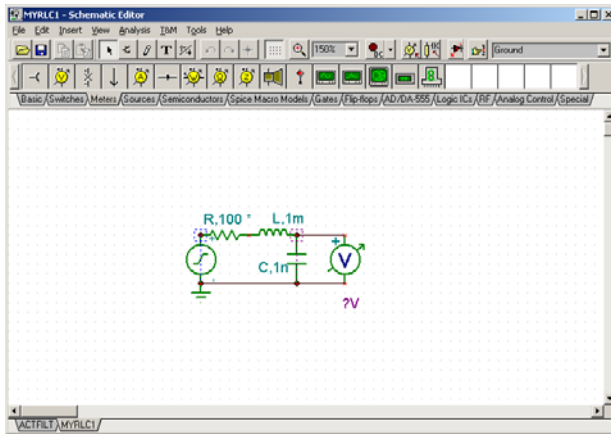
Next, run Analysis.Symbolic Analysis.Semi-symbolic Transient. The transient response of this circuit to a voltage generator with Unit step waveform is then the following:

TR Result :

$$v(t)=(1+1 \cdot e^{-5 \cdot 10^4 \cdot t} \cdot \cos(9.99 \cdot 10^5 \cdot t - 182.87°)) \cdot \varepsilon(t)$$

Now press the small calculator (interpreter) button on the toolbar. The interpreter window appears, and our transient function is already defined as a v_1 function.

As you can see, the function definition is followed by a few drawing preference settings, where you can define the ranges of the function and the number of points at which the function will be calculated. Finally, the last command draws this function in the diagram window. The second parameter of the draw command defines the name of our newly generated function. After pressing the ▶ Run button, the function appears in the diagram window.
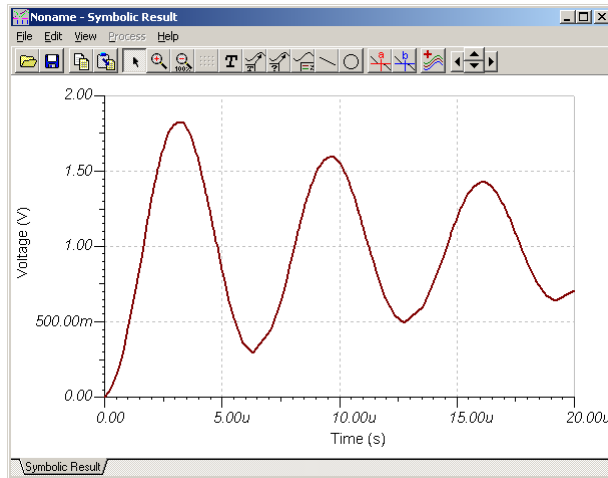
Symbolic analysis can also calculate the DC and AC results (response) of a linear network containing one or more sources. Just as with symbolic analysis of AC transfer functions, these results can be generated both in symbolic and semi-symbolic form. If we choose the symbolic form, all variables in the circuit that have symbolic identifiers (labels) will be treated as pure symbols. If we choose the semi-symbolic form, every parameter of the circuit will be handled as a number. When the circuit contains more than one generator (source), the result due to each of these generators will be calculated and added together (using the superposition law).

Let's use a simple parallel resistor circuit to get to know symbolic analysis of circuit results. Open the circuit examples\symbolic\suppos.sch.

DesignSoft
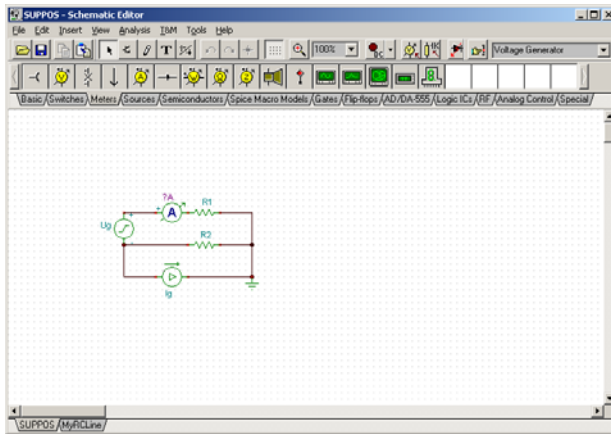
This circuit contains two generators: a voltage and a current generator. Now press Analysis.Symbolic Analysis.DC Result.

As you can see the result is a fully symbolic combination of the circuit's response to each of the generators.

Now press Analysis.Symbolic Analysis.Semi-symbolic DC Result.

In this Semi-symbolic DC Result, the coefficients of Igdc and Ugdc are calculated numerically. Of course, this requires that R1 and R2 be assigned both a label and a numerical value.

DesignSoft

9.8.2 Hierarchical Semi-symbolic Analysis

Depending on the circuit size and topology, the symbolic result of an average circuit can easily become a huge, complicated expression. This is especially true for circuits that contain components whose models contain further symbolic components, such as transistors.

The symbolic expressions can be simplified by identifying parts of the circuit that are less important, or whose effect are already better understood, and setting only numeric values (no entry in the label field) for these components. For other components that are more important, be sure to enter a label so that they will be treated symbolically in the analysis. For visual convenience you can use TINA's advanced macro feature to create two different macros for each of the component you would like to examine in detail: one with a symbolic model and another with a purely numeric model. To demonstrate this feature, let's create an RC transmission line where some of the RC sections are numeric and others are symbolic. The numeric macro will look this.
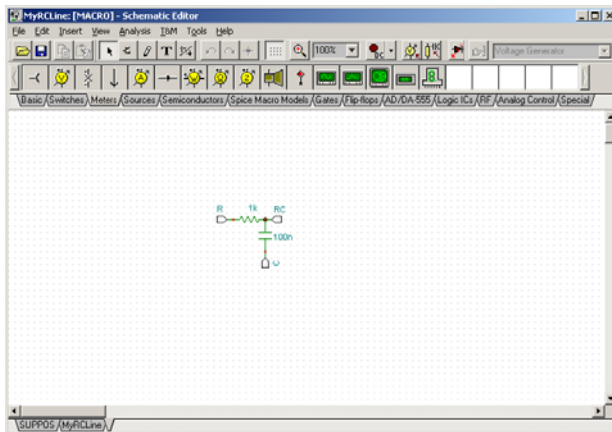
The resistor is 1k Ohm, and the capacitor is 100nF.

The symbolic model (below) has the same topology, but has symbolic R and C parameters.
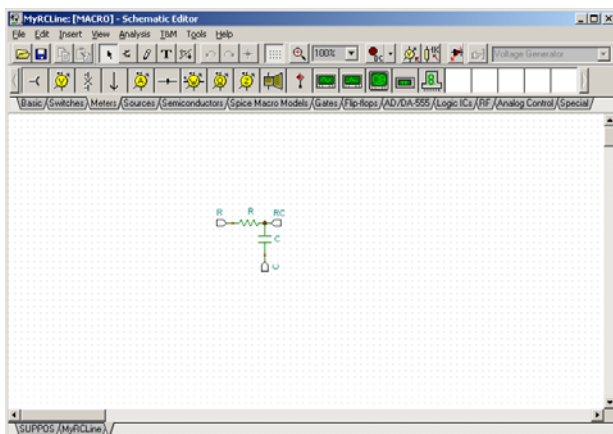
Now let's consider an RC transmission line circuit with three RC sections. Only the middle section is numeric.
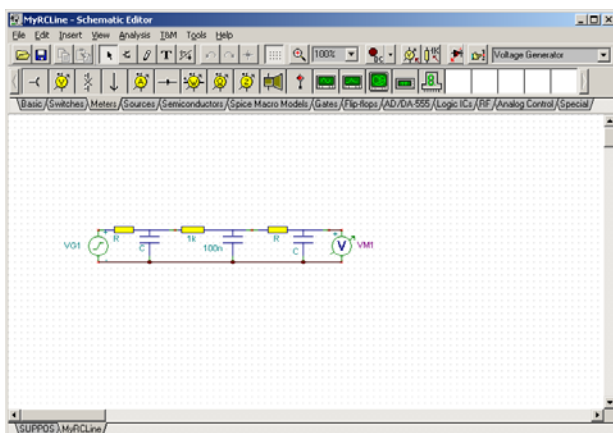
Pressing Analysis.Symbolic Analysis. AC Transfer, TINA calculates the result:

Transfer function:

$$W(s)=\frac{10\cdot10^6}{10\cdot10^6+(R+3\cdot10^7\cdot C\cdot R+1000+10\cdot10^9\cdot C)\cdot s+(R+10\cdot10^6\cdot C\cdot R+2000+10\cdot10^9\cdot C)\cdot C\cdot R\cdot s^2+1000\cdot C^2\cdot R^2\cdot s^3}$$

ADVANCED TOPICS

If we change the position of the numeric section of our circuit (by simply dragging the whole macro to its new position), we see:
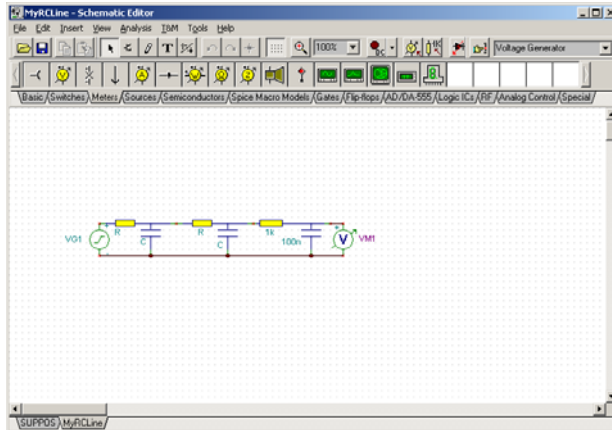
And the result changes as well:

Note that TINA's built in transistor models are purely symbolic: i.e., the bipolar transistor contains a base resistor and a current controlled voltage source, while the MOS and JFET contain a drain-source resistance and a voltage controlled current source. If you want to calculate the DC or AC symbolic transfer or result function of a circuit with several transistors, you can simplify the expressions by creating your own transistor models using only numeric components.

ADVANCED TOPICS

DesignSoft

9.9   Post-processing Analysis Results

9.9.1   Adding new curves to diagrams and post-processing analysis curves

TINA has a great tool to add new curves of virtually any node and component voltage or currents. In addition, you can post process existing curves by adding or subtracting curves, or by applying mathematical functions. You can also draw trajectories; i.e., draw any voltage or current as a function of another voltage or current.

If you check "Save all analysis results" in the Analysis.Options dialog, TINA will calculate and store all the results that are reasonable in the current circuit. However, only those results that have been set as output are inserted automatically into the diagram window.

To add these curves, or to add new, user-defined curves to a diagram, press the *Add more curves* button 📈 on the toolbar or call the *Add more curves* menu in the Edit menu. The post-processing dialog appears on the screen.

This dialog box consists of two parts: in the upper part you can select and insert/remove existing curves into/from the current diagram, while in the lower part you can define your own curves. The lower part of the dialog box is hidden by default. Press the "More >>" button if you want to create new curves.

<u>Upper part:</u>

When this dialog first opens, the *Available curves* list box contains all the curves that have already been calculated. The status bar shows the circuit file name and the analysis which was run. You can select and add any of the curves into the *Curves to insert* list box by selecting them in the *Available curves* list and pressing "Add>>". Similarly, you can move back any curves from the *Curves to insert* listbox into the *Available curves* list box by selecting them and pressing <<Remove. To make selection easier, you can filter out some of the available curves by unchecking any of the *Show* types. When you press OK, all the curves in the *Curves to insert* list box will be inserted into the active diagram page.

DesignSoft

TINA offers another way to select curves for inclusion in the diagram. With the Post processing dialog box open, move the cursor over the schematic. The cursor will change into a pencil shape. Move it over a node or circuit component in the schematic and click. The corresponding curve will be selected in the Available Curves list, and you can add it to the diagram by clicking the Add>> button.

Selected curves can also be deleted with the Delete button. Deleted curves will be removed both from the list boxes and from the dialog window pages!

<u>Lower part:</u>

This part is hidden by default. If you want to create new curves, press the "More >>" button.

Here you can define your own curves using arithmetic operators and mathematical functions. These custom curves must be defined either as a single-line mathematical expression or as a Pascal-like function.

Take special care if you transform the independent variable of curves; for example, writing v(t+1) instead of v(t). The transformed variable expects results from beyond the interval for which the calculation was run. TINA in this case would have no choice but to set the new curve to zero outside of the calculated interval.

While single-line expressions can be entered in the *Line Edit* field, more complex functions should be defined in the *Advanced Edit* field. The *Advanced Edit* checkbox enables/disables the *Advanced Edit* field.

Pressing the ⬚ *"Add to edit area"* button above the *Line Edit* field, you can insert the currently selected curve from the *Available curves* list into the editor field.

Pressing the ⬚ *"Add to edit area"* button to the right of the Built-in Functions drop down list, you can insert the currently selected function from the Built-in Functions list into the editor field.

You can also display a result as a function of another result (e.g., v(t) as a function of i(t)) using the *XY Plot* checkbox. If you check this box, the Line Edit field is divided into an X-part and a Y-part. Bring the curve names into the two fields using the ⬚ *"Add to Edit-area"* button or type the curve names. You can use several curves, arithmetic operations, and functions - but you cannot use *Advanced Edit* in this mode.

ADVANCED TOPICS

DesignSoft

You can use the "*New function name*" edit field to enter your name for the new function (curve).

After you have completed the definition, you can either draw your new curve directly into the diagram window by pressing *Preview*, or you can first store the result by pressing *Create,* and then insert it with the OK button.

Now let's run a transient analysis on the AmpliOpt circuit in the *EXAMPLES* directory. Two curves are drawn in the diagram window: the input, Vin, and the collector voltage, Vout.
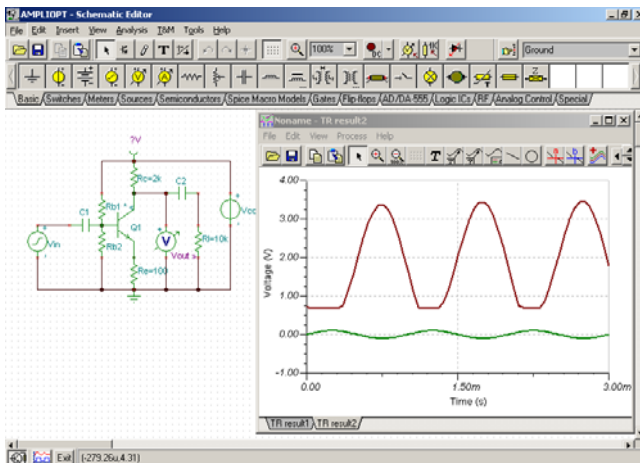
Now press the *Add more curves button* on the toolbar. The post-processing dialog appears. The *available curves* listbox on the left side shows all the curves that have already been calculated.
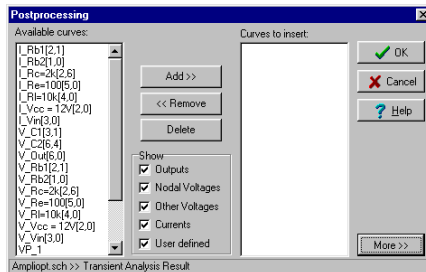
Here, the symbols V_*label*[i,j] and I_*label*[i,j] denote the voltages and currents respectively of the *labeled* components between the node *i* and *j*. The symbol VP_n denotes the nodal voltage of the node *n*.

Now, select V_C2[6,4] (i.e., the voltage of capacitor C2 connected between node 6 and 4) and press the Add>> button. Pressing OK inserts this curve into the current diagram page.

As the next step, let's draw the voltage across the transistor's collector and emitter. Press the *Add more curves* button again and then press the More>> button to reveal the lower part of the diagram. Type VP_6-VP_5 in the *Line edit* field.

Note: another, perhaps easier way, to insert VP_6 or VP_5 into the Edit field is to move the cursor over Q1's collector or emitter, click, and press the [⎯⎯⎯⎯] "Add to Edit-area" button below the listbox.

Note also that if you double click on any node in the circuit with the mouse, the corresponding nodal voltage symbol will be automatically inserted into the *Line edit* field at the current cursor position.

Now change the default name of our function to VCE and press Create.

DesignSoft
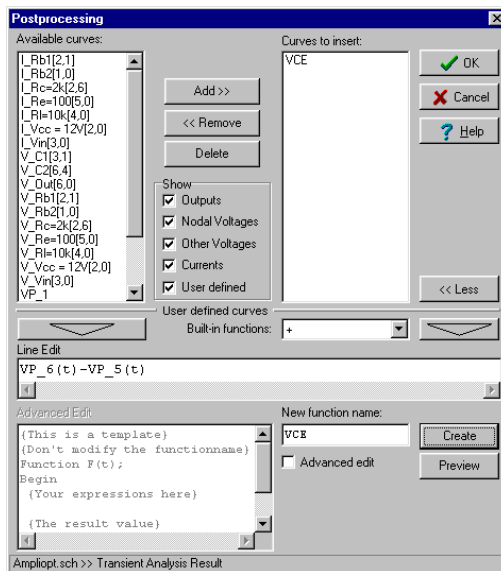
A new function (VCE) is created and inserted automatically into the *Curves to insert* listbox. If you press OK, our new curve will immediately appear on the diagram.
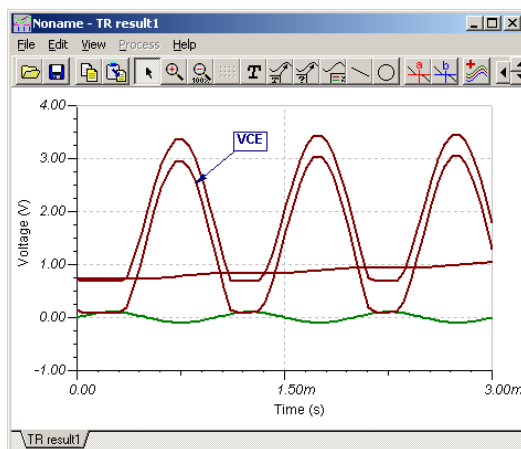
Sometimes the definition of a new curve requires more editing space or other variables. In these cases we can use the post-processing dialog's *Advanced edit* field. To illustrate this, let's draw the output power dissipation of a differential amplifier.

Open the circuit called DiffAmp in the examples directory and run a transient analysis. Press *Add more curves* again and place a check in the *Advanced edit* field. Position the cursor in the *Advanced edit* field below the comment {*Your expressions here*}.

Type :

IR2 := I_R2[14,16](t);

VCE := VP_15(t)-VP_3(t);

Change the line F := 1; {*The result value*} to: F := IR2*VCE;

Enter the name of our new function, OutDissipation, and press Create.
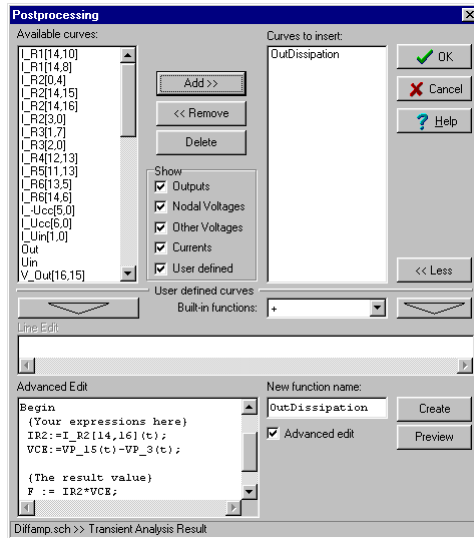
Finally, press OK to draw our new function in the diagram window.

ADVANCED TOPICS

DesignSoft
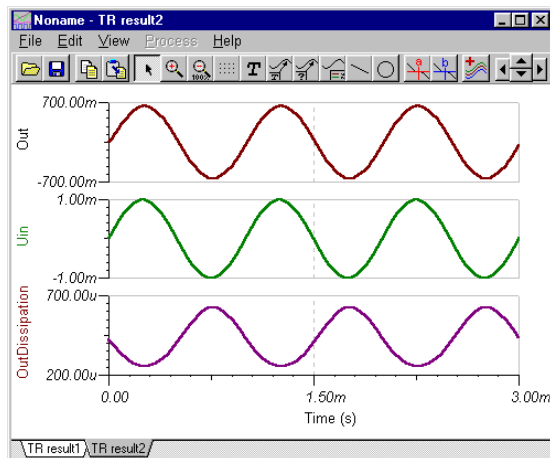
9.9.2   Drawing trajectories (xy-plots) in TINA

In the Post-processor you can also display a signal as a function of another signal, by drawing XY-plots.
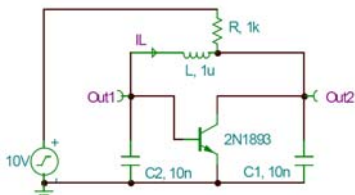
For example, let's draw the voltage of capacitor voltage Out2 as a function of the inductor current IL in the oscillator shown in the figure above.

First load the circuit limitc.tsc from the *EXAMPLES* directory. Run the Analysis.Transient analysis. Now the diagram window shows two curves, the capacitor voltage, out1, and the inductor current, IL.
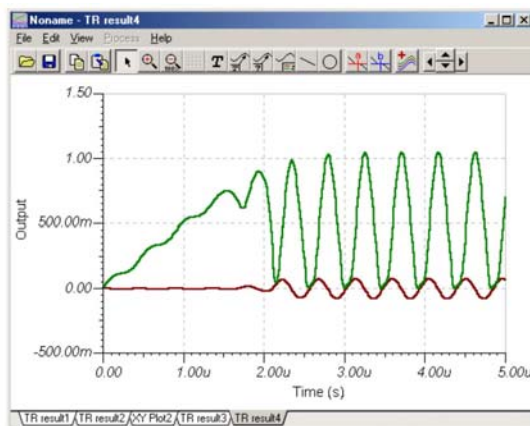
Press the ▣ *Add more curves* button and check the XY-plot checkbox. As you can see in the next figure, the line-edit section is divided into an X-part and a Y-part. Type Out2(t) in the X part and IL(t) in the Y part and press Create and OK. Alternatively, you could have brought the names into the two fields using the ⬚ "Add to Edit-area" button.
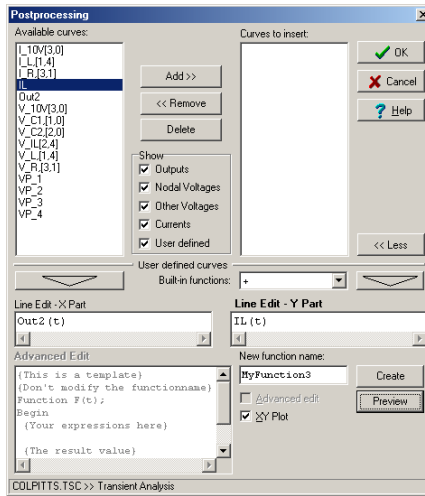
The calculated curve is shown in the next figure. Place a cursor on any curve and move the cursor. As the curve starts from the (0,0) point and converges to a closed trajectory, we see a "limit cycle." Limit cycles are very important in the theory of oscillators.

ADVANCED TOPICS

DesignSoft
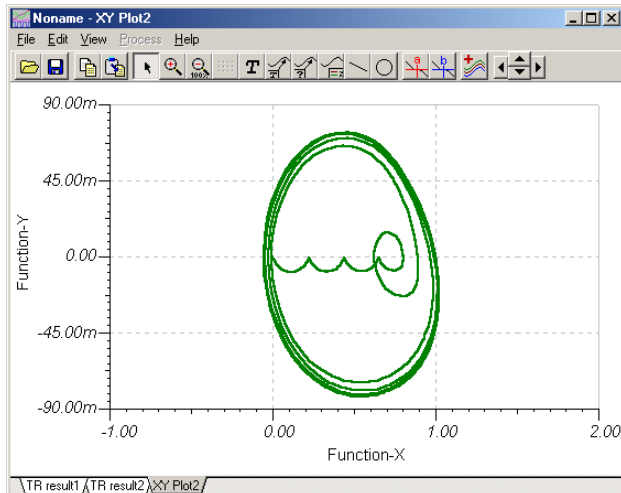
9.10   Design Tool

TINA's Design Tool works with the design equations of your circuit to ensure that the specified inputs result in the specified output response. The tool requires of you a statement of inputs and outputs and the relationships among the component values. The tool offers you a solution engine that you can use to solve repetitively and accurately for various scenarios. The calculated component values are automatically set in place in the companion TINA schematic and you can check the result by simulation.

As an example, this tool can calculate feedback or other resistor and capacitor values of an amplifier in order to achieve a certain gain and bandwidth, and it can calculate component parameters of power supply circuits to meet output voltage and ripple requirements.

The TINA Design Tool promotes good documentation by storing the design procedure together with the circuit.

It is also very useful for semiconductor and other electronics component manufacturers to provide application circuits along with the design procedure.

Let's demonstrate the use of this tool through a simple operational amplifier example.

Open the Invert Gain OPA350 Test Circuit Design.TSC circuit from the Examples\Design Tool folder of TINA. In the TINA Schematic Editor the following circuit will appear:

With the Design Tool we will set Rf and Vref to achieve the specified Gain and DC output voltage.

Now invoke the Design Tool from the Tools menu of TINA. The following dialog will appear.

ADVANCED TOPICS

DesignSoft

Here you can specify the Gain (Vout/Vin), the DC output voltage (Vout_DC) and some other parameters. The simple design procedure will calculate Rg and Vref. The allowed minimum (Min) and maximum (Max) parameters are also shown. To   enable or disable modification of Min and Max select Options from the Settings menu of the Design Tool.

Note that in the Design Tool dialog you can also refer to component parameter names. For example in the Vout_DC line   the maximum value is set as V1-200m, telling that the DC output voltage must be at least by 200mV less than the V1 supply voltage of the IC.

If you just want to Run the design procedure press the button or the F9 key or use the Run command in the menu of the tool. If you run TINA in interactive mode you can immediately see the effect of the changes made by the Design Tool.

To see the design procedure itself, press the More button in the dialog. The code of the design procedure, written in TINA's Interpreter, will appear.

DesignSoft

Note that the code part also shows the calculated parameter values according to the last stored calculation (Now Rf=[10.011k], Vref=[227.0455m]).

Now let's change the Gain input parameter to -1, Vout_DC to 3V and run the procedure by clicking Run in the menu or pressing the green ▶ utton or F9 on the keyboard. In the code part we will see:

A:= 10^(Aol/20) Rg:= Rscale

Rf:=-Gain*Rg*(1 + 1/A)/(1+Gain/A) Rf=[1.0002k]

Vref:= Vout_DC/(1+Rf/Rg) Vref=[1.4998]

and the new values will immediately appear in the schematic editor, drawn in brown color. Press the DC button to display the DC output voltage:

DesignSoft

Now run an AC Transfer analysis, the Bode diagram will appear.

The small frequency Gain is 0dB which complies with the specified Vout/Vin=-1 value.

You can find more complex examples in the Design Tool folder of TINA.

You can make your own design procedure in any TINA circuits and save it together with the circuit itself.

For more information on the use and the controls see the on-line help in TINA by pressing the Help button on the tool.

9.11   Optimization

TINA's optimization tool will determine for you one or more unknown circuit parameters (Control Object) which satisfy a predefined circuit target response (Optimization Target). The target circuit response (voltage, current, impedance, or power) must be "monitored" by a meter–or meters–previously inserted at the desired location(s). The unknown circuit parameter(s) will be determined automatically so that the circuit will produce the target output. Two alternative iterative methods (simple and pattern search) are used to find the optimum.

DesignSoft

Use the commands Analysis.Control Object, Analysis.Optimization Target, or the speed buttons Select Control Object [⟨icon⟩], Select Optimization Target [⟨icon⟩], to set up Optimization mode or to change their settings.

First, place the voltage, current, impedance or power meter(s) onto the schematic to represent the Optimization Targets. Click on the Select Optimization Target speed button [⟨icon⟩] or select the Analysis.Optimization Target menu item. A special cursor [⟨icon⟩] will appear in the schematic editor window, waiting for you to identify the meter. Should you need more meters to represent the optimization targets, repeat the procedure above for the other meters.

There are several target functions you can choose depending upon the analysis mode of the optimization (DC or AC).

For DC optimization problems, you'll need DC goal functions. When you click on a meter with the special target selector [⟨icon⟩] cursor, a dialog box appears showing several goal functions:
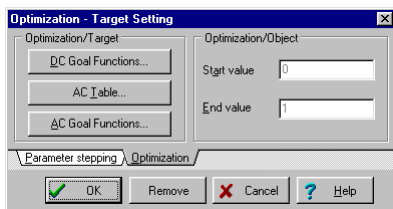
Click on the DC Goal Functions… button. Another dialog box appears which allows you to set a DC goal function:
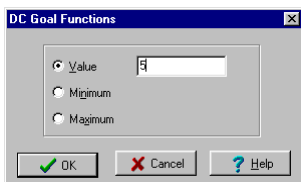
To optimize to a target Value, enter the desired value. To optimize to a Maximum or Minimum, select the appropriate option.

For AC optimization problems, you'll need AC goal functions. With a meter selected, click on the AC Table… or on the AC Goal Functions… button. The AC Table supports a special AC goal

function defined by a series of frequency and amplitude values. Using these frequency-amplitude pairs, an AC transfer function can be defined.
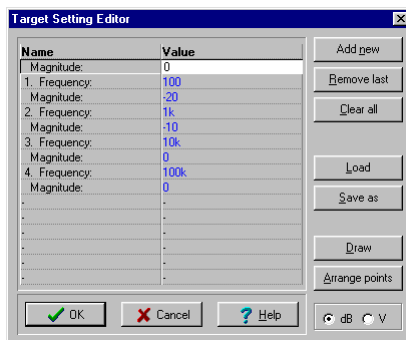
You can access other AC goal functions by pressing the AC Goal Functions… button. Here you'll find Center Frequency, High Pass, Band Pass, and Low Pass target functions, as well as Minimum and Maximum target functions. You can combine these functions, effectively setting more than one criterion.
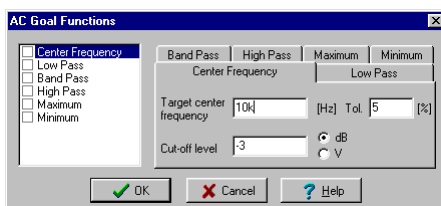
Now that you have established the target details, you must identify the control object(s) and their parameter(s) to be determined by optimization. To select a component, click on the Analysis.Control Object menu item or on the Select Control Object speed button.

A special cursor will appear in the schematic editor window, waiting for you to identify the component to which the optimized parameter belongs. If you want TINA to optimize using more than one component, repeat the procedure for the other components.

When you point at the component and click, TINA will display all of its associated parameters in a dialog window.

ADVANCED TOPICS
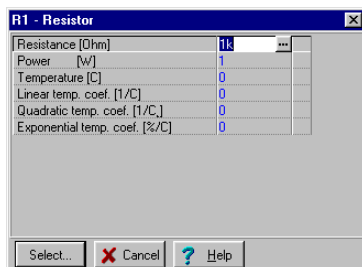
Highlight the parameter to be determined by the optimization process, press the Select button, and enter the Start and End values for the variation of this component's value.
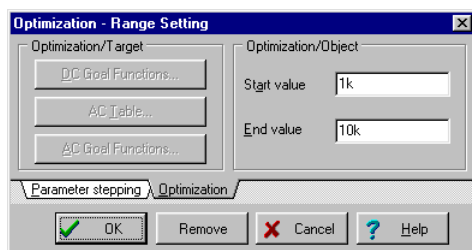
If the desired parameter belongs to a catalog component, or if it is an excitation attribute, you must first select the semiconductor or waveform type so that you can access the appropriate data sheet.

The selected parameter will be varied in the range between the Start and End values while TINA searches for the optimum result at the target. If no optimum is found, then the range will have to be enlarged. If there is more than one optimum case in the range, TINA will only find one of them. If you suspect that this is the case, try using multiple, narrower ranges, each of which has one or fewer optimums.

You can use the Remove button to deselect a previously assigned control object or optimization target.

Optimization is useful not only in the design of electronic circuits, but in teaching, where you can use it to construct examples and problems. Let us see some examples.

Two examples

Example 1: Differential Amplifier (DC Optimization)

This example illustrates how to use the DC Optimization to set the operating point of a differential amplifier.

1. **Specification:** First load the file named DIFFAMP.TSC from TINA's EXAMPLES/OPTI subdirectory and run Analysis/DC Analysis/Calculate Nodal Voltages to see the original VF1, VF2 collector voltages (3.78V). The goal of the optimization is to change the values of R1A, R1B in order to find those values which meet the specification–2.5V–for both collector voltages.

2. **Set up the optimization:** Before you start the optimization, you have to select those meters which will be monitored during the iteration and which represent the goal of the optimization. In this example, the meters are VF1 and VF2, and the goal establish 2.5V at the collector of the transistors. You also must select those parameters which will be determined by the optimization. In this example, they are the resistances of R1A and R1B.

a) **Target:** Use the command Analysis.Optimization Target or the Select Optimization Target speed button 🔧 and click on the first voltage pin (VF1). A dialog box appears. When you press the DC Goal Functions… button another dialog appears which allows you to set the goal function for this meter. Select the Value option and set the value to 2.5. (Note that you mustn't use units here.) Then press OK twice. Repeat this procedure for the other voltage pin, VF2.

b) **Control object:** Use the command Analysis.Control Object or the Select Control Object speed button 🔧 and click on the first control object (R1A). In the property dialog that appears, click on the first line (the resistance parameter). Press the Select button and another dialog appears which allows you to define the Start and End values for the variation of this parameter. Enter 1k and 100k respectively and press OK. Then repeat this procedure for the other resistor, R1B.

3. **Start the optimization:** From the Analysis menu choose Optimization, then DC Optimization… Accept the settings in the dialog that appears and press OK. Now TINA will find new R1A, R1B values to achieve the desired 2.5V collector voltages. After a short iteration the result–the new resistor values–will be shown in a dialog.

ADVANCED TOPICS

4. **Checking the result:** TINA will retain the optimized resistor values. Run Analysis/DC Analysis/Calculate Nodal Voltages to check the new collector voltages with these values. Note that they are now exactly 2.5V as we previously defined in the specification, verifying that the optimization was successful.

Example 2: Band pass Filter (AC Optimization)

This example illustrates how to use the AC Optimization to obtain a band pass filter which meets transfer function goals.

1. **Specification:** First load the file called BANDPASS.TSC from TINA's EXAMPLES/OPTI subdirectory and run Analysis/AC Analysis/AC Transfer Characteristic… to see the original transfer function. The diagram shows that the maximum of the curve is at about 10kHz, while the peak amplitude of the curve is -10dB. The goal of the optimization is to set the peak amplitude to -7dB and the frequency at which the curve reaches its maximum to 14kHz.

2. **Set up the optimization:** To do an optimization, you have to select those meters which will be monitored during the iteration and which represent the goal of the optimization. In this example, this is the meter "Out." You must also select those parameters which will be varied during the optimization. In this example, these parameters are the resistance of R1 and the capacitance of C1.

a) **Target:** Use the command Analysis.Optimization Target or the Select Optimization Target speed button 🔧 and click on the voltmeter (Out). A dialog box appears. When you press the AC Goal Functions… button another dialog appears which allows you to set the goal function for this meter. The specification contains two conditions. The first defines the new maximum point of the curve. To set this value, click on the Center Frequency tab, then enter 14k in the Target center frequency edit field. The other condition is the new maximum, -7dB. Select the Maximum tab and enter -7 in the Maximum edit field. The last step is to activate the current goal functions which are the combination of two functions. On the left side of the dialog, select both the Center Frequency and the Maximum function by clicking on the checkbox next to them. Finally press OK twice.

b) **Control object:** Use the command Analysis.Control Object or the Select Control Object speed button and click on the first control object (R1). In the resistor property dialog, click on the first line, the resistance parameter. Press the Select button to reach another dialog where you can define the Start and End values for the variation of this parameter. Enter 1k and 10k respectively and press OK. Repeat this procedure for the capacitor (C1), and specify 1n and 10n for the Start and End values respectively.

3. **Start the optimization:** From the Analysis menu, choose Optimization then AC Optimization (Transfer)… Accept the settings in the dialog and press OK. Now TINA will find new R1 and C1 values that will satisfy the specification for the transfer function. After a short iteration the result–the new resistor and capacitor values–will be shown in a dialog box.

4. **Checking the result:** Run Analysis/AC Analysis/AC Transfer Characteristic… to check the new transfer function. The maximum now occurs at 14kHz, with a level at the maximum of -7dB. The optimization was successful.

9.12   Design Tool vs. Optimization in TINA

There are cases when writing a design procedure is not obvious or needs iteration or simply we do not have the time to implement it. In this case you may use the Optimization tool in TINA to determine the required parameter numerically in order to meet predefined circuit responses: voltage, current, power, gain, etc.

Note: Optimization for multiple parameters and target values is available in the Industrial version of TINA only.

Generally speaking, although optimization is a very powerful tool, it is better to use a design procedure, if available because numerical optimization might need significant calculation time and it does not guaranty physically realistic results. But it is a very good tool to refine the results provided by a design procedure or tune already working circuits.

ADVANCED TOPICS

DesignSoft

9.13   Fourier Analysis

9.13.1 Introduction

In Tina there are many advanced analysis features that provide powerful professional tools to study and evaluate electronic circuits both numerically and graphically. One of them is called spectral or Fourier analysis. In this booklet we will demonstrate Tina's implementation of Fourier analysis which covers two different methods: Fourier series and Fourier spectrum. We will especially emphasize applications that are not described in detail in the manual of Tina. In the following we will shortly summarize the theory behind Fourier analysis, and then present examples on the topic. All the examples are available on floppy disk, as well.

In Tina Fourier analysis must be always performed in conjunction with the transient analysis. This means, that first you always have to carry out a transient analysis on the circuit. When the diagram window appears you have to select the curve you want to process using the cursor and from the *Process* menu choose *Fourier series* or *Fourier spectrum*. In the first part of this booklet we will investigate Fourier series and later on Fourier spectrum.

9.13.2 Fourier Series

One type of spectral analysis is called Fourier or harmonic decomposition. It's based on the theory, that any periodic function can be expressed as the sum, or series of sinusoidal functions. If a periodic function is expressed this way, each component in the series must be periodic over the same interval as the original function. The components are integer multiples or harmonics of the original function's fundamental (base) frequency. This decomposition is described by the following formulas:

$$f(t) = \sum_{k=0}^{\infty} (A_k \cos k\omega t + B_k \sin k\omega t), \text{ or in complex form } f(t) = \sum_{k=-\infty}^{\infty} C_k e^{jk\omega t},$$

where A_k, B_k, and C_k are the Fourier coefficients.

Let us see this decomposition in case of some practical waveforms:

Triangle wave
Square wave

$$f(t) = \frac{8}{\pi^2}\left[\sin \omega t - \tfrac{1}{3^2}\sin 3\omega t + \tfrac{1}{5^2}\sin 5\omega t - ...\right]$$

$$f(t) = \frac{4}{\pi}\left[\sin \omega t + \tfrac{1}{3}\sin 3\omega t + \tfrac{1}{5}\sin 5\omega t + ...\right]$$

Let us demonstrate the above theorem with Tina in case of a square wave of f=1kHz frequency and 1V amplitude up to three sine waves. The easiest way is to put together a simple circuit (fourex1.sch). The circuit consists of three sine wave generators connected in series and the reference square wave generator. The frequencies of the sine wave generators are $f_1 = 1kHz$, $f_2 = 3f_1 = 3kHz$ and $f_3 = 5f_1 = 5kHz$, however the amplitudes are $A_1 = \frac{4}{\pi} = 1.2732$, $A_2 = \frac{1}{3}A_1 = 0.4244$ and $A_3 = \frac{1}{5}A_1 = 0.2546$ according to the above formula.
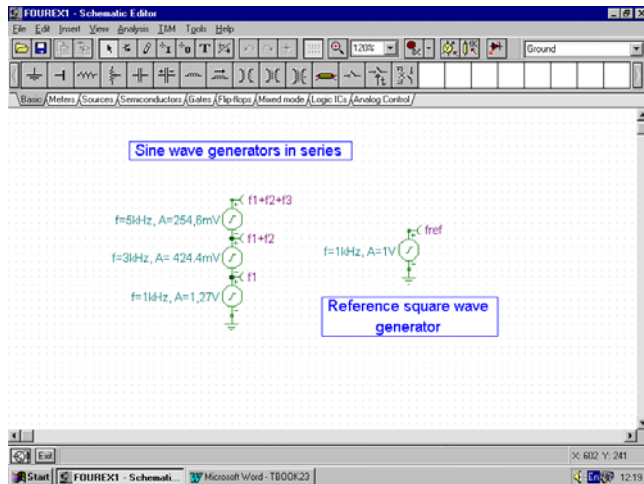
The result of the transient analysis can be seen on the following figure.
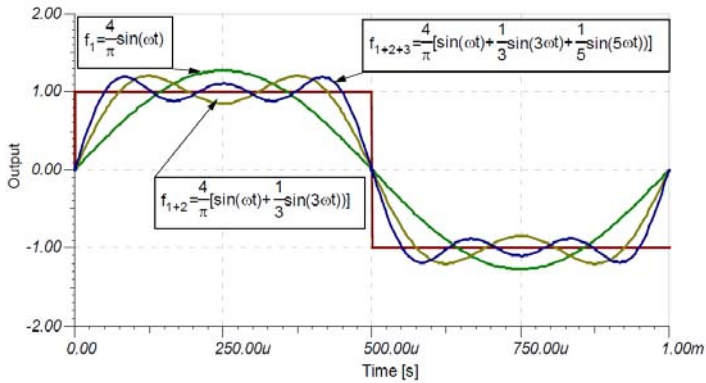
Note that the above figure was entirely made by Tina using its Equation Editor in the Diagram Window. For example the X is described in the following way in the Equation Editor's text mode:

\i(f,1)=\f(4,\s(p))sin(\s(w)t) as it can be seen on the next figure.
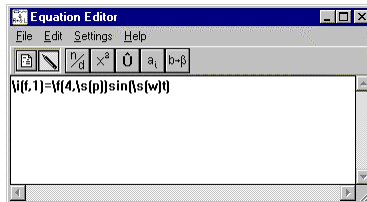
If you press the View button 🗈 you can preview the formula in the Equation Editor. Press the Edit button ✎ to return into the text mode if you want to make further changes. Finally press the Copy button 🗎 to place the formula in the Diagram Window.

ADVANCED TOPICS

DesignSoft
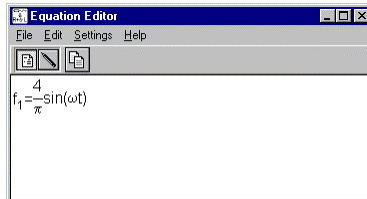
Obviously if you connected more than 3 generators in series, you would have achieved a better match. There is an easy way to demonstrate this statement. Using Tina's Interpreter you can easily write a function that calculates the sum of any number of sine waves (fourex1.ipr).

```
Function Square(t, n);
{Sum of n sine waves, n: number of sine waves, t: time}
Begin
   f := 1k;
   w := 2 * pi * f;
   x := 0;
   For i := 0 To n - 1 Do
      x := x + 1 / (2 * i + 1) * sin((2 * i + 1) * w * t);
   Square := 4 / pi * x;
End;

{Sum of 20 sine waves}
Draw(Square(Time, 20), Square);
```

To run the above example invoke the Interpreter from the Tools menu, type in the above text or read in fourex1.ipr by selecting the Open command from the File menu. If you press the Run ▶ button after a short calculation the following function appears:
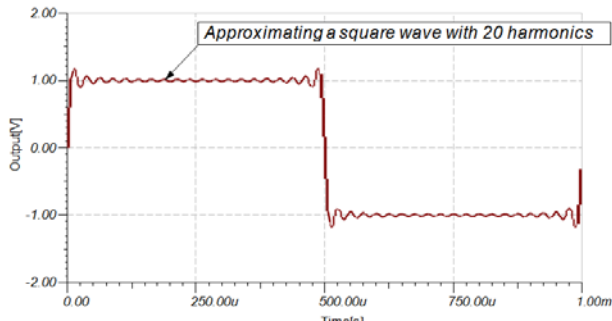
The more sine waves you would sum up the better match you would get. Note however the small peaks at the edges of the square wave. This is called Gibbs phenomenon. According to the mathematical theory the Fourier series can not approximate properly

non- continuous functions in the point of discontinuity. In case of a square wave there will always be 18% overrun at the edges at any number of approximating sine waves. Let us demonstrate this with 100 harmonics; simple change 20 to 100 in the Draw command of the above example (fourex1.ipr)
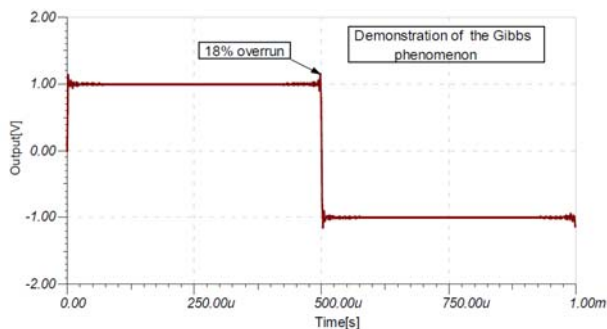
However there are methods which improve the convergence of the Fourier series near the discontinuity. The essence of these techniques is to modify the Fourier coefficients by multiplying each coefficient by a weight factor. This method is also called window function technique. Two Hungarian mathematicians - Fejér, and Lánczos - invented such window functions that are commonly used in practice.

In practice however there are no discontinuous waveforms though to the rise/fall time could be very short in comparison to the period time. This means, that the Fourier series will always converge. Let us demonstrate this with the approximation of the triangular waveform with just 3 harmonics using the above circuit. Change the amplitude of the sinusoidal generators to $A_1 = \dfrac{8}{\pi^2} = 0.8106$, $A_2 = -\dfrac{1}{3^2} A_1 = 0.0901$ and $A_3 = \dfrac{1}{5^2} A_1 = 0.0324$. Change the waveform of the reference generator to triangle wave (fourex12.sch).

DesignSoft
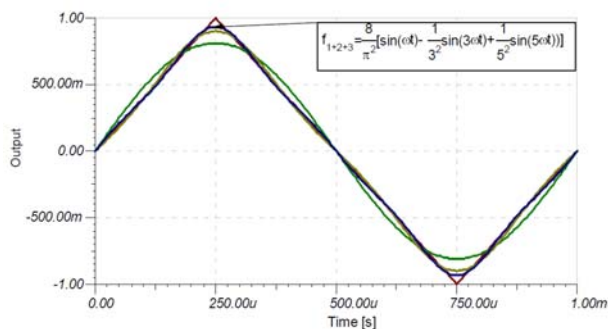
In practice of course you use the Fourier series option of TINA in opposite direction. You have a periodic function and you want to know the Fourier components. To get them in Tina first you have to run a transient analysis and then in the appearing Diagram window select a curve. Then in the diagram window you have to select Fourier series from the Process menu. After selecting this menu item Tina After selecting this menu item Tina performs a harmonic decomposition, calculating the Fourier coefficients for the sinusoidal components of any voltage or current. After you select Fourier series a dialog box appears, where you have to set the sampling start time, the base frequency, the number of samples, the number of harmonics, and the format.

Example: Let us determine the Fourier decomposition of the previous square wave (f=1kHz, A=1V). We will compare the theoretical decomposition with the result calculated by TINA.

The Fourier series of a square wave function is

$$f(t) = \frac{4}{\pi}\left[\sin \omega t + \tfrac{1}{3}\sin 3\omega t + \tfrac{1}{5}\sin 5\omega t + ...\right].$$

Because square wave function is an odd function $/f(-x) = -f(x)/$, all the coefficients $A_k = 0$, and $A_0 = 0$, too. Comparing the above function with the definition of the Fourier series it's quite clear that:

$$B_1 = \frac{4}{\pi} = 1.27, \quad B_2 = 0, \quad B_3 = \frac{1}{3}B_1 = 0.42441$$

$$B_4 = 0, \quad B_5 = \frac{1}{5}B_1 = 0.25464$$

In TINA we get approximately the same result. However due to numerical inaccuracy we don't get exactly 0 for the theoretically 0 coefficients. If you examine these values more carefully you'll see, that they are not significant because they are $10^{-6} - 10^{-9}$ times less than the fundamental value.

Load this example (fourex2.sch). Run transient analysis then select the appearing square wave and choose Fourier series from the *Process* menu. In the dialog box change the default format from complex representation $/ \quad C * \exp(k\omega t + \phi)/$ to real representation

$/ A * \cos(k\omega t) + B * \sin(k\omega t) /.$

If you have set all the parameters, press *Calculate*. At the bottom of the dialog box the Fourier coefficients appear. Tina allows you to use these values in other programs. Every time you press *Calculate* the coefficients are placed not only at the bottom of the dialog box but on the clipboard, as well. You can import this table of Fourier components in other programs, for example Word or Excel to process.

Harmonics	Amplitude (A)	Amplitude (B)
0.	0	0
1.	976.56u	1.27
2.	-1.15n	-749.01n
3.	976.56u	424.41m
4.	-4.6n	-1.5u
5.	976.56u	254.64m

It's also possible to evaluate the above result graphically. You only need to press *Draw*, then Tina generates you a diagram with the Fourier coefficients.

DesignSoft
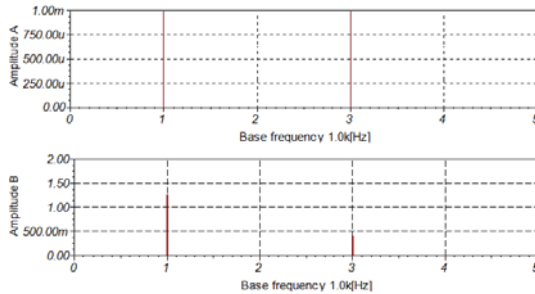
You can also get complex Fourier components with Tina. To do this select the complex representation in the Fourier series dialog box and press *Calculate* again. On the following three figures you'll see the harmonic decomposition of the same sine wave (f=1kHz, A=1V) but this time using the complex representation. On the first figure the dialog box can be seen, on the next the Fourier coefficients as they are placed on the clipboard, and on the last the graphical representation of the result.
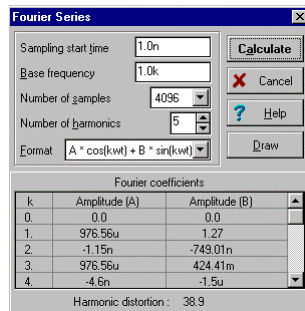
Harmonics	Amplitude (A)	Phase (ø)
0.	0	0
1.	636.62m	-89.96
2.	374.51n	90.09
3.	212.21m	-89.87
4.	749.01n	90.18
5.	127.32m	-89.78

DesignSoft
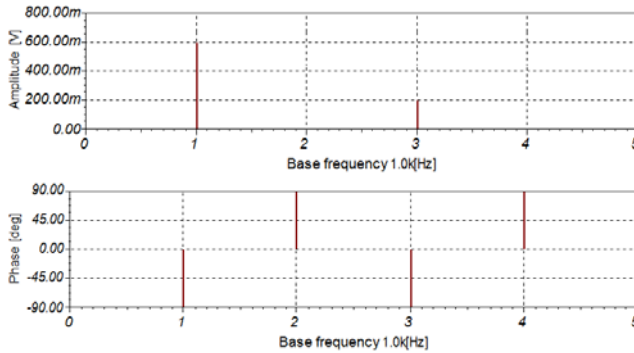
To compare this result with the previous one we have to perform a simple mathematical transformation. Using the Euler-formula:

$$\frac{A_k + jB_k}{2} = C_{1k}, \text{ and } \frac{A_k \mid jB_k}{2} = C_k \text{ or vice versa } C_k + C_{1k} = A_k, \text{ or } C_{1k} \mid C_k = B_k.$$

From   this   we   get,   that

$$C_1 = \frac{A_1}{2} = 0.63662, \quad C_2 = 0, \quad C_3 = \frac{A_3}{2} = 0.21221, \quad C_4 = 0 \; C_5 = \frac{A_5}{2} = 0.12732$$

Again it is important to mention, that due to numerical inaccuracy we don't get exactly 0 for the theoretically 0 coefficients. It's especially true in case of the phase values. As harmonic magnitudes become small it becomes difficult to determine phase. Again, this is a numerical problem only.

9.13.3 Fourier Spectrum

Tina provides another type of spectral analysis called Fourier transform. Fourier transform can also be thought of as an extension of the Fourier series. This means the Fourier transform converts a function of time to a function of frequency, and vice versa. The physical interpretation of Fourier transform is the conversion of a time-domain signal to the AC steady-state frequency spectrum.

$$F(j\omega) \equiv F f(t) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

However the inverse Fourier transform converts the AC steady-state signal into the time-domain.

$$f(t) = F^{-1}F(j\omega) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} F(j\omega)e^{j\omega t} d\omega$$

The Fourier transform in Tina is a discrete Fourier transform (DFT), where the Fourier integral has been replaced by a nearly equivalent summation formula applied to evenly spaced samples of the signal. To speed up the process the transform is accomplished by a special technique, called fast Fourier transform (FFT). The FFT gets its data from a set of samples with the appropriate number of data points.

Sometimes it's useful to define the real spectrums (amplitude-densities):

$F(j\omega) = \dfrac{A(\omega) - jB(\omega)}{2}$, where $A(\omega) = 2\,\mathrm{Re}\,F(j\omega)$ and

$B(\omega) = -2\,\mathrm{Im}\,F(j\omega)$

Note, that Tina calculates $F(j\omega)$ and displays in the $0 < \omega < \infty$ region only. For the $-\infty < \omega < 0$ region applies $F(j\omega) = F(j\omega)^*$, therefor the amplitude is the same and the phase has opposite sign.

$A(\omega),\ B(\omega)$ can also be calculated directly from the time function.

$A(\omega) = 2\int\limits_{-\infty}^{\infty} f(t)\cos\omega t\, dt$ and $B(\omega) = 2\int\limits_{-\infty}^{\infty} f(t)\sin\omega t\, dt$

$A(\omega) = A(-\omega),\ B(-\omega) = -B(\omega)$ thus $A(\omega)$ is an even, $B(\omega)$ is an odd function.

Example: Let's determine the complex spectrum of a pulse (amplitude 1, width T=1n; fourex3.sch) and compare it with the theoretical result. First carry out a transient analysis then select the pulse wave and choose Fourier spectrum from the *Process* menu.
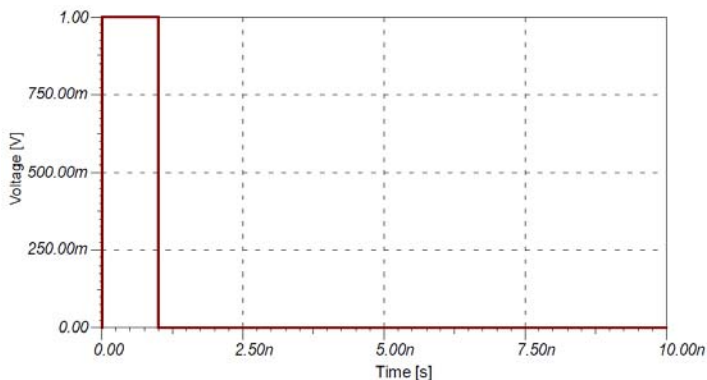
After you select Fourier spectrum a dialog box appears, where you have to set the sampling start and end time, the minimum and maximum frequency and the number of samples. You can also choose the diagrams you want to display.
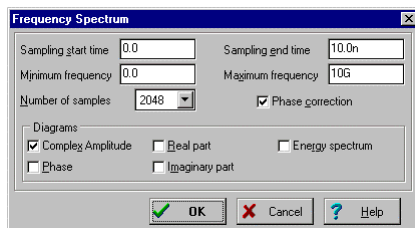
If you press OK Tina performs discrete Fourier transform. The complex amplitude spectrum can be seen on the following figure:

DesignSoft
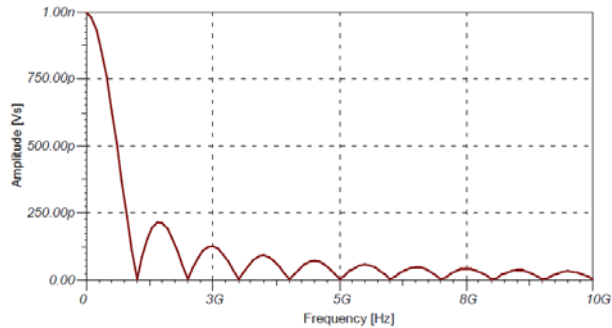
This function is described by the well-known formula:

$$F(j\omega) = \int_0^T 1*e^{-j\omega t}\,dt = \frac{1-e^{-j\omega T}}{j\omega} = e^{-j\omega T/2}\,\frac{e^{j\omega T/2}-e^{-j\omega T/2}}{j\omega} = \frac{2}{\omega}\sin\frac{\omega T}{2}\,e^{-j\omega T/2}\ ,$$

therefore the amplitude spectrum is: $\left|F(j\omega)\right| = \frac{2}{\omega}\left|\sin\frac{\omega T}{2}\right|,\ 0<\omega<\infty$

To check our previous result now we will recalculate the amplitude spectrum using the above formula in Tina's Interpreter (fourex2.ipr). The following function calculates the amplitude spectrum.

```
Function PulseAmplSpe(f);
Begin
    w := 2 * pi * f;
    T := 1n;
    PulseAmplSpe := 2 / w * Abs(Sin(w * T / 2));
End;

Draw(PulseAmplSpe(Frequency), AmplSpe);
```

To try out the above example invoke the Interpreter from the Tools menu, type in the above text or read in fourex2.ipr by selecting the Open command from the File menu. If you press the Run ▶ button after a short time the following function appears:

You can easily check, that the above spectrum provided by the closed formula and the spectrum provided by Tina's Fourier Spectrum command are exactly the same. It's rather easy to compare these curves using the clipboard. First select the curve which was generated by the Interpreter, then copy this curve onto the clipboard then in the diagram window go back to the previous page and paste the curve back. The curves will be identical, so we do not show here both.

Example: Transform of a sine wave (fourex4.sch)

The Fourier transform of a sine wave results in a Dirac-delta function, that is an infinitely high spike of zero width at the frequency of a sine wave. We can not reproduce that with DFT. We can actually examine sine waves of finite length. The longer the sine wave is, the taller and narrower the spectrum becomes.

Let us first examine a 3ms long sine wave with 1kHz frequency.
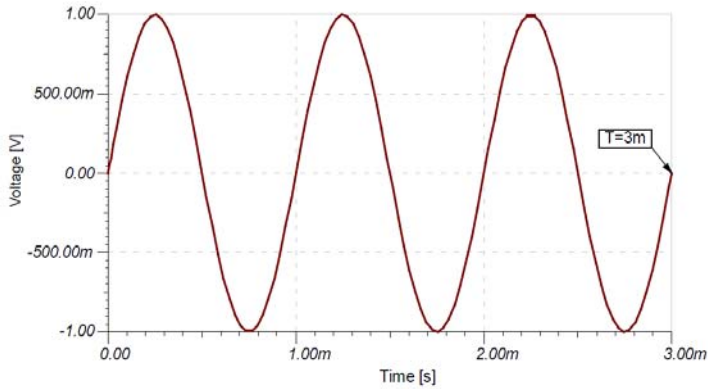
If you run Fourier spectrum on this curve you will get the following result.
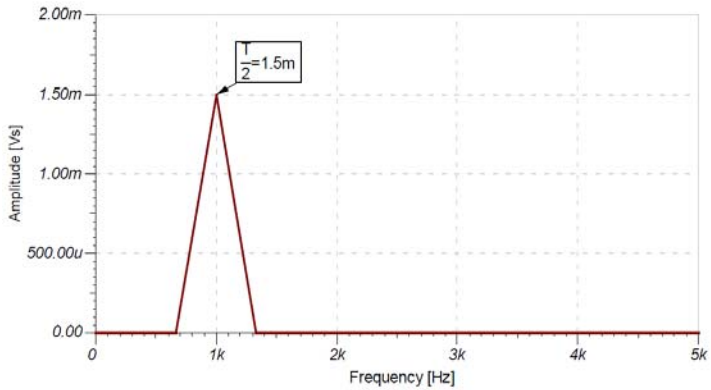
Although the peak value of this spectrum is correct the shape of the spectrum is quite rough. To get a better spectrum change the simulation time to 21ms and carry out the analysis again. At FFT the frequency resolution is inversely proportional to the duration of the signal.

DesignSoft

ADVANCED TOPICS
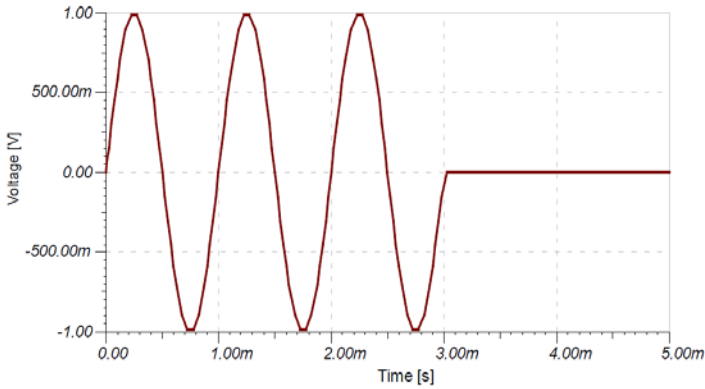
Note that the actual simulation time in this example was 21ms to get a better spectrum. (Not shown on the figure.)
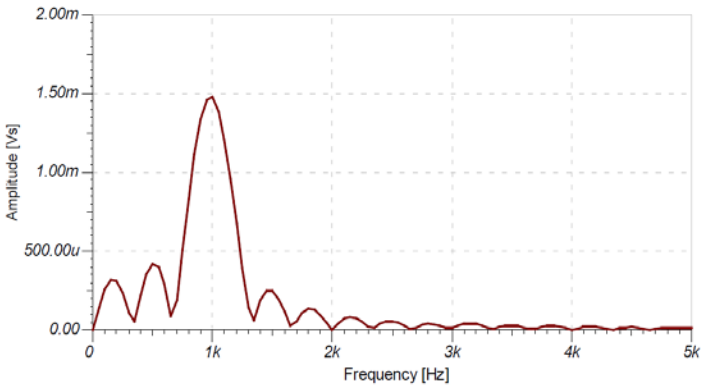
Let us check this result using the theory of Fourier transform:

$$F\{d(t)\sin\omega_0 t\}, \text{ where } d(t) = \varepsilon(t) - \varepsilon(t-T) = \begin{cases} 0, & \text{if } t < 0 \text{ or } t > T \\ 1, & \text{if } 0 \le t \le T \end{cases}$$

$$F(j\omega) = \int_{-\infty}^{\infty} d_T(t)\sin\omega_0 t e^{-j\omega t} dt = \int_{-\infty}^{\infty} d_T(t)\frac{e^{j\omega_0 t} - e^{-j\omega_0 t}}{2j} e^{-j\omega t} dt = \frac{1}{2j}\left\{\int_0^T e^{-j(\omega-\omega_0)t} dt - \int_0^T e^{-j(\omega+\omega_0)t} dt\right\} =$$

ADVANCED TOPICS

$$e^{-j(\omega-\omega_0)\frac{T}{2}}\frac{\sin(\omega-\omega_0)\dfrac{T}{2}}{j(\omega-\omega_0)} - e^{-j(\omega+\omega_0)\frac{T}{2}}\frac{\sin(\omega+\omega_0)\dfrac{T}{2}}{j(\omega+\omega_0)}$$

Note, that the main amplitude of the spectrum at $\omega = \omega_0$ and at $\omega = -\omega_0$ is $T/2$.

Let us draw this function with Tina's Interpreter (fourex3.ipr). In this example we will use a simplified expression, because this part is significant only regarding the amplitude of the spectrum. The following simplified function calculates the amplitude spectrum.

```
Function SinusAmplSpe(f);
Begin
  w := 2 * pi * f;
  w0 := 2 * pi * 1k;
  T := 3m;
  SinusAmplSpe := Abs(Sin((w - w0) * T / 2) / (w - w0) –
                       Sin((w + w0) * T / 2) / (w + w0));
End;

Draw(SinusAmplSpe(Frequency), AmplSpe)
```

To try out the above example invoke the Interpreter from the Tools menu, type in the above text or read in fourex3.ipr by selecting the Open command from the File menu. If you press the Run ▶ button after a short time the following function appears:

ADVANCED TOPICS

DesignSoft

You can easily check, that the above spectrum provided by the closed formula and the spectrum provided by Tina's Fourier Spectrum command are exactly the same. It's rather easy to compare these curves using the clipboard. First select the curve which was generated by the Interpreter, then copy this curve onto the clipboard then in the diagram window go back to the previous page and paste the curve back. The curves will be identical, so we do not show here both.

Example: Amplitude modulation (am.sch).

Let us see the spectrum derived by amplitude modulation. In our example the amplitude modulated signal is described by the following formula: $\sin(\omega_0 t) * (1 + m\cos(\omega_m t))$, where

$\omega_0 = 1MHz$, $\omega_m = 100kHz$, $m = 0.5$
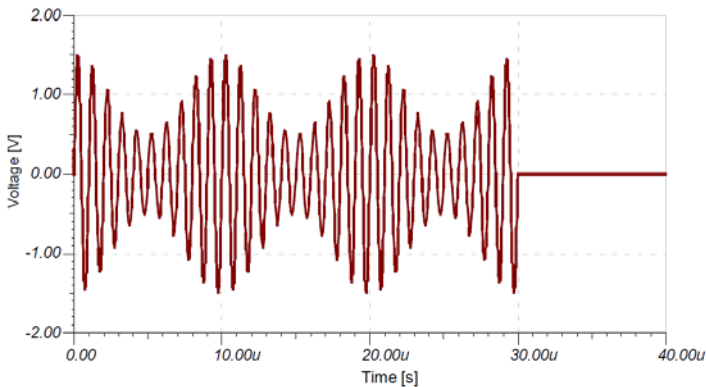
Let's calculate the Fourier spectrum of the modulated waveform:

$$\sin(\omega_0 t) * (1 + m\cos(\omega_m t)) = \sin\omega_0 t + \frac{m}{2}\sin(\omega_0 + \omega_m)t + \frac{m}{2}\sin(\omega_0 - \omega_m)t.$$

In our case:

$$\sin(1e6t) * (1 + 0.5\cos(1e5t)) = \sin 1e6t + 0.25\sin(1.1e6)t + 0.25\sin(0.9e6)t$$

From this formula it's clear, that the spectrum of the modulated waveform is the sum of the spectrum of three sinusodial signals. First check our result using Fourier series. Select Fourier Series from

the Process menu and set the Format field to $A\cos(k\omega t) + B\sin(k\omega t)$.
Then press Calculate.
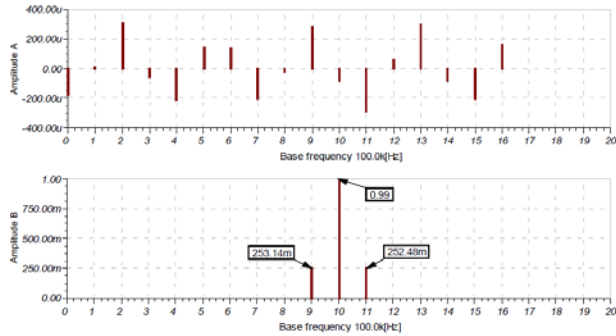
The cosinusodial coefficients (A) are practiacally zero and for the
sinusodial coefficients (B) we get the theoretical result.

On the next figure the Fourier spectrum of the same signal can be
seen.
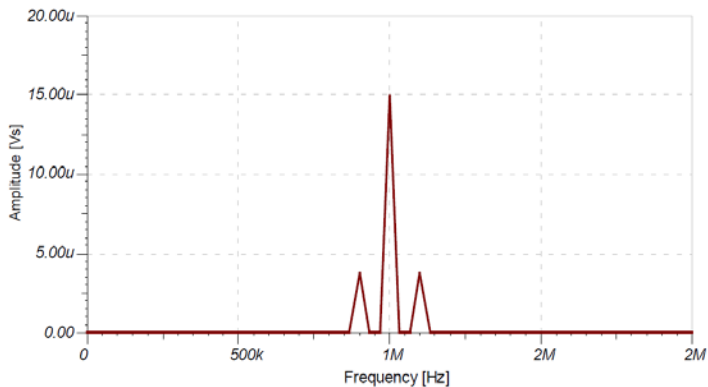
You can see, that the amplitude of the smaller spikes is the quarter
of the amplitude of the peak at the base frequency, which is 15u=T/
2, where T=30u is the length of the modulated wave.

And finally on the last figure you can see the spectrum in case of
210us simulation time.
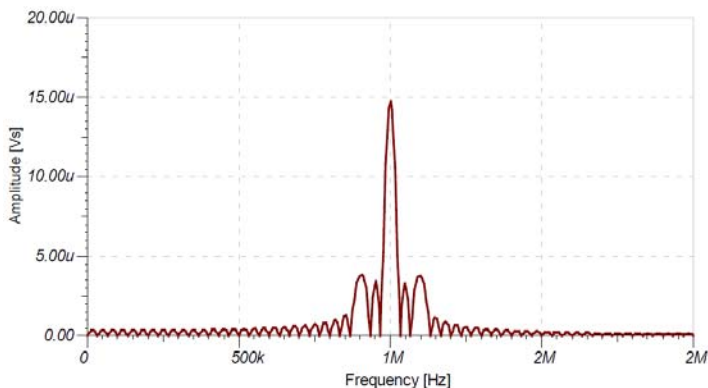
ADVANCED TOPICS

DesignSoft

The Fourier spectrum and the Fourier series dialog box can also be obtained directly from the Analysis.Fourier Analysis menu. This way you do not need to calculate the transient function manually–TINA will automatically do it before generating the Fourier series or spectrum.

9.13.4 Harmonic Distruction

In Tina you can also calculate the harmonic distortion.

DesignSoft
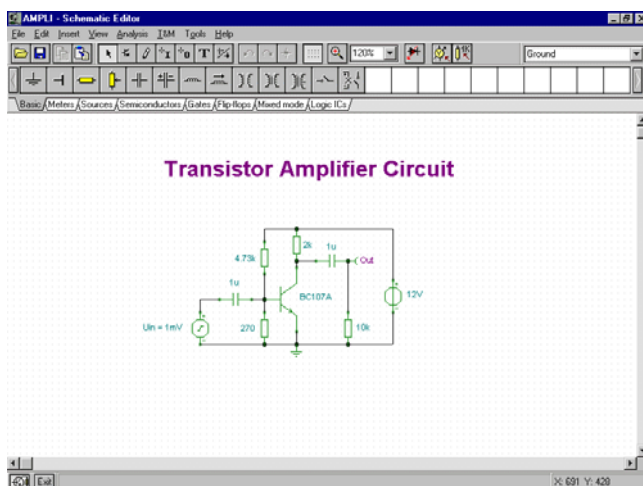
To illustrate this, run the transient analysis with the above circuit (ampli.sch) then the following figure appears:
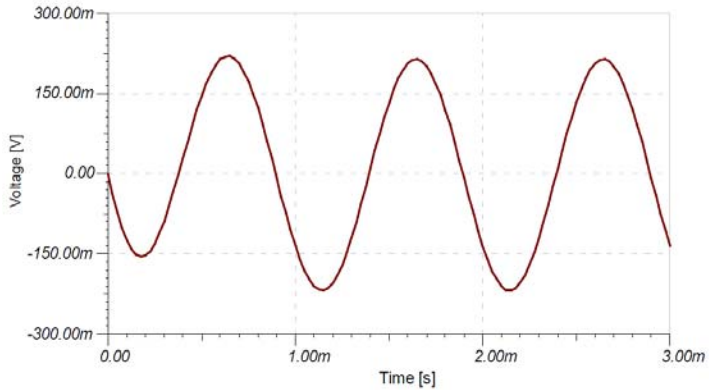
Select the output curve then press the right button of the mouse and select Fourier Series from the popup menu: the dialog box of the Fourier series will appear. Set **Sampling start time** to 1ms and the **Number of samples** to 4096. Note that for best accuracy, it is very important to set the starting time for the Fourier Series analysis to after the initial transient has died away. Now press **Calculate**. Tina calculates you the Fourier coefficients and the harmonic distortion, which is defined to be the ratio of the root-mean-square (RMS) sum of the magnitude of the harmonics to the magnitude of the fundamental, or as a percentage: X, called percent total harmonic distortion (%THD).

9.14   Using IBIS Models in TINA

IBIS (Input/Output Buffer Information Specification) is a method to provide modeling information about the input/output buffers of integrated circuits. The good thing about IBIS models is that they are often available even for devices where complete device models are not available from manufacturers for any reason (e.g., complexity,proprietary information protection, etc.). One of *the most popular uses of IBIS models is Signal Integrity Analysis*, including impedance matching and more. TINA currently supports the most widely used IBIS 4.2 version.

In TINA, you can convert IBIS models to TINA Spice macros and then use them in any circuits in TINA. You can also complete simplified digital device models—e.g., MCUs with IBIS models—to better describe their analog behavior.

In the following, we will show the use of IBIS models through an example of fixing signal integrity between a Texas Instrument TMS320C6748DSP and an ADS1259 delta-sigma ADC.

Select *File/Import/IBIS File (*.ibs)*, select *c6748zce.ibs* from *<TINA directory>\Examples\IBIS*. The following dialog will be displayed. In this dialog, you can select the model to import.

Now select *SPI1CLK_GP213*signal, PBFZP18LL_X50_PI_3P3 model (cell operated at 3.3Vwithout pullup or pulldown),and *Typ* value set. Press OK. The IBIS model is automatically converted to a Spice macro.

DesignSoft

U1 I/O
POLARITY=0 C_COMP=1.94P R_PKG=469.26M L_PKG=4.17N
C_PKG=711.45F DVR=1.16 DT_R=315.35P DVF=1.21
DT_F=385.08P VCC=3.3 ENABLE=0 VINL=0.8 VINH=2
RP_LPKG=100

SPI1CLK_GP213is the master configuration serial clock signal of TMS320C6748chip to drive SPI clock input of an AD converter, Texas Instruments ADS1259.

Select *File/Import/IBIS File (*.ibs)*, select *ads1259.ibs* from *<TINA directory>\Examples\IBIS*. The following dialog will be displayed. In this dialog, you can select the model to import.

ADVANCED TOPICS

DesignSoft

Now select *SCK* input signal, DIN_PD_3 model and *Max*value (for 3.3V DVDD voltage range). Press OK. The IBIS model is automatically converted to a Spice macro.

Connect the DSP I/O buffer to the input of the ADC with a lossless transmission line. Add the power source and voltage generator to create clock signal of DSP side. Place voltage pins for the simulation onto the signal nodes.

We adjust the transmission line parameters to a few inches of microstrip trace routed on a four-layer PCB. This produces cc. 500ps delay and 90 Ohms characteristic impedance.

File from *<TINA directory>\Examples\IBIS\Impedance matching of TMS320C6748.TSC*is ready to be used.

Now, click Analysis, Transient. The DSP transmits the SPI clock signal where the impedance mismatch creates reflections. The result shows the reflections created by the impedance mismatches in this circuit simulation.

At the ADC side(pin EoTL),the voltage is beyond the ground and the supply voltage, which violates the absolute maximum rating of the digital input.

To avoid under and overshoots at the line end is to match the output impedance of the driver to the trace impedance by inserting a resistor between the output and the trace. Let us place a 100 ? resistor in series now with the output.

Run the transient analysis again, and compare the results by copying the important curves with each other.

DesignSoft

Now, we can see that using the IBIS model to understand and find the critical issues with the simulation helped to solve this problem.

DesignSoft
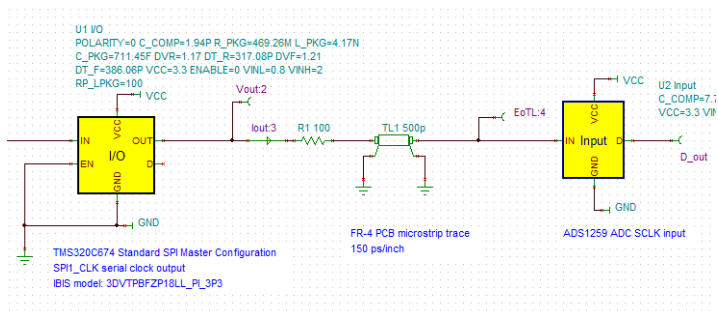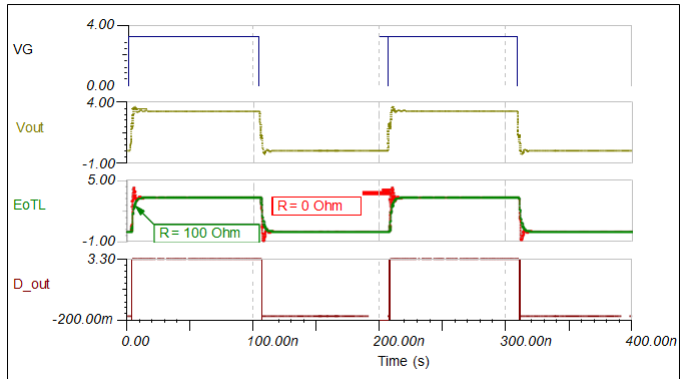
INTERPRETER

10.1   Introduction

The Interpreter is a powerful tool which extends TINA's usefulness. It allows the evaluation of mathematical expressions, the solution of equations and systems of equations, the calculation of derivatives and integrals, the plotting of results, and much more.

You may liken the Interpreter to using paper and pencil for solving problems, except that you can have TINA check the results. If you use the Interpreter after DC, AC or Transient analysis, it is extremely handy for post processing the results.

Another interesting application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators.

For educational users, TINA has two special modes. In examination mode, the student has to solve a series of problems (a problem set) either by traditional pencil-and-paper methods or by using the Interpreter and analysis functions. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. In training mode, operation is similar, except that TINA gives the student feedback about the correctness of his answer, and the student may turn to the Advisor to get help prepared by the teacher.

The Interpreter can be activated from the Tools menu. Its interface is similar to that of TINA's text editor: you enter expressions as text and then execute it as a program or an expression to evaluate or plot.

10.2   Getting Started

In this chapter we will consider some simple examples to see the types of problems we can solve with the Interpreter.

The Interpreter and the Schematic Window

Although the Interpreter can be used entirely on its own with no schematic file open in the Schematic Window, it gains in power when used in relation to an underlying schematic. When a schematic file is open in the Schematic Window, you can access its circuit variables and component parameters.

Note that, in order to make component parameters accessible, you must enter a component reference symbol in the Label field of each component's properties window. The label for a 250 ohm resistor, for example, could read *R,250, R, or R=250*. With any of these three labels, the Interpreter will use R as the reference name and will pick up the value from the Value field. The labels showing the value aid in user documentation on the schematic. In the case of components such as inductors or capacitors, additional parameters (parallel resistance or series resistance, respectively) become available. In general, each component reference symbol should be unique (an exception being where there are multiple components all with the same values).

The various parameters of sources (e.g., for a sine wave, DC level, AC amplitude, frequency, and phase) in a schematic are also available to you in the Interpreter. They appear in the symbol table under the heading Circuit variables. Be sure to fill in the label field for sources.

Finally, the results of analyses run on the schematic are also available as functions (of time or frequency). These appear in the Interpreter symbol table as External Functions, under the names they have been assigned by the Diagram Window (e.g., *AC_Ampli2, AC_Group_delay3)*.

Calculations

In the following example we will see how to do simple calculations in the Interpreter. The Schematic Editor will contain a basic RLC circuit.(*e1.sch*)

Run transient analysis with the given defaults. Click on the Tools menu and select the Interpreter. The Interpreter window appears.
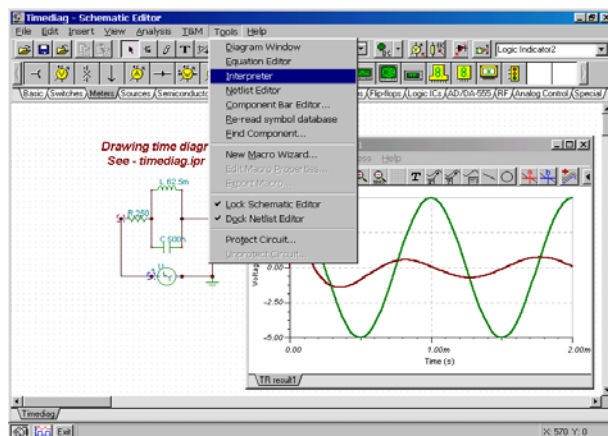
NOTE:

To insert these TINA screen views into this manual (a MicroSoft Word document), we used the *PrintScreen* key in TINA to copy the screen to the clipboard. You could also copy any windows of TINA to the clipboard by selecting the window and pressing the *Alt+PrintScreen* keys. Then we pasted the clipboard contents into this document.
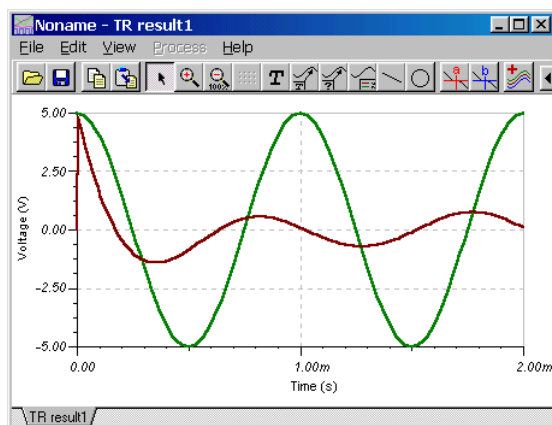
i) Let`s first check R, one of the circuit parameters.

Write in the Interpreter:

     R=   , *Then press enter*

The Interpreter will give the result

     R = [250]   , *That is 250 Ohm*

ii) Calculate the impedance of the circuit at a radian frequency of 100k:

$$W(s) = R + sL \parallel \frac{1}{sC}$$

Write in the Interpreter,

```
R=[250]
w:=100k
Z:=R+Replus(j*w*C))
Z=[250-20.0642*j]
Abs(Z)=[250.8039]      { the absolute value }
Arc(Z)=[-80.0852m]   {the phase value(radian) }
```
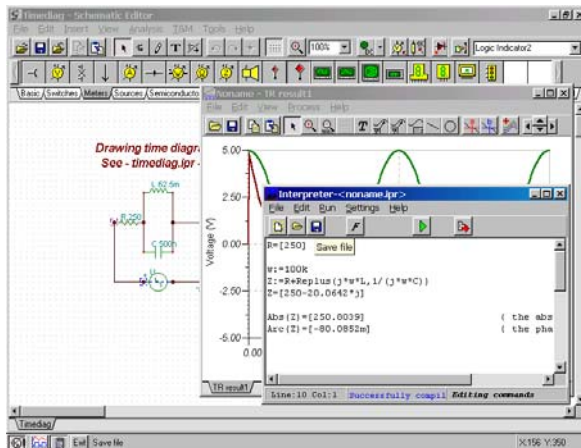

There are more functions available (See the *Appendix*).

To save these calculations, click the *Save* button on the toolbar, and follow the prompts. To recalculate results, press the *Run* button on the toolbar or use the Run menu.

10.3 Signal processing

We can use TINA to post-process the results of an analysis of a circuit.
After a DC, AC or transient analysis has been performed on a schematic,
the analysis results are placed in the symbol table of the Interpreter so
you that can refer to them as you write expressions in the Interpreter.
In this example, we will plot the instantaneous power in the resistor
using the previous TINA transient analysis result.
The instantaneous power in the resistor is:

$$p(t) = u(t) \cdot i(t) = \frac{u^2(t)}{R}$$

Before plotting this signal, we have to set the ending interval time
value (right limit parameter) in the *Drawing Preferences* dialog. Set this
parameter to the value of the transient analysis end time, 2.0m.
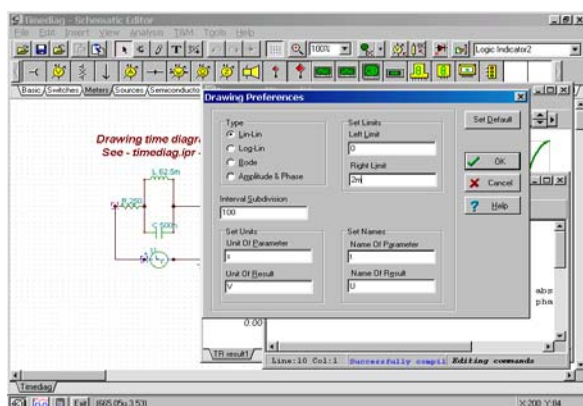
Type in the Interpreter:

```
Draw( Sqr(TR_result1(t))/R, MyResult )
```

The result appears after a moment in the Diagram Window.
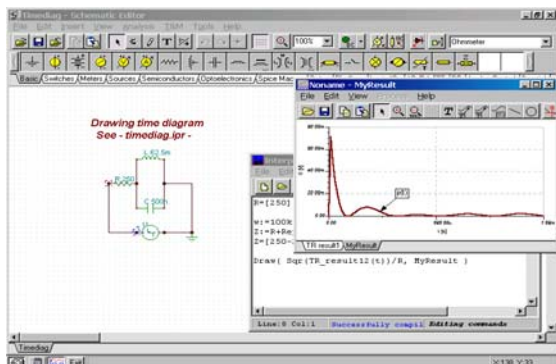
Signal definition

Another interesting application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators. User defined waveforms are extremely general and call up an equation editor window where the waveform may be defined as a mathematical expression. To make signal definition even easier, a template which you can modify to create your own signal is provided in the editor window. *Note that while TINA's Interpreter is utilized for signal definition, you access signal definition via the Schematic Editor window, not the Tools/Interpreter menu.*

As a simple example of this feature, pick up a voltage generator from the toolbar, place it in the *Schematic Editor* window, and then double-click on it. The dialog box of the Voltage Generator will appear. Click on Unit Step in the Signal field and then press the **...** button that appears. A new dialog box opens showing the available signals. Now press the rightmost button, *User Defined*, in the *Signal Editor* window. The curve and the description of a simple linear time function will appear. To change this into a quadratic function, replace *t* with *t*t* in the description on the right side and then press the *Test* button. The waveform versus time diagram of the new function will appear. Now you can use this new signal with any circuit during transient analysis. For more details of User defined excitations, refer to *Signal definition.*

Problem solving

TINA has two special modes for educational purposes. In examination mode, the student has to solve a series of problems (*a problem set*), either using pencil-and-paper or the *Interpreter* and analysis functions. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. Operation is similar in training mode**,** except that TINA gives the student feedback about the correctness of his answer. Furthermore, in training mode the student may turn to the Advisor to get help prepared by the teacher.

To learn more about this feature see *Solving DC problem.*
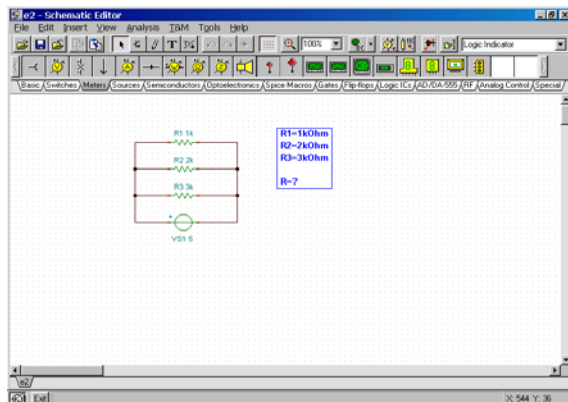
10.4 Calculations using the Interpreter

You can also use the Interpreter to carry out calculations. In this chapter, we will present examples of the calculations that you can do with the Interpreter.

To solve a problem, you first create the "source code" that the Interpreter will compile and execute. This code is structured in two parts: the first part contains function definitions, while the second part contains executable commands. Functions help you to solve complicated problems.

DesignSoft

Solving DC problems

Example

Resistors R1, R2, R3 in the circuit below are connected in parallel. Calculate the total resistance.
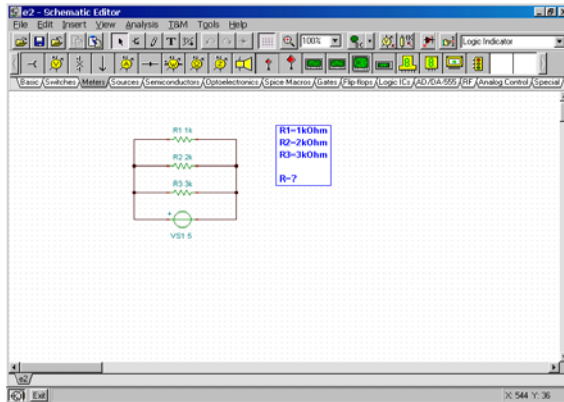
Solution

i) We will solve the problem first by entering the equation for R. Write in the Interpreter

```
R := 1/( 1/R1+1/R2+1/R3 )
R = , Then press enter
```

The Interpreter will give the result,

```
R = [545.4545]
```

Since we have a schematic file open, we can access variables derived from the component name labels—R1, R2, and R3, for example. Click on the menu to *Settings/View Symbol Table* to view which variables are available to you.

ii) We will define a function to compute the total resistance of arbitrary parallel connected resistances labeled R1, R2, R3. The Interpreter's Function declaration is used to solve more complicated problems. The syntax of the function declaration is:
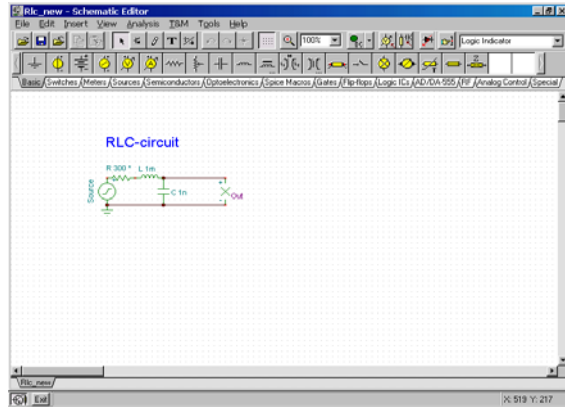
Function FunctionName(parname*1*,...,parname*n*)
Begin
 function body
End;

The function that computes the parallel resistance is:

```
Function Parallel( r1, r2, r3 );
Begin
 Parallel := 1/( 1/r1+1/r2+1/r3 );
End;
```

For a numerical solution based on the values of R1, R2, and R3 in
the underlying schematic, type the following and press Enter:

```
Parallel( R1, R2, R3 )
```

See the result:

```
=[545.4545]
```

DesignSoft

INTERPRETER
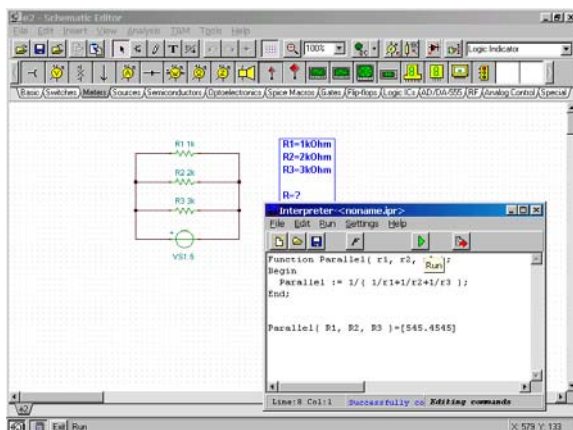
Solving AC problems

Example

Consider the following circuit. Calculate the phase and absolute value and the time function of the voltage named *Out*, when f=1MHz. The time function of the voltage generator is:

$$V_{source} \ (t) = 10 \ \cdot \cos(\ w \cdot t + 45 \ °)$$
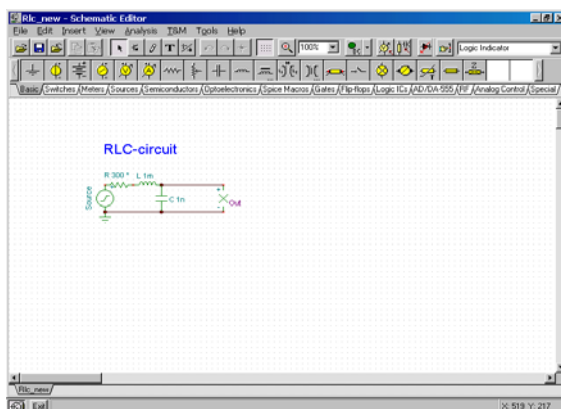
Solution

First we find the AC transfer function,

DesignSoft

$$W(s) = \frac{\frac{1}{s \cdot C}}{R + s \cdot L + \frac{1}{s \cdot C}} = \frac{1}{1 + s \cdot R \cdot C + s^2 \cdot C \cdot L}$$

Then we represent the complex source and its peak value,

$$V_{source} = 10 \cdot e^{j \cdot 45°}$$

The solution will be,

$$V_{out}(j \cdot \omega_0) = V_{source} \cdot W(j \cdot \omega_0)$$

where w_0 is the desired frequency.

In our program, we first define a variable **s** to hold the complex frequency value, **s**=jw_0. Then we calculate the transfer function W(s) using the complex peak value V_{source}. By default, TINA gives us the phase angle in radians. To present it in degrees, simply change the *Settings/Numeric formats & precisions/Angle* to DEG

With the help of the Interpreter,

```
s := j*2*pi*1e6
W := 1/(1+R*C*s+C*L*s^2)
VSource := 10*exp(j*pi/4)          {Complex peak
                                    value of the
                                    source}
VOut := VSource*W
VAbs := Abs( VOut )                {Abs. value of
                                    the output
                                    voltage }
VFi  := Arc( VOut )                {Phase value of
                                    the output
                                    voltage}

VFi=[3.9759]
VFiDeg := RadToDeg( VFi )

VAbs=[259.5747m]
VFiDeg=[227.8045]
```

Using these results, we write the output voltage:

$$V_{out}(t) = V_{Abs} \cdot \cos(\omega \cdot t + V_{Fi}) = 0.259 \cdot \cos(\omega \cdot t + 3.97)$$

Evaluating integrals

Example

Another application of the Interpreter is numerical solution of arbitrary integrals.

$$Calculate \qquad \int_{-\pi}^{+\pi} (\sin\ 2x + \cos^2 x)\,dx$$

Solution

We define a function f(x) to hold the function to be integrated; then we compute the integral using the help of TINA's *Integ* function.

```
Function f(x);
Begin
 f := sin(2*x)+Sqr(cos(x));
End;
Integ( f(s), -pi, pi, s )=[3.1416]
```

The syntax of the *Integ* command used above is,

Integ(Expression(t),Limit1,Limit2,t)
Expression(t) is the expression to be integrated; the left and right limits are Limit1, Limit2; and the variable of integration is t.

There are a large number of built-in functions that you can use in the Interpreter. The formal parameters of some functions (such as *sin, cos, exp,* etc.) are complex. See the *Appendix* for details.

Solving linear systems

One of the most powerful features of the Interpreter is the linear equation solver. With it, there is no need to reduce the equations manually to achieve a solution. The syntax of the equation solver is:

sys name*1*, ..., name*n*
  equation*1*
  .
  .
  equation*n*
end;

where equation*i* is the ith equation, and name*i* is the ith unknown variable. If there is no solution, or if there are multiple solutions, the Interpreter will present an error message.

Our linear system example will be the solution of a system of nodal equations.

Example

Let's consider the following circuit. Determine the nodal voltage at F_1.
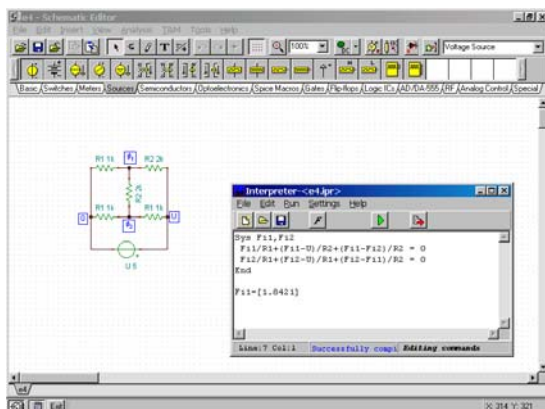
Solution

We need only enter the correct equations: the Interpreter will then solve them automatically.

The nodal equations are:

$$\frac{\phi_1}{R_1} + \frac{\phi_1 - U}{R_2} + \frac{\phi_1 - \phi_2}{R_2} = 0$$

$$\frac{\phi_2}{R_1} + \frac{\phi_2 - U}{R_1} + \frac{\phi_2 - \phi_1}{R_2} = 0$$

Note that we assume that the underlying schematic as shown, with component values, has already been created. We will reference variables from the schematic.

Entering these as source code in the Interpreter:

```
Sys Fi1,Fi2
 Fi1/R1+(Fi1-U)/R2+(Fi1-Fi2)/R2 = 0
 Fi2/R1+(Fi2-U)/R1+(Fi2-Fi1)/R2 = 0
End

Fi1=[1.8421]
```

DesignSoft

In this example, the system of equations of variables Fi1, Fi2 is solved assuming that U, R1, R2 are already defined. In TINA, these parameters are automatically referenced by the labels of the appropriate components. To help in documentation, we used TINA's text editor to put the F_1, F_2 node variable labels on the schematic as text items.

Verification

We use the *Analysis/DC Analysis/Calculate nodal voltages* to check the solution obtained by the Interpreter. After choosing *Calculate nodal voltages*, the cursor changes into a pencil shape. Click with the cursor on the F_1 node to verify its voltage.

Using *Calculate nodal voltages*, we get *1.84V,* which agrees with the value computed by the Interpreter.

10.5   Diagram Drawing

When you use the Interpreter you can access the results of previous analyses. You can also define and plot arbitrary functions and compare the theoretical solution with the analysis result.

DesignSoft
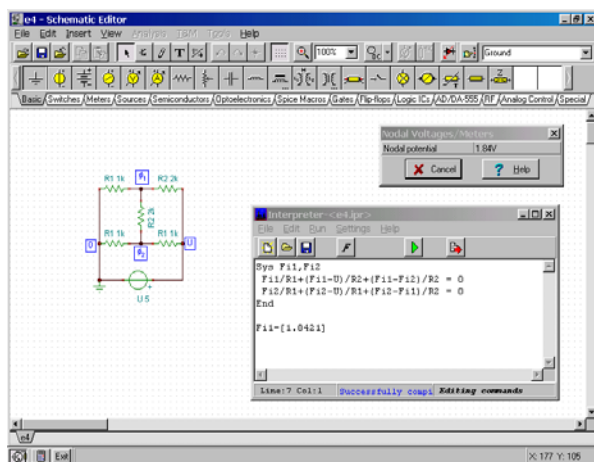
Drawing the diagrams

Using the Interpreter, you can define arbitrary functions of time and plot the results in the time domain.

Referring to the following RL circuit entered as a schematic diagram, run transient analysis with the given defaults. Then use the Interpreter to plot the theoretical result.
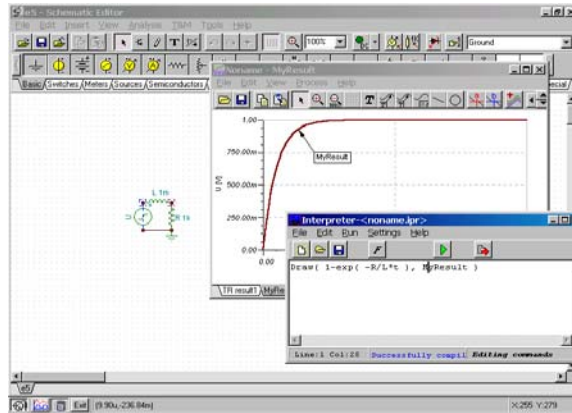
i) The time response is:

$$u(t) = 1 - e^{-\frac{R}{L} \cdot t}$$

Enter this response directly in the Interpreter, referring to the schematic parameters and using the Draw command:

```
Draw( 1-exp( -R/L*t ), MyResult )
```

We can compare the computed result and the theoretical result (in this case, the results are almost the same).

ii) Changing the R parameter, we can see how the output signal changes. We use three instances of the Draw command, each with a different R parameter, to place three curves on the diagram:

```
Draw( 1-exp( -R/L*t ), MyResult1 )
Draw( 1-exp( -2*R/L*t ), MyResult2 )
Draw( 1-exp( -3*R/L*t ), MyResult3 )
```

DesignSoft
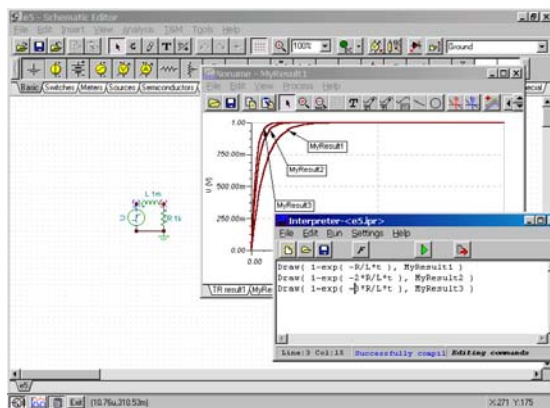
Bode diagrams

You can define arbitrary functions in the frequency domain and plot them.

Consider the following RC circuit. Draw the *AC transfer function* and calculate the amplitude and phase for *f*=5kHz.

Run AC analysis with the given defaults. Using Symbolic Analysis we can get the AC transfer function.

INTERPRETER

DesignSoft
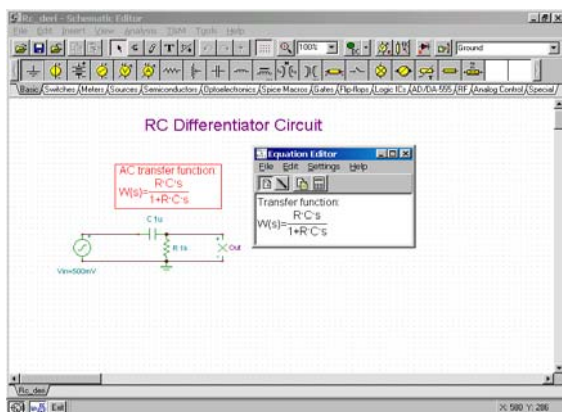
<u>Drawing the AC transfer function.</u>

Now let's use the Interpreter to draw the AC transfer function with the given defaults. Before plotting the result, we have to set some parameter in the Drawing Preferences dialog.

- Set the *left and the right limit* parameter in the Drawing Preferences dialog. Set these to the AC analysis *start and end frequency*.
- Set the *name of parameter* to *s* in this dialog box also.
- Set the *type* to Bode.

Then we can plot the desired function:

```
Draw( 1/(1+C*R*s), MyResult )
```

Two new pages will be opened in the Diagram Window, for the amplitude and the phase diagrams of the function.
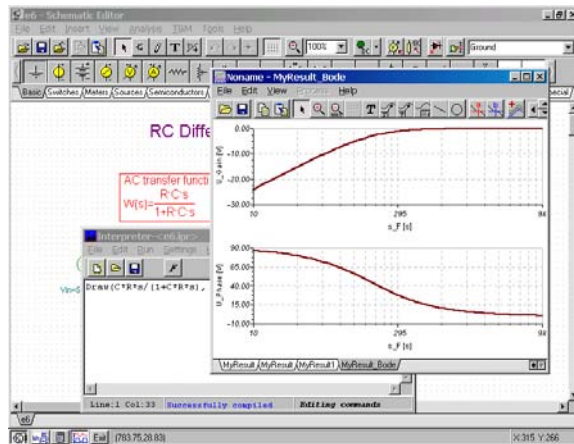
<u>Calculating the amplitude and the phase when *f* = 5kHz</u>

The Interpreter provides a number of built-in functions that help in solving complicated problems. Using the *Abs* and the *Arc* functions, we can solve this problem easily.

```
f:=5k
ZAbs := Abs( 1/(1+C*R*j*2*pi*f) )
ZAbs =[31.8149m]
Db( ZAbs )=[-29.9474]
```

```
ZFi  := Arc( 1/(1+C*R*j*2*pi*f) )
ZFi  =[-1.539]
```

In this example, we define a variable *f* for frequency and set it to the given value. Then, using the *Abs* function, we calculate the absolute value (magnitude) of the transfer function at the given frequency. Next, using the *Db* function (returns $20*\log_{10}(x)$, provided $x > 0$), we get the amplitude in decibels. We can check these results by running an AC Analysis on the circuit in the schematic window, going to the diagram window, and placing a cursor over the amplitude or phase curves at 5kHz. Finally, we use the *Arc* function to get the phase angle in radians per second.
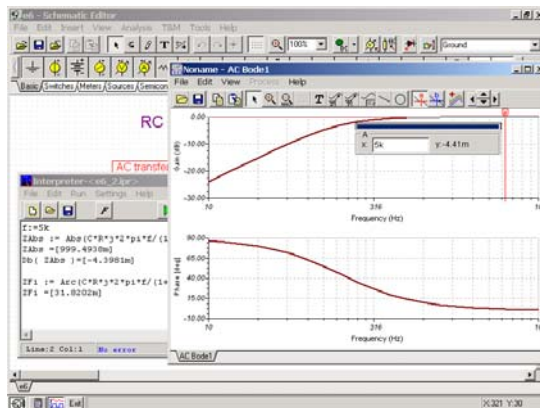

Drawing impedances

The Interpreter can also draw impedance functions, with a choice of *linear*, *log-lin* or *Bode* diagrams.

Starting with the RL circuit shown below, we will use the Interpreter's LOG-LIN diagram to draw the impedance as a function of frequency. We are after the impedance that the source sees looking into the network from its terminals. Note that we have two identically named and valued inductors and two identically named and valued resistors. If the resistors had different values, we would have to label them with different reference designators in order for the Interpreter to pick up the correct values.
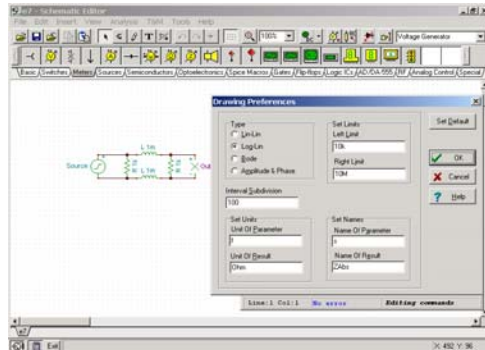
Drawing the impedance in the frequency domain

Before plotting the impedance function, we have to setup some parameters in the Interpreter's Drawing Preferences dialog.

- Set the *left and the right limit* parameter in the Drawing Preferences dialog. Set this parameter to the AC analysis *start and end frequency.*(10k and 10M)
- Set the *name of parameter* to *s*.
- Set the *name of result* to *ZAbs.*
- Set the *unit of parameter* to *f*
- Set the *unit of result* to *Ohm*
- Set the *type* to LOG-LIN.

We can draw the desired function:

```
Function Z(u);
Begin
 Z := u*L+Replus( R, R+u*L );
End;

Draw( Z(s), Impedance )
```

At first we define a function to hold the impedance expression. Note in the function that *u* is a parameter that is replaced with **s** when the function is actually "called" in the Draw statement. We use here the built-in *Replus* function to calculate the impedance of two parallel connected impedances. The *Replus* function accepts either complex or real arguments. Then, using this function in the Draw statement, and relying on the current component values of the underlying schematic, we draw the impedance function in the Diagram Window.

Two new pages are created in the Diagram Window—the amplitude and the phase diagrams of the given function.
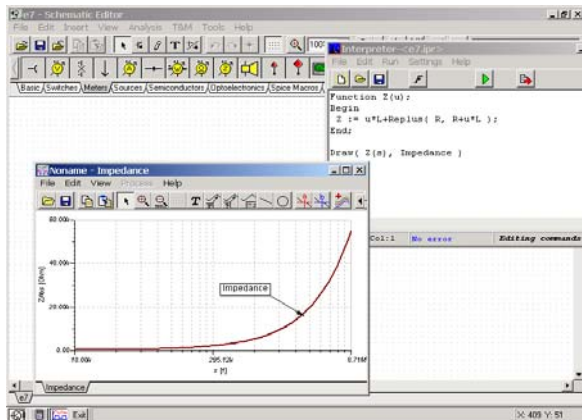
10.6 Calculations and diagram drawing

In this chapter we will use convolution to obtain the response of a circuit in the time domain. Convolution is a mathematical procedure in circuit theory that is used to be determine the response in time domain.

An example of convolution

Let's use convolution to find the time domain response of the RL circuit below. The input to the circuit is a *unit step* voltage generator and the output is the output voltage on the resistor R.
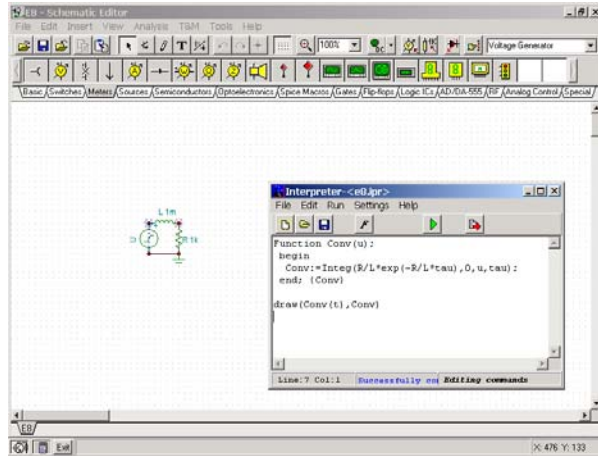
Solution

The theoretical response of this circuit in the time domain is,

$$U_R(t) = 1 - e^{-\frac{R}{L}t}$$

What does convolution mean and how can it be used to determine the response of this circuit in the time domain? The convolution of two arbitrary signals is,

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) d\tau$$

It can be proven that the time response of a circuit to a source *s(t)* is:

$$y(t) = w(t) * s(t) = \int_{-\infty}^{\infty} w(\tau) s(t-\tau) d\tau$$

where *w(t)* is the response of the circuit to the Dirac impulse. In this case,

$$w(t) = \frac{R}{L} e^{-\frac{R}{L}t}$$

So, we have to define a function that computes the desired convolution integral.

```
Function  Conv(u);
 begin
   Conv:=Integ(R/L*exp(-R/L*tau),0,u,tau);
 end;  {Conv}

draw(Conv(t),Conv)
```
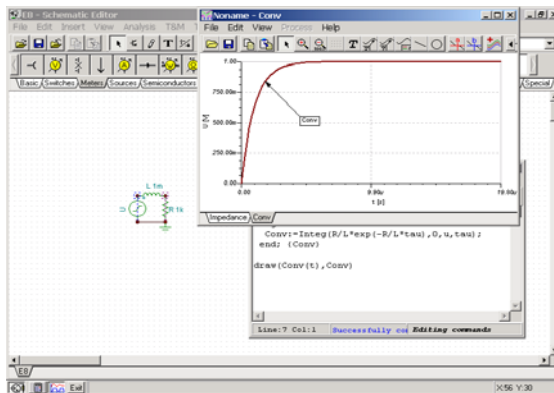

Verification

Let's use Transient Analysis on the circuit to check the result just found using the theoretical convolution function:

$$U_R(t) = 1 - e^{-\frac{R}{L}t}$$

We will check this result after running Transient Analysis by using the *copy-paste* functions in the *Diagram Window*. Moving to the Transient Analysis window, *TR_result1*, by clicking on the tab, we select the curve by clicking on it. We put it in the *Clipboard* by pressing the *Ctrl+C* key. Changing now to the *Resp tab* of the Diagram Window, we copy the selected curve into this diagram by pressing the *Ctrl+V* keys.

We can see that the results are the same—one overlays the other xactly.
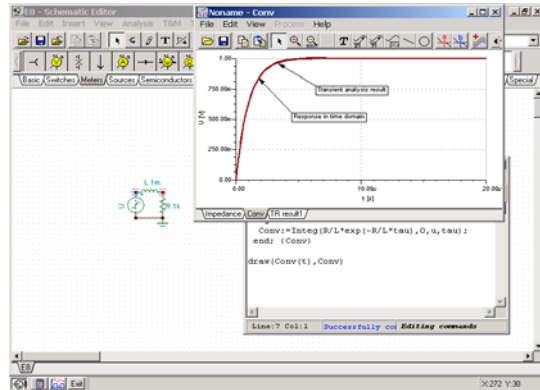
10.7 Processing TINA's results

Another interesting application of the Interpreter is post- processing the results of TINA analyses with arbitrary functions. When you carry out DC, AC or transient analysis on a circuit in the Schematic Window, the analysis results also appear in the Interpreter symbol table, so you can reference them for additional processing.
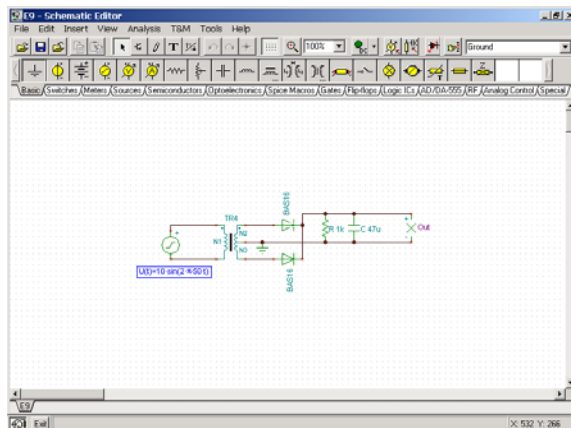
Calculating average power

Example

Let's consider the following circuit. Determine the average power appearing in the resistor R, sensed by the voltage labeled *Out,* as a function of time.

Solution

The instantaneous power in the resistor is:

$$p(t) = u(t) \cdot i(t) = \frac{u^2(t)}{R}$$

The average power,

$$P(t) = \frac{1}{R \cdot t} \int_0^t p(s)\,ds$$

And,

$$P(t) = \frac{1}{R \cdot t} \int_0^t u^2(s)\,ds$$

We will compute and draw the power function with the Interpreter. The *u(t)* function in the integral (the voltage across the resistor) will be picked up from a previous TINA result produced by transient analysis if we reference it correctly, as TR_result1.
The solution as found via the Interpreter,

```
Function P(x);
 begin
P := Integ( Sqr( TR_result1( tau ) ),0,x,tau );
 end; {P}

Function Power(x);
 begin
  Power := P(x)/(R*x);
 end;

Draw( Power(t), PowerRes )
```
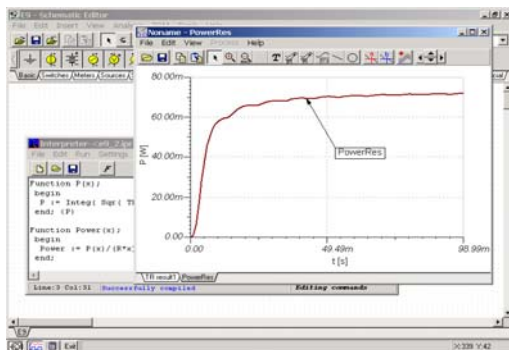
Verification

For verification we will use the fact that in this case the instantaneous voltage across the resistor settles down to an average amount with a ripple component, and the instantaneous and average values come out very close.

Using the TINA transient analysis output, let's check the result computed by the Interpreter.

In the TINA transient analysis time diagram, we see that after awhile, the voltage across the resistor doesn't change much. This will let us use an average value which we will read off the curve in the Diagram Editor. Press the '*Cursor a'* button, and the cursor will change into a **+**. Click this new cursor on the output curve, position it to around *80m*, and read the voltage:

$U_{TRAN}(80m) = 8.4$

The average power at this time is:

$$p(80\,m) = \frac{u^2(80\,m)}{R} = 0.0705\ W = 70.05\ mW$$

Using the Interpreter we see that
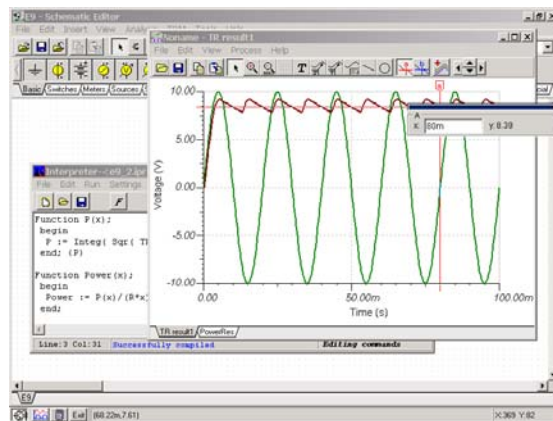
Power(80m)=[72.7884]; that is, 72.79mW

We find just a small difference of *2.74mW*.

10.8 Signal definition

Another application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators. User defined waveforms are extremely general and call up an equation editor window where the waveform may be defined as a mathematical expression. To make signal definition even easier, a template which you can modify to create your own signal is provided in the editor window. *Note that while TINA's Interpreter is utilized for signal definition, you access signal definition via the Schematic Editor window, not the Tools/Interpreter menu.*

Clicking on a generator, we can establish the *Signal* properties. Choose type *user defined* and the *Signal Editor* opens. In this editor, as in the Interpreter editor, we can edit, save and compile source code.

A separate window, visible just below the editor, contains a list of the symbols for pre-defined variables, circuit variables, and other user defined functions. We can use these to create a new user defined waveform. Here's how we would define an amplitude modulated signal:

```
Function Signal(t);
Begin
fCarrier:=1M;
wC:=fCarrier*2*pi;
m:=0.5;
fmodsig:=100k;
wS:=fmodsig*2*pi;
Signal := 1*sin(wC*t)*(1+m*sin(wS*t));
End;
```

If we continue and define more functions, the function named *Signal* will appear at the top of the symbol list. In the body of the functions, we can define variables and use arbitrary assignments of the *built-in* functions.
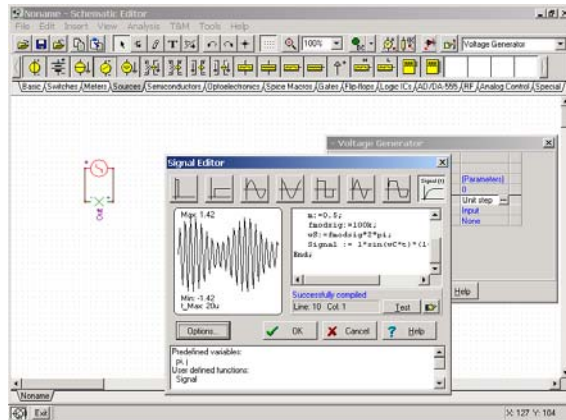
10.9 Using the Interpreter in the training and examination mode of TINA

For educational user, *TINA* has two special modes. In examination mode, the student has to solve a series of problems (*a problem set*) either by traditional pencil-and-paper methods or by using the *Interpreter and analysis functions*. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. In training mode, operation is similar**,** except that TINA gives the student feedback about the correctness of his answer, and the student may turn to the Advisor to get help prepared by the teacher.

Solving a DC problem

The following example is a DC problem. The problem requires the student to calculate the voltage on the resistor R2. Clicking on the *Voltage division entry* in the examination panel, the circuit appears.

There are two ways to solve problems in TINA's training & examination mode. The first is by using the *Result field* in the Training & Examination panel, while the second is by using the *Interpreter* directly.

Using the Interpreter directly, we can use the *Result* and the *Submit* functions to pass the result value to the training and examination module of TINA.

INTERPRETER

Using the Interpreter in the Result field

In the result field, you may type in not only a number, but also a formula based on the symbols in the circuit. In this case, the Interpreter is automatically (internally) invoked and evaluates the formula.

Submitting a solution from the Interpreter window

After clicking the Interpreter button, the Interpreter window opens. Remember that our example problem is to calculate the voltage on the resistor R2. We define two variables (I and U2) to hold the current and the voltage on R2 using Ohm's law. Using the *Submit* function, we pass the result to the examination module.

The solution with the Interpreter,

```
I  := VG/( R1+R2 )
I = [2]


U2  :=  I*R2
U2 = [60]

   Submit( 60 [V] )
```
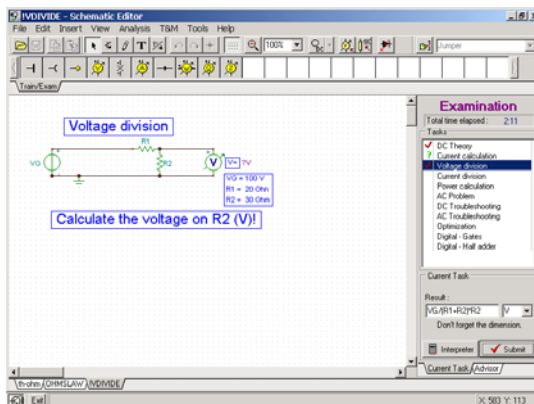
Solving an AC problem

The following is an example of an AC problem. Click on the AC problem entry in the examination and training panel, and the circuit appears. Click the Interpreter button and the Interpreter window opens.

AC problems are normally too complex to allow a one line solution in the result field, so we use the Interpreter Window. But if the expression is small enough, we could use the result field also.

The problem is calculating the complex current through the inductor. First we define two complex variables (Z and I) to hold the complex impedance and the complex current. Using the *Submit2* function we pass the result to the examination module.

The solution with the Interpreter

```
Z := j*VGw*L
I := VGA / Z
Abs(I)=[10]
Arc(I)=[-1.5708]
Submit2( 10[A], -1.5708 [rad] )
```
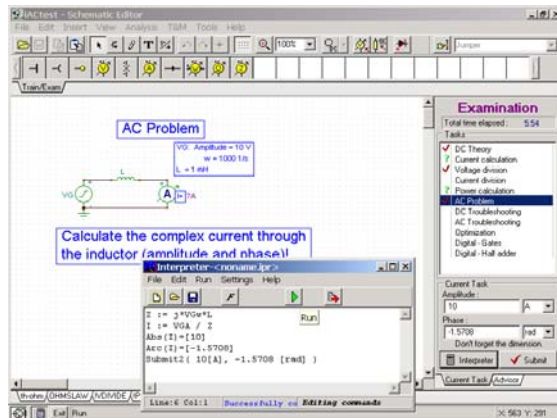
VGA stands for the voltage generator amplitude, while *VGw* stands for its radian frequency. It is good practice to review the results carefully before adding the Submit command to your solution. Once you press the Run button in a program containing the submit command, you are committed: TINA will check your solution, tell you if it is right or wrong, and submit it.

Appendix

There are a large number of *built-in* functions that you may use in the Interpreter. The formal parameters For some of these functions (identified with a C), the formal parameters accept complex numbers.

General purpose functions

Sin(x)
Calculates the sine of x

Cos(x)
Calculates the cosine of x

Tan(x)
Calculates the tangent of x

Atan(x)
Calculates the arc tangent of x

Exp(x)   C

Calculates the exponential function e^x

Ln(x)   C
Calculates the natural logarithm of x, x>0

Sqr(x)   C
Calculates the square of x

Sqrt(x)   C
Calculates the positive square root of x

Abs(x)   C
Calculates the absolute value of x

Sgn(x)

Sign function: $\begin{cases} 1, \text{ if x} > 0 \\ 0, \text{ if x} = 0 \\ -1, \text{ if x} < 0 \end{cases}$

Re(x)   C
Calculates the real part of x

Im(x)   C
Calculates the imaginary part of x

Arc(x)   C
Calculates the phase of x in radian

Sum(f(x))
Integral of f(x) in the [0..x] interval

Example:

```
Sum(sin(pi))=[2]
```

Integ(Expression(t),Limit1,Limit2,t)
Integral of Expression(t) in the [Limit1..Limit2] interval. The
variable of integration is t.

Example:

a)
```
Function Con(t);
Begin
 Con := t;
End;
Integ(Con(t),0,1,t)=[0.5]
```
b)

The meaning of Integ(Con(t),0,x,t) is: $\int\limits_{0}^{x} Con(t)dt$

D(f(x))
Derivative of f(x) at x

Example:

```
D(sin(0))=[1]
```

Periodic(x,y)
Where y is the period value. The function transforms x into the [0..y]
interval according to y.

Example:
```
Periodic(5.3,2)=[1.3]
```

Not(x)
Calculates the bitwise negation of x

RadToDeg(x)
Converts x, in radians to degrees

DegToRad(x)
Converts x, in degrees to radians

Special functions for electrical engineering applications

Calculation of parallel impedances

$$\textbf{Replus}(x,y)=\frac{x*y}{x+y} \qquad\qquad C$$

Unit step

$$\textbf{E}(x)= \begin{cases} 0, \text{ if } x \leq 0 \\ 1, \text{ if } x > 0 \end{cases}$$

Calculation of a value in decibels

$$\textbf{Db}(x)=20*lg(x), x>0 \qquad\qquad C$$