
Qucs Help Documentation

Выпуск 0.0.19

Qucs Team

нояб. 05, 2017

1	Фон	3
2	Быстрый старт Qucs в аналоговом моделировании цепи	5
3	Быстрый старт в оптимизации	11
4	Getting Started with Octave Scripts	23
5	Краткое описание действий	25
6	Работа с подсхемами	29
7	Быстрый старт в цифровом моделировании	33
8	Краткое описание математических функций	37
9	Перечень специальных символов	45
10	Согласование цепей	49
11	Установленные файлы	51
12	Qucs File Formats	53
13	Subcircuit and Verilog-A RF Circuit Models for Axial and Surface Mounted Resistors	61

Содержание:

The ‘Quite universal circuit simulator’ Qucs (pronounced: kju:ks) is an open source circuit simulator developed by a group of engineers, scientists and mathematicians under the GNU General Public License (GPL). Qucs is the brain-child of German Engineers Michael Margraf and Stefan Jahn. Since its initial public release in 2003 around twenty contributors, from all regions of the world, have invested their expertise and time to support the development of the software. Both binary and source code releases take place at regular intervals. Qucs numbered releases and day-to-day development code snapshots can be downloaded from (<http://qucs.sourceforge.net>). Versions are available for Linux (Ubuntu and other distributions), Mac OS X © and the Windows © 32 bit operating system.

In the period since Qucs was first released it has evolved into an advanced circuit simulation and device modelling tool with a user friendly “graphical user interface” (GUI) for circuit schematic capture, for investigating circuit and device properties from DC to RF and beyond, and for launching other circuit simulation software, including the FreeHDL (VHDL) and Icarus Verilog digital simulators. Qucs includes built-in code for processing and visualising simulation output data. Qucs also allows users to process post-simulation data with the popular Octave numerical data analysis package. Similarly, circuit performance optimisation is possible using the A SPICE Circuit Optimizer (ASCO) package or Python code linked to Qucs.

Between 2003, and January 2015, the sourceforge Qucs download statistics show that over one million downloads of the software have been recorded. As well as extensive circuit simulation capabilities Qucs supports a full range of device modelling features, including non-linear and RF equation-defined device modelling and the use of the Verilog-A hardware description language (HDL) for compact device modelling and macromodelling. Recent extensions to the software aim to diversify the Qucs modelling facilities by running the Berkeley “Model and Algorithm Prototyping Platform” (MAPP) in parallel with Qucs, using Octave launched from the Qucs GUI. In the future, as the Qucs project evolves, the software will also provide circuit designers with a choice of simulation engine selected from the Qucs built-in code, ngspice and Xyce ©.

Qucs is a large software package which takes time to learn. Incidentally, this statement is also true for other GPL circuit simulators. New users must realise that to get the best from the software some effort is required on their part. In particular, one of the best ways to become familiar with Qucs is to learn a few basic user rules and how to apply them. Once these have been mastered users can move on with confidence to next level of understanding. Eventually, a stage will be reached which allows Qucs to be used productively to model devices and to investigate the performance of circuits. Qucs is equally easy to use by absolute beginners, like

school children learning the physics of electrical circuits consisting of a battery and one or more resistors, as it is by cutting edge engineers working on the modelling of sub-nano sized RF MOS transistors with hundreds of physical parameters.

The primary purpose of these notes is to provide Qucs users with a source of reference for the operation and capabilities of the software. The information provided also indicates any known limitations and, if available, provides details of any work-arounds. Qucs is a high level scientific/engineering tool who's operation and performance does require users to understand the basic mathematical, scientific and engineering principles underlying the operation of electronic devices and the design and analysis of electronic circuits. Hence, the individual sections of the Qucs-Help document include material of a technical nature mixed in with details of the software operation. Most sections introduce a number of worked design and simulation examples. These have been graded to help readers with different levels of understand get the best from the Qucs circuit simulator. Qucs-Help is a dynamic document which will change with every new release of the Qucs software. At this time, Qucs release 0.0.19, the document is far from complete but given time it will improve.

Быстрый старт Qucs в аналоговом моделировании цепи

Qucs is a scientific/engineering software package for analogue and digital circuit simulation, including linear and non-linear DC analysis, small signal S parameter circuit analysis, time domain transient analysis and VHDL/Verilog digital circuit simulation. This section of the Qucs-Help document introduces readers to the basic steps involved in Qucs analogue circuit simulation. When Qucs is launched for the first time, it creates a directory called `.qucs` within the user's home directory. All files involved in Qucs simulations are saved in the `.qucs` directory or in one of its sub-directories. After Qucs has been launched, the software displays a Graphical User Interface window (GUI) similar, or the same, to the one shown in Figure 1.

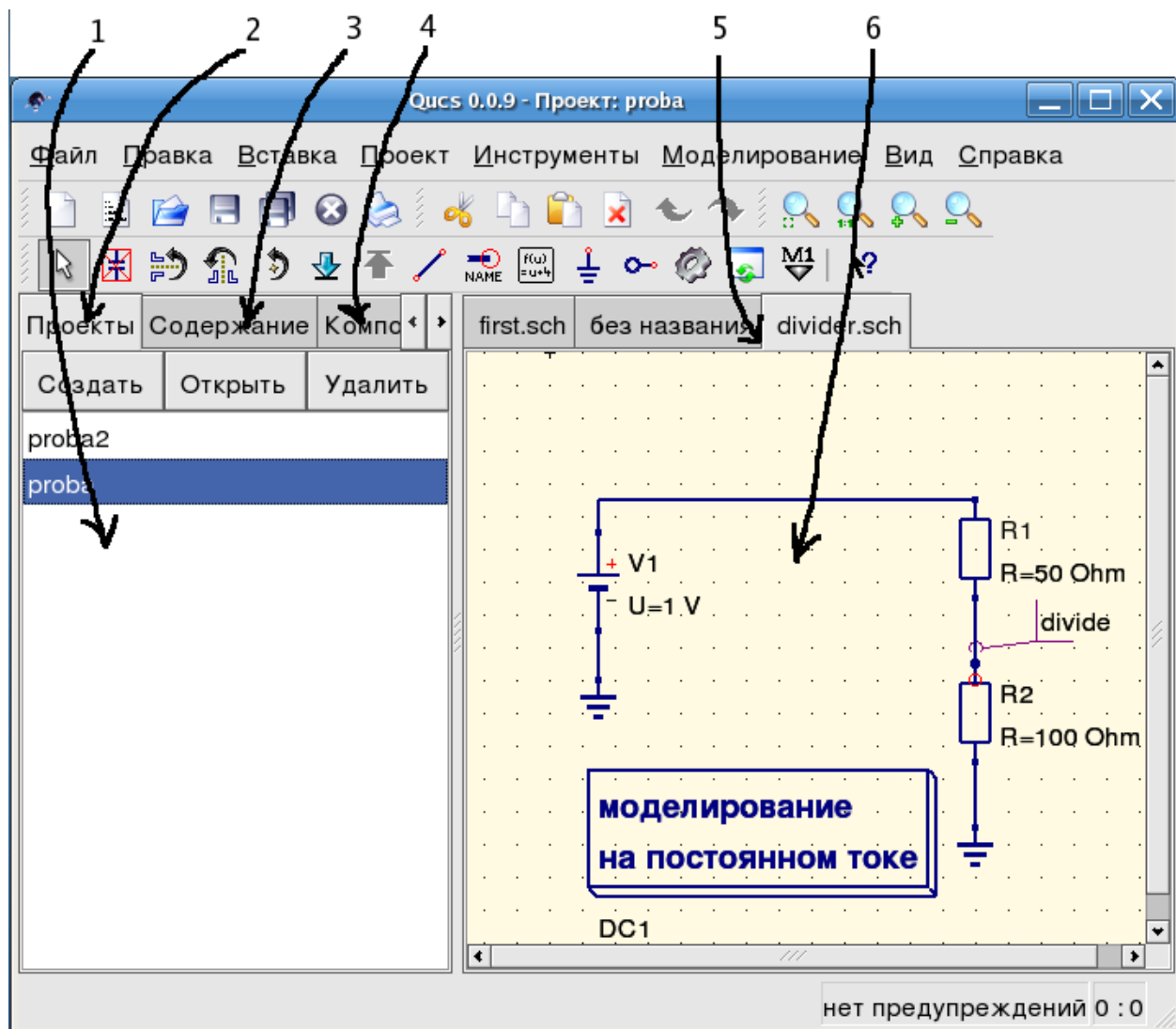


Рис. 1 - Главное окно Qucs

Before using Qucs it is advisable to set the program application settings. This is done from the **File** → **Application Settings** menu. Clicking on **Application Settings** causes the **EditQucsProperties** window to be displayed, see Figure 2. Complete, with appropriate entries for your Qucs installation, the **Settings**, **Source Code Editor**, **File Types** and **Locations** menus.

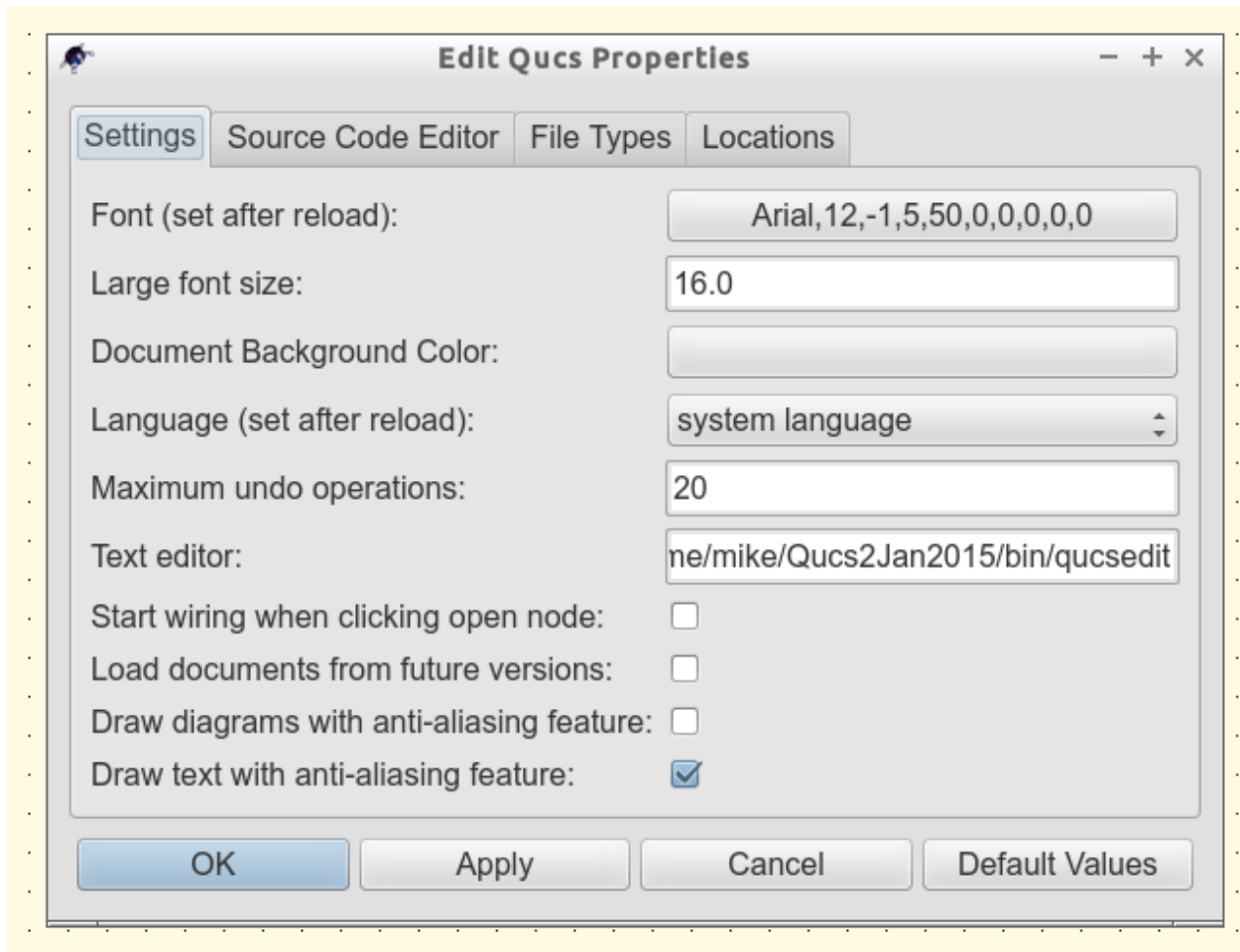


Рис. 2 - Окно Qucs с редактированием свойств.

On launching Qucs a working area labelled (6) appears at the centre of the GUI. This window is used for displaying schematics, numerical and algebraic model and circuit design data, numerical output data, and signal waveforms and numerical data visualised as graphs, see Figure 3. Clicking, with the left hand mouse button on any of the entries in the tabular bar labelled (5) allows users to quickly switch between the currently open documents. On the left hand side of the Qucs main window is a third area labelled (1) whose content depends on the status of **Projects** (2), **Content** (3), **Components** (4) or **Libraries**. After running Qucs, the **Projects** tab is activated. However note, when Qucs is launched for the first time the **Projects** list is empty.

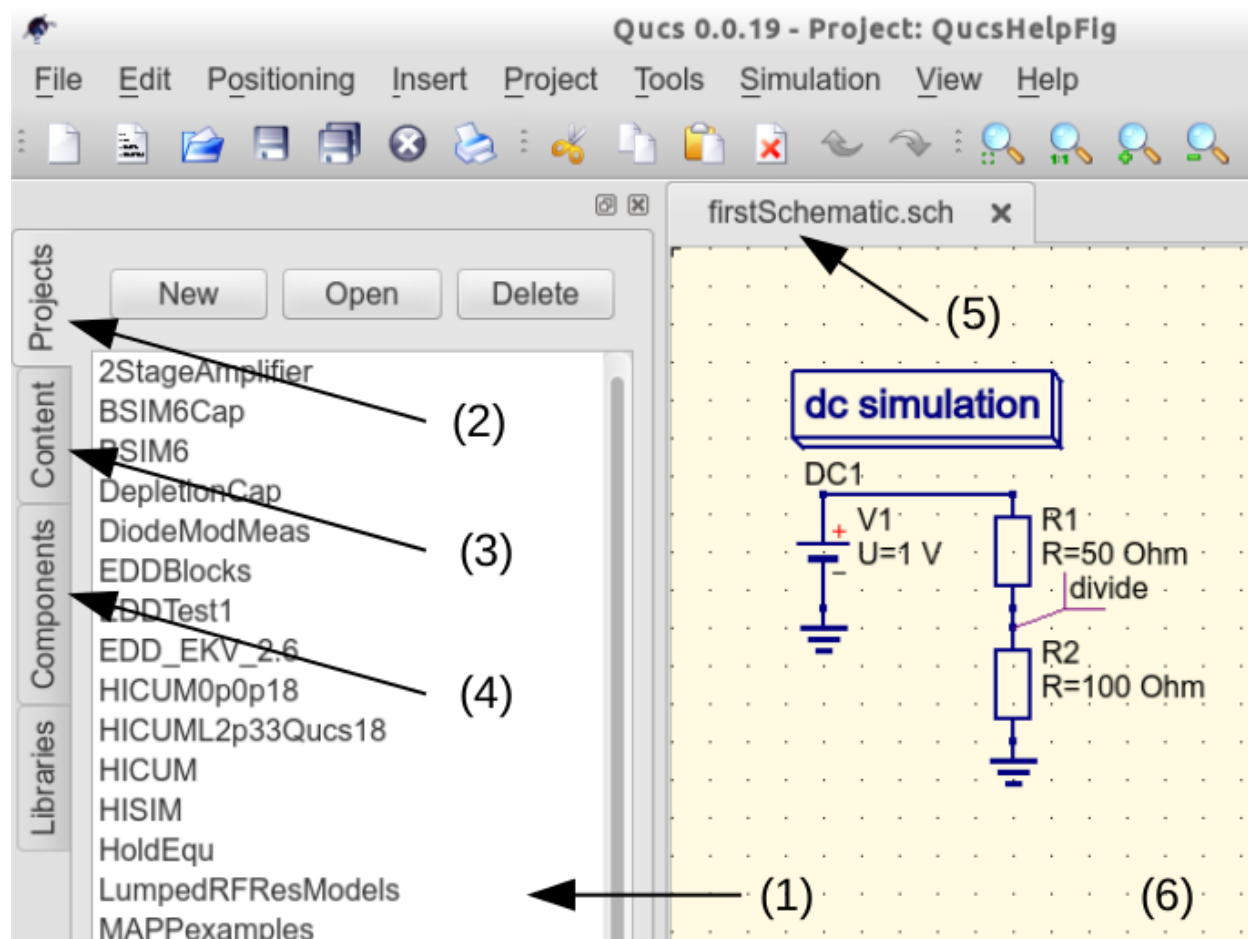


Рис. 3 - Главное окно Qucs с отмеченными рабочими зонами

To enter a new project left click on the **New** button located on the right above window (1). This action causes a Qucs GUI dialogue to open. Enter the name of a Qucs project in the box provided, for example enter *QucsHelpFig* and click on the **OK** button. Qucs then creates a project directory in the `~/qucs` directory. In the example shown in Figure 3 this is called **QucsHelpFig_prj**. Every file belonging to this new project is saved within the **QucsHelpFig_prj** directory. On creation a new project is immediately opened and its name displayed on the Qucs window title bar. The left tabular bar is then switched to **Content**, and the content of the currently opened project displayed. To save an open document click on the **save** button (or use the main menu: **File** → **Save**). This step initiates a sequence which saves the document displayed in area (6). To complete the save sequence the program will request the name of your new document. Enter *firstSchematic*, or some other suitable name, and click on the **OK** button to complete the save sequence.

As a first example to help you get started with Qucs enter and run the simple DC circuit shown in Figure 3. The circuit illustrated is a two resistor voltage divider network connected to a fixed value DC voltage source. Start by clicking on the **Components** tab. This action causes a combo box to be displayed from which a component group may be chosen and the required components selected. Choose components group **lumped components** and click on the first symbol: **Resistor**. Next move the mouse cursor into area (6). Pressing the right mouse button rotates the **Resistor** symbol. Similarly, pressing the left mouse button places the component onto the schematic at the place the mouse cursor is pointing at. Repeat this process for all components shown in Figure 3. The independent DC voltage source is located in the **sources** group. The ground symbol can be found in the **lumped components** group or selected from the Qucs toolbar. The icon requesting DC simulation is listed in the **simulations** group. To edit the parameters of the second resistor, double-click on it. A dialogue opens which allows the resistor value to be changed; enter *100 Ohm* in the edit field on the right hand side and click enter.

To connect the circuit components shown in Figure 3, click on the wire toolbar button (or use the main menu: **Insert** → **Wire**). Move the cursor onto an open component port (indicated by a small red circle at the end of a blue wire). Clicking on it starts the wire drawing sequence. Now move the drawing cursor to the end point of a wire (normally this is a second red circle attached to a placed component) and click again. Two components are now connected. Repeat the drawing sequence as many times as required to wire up the example circuit. If you want to change the corner direction of a wire, click on the right mouse button before moving to an end point. You can also end a wire without clicking on an open port or on a wire; just double-click the left mouse button.

As a final step before DC simulation label the node, or nodes, whose DC voltage is required, for example the wire connecting resistors **R1** and **R2**. Click on the label toolbar button (or use the menu: **Insert** → **Wire Label**). Now click on the chosen wire. A dialogue opens allowing a node name to be entered. Type *divide* and click the **OK** button. If you have drawn the test schematic correctly the entered schematic should look the same, or be similar to, the one shown in Figure 3.

To start DC simulation click on the **Simulate** toolbar button (or use menu: **Simulation** → **Simulate**). A simulation window opens and a sliding bar reports simulation progress. Normally, all this happens so fast that you only see a short flickering on the PC display (this depends on the speed of your PC). After finishing a simulation successfully Qucs opens a data display window. This replaces the schematic entry window labelled (6) in Figure 3. Next the **Components** → **diagrams** toolbar is opened. This allows the simulation results to be listed. Click on the **Tabular** item and move it to the display working area, placing it by clicking the left hand mouse button. A dialogue opens allowing selection of the named signals you wish to list, see Figure 4. On the left hand side of the **Tabular** dialogue (called Edit Diagram Properties) is listed the node name: **divide.V**. Double-click on it and it will be transferred to the right hand side of the dialogue. Leave the dialogue by clicking the **OK** button. The DC simulation voltage data for node **divide** should now be listed in a box on the data display window, with a value of 0.666667 volts.

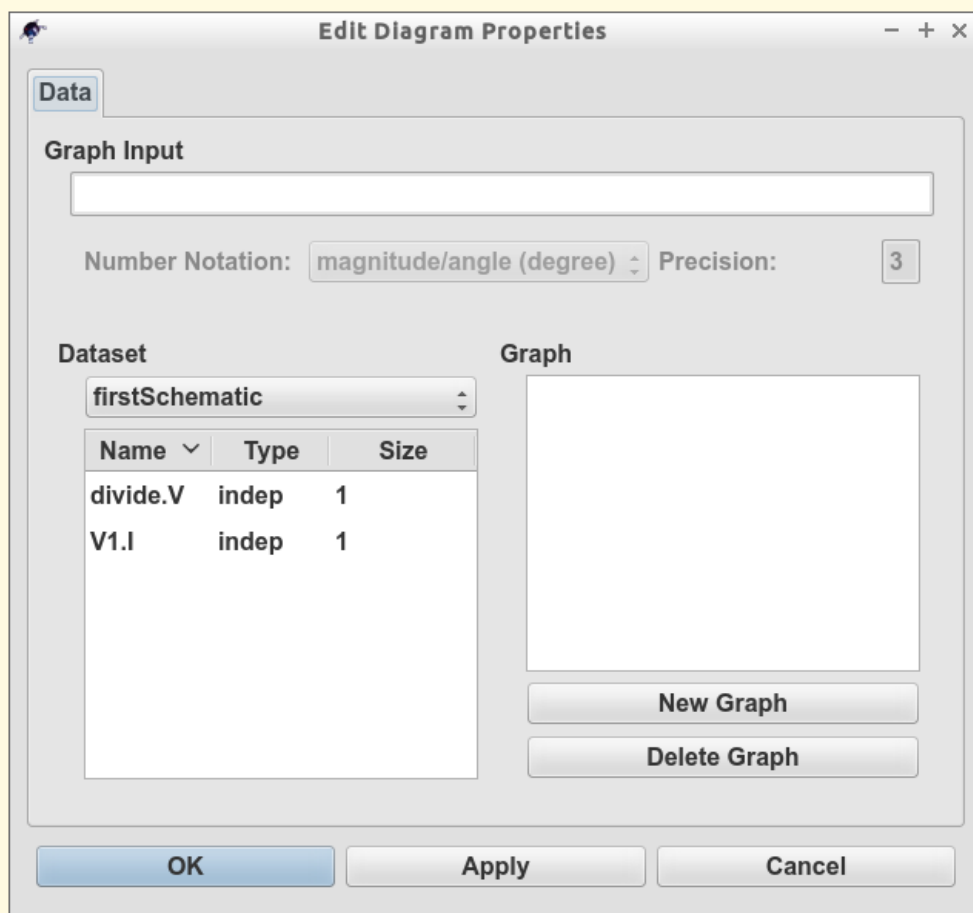


Рис. 4 - Окно отображения данных Qucs, показывающее табличный диалог

Быстрый старт в оптимизации

Для оптимизации цепей в Qucs используется утилита ASCO (<http://asco.sourceforge.net/>). Ниже дается краткое описание того, как подготовить схему, выполнить утилиту и интерпретировать результат. Перед этим в системе должна быть установлена ASCO.

Оптимизация цепи - это ничто иное, как минимизация функции стоимости. Это может быть время задержки или нарастания цифровой цепи, либо мощность или усиление аналоговой цепи. Еще одна возможность - определить задачу оптимизации как сочетание функций, что в этом случае ведет к определению показателя добротности.

Для подготовки списка связей к оптимизации две вещи должны быть добавлены к уже существующей схеме: нужно вставить уравнение(я) и блок компонента оптимизации. Возьмите схему из рис. 1 и внесите в нее изменения так, чтобы получить в результате схему на рис. 2.

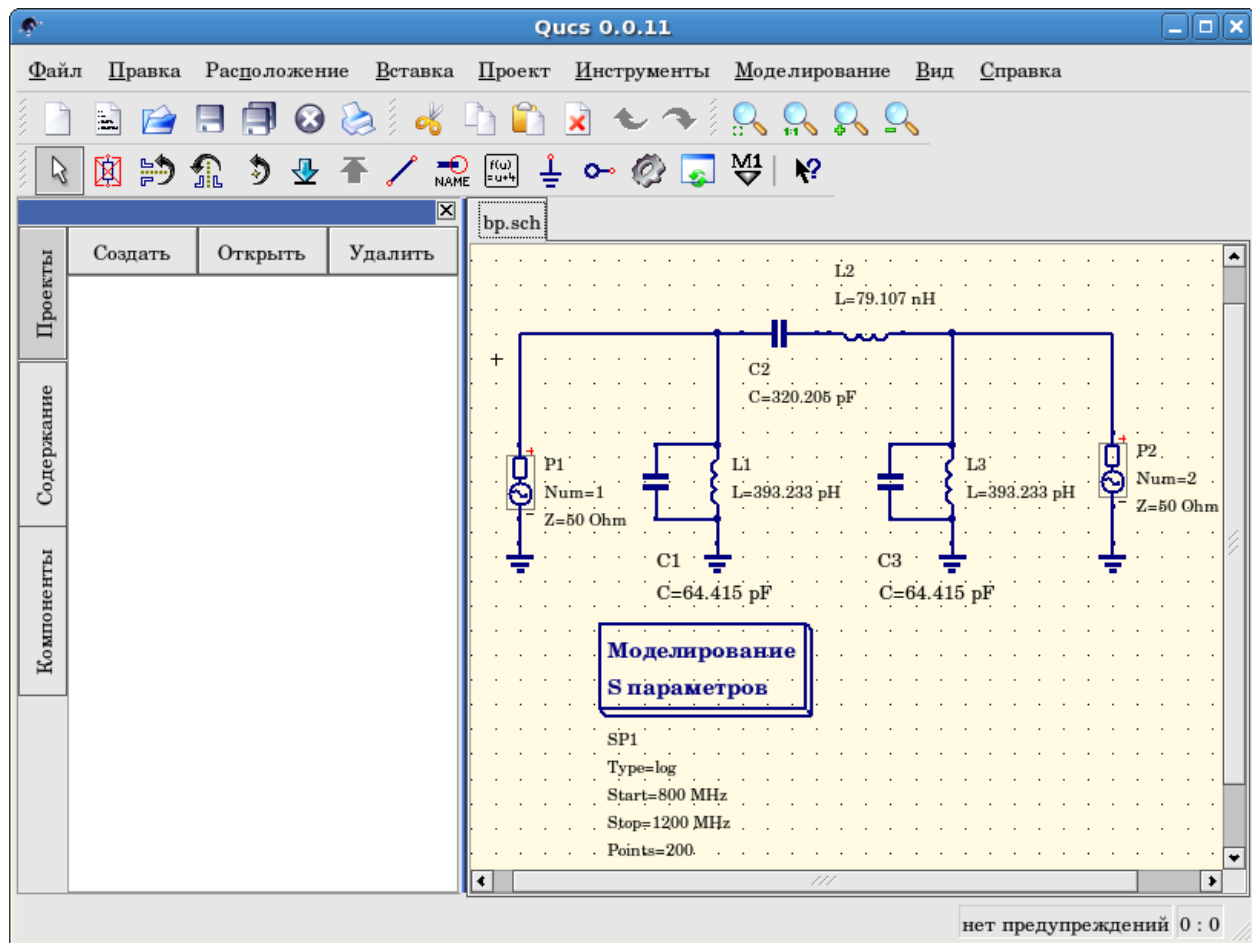


Рис. 1 - Исходная схема.

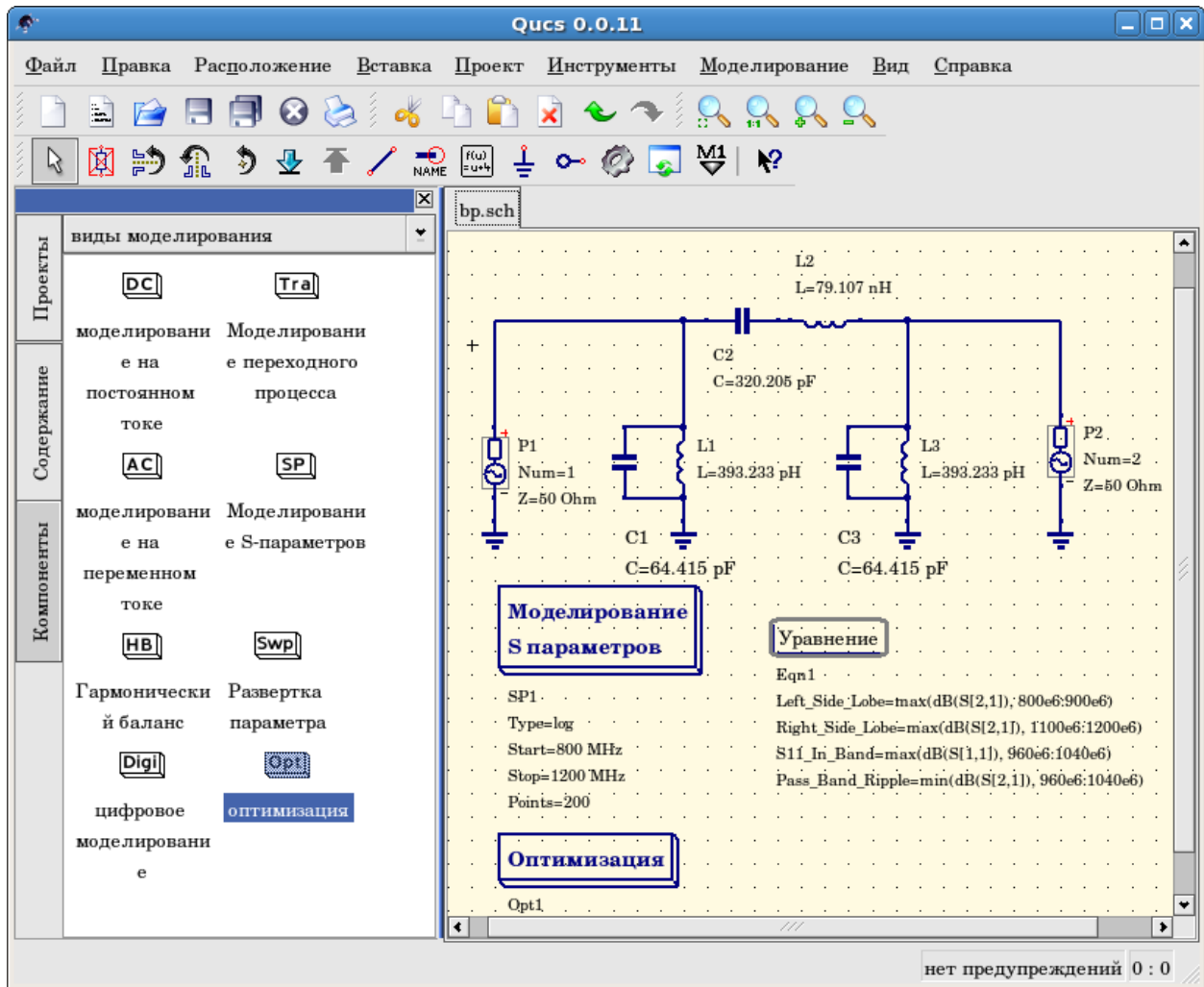


Рис. 2 - Подготовленная схема.

Теперь откройте компонент оптимизации и выберите вкладку 'Алгоритм'. Из имеющихся тут параметров особое внимание должно быть уделено параметрам 'Максимальное число итераций', 'Постоянная F' и 'Фактор пересечения'. Завышение или занижение этих параметров может привести к преждевременной сходимости оптимизатора, к локальному минимуму или к очень долгой оптимизации.

Изменение свойств оптимизации			
Общее	Алгоритм	Переменные	Цели
Метод:	DE/rand-to-best/1/exp		
Максимальное число итераций:	50		
Цикл обновления выхода:	2		
Число источников:	20		
Постоянная F:	0.85		
Фактор пересечения:	1		
Начальное число для генератора псевдослучайных чисел:	3		
Минимальный разброс стоимости:	1e-6		
Стоимостные цели:	10		
Стоимостные ограничения:	100		
ОК		Применить	
Отменить			

Рис. 3 - Диалог оптимизации, параметры алгоритма.

На вкладке “Переменные” определяется, какие элементы цепи будут выбраны и диапазоны их допустимых значений (рис. 4). Имена переменных соответствуют идентификаторам, помещенным в свойствах компонентов, а не именам компонентов.

Изменение свойств оптимизации

Общее Алгоритм Переменные Цели

Имя	активно	начальное	мин	макс	Тип
L3	да	393.2e-12	350e-12	450e-12	линейное вещественное
C3	да	64.4e-12	50e-12	80e-12	линейное вещественное
C2	да	320.2e-15	300e-15	340e-15	линейное вещественное
L2	да	79.1e-9	60e-9	100e-9	линейное вещественное
L1	да	393.2e-12	350e-12	450e-12	линейное вещественное
C1	да	64.4e-12	50e-12	80e-12	линейное вещественное

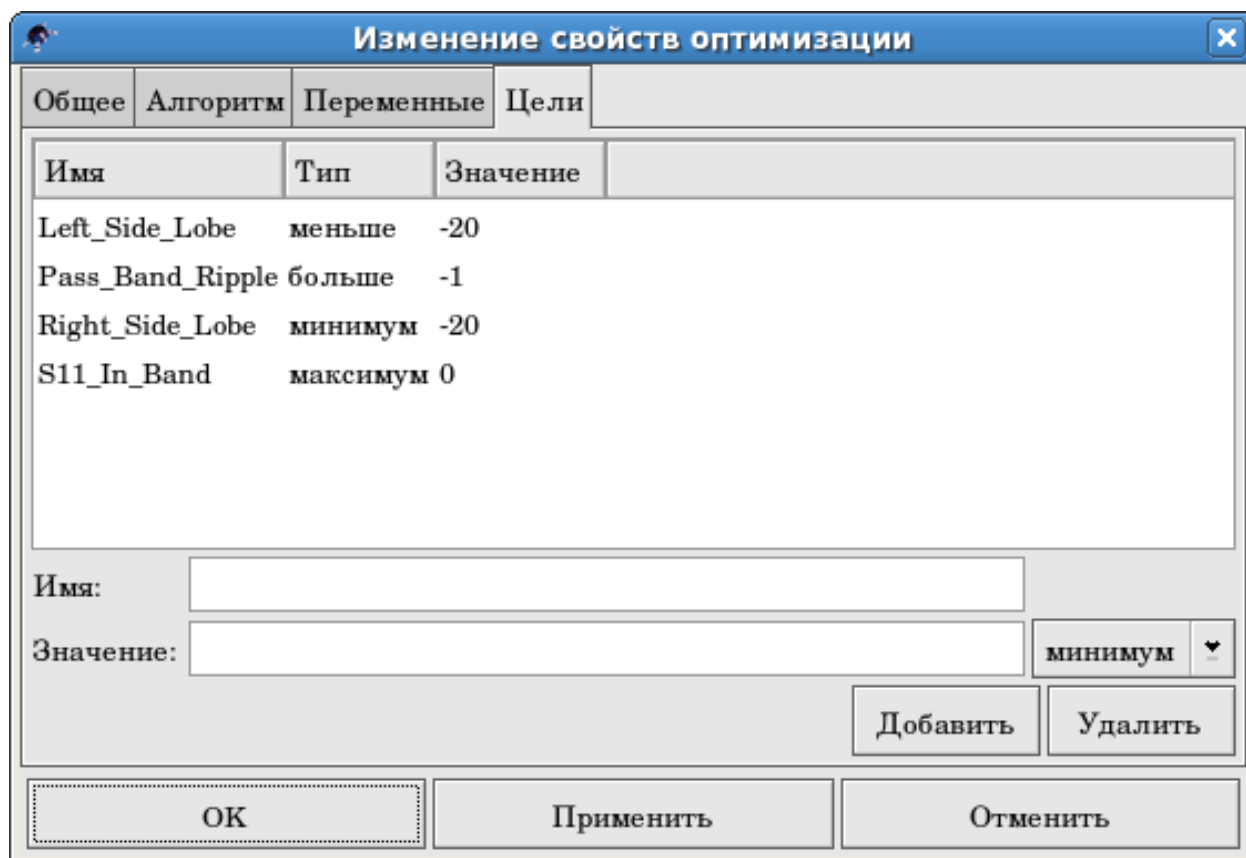
Имя: ☒ активно

начальное: мин: макс:

Тип: ▼

Рис. 4 - Диалог оптимизации, параметры переменных.

Наконец, переходите к “Целям”, где задаются цели оптимизации (сделать максимальным, сделать минимальным) и ограничения (меньше, больше, равно). Затем ASCO автоматически объединяет их в одну функцию стоимости, которая делается минимальной.



Имя	Тип	Значение
Left_Side_Lobe	меньше	-20
Pass_Band_Ripple	больше	-1
Right_Side_Lobe	минимум	-20
S11_In_Band	максимум	0

Имя:

Значение: минимум ▼

Добавить Удалить

ОК Применить Отменить

Рис. 5 - Диалог оптимизации, параметры целей.

Следующий шаг состоит в изменении схемы и определении, какие элементы требуется оптимизировать. Полученная в результате схема изображена на рис. 6.

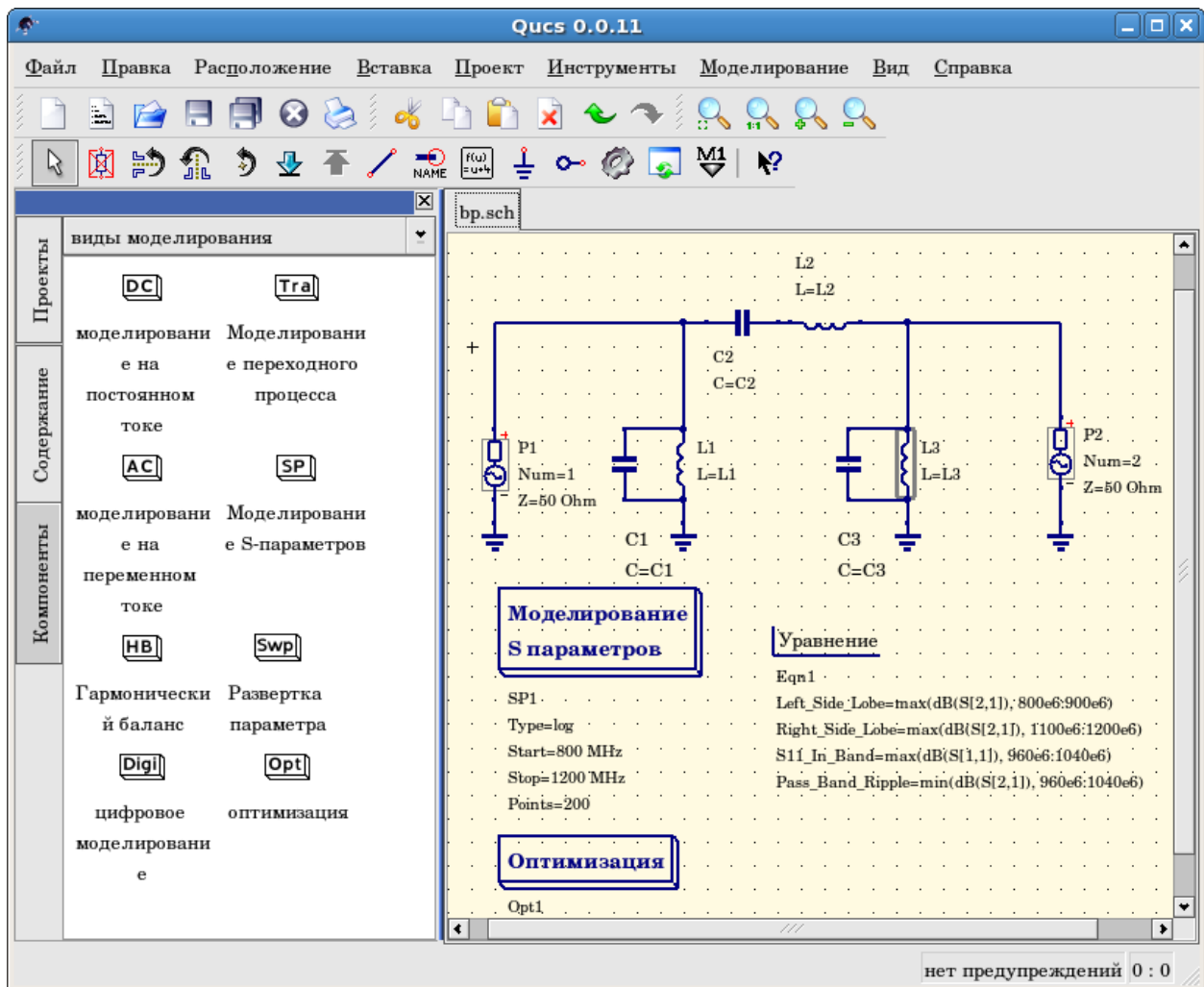


Рис. 6 - Новое главное окно Qucs.

Последний шаг - запуск оптимизации, то есть моделирования, нажатием клавиши F2. По окончании работы, на которую на современном компьютере уйдет всего несколько секунд, наилучшие результаты моделирования будут показаны в графическом виде.

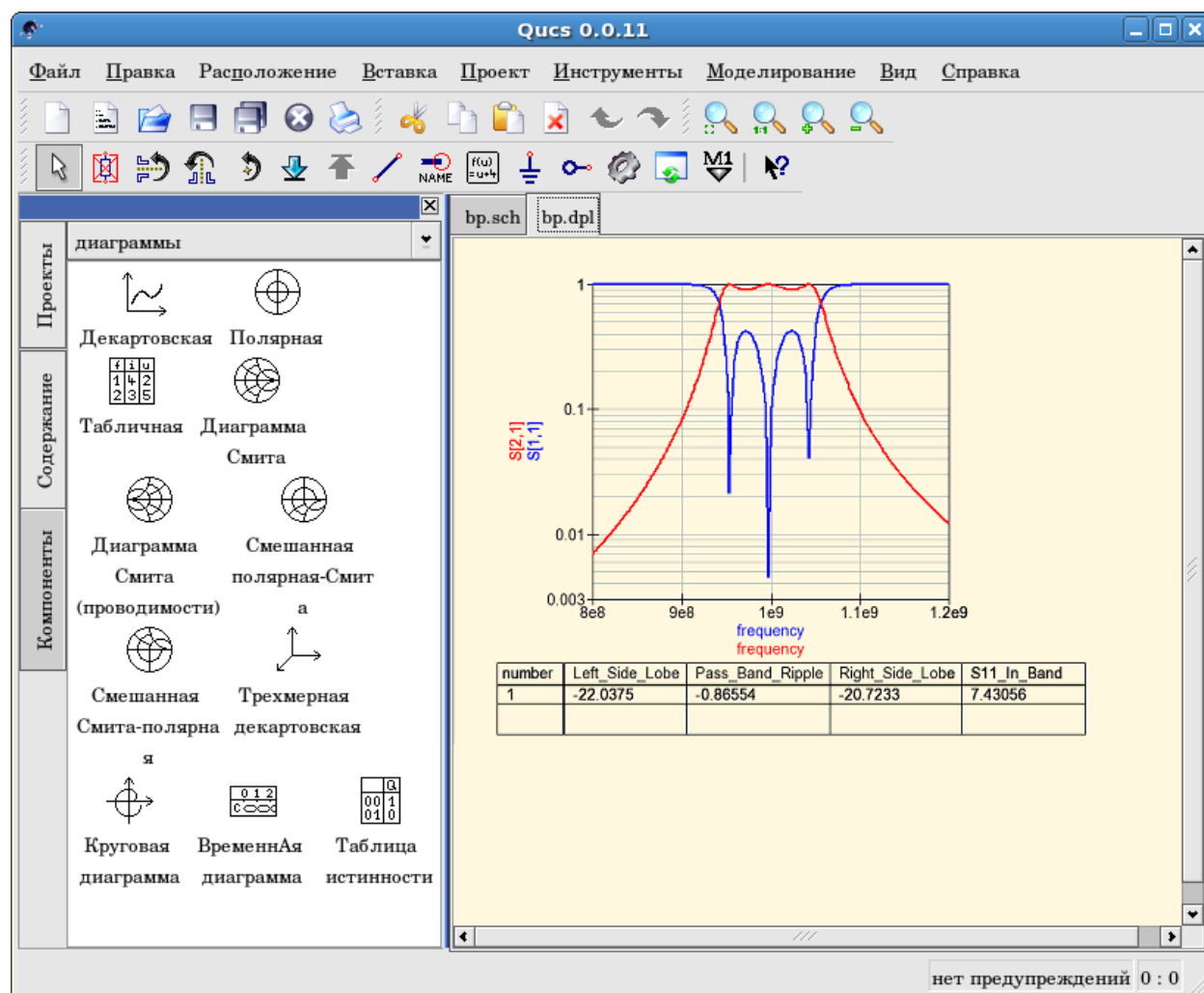


Рис. 7 - Окно Qucs с результатами.

Оптимальные параметры цепи могут быть найдены в диалоге оптимизации, на вкладке "Переменные". Теперь они являются начальными значениями каждой из представленных переменных (рис. 8).

Name	active	initial	min	max	Type
L3	yes	3.93e-10	350e-12	450e-12	linear double
C3	yes	6.52e-11	50e-12	80e-12	linear double
C2	yes	3.04e-13	300e-15	340e-15	linear double
L2	yes	8.4e-08	60e-9	100e-9	linear double
L1	yes	4.31e-10	350e-12	450e-12	linear double
C1	yes	5.92e-11	50e-12	80e-12	linear double

Name: ☒ active

initial: min: max:

Type:

Рис. 8 - Лучшие найденные параметры цепи.

By clicking the “Copy current values to equation” button, an equation component defining all the optimization variables with the values of the “initial” column will be copied to the clipboard and can be pasted to the schematic after closing the optimization dialog. The resulting schematic will be as shown in the next figure.

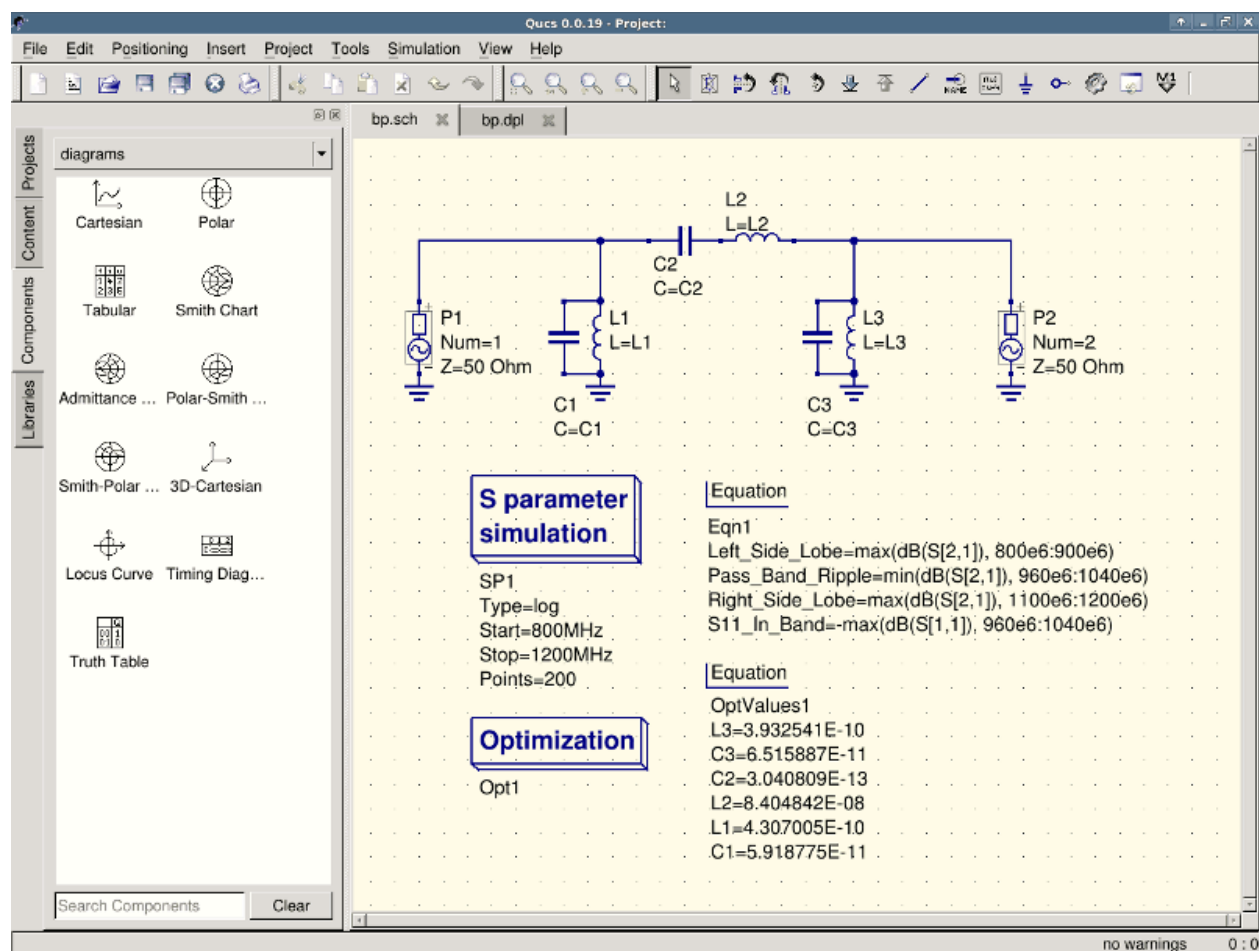


Рис. 9 - Схема с оптимизированными значениями.

в случае, если вам нужно сделать дополнительные изменения в схеме, компонент оптимизации теперь может быть отключен и будут использоваться оптимизированные значения из вставленного уравнения.

Вы можете изменить количество фигур, показанных для оптимизированных значений в диалоге оптимизации, щёлкнув правой кнопкой мыши на “начальном” заголовке таблицы и выбрав меню “Установить точность”, как показано на следующем рисунке.

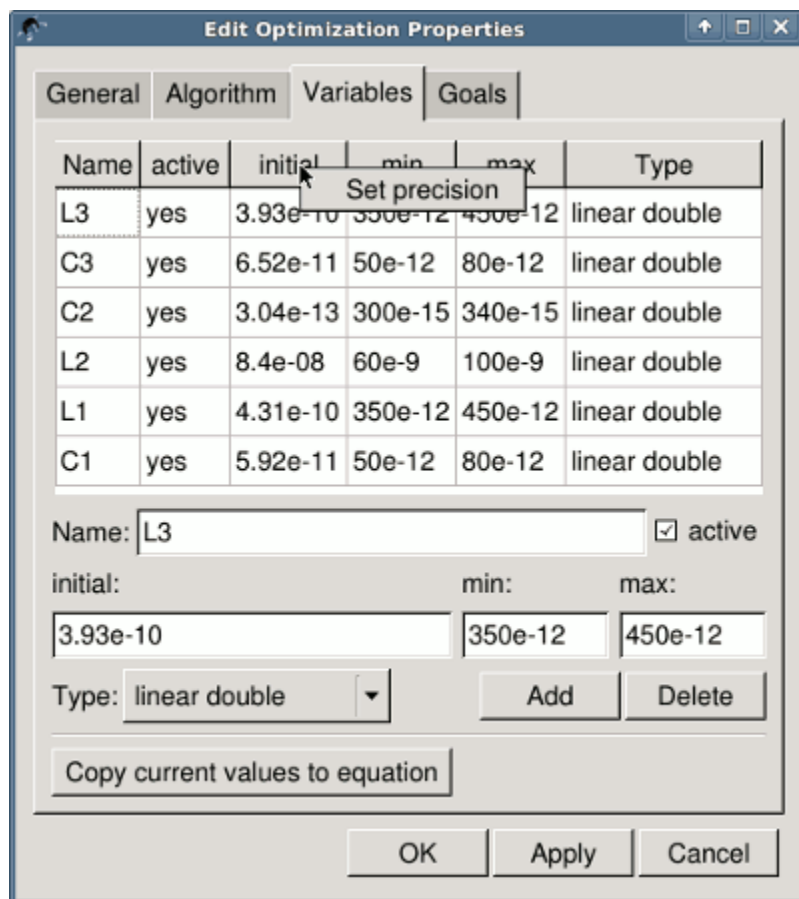


Рис. 10 - Изменение точности отображаемых переменных.

Getting Started with Octave Scripts

Qucs can also be used to develop Octave scripts (see <http://www.octave.org>). This document should give you a short description on how to do this.

If the user creates a new text document and saves it with the Octave extension, e.g. ‘name.m’ then the file will be listed at the Octave files of the active project. The script can be executed with F2 key or by pressing the simulate button in the toolbar. The output can be seen in the Octave window that opens automatically (per default on the right-hand side). At the bottom of the Octave window there is a command line where the user can enter single commands. It has a history function that can be used with the cursor up/down keys.

There are two Octave functions that load Qucs simulation results from a dataset file: `loadQucsVariable()` and `loadQucsDataset()`. Please use the help function in the Octave command line to learn more about them (i.e. type `help loadQucsVariable` and `help loadQucsDataset`).

4.1 Postprocessing

Octave can also be used for automatic postprocessing of a Qucs simulation result. This is done by editing the data display file of a schematic (Document Settings... in File menu). If the filename of an Octave script (filename extension m) from the same project is entered, this script will be executed after the simulation is finished.

Краткое описание действий

5.1 Действия общего назначения

(действуют во всех режимах)

колесо мыши	Прокручивает область рисования по вертикали. Можно также прокручивать за пределы текущего размера.
колесо мыши + клавиша Shift	Прокручивает область рисования по горизонтали. Можно также прокручивать за пределы текущего размера.
колесо мыши + клавиша Ctrl	Увеличивает или уменьшает масштаб области рисования.
перетаскивание файла в документную область	Пытается открыть файл как схему Qucs или как документ показа данных.

5.2 Режим “Выделения”



(Меню: Правка->Выделить)

левая кнопка мыши	Выделяет элемент, находящийся под курсором мыши. Если там расположено несколько компонентов, то можно нажимать на кнопку несколько раз, чтобы выбрать нужный. Держа эту кнопку мыши нажатой, можно перемещать компонент, находящийся под курсором мыши, и все выделенные компоненты. Если нужно точно разместить компоненты, нажмите при движении клавишу CTRL, и сетка будет отключена. Если держать кнопку мыши нажатой без каких-либо элементов под курсором, то получится прямоугольник. После отпускания кнопки мыши все элементы, находящиеся внутри этого прямоугольника, будут выделены. Размеры выделенной диаграммы или рисунка могут быть изменены, если нажать левую кнопку мыши над одним из углов и двигать курсор, держа кнопку нажатой. Если нажать кнопку мыши на тексте компонента, то его можно непосредственно редактировать. Нажатие клавиши Enter переводит к следующему свойству. Если это свойство является списком выбора, то его можно изменить только с помощью клавиш управления курсором (стрелки вверх/вниз). Если нажать кнопку мыши на узле цепи, то произойдет вход в “режим проводника”.
левая кнопка мыши + клавиша Ctrl	Позволяет выделить более одного элемента, то есть, выделение одного элемента не снимает выделения с других. Нажатие кнопки на выделенном элементе приводит к снятию выделения. Этот режим также пригоден для выделения с помощью прямоугольника (см. предыдущий пункт).
правая кнопка мыши	Нажатие кнопки на проводнике выделяет одну прямую линию, а не весь проводник.
двойное нажатие правой кнопки мыши	Открывает диалог редактирования свойств элемента (метки проводников, параметры компонентов и т.д.).

5.3 Режим “Вставка компонента”

(Нажмите на компонент/диаграмму в левой области)

левая кнопка мыши	Поместить новый экземпляр компонента на схему.
правая кнопка мыши	Вращать компонент. (Не действует на диаграммы.)

5.4 Режим “Проводник”



(Меню: Вставка->Проводник)

левая кнопка мыши	Устанавливает начальную/конечную точку проводника.
правая кнопка мыши	Изменяет направление изгиба проводника (сперва налево/направо или сперва вверх/вниз).
двойное нажатие правой кнопки мыши	Заканчивает проводник, не находясь на проводнике или выводе.

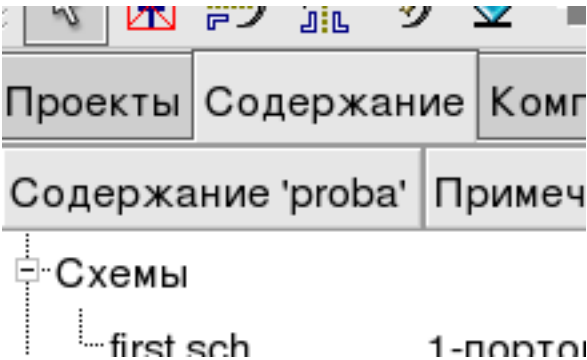
5.5 Режим “Вставка”



(Меню: Правка->Вставить)

левая кнопка мыши	Поместить элементы на схему (из буфера обмена).
правая кнопка мыши	Вращать элементы.

5.6 Мышь во вкладке “Содержание”



нажатие левой кнопки	Выделяет файл.	
двойное нажатие левой кнопки	Открывает файл.	
нажатие правой кнопки	Отображает меню с:	
	открыть	• открыть выделенный файл
	“переименовать”	• изменить имя выделенного файла
	“удалить”	• удалить выделенный файл
	“copy file”	• copy schematic file. Only file operations are performed. Dataset and display settings in schematic properties are kept untouched.

5.7 Клавиатура

Многие действия могут быть вызваны/сделаны с помощью клавиш клавиатуры. Выполняемые действия поясняются в строке статуса при выборе команды из главного меню. Некоторые дополнительные команды, выполняемые с помощью клавиш, приводятся в следующем списке:

“Delete” или “Backspace”	Удаляет выделенные элементы или входит в режим удаления, если ни один элемент не выделен.
Клавиши со стрелками влево/вправо	Изменяют положение выделенных маркеров на графиках. Если ни один маркер не выделен, перемещают выделенные элементы. Если ни один элемент не выделен, прокручивают область документа.
Клавиши со стрелками вверх/вниз	Изменяют положение выделенных маркеров на многомерных графиках. Если ни один маркер не выделен, перемещают выделенные элементы. Если ни один элемент не выделен, прокручивают область документа.
Клавиша Tab	Переходит к следующему открытому документа (в соответствие со вкладками).

Работа с подсхемами

Подсхемы используются, чтобы внести большую ясность в схему. Это очень полезно в больших схемах или в схемах, в которых какой-либо блок компонентов появляется несколько раз.

В Qucs каждая схема, содержащая вывод подсхемы, является подсхемой. Вывод подсхемы можно получить с помощью панели инструментов, списка компонентов (в дискретных компонентах) или меню (Вставка->Вставить вывод). После того, как вставлены все выводы подсхемы (например, два), нужно сохранить подсхему (например, нажав CTRL-S). Если посмотреть в просмотр содержимого проекта (рис. 1), можно увидеть, что теперь справа от имени схемы стоит “2-портовый” (столбец “Примечание”). Эта пометка есть у всех документов, которые являются подсхемами. Теперь перейдите в схему, где Вы хотите использовать эту подсхему. Затем нажмите на имя подсхемы (в просмотре содержимого). Снова войдя в область документов, Вы видите, что теперь можно поместить подсхему в главную схему. Сделайте так и закончите схему. Теперь можно выполнить моделирование. Результат будет такой же, как если бы все компоненты подсхемы были помещены непосредственно на схему.

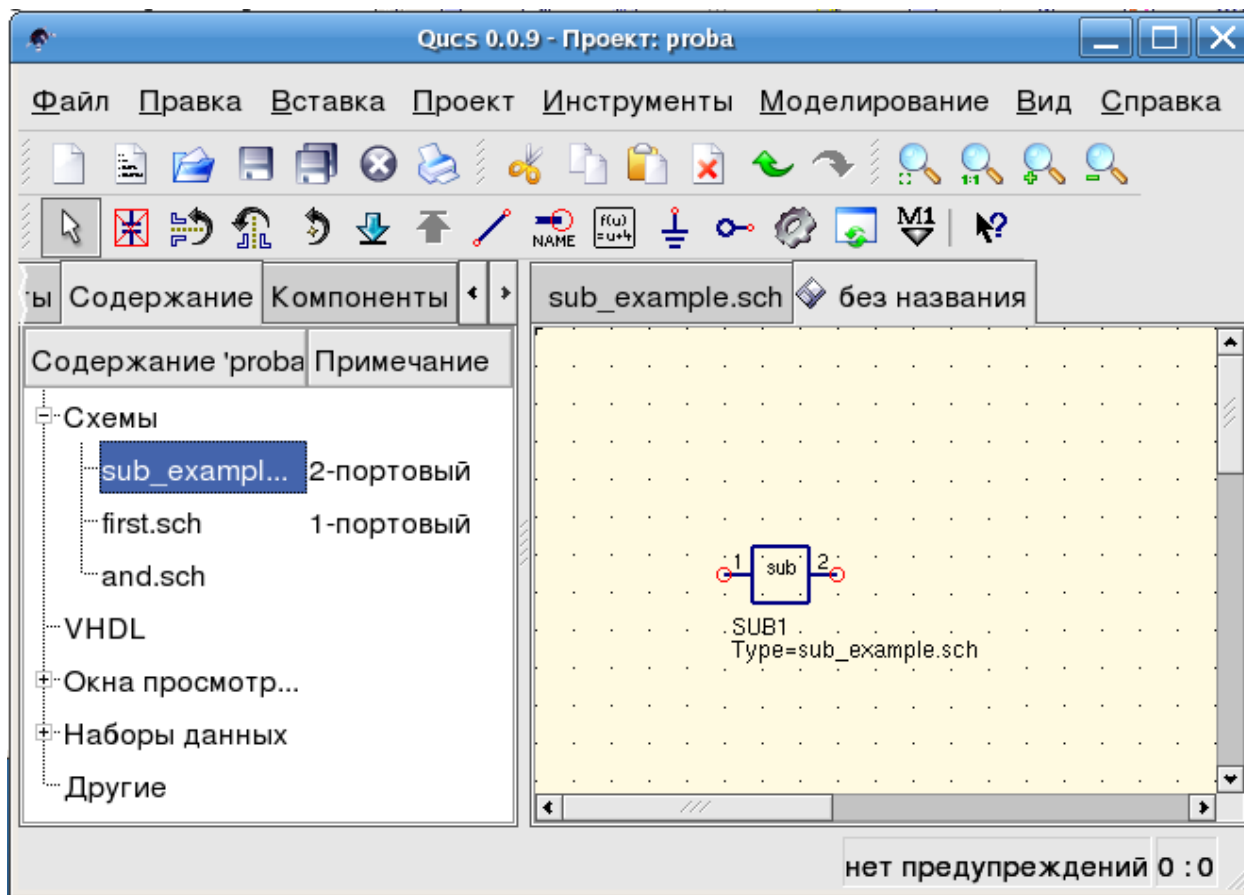


Рис. 1 - Получение доступа к подсхеме

Если выбрать компонент-подсхему (нажав на ее обозначение в схеме), то можно войти в подсхему, нажав CTRL-I (конечно, эта функция доступна через панель инструментов и через меню). Можно вернуться, нажав CTRL-H.

Если Вам не нравится обозначение компонента подсхемы, то Вы можете нарисовать свое собственное и поместить текст компонента туда, где Вам нравится. Просто сделайте схему подсхемы текущим документом и перейдите к меню: Файл->Изменить обозначение схемы. Если Вы уже не нарисовали обозначение для этой схемы, то автоматически будет создано простое обозначение. Это обозначение можно редактировать, рисуя линии и дуги. После завершения, сохраните его. Теперь поместите его на другую схему, и вот у Вас новое обозначение.

Как и у всех других компонентов, у подсхем могут быть параметры. Для создания своих собственных параметров, вернитесь в редактор, где вы редактировали обозначение подсхемы, и дважды нажмите левую кнопку на тексте параметра подсхемы. Появится диалоговое окно, в котором Вы можете заполнить параметры начальными значениями и описаниями. Когда Вы это закончите, закройте диалоговое окно и сохраните подсхему. Везде, где вставляется подсхема, у нее есть эти новые параметры, и их можно редактировать так же, как и во всех других компонентах.

6.1 Subcircuits with Parameters

A simple example using subcircuits with parameters and equations is provided here.

Create a subcircuit:

- Создайте новый проект
- New schematic (for subcircuit)
- Добавить резистор, катушку индуктивности, конденсатор, соедините их последовательно, добавьте два порта
- Сохраните подсхему как RLC.sch
- Give value of resistor as 'R1'
- Добавьте уравнение 'ind = L1',
- Give value of inductor as 'ind'
- Give value of capacitor as 'C1'
- Сохранить
- File > Edit Circuit Symbol
- Double click on the 'SUB File=name' tag under the rectangular box
 - Add name = R1, default value = 1
 - Add name = L1, default value = 1
 - Add name = C1, default value = 1
 - OK

Insert subcircuit and define parameters:

- New schematic (for testbench)
- Сохраните Test_RLC.sch
- Project Contents > pick and place the above RLC subcircuit
- Add AC voltage source (V1) and ground
- Add AC simulation, from 140Hz to 180Hz, 201 points
- Set on the subcircuit symbol
 - R1=1
 - L1=100e-3
 - C1=10e-6
- Моделировать
- Add a Cartesian diagram, plot V1.i
- The result should be the resonance of the RLC circuit.
- The parameters of the RLC subcircuit can be changed on the top schematic.

Быстрый старт в цифровом моделировании

Qucs - это также графический интерфейс пользователя для выполнения цифрового моделирования. Этот документ вкратце описывает, как для этого пользоваться Qucs.

For digital simulations Qucs uses the FreeHDL program (<http://www.freehdl.seul.org>). So the FreeHDL package as well as the GNU C++ compiler must be installed on the computer.

There is no big difference in running an analog or a digital simulation. So having read the [Getting Started for analog simulations](#), it is now easy to get a digital simulation work. Let us compute the truth table of a simple logical AND cell. Select the digital components in the combobox of the components tab on the left-hand side and build the circuit shown in figure 1. The digital simulation block can be found among the other simulation blocks.

Цифровые источники S1 и S2 подключены ко входам, узел с меткой Output является выходом. После выполнения моделирования открывается страница показа данных. Поместите на нее диаграмму Таблица истинности и вставьте переменную Output. Теперь показывается таблица истинности двух-входного элемента И. Поздравления, первое цифровое моделирование сделано!

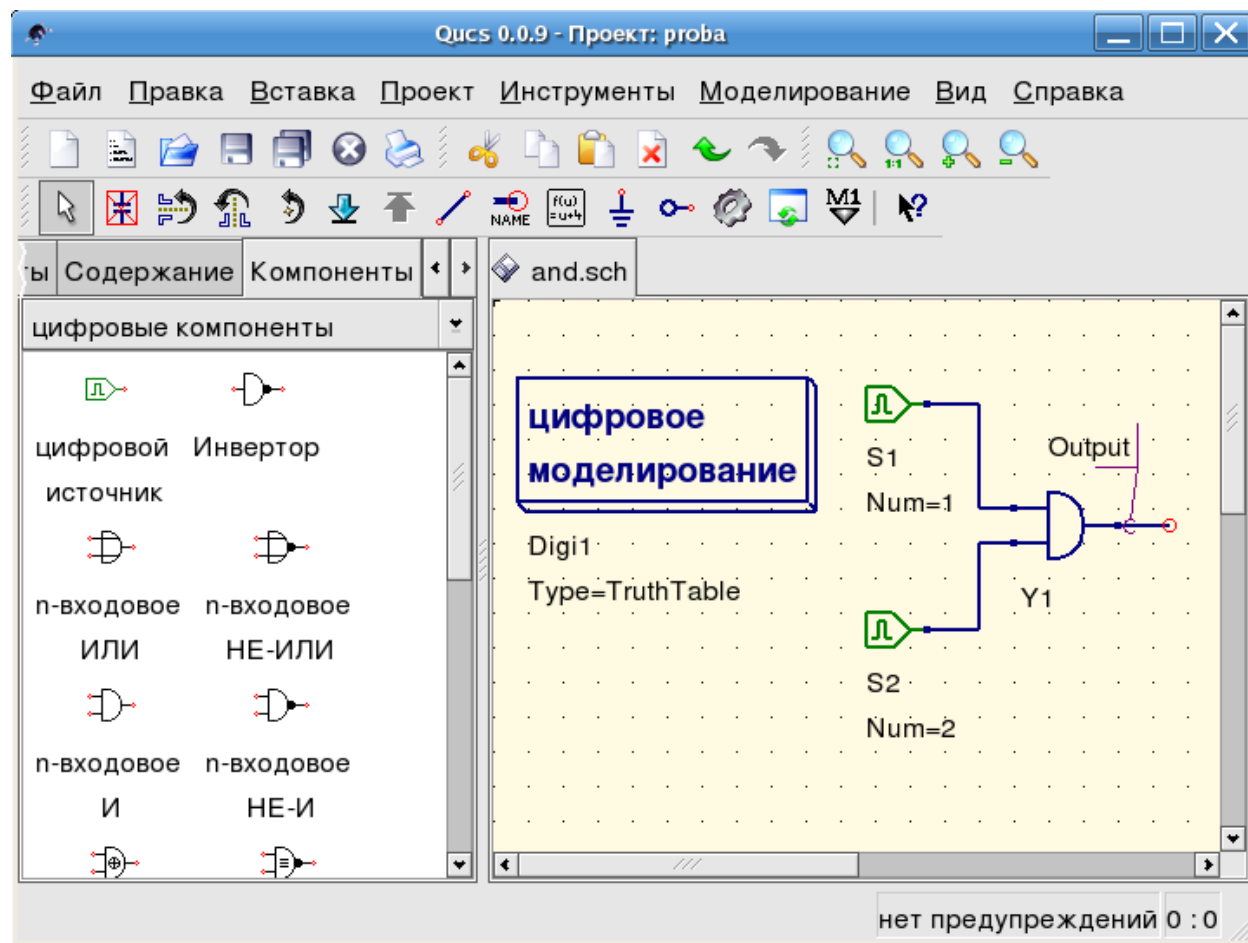


Рис. 1 - Главное окно Qucs

Таблица истинности - не единственный вид моделирования, который может выполнить Qucs. Возможно также подать в схему случайный сигнал и посмотреть выходной сигнал во временной диаграмме. Чтобы это сделать, нужно изменить параметр Type блока моделирования на TimeList и в следующем параметре нужно ввести продолжительность моделирования. Теперь у цифровых источников другой смысл: они могут выдавать случайную последовательность битов, для чего им нужно задать первый бит (низкий или высокий) и список моментов времени следующей смены состояния. Обратите внимание, что этот список повторяется после конца. Поэтому, чтобы получить тактовые импульсы с частотой 1 ГГц и скважностью 1:1, в этом списке должно быть записано: 0.5ns; 0.5ns.

Для отображения результатов этого типа моделирования есть временная диаграмма. В ней результаты всех выходных сигналов могут быть изображены построчно в одной диаграмме. Так что успехов в этом деле...

7.1 Файловый компонент VHDL

Более сложные и более универсальные виды моделирования могут быть выполнены с помощью компонента "файл VHDL". Этот компонент может быть взят из списка компонентов (раздел "цифровые компоненты"). Тем не менее, рекомендуется следующий способ: файл VHDL должен быть в составе проекта. Затем перейдите в просмотр содержимого проекта и нажмите на имя файла. Войдя в область построения схем, поместите компонент VHDL.

Последний объектный блок в файле VHDL определяет интерфейс, то есть здесь должны быть объявлены все входные и выходные выводы. Эти выводы показываются также на схемном обозначении и могут быть соединены с остальной схемой. Во время моделирования исходный код файла VHDL помещается в VHDL-файл верхнего уровня. Это следует учитывать, поскольку это приводит к некоторым ограничениям. Например, имена объектов в VHDL-файле должны отличаться от имен, уже данных подсхемам. (После моделирования полный исходный код можно увидеть, нажав F6. Пользуйтесь этим, чтобы прочувствовать процедуру.)

Краткое описание математических функций

В уравнениях Qucs могут применяться следующие операции и функции. Подробное описание смотрите в “Справочном руководстве по выражениям”. Параметры в квадратных скобках “[]” необязательны.

8.1 Операторы

8.1.1 Арифметические операторы

$+x$	Унарный плюс
$-x$	Унарный минус
$x+y$	Сложение
$x-y$	Вычитание
$x*y$	Умножение
x/y	Деление
$x\%y$	Остаток от деления
x^y	Возведение в степень

8.1.2 Логические операторы

$!x$	Отрицание
$x\&\&y$	И
$x y$	ИЛИ
$x\wedge y$	Исключающее ИЛИ
$x?y:z$	Сокращение для условного выражения - if x then y else z
$x==y$	Равно
$x!=y$	Не равно
$x<y$	Меньше
$x<=y$	Меньше или равно
$x>y$	Больше
$x>=y$	Больше или равно

8.2 Математические функции

8.2.1 Векторы и матрицы: создание

<code>eye(n)</code>	Создает единичную матрицу $n \times n$
<code>length(y)</code>	Возвращает длину вектора <code>y</code>
<code>linspace(from, to, n)</code>	Вещественный вектор с <code>n</code> линейно расставленными компонентами между <code>from</code> и <code>to</code>
<code>logspace(from, to, n)</code>	Вещественный вектор с <code>n</code> логарифмически расставленными компонентами между <code>from</code> и <code>to</code>

8.2.2 Векторы и матрицы: основные матричные функции

<code>adjoint(x)</code>	Сопряженная с <code>x</code> матрица (транспонированная и комплексно-сопряженная)
<code>det(x)</code>	Детерминант матрицы <code>x</code>
<code>inverse(x)</code>	Инверсия матрицы <code>x</code>
<code>transpose(x)</code>	Транспонированная матрица для <code>x</code> (столбцы и строки меняются местами)

8.2.3 Элементарные математические функции: основные вещественные и комплексные функции

<code>abs(x)</code>	Абсолютное значение, модуль комплексного числа
<code>angle(x)</code>	Фазовый угол (в радианах) комплексного числа. Синоним <code>arg()</code>
<code>arg(x)</code>	Фазовый угол (в радианах) комплексного числа
<code>conj(x)</code>	Сопряжение комплексного числа
<code>deg2rad(x)</code>	Преобразует фазу из градусов в радианы
<code>hypot(x, y)</code>	Функция евклидова расстояния
<code>imag(x)</code>	Мнимая часть комплексного числа
<code>mag(x)</code>	Модуль комплексного числа
<code>norm(x)</code>	Квадрат абсолютного значения вектора
<code>phase(x)</code>	Фазовый угол (в градусах) комплексного числа
<code>polar(m, p)</code>	Преобразование полярных координат <code>m</code> и <code>p</code> в комплексное число
<code>rad2deg(x)</code>	Преобразует фазу из радианов в градусы
<code>real(x)</code>	Вещественная часть комплексного числа
<code>sign(x)</code>	Знаковая функция
<code>sqr(x)</code>	Квадрат (вторая степень) числа
<code>sqrt(x)</code>	Квадратный корень
<code>unwrap(p[,tol[,step]])</code>	Разворачивает угол <code>p</code> (радианы) – по умолчанию <code>step = 2pi</code> , <code>tol = pi</code>

8.2.4 Элементарные математические функции: экспоненциальные и логарифмические функции

<code>exp(x)</code>	Экспоненциальная функция с основанием <code>e</code>
<code>limexp(x)</code>	Ограниченная экспоненциальная функция
<code>log10(x)</code>	Десятичный логарифм
<code>log2(x)</code>	Двоичный логарифм
<code>ln(x)</code>	Натуральный логарифм (основание <code>e</code>)

8.2.5 Элементарные математические функции: тригонометрия

$\cos(x)$	Косинус
$\operatorname{cosec}(x)$	Косеканс
$\cot(x)$	Котангенс
$\sec(x)$	Секанс
$\sin(x)$	Синус
$\tan(x)$	Тангенс

8.2.6 Элементарные математические функции: обратные тригонометрические функции

$\arccos(x)$	Арккосинус (известный также как “обратный косинус”)
$\operatorname{arccosec}(x)$	Арккосеканс
$\operatorname{arccot}(x)$	Арккотангенс
$\operatorname{arcsec}(x)$	Арксеканс
$\arcsin(x)$	Арсинус (известный также как “обратный синус”)
$\arctan(x[,y])$	Арктангенс (известный также как “обратный тангенс”)

8.2.7 Элементарные математические функции: гиперболические функции

$\cosh(x)$	Гиперболический косинус
$\operatorname{cosech}(x)$	Гиперболический косеканс
$\coth(x)$	Гиперболический котангенс
$\operatorname{sech}(x)$	Гиперболический секанс
$\sinh(x)$	Гиперболический синус
$\tanh(x)$	Гиперболический тангенс

8.2.8 Элементарные математические функции: обратные гиперболические функции

$\operatorname{arcosh}(x)$	Гиперболический ареакосинус
$\operatorname{arcosech}(x)$	Гиперболический ареакосеканс
$\operatorname{arcoth}(x)$	Гиперболический ареакотангенс
$\operatorname{arsech}(x)$	Гиперболический ареасеканс
$\operatorname{arsinh}(x)$	Гиперболический ареасинус
$\operatorname{artanh}(x)$	Гиперболический ареатангенс

8.2.9 Элементарные математические функции: округление

$\operatorname{ceil}(x)$	Округление до ближайшего большего целого
$\operatorname{fix}(x)$	Отбрасывание десятичных разрядов вещественного числа
$\operatorname{floor}(x)$	Округление до ближайшего меньшего целого
$\operatorname{round}(x)$	Округление до ближайшего целого

8.2.10 Элементарные математические функции: специальные математические функции

besseli0(x)	Модифицированная функция Бесселя нулевого порядка
besselj(n,x)	Функция Бесселя первого рода и n-го порядка
bessely(n,x)	Функция Бесселя второго рода и n-го порядка
erf(x)	Функция ошибки
erfc(x)	Комплиментарная функция ошибки
erfinv(x)	Обратная функция ошибки
erfcinv(x)	Обратная комплиментарная функции ошибки
sinc(x)	Sinc-функция ($\sin(x)/x$ или 1 при $x = 0$)
step(x)	Шаговая функция

8.2.11 Анализ данных: основы статистики

avg(x[, range])	Среднее значение вектора x. Если задан диапазон, то в x должна быть однозначная зависимость данных
cumavg(x)	Накопительное среднее элементов вектора
max(x,y)	Возвращает большее из значений x и y
max(x[, range])	Максимум вектора x. Если задан диапазон, то в x должна быть однозначная зависимость данных
min(x,y)	Возвращает меньшее из значений x и y
min(x[, range])	Минимум вектора x. Если задан диапазон, то в x должна быть однозначная зависимость данных
rms(x)	Среднеквадратичное значение элементов вектора
runavg(x)	Скользящее среднее элементов вектора
stddev(x)	Стандартное отклонение элементов вектора
variance(x)	Дисперсия элементов вектора
random()	Случайное число между 0.0 и 1.0
srandom(x)	Устанавливает начальное число для random

8.2.12 Анализ данных: основные операции

cumprod(x)	Накопительное произведение элементов вектора
cumsum(x)	Накопительная сумма элементов вектора
interpolate(f,x[,n])	Spline interpolation of vector f using n equidistant points of x
prod(x)	Произведение элементов вектора
sum(x)	Сумма элементов вектора
xvalue(f,yval)	Returns x-value nearest to yval in single dependency vector f
yvalue(f,xval)	Returns y-value nearest to xval in single dependency vector f

8.2.13 Анализ данных: дифференцирование и интегрирование

ddx(expr,var)	Дифференцирует математическое выражение expr по переменной var
diff(y,x[,n])	Дифференцировать вектор y по вектору x n раз. По умолчанию n = 1
integrate(x,h)	Численно интегрировать вектор x, считая, что размер шага h постоянный

8.2.14 Анализ данных: обработка сигналов

dft(x)	Дискретное преобразование Фурье для вектора x
fft(x)	Быстрое преобразование Фурье для вектора x
fftshift(x)	Shuffles the FFT values of vector x to move DC to the center of the vector
Freq2Time(V, f)	Обратное дискретное преобразование Фурье функции V(f) с физической интерпретацией
idft(x)	Обратное дискретное преобразование Фурье для вектора x
ifft(x)	Обратное быстрое преобразование Фурье для вектора x
kbd(x[,n])	Производное окно Кайзера-Бесселя
Time2Freq(v, t)	Дискретное преобразование Фурье функции v(t) с физической интерпретацией

8.3 Функции из электроники

8.3.1 Преобразование единиц измерения

dB(x)	Значение в дБ
dbm(x)	Преобразовать напряжение в мощность в дБм
dbm2w(x)	Преобразовать мощность в дБм в мощность в ваттах
w2dbm(x)	Преобразовать мощность в ваттах в мощность в дБм
vt(t)	Температурный потенциал для данной температуры t в градусах Кельвина

8.3.2 Коэффициенты отражения и КСВН

rtoswr(x)	Преобразует коэффициент отражения в коэффициент стоячей волны по напряжению (КСВН)
rtoz(x[, zref])	Преобразует коэффициент отражения в полную проводимость; по умолчанию zref равно 50 Ом
rtoz(x[, zref])	Преобразует коэффициент отражения в полное сопротивление; по умолчанию zref равно 50 Ом
ytor(x[, zref])	Преобразует полную проводимость в коэффициент отражения; по умолчанию zref равно 50 Ом
ztor(x[, zref])	Преобразует полное сопротивление в коэффициент отражения; по умолчанию zref равно 50 Ом

8.3.3 N-портовые матричные преобразования

stos(s, zref[, z0])	Преобразует матрицу S-параметров в матрицу S-параметров с другим Z0
stoy(s[, zref])	Преобразует матрицу S-параметров в матрицу Y-параметров
stoz(s[, zref])	Преобразует матрицу S-параметров в матрицу Z-параметров
twoport(m, from, to)	Преобразует двух-портовую матрицу: from и to - это 'Y', 'Z', 'H', 'G', 'A', 'S' и 'T'.
ytoz(y[, z0])	Преобразует матрицу Y-параметров в матрицу Z-параметров
ytoz(y)	Преобразует матрицу Y-параметров в матрицу Z-параметров
ztoz(z[, z0])	Преобразует матрицу Z-параметров в матрицу S-параметров
ztoz(z)	Преобразует матрицу Z-параметров в матрицу Y-параметров

8.3.4 Усилители

GaCircle(s,Ga[,arcs])	Доступный коэффициент усиления мощности Ga окружностей (плоскость источника)
GpCircle(s,Gp[,arcs])	Операционный коэффициент усиления мощности Gp окружностей (плоскость нагрузки)
Mu(s)	Mu - фактор устойчивости двухпортовой матрицы S-параметров
Mu2(s)	Mu' - фактор устойчивости двухпортовой матрицы S-параметров
NoiseCircle(Sopt,Fmin,Rn,F[,Arcs])	Окружности диаграммы шума(ов) F
PlotVs(data,dep)	Возвращает данные, выбранные из data: зависимость dep
Rollet(s)	Фактор устойчивости Роллета для двухпортовой матрицы S-параметров
StabCircleL(s[,arcs])	Окружность устойчивости в плоскости нагрузки
StabCircleS(s[,arcs])	Окружность устойчивости в плоскости источника
StabFactor(s)	Фактор устойчивости двухпортовой матрицы S-параметров
StabMeasure(s)	Мера устойчивости B1 двухпортовой матрицы S-параметров

8.4 Номенклатура

8.4.1 Интервалы

L0:HI	Интервал от LO до HI
:HI	Вплоть до HI
L0:	От LO
:	Нет ограничений интервала

8.4.2 Матрицы и элементы матриц

M	Вся матрица M
M[2,3]	Элемент, находящийся во 2-м ряду и 3-м столбце матрицы M
M[:,3]	Вектор, состоящий из 3-го столбца матрицы M

8.4.3 Непосредственно

2.5	Вещественное число
1.4+j5.1	Комплексное число
[1,3,5,7]	Вектор
[11,12;21,22]	Матрица

8.4.4 Приставки к единицам

E	экса, 1e+18
P	пета, 1e+15
T	тера, 1e+12
G	гига, 1e+9
M	мега, 1e+6
k	кило, 1e+3
m	милли, 1e-3
u	микро, 1e-6
n	нано, 1e-9
p	пико, 1e-12
f	фемто, 1e-15
a	атто, 1e-18

8.4.5 Названия величин

S[1,1]	Значение S-параметра
<i>nodename.V</i>	Напряжение постоянного тока в узле <i>nodename</i>
<i>name.I</i>	Постоянный ток через компонент <i>name</i>
<i>nodename.v</i>	Переменное напряжение в узле <i>nodename</i>
<i>name.i</i>	Переменный ток через компонент <i>name</i>
<i>nodename.vn</i>	Шумовое напряжение переменного тока в узле <i>nodename</i>
<i>name.in</i>	Шумовой переменный ток через компонент <i>name</i>
<i>nodename.Vt</i>	Переходное напряжение в узле <i>nodename</i>
<i>name.It</i>	Переходной ток через компонент <i>name</i>

Примечание: Все напряжения и токи - пиковые значения. Примечание: Шумовые напряжения - СКЗ значения в полосе частот 1 Hz.

8.5 Константы

i, j	Мнимая единица (“квадратный корень из -1”)
pi	$4 \cdot \arctan(1) = 3.14159\dots$
e	$e = 2.71828\dots$
kB	Постоянная Больцмана = $1.38065e-23$ Дж/К
q	Элементарный заряд = $1.6021765e-19$ Кл

Перечень специальных символов

В компоненте “Текст” и в тексте меток осей диаграмм можно использовать специальные символы. Это делается с помощью тэгов LaTeX. В следующей таблице приводится перечень символов, имеющихся в настоящее время.

Примечание: Правильное отображение этих символов зависит от шрифта, используемого Qucs!

Строчные греческие буквы

Тэг LaTeX	Юникод	Описание
<code>\alpha</code>	0x03B1	alpha
<code>\beta</code>	0x03B2	beta
<code>\gamma</code>	0x03B3	gamma
<code>\delta</code>	0x03B4	delta
<code>\epsilon</code>	0x03B5	epsilon
<code>\zeta</code>	0x03B6	zeta
<code>\eta</code>	0x03B7	eta
<code>\theta</code>	0x03B8	theta
<code>\iota</code>	0x03B9	iota
<code>\kappa</code>	0x03BA	kappa
<code>\lambda</code>	0x03BB	lambda
<code>\mu</code>	0x03BC	mu
<code>\textmu</code>	0x00B5	mu
<code>\nu</code>	0x03BD	nu
<code>\xi</code>	0x03BE	xi
<code>\pi</code>	0x03C0	pi
<code>\varpi</code>	0x03D6	pi
<code>\rho</code>	0x03C1	rho
<code>\varrho</code>	0x03F1	rho
<code>\sigma</code>	0x03C3	sigma
<code>\tau</code>	0x03C4	tau
<code>\upsilon</code>	0x03C5	upsilon
<code>\phi</code>	0x03C6	phi
<code>\chi</code>	0x03C7	chi
<code>\psi</code>	0x03C8	psi
<code>\omega</code>	0x03C9	omega

Прописные греческие буквы

Тэг LaTeX	Юникод	Описание
<code>\Gamma</code>	0x0393	Gamma
<code>\Delta</code>	0x0394	Delta
<code>\Theta</code>	0x0398	Theta
<code>\Lambda</code>	0x039B	Lambda
<code>\Xi</code>	0x039E	Xi
<code>\Pi</code>	0x03A0	Pi
<code>\Sigma</code>	0x03A3	Sigma
<code>\Upsilon</code>	0x03A5	Upsilon
<code>\Phi</code>	0x03A6	Phi
<code>\Psi</code>	0x03A8	Psi
<code>\Omega</code>	0x03A9	Omega

Математические символы

Тэг LaTeX	Юникод	Описание
<code>\cdot</code>	0x00B7	знак умножения - точка (центрированная точка)
<code>\times</code>	0x00D7	знак умножения - крестик
<code>\pm</code>	0x00B1	знак плюс минус
<code>\mp</code>	0x2213	знак минус плюс
<code>\partial</code>	0x2202	знак частного дифференцирования
<code>\nabla</code>	0x2207	набла-оператор
<code>\infty</code>	0x221E	знак бесконечности
<code>\int</code>	0x222B	знак интеграла
<code>\approx</code>	0x2248	символ приближения (волнистый знак равенства)
<code>\neq</code>	0x2260	знак не равно
<code>\in</code>	0x220A	символ “содержится в”
<code>\leq</code>	0x2264	знак меньше-равно
<code>\geq</code>	0x2265	знак больше-равно
<code>\sim</code>	0x223C	(центрально-европейский) знак пропорциональности
<code>\propto</code>	0x221D	(американский) знак пропорциональности
<code>\diameter</code>	0x00F8	знак диаметра (также знак среднего)
<code>\onehalf</code>	0x00BD	половина
<code>\onequarter</code>	0x00BC	четверть
<code>\twosuperior</code>	0x00B2	квадрат (степень 2)
<code>\threesuperior</code>	0x00B3	степень 3
<code>\ohm</code>	0x03A9	единица для сопротивления (прописная греческая омега)

Согласование цепей

Создание согласованных цепей часто требуется в микроволновой технологии. Qucs может делать это автоматически. Необходимые шаги для этого:

Выполнить моделирование S-параметров, чтобы рассчитать коэффициент отражения.

Вставить диаграмму, чтобы показать коэффициент отражения (то есть, $S[1,1]$ для порта 1, $S[2,2]$ для порта 2 и т.д.)

Поместить на график маркер и двигаться шагами к требуемой частоте.

Нажать правую кнопку мыши на маркере и выбрать “согласование мощности” в появившемся меню.

Появляется диалоговое окно, в котором можно настроить значения, например, опорное полное сопротивление может быть выбрано отличающимся от 50 Ом.

После нажатия на кнопку “создать” происходит возврат к схеме, и с помощью курсора мыши может быть выбрано место для вставки согласованной цепи.

Левая сторона согласованной цепи является входом, а правая сторона должна быть подсоединена к схеме.

Если маркер указывает на переменную под названием “Sopt”, в меню появляется опция “согласование шума”. Обратите внимание, что единственным отличием от “согласования мощности” является то, что берется сопряженный комплексный коэффициент отражения. Поэтому, если у переменной другое имя, согласование шума может быть выбрано путем перенастройки значений в диалоговом окне.

Диалог создания согласования может быть также вызван с помощью меню (Инструменты->согласование цепи) или с помощью комбинации клавиш (<CTRL-5>). Но тогда все значения должны вводиться вручную.

10.1 Согласование четырехполюсников

Если имя переменной в тексте маркера является S-параметром, то существует возможность одновременного согласования входа и выхода четырехполюсной цепи. Это работает очень похоже на вышеописанные шаги. Результатом являются две согласованные цепи: самый левый узел должен соединяться

с портом 1, самый правый узел - с портом 2, а два узла в середине должны соединяться с четырехполюсником.

Установленные файлы

В состав Qucs входят несколько программ. Они устанавливаются во время процесса инсталляции. Путь, куда устанавливается Qucs, определяется при установке (скриптом `configure`). В следующих объяснениях принимается путь по умолчанию (`/usr/local/`).

- `/usr/local/bin/qucs` - графический интерфейс
- `/usr/local/bin/qucsator` - симулятор (консольное приложение)
- `/usr/local/bin/qucsedit` - простой текстовый редактор
- `/usr/local/bin/qucshelp` - небольшая программа для отображения справочной системы
- `/usr/local/bin/qucstrans` - программа для расчета параметров линий передач
- `/usr/local/bin/qucsfilter` - программа синтеза фильтров
- `/usr/local/bin/qucsconv` - преобразователь форматов файлов (консольное приложение)

Все программы являются самостоятельными приложениями и могут быть запущены независимо друг от друга. Главная программа (графический интерфейс)

- вызывает `qucsator` при выполнении моделирования,
- вызывает `qucsedit`, когда показываются текстовые файлы,
- вызывает `qucshelp`, когда показывается справочная система,
- вызывает `qucstrans` при вызове этой программы из меню “Инструменты”,
- вызывает `qucsfilter` при вызове этой программы из меню “Инструменты”,
- вызывает `qucsconv`, когда вставляется компонент SPICE и когда выполняется моделирование с использованием компонента SPICE.

Аргументы командной строки

- `/usr/local/share/qucs/bitmaps` - содержит все растровые изображения (значки и т.д.)
- `/usr/local/share/qucs/docs` - содержит HTML-документы справочной системы
- `/usr/local/share/qucs/lang` - содержит файлы переводов

11.1 Аргументы командной строки

qucs [файл1 [файл2 ...]]

qucsator [-b] -i список_цепей -o набор_данных (b = полоса прогресса)

qucsedit [-r] [файл] (r = только для чтения)

qucshelp (без аргументов)

qucsconv -if spice -of qucs -i netlist.inp -o netlist.net

This document describes the schematic and library file formats of Qucs.

12.1 Schematic file format

This format is used for schematics (usually with suffix `.sch`) and for data displays (usually with suffix `.dpl`). The following text shows a short example of a schematic file.

```
<Qucs Schematic 0.0.6>
<Properties>
  <View=0,0,800,800,1,0,0>
</Properties>
<Symbol>
  <.ID -20 14 SUB>
</Symbol>
<Components>
  <R R1 1 180 150 15 -26 0 1 "50 Ohm" 1 "26.85" 0 "european" 0>
  <GND * 1 180 180 0 0 0 0>
</Components>
<Wires>
  <180 100 180 120 "" 0 0 0 "">
  <120 100 180 100 "Input" 170 70 21 "">
</Wires>
<Diagrams>
  <Polar 300 250 200 200 1 #c0c0c0 1 00 1 0 1 1 1 0 5 15 1 0 1 1 315 0 225 "" "" "">
    <"acnoise2:S[2,1]" #0000ff 0 3 0 0 0>
    <Mkr 6e+09 118 -195 3 0 0>
  </Polar>
</Diagrams>
<Paintings>
  <Arrow 210 320 50 -100 20 8 #000000 0 1>
</Paintings>
```

В файле несколько разделов. Все они объясняются ниже. Каждая линия состоит из не более чем одного информационного блока, начинающегося знаком **меньше** (<) и заканчивающегося знаком **больше** (>).

12.1.1 Properties

Первый раздел начинается с <Properties> и заканчивается </Properties>. Он содержит свойства документа, находящегося в файле. Каждая строка необязательна. Поддерживаются следующие свойства:

- <View=x1,y1,x2,y2,scale,xpos,ypos> содержит положение в пикселах окна схемы (первые четыре числа), его текущий масштаб и текущее положение верхнего левого угла (два последних числа).
- <Grid=x,y,on> содержит шаг сетки в пикселах (первые два числа) и включена сетка (последнее число = 1) или выключена (последнее число = 0).
- <DataSet=name.dat> содержит имя файла набора данных, связанного с этой схемой.
- <DataDisplay=name.dpl> содержит имя файла со страницей просмотра данных, связанной с этой схемой (или имя схемного файла, если этот документ является просмотром данных).
- <OpenDisplay=yes> содержит 1, если страница показа данных открывается автоматически после моделирования, в противном случае - 0.
- <Script=name.m> contains the file name of the octave script associated with this schematic.
- <RunScript=0> contains 1 if the octave script is executed after the simulation.
- <showFrame=0> specify if a frame is drawn and if so which size it is. valid values are 0 (do not show a frame), 1 (A5 landscape), 2 (A5 portrait), 3 (A4 landscape), 4 (A4 portrait), 5 (A3 landscape), 6 (A3 portrait), 7 (letter landscape) and 8 (letter portrait).
- <FrameText0=NE555 sub-circuit model>, FrameText1=Draw by: anonymous, FrameText2=Date: 1984, and <FrameText3=Revision: 42> specify the texts to be placed into the frame text boxes.

12.1.2 Symbol

Этот раздел начинается с <Symbol> и заканчивается </Symbol>. Он содержит элементы рисования, составляющие схемное обозначение для файла. Это обычно используется только для схемных файлов, которые считаются подсхемой.

Refers to “Symbol definition” in the “Shared file format” section at the end of this document.

12.1.3 Components

Этот раздел начинается с <Components> и заканчивается </Components>. Он содержит компоненты цепей схемы. Формат строки:

```
<type name active x y xtext ytext mirrorX rotate "Value1" visible "Value2" visible ...>
```

- **type** (“тип”) означает компонент, например, R для резистора, C для конденсатора.
- **name** (“имя”) - уникальное обозначение компонента на схеме, например, R1 для первого резистора.
- **1** в поле **active** (“активный”) показывает, что этот компонент активный, то есть используется в моделировании. **0** показывает, что он неактивный.
- Следующие два числа являются **x** и **y** координатами центра компонента.

- Следующие два числа являются x и y координатами верхнего левого угла текста компонента. Они отсчитываются от центра компонента.
- Следующие два числа указывают на зеркальное отображение относительно оси x (1 - зеркальное отображение, 0 - нет зеркального отображения) и вращение против часовой стрелки (кратно 90 градусам, то есть 0...3).
- Следующие параметры являются значениями свойств компонента (в кавычках), за которыми следует 1, если это свойство видимо на схеме (в противном случае 0).

12.1.4 Wires

Этот раздел начинается с `<Wires>` и заканчивается `</Wires>`. Он содержит проводники (электрическое соединение между компонентами цепи), их метки и узлы. Формат строки:

```
<x1 y1 x2 y2 "label" xlabel ylabel dlabel "node set">
```

- Первые четыре числа являются координатами проводника в пикселах: x-координата начальной точки, y-координата начальной точки, x-координата конечной точки и y-координата конечной точки. Все проводники должны быть либо горизонтальными (обе x-координаты равны), либо вертикальными (обе y-координаты равны).
- Первая строка в кавычках - имя метки. Она пустая, если пользователь не установил метку на этот проводник.
- Следующие два числа - x- и y-координаты метки или ноль, если метки не существует.
- The next number is the distance between the wire starting point and the point where the label is set on the wire.
- Последняя строка в кавычках - параметры цепи проводника, то есть начальное напряжение узла, используемое ядром симулятора для поиска решения. Эта строка пустая, если пользователь не установил параметры узла цепи для этого проводника.

12.1.5 Diagrams

Этот раздел начинается с `<Diagrams>` и заканчивается `</Diagrams>`. Он содержит диаграммы с их графиками и маркерами." The line format is as follows (line break not allowed):

```
<diatype x y width height grid gridcolor gridstyle log xAutoscale xmin
xstep xmax yAutoscale ymin ystep ymax zAutoscale zmin zstep zmax
xrotate yrotate zrotate "xlabel" "ylabel" "zlabel" "[freq Hz;]*">
  <"graphvar" color thickness precision numberformat style axisside>
  <Mkr x y precision numberformat transparent>
</diatype>
```

Diagram line format:

- The `diatype` token specifies the type of diagram.
- The `x` and `y` numbers are the coordinate of lower left corner.
- The `width` and `height` numbers of diagram boundings.
- The `grid` flags with 1 if grid is on and 0 if grid is off.
- The `gridColor` in 24 bit hexadecimal RGB value, e.g. `#FF0000` is red.
- The `gridstyle` is the line style sued of the grid.

- The `log` has two field to flag which axes have logarithmical scale.
- The `xAutoscale`, `xmin`, `xstep`, `xmax` configure the x-axis scaling, limits.
- The `yAutoscale`, `ymin`, `ystep`, `ymax` configure the y-axis scaling, limits.
- The `zAutoscale`, `zmin`, `zstep`, `zmax` configure the z-axis scaling, limits.
- The `xrotate`, `yrotate`, `zrotate` numbers set the 3D rotation.
- The `xlabel`, `ylabel`, `zlabel` hold the labels used on each axis.
- The list of frequencies "[freq Hz;]*" is used by `Phasor` and `Waveac`.

Here is a list of known diagram types:

- `Curve` for a locus curve diagram.
- `Smith` for an impedance Smith diagram.
- `ySmith` for an admittance Smith diagram.
- `PS` for a mixed polar/smith diagram.
- `SP` for a upper-half mixed polar/smith diagram.
- `Polar` for a polar diagram.
- `Rect` for a 2D-cartesian diagram.
- `Rect3D` for a 3D-cartesian diagram.
- `Tab` for a tabular diagram.
- `Time` for a timing diagram.
- `Truth` for a truth-table diagram.
- `Phasor` for a complex phasor diagram.
- `Waveac` for a wave as temporal diagram.

Graph line format:

- The `graphvar` specify the variable this graph is plotting for.
- The `color`, `thickness` and `style` refers to the pen used to draw the curve.
- The `precision` specify the number of digits used when displaying data values.
- The `numberformat` is an integer that specify how the number are formatted (0 for real/imag, 1 for polar/deg and 2 for polar/rad).
- The `axisside` is an integer indicating on which side the Y axis should be placed (0).

Marker line format:

- The `x` and `y` are the location of the marker.
- The `precision` ...
- The `numberformat` ...
- The `transparent`

12.1.6 Paintings

Этот раздел начинается с `<Paintings>` и заканчивается `</Paintings>`. Он содержит элементы рисования, которые используются в схеме.

Refers to “Shared file format” section below.

12.2 Library file format

This format is used for libraries (usually with suffix `.lib`). The following text shows a short example of a library file.

```
<Qucs Library 0.0.14 "Ideal">
<DefaultSymbol>
  <.ID -26 13 D>
  <Line -30 0 60 0 #000080 2 1>
  <Line -6 -9 0 18 #000080 2 1>
  <Line 6 -9 0 18 #000080 2 1>
  <Line -6 0 12 -9 #000080 2 1>
  <Line -6 0 12 9 #000080 2 1>
  <Line -6 9 4 0 #000080 2 1>
  <.PortSym -30 0 1 0>
  <.PortSym 30 0 2 180>
</DefaultSymbol>
<Component VSum>
  <Description>
Voltage adder
  </Description>
  <Model>
.Def:Ideal_AP1 _net3 _net2 fc="1E3"
Sub:VSub1 _net0 _net1 _net2 Type="VSub"
Sub:LP1F1 _net3 _net0 Type="LP1" fc="fc2" V0="0"
Sub:HP1F1 _net3 _net1 Type="HP1" fc="fc2"
Eqn:Eqn1 fc2="fc/0.6436" Export="yes"
.Def:End
  </Model>
  <ModelIncludes "HP1.sch.lst" "LP1.sch.lst" "VSub.sch.lst">
  <Symbol>
    <Ellipse -20 -20 40 40 #000080 2 1 #c0c0c0 1 0>
    <Line -10 0 20 0 #000080 1 1>
    <Line 0 -10 0 20 #000080 1 1>
    <Line 0 30 0 -10 #000080 2 1>
    <.PortSym 0 30 2 0>
    <.PortSym 30 0 3 180>
    <Line 20 0 10 0 #000080 2 1>
    <.ID 10 14 VADD>
    <Line 0 -20 0 -10 #000080 2 1>
    <.PortSym 0 -30 1 0>
  </Symbol>
</Component>
```

The first line specifies that this file is a Qucs library file generated by Qucs 0.0.14 and that the library is named “Ideal”.

The file contains an optional `DefaultSymbol` section, followed by `Component` sections. Each section is explained below.

12.2.1 Default symbol

This section starts with `<DefaultSymbol>` and ends with `</DefaultSymbol>`. It contains painting elements creating a default schematic symbol for any subsequent component declaration that doesn't define its own.

Refers to "Shared file format" section below.

12.2.2 Component

This section starts with `<Component>` and ends with `</Component>`. It contains the component definition for use with schematic documents.

The component section is an aggregation of the following sub-sections:

- `<Description>` and `</Description>` contain lines of free text describing the component function.
- `<Model>` and `</Model>` contain the Qucsator netlist lines for this component.
- `<ModelIncludes "value0value1"...>` ...
- `<Spice>` and `</Spice>` are optional and contain the Spice netlist lines for this component.
- `<Symbol>` and `</Symbol>` are optional and contain painting elements defining the schematic symbol to be used with this component. Refers to "Symbol definition" section below.

12.3 Shared file format

12.3.1 Painting elements

A painting line can be found in:

- The `Paintings` section of a schematic file.
- The `Symbol` sections of a schematic file.
- The `DefaultSymbol` section of a library file.
- The `Symbol` section (sub-section of `Component`) of a library file.

A painting line has one of the following format:

- `<Rectangle x y width height pencolor penwidth penstyle brushcolor brushstyle filled>`
...
- `<Ellipse x y width height pencolor penwidth penstyle brushcolor brushstyle filled>` ...
- `<EArc x y startangle spanangle width height pencolor penwidth penstyle brushcolor brushstyle filled>` ...
- `<Text x y size color angle "text">` ...
- `<Line x1 y1 x2 y2 pencolor penwidth penstyle >` ...
- `<Arrow x1 y1 x2 y2 x3 y3 pencolor penwidth penstyle >` ...

12.3.2 Symbol definition

A symbol definition can contains any painting element as described in the previous section. In addition to the painting elements, a symbol definition must contain one `.ID` line and one or more `.PortSym` lines.

The `.ID` line has the following format:

```
<.ID x y name "property1" "property2" ...>
```

Where:

- `x` and `y` are the center coordinates of the symbol.
- `name` will be used as a name prefix when instanciating this symbol on a schematic sheet.
- `propertyX` are used for symbol definition within a schematic file, these parameter will be associated with the symbol instance and communicated to the sub-schematic. The format for such a property is ``displayed=name=value=description=unknown``.

The `.PortSym` line has the following format:

```
<.PortSym x y caption angle>
```

Where:

- `x` and `y` are the coordinates of the port.
- `caption` is the name/caption of the port.
- `angle` is an angle value, it is ignored (backward compatibility).

Subcircuit and Verilog-A RF Circuit Models for Axial and Surface Mounted Resistors

Mike Brinson

Copyright 2014, 2015 Mike Brinson, Центр Коммуникационных технологий, Лондонский Университет Metropolitan, Лондон, Соединённое Королевство. (<mailto:mbrin72043@yahoo.co.uk>)

Разрешается копировать, распространять и/или изменять этот документ в соответствии с условиями GNU Free Documentation License версии 1.1 или более поздней версии, опубликованной Free Software Foundation.

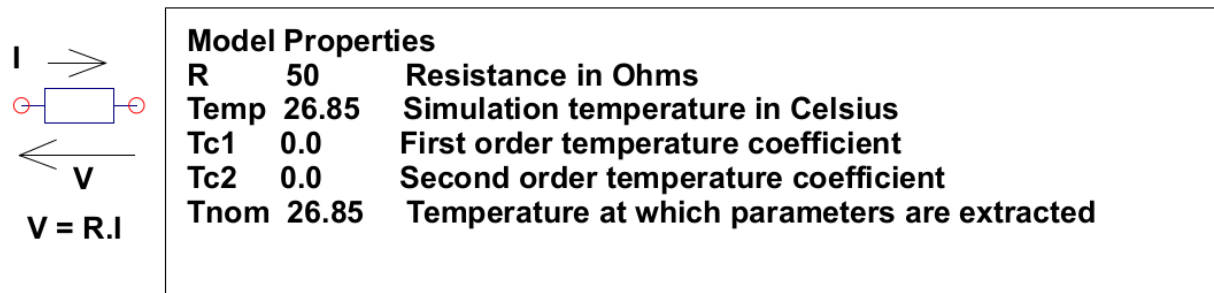
13.1 Введение

Resistors are one of the fundamental building blocks in electronic circuit design. In most instances conventional resistor circuit simulation models are characterized by I/V characteristics specified by Ohm's law. In reality the impedance of RF resistors is frequency dependent, being determined by component physical properties, component manufacturing technology and how components are connected in a circuit. At low frequencies fixed resistors have a nominal value at roomtemperature and can be modelled accurately by Ohm's law. At RF frequencies the fact that a resistor acts more like an inductance or a capacitance can play a crucial role in determining whether or not a circuit operates as designed. Similarly, if a resistor is modelled as an ideal component at a frequency where it exhibits significant reactive properties then the resulting simulation data are likely to be incorrect. The subcircuit and Verilog-A compact resistor models introduced in this Qucs note are designed to give good performance from low frequencies to RF frequencies not greater than a few GHz.

13.2 Модели радиочастотных резисторов

The schematic symbol, I/V equation and parameters of the Qucs linear resistor model are shown in Figure 1. In contrast to this model Figure 2 illustrates the structure of a printed circuit board (PCB) mounted metal

film (MF) axial RF resistor (a), its Qucs schematic symbol (b) and its equivalent circuit model (c). A thin film surface mounted (SMD) resistor can also be represented by the model shown in Figure 2 (c).



where $R(\text{Temp}) = R(\text{Tnom}) \cdot (1 + \text{Tc1} \cdot (\text{Temp} - \text{Tnom}) + \text{Tc2} \cdot (\text{Temp} - \text{Tnom})^2)$

Figure 1 - Qucs built-in resistor model.

At signal frequencies where the largest dimension of an axial or SMD resistor is less than approximately 20 times the smallest signal wavelength a resistor can be modelled by a lumped passive circuit consisting of a resistor **Rs** in series with a small inductance **Ls** with the combination shunted by parasitic capacitor **Cp**. In Figure 2 **Rs** is the nominal value of resistor at its parameter extraction temperature **Tnom**, **Ls** represents the inductance associated with **Rs** where the value of **Ls** is largely determined by the trimming method employed during component manufacture to set the value of **Rs** to a specified tolerance. Similarly, capacitor **Cp** models a parasitic capacitance associated with **Rs** where the value of **Cp** is a function of the physical size of **Rs**. At RF frequencies it is important, for accurate operation, to add lead parasitic elements to the intrinsic equivalent circuit model shown within the red box draw in Figure 2. In Figure 2 **Llead** and **Cshunt** represent resistor series lead inductance and shunt capacitance to ground respectively.

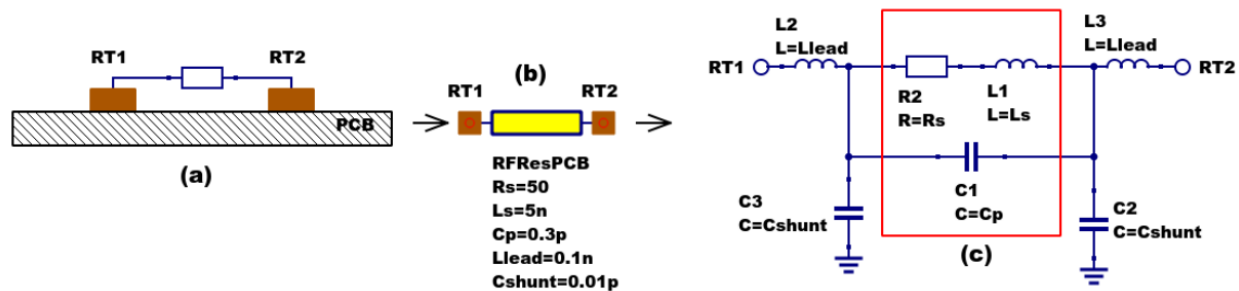


Figure 2 - PCB mounted resistor: (a) axial component mounting, (b) Qucs symbol and (c) equivalent circuit model.

A typical set of model parameters for a 51Ω 5 % MF axial resistor are (1) **Ls** = 8nH, **Cp** = 1pF, **Llead** = 1nH and **Cshunt** = 0.1pF. Illustrated in Figure 3 is a basic S parameter test bench circuit for measuring the S parameters of an RF resistor over a frequency range 1 MHz to 1.3 GHz. This example also demonstrates how the real and imaginary parts of a resistor model impedance can be extracted from S parameter simulation data. The graphs in Figure 3 clearly demonstrate that the impedance of the typical MF RF resistor described in previous text and modelled by the equivalent circuit shown in Figure 2 is a strong function of frequency at higher frequencies in the band 1 MHz to 1.3 GHz.

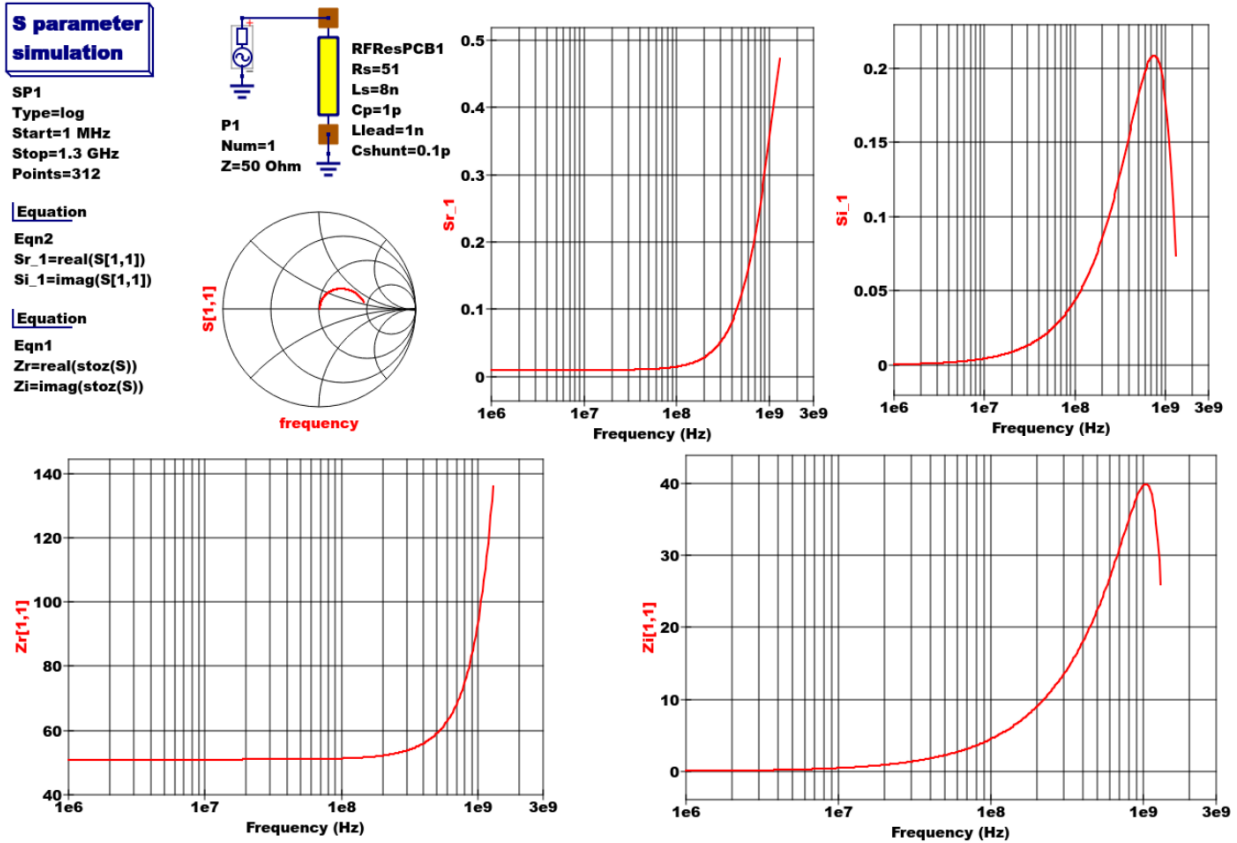


Figure 3 - Qucs S parameter simulation test circuit and plotted output data for a MF axial resistor: $R_s=51\Omega$, $L_s=8\text{nH}$, $C_p=1\text{pF}$, $L_{lead}=1\text{nH}$ and $C_{shunt}=0.1\text{pF}$.

13.3 Анализ модели радиочастотного резистора

A component level version of the proposed RF resistor model is shown in Figure 4, where

$$\begin{aligned}
 Z1 &= j \cdot \omega \cdot L_{lead} \\
 Z2 &= \frac{R_s + j \cdot \omega \cdot L_s \cdot (1 - \omega^2 \cdot C_p \cdot L_s) - j \cdot \omega \cdot C_p \cdot R_s^2}{(1 - \omega^2 \cdot C_p \cdot L_s)^2 + (\omega \cdot C_p \cdot R_s)^2} \\
 Z3 &= \frac{j \cdot \omega \cdot L_{lead}}{(1 - \omega^2 \cdot L_{lead} \cdot C_{shunt})} \\
 Z_{series} &= Z1 + Z2 = R_{series} + j \cdot X_{series} \\
 Zb &= Z_{series} || XC_{shunt} = \frac{Z_{series}}{(1 + j \cdot \omega \cdot C_{shunt} \cdot Z_{series})} = ZBR + j \cdot \omega \cdot ZBI, \\
 Z &= j \cdot \omega \cdot L_{lead} + Zb = ZR + j \cdot \omega \cdot ZI.
 \end{aligned}$$

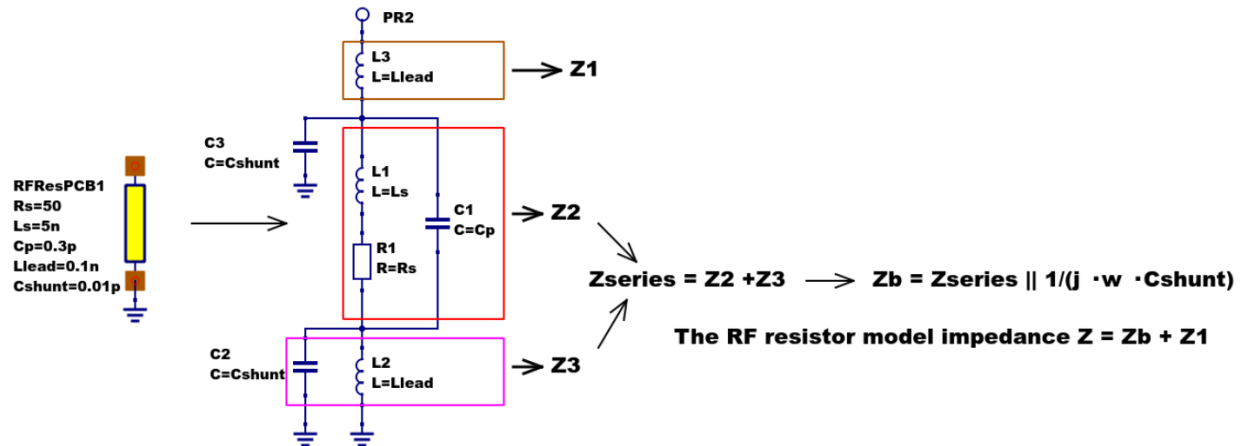


Figure 4 - RF resistor model rotated through 90 degrees and connected with one terminal grounded, similar to the test circuit in Figure. Sections of the model are shown grouped for calculation of the model impedance Z .

Figure 5 illustrates how a set of theoretical equations can be converted into Qucs equations for model simulation and post simulation data processing. In this example Qucs equation **Eqn1** holds values for RF resistor model parameters and Qucs equation **Eqn2** lists the model equations introduced at the start of this section. Figure 5 also gives a set of cartesian graphs of post simulation output data which illustrate how **ZR** and **ZI**, and other calculated items, vary with frequency over the range 1 MHz to 1.3 GHz.

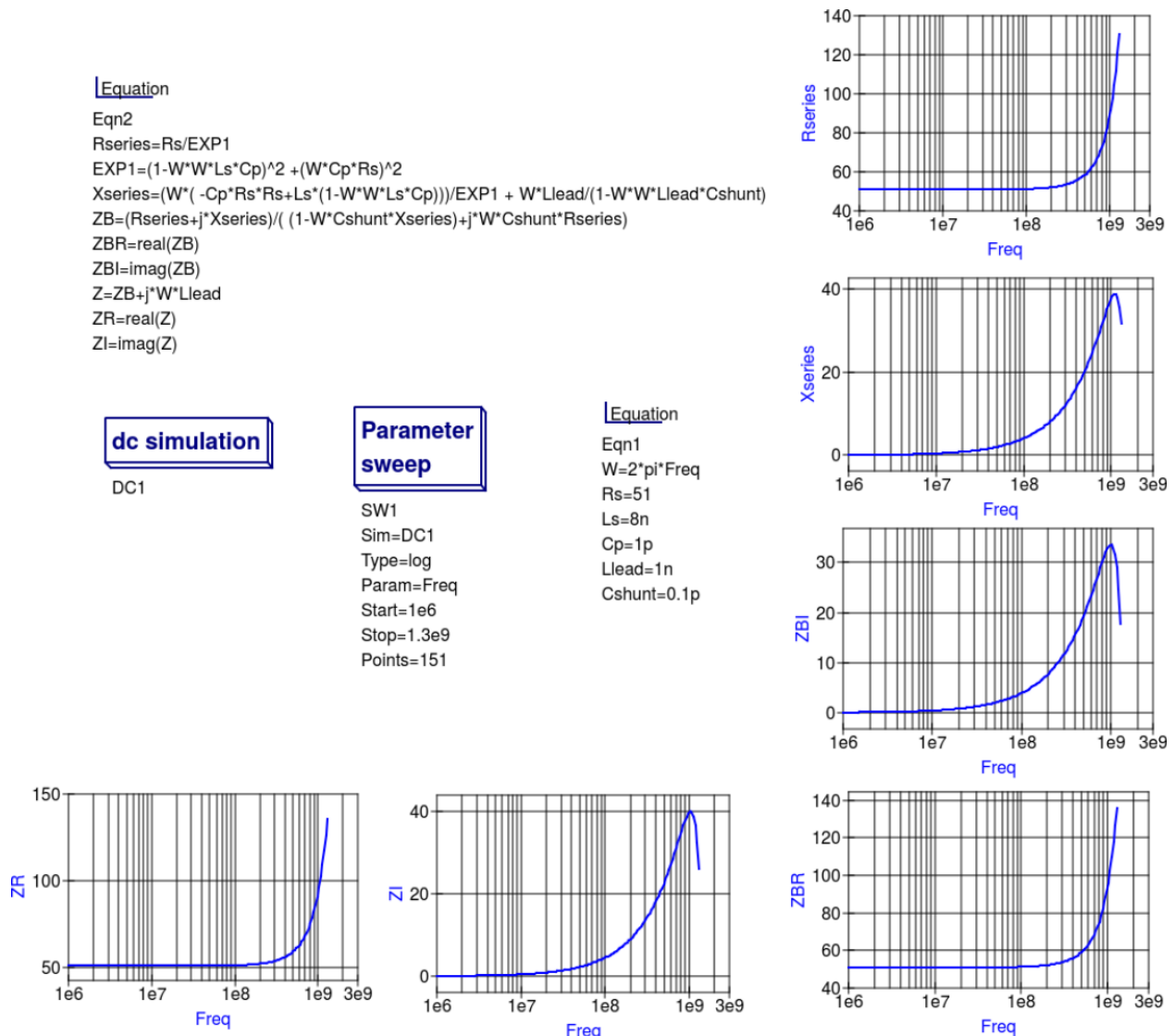


Figure 5- Theoretical analysis of RF resistance impedance Z using Qucs post processing facilities: note a dummy simulation icon, in this example DC simulation, is required to force Qucs to complete the analysis calculations.

13.4 Direct measurement of RF resistor impedance using a simulated impedance meter

A simple impedance meter for measuring the real and imaginary components of component and circuit impedance, using small signal AC simulation, is shown in Figure 6. The impedance measuring technique uses a 1 Amp AC constant current source applied to one terminal of a two port electrical network. The second terminal is grounded. A parallel high resistance resistor ($1\text{E}9\ \Omega$ in Figure 6) shunts the network under measurement to ensure that there is always a direct current path to ground as required by the Qucs simulator during the calculation of simulation results. If required the 1 Amp AC source can be set at a lower value. In such cases the value of **VRes** must also be scaled to give the network impedance.

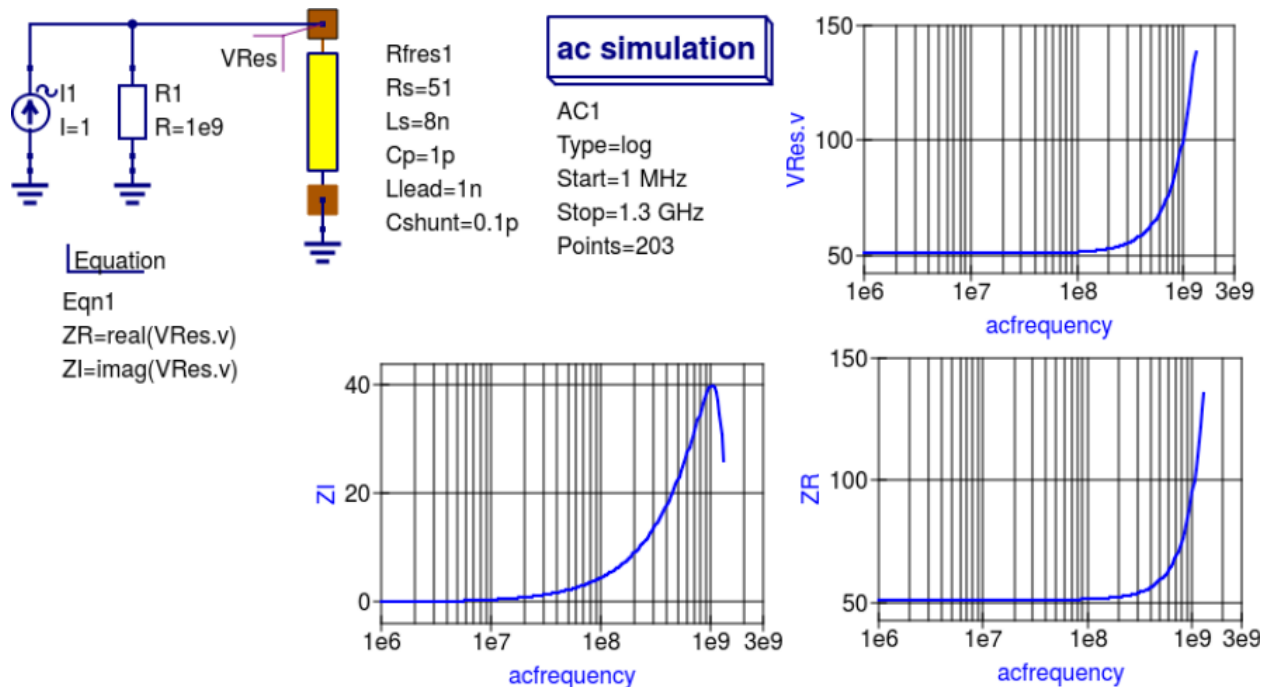


Figure 6 -A simple Qucs test circuit for demonstrating the use of an AC constant current source to measure electrical network impedance.

13.5 Extraction of RF resistance data from measured S parameters

In the past the cost of Vector Network Analyser systems for measuring S parameters has been prohibitively expensive for individual engineers to purchase. However, this scene is changing with the introduction of low cost systems like the DGSAQ Vector Network Analyser (VNA)¹. This instrument operates over a frequency band width of 1.3 GHz, providing a range of useful functions with highest accuracy at frequencies up to 500 MHz. This form of VNA is particularly suited to Radio Amateur requirements and Qucs users interested in RF circuit analysis and design. Such equipment is ideal for measuring RF circuit S parameters and providing measured data for subcircuit and Verilog-A compact device model parameter extraction. Shown in Figure 7 is a graph of measured S parameter data for a nominal 47 Ω resistor². As well as displaying, and printing, measured data the DGSAQ Vector Network Analyser software can output data tabulated in Touchstone³ “SnP” file format. These files can be read by Qucs and their contents attached to an S parameter file icon for inclusion in circuit schematic diagrams. Figure 8 shows this process as part of an RF resistor model parameter extraction technique involving DGSAQ VNA measured S parameter data and Qucs simulated S parameter data.

The brown “Test circuits” box shows test circuits for firstly reading and processing the DGSAQ VNA measured data listed in file mike3.s1p, and for secondly generating simulated S parameter data for an RF resistor specified by parameters **Ls** = **L**, **Cp** = **C**, **Llead** = **LL**, **Cshunt** = **0.08 pF**, and **Rs** = **47.3 Ω** . Presented in Figure 9 are the Qucs Optimization controls⁴ which are used to set the range of **L**, **C** and **LL** values that optimizer ASCO will select from to obtain the best fit between the measured and simulated S parameter data. Note in this parameter extraction system that **S[1,1]** refers to measured S parameter data and **S[2,2]** to simulated S parameter data. Two least squares cost functions called **CF1** and **CF2** are used as targets in the minimisation process. Values for **CF1** and **CF2** can be found in the red box called “Simulation

¹ DG8SAQ VNA 3 & 3E- Vector Network Analysers, SDR Kits Limited, Grangeside Business Centre, 129 Devizes Road, Trowbridge, Wilts, BA14-7sZ, United Kingdom, 2014.

² See DG8SAQ VNA 3 & 3E- Vector Network Analysers- Getting Started Manual for Windows 7, Vista and Windows XP.

³ (http://www.vhdl.org/ibis/connector/touchstone_spec11.pdf).

Controls“. In this parameter extraction example the least squares cost function **CF1** is employed to minimize the square of the difference between the real values of the S parameters and least squares cost function **CF2** is employed to minimize the square of the difference between the imaginary values of the S parameters. Qucs post-simulation processing is also used to extract values for the real and imaginary components of the RF resistor impedance. Both the S parameter data and the impedance data are displayed as graphs in Figure 8.

Notice in this example the SPICE optimizer ASCO is used to find the values of **L**, **C** and **LL** which minimize **CF1** and **CF2**. Also note that **Rs** and **Cshunt** are held at fixed values during optimization. In the case of *Rs* its nominal value can be found from DC or low frequency AC measurements. Similarly the value selected for *Cshunt* has been chosen to give a very small but representative value of the parasitic shunt capacitance.. After optimization finishes the minimized values of **L**, **C** and **LL** are given in the initial value column of the Qucs optimization Variables list, see Figure 9. For the 47 Ω resistor the post-minimization RF resistor model parameters are **Rs** = 47.3 Ω , **Ls** = 10.43 nH, **Cp** = 0.69 pF, **Llead** = 1.46 nH and **Cshunt** = 0.08 pF. The theoretical simulation data illustrated in Figure 10 shows good agreement with the measured and the optimized simulation data.

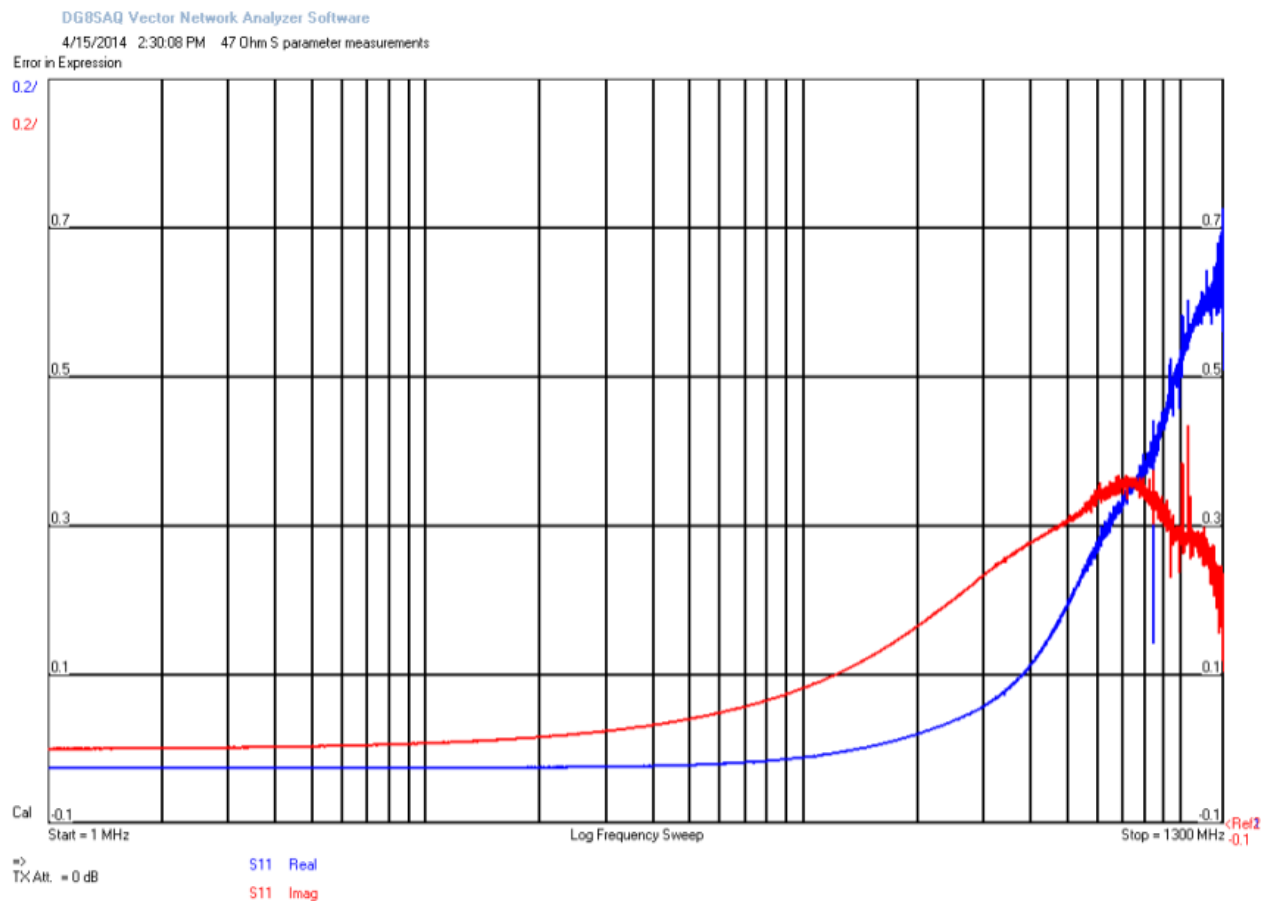


Figure 7 - DGSAQ Vector Network Analyser S parameter measurements for a 47 Ω axial RF resistor.

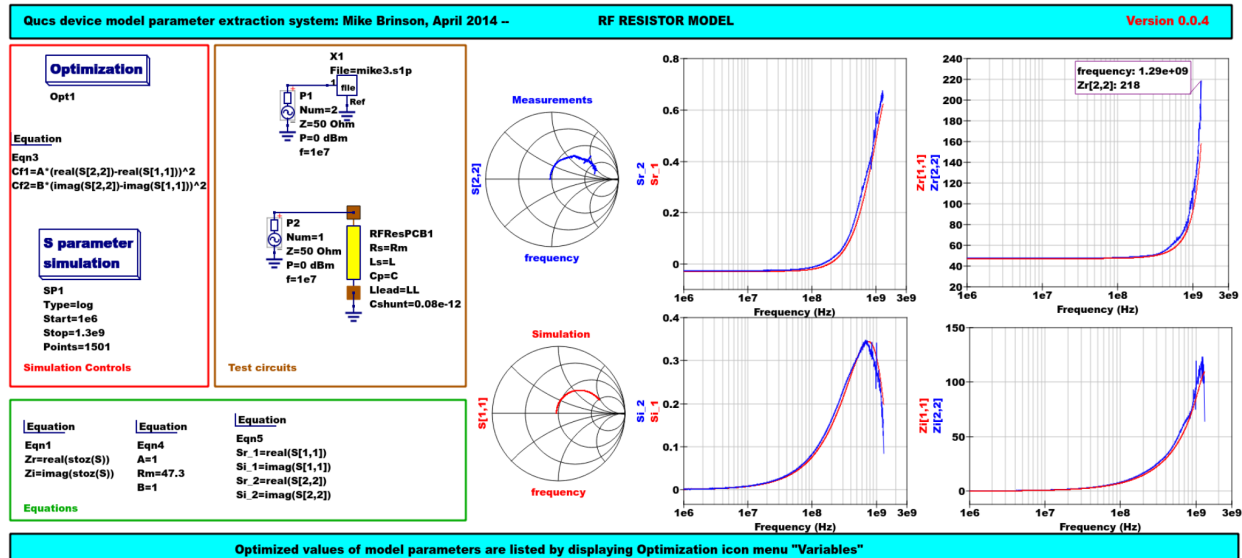
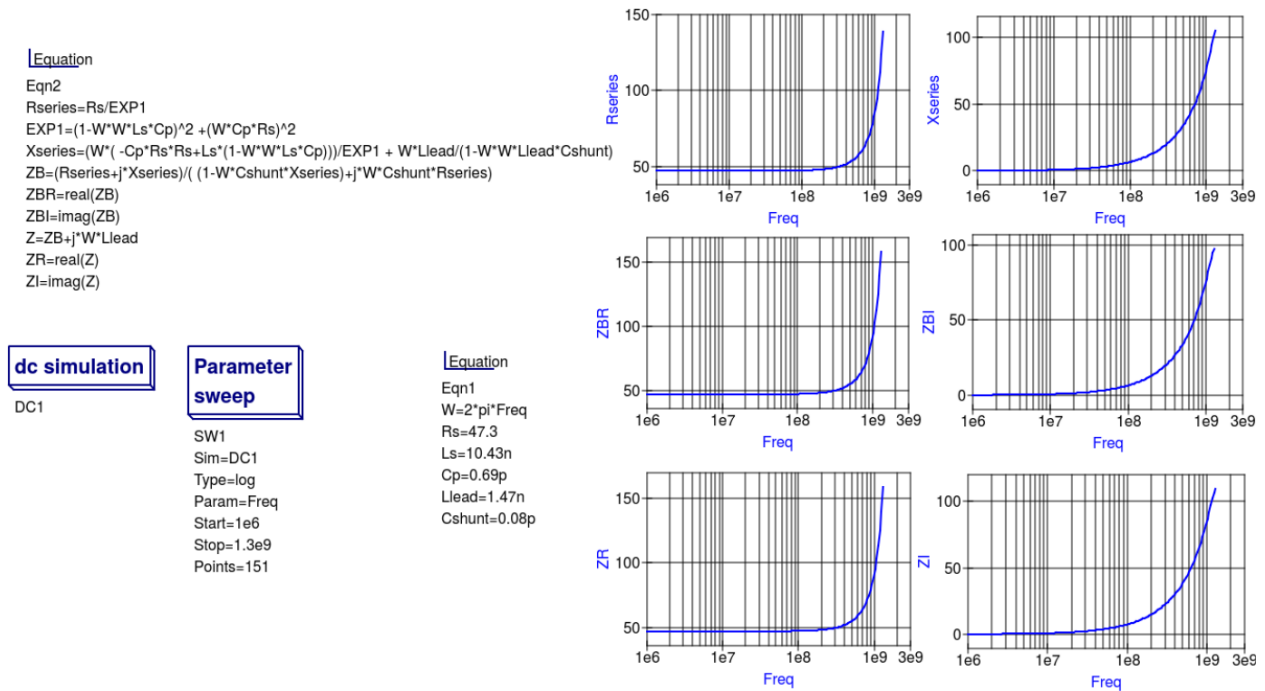
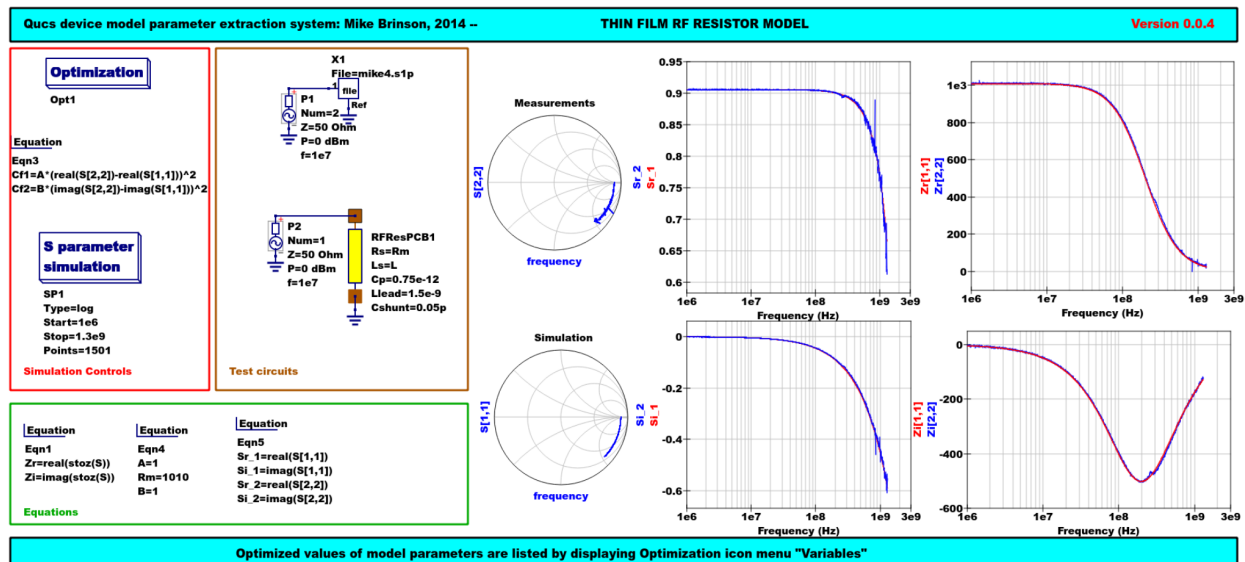
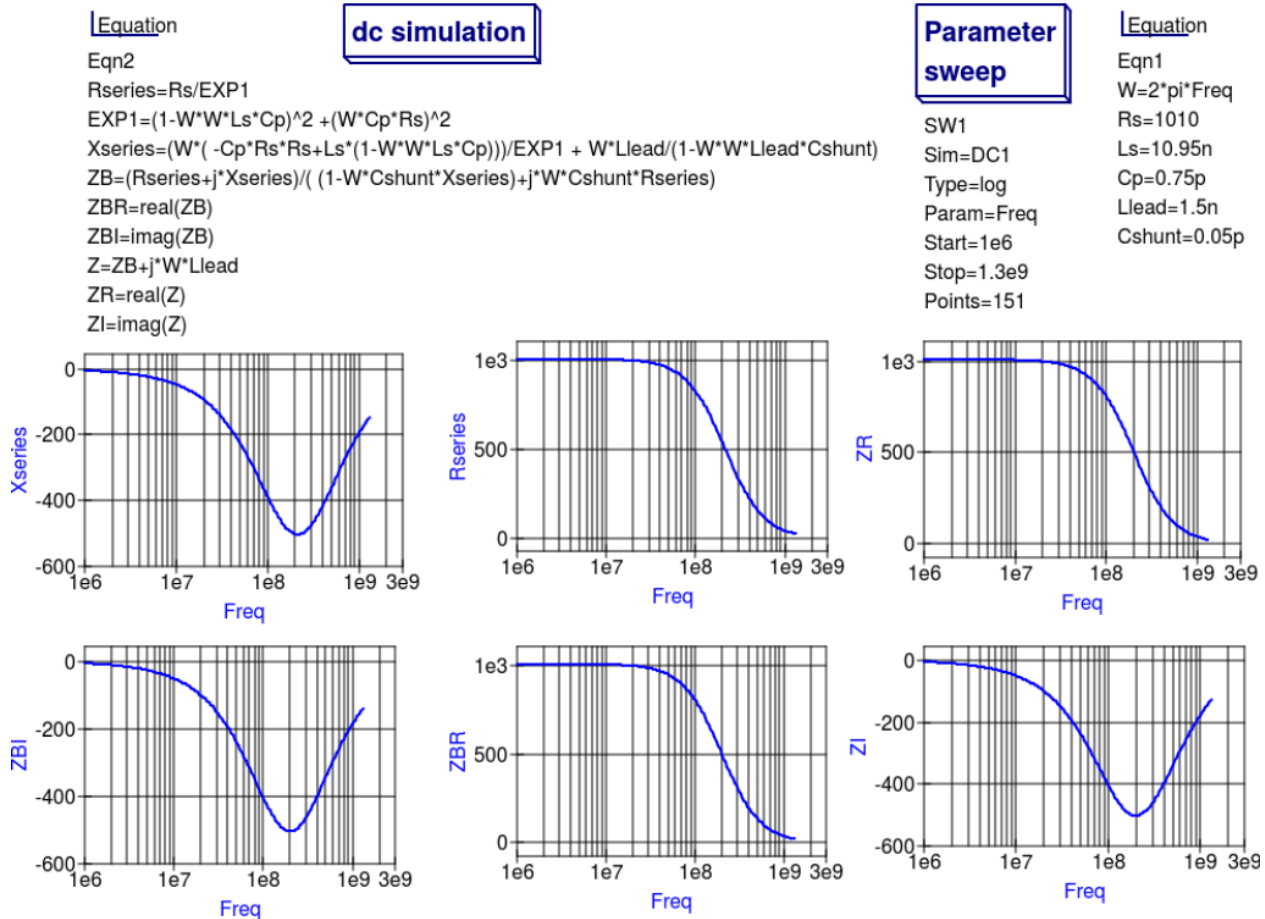


Figure 8 - Qucs device model parameter extraction system applied to a nominal $47\ \Omega$ resistor represented by the subcircuit model illustrated in Figure 2 (c). Fixed model parameter values: $R_s = R_m = 47.3\ \Omega$, $C_{Shunt} = 0.08\text{pF}$; Optimised values: $L_s = L = 10.43\text{nH}$, $L_{lead} = LL = 1.47\text{nH}$, $C_p = C = 0.69\text{pF}$. To reduce simulation time the ASCO cost variance was set to $1\text{e-}3$. The ASCO method was set to DE/best/1/exp.



Figure 9 - Qucs Minimization Icon drop down menus: left "Variables" and right "Goals".

Figure 10 - Qucs simulation of nominal 47 Ω resistor based on theoretical analysis.Figure 11 - Qucs device model parameter extraction system applied to a nominal 1000 Ω resistor represented by the subcircuit model illustrated in Figure 2(c).

Figure 12 - Qucs simulation of nominal 1000 Ω resistor based on theoretical analysis.

13.6 Extraction of RF resistor parameters from measured S data for a nominal 1000 Ω axial resistor

At low resistance values the impedance of an RF resistor becomes inductive as the signal frequency is increased. This is due to the fact that the inductance **Ls** contribution dominates any reactance effects by **Cp**, **Llead** and **Cshunt**. However, as **Rs** is increased above a few hundred Ohm's the reverse becomes true with reactive effects dominated by contributions from **Cp**. Figures 11 and 12 demonstrate the dominance of **Cp** reactive effects at low to mid-range frequencies.

13.7 One more example: extraction of RF resistor parameters from measured S data for a nominal 100 Ω SMD resistor

Figure 13 is included in this Qucs note purely for comparison purposes. SMD resistors are in general physically very small when compared to axial resistors. This results in lower values for the inductive and capacitive parasitics which in turn ensures that the high frequency performance of SMD resistors is much improved.

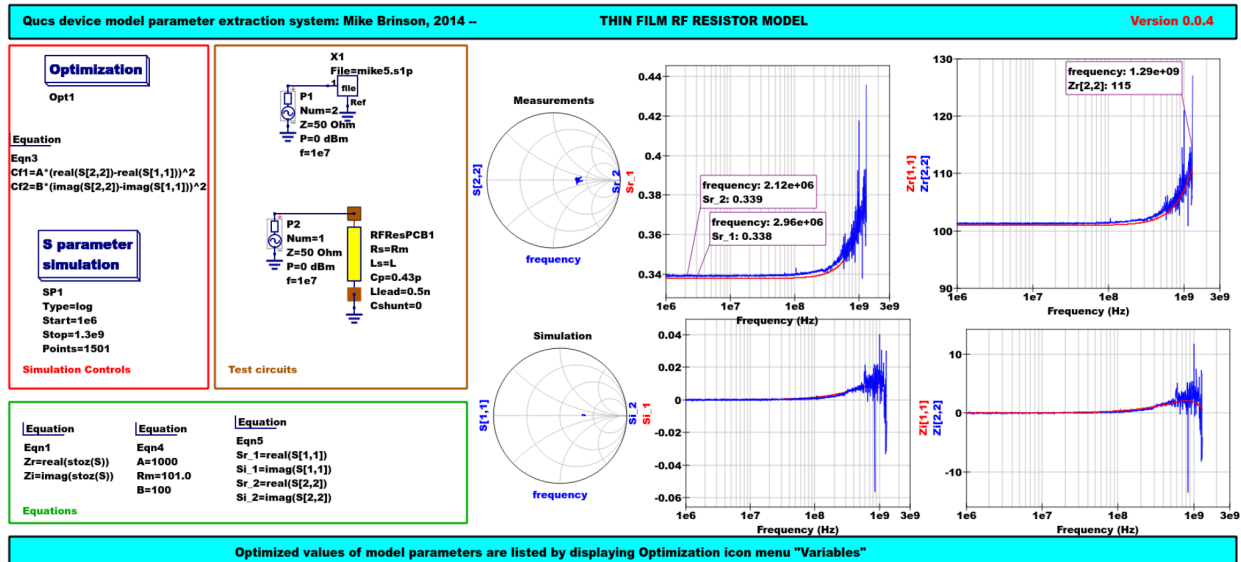


Figure 13 - Qucs device model parameter extraction system applied to a nominal $100\ \Omega$ SMD resistor represented by the subcircuit model illustrated in Figure 2 (c).

13.8 Verilog - Модель радиочастотного резистора

Listed below is an example Verilog-A code model for the RF resistor model introduced in Figure 2 (c). Due to the limitations of the Verilog-A language subset provided by version 2.3.4 of the "Analogue Device Model Synthesizer" (ADMS)⁴ inductors **Ls** and **Llead** are modelled by gyrators and capacitors with values identical to **Ls** or **Llead**.

```
// Verilog-A module statement.
//
// RFresPCB.va RF resistor (Thin film resistor, axial type, PCB mounting)
//
// This is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation; either version 2, or (at your option)
// any later version.
//
// Copyright (C), Mike Brinson, mbrin72043@yahoo.co.uk, April 2014.
//
`include "disciplines.vams"
`include "constants.vams"
// Verilog-A module statement.
module RFresPCB(RT1, RT2);
  inout RT1, RT2;          // Module external interface nodes.
  electrical RT1, RT2;
  electrical n1, n2, n3, nx, ny, nz;      // Internal nodes.
  `define attr(txt) (*txt*)
  parameter real Rs = 50      from [1e-20 : inf)
    `attr(info="RF resistance" unit="Ohm's");
  parameter real Cp = 0.3e-12 from [0 : inf)
    `attr(info="Resistor shunt capacitance" unit="F");
  parameter real Ls = 8.5e-9  from [1e-20 : inf)
    `attr(info="Series inductance" unit="H");
```

⁴ (<http://sourceforge.net/projects/mot-adms/>).

```
parameter real Llead = 0.1e-9 from [1e-20 : inf)
`attr(info="Parasitic lead inductance" unit="H");
parameter real Cshunt = 1e-10 from [1e-20 : inf)
`attr(info="Parasitic shunt capacitance" unit="F");
parameter real Tc1 = 0.0 from [-100 : 100]
`attr(info="First order temperature coefficient" unit ="Ohm/Celsius");
parameter real Tc2 = 0.0 from [-100 : 100]
`attr(info="Second order temperature coefficient" unit ="(Ohm/Celsius)^2");
parameter real Tnom = 26.85 from [-273.15 : 300]
`attr(info="Parameter extraction temperature" unit="Celsius");
parameter real Temp = 26.85 from [-273.15 : 300]
`attr(info="Simulation temperature" unit="Celsius");
branch (RT1, n1) bRT1n1; // Branch statements
branch (n1, n2) bn1n2;
branch (n1, n3) bn1n3;
branch (n2, n3) bn2n3;
branch (n3, RT2) bn3RT2;
real Rst, FourKT, n, Tdiff, Rn;
analog begin // Start of analog code
@(initial_model)
begin
  Tdiff = Temp-Tnom; FourKT =4.0*`P_K*Temp;
  Rst = Rs*(1.0+Tc1*Tdiff+Tc2*Tdiff*Tdiff); Rn = FourKT/Rst;
end
I(n1) <+ ddt(Cshunt*V(n1)); I(bn1n2) <+ V(bn1n2)/Rst;
I(bn1n3) <+ ddt(Cp*V(bn1n3)); I(n3) <+ ddt(Cshunt*V(n3));
I(bRT1n1) <+ -V(nx); I(nx) <+ V(bRT1n1); // Llead
I(nx) <+ ddt(Llead*V(nx));
I(bn2n3) <+ -V(ny); I(ny) <+ V(bn2n3); // Ls
I(ny) <+ ddt(Ls*V(ny));
I(bn3RT2) <+ -V(nz); I(nz) <+ V(bn3RT2); // Llead
I(nz) <+ ddt(Llead*V(nz));
I(bn1n2) <+ white_noise(Rn, "thermal"); // Noise contribution
end // End of analog code
endmodule
```

Module RFresPCB

Input Variables

Input Variables: instance=0 (bold) and model=9

name	description	default
Rs	RF resistance	50
Cp	Resistor shunt capacitance	0.3e-12
Is	Series inductance	8.5e-9
Llead	Parasitic lead inductance	0.1e-9
Cshunt	Parasitic shunt capacitance	1e-10
Tc1	First order temperature coefficient	0.0
Tc2	Second order temperature coefficient	0.0
Tnom	Parameter extraction temperature	26.85
Temp	Simulation temperature	26.85

Output Variables

Output Variables: instance=0
(bold) and model=0
(red-underlined: temperature
dependent)

name	description	dependencies
------	-------------	--------------

Nature/Discipline Definition

Nature

name	access	abstol	units
Current	I	1e-12	A
Charge	Q	1e-14	coul
Voltage	V	1e-6	V
Flux	Phi	1e-9	Wb
Magneto_Motive_Force	MMF	1e-12	A*turn
Temperature	Temp	1e-4	K
Power	Pwr	1e-9	W
Position	Pos	1e-6	m
Velocity	Vel	1e-6	m/s
Acceleration	Acc	1e-6	m/s^2
Impulse	Imp	1e-6	m/s^3
Force	F	1e-6	N
Angle	Theta	1e-6	rads
Angular Velocity	Omega	1e-6	rads/s
Angular Acceleration	Alpha	1e-6	rads/s^2
Angular_Force	Tau	1e-6	N*m

Discipline

name	potential	flow
logic		
electrical	Voltage	Current
voltage	Voltage	
current	Current	
magnetic	Magneto_Motive_Force	Flux
thermal	Temperature	Power
kinematic	Position	Force
kinematic_v	Velocity	Force
rotational	Angle	Angular_Force
rotational_omega	Angular_Velocity	Angular_Force

Model Equations

Notations used:

- green: input parameter
- bar over: variable never used
- bar under: temperature dependent variable
- red: voltage dependent variable

Initial Model

$T_{diff} = (Temp - T_{nom});$

$FourKT = ((4.0 \cdot 1.3806503e-23) \cdot Temp);$

$R_{st} = (Rs \cdot ((1.0 + (Tc1 \cdot T_{diff})) + ((Tc2 \cdot T_{diff}) \cdot T_{diff})));$

$R_n = \frac{FourKT}{R_{st}};$

----- end of Initial Model

$I(n1, n1) <+ ddt((C_{shunt} \cdot V(n1, n1)));$

$I(n1, n1) <+ \frac{V(n1, n1)}{R_{st}};$

$I(n1, n1) <+ ddt((C_p \cdot V(n1, n1)));$

$I(n3, n3) <+ ddt((C_{shunt} \cdot V(n3, n3)));$

$I(RT1, RT1) <+ (-V(nx, nx));$

$I(nx, nx) <+ V(RT1, RT1);$

$I(nx, nx) <+ ddt((L_{lead} \cdot V(nx, nx)));$

$I(n2, n2) <+ (-V(ny, ny));$

$I(ny, ny) <+ V(n2, n2);$

$I(ny, ny) <+ ddt((L_s \cdot V(ny, ny)));$

$I(n3, n3) <+ (-V(nz, nz));$

$I(nz, nz) <+ V(n3, n3);$

$I(nz, nz) <+ ddt((L_{lead} \cdot V(nz, nz)));$

$I(n1, n1) <+ white_noise(Rn, "thermal");$

Figure 14 - Details of the proposed RF resistor model: equations, variables and other data.

13.9 Extraction of Verilog-A RF resistor model parameters from measured S data for a 100 Ω axial resistor

This example demonstrates the use of ASCO for extracting Verilog-A model parameters from measured S parameter data. ASCO optimization yields a figure of 4nH for **L** in the model shown in Figure 2 (c). Other model parameter values are given with the test circuit, see Figure 15.

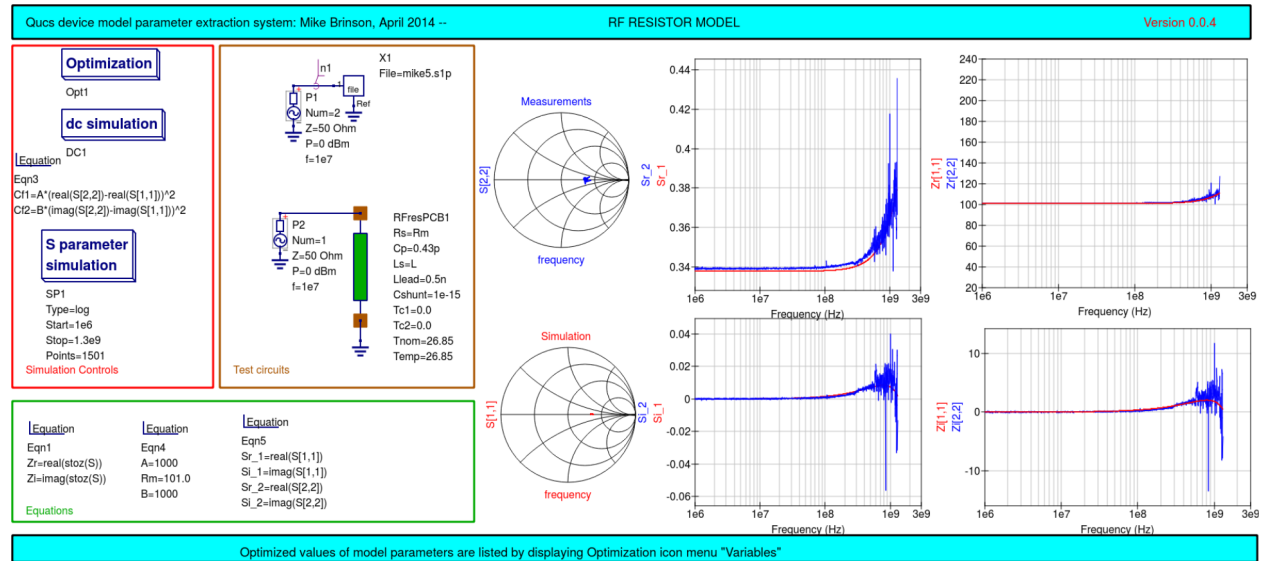


Figure 15 - Verilog-A models parameter data extraction for a 100 Ω axial thin film resistor. Fixed model parameter values: **$R_s = R_m = 101 \Omega$** , **$C_{Shunt} = 1e-15 \text{ F}$** , **$L_{lead} = L_L = 0.5 \text{ nH}$** , **$C_p = C = 0.43 \text{ pF}$** ; Optimised values: **$L_s = L = 3.99 \text{ nH}$** . To reduce simulation time the ASCO cost variance was set to 1e-3. The ASCO method was set to DE/best/1/exp.

13.10 Конец примечаний

This brief Qucs note outlines the fundamental properties of subcircuit and verilog-A compact component models for RF resistors. The use of optimization for the extraction of subcircuit and Verilog-A compact model parameters from measured S parameters is also demonstrated. The presented techniques form part of the simulation and device modelling capabilities available with the latest Qucs release⁵.

Технические описания, касающиеся симулятора

Имеются на <http://qucs.sourceforge.net/tech/technical.html>

Примеры схем

Имеются на <http://qucs.sourceforge.net/download.html#example>

⁵ Релиз Qucs 0.0.18 или выше.