
Qucs Help Documentation

Vydání 0.0.18

Qucs Team (2014)

20.01.2015

1	Začínáme s Analogovými simulacemi	3
2	Začínáme s Digitálními simulacemi	7
2.1	Komponenta VHDL soubor	8
3	Začínáme s optimalizací	11
4	Getting Started with Octave Scripts	17
4.1	Postprocessing	17
5	Krátký popis k ovládání	19
5.1	Hlavní ovládání	19
5.2	“Výběr”-Mód	19
5.3	“Vložit komponentu”-Mód	20
5.4	“Vodič”-Mód	20
5.5	“Vložit”-Mód	20
5.6	Myš v záložce “Obsah”	20
5.7	Klávesnice	21
6	Práce s vnořenými obvody	23
6.1	Subcircuits with Parameters	24
7	Krátký popis matematických funkcí	27
7.1	Operátory	27
7.2	Matematické funkce	28
7.3	Electronics Functions	31
7.4	Nomenclature	32
7.5	Constants	33
8	Seznam speciálních symbolů	35
9	Ladění obvodů	37
9.1	2-vývodové ladění obvodů	37
10	Instalované soubory	39
10.1	Příkazová řádka - argumenty	39
11	Popis k formátu souborů	41
11.1	Vlastnosti (Properties)	41

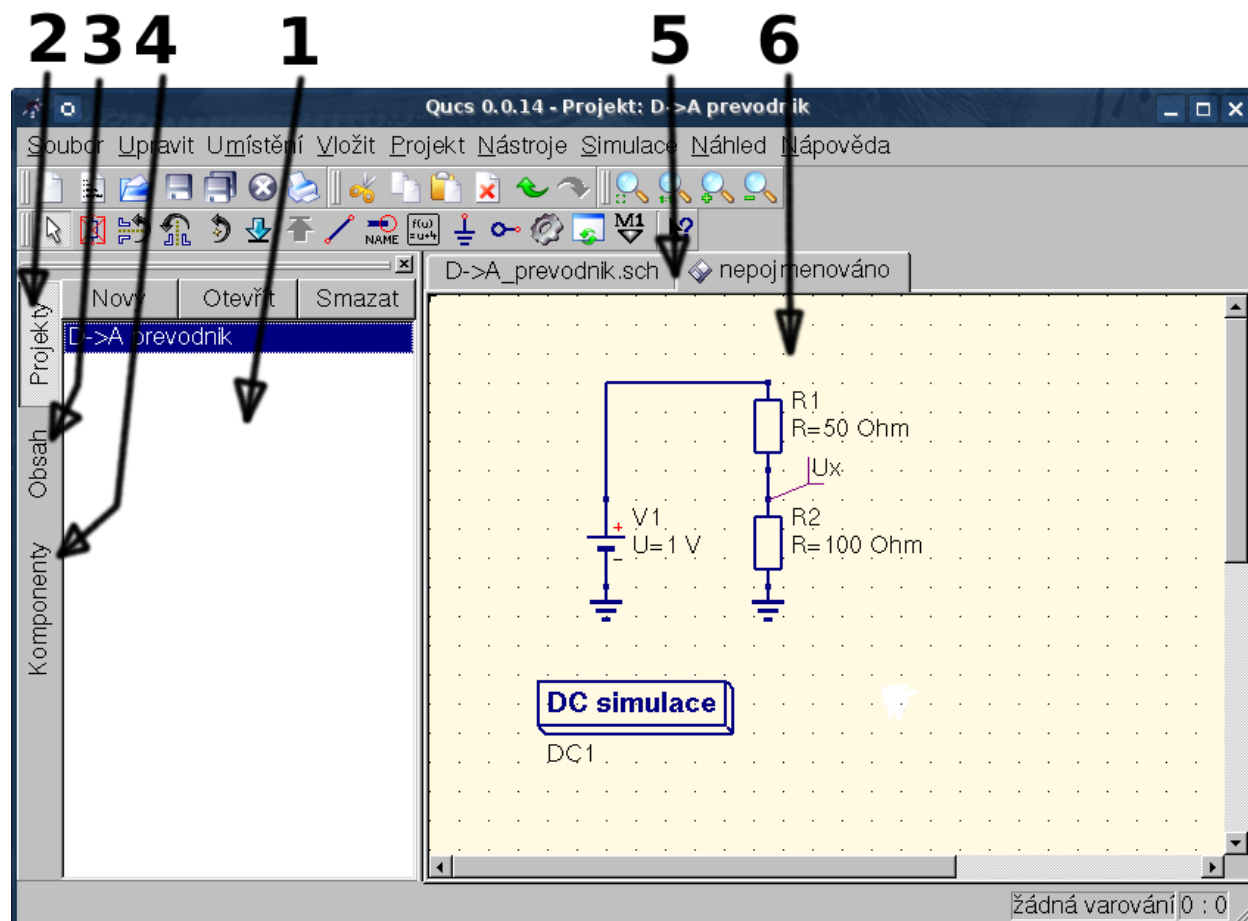
11.2	Symbol	42
11.3	Components (Komponenty)	42
11.4	Vedení	42
11.5	Diagramy	43
11.6	Obrazce	43

Contents:

Začínáme s Analogovými simulacemi

Qucs (vyslovovat kju:ks) je simulátor elektronických obvodů s grafickým rozhraním. Je schopen provádět mnoho rozdílných typů simulace (například DC simulace s parametrem) Tento manuál by vám měl seznámit se základy, jak používat Qucs.

Když spustíte Qucs poprvé, vytvoří adresář ".qucs" ve vašem domovském adresáři. Každý soubor je uložen do této složky, nebo do některých podadresářů. Po tom, co se Qucs načte, uvidíte hlavní okno, podobné tomu na obrázku 1. Na pravé straně je pracovní oblast (6) obsahující schémata, zobrazuje data, atd.. Použitím horních záložek (5) můžete rychle přepínat mezi dokumenty, které máte současně otevřené. Na levé straně hlavního okna je další oblast (1) která zobrazuje svůj obsah v závislosti na zvolené záložce: "Projekty" (2), "Obsah" (3) a "Komponenty" (4). Po spuštění Qucs se automaticky aktivuje záložka "Projekty" (2). A protože toto je poprvé, co spouštíte tento program, záložka je prázdná, protože jste zatím nevytvořili žádný projekt. Stiskněte tlačítko "Nový" právě v oblasti (1). Poté se otevře malé okno. Zde se píše název projektu, například "prvni_projekt" a stiskněte tlačítko "Hotovo". Qucs vytvoří projektu vlastní složku v ~/.qucs, pro tento příklad "prvni_prjekt_prj". Každý soubor patřící k tomuto projektu bude uložen do tohoto adresáře. Nový projekt se okamžitě otevře (lze si název projektu přecíst na horní liště) a záložka vlevo se přepne na "Obsah" (3), kde je zobrazen obsah právě otevřeného projektu. Zatím nemáte vytvořený žádný dokument, takže stiskněte tlačítko "Uložit" na panelu nástrojů (nebo použijte hlavní menu: Soubor->Uložit). Budete dotázáni na jméno nového dokumentu. Zadejte například "prvni_schema" a stiskněte tlačítko "Hotovo".



Obrázek 1 - Qucs - hlavní okno

Nyní budeme chtít vytvořit jednoduchou DC simulaci. Budeme chtít analyzovat obvod na obrázku 1. Aktivujte záložku “Komponenty” (4). Zde můžete vidět combo box, kde můžete vybrat skupinu komponent (zdroje, sondy,...) a pod jsou komponenty (většinou součástky, voltmetr, ampérmetr, atd) z vybrané skupiny komponent. Vyberte “Diskrétní komponenty” a klikněte na první symbol: “Odpor” (Pouze kliknout, není potřeba držet). Posouváním kurzoru myši do pracovní oblasti (6) přenesete rezistor na požadované místo. Kliknutím pravým tlačítkem myši můžete součástku otáčet. Levým kliknutím myši součástku umístíte do schématu. Opakováním tohoto procesu dodejte do schématu i zbývající komponenty, které jsou zakresleny na obrázku 1. Zdroj napětí můžete nalézt ve skupině komponent “zdroje”, symbol zem můžete vzít z “Diskrétní komponenty”, nebo z panelu nástrojů. Chtěná simulace je definována velkým čtvercem, který nalezneme ve skupině komponent “Simulace”. Pro změnu parametrů druhého rezistoru na něj jen dvakrát klikněte. Otevře se malé okno, ve kterém můžete měnit odpor. Zadejte “100 Ohm” do pravého políčka a stiskněte enter.

Pro propojení součástek stiskněte tlačítko “Vodič” na panelu nástrojů (nebo použijte hlavní menu: Vložit->Vodič). Kurzorem přejed'te na malý červený kroužek. Klikněte na něj pro započatí vodiče. Nyní klikněte na další červený kroužek jiné součástky, kterou chcete propojit. Součástky jsou nyní propojeny. Pokud chcete změnit směr vodiče, klikněte pravým tlačítkem myši ještě předtím, než nastavíte konec vodiče. Můžete také ukončit vodič bez jakéhokoliv připojení. Jen dvakrát klikněte levým tlačítkem myši.

Dále je potřeba označit vodič, kde Qucs vypočítá napětí. Klikněte na “Označení vodiče” na panelu nástrojů (nebo Vložit->Označení vodiče). Nyní klikněte na vybraný vodič. Objeví se okno. Sem napište například “Ux” (název je jen pro větší přehlednost, je na vás, jaký název si zvolíte) a klikněte na tlačítko “Hotovo”. Nyní by obvod měl vypadat jako obvod na obrázku 1.

Pro spouštění simulace stiskněte tlačítko “Simulovat” na panelu nástrojů (nebo použijte menu: Simulace->Simulovat). Otevře se okno, které ukazuje postup výpočtů. Po úspěšném dokončení simulace se objeví list, ve kterém se vypisují data z simulace. Obyčejně se tato operace provede tak rychle, že pouze vidíte krátký “záblesk”. Nyní máte možnost podívat se na výsledky simulace. Na levé straně je automaticky vybrána komponenta “Diagramy”. Klikněte na tlačítko “Tabulka” a přesuňte ji do pracovní oblasti. Tabulku přidáme do pracovní oblasti opět kliknutím levým tlačítkem myši. Otevře se okno, kde si můžeme vybrat, co chceme v novém diagramu zobrazit. V levé oblasti vidíme námi definované “Ux” a dále V1.1, což je označení pro zdroj napětí. Pro zobrazení Ux dvakrát klikněte na “Ux.V”. Tím se přidá proměnná do diagramu. Dále klikněte na “Hotovo”. Nyní můžete vidět výsledky simulace: 0.666667 voltů. Náhle, můžete se poplácat na rameno ;-)

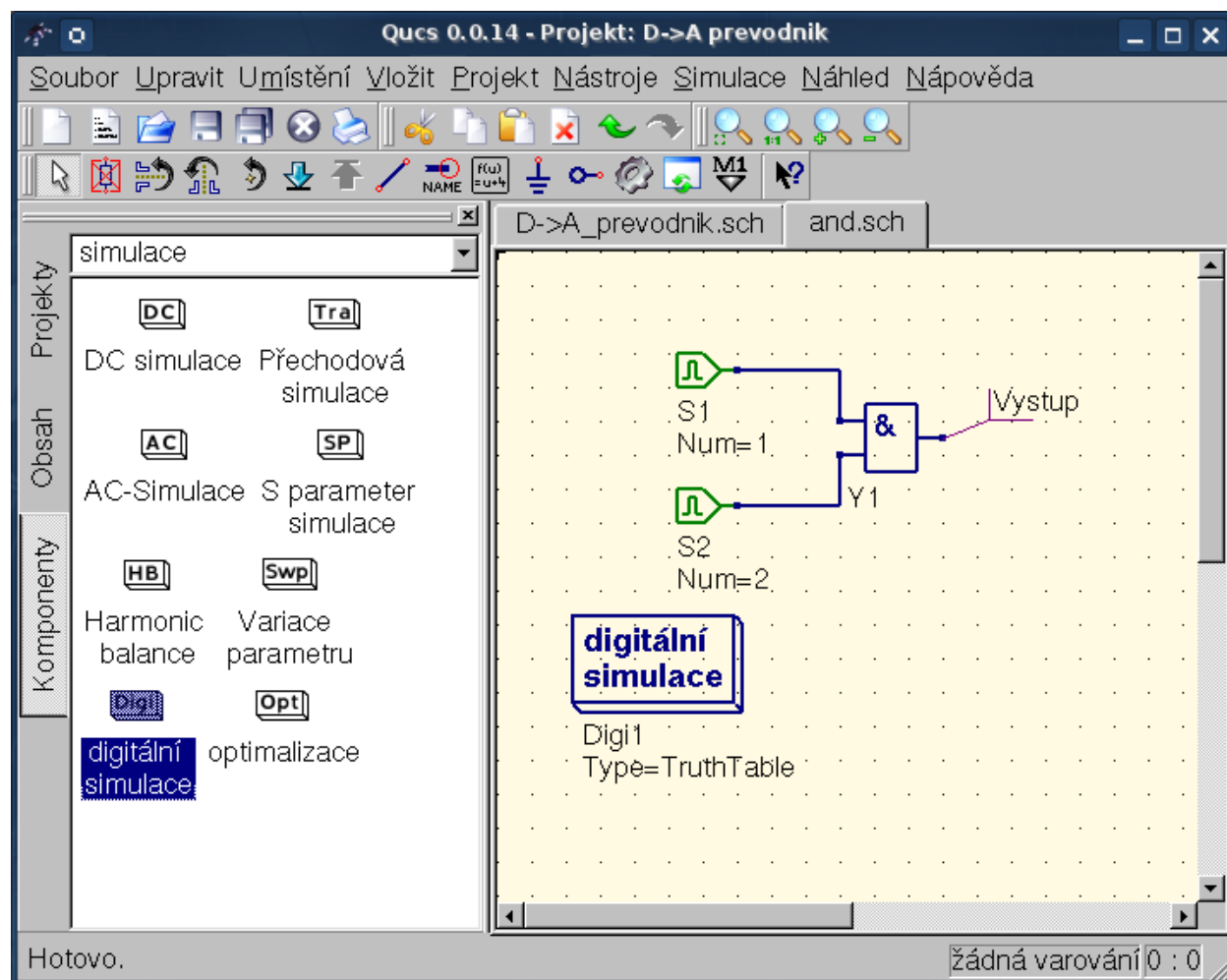
Začínáme s Digitálními simulacemi

Qucs obsahuje také grafické rozhraní umožňující digitální simulace. Tento manuál by vám měl ukázat “Jak nato”.

Pro digitální simulace Qucs používá FreeHDL program (<http://www.freehdl.seul.org>) , takže FreeHDL a GNU C++ kompilátor musí být nainstalován na vašem počítači.

Zde není příliš velký rozdíl mezi zprovozněním analogové, nebo digitální simulace. Takže pokud jste si přečetli manuál Začínáme s analogovými simulacemi, bude pro vás hračka zprovoznit digitální simulaci. Pokud chcete, aby hradlo AND vypadalo jako na ukázkovém obrázku, pak dvakrát klikněte na hradlo a u položky “Symbol” změňte “Old” na “DIN40900”. Nechme na programu, aby nám vypočítal pravdivostní tabulku z jednoduchého hradla AND. Vyberte ze skupiny komponent digitální komponenty v comboboxu. Čtverec “Digitální simulace” naleznete v kategorii skupin komponent “Simulace”.

Digitální zdroje S1 a S2 jsou vstupy. Uzel nadepsaný jako Output je výstup. Po provedení simulace se otevře okno, ve kterém se vypisují data ze simulace. Umístěte diagram Logická tabulka (Pravdivostní tabulka). Vyberte proměnnout Output. Nyní se nám zobrazí pravdivostní tabulka dvou-vstupového hradla AND. Gratuluji, první digitální simulace je hotová!



Obrázek 1 - Qucs - hlavní okno

Pravdivostní tabulka není jediné, co Qucs zvládne. Je zde také možnost poslat náhodný signál do obvodu a podívat se na na výstup v časovém diagramu. Pro to ale musíme změnit parametr simulace. Vraťte se ke schématu a změňte v Type (stačí jednou kliknout) TruthTable na TimeList. Nebo jednodušeji dvakrát klikněte na ikonu “Digitální simulace” a zde změňte “TruthTable” na “TimeList”. Ale během simulace musí být zadán další parametr. Digitální zdroje se nyní chovají jinak. Na jejich výstupech se náhodně mění sekvence bitů od prvního (definovaného) bitu (logická “0” nebo logická “1”). Dále je třeba nastavit seznam, který bude určovat kdy se bude měnit jejich logické stavy. Po “přečtení” tohoto seznamu se “přečte” ještě jednou a pak se program ukončí. Takže vytvořme generátor s taktovací frekvencí 1GHz se střídou 1:1, do listu se zapíše: 0.5ns, 0.5ns

Pro zobrazení výsledků typu této simulace je zde navržený diagram Časový diagram. Zde mohou být zobrazeny výsledky všech výstupů za sebou na řádkách. Takže, teď si můžete hrát ;-)

2.1 Komponenta VHDL soubor

Více složité a více univerzální simulace mohou být zrealizovány použitím komponenty “VHDL soubor”. Tuto komponentu můžete najít v skupině komponent “Digitální komponenty”. Přesto je však doporučeno, že VHDL soubor by měl být součástí projektu. Vraťte se zpět na “Obsah” a klikněte na název souboru. Po “vstupu” do schématu by se měla komponenta VHDL vložit

Poslední část ve VHDL souboru definuje prostředí, to jsou všechny vstupy a výstupy. Ty také musejí být deklarovány

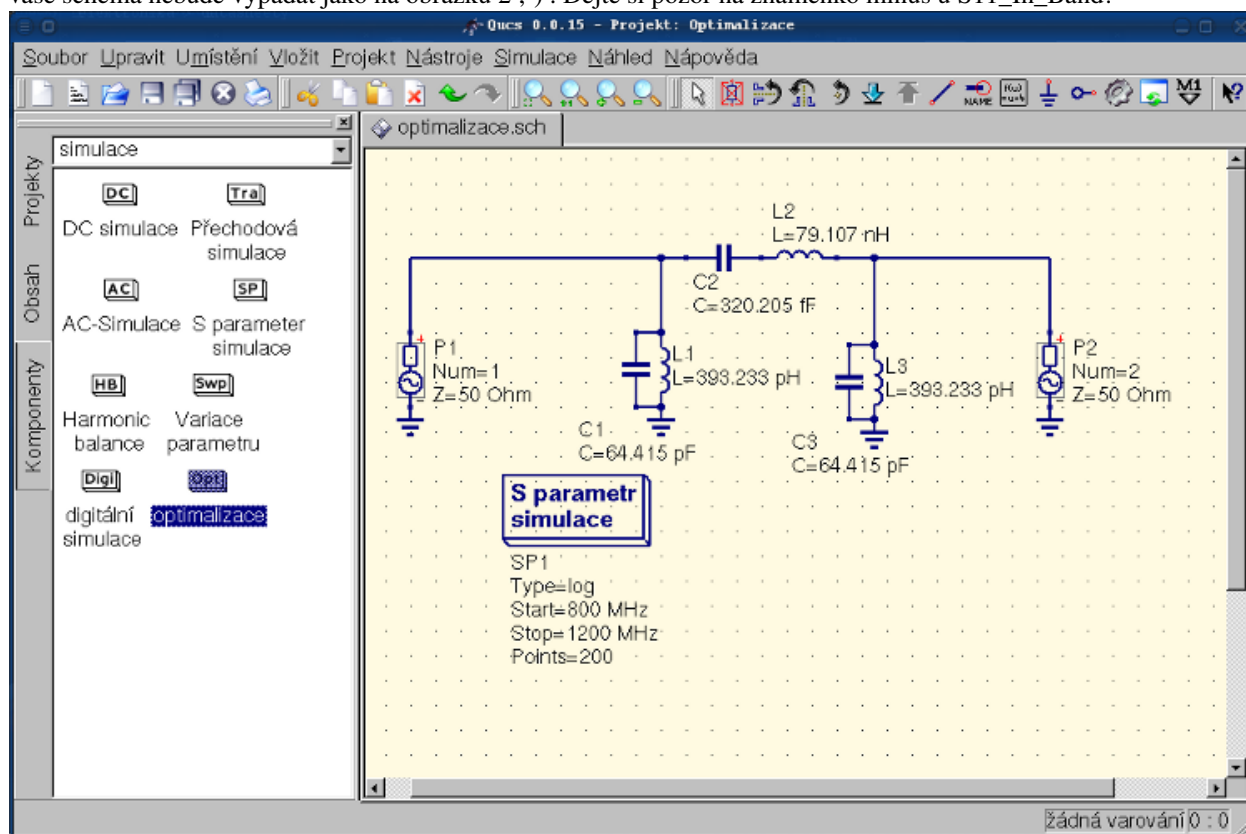
právě zde. Kontakty jsou rovněž ve schématu a mohou být propojeny se zbytkem obvodu. Během simulace je zdrojový kód VHDL souboru umístěn do nejvyšší úrovně VHDL souboru. Toto musí být šetrné kvůli jistým limitacím. Například celky názvů ve VHDL souboru musí být různé než jsou názvy vnořených obvodů. Po simulaci můžete zobrazit kompletní zdrojový kód stisknutím klávesy F6. Zobrazte si jej pokud se chcete o této proceduře dozvědět více.

Začínáme s optimalizací

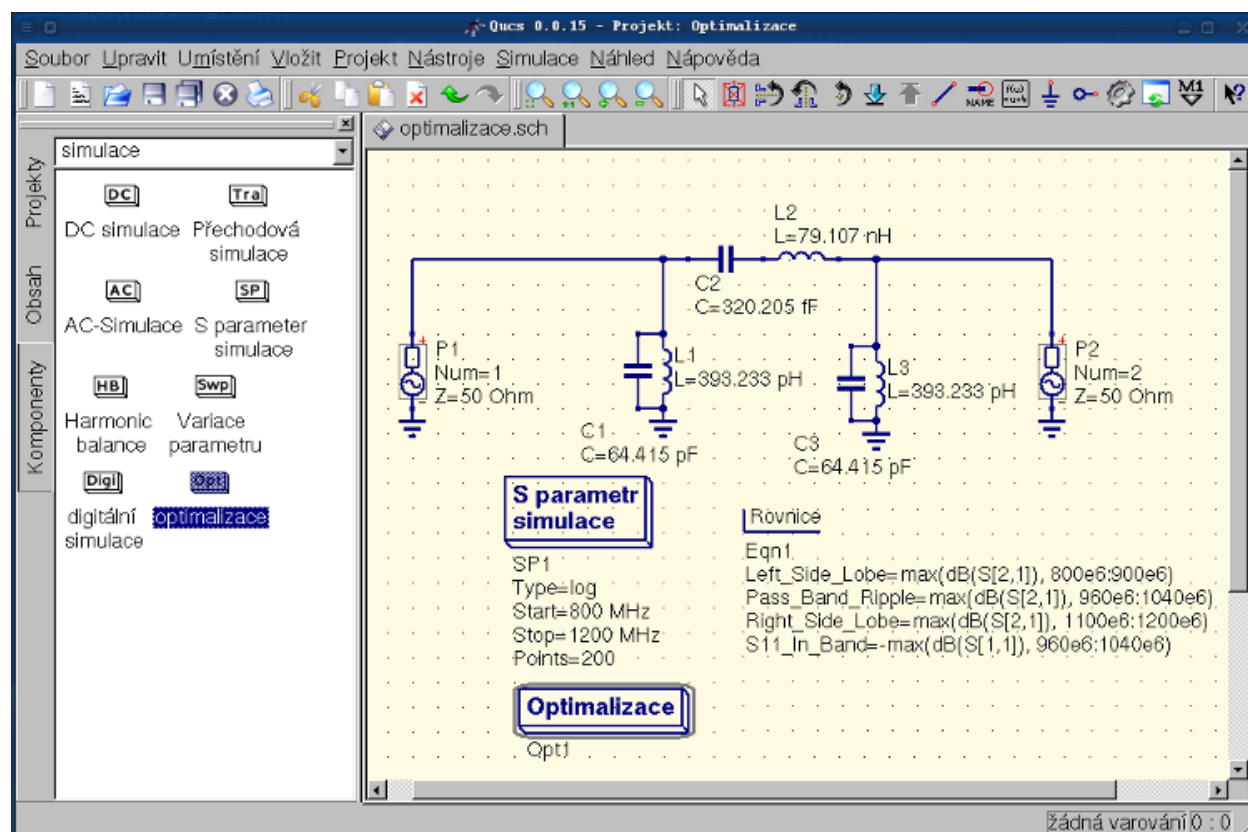
Pro optimalizaci obvodu používá Qucs utilitku ASCO (<http://asco.sourceforge.net/>). Zde předložím popis jak připravit vaše schéma, jak to celé spustit a jak porozumět výsledkům. Předtím než budete chtít využívat tuto funkci, musíte ASCO nainstalovat na váš počítač.

Obvod s minimalizací není nic víc, než program, který minimalizuje počet funkcí. Může to být každé časové zpoždění, nebo doba náběžné hrany v digitálních obvodech, nebo jakýkoli zdroj v analogových obvodech. Další možností je definovat optimalizační problém jako skládání funkcí, nebo jako v tomto případě, definovat “figure-of-merit”.

Pro nastavení optimalizace musíme dvě věci do schématu přidat: rovnici/rovnice (Vložit -> Vložit rovnici) a velký čtverec s nápisem “Optimalizace” (Ve skupině “Simulace”). Sestavte schéma podle obrázku 1 a hrajte si s Qucs, díkyd vaše schéma nebude vypadat jako na obrázku 2 ;-). Dejte si pozor na znaménko minus u S11_In_Band!

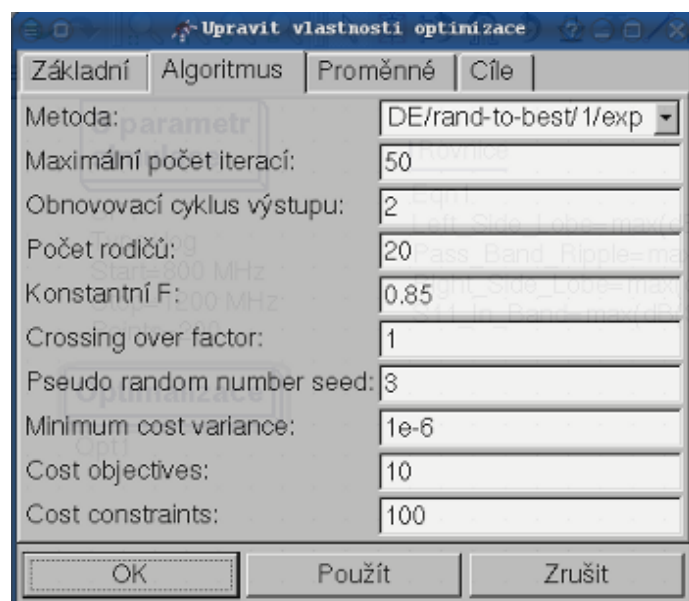


Obrázek 1 - Počáteční schéma



Obrázek 2 - Připravené schéma

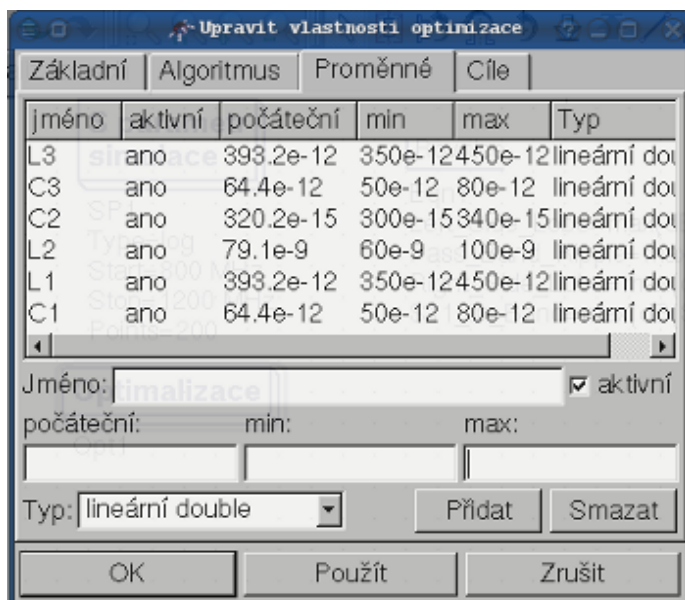
Nyní vyberte ze skupin komponent komponentu “Optimalizace”. Z existujících parametrů je třeba věnovat speciální pozornost “Maximální počet opakování”, “Konstanta F” a “Přechod přes faktor”. Někdy může optimalizace trvat jen chvíli, ale někdy i celkem dlouho.



Obrázek 3 - Optimalizace - dialog, možnosti algoritmu.

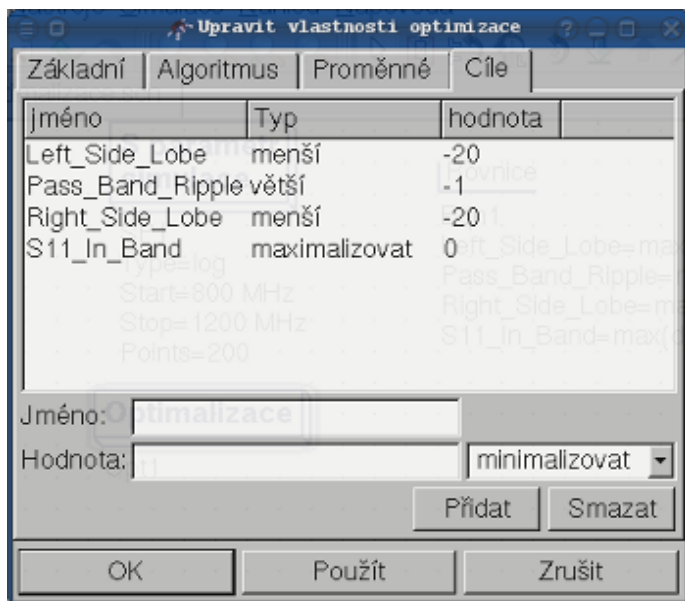
V tabulce Proměnné, které definují jednotlivé komponenty bude vybráno z rozsahu, jak je zobrazeno na obrázku 4.

Jména proměnných jsou ve shodě s jejich identifikátory umístěných ve vlastnostech komponent a NE jmen komponent.



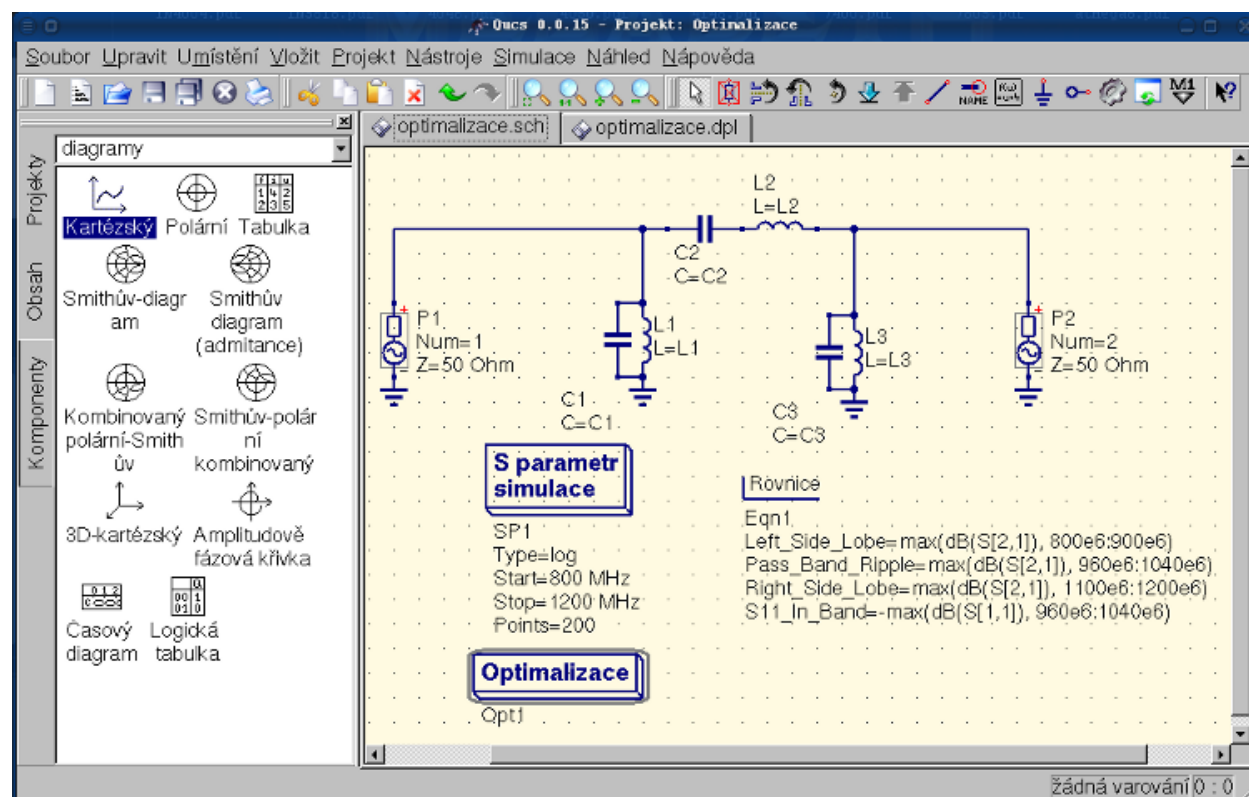
Obrázek 4 - Optimalizace - dialog, možnosti proměnných.

Konečně, se dostáváme do cíle kde objekty optimalizace (maximalizace, minimalizace) a omezení (menší, větší, rovno) jsou definovány. Poté ASCO je automaticky zkombinuje do jediné funkce a to je minimalizace.



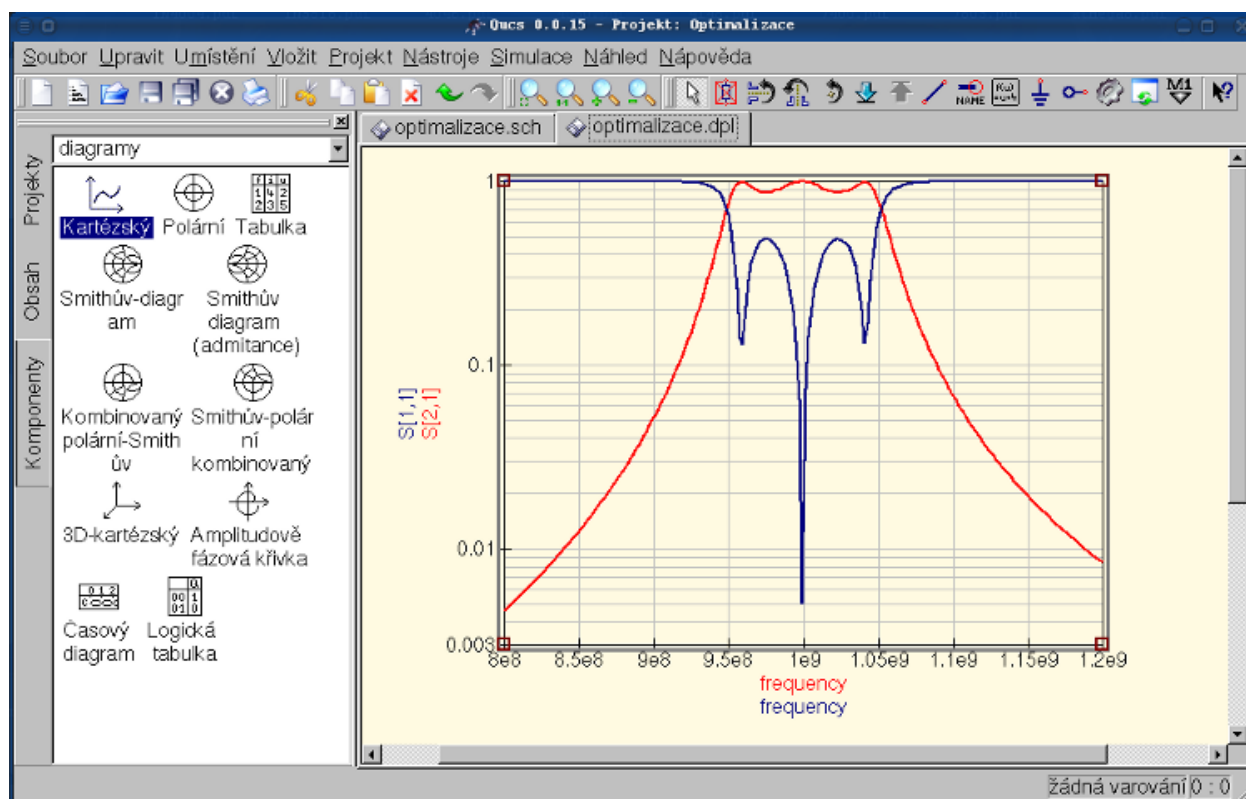
Obrázek 5 - Optimalizace - dialog, cílové možnosti.

Dalším krokem je změna schématu a definice které součásti obvodu budou optimalizovány. Výsledné schéma se zobrazeno na obrázku 6.



Obrázek 6 - Nové hlavní okno QUCS

Posledním krokem optimalizace je spuštění simulace stisknutím klávesy F2. Simulace může trvat několik sekund na moderním počítači. Nejlepší bude, když si výsledky necháte zobrazit do grafu. Vyberte si diagram kartézský. Přidejte do něj $S[2,1]$ a $S[1,1]$. Už jen nastavit barvy a tloušťku grafu (čím větší číslo, tím tlustější).



Obrázek 7 - Qucs - okno s výsledky.

Nejlepší nalezené obvody můžete najít v optimalizačním dialogu v záložce Proměnné. Zde jsou hodnoty pro každý z uvedených prvků (Obrázek 8).

Upravit vlastnosti optimalizace						
Základní		Algoritmus		Proměnné		Cíle
Jméno	aktivní	počáteční	min	max	Typ	
C1	ano	7.019977E-11	50e-12	80e-12	lineární doub	
L1	ano	3.593254E-10	350e-12	450e-12	lineární doub	
L2	ano	8.060568E-08	60e-9	100e-9	lineární doub	
C2	ano	3.142795E-13	300e-15	340e-15	lineární doub	
C3	ano	7.118198E-11	50e-12	80e-12	lineární doub	
L3	ano	3.590758E-10	350e-12	450e-12	lineární doub	

Jméno: ☒ aktivní
 počáteční: min: max:
 Typ: lineární double

Obrázek 8 - Nejlepší nalezené obvody.

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.

Krátký popis k ovládání

5.1 Hlavní ovládání

(platné pro všechny módy)

kolečko myši	Posouvá vertikálně pracovní oblast.
kolečko myši + klávesa Shift	Posouvá horizontálně pracovní oblast.
kolečko myši + klávesa Ctrl	Přiblíží, nebo oddálí pracovní oblast.
“přetáhnout” soubor do oblasti s dokumenty (viz Začínáme - Analogové obvody obrázek 1 (5))	Qucs se pokusí otevřít soubor jako schéma, nebo jako soubor s výstupními daty.

5.2 “Výběr”-Mód



(Menu: Upravit->Vybrat)

levé tlačítko myši	Vybere komponentu pod kurzorem. Pokud je zde několik komponent, můžete klikat tak dlouho, dokud nebude vybrána taková komponenta, kterou chcete. Pokud necháte tlačítko myši stisklé, můžete komponenty pod kurzorem přesouvat. Pokud chcete nastavit přesně pozici komponent, držte klávesu CTRL během přesouvání. Tím se vypne automatické přichytávání k mřížce. Pokud budete držet tlačítko myši v prázdném poli, vytvoříte obdélník. Po uvolnění myši všechny komponenty umístěné uvnitř obdélníku budou vybrané. Vybrané schéma, nebo kresba může být zvětšena, nebo zmenšena pomocí stisklého levého tlačítka myši na jednom z jeho rohů a přesunutím kurzoru se stisklým tlačítkem myši. Po kliknutí na text u komponenty, může být obsah upravován přímo. Po stisknutí Enteru automaticky skočí na další řádek. Pokud je na dalším řádku možnost si vybrat (například ze 2 možností), můžete tyto možnosti projít pomocí šipky nahoru a šipky dolů. Kliknutím v obvodu na uzel vstoupíte do “propojovacího módu”.
Levé tlačítko myši + klávesa Ctrl	Povolí vybrat více jak jednu komponentu. Například vyberete několik komponent, ale z tohoto výběru potřebujete mít dvě nevybrané. Kliknutím na vybranou komponentu ji odeberete z výběru. Tento mód je také platný pro výběr pomocí obdélníku (podívejte se o odstavce výše).
pravé tlačítko myši	Kliknutím na vodič vyberete pouze čist vodiče namísto celého vodiče.
Dojklik levým tlačítkem myši	Otevře se okno, ve kterém můžeme měnit vlastnost (značky vodičů, parametry komponent, atd.)

5.3 “Vložit komponentu”-Mód

(Kliknout na komponentu/diagram v levé oblasti)

levé tlačítko myši	Umístí novou komponentu do schématu.
pravé tlačítko myši	Otočí komponentu. (Neplatí pro diagramy.)

5.4 “Vodič”-Mód



(Menu: Vložit->Vodič)

levé tlačítko myši	Nastaví začátek/konec vodiče.
pravé tlačítko myši	Změní směr vodiče v rohách (první doleva/dopravas, nebo první nahoru/dolů).
Dojklík levým tlačítkem myši	Ends a wire without being on a wire or a port.

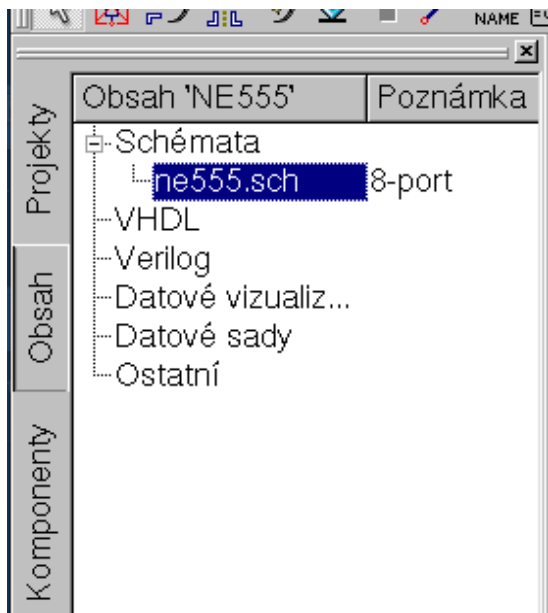
5.5 “Vložit”-Mód



(Menu: Upravit->Vložit)

levé tlačítko myši	Umístí komponentu do schématu (ze schránky).
pravé tlačítko myši	klik levým tlačítkem

5.6 Myš v záložce “Obsah”



dvojklik dvojklik	Otevře soubor. Otevře soubor. Zobrazí se menu: “Otevřít”	
Klávesnice	Přejmenovat	<ul style="list-style-type: none"> • otevře vybraný soubor • změní jméno vybraného souboru • Smaže vybraný soubor • smaže vybraný soubor i data s tím spojená (schémata, výstupy dat, nastavení)
	“Smazat”	
	“Smazat skupinu”	

5.7 Klávesnice

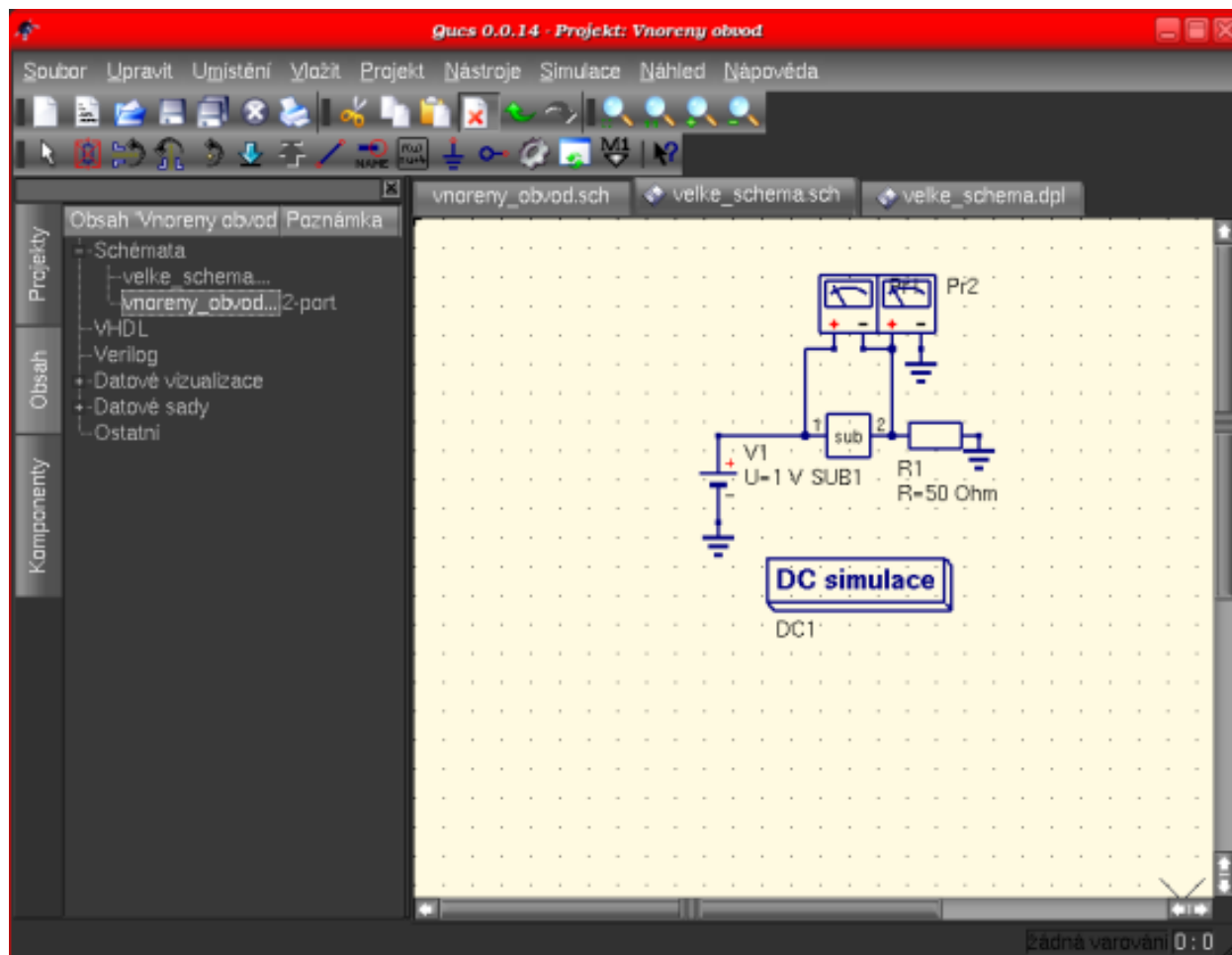
Mnoho akcí můžeme aktivovat/potvrdit pomocí klávesnice. Tyto klávesové zkratky můžete najít v hlavním menu. Další klávesové zkratky najdete v tomto seznamu:

Šipky doleva/doprava	Smaže vybrané komponenty, nebo se zapne mazací mód, pokud není nic vybráno.
Šipky nahoru/dolů	Změní pozici vybraných popisovačů v jejich grafech. Pokud není žádný popisovač vybrán, posune vybrané objekty. Pokud není žádná komponenta vybrána, bude se rolovat mezi záložkami dokumentů.
Tabulátor	Změní pozici vybraných popisovačů ve více rozměrných grafech. Pokud není žádný popisovač vybrán, po Changes the position of selected markers on more-dimensional graphs. Pokud není žádný popisovač vybrán, posune vybrané objekty. Pokud není žádná komponenta vybrána, bude se rolovat mezi záložkami dokumentů.
Nahoru	Skočí na další otevřený dokument. (souhlasně s další záložkou).

Práce s vnořenými obvody

Vnořené obvody se používají pro větší přehlednost ve schématu. Toto je velice užitečné při sestavování rozsáhlejších obvodů, kde se mnohokrát objevují stejné bloky součástek.

Nejprve vytvoříme vnořený obvod samotný. Vytváří se stejně jako každé jiné schéma. Nejdříve ale musíme program říct, kolik bude mít vlastně vnořený obvod pinů (pro začátek stačí třeba 2). Tyto piny najdete v liště nástrojů “Vložit připojení”, nebo v menu Vložit->Vložit připojení. Na ty červené kroužky připojte například dva rezistory 50 Ohmů zapojeny paralelně. Nyní schéma uložte (například vnoreny_obvod.sch). Všimněte si, že v záložce “Obsah” je u názvu schématu poznámka “2-port”. To znamená, že se jedná o vnořený obvod se dvěma piny (konektory). Ted’ už jen zbývá otevřít (popřípadě vytvořit nové) schéma a do něj vložit komponentu “Podobvod”. Klikněte na náš vnořený obvod následně dvakrát klikněte na schéma, kam chcete vložit vnořený obvod. Komponenta “Podobvod” by se měla sama nabídnout ke vložení. Nyní už zbývá vybrat pro “Podobvod” vybrat vhodné místo a napojit k němu další součástky. Vzorový příklad můžete vidět na obrázku 1. Nyní můžete obvod odsimulovat. Výsledky jsou naprosto shodné, jako kdyby součástky ze vnořeného obvodu byli připojené přímo.



Obrázek 1 - Obvod s vnořeným obvodem

Pokud vyberete komponentu “Podobvod” (jednou kliknout na symbol ve schématu) můžete se snadno a rychle dostat do vnořeného obvodu, pokud stisknete klávesu CTRL současně s klávesou I (Samozřejmě, že tato funkce je dostupná z lišty nástrojů i hlavního menu). Vrátit zpět se můžete pomocí CTRL-H (současně stisknout CTRL a klávesu H).

Pokud se vám nelíbí symbol komponenty pro vnořený obvod, můžete si nakreslit svůj vlastní symbol a vložit vlastní popisky komponenty na vaše oblíbené místo. Pouze vytvořte vnořený obvod a v menu klikněte na Soubor->Upravit symbol komponenty. Pokud ještě nemáte nakreslený symbol pro tento obvod, jednoduchý symbol je vytvořen automaticky. Nyní můžete upravovat symbol kreslením čar a vlastně čímkoliv, co najdete na levé straně programu. Nyní máte místo schématu nový symbol.

Jako všechny ostatní komponenty, má i komponenta “Podobvod” nějaké parametry. Pro definování vlastních parametrů se vraťte do schématu, kde je vložena komponenta “Podobvod” a dvakrát na ni klikněte. Objeví se okno, ve kterém můžete vyplnit parametry a popisy. Až budete připraveni, zavřete okno a uložte vnořený obvod. V každém schématu, kde je vnořený obvod umístěn, bude vlastnit nové parametry, které lze měnit jako ostatní komponenty.

6.1 Subcircuits with Parameters

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

Create a subcircuit:

- Create a new project

- New schematic (for subcircuit)
- Add a resistor, inductor, and capacitor, wire them in series, add two ports
- Save the subcircuit as RLC.sch
- Give value of resistor as 'R1'
- Add equation 'ind = L1',
- Give value of inductor as 'ind'
- Give value of capacitor as 'C1'
- Save
- 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)
- Save 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
- Simulovat
- 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.

Krátký popis matematických funkcí

Následující operace a funkce mohou být aplikovány v Qucs rovnicích. Parametry v závorkách “[]” jsou volitelné.

7.1 Operátory

7.1.1 Aritmetické Operátory

$+x$	Jednočlenný plus
$-x$	Jednočlenný mínus
$x+y$	Součet
$x-y$	Rozdíl
$x*y$	Násobení
x/y	Dělení
$x\%y$	Dělení beze zbytku (Modulo)
x^y	Mocnina

7.1.2 Logické Operátory

$!x$	Negace
$x\&y$	Logický součin
$x y$	Logický součet
x^y	Nonekvivalence
$x?y:z$	Zkratka pro podmíněné výrazy “Když x tak y jinak z” (if x then y else z)
$x==y$	Rovnost
$x!=y$	Nerovnost
$x<y$	Menší než
$x<=y$	Menší, nebo rovno
$x>y$	Větší
$x>=y$	Větší, nebo rovno

7.2 Matematické funkce

7.2.1 Vektory a matice: Vytváření

<code>eye (n)</code>	Vytvoří $n \times n$ identickou matici
<code>length (y)</code>	Returns the length of the given vector
<code>linspace (from, to, n)</code>	Vytvoří reálný vektor s n lineárně rozloženou komponentou mezi od a do, zahrnující oboje
<code>logspace (from, to, n)</code>	Vytvoří reálný vektor s n logaritmičsky rozloženou komponentou mezi od a do, zahrnující oboje

7.2.2 Vektory a matice: Základní maticové funkce

<code>adjoint (x)</code>	Adjungovaná matice x (Přenesená a složená)
<code>det (x)</code>	Determinant matice x
<code>inverse (x)</code>	Obrácená matice x
<code>transpose (x)</code>	Přenesená matice x (řady a sloupce jsou prohozeny)

7.2.3 Absolutní hodnota, důležité u komplexních čísel

<code>abs (x)</code>	Absolutní hodnota, důležité u komplexních čísel
<code>angle (x)</code>	Fáze úhlu v radiánech komplexního čísla. Synonymum pro <code>arg()</code>
<code>arg (x)</code>	Komplexně sdružené číslo
<code>conj (x)</code>	Komplexně sdružené číslo
<code>deg2rad (x)</code>	Převádí fázi ze stupňů na radiány
<code>hypot (x, y)</code>	Euklidova distantní funkce
<code>imag (x)</code>	Velikost komplexního čísla
<code>mag (x)</code>	Absolutní hodnota vektoru na čtverec
<code>norm (x)</code>	Fázový úhel komplexního čísla (ve stupních)
<code>phase (x)</code>	Fázový úhel komplexního čísla (ve stupních)
<code>polar (m, p)</code>	Mění protichůdné souřadnice (velikost, fáze p) na komplexní číslo
<code>rad2deg (x)</code>	Reálná část komplexního čísla
<code>real (x)</code>	Reálná část komplexního čísla
<code>sign (x)</code>	Funkce signum
Druhá mocnina čísla	Druhá mocnina čísla
<code>sqrt (x)</code>	Druhá odmocnina
<code>unwrap (p[, tol], 0)</code>	Unwraps the angle (opravdu netuším jak to přeložit) p (v radiánech – základně ; step je 2π) s použitím volitelné hodnoty tolerance tol (základně je π)

7.2.4 Základní matematické funkce: Exponenciální a logaritmické funkce

<code>exp (x)</code>	Limitovaná exponenciální funkce
<code>limexp (x)</code>	Dekadický logaritmus
<code>log10 (x)</code>	Binární logaritmus
<code>log2 (x)</code>	Binární logaritmus
<code>ln (x)</code>	Přirozený logaritmus (základ e)

7.2.5 Základní matematické funkce: Trigonometrie

<code>cos (x)</code>	Funkce cosinus
<code>cosec (x)</code>	Kosekans
<code>cot (x)</code>	Funkce kotangens
<code>sec (x)</code>	Sekans
<code>sin (x)</code>	Funkce tangens
<code>tan (x)</code>	Funkce tangens

7.2.6 Základní matematické funkce: Opačné trigonometrické funkce

<code>arccos (x)</code>	Arc cosinus
<code>arccosec (x)</code>	Arc kotangens
<code>arccot (x)</code>	Arc sekans
<code>arcsec (x)</code>	Arc sinus
<code>arcsin (x)</code>	Arc sinus
<code>arctan (x[, y])</code>	Arc tangents

7.2.7 Základní matematické funkce: Hyperbolické funkce

<code>cosh (x)</code>	Hyperbolický kosekans
<code>cosech (x)</code>	Hyperbolický cotangent
<code>coth (x)</code>	Hyperbolický sekans
<code>sech (x)</code>	Hyperbolický sinus
<code>sinh (x)</code>	Hyperbolický sinus
<code>tanh (x)</code>	Hyperbolický tangents

7.2.8 Základní matematické funkce: Opačné hyperbolické funkce

<code>arcosh (x)</code>	Opačná funkce k hyperbolickému cosinu
<code>arcosech (x)</code>	Opačná funkce k hyperbolickému kosekanu
<code>arcoth (x)</code>	Opačná funkce k hyperbolickému cotangetu
<code>arsech (x)</code>	Opačná funkce k hyperbolickému sekanu
<code>arsinh (x)</code>	Opačná funkce k hyperbolickému sinu
<code>artanh (x)</code>	Opačná funkce k hyperbolickému tangentu

7.2.9 Zaokrouhlí na další vyšší celé číslo

<code>ceil (x)</code>	Zaokrouhlí na další vyšší celé číslo
<code>fix (x)</code>	Zaokrouhlí na další nižší celé číslo
<code>floor (x)</code>	Zaokrouhlí na další nižší celé číslo
<code>round (x)</code>	Zaokrouhlí na nejbližší celé číslo

7.2.10 Základní matematické funkce: Speciální matematické funkce

besseli0(x)	Modifikovaná Besselova funkce
besselj(n, x)	Modifikovaná Besselova funkce prvního druhu a n-tého druhu
bessely(n, x)	Bessel function of second kind and n-th order
erf(x)	Chybná funkce
erfc(x)	Invertovaná chybná funkce
erfinv(x)	Invertovaná chybná funkce
erfcinv(x)	Inverzní doplňková chybná funkce
sinc(x)	Synchronizační funkce (sin(x)/x nebo 1 na x = 0)
step(x)	avg(x[,range])

7.2.11 Rozbor dat: Základní statistiky

avg(x[, range])	Aritmetický průměr vektorových prvků; pokud je dán rozsah pak x must have a single data dependency
cumavg(x)	Souhrnný průměr vektorových prvků
max(x, y)	Vrátí větší číslo z x a y
max(x[, range])	Maximální hodnota ve vektoru. Pokud je dán rozsah, pak x must have a single data dependency
min(x, y)	Vrátí menší číslo z x a y
min(x[, range])	Minimální hodnota ve vektoru; Pokud je dán rozsah, pak x must have a single data dependency
rms(x)	Running average of vector elements
runavg(x)	Standard deviation of vector elements
stddev(x)	Variance of vector elements
variance(x)	Náhodné číslo mezi 0,0 a 0,1
random()	Náhodné číslo mezi 0.0 a 0.1
srandom(x)	Give random seed

7.2.12 Cumulative product of vector elements

cumprod(x)	Cumulative sum of vector elements
cumsum(x)	Cumulative sum of vector elements
interpolate(f, xval)	Equidistant spline interpolation of real function vector f(x) using n equidistant datapoints; the latter can be omitted and defaults to a reasonable value
prod(x)	Sum of vector elements
sum(x)	Sum of vector elements
xvalue(f, yval)	Returns the x-value which is associated with the y-value nearest to a specified y-value yval in a given vector f; therefore the vector f must have a single data dependency
yvalue(f, xval)	Returns the y-value of the given vector f which is located nearest to the x-value xval; therefore the vector f must have a single data dependency

7.2.13 Data Analysis: Differentiation and Integration

ddx(expr, var)	Derives mathematical expression expr with respect to the variable var
diff(y, x[, n])	Differentiate vector y with respect to vector x n times. If n is omitted it defaults to n = 1
integrate(x, h)	Integrate vector x numerically assuming a constant step-size h

7.2.14 Data Analysis: Signal Processing

<code>dft(x)</code>	Discrete Fourier Transform of vector x
<code>fft(x)</code>	Fast Fourier Transform of vector x
<code>fftshift(x)</code>	Shuffles the FFT values of vector x in order to move the frequency 0 to the center of the vector
<code>Freq2Time(V, f)</code>	Inverse Discrete Fourier Transform of function $V(f)$ interpreting it physically
<code>idft(x)</code>	Inverse Discrete Fourier Transform of vector x
<code>ifft(x)</code>	Inverse Fast Fourier Transform of vector x
<code>kbd(x[, n])</code>	Kaiser-Bessel derived window
<code>Time2Freq(v, t)</code>	Discrete Fourier Transform of function $v(t)$ interpreting it physically

7.3 Electronics Functions

7.3.1 Unit Conversion

<code>dB(x)</code>	dB value
<code>dbm(x)</code>	Convert voltage to power in dBm
<code>dbm2w(x)</code>	Convert power in dBm to power in Watts
<code>w2dbm(x)</code>	Convert power in Watts to power in dBm
<code>vt(t)</code>	Thermal voltage for a given temperature t in Kelvin

7.3.2 Reflection Coefficients and VSWR

<code>rtoswr(x)</code>	Converts reflection coefficient to voltage standing wave ratio (VSWR)
<code>rtoz(x[, zref])</code>	Converts reflection coefficient to admittance; by default reference $zref$ is 50 ohms
<code>rtoz(x[, zref])</code>	Converts reflection coefficient to impedance; by default reference $zref$ is 50 ohms
<code>ytor(x[, zref])</code>	Converts admittance to reflection coefficient; by default reference $zref$ is 50 ohms
<code>ztor(x[, zref])</code>	Converts impedance to reflection coefficient; by default reference $zref$ is 50 ohms

7.3.3 N-Port Matrix Conversions

<code>stos(s, zref[, z0])</code>	Converts S-parameter matrix to S-parameter matrix with different reference impedance(s)
<code>stoy(s[, zref])</code>	Converts Y-parameter matrix to S-parameter matrix
<code>stoz(s[, zref])</code>	Converts Y-parameter matrix to Z-parameter matrix
<code>twoport(m, from, to)</code>	Converts a two-port matrix from one representation into another, possible values for $from$ and to are 'Y', 'Z', 'H', 'G', 'A', 'S' and 'T'.
<code>ytoz(y[, z0])</code>	Converts Z-parameter matrix to Y-parameter matrix
<code>ytoz(y)</code>	Converts Y-parameter matrix to Z-parameter matrix
<code>ztos(z[, z0])</code>	Converts Z-parameter matrix to S-parameter matrix
<code>ztos(z)</code>	Converts Z-parameter matrix to Y-parameter matrix

7.3.4 Amplifiers

GaCircle(s, Ga[, arcs])	Circle(s) with constant available power gain Ga in the source plane
GpCircle(s, Gp[, arcs])	Circle(s) with constant operating power gain Gp in the load plane
Mu(s)	Mu stability factor of a two-port S-parameter matrix
Mu2(s)	Rollet stability factor of a two-port S-parameter matrix
NoiseCircle(Sopt, Fm, NoiseFigure[, arcs])	Generates circle(s) with constant Noise Figure(s) F. Arcs specifies the angles in degrees created by e.g. linspace(0, 360, 100). If Arcs is a number it specifies the number of equally spaced circle segments, if it is omitted this number defaults to a reasonable value
PlotVs(data, dep)	Returns a data item based upon vector or matrix vector data with dependency on a given vector dep, e.g. PlotVs(Gain, frequency/1e9)
Rollet(s)	Rollet stability factor of a two-port S-parameter matrix
StabCircleL(s[, arcs])	Stability circle in the load plane
StabCircleS(s[, arcs])	Stability circle in the source plane
StabFactor(s)	Stability factor of a two-port S-parameter matrix. Synonym for Rollet()
StabMeasure(s)	Stability measure B1 of a two-port S-parameter matrix

7.4 Nomenclature

7.4.1 Ranges

LO:HI	Range from LO to HI
:HI	Up to HI
LO:	From LO
:	No range limitations

7.4.2 Matrices and Matrix Elements

M	The whole matrix M
M[2, 3]	Element being in 2nd row and 3rd column of matrix M
M[:, 3]	Vector consisting of 3rd column of matrix M

7.4.3 Vector

2.5	Real number
1.4+j5.1	Complex number
[1, 3, 5, 7]	Vector
[11, 12; 21, 22]	Matrix

7.4.4 tera, * 1e+12

E	exa, 1e+18
P	peta, 1e+15
T	tera, 1e+12
G	giga, 1e+9
M	mega, 1e+6
k	kilo, 1e+3
m	milli, 1e-3
u	micro, 1e-6
n	nano, 1e-9
p	pico, 1e-12
f	femto, 1e-15
a	atto, 1e-18

7.4.5 Name of Values

$S[1,1]$	S-parameter value
<i>nodename.V</i>	AC current through component <i>name</i>
<i>name.I</i>	AC noise voltage at node <i>nodename</i>
<i>nodename.v</i>	AC voltage at node <i>nodename</i>
<i>name.i</i>	Transient voltage at node <i>nodename</i>
<i>nodename.vn</i>	Transient current through component <i>name</i>
<i>name.in</i>	AC noise current through component <i>name</i>
<i>nodename.Vt</i>	Imaginary unit (“square root of -1”)
<i>name.It</i>	Transient current through component <i>name</i>

Note: All voltages and currents are peak values. Note: Noise voltages are RMS values at 1 Hz bandwidth.

7.5 Constants

i, j	Boltzmann constant = 1.38065e-23 J/K
pi	$4 \cdot \arctan(1) = 3.14159\dots$
e	Euler = 2.71828...
kB	Boltzmann constant = 1.38065e-23 J/K
q	Elementary charge = 1.6021765e-19 C

Seznam speciálních symbolů

Poznámka: Zda se vám správně zobrazí daný symbol, či ne závisí na typu písma, které používá Qucs! Proto si prosím volte takový typ písma, který dokáže zobrazit dané znaky.

Poznámka: Zda se vám správně zobrazí daný symbol, či ne závisí na typu písma, které používá Qucs! Proto si prosím volte takový typ písma, který dokáže zobrazit dané znaky.

Malá řecká písmena

LaTeX tag	Unicode	Popis
<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

Velká řecká písmena

LaTeX tag	Unicode	Popis
<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

Matematické symboly

LaTeX tag	Unicode	Popis
<code>\cdot</code>	0x00B7	Mnohonásobné body (střed bodu)
<code>\times</code>	0x00D7	Mnohonásobný křížek
<code>\pm</code>	0x00B1	Znak plus mínus
<code>\mp</code>	0x2213	Znak mínus plus
<code>\partial</code>	0x2202	Symbol pro částečné derivování
<code>\nabla</code>	0x2207	Operátor nabla
<code>\infty</code>	0x221E	Symbol nekonečno
<code>\int</code>	0x222B	Symbol integrátu
<code>\approx</code>	0x2248	Symbol pro aproximaci
<code>\neq</code>	0x2260	Znaménko nerovnosti
<code>\in</code>	0x220A	Symbol “obsaženo v”
<code>\leq</code>	0x2264	Znaménko menší, nebo rovno
<code>\geq</code>	0x2265	Znaménko větší než
<code>\sim</code>	0x223C	Znaménko pro přímou úměru (střední Evropa)
<code>\propto</code>	0x221D	Znaménko pro přímou úměru (Amerika)
<code>\diameter</code>	0x00F8	Znaménko pro průměr kružnice (také znaménko pro aritmetický průměr)
<code>\onehalf</code>	0x00BD	Jedna polovina
<code>\onequarter</code>	0x00BC	První mocnina
<code>\twosuperior</code>	0x00B2	Druhá mocnina
<code>\threesuperior</code>	0x00B3	Třetí mocnina
<code>\ohm</code>	0x03A9	jednotka pro rezistivitu (velké řecké písmeno Omega)

Laděné obvody

Vytváření laděných obvodů je občas potřeba v mikrovlnné technologii. Qucs může toto dělat automaticky. Zde jsou nezbytné kroky:

Vykonat simulaci s S-parametrem aby se vypočítal činitel odrazu.

Umístit diagram a zobrazit činitel odrazu (Např.: $S[1,1]$ pro port 1, $S[2,2]$ pro port 2 atd.)

Nastavit tvůrce grafu a nastavit požadovanou frekvenci.

Kliknout pravým tlačítkem myši na tvůrce grafu a vyberte “power matching” v právě objeveném menu.

Objeví se dialog. Zde si můžete přizpůsobit hodnoty. Např. referenční impedance se může změnit o 50 Ohmů.

Po kliknutí na “vytvořit” se vrátíte zpátky do schématu a pomocí kurzoru myši můžete umístit vlastní laděný obvod.

Levá strana laděného obvodu je vstup a na pravé straně musí být zapojen obvod.

If the marker points to a variable called “Sopt”, the menu shows the option “noise matching”. Note that the only different to “power matching” is the fact that the conjugate complex reflexion coefficient is taken. So if the variable has another name, noise matching can be chosen by re-adjusting the values in the dialog.

Dialog pro laděné obvody můžete také najít přímo v menu (Nástroje -> Přizpůsobovací obvod) a nebo použít klávesovou zkratku (<CTRL-5>). Potom ale veškeré hodnoty musí být zadány ručně.

9.1 2-vývodové laděné obvody

Pokud název proměnné v tvůrci textu je jako S-parametr, pak tato volba existuje pro současný laděný vstup a výstup dvou-vývodového obvodu. Toto pracuje podobně jako již bylo výše zmíněno. Ve výsledku je vše ve dvou L-obvodech: Levý uzel je pro propojení s propojkou č. 1. Pravý uzel je pro propojení s propojkou č. 2 a dva uzly vprostřed jsou pro připojení dvou-vývodového obvodu.

Instalované soubory

Qucs potřebuje několik programů. Ty jsou nainstalovány během instalace. Cesty k Qucs se určují během instalace (configure skript). Následující vysvětlivky předpokládají, že program je nainstalován defaultně do defaultních adresářů (/usr/local/).

- /usr/local/bin/qucs - GUI - grafické rozhraní
- /usr/local/bin/qucsator - simulátor (aplikace v konzoli)
- /usr/local/bin/qucsedit - jednoduchý textový editor
- /usr/local/bin/qucshelp - malý program zobrazující nápovědu
- /usr/local/bin/qucstrans - program pro přenos výpočtů a parametrů
- /usr/local/bin/qucsfilter - program syntetizující filtry obvodů
- /usr/local/bin/qucsconv - konvertor formátu souborů (aplikace v konzoli)

Všechny programy jsou samostatné aplikace a mohou být spuštěny samostatně. Hlavní program (GUI)

- spustí qucsator když provádí simulaci,
- spustí qucsedit když zobrazuje textové soubory,
- spustí qucshelp když zobrazuje nápovědu,
- spustí qucstrans když spustíme tento program z menu “Nástroje”,
- spustí qucsfilter když spustíme tento program z menu “Nástroje”,
- spustí qucsconv když umístíte SPICE komponentu a když vykonává simulaci s SPICE komponentou.

Krom toho, následující adresáře jsou vytvořeny během instalace:

- /usr/local/share/qucs/bitmaps - obsahuje všechny bitmapy (ikony atd.)
- /usr/local/share/qucs/docs - obsahuje HTML dokumenty, které pak používá nápověda
- /usr/local/share/qucs/lang - obsahuje soubory s překladem

10.1 Příkazová řádka - argumenty

```
qucs [soubor1 [soubor2 ...]]
```

```
qucsator [-b] -i netlist -o dataset (b = ukazatel stavu - progress bar)
```

```
qucsedit [-r] [soubor] (r = pouze pro čtení)
```

qucs-help (bez argumentů)

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

Popis k formátu souborů

Tento dokument popisuje formát souborů. Tento formát je používán pro schémata (obvykle s příponou `.sch`) a pro soubory, které zobrazují výstupní data (obvykle s příponou `.dpl`). Následující text názorně ukazuje příklad souboru se schématem.

```
<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>
```

Každý řádek obsahuje mnoho sekcí. Každá je vysvětlena níže. Každá řádka neobsahuje více jak jeden blok informací které začínají znakem `<` a končí znakem `>`.

11.1 Vlastnosti (Properties)

První část začíná s `<Properties>` a končí `</Properties>`. Tento blok obsahuje vlastnosti souboru dokumentu. Každá řádka je volitelná (neí třeba vše definovat). Následující vlastnosti jsou podoprovány:

- `<View=x1, y1, x2, y2, scale, xpos, ypos>` první čtyři čísla udávají pozici okna se shématem. Je to současná velikost tohoto okna a pozice levého horního rohu (poslední dvě čísla).

- `<Grid=x, y, on>` udává rozestup v mřížce v pixelech (první dvě čísla) a jestli je zapnut (poslední číslo je 1), nebo vypnut (poslední číslo je 0).
- `<DataSet=name.dat>` Do tohoto souboru se ukládají výsledky ze simulace.
- `<DataDisplay=name.dpl>` Do tohoto souboru se ukládají další informace o simulaci.
- `<OpenDisplay=yes>` obsahuje 1 pokud se stránka `DataDisplay` má automaticky otevřít po simulaci. V opačném případě obsahuje 0.

11.2 Symbol

Začíná znaky `<Symbol>` a končí `</Symbol>`. Obsahuje grafické součásti, které tvoří schématický symbol pro soubor. Toto je často používáno pro soubory schémat, které bývají později použity jako vnořené obvody.

11.3 Components (Komponenty)

Začíná znaky `<Components>` a končí `</Components>`. Obsahuje komponenty obvodů ve schématech. Formát je následující:

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

- `type` - identifikuje komponenty. Např.: R jako rezistor, C jako kapacitu.
- `name` - toto je zcela jedinečný identifikátor ve schématu. Např.: R1 pro první rezistor.
- `active` - pokud je zde 1, znamená to, že komponenta je aktivní. Například je použita v simulaci. Pokud je zde 0, je neaktivní.
- `"x y"` - Tyto dvě čísla určují, polohu komponenty (resp. kde se bude nacházet její střed).
- `"xtext ytext"` - Tato čísla určují polohu textu, který slouží jako popis pro určitou komponentu (resp. určuje, kde se bude nacházet horní levý roh popisku). Tyto údaje udávají vzdálenost od středu komponenty.
- `mirrorX rotate` - Následující dvě čísla definují zrcadlení podle osy x (1 pro zrcadlení, 0 nezrcadlí se) a rotaci ve stupních (proti směru hodinových ručiček).
- `Value1 visible` - Zde se udává hodnota komponenty (v uvozovkách). Pokud je za ní 1, pak bude ve schématu zobrazena. Pokud bude hodnota 0, pak nebude ve schématu zobrazena.

11.4 Vedení

Začíná `<Wires>` a končí `</Wires>`. Obsahuje informace o vedení, které spojuje jednotlivé komponenty (co spojuje, název, atd.). Formát je následující:

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

- `"x1 y1 x2 y2"` - Tyto čtyři čísla určují počátek (x1, y1) a konec (x2, y2) vodiče. Veškeré vodiče musí být ve vodorovné, nebo horizontální poloze (tzn. že budou obě xové, nebo obě ypsylonové souřadnice stejné).
- `"label"` - Tato proměnná nastavuje popis. Pokud je prázdná, znamená to, že vodiči nedal uživatel žádný název.
- `"xlabel ylabel"` - Další dvě čísla jsou xové a ypsylonové souřadnice popisku. Pokud jsou zde nuly, znamená to, že popisek neexistuje.

- “dlabel” - Číslo určuje vzdálenost mezi počátečním bodem vodiče a popiskou vodiče.
- “node set” - Text v uvozovkách udává jméno uzlu vodiče. Např.: počáteční napětí na tomto uzlu je právě jméno uzlu tohoto vodiče, pak se engine pokusí najít řešení. Pokud je tato položka prázdná, znamená to, že uživatel nenastavil jméno uzlu pro daný vodič.

11.5 Diagramy

Začíná <Diagrams> a končí </Diagrams>. Obsahuje diagramy s jejich grafy a značkami. The line format is as follows (line break not allowed):

```
<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">
```

- “x y” - Tyto čísla určují pozici spodního levého rohu.
- “width height” - Následující čísla udávají šířku a výšku diagramu.
- “grid” - Pokud je zde 1, pak bude zobrazena mřížka. Pokud zde bude 0, pak mřížka nebude zobrazena.
- “gridcolor” - Zde je udána 24. bitová barva v hexadecimální RGB hodnotě. Např.: #FF0000 je červená.
- “gridstyle” - Určuje styl mřížky.
- Zbylá čísla určují jak “osekáme” logaritmickou stupnici.

11.6 Obrazce

Začíná znaky <Paintings> a končí </Paintings>. Obsahuje obrazce, které jsou ve schématu.

Technický popis týkajícího se simulace

je dostupný na: <http://qucs.sourceforge.net/tech/technical.html> (anglicky)

Příklady shémat

jsou dostupné na: <http://qucs.sourceforge.net/download.html#example> (anglicky)