Симулятор электронных схем Qucs-S: обзор новшеств версии 2.1.0

**6 мин**

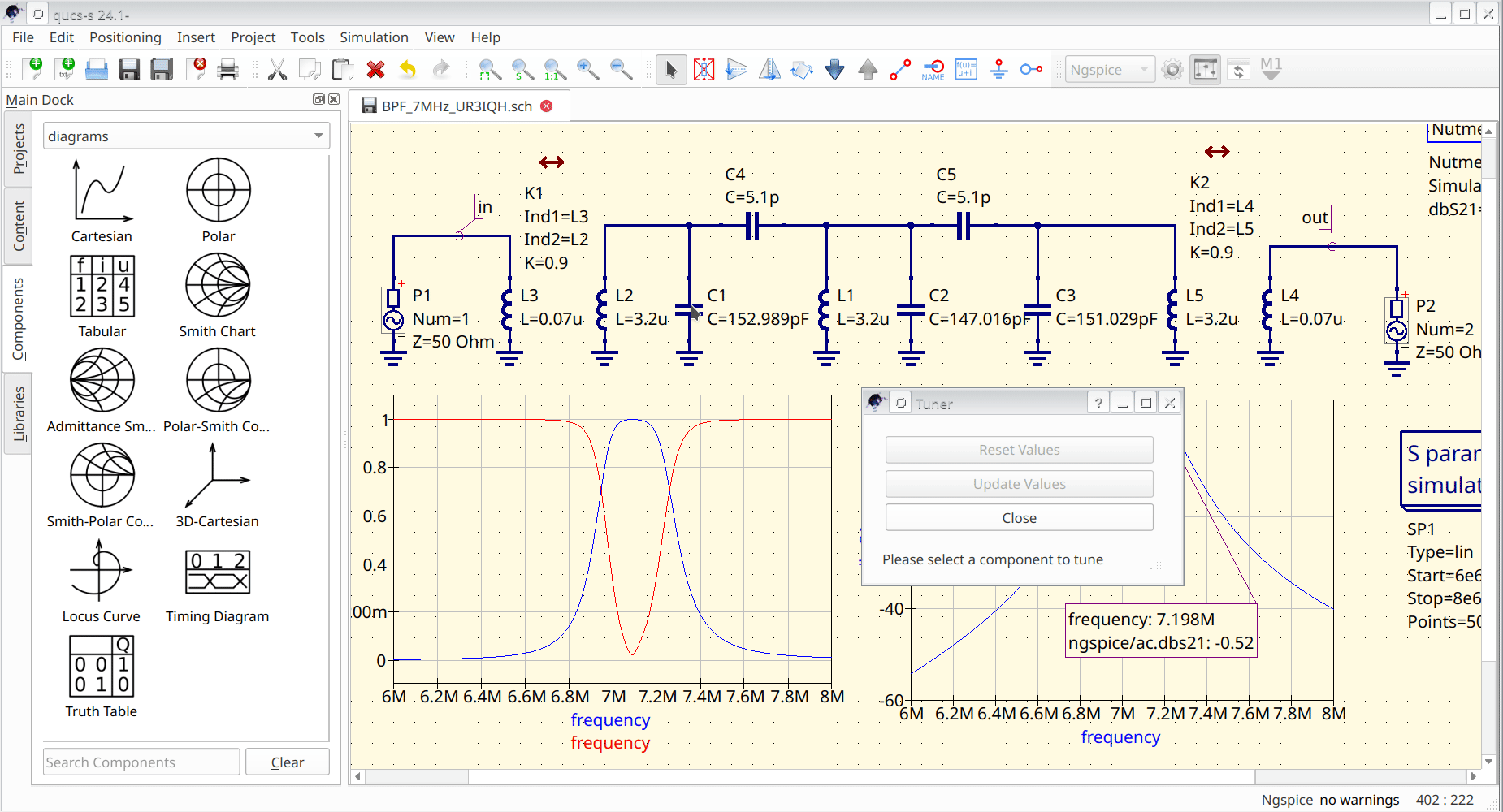
**10K**

[Open source\*](https://habr.com/ru/hubs/open_source/)[CAD/CAM\*](https://habr.com/ru/hubs/cad_cam/)[DIY или Сделай сам](https://habr.com/ru/hubs/DIY/)

Qucs‑S является программой с открытым исходным кодом для моделирования электронных схем. Qucs‑S кроссплатформенный (поддерживаются Linux и Windows) и написан на С++ с использованием набора библиотек Qt. Для работы Qucs рекомендуется использовать также открытый движок моделирования Ngspice. Первый релиз Qucs, на котором основан Qucs‑S, состоялся в 2003 году. В декабре этого года программе исполняется 20 лет. Актуальным релизом Qucs‑S на текущий момент является версия 2.1.0. Далее будет рассказано о функциях, добавленных в релизах, вышедших в этом году.

Тюнер элементов

Тюнер был добавлен в версии 2.1.0. Данный режим симуляции позволяет производить подстройку компонентов при помощи ползунков в специальном окне и сразу видеть результат на графиках. Тюнер в ряде случаев может быть удобнее, чем параметрический анализ. Чтобы запустить тюнер следует выбрать в главном меню Simulation→Tune или нажать кнопку на панели инструментов. Предварительно в схеме должны быть заданы симуляции. Если ни того, ни другого нет, то будет выдано предупреждение. Открывается пустое диалоговое окно. Затем требуется выбрать компоненты, которые нужно подстроить, щёлкнув по ним мышью. Можно также добавлять в диалог и свойства компонентов. Теперь в диалоговом окне появятся ползунки. Можно двигать их мышью и на графиках смотреть результат. Данный процесс иллюстрирует анимированный GIF. Схема представляет собой полосовой фильтр на любительский КВ диапазон 7 МГц.



Подстройка конденсаторов ДПФ при помощи тюнера

Быстрое переключение движков моделирования

Особенностью Qucs‑S является то, что этот симулятор позволяет выбирать между несколькими движками моделирования. Движком по умолчанию начиная с версии 0.0.23 предлагается Ngspice, который лучше всего подходит для моделирования любых электронных схем. Всего в Qucs‑S доступно три движка моделирования:

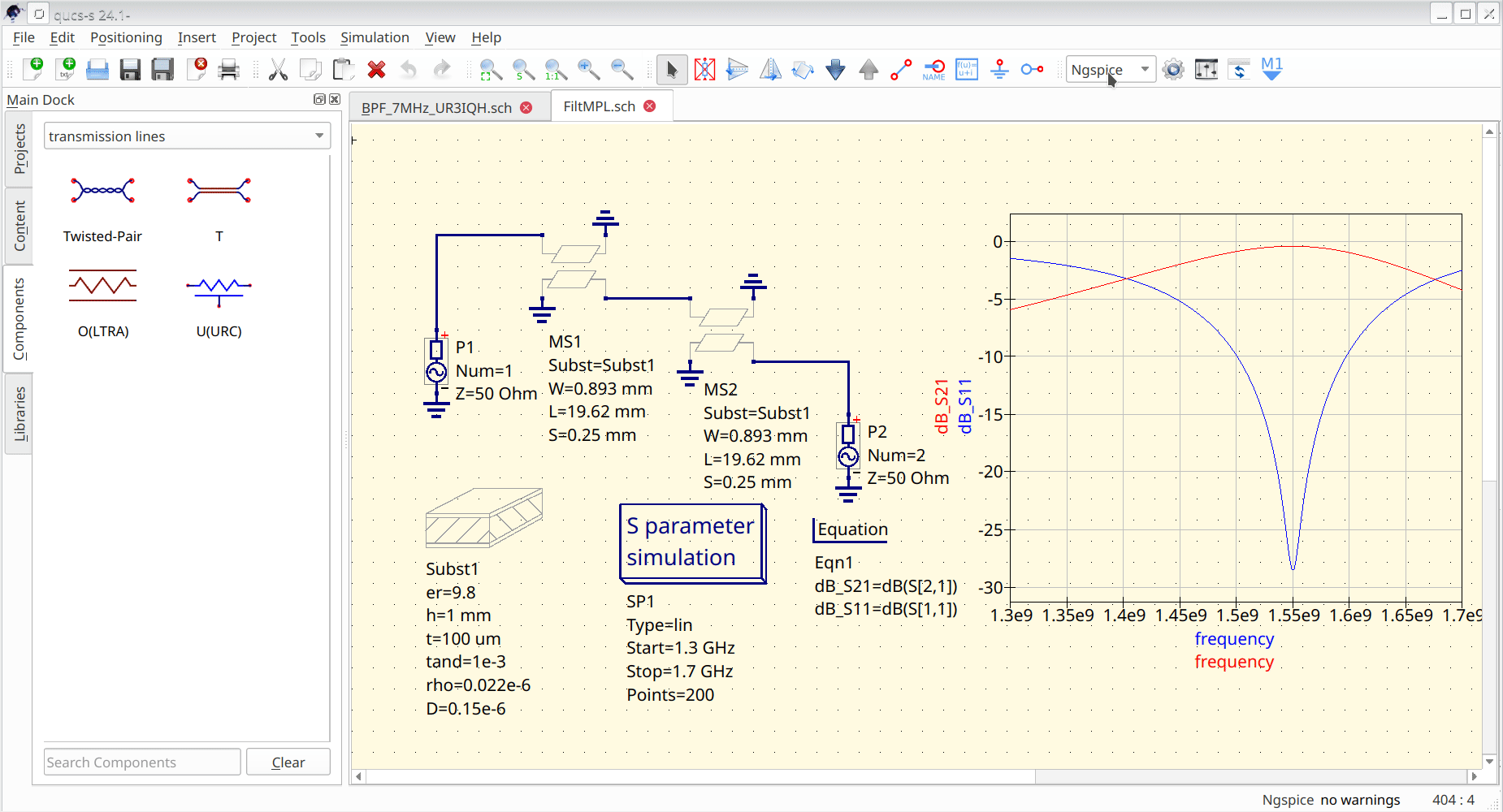
* Ngspice, упомянутый ранее <https://ngspice.sourceforge.io/>
* XYCE <https://xyce.sandia.gov/>
* Qucsator <https://github.com/qucs/qucsator>

Наибольший интерес из них представляет Qucsator, который был движком по умолчанию в старом Qucs. Он содержит расширенный набор компонентов для анализа СВЧ устройств, таких как модели волноводов и микрополосковых линий. Реализовать данные компоненты в SPICE проблематично. Но Qucsator имел проблемы с моделированием во временной области особенно при импульсных сигналов и не совместим со SPICE по синтаксису нетлиста, поэтому в Qucs‑S я от него отказался и перешёл на Ngspice по умолчанию. Но оказалось, что полностью списывать Qucsator в утиль пока рано.

К сожалению, разработчик, который сейчас занимается Qucsator, не развивает этот уникальный симулятор и не собирает бинарные пакеты. Для Linux придётся собирать Qucsator из исходников. От Qt он не зависит. Для Windows можно добыть Qucsator из версии Qucs-0.0.19. Если нужна последняя версия 0.0.20, то опять же придётся собирать из исходников.

В настоящее время имеются планы вернуть Qucsator в состав Qucs‑S в качестве особого движка моделирования для анализа ВЧ схем. Подробнее можно прочитать здесь: <https://github.com/ra3xdh/qucsator_rf/issues/1>

В Qucs‑S начиная с версии 2.0.0 можно переключаться между движками моделирования на ходу без перезагрузки программы. Для этого рядом с кнопкой запуска симуляции на панели инструментов добавлен выпадающий список, из которого можно выбрать симулятор. При этом компоненты на левой панели будут автоматически перезагружены, и остаются только совместимые с выбранный движком. Несовместимые компоненты также помечаются в самой схеме. Процесс иллюстрирует анимированный GIF:



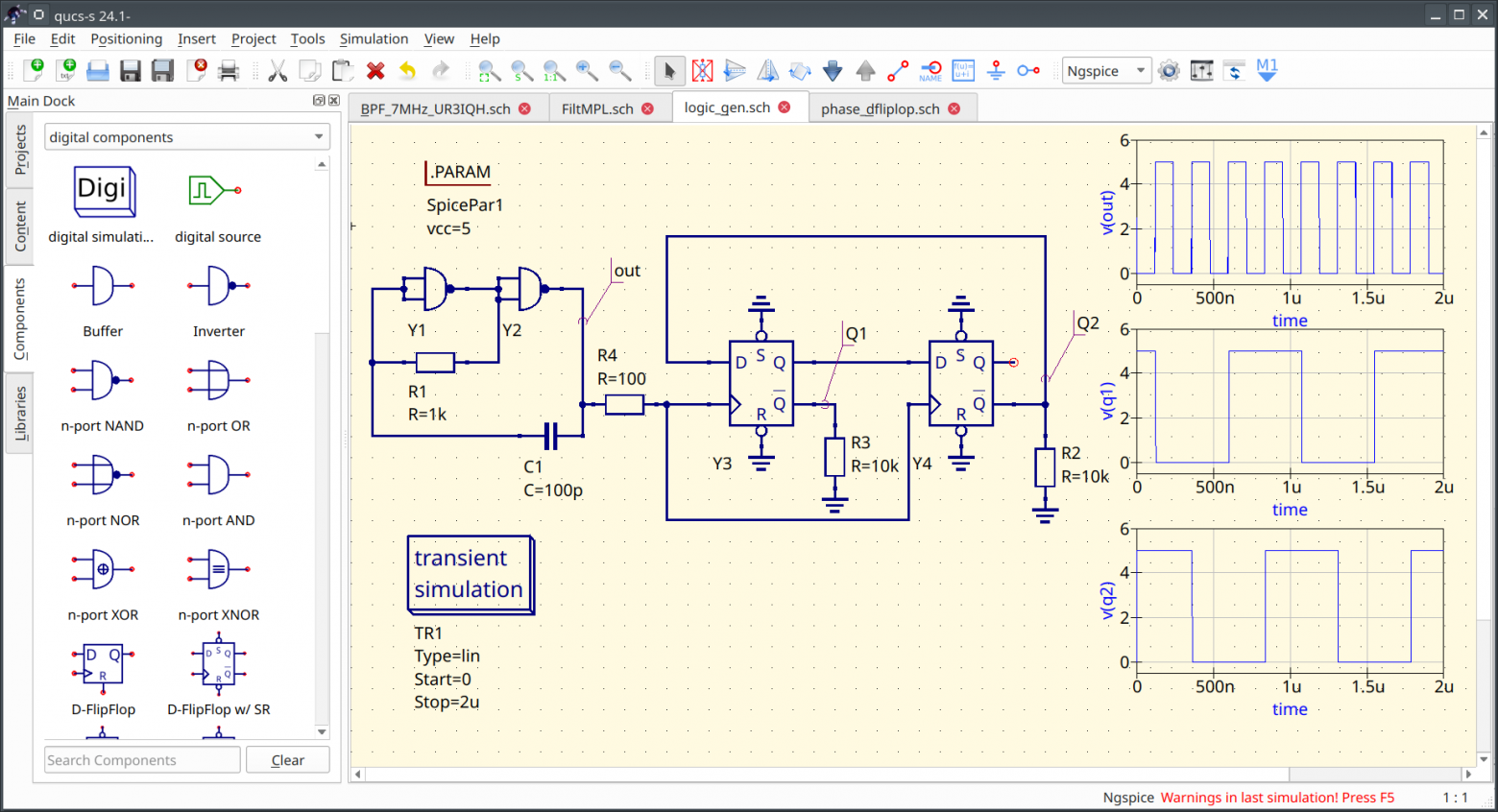
Быстрое переключение между движками моделирования

Цифровое моделирование при помощи IcarusVerilog

Начиная с версии 1.1.0 Qucs‑S поддерживает два режима цифрового моделирования:

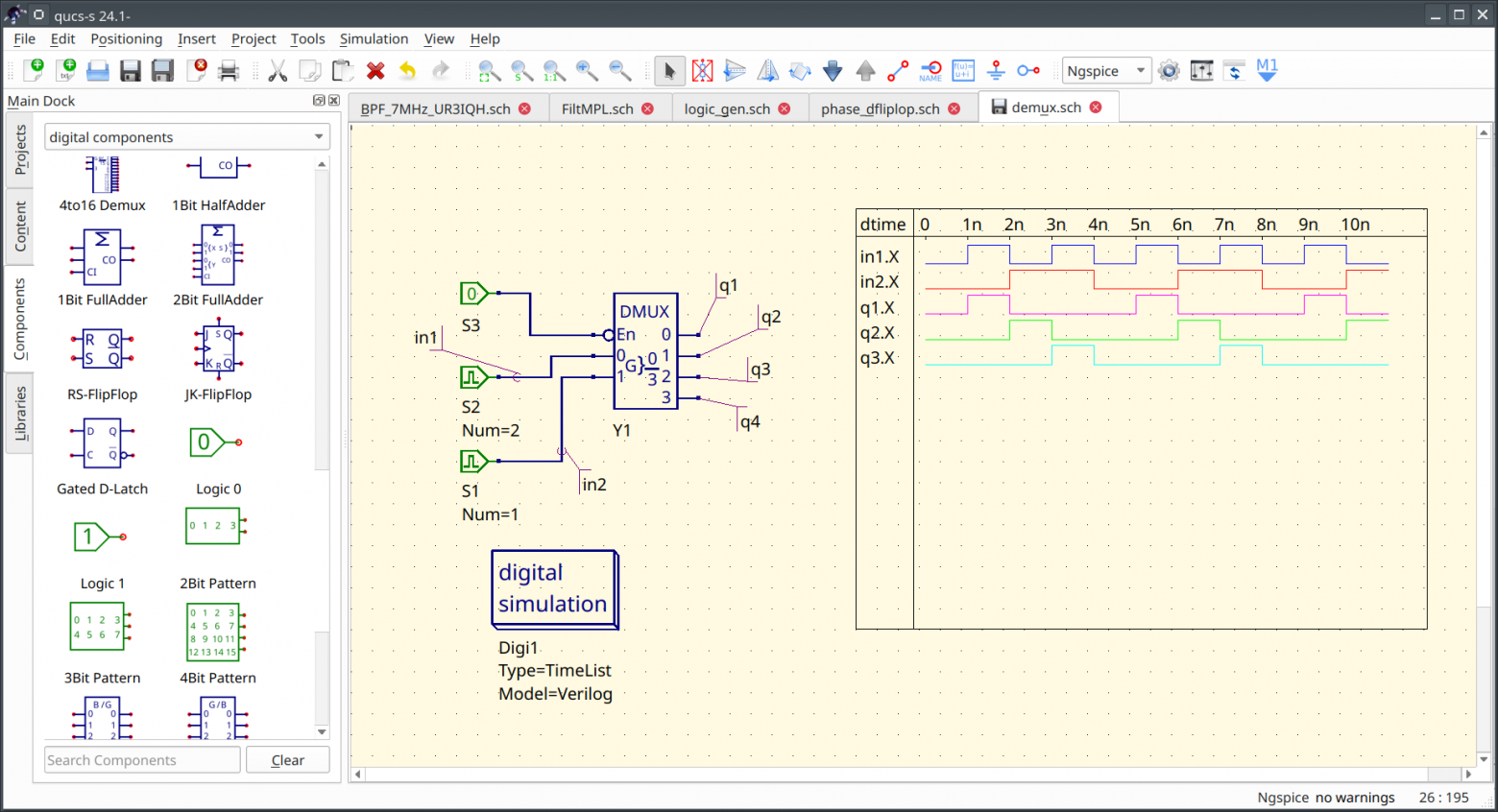
* Моделирование цифровых компонентов в аналоговом режиме. Возможно для ограниченного круга компонентов, таких как логические элементы и триггеры. Можно комбинировать цифровые и аналоговые компоненты в одной схеме.
* Моделирование чисто цифровой схемы при помощи IcarusVerilog <https://sourceforge.net/projects/iverilog/> При этом аналоговые компоненты в схеме не допускаются.

Пример моделирования генератора на логических элементах с делителем частоты в аналоговом режиме показан на скриншоте.



Моделирование цифровых компонентов в аналоговом режиме

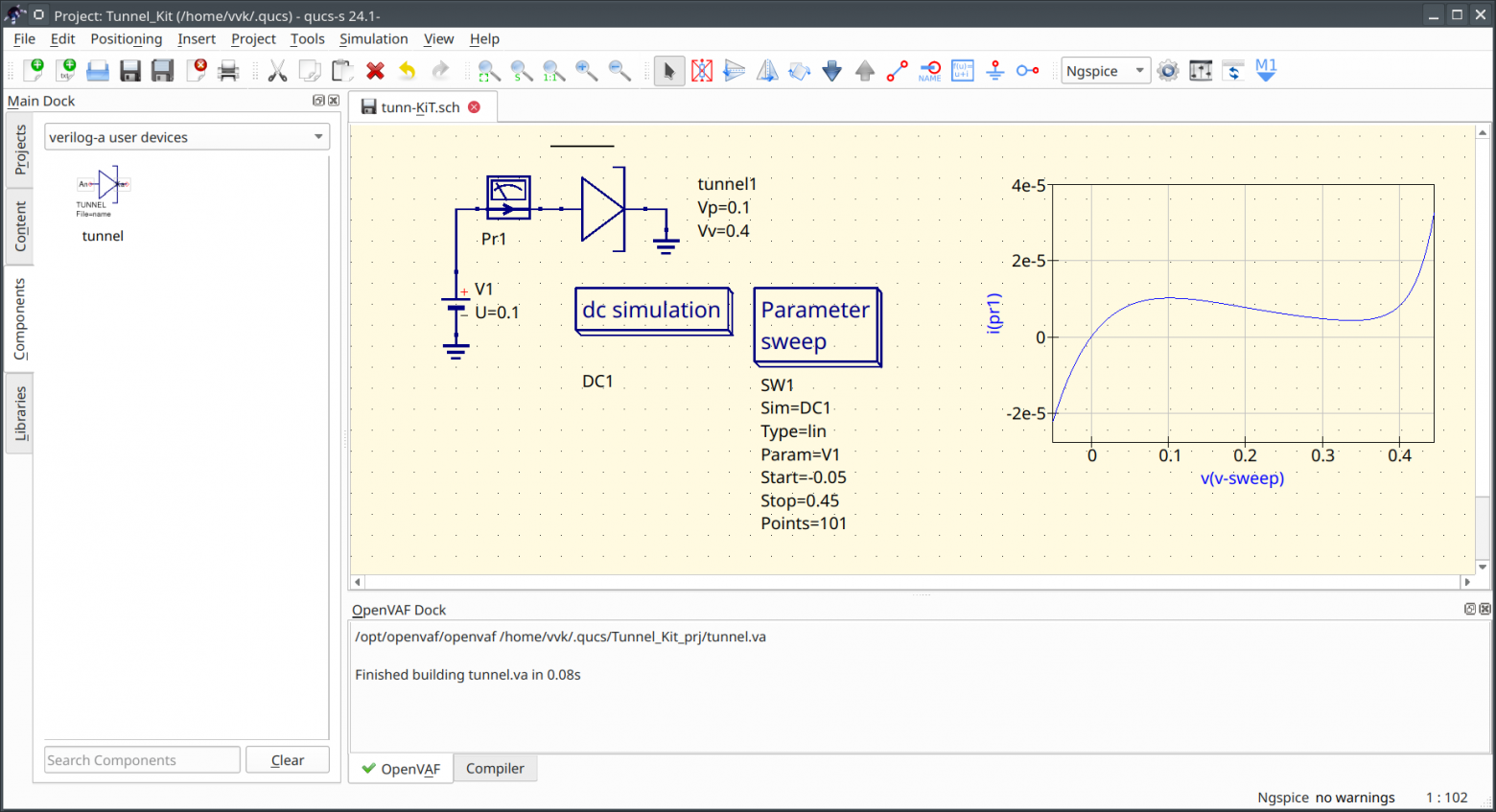
Пример цифрового моделирования схемы с демультиплексором с движком Icarus Verilog показан ниже. Icarus Verilog должен быть установлен в системе и доступен через $PATH. Цифровое моделирование запускается автоматически, если на схему добавлен компонент Digital simulation.



Цифровое моделирование при помощи Icarus Verilog

Поддержка моделей Verilog-A при помощи компилятора OpenVAF

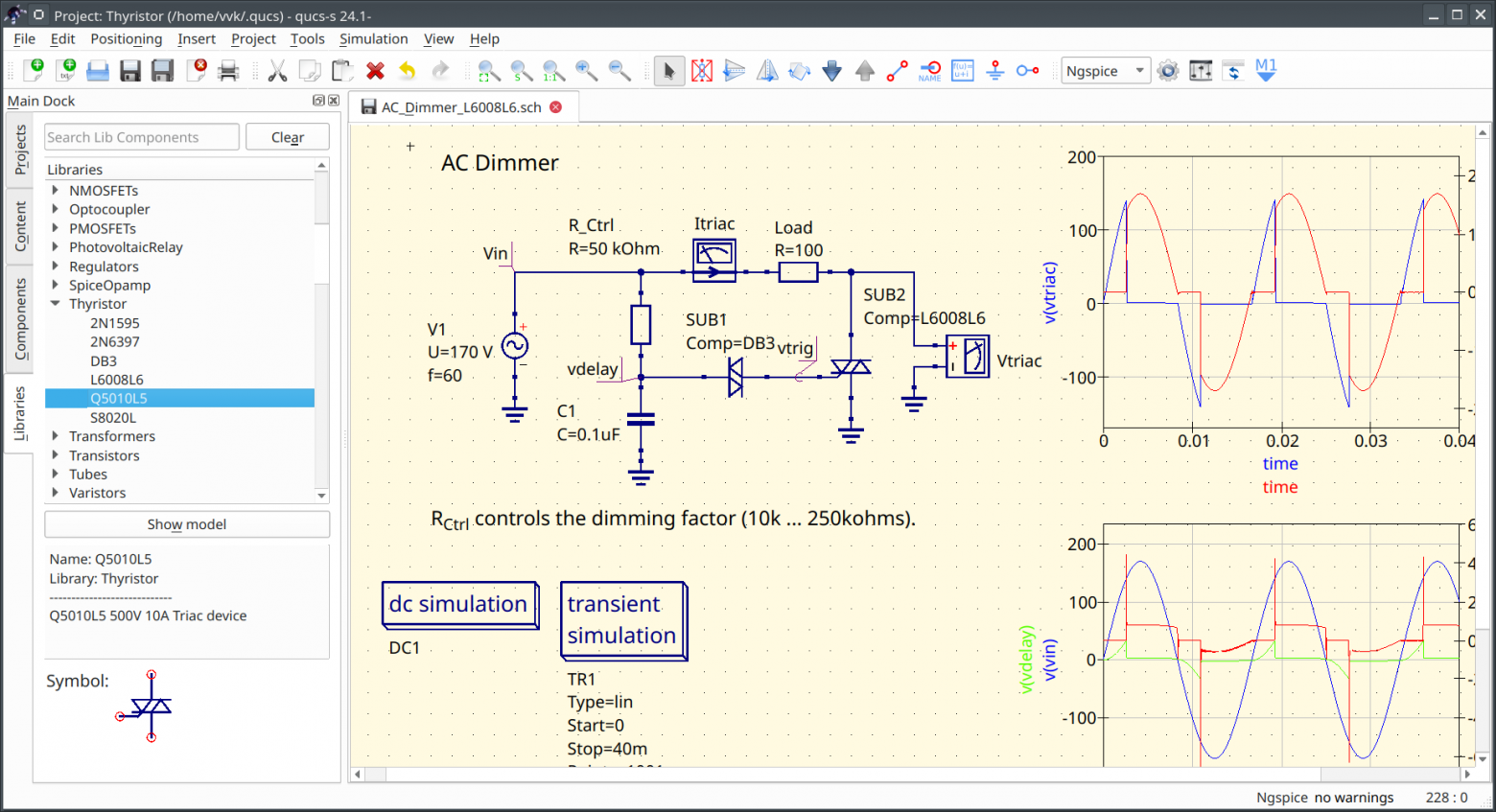
Verilog‑A — это язык описания аналоговых моделей. При этом возможно добавить в симулятор собственные модели компонентов без перекомпиляции всего симулятора. Чтобы использовать модели Verilog‑A нужна специальная программа‑компилятор, которая формирует двоичный подгужаемый модуль. Старый Qucs использовал для поддержки моделей Verilog‑A компилятор ADMS. В 2022 году был разработан новый компилятор OpenVAF <https://openvaf.semimod.de/> , который совместим с Ngspice. Поддержка OpenVAF добавлена в Qucs‑S начиная с версии 1.0.1. Процедура добавления моделей Verilog‑A точно такая же, как и для старого Qucs. Путь к компилятору должен быть указана в настройках программы в диалоге File→Application Settings на вкладке Locations. Описание можно найти в данной презентации [https://www.mos‑ak.org/venice\_2014/publications/T\_4\_Brinson\_MOS‑AK\_Venice\_2014.pdf](https://www.mos-ak.org/venice_2014/publications/T_4_Brinson_MOS-AK_Venice_2014.pdf) На скриншоте можно видеть пример компонента, заданного в виде модели Verilog‑A:



Моделирование туннельного диода при помощи Ngspice и OpenVAF

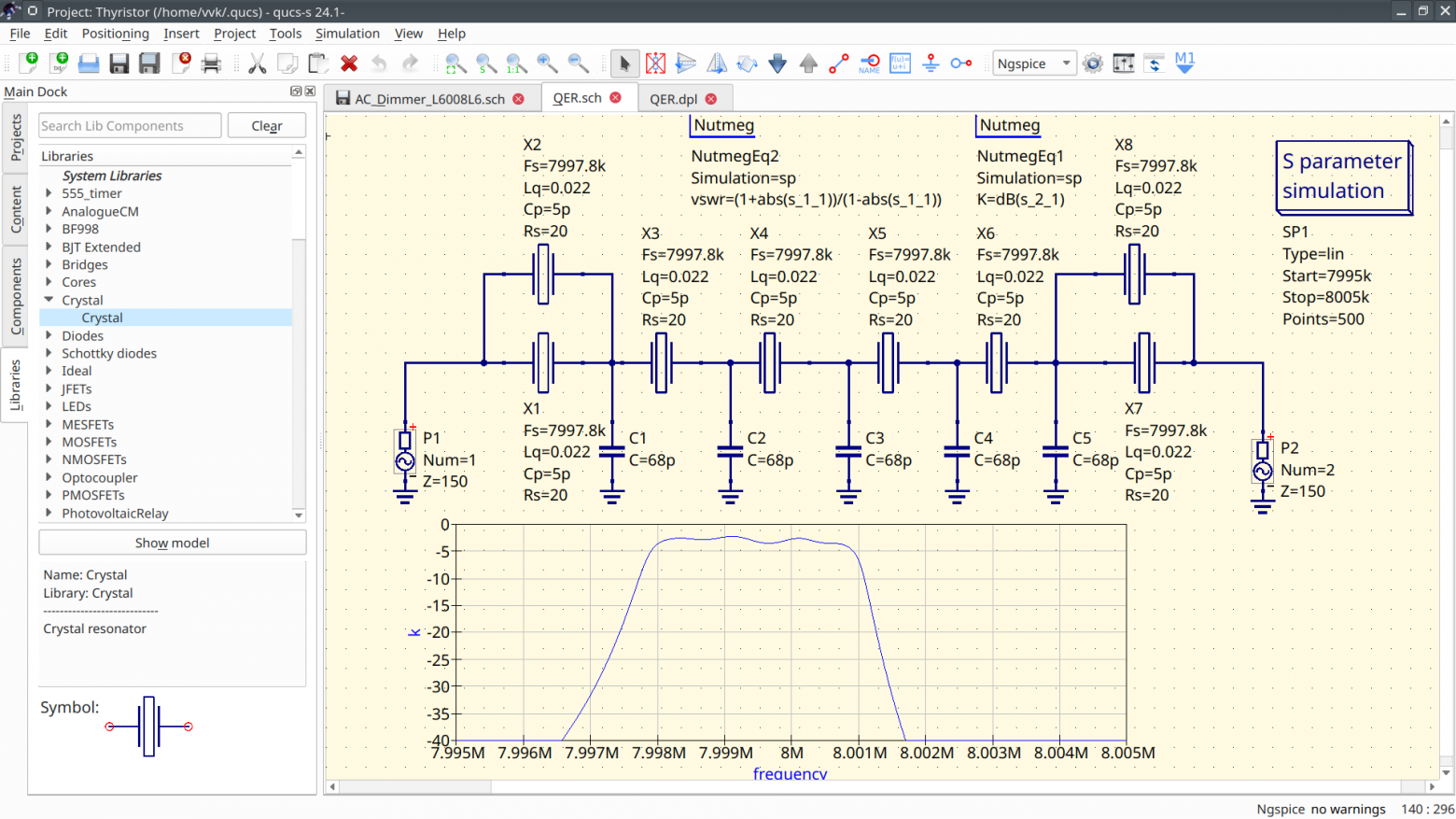
Расширена библиотека компонентов

В недавних релизах добавлено много новых библиотек компонентов, в том числе с тиристорами и оптопарами. Добавлена библиотека BJT Extended с дополнительными моделями транзисторов. На скриншоте можно видеть пример моделирования тиристорного регулятора мощности с компонентами из новой библиотеки.



Модель симисторного регулятора мощности

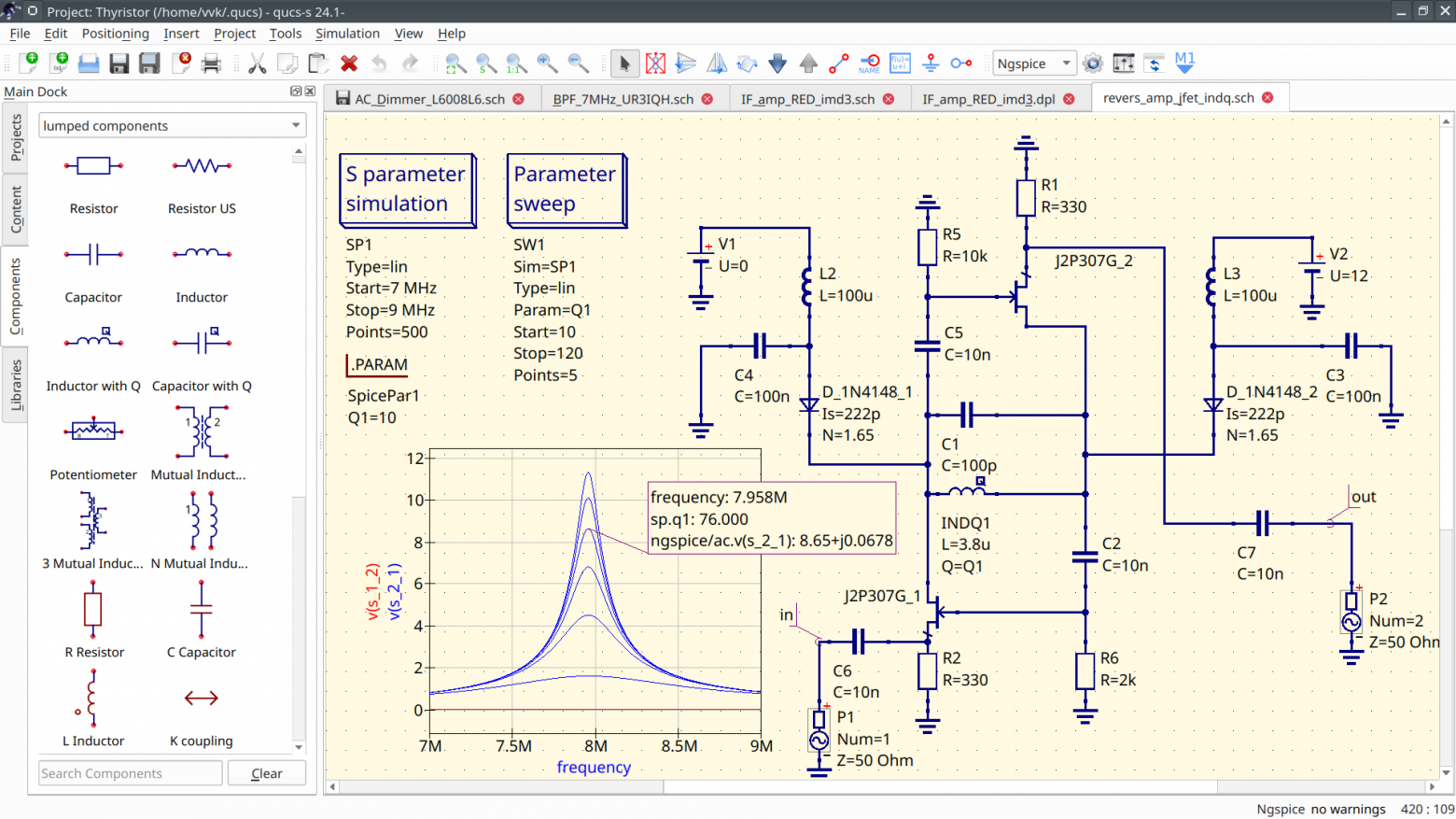
В релизе 0.0.24 добавлена библиотека Crystal, которая содержит модель кварцевого резонатора, которая специально адаптирована для моделирования радиолюбительских кварцевых фильтров. На скриншоте пример моделирования QER фильтра для КВ трансивера.



QER кварцевый фильтр

Новые системные компоненты

Добавлены новые системные компоненты. Наибольший интерес представляют катушки и конденсаторы с добротностью (INDQ и CAPQ), а также файловый компонент для файлов s2p. На скриншоте пример параметрического анализа зависимости коэффициента усиления реверсивного усилителя на 8 МГц от добротности катушки в резонансном контуре. Подобная схема используется для построения трактов ПЧ в радиолюбительских трансиверах.



Моделирование усилителя ПЧ с учётом добротности контура

Векторные иконки

В релизе 2.1.0 переделаны все иконки. Растровые иконки 22×22 больше не используются. Теперь для панелей инструментов используются вектроные SVG иконки, а иконки для компонентов генерируются автоматически. Данное новшество улучшает внешний вид программы на HiDPI мониторах. Примеры можно видеть на всех скриншотах в данной статье.

Инженерная нотация на диаграммах

Числа на всех диаграммах теперь можно переключить в инженерную нотацию (будет отображаться 1k вместо 1e3 или 2.1u вместо 2.1e-6). Данный настройки доступны в диалоге свойств диаграммы (выпадающий список Number notation).

