

Пакет средств проектирования электронных устройств gEDA

Проект gEDA показывает, что созданный усилиями сообщества пакет САПР, распространяемый в виде исходных кодов бесплатно, может составить конкуренцию коммерческим аналогам. Статья знакомит читателя с некоторыми возможностями gEDA и дает сведения, необходимые для быстрого начала работы.

Дмитрий Осипов

ossipov.dmitry@gmail.com

Юрий Бочаров

Введение

Проект gEDA [1] представляет набор программных инструментов для проектирования электронных устройств, доступных по лицензии GNU GPL. Это подразумевает свободу получения и модификации исходных кодов любых программ, распространяемых под данной лицензией. В большинстве случаев, хотя и необязательно, это подразумевает и бесплатность. Проект gEDA — это объединение достаточно большого количества других проектов, причем после объединения ни один из них не потерял своей самостоятельности. В настоящий момент gEDA предоставляет следующие инструменты:

- Графический ввод и редактирование схем. Возможна автоматическая генерация файла списков межсоединений во множестве стандартных форматов (SPICE, Spectre, Verilog и т. д.). Эта часть проекта разрабатывается собственно сообществом gEDA.
- Моделирование аналоговых и аналого-цифровых схем. Доступно сразу несколько разных симуляторов, поддерживаемых различными группами разработчиков. Во-первых, это NGSPICE [2] — развитие классического симулятора Spice3 f, с дополнительными возможностями для моделирования цифровых схем и поведенческого моделирования (расширение XSPICE). Кроме того, этот симулятор позволяет проводить совместное моделирование с физическими TCAD-моделями. Во-вторых, GNUCAP [3] — полностью независимая разработка, не основанная на исходном коде SPICE. Обеспечивает совместимость со Spectre

(но не Spectre-HDL или Verilog-A) и неполную совместимость со SPICE.

- Графическая оболочка для симуляторов GspiceUI [4]. Позволяет производить автоматическую трансляцию схем из формата графического редактора gEDA в формат файла списка межсоединений выбранного симулятора (NGSPICE или GNUCAP). Существенно облегчает доступ ко всем настройкам симулятора.
- Средства разработки и моделирования цифровых схем Icarus Verilog [5]. Полная поддержка стандарта языка Verilog IEEE Std 1364-2005. Поддержка VHDL отсутствует.
- Отображение результатов моделирования GtkWave [6].
- Проектирование печатных плат. Поддерживается сообществом gEDA.

Таким образом, gEDA позволяет пройти весь путь от разработки и моделирования устройства до его исполнения на печатной плате и не уступает коммерческим аналогам, а в чем-то и превосходит их.

Одно из преимуществ gEDA — нулевая стоимость, причем пользователь получает бесплатно полнофункциональные версии всех программ, без каких-либо ограничений. Другим, менее очевидным достоинством является распространение в виде исходных кодов. Это позволяет легко сделать gEDA основой собственной САПР, включив, например, собственные модели или добавив необходимую функциональность.

«Родной» операционной системой для всех инструментов gEDA является Linux, однако открытость исходного кода позволяет запустить любую программу и на другой платформе. Существуют неофициальные сборки некоторых программ для Windows и Mac OS. Однако даже на «родной» платформе самостоятельная установка всех частей gEDA по отдельности — достаточно трудоемкая задача. Поэтому многие дистрибутивы Linux (например, Fedora, OpenSuse, Mandriva) предлагают решения на основе gEDA, устанавливаемые штатными средствами управления программой. В таком случае установка любых необходимых компонентов может быть выполнена в несколько кликов мышкой.

Для того чтобы познакомиться с gEDA и другими свободными средствами САПР для проектирования электроники, можно воспользоваться специальным загрузочным диском, который подготовила группа разработчиков Fedora — Fedora Electronic Lab [7]. Он по-

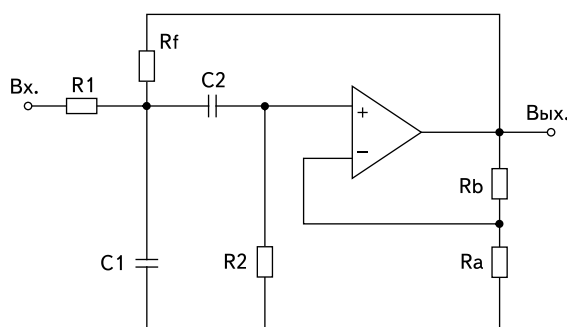


Рис. 1. Полосовой фильтр Саллена-Ки

зволяет загрузить без установки на жесткий диск компьютера операционную систему Fedora с заранее сконфигурированными средствами САПР для электроники, включая gEDA. При желании с этого же диска можно произвести установку предустановленной ОС.

Минус таких подходов к установке — отсутствие тонкой настройки инструментов, возможной только при компиляции программ из исходных кодов.

Наряду с очевидными достоинствами gEDA большим недостатком этого пакета средств САПР является разобщенность отдельных средств. Так как все программы разрабатываются различными сообществами, то они не обладают единым интерфейсом и документацией. Большинство доступных документов описывают отдельные инструменты или программы, тогда как для быстрого освоения САПР необходимо описание процесса работы над реальным проектом, а не перечисление опций симулятора или редактора схем. Отсутствие единого интерфейса отталкивает рядового пользователя, который привык выполнять все основные операции в единственном окне, переходя от одного инструмента к другому. Однако в действительности процесс создания и моделирования схем электронных устройств в gEDA ненамного труднее, чем в популярных программах типа OrCAD.

Эта статья должна дать возможность быстро начать работу в среде gEDA, которую можно рассматривать как альтернативную замену коммерческим программам. Естественно, что в одной статье нельзя рассказать обо всех возможностях gEDA, поэтому в ней не говорится об использовании моделей, представленных на языках Verilog, Verilog-AMS, а также проектировании печатных плат.

На примере разработки полосового фильтра с топологией Саллена-Ки [8] (рис. 1) мы рассмотрим основные моменты процесса создания и моделирования электронной схемы:

- ввод схемы с использованием макромодели;
- оформление схемы как макромодели;
- моделирование схемы с использованием графического интерфейса и отображение результатов.

Системные требования и версии программ

Программные средства, входящие в пакет gEDA, поддерживаются большинством современных дистрибутивов ОС Linux, в частности Fedora, Mandriva, OpenSuse, Ubuntu. Работа в ОС Windows не поддерживается, однако возможна компиляция программ gEDA в среде Cygwin [9]. При написании этой статьи авторы использовали ОС Scientific linux 6.0 (RHEL 6.0) и следующие версии инструментов из состава gEDA:

- gSCHEM 1.6.2.20110115 — редактор схем;
- gnetlist 1.6.2.20110115 — конвертер различных схемных описаний;
- ngspice 22 — SPICE-симулятор;
- gspiceui 1.0.0 — графическая оболочка для SPICE-симуляторов;
- gtkWave 3.3.15 — программа для просмотра и анализа сигналов.

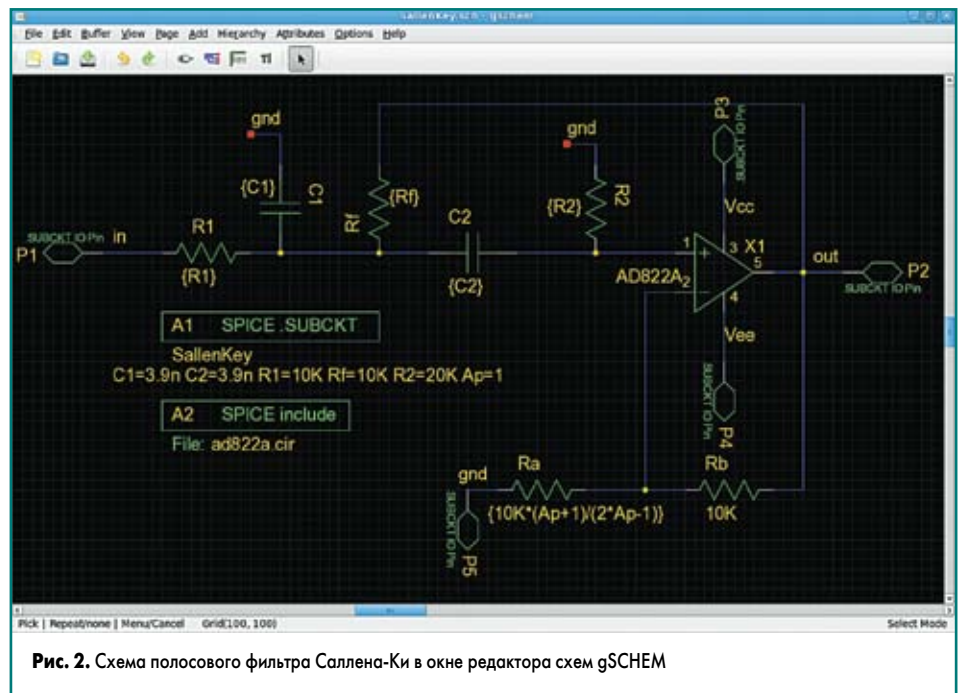


Рис. 2. Схема полосового фильтра Саллена-Ки в окне редактора схем gSCHEM

Ввод схемы

Ввод схемы осуществляется с помощью программы gSCHEM, которая может быть вызвана либо из консоли командой `gchem`, либо из меню `OS Applications → Electronics → Circuit and PCB design → GEDA Schematic editor`.

Сохраним новый файл как `SallenKey.sch`. Добавим на схему символ усилителя, для этого используется либо команда меню `Add → Component`, либо кнопка на панели инструментов. В появившемся окне `Select Component...` переключимся во вкладку `Libraries` и в раскрывающемся списке `Basic Devices` выберем символ `opamp-1.sym`. Всегда необходимо обращать внимание на нумерацию ножек символа: она должна совпадать с порядком следования портов модели, которую вы собираетесь использовать с данным символом. Библиотека редактора схем gSCHEM предоставляет только символы элементов, процесс подключения моделей элементов будет рассмотрен позднее.

Для соединения элементов проводниками воспользуемся командой `Add → Net` либо кнопкой на панели инструментов. Для того чтобы присвоить имя линии связи, нужно дважды щелкнуть по ней левой кнопкой мыши и в появившемся окне добавить значение атрибуту `netname` (поле `Value`), а затем нажать кнопку `Add Value`. Присваивать имена всем линиям связи необязательно: те проводники, которым не присвоены имена, получают их автоматически. Для переименования символа дважды кликните по нему левой клавишей мыши и в окне `Edit Attributes` измените атрибут `refdef`. Обратите внимание, что начальная буква имени элемента (значение `refdef`) используется генератором файла списка межсоединений и, соответственно, самим симулятором для идентификации типа модели, поэтому ее изменять не следует. Существуют следующие стандартные типы SPICE-моделей [10]:

- R<имя> — резистор.
- C<имя> — конденсатор.

- L<имя> — индуктивность.
- S<имя> — ключ, управляемый напряжением.
- W<имя> — ключ, управляемый током.
- V<имя> — источник напряжения.
- I<имя> — источник тока.
- E<имя> — источник напряжения, управляемый напряжением.
- G<имя> — источник тока, управляемый напряжением.
- H<имя> — источник напряжения, управляемый током.
- F<имя> — источник тока, управляемый током.
- X<имя> — макромодель (подсхема).
- D<имя> — диод.
- Q<имя> — биполярный транзистор.
- M<имя> — МОП-транзистор.
- J<имя> — полевой транзистор с p-n-переходом.

Кроме того, NGSPICE добавляет еще несколько типов для поведенческого описания схем и цифровых элементов [11].

Вид подготовленной для моделирования схемы фильтра в окне редактора принципиальных схем показан на рис. 2.

Привязка символов к моделям элементов

После того как схема введена, необходимо указать типы моделей используемых элементов и задать их параметры. В качестве модели операционного усилителя (ОУ) используем макромодель ОУ типа AD822A компании Analog Devices, скачать которую можно с официального сайта производителя [12].

Загруженный файл `ad822a.cir` сохраним в рабочей директории. Добавим к символу усилителя атрибут `model-name` с именем модели, в данном случае AD822A. Название модели можно посмотреть, открыв файл `ad822a.cir` в любом текстовом редакторе. Кроме того, следует проверить значение атрибута `refdes`: название символа должно начинаться с «X».

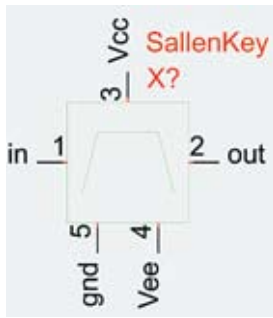


Рис. 3. Символ для макромоделли фильтра

Подключим файл, содержащий описание модели усилителя, к проекту. Для этого в окне схемного редактора прямо на поле изображения схемы добавим символ `spice-include-1.sym` с атрибутом `file=ad822a.cir`.

Встроенные модели резисторов и конденсаторов будут автоматически привязаны к символам, имена которых начинаются с «R» и «C», генератором списков межсоединений. Однако необходимо задать значения параметров этих моделей.

Так как схема будет использоваться как параметризованная макромоделль, не будем задавать значения параметров отдельных элементов (резисторов и конденсаторов) численно. В качестве параметров фильтра будем использовать максимальное усиление в полосе пропускания — A_p , а также номиналы компонентов R_f , R_1 , R_2 , C_1 , C_2 . Номиналы резисторов R_a и R_b зададим через величину A_p .

Рассмотрим присвоение параметризованного значения номиналу резистора R_a . Дважды кликнув по символу резистора и в появившемся окне **Edit Attributes** добавим атрибут `value` со значением $\{10e3*(Ap+1)/(2*A-1)\}$. Аналогично поступим с другими пассивными элементами схемы.

Оформление схемы как макромоделли

Для того чтобы оформить схему как макромоделль, необходимо добавить несколько SPICE-директив. Во-первых, дадим имя новой макромоделли. Для этого добавим в поле редактирования схемы специальный элемент — `spice-subcircuit-LL-1.sym` (**Add Component** → **Libraries** → **SPICE simulation elements**) и изменим его атрибуты: `model-name` на `SallenKey` (это имя новой подсхемы), а `refdes` на `A1` (это имя экземпляра директивы SPICE).

Кроме того, добавим атрибут `value` со значением параметров макромоделли по умолчанию, например: $C_1=3.9n$ $C_2=3.9n$ $R_1=10K$ $R_2=20K$ $A_p=1$. К сожалению, последняя официальная версия gEDA не поддерживает добавление значений по умолчанию для параметров макромоделлей. Чтобы исправить этот недостаток, необходимо модифицировать файл `/usr/local/share/gEDA/scheme/gnet-spice-sdb.scm` согласно [13].

Добавим выводы к макромоделли. Для этого к каждому узлу, который необходимо сделать выводом, следует подключить компонент

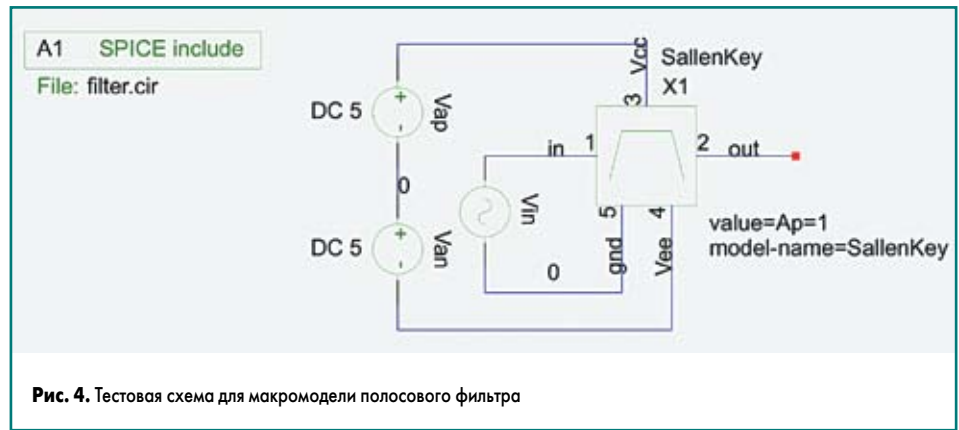


Рис. 4. Тестовая схема для макромоделли полосового фильтра

`spice-subcircuit-IO-1.sym` (**Add Component** → **Libraries** → **SPICE simulation elements**). Атрибут `refdes` элемента вывода отвечает за имя порта. Выводы должны именоваться в соответствии со схемой `P<номер вывода>`.

Для создания файла макромоделли, аналогичного скачанному с сайта Analog Devices, воспользуемся входящей в состав gEDA программой `gnetlist` — средством конвертации различных схемных описаний.

Для создания модели следует ввести в терминале, находясь в рабочей директории, следующую строку:

```
$ gnetlist -I -g spice-sdb -o SallenKey.cir SallenKey.sch
```

`Sallenkey.cir` — желаемое название файла текстового SPICE-описания модели фильтра, `SallenKey.sch` — созданная в gSCHEM схема. Полученный файл можно использовать аналогично файлу `ad822.cir`.

Создание графического изображения символа модели

Макромоделль фильтра готова. Осталось создать символ для нового элемента. С этой целью также воспользуемся программой gSCHEM. Откроем gSCHEM, удалим рамку и сохраним новый файл как `SallenKey-1.sym` в рабочей директории. Для рисования символа можно пользоваться инструментами из меню **Add**, расположенными под горизонтальной чертой, такими как `line` — линия, `box` — прямоугольник, `circle` — окружность, `ark` — дуга, `pin` — вывод. Созданный символ фильтра Саллена-Ки показан на рис. 3. Для выводов нужно задать некоторые обязательные атрибуты:

- `pinnumber` — номер;
- `pinseq` — номер для симулятора (в данном случае совпадают);
- `pintype` — тип: вход (in), выход (out), двунаправленный (io);
- `pinlabel` — имя (для удобства использования символа).

Добавим атрибуты всему символу. Для этого, убедившись, что ничего не выделено в поле редактора, выберем пункт меню **Add** → **Attribute**.

Необходимые атрибуты:

- `device` — имя созданного устройства, например `SallenKey`;

- `refdes` — начальная буква имени SPICE типа устройства со знаком вопроса. Для макромоделей это «X?».

После того как символ нарисован, необходимо убедиться, что размер сетки установлен в значение 100 (**Options** → **Snap Grid Spacing**), а также включена привязка к узлам сетки (**Options** → **Toggle Snap On/Off**). После этого выполним привязку символа к узлам сетки (это необходимо для корректной вставки символа в схему): **Edit** → **Symbol Translate** → **offset 0**. Символ готов.

Для того чтобы символ стал доступен из меню gSCHEM, необходимо отредактировать или создать настроечный файл `~/gEDA/gafrc`, добавив в него строки:

```
;; mylib  
(component-library <Путь до рабочей директории>)
```

Создание тестовой схемы

Протестируем созданную модель. Для этого создадим новый файл в gSCHEM, например `SallenKey_test.sch`. Добавим символы созданного нами фильтра и источников. Установим атрибут `value` для источника питания атрибут «5», для источника сигнала — «DC 0 AC 1». С созданной на предыдущем шаге моделью фильтра Саллена-Ки можно обходиться так же, как и с моделью усилителя Analog Devices. Кроме того, можно изменить значения параметров макромоделли с помощью атрибута `value`, например, чтобы изменить коэффициент усиления фильтра, следует присвоить атрибуту `value` « $A_p=<\text{новое значение}>$ ». Вид схемы показан на рис. 4.

Моделирование тестовой схемы

Для моделирования работы схемы будем использовать удобную графическую оболочку `gspiceui` над двумя gEDA-симуляторами — `NGSPICE` [2] и `gnucap` [3]. Выбор симулятора зависит от типа решаемой задачи. Так, хотя симулятор `gnucap` и содержит некоторые дополнительные опции, связанные в основном с поведенческим моделированием, но он не полностью совместим с SPICE, таким образом, могут возникнуть некоторые трудности с использованием моделей, которые представляют фирмы-изготовители ИС.

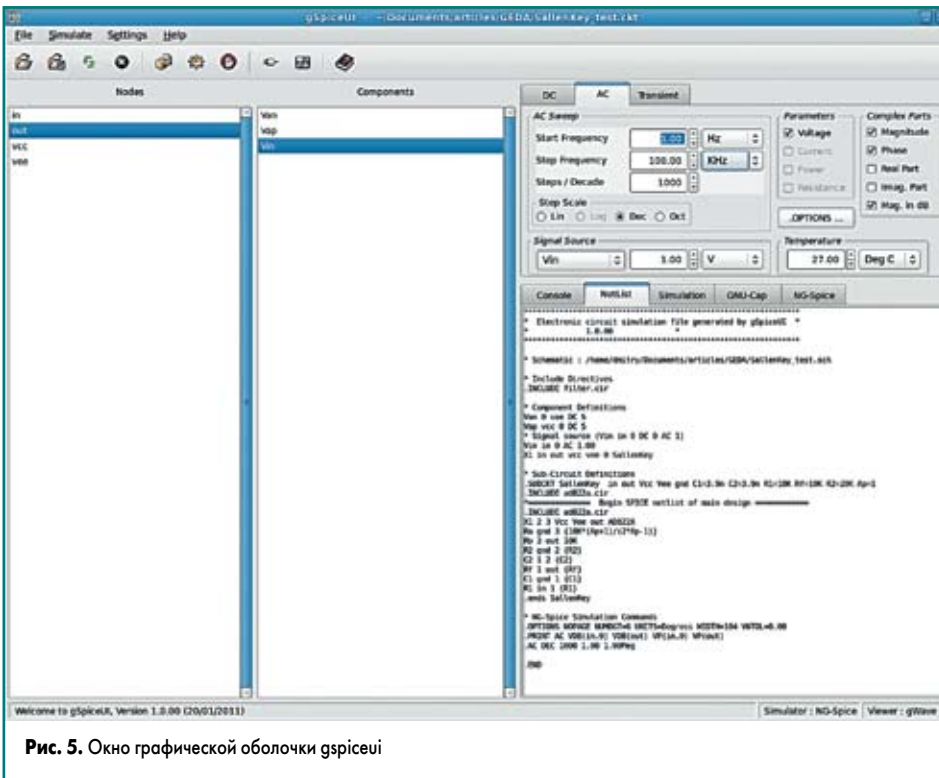


Рис. 5. Окно графической оболочки gspiceui

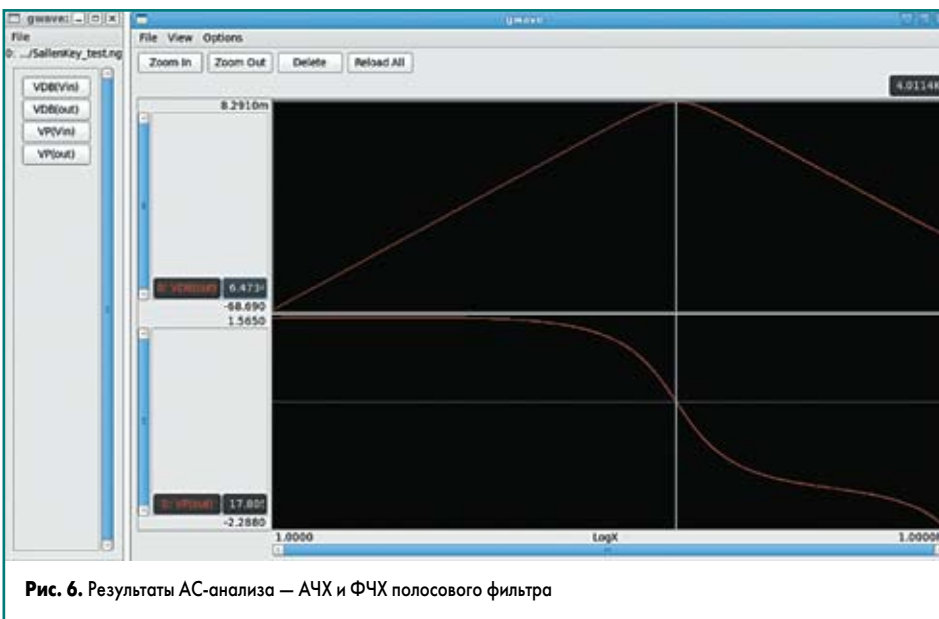


Рис. 6. Результаты АС-анализа — АЧХ и ФЧХ полосового фильтра

Программа GspiceUI может быть вызвана как из меню **Applications** → **Electronics** → **GspiceUI**, так и из терминала. Окно программы показано на рис. 5. Так как мы использовали модель ad822, не совместимую с **gnucap**, то в качестве симулятора необходимо выбрать **ngspice** (**Settings** → **NG-Spice**).

Загрузив тестовую схему **SallenKey_test.sch** (**File** → **Import...**). Далее перейдем на вкладку АС и внесем необходимые настройки анализа, воспользовавшись интуитивно понятным интерфейсом. В поле **Nodes** и **Components** можно выбрать узлы и компоненты схемы, на которых будет произведен расчет. Для начала моделирования воспользуемся пунктом меню **Simulate** → **Run**. Чтобы просмотреть результаты в программе gtkWave, выберем **Simulate** → **Data Viewer**. Появятся два окна **gtkWave**: одно с двумя пустыми полями графиков, второе — со списком доступных для отображения сигна-

лов. Необходимые для отображения сигналы перетащим мышью из второго окна в первое. Пример отображения результатов АС-анализа показан на рис. 6.

Заключение

Приведенный пример показывает, что, несмотря на некоторую непривычность САПР, gEDA позволяет выполнять достаточно сложные задачи по проектированию и анализу схем и в целом не уступает аналогам, распространяемым на коммерческой основе. Кроме того, пользователь сразу получает полнофункциональные версии всех программ, избавляясь от сложного выбора типа лицензии, количества инструментов и т. д. Это особенно удобно в учебном процессе, поскольку любой студент может повторить у себя дома лабораторную работу или заниматься разработкой своего

курсового проекта, не задумываясь об ограничениях пробной версии САПР.

Авторы использовали этот пакет САПР в лабораторных работах, проводимых в рамках курса «САПР в микроэлектронике» на кафедре микро- и нанозлектроники МИФИ. Как показал опыт, использование gEDA в учебном процессе позволяет студентам попутно овладеть навыками работы в UNIX-подобных системах, а также глубже понять принципы работы инструментов САПР, благодаря необходимости работать с каждым инструментом отдельно. Студенты, изучившие приемы работы в среде gEDA, на дальнейших этапах обучения быстрее осваивают сложные коммерческие САПР, такие как Cadence Virtuoso.

Литература

1. www.gpleda.org
2. <http://ngspice.sourceforge.net/>
3. <http://gnucap.org/dokuwiki/doku.php?id=gnucap:start>
4. <http://gspiceui.sourceforge.net/>
5. <http://www.icarus.com/eda/verilog/>
6. <http://gtkwave.sourceforge.net/>
7. <http://spins.fedoraproject.org/fel/>
8. Linear Circuit Design Handbook. Analog Devices. Ed. H. Zumbahlen. Newnes, 2008.
9. www.cygwin.com
10. Williams C. D. H. Quickstart Tutorial for Spice 3. <http://newton.ex.ac.uk/teaching/CDHW/Electronics2/Spice3Quickstart.html>
11. Nenzi P., Vogt H. Ngspice Users Manual. Version 22plus. 2011. <http://ngspice.sourceforge.net/docs.html>
12. www.analog.com
13. <https://bugs.launchpad.net/geda/+bug/698736>