

ПРОГРАММА ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ГРАФИЧЕСКИХ ТЕСТОВ

Камилов И.К.

Научный руководитель профессор Бронов С. А.

Сибирский федеральный университет

Тест (от англ. test — испытание, исследование) — система заданий различной сложности и специфической формы, позволяющая оценить знания, умения и навыки, измерить их уровень.

Появились тесты очень давно, во времена античности. Древние греки, например, в образовании использовали систему разных испытаний для определения физических и интеллектуальных навыков учеников.

В своей статье «Интеллектуальные тесты и измерения» Джеймс Кеттел, американский психолог, впервые сформулировал термин «интеллектуальный тест» в 1980 г.

В наше время различные тестирования предлагаются всяческими сетевыми ресурсами, посредством которых можно узнать уровень своих знаний по определенной дисциплине. Так же они широко применяются в высших учебных заведениях.

В институте космических и информационных технологий (ИКИТ) под руководством директора ИКИТ д-ра техн. наук, профессора Г.М. Цибульского разрабатывается и реализуется система автоматизированного управления учебным процессом на основе концепции электронного университета [1].

Данная система реализована с использованием виртуальной обучающей среды Moodle и дополнительных модулей, обеспечивающих автоматическое формирование данных о текущей успеваемости студентов. Для контроля успеваемости используются тесты, которые можно разделить на четыре типа:

- 1) Возможность выбора одного варианта ответа;
- 2) Возможность указать соответствие вариантов ответов;
- 3) Множественность выбора ответов;
- 4) Ввод ответа с помощью клавиатуры (отдельных слов или словосочетаний).

Все вышеперечисленные варианты тестов являются недостаточно информативными. Первые три из них можно назвать пассивными, когда тестируемый только выбирает правильный ответ из предложенных. В четвёртом варианте тестируемый создаёт ответ сам, но возможности диалога ограничены цифро-буквенным представлением ответа и его простотой. В рамках существующих тестов можно проверить только знания тестируемых (хотя и в ограниченных пределах). Проверить умения и, тем более, навыки с помощью таких тестов невозможно. Для этого необходимо увеличить степень интерактивности, предоставив возможность тестируемому самому формировать ответ на вопрос и обеспечив автоматизированную обработку сложного ответа. Такие тесты можно назвать интерактивными. Работа над такими тестами выполняется в ИКИТ под руководством д-ра техн. наук, профессора, руководителя НУЛ САПР С. А. Бронова.

В данное время создана эскизная версия программы, в которой будет реализован такой вид тестирования. Программа «CircuitSoftware» разрабатывается на платформе .NET, в среде разработки Microsoft Visual Studio 2010, язык программирования C#.

Данный тестовый инструмент позволяет собирать заданные типы схем и проверяет правильность их конфигурации. Этот тип тестов может использоваться при изучении структурных и поэлементных схем различного назначения. В частности, аналогично работают графические редакторы схем в системах моделирования электронных устройств (MicroCAP, OrCAD и др.). Но в данном случае требуется проверка схемы.

Например, может быть задание собрать схему диодного моста или интегратора на операционном усилителе из библиотеки готовых элементов. Особенностью тестовой программы является способность её определять правильность схемы в условиях неоднозначности выполнения некоторых заданий. Для этого программа анализирует соединения элементов, формирует матрицу соединений и сравнивает её с эталонной матрицей, задаваемой как ответ на поставленное задание. Многовариантность решения задания приводит к нескольким матрицам правильных решений, которые путём специальных преобразований приводятся к одной.

На данном этапе реализованы следующие функции:

- перемещение и поворота элемента;
- изменение размера элемента;
- создание связи между элементами.

Одной из основных функций является функция создания связи между объектами. Каждая связь содержит в себе как входные, так и выходные параметры (индексы):

- «1» — входная связь;
- «-1» — выходная связь;
- «0» — связь отсутствует.

Библиотека элементов реализована при помощи класса `UserControl`. Данный класс предоставляет простой способ создания элемента схемы.

Пример реализации конденсатора с использованием класса `UserControl` и его графическое изображение — на рис. 1.

```
namespace CurcuitST.UserControls
{
    public partial class Capacitor :
    UserControl
    {
        public Capacitor()
        {
            InitializeComponent();
        }
    }
}
```

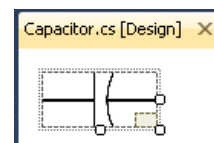


Рисунок 1 — Графическое изображение конденсатора

Например, в задании предусматривается построение схемы диодного моста (рис. 2).

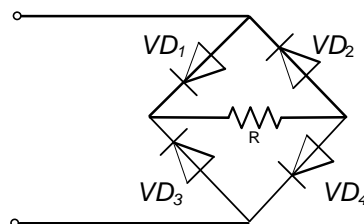


Рисунок 2 — Схема диодного моста

В главном окне программы расположены: системное меню, панель инструментов, рабочая область (рис. 3).

Для того чтобы добавить элемент, следует зайти в меню *Инструменты* и выбрать соответствующий элемент. Для простоты можно воспользоваться панелью инструментов, где расположены кнопки для создания новых элементов, таких как: резистор, диод, конденсатор, транзистор, земля, так же кнопка *Check*.

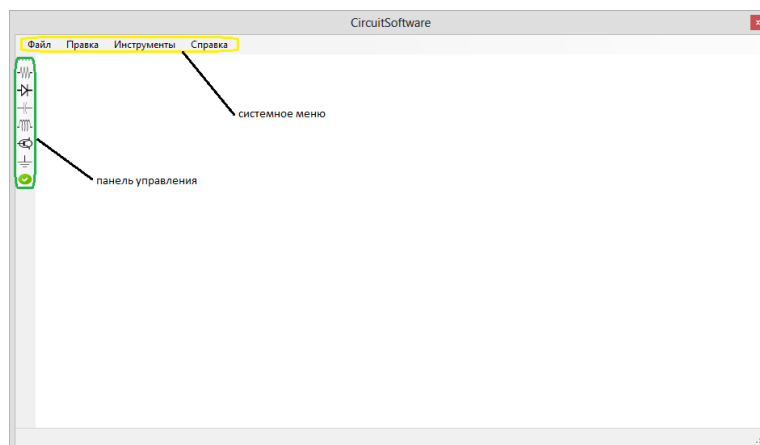


Рисунок 3 — Главное окно программы

Для построения диодного моста потребуется 4 диода и 1 резистор. Элементы добавляются в пространство рабочей панели. Чтобы повернуть элемент, нужно нажать на правую кнопку мыши и выбрать угол поворота (рис. 4): 45, 90 или 135 градусов.

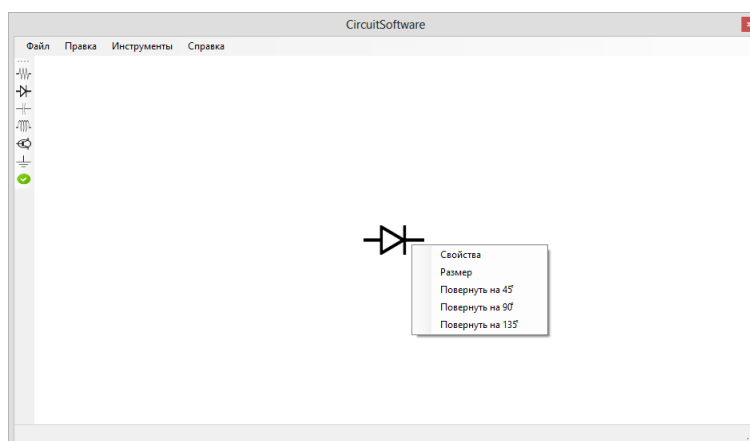


Рисунок 4 — Поворот элемента

Добавляются все остальные элементы, получается схема без связей (рис. 5):

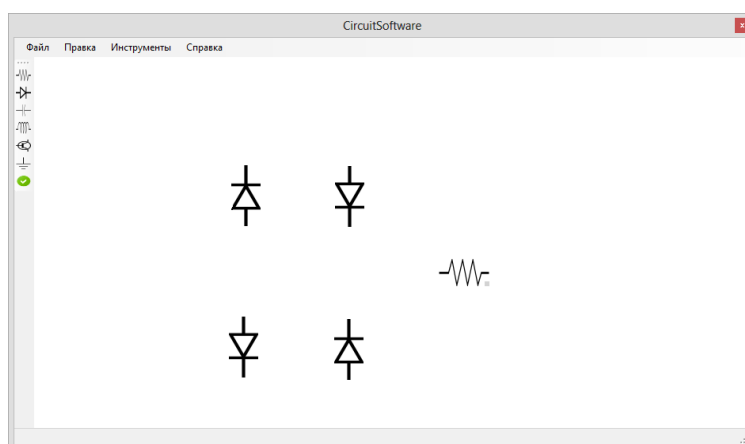


Рисунок 5 — Схема без связей

Для добавления связей нужно зайти в меню *Инструменты* и нажать на *Создать связь*. Создаются связи между элементами с образованием готовой схемы (рис. 6).

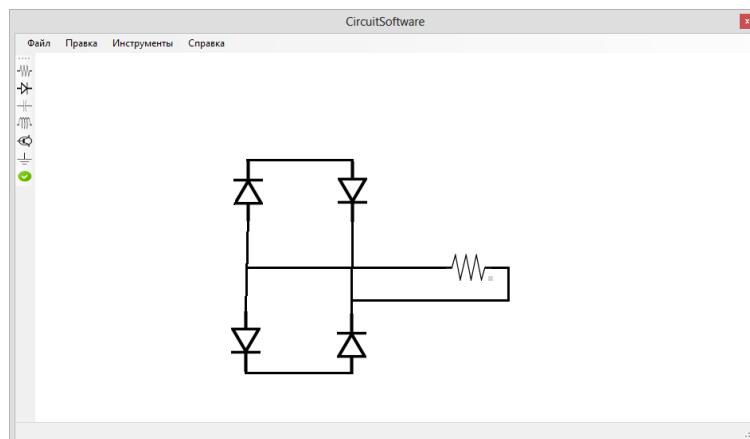


Рисунок 6 — Схема со связями

После создания всех связей и проверки (самим тестируемым) на правильность схемы нужно нажать на кнопку *Check* на панели инструментов. В результате создаётся квадратная матрица:

	D1	D2	D3	D4	R
D1	0	-1	1	0	1
D2	1	0	0	-1	-1
D3	1	0	0	-1	1
D4	0	-1	1	0	-1
R	-1	1	-1	1	0

Для автоматической проверки построенной схемы потребуется предварительно заданный правильный вариант ответа в виде матрицы (шаблон). Размерность матрицы шаблона будет зависеть от количества элементов, используемых в схеме.

В настоящее время в программе реализованы все выше продемонстрированные функции. Задачей дальнейших исследований является нахождение метода преобразования матриц (матрица шаблона, проверочная матрица). При преобразовании матриц найденным методом матрицы должны преобразоваться таким образом, чтобы все элементы исходной и проверочной матриц оказались равными, т.е. одинаковыми по размеру и по значениям элементов. Так же будет предусмотрено применение данного программного обеспечения по другим дисциплинам, возможность создавать, редактировать и удалять шаблоны по предметам, а в последующем — интегрирование данного типа тестов в систему тестирования ИКИТ.

СПИСОК ЛИТЕРАТУРЫ

1 Камиллов, И. К. Методологические проблемы автоматизированного формирования образовательных программ в рамках ФГОС ВПО / С. А. Броннов, Е. А. Степанова, Ю. М. Кудрявцева, Н. С. Афонасенко, И. К. Камиллов // Современные информационные технологии и ИТ-образование : сб. избр. тр. VII Научно-практической конференции в МГУ им. М. В. Ломоносова 9—11.11.2012 / под ред. В. А. Сухомлина. — М. : ИНТУИТ.РУ, 2012. — С. 103—111. — ISBN 978-5-9556-0140-3.