

Покрытие схемы модулями из заданного набора

*М.В. Платонов,
студ. 3-го к., krompyg@gmail.com,
И.В. Герасимов,
студ. 3-го к., gerasimov_981@mail.ru,
Л.А. Старостина,
доц. к.т.н., доц., StarostinaLA@gmail.com,
НИУ МЭИ, г. Москва*

Целью работы является исследование возможностей использования облачных технологий при создании систем автоматизированного проектирования. Рассматривается реализация алгоритма покрытия схемы несвязными модулями из заданного набора и полученные результаты.

The aim of the research is to study the possibilities of using cloud technology in computer-aided design systems. How incoherent scheme coverage algorithm modules from a given set of results are shown.

Сегодня особую актуальность приобретают концепции «мобильных работников» и «домашнего офиса», которые предполагают удаленную работу с основными корпоративными системами. В настоящее время идет массированная миграция традиционных приложений и систем на веб-интерфейсы. Например, специалисты Autodesk выпустили мобильные приложения для Android, iPad и iPhone online - конструктор circuits.io, который позволяет строить схему и печатную плату, в приложении есть множество модулей, для построения и тестирования электронных схем Упрощенный интерфейс предоставляет минимальный набор команд для быстрого взаимодействия с существующими проектами.

Тема предлагаемого доклада связана с разработкой Web приложения для решения одной из задач конструкторского проектирования "покрытие схемы несвязными модулями из заданного набора". Приложение состоит из нескольких связанных друг с другом модулей. Это собственно алгоритм покрытия, графический редактор и база данных, содержащая набор микросхем различных типов. В докладе рассматриваются структура разработанной системы, возможности интерфейса, принципы работы алгоритма покрытия схемы, направления дальнейшего развития проекта и его улучшения.

Типовые микросхемы и вентили хранятся в отдельной облачной базе данных. Ввод данных от пользователя осуществляется через сайт, ему предоставляется возможность сделать собственный набор вентиляей, из которых построена схема, а также набор микросхем, которыми эта схема должна покрываться. На следующем этапе пользователь, при помощи интерфейса сайта, должен собрать свою схему. Сборка схемы заключается в том, что на рабочем поле устанавливаются вентили, а затем создаются связи между этими вентилями. Помимо этого, имеется возможность получить схему, создавая ее по определенным правилам, либо описать ее в текстовом файле прописывая все вентили и связи между ними. Либо сначала построить схему при помощи интерфейса сайта, затем скачать файл с этой схемой и использовать его для дальнейшей работы. Схему можно редактировать удалять или добавлять новые вентили или новые связи между вентилями. Если схема слишком большая, для удобства построения и редактирования система позволяет схему масштабировать. На этапе покрытия схемы микросхемами пользователь может самостоятельно, при помощи редактора покрыть некоторые вентили микросхемами, если это необходимо, это покрытие обязательно будет присутствовать в итоговом покрытии.

Используются критерии качества при покрытии схемы:

- Минимизация связей между элементами.
- Максимальная заполняемость микросхем.
- Максимальная частота повторения каждого типа микросхем.
- Минимизация типов микросхем.

Порядок работы программы:

- Получение схемы соединения элементов от пользователя.
- Преобразование этой схемы в компьютерное представление.
- Работа алгоритма (составление набора микросхем).
- Выдача результатов пользователю.

Полный алгоритм представлен на рис.1.

В настоящее время частично реализован общий алгоритм, позволяющий выполнять покрытие схемы для любого набора вентиляей и микросхем. Реализованный алгоритм позволяет выполнять покрытие как для односвязных схем, так и для многосвязных. Среди особенностей можно отметить то, что все вычисления производятся на сервере (в облаке), а не на ПК пользователя, это позволяет даже на маломощных машинах быстро получать хорошие результаты.

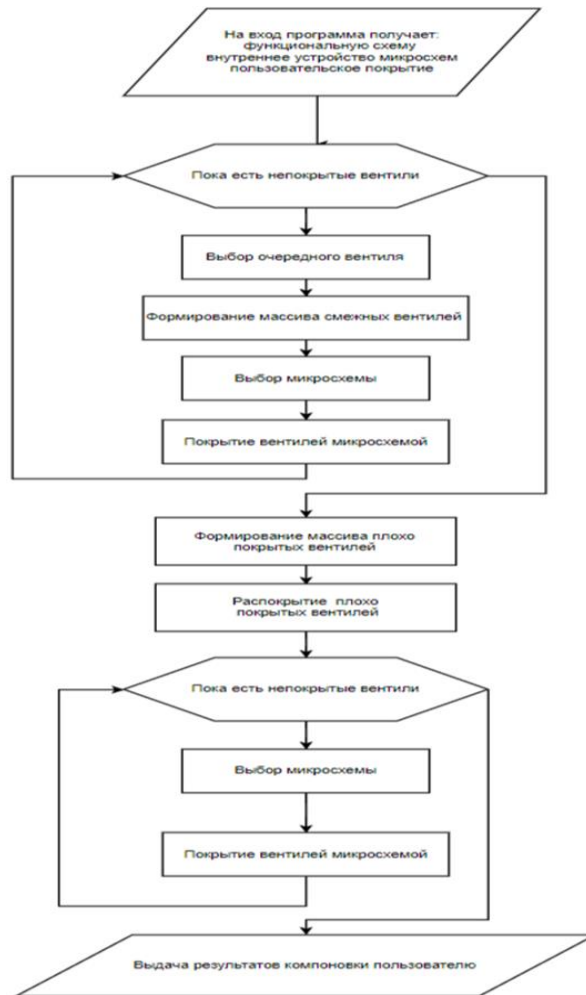


рис.1 Блок-схема алгоритма "покрытия"

Работа алгоритма проверялась на следующем наборе вентилях и микросхем, который представлен на рис.2.

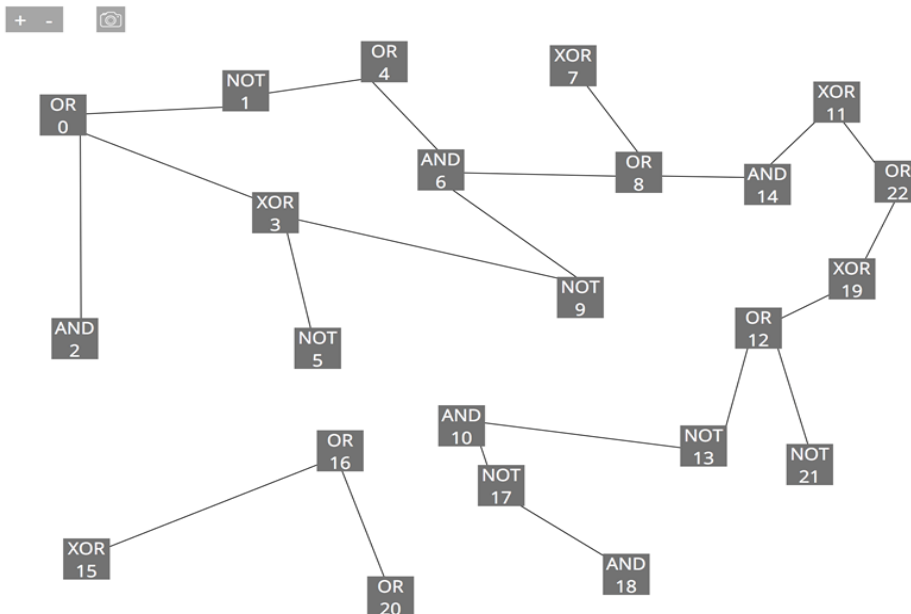


рис.2 Исходная схема для тестирования алгоритма

Результат работы алгоритма предоставлялся пользователю в указанном на рисунке 3 виде.

Вам понадобятся:

количество: 12	0 4 0 8 0 5 1	2 3 3 1 2 0	0 1 1 0 0	2 1 9 3
гво первого типа: 3	-----			
гво второго типа: 4	-----			
гво третьего типа: 5	-----			
	2 1 1 3 2 0 0	0 9 1 1 6 0	1 1 3 1 1 4 2	2 7 3
	1 2 2 6 2	1 1 0 2 1 7 1	1 1 8 2	2 1 5 3
	Назад		Скачать файлом	

рис.3 Предоставляемые пользователю результаты

Возможен ввод данных через форму, представленную на рис.4.

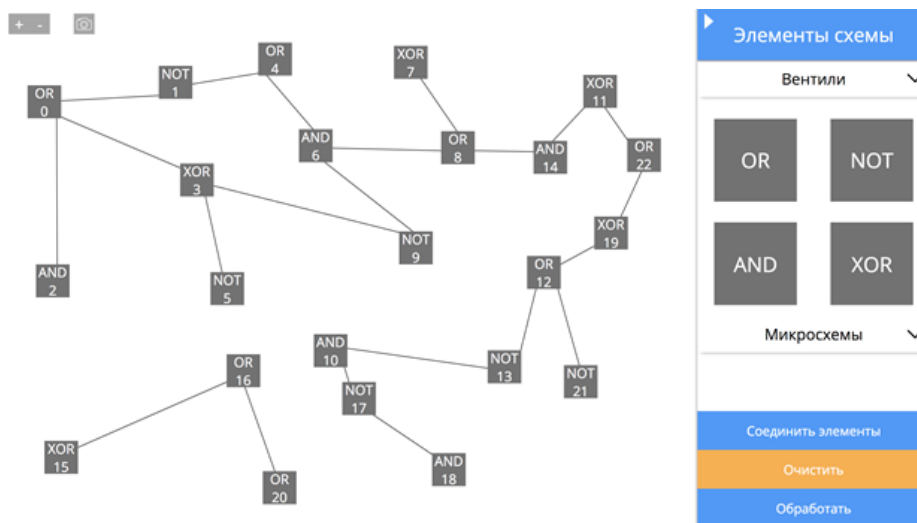


рис.4 Форма для ввода данных

Все вычисления производятся на сервере (в облаке), а не на ПК пользователя, это позволяет даже на маломощных машинах быстро получать хорошие результаты. Большая часть интерфейса реализована и работает. Программное обеспечение написано на C++. Первая версия тестировалась на схемах размерностью до 30 и модулях 3 типов. Для верстки интерфейса web-сайта использовались языки HTML5, CSS3. Работа с событиями была написана на VanillaJS (NativeJS). Сервер был написан на Python с использованием фреймворка Flask. Во время разработки сервер был развернут на локальном компьютере.

Литература

1. Курейчик В.М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР. - М: Радио и связь, 1990-352с., с.143-150.