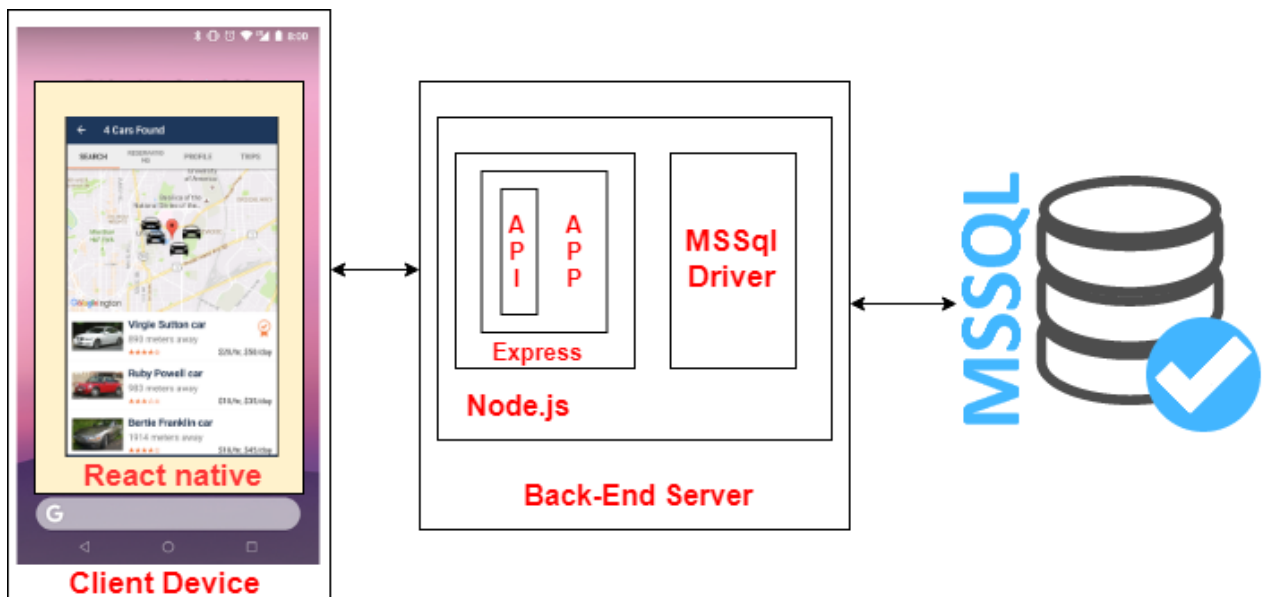


Приложение позволяет просматривать список или карту с местоположениями доступных автомобилей, фильтруемых по различным параметрам, войти в систему или авторизоваться в системе, оставлять отзывы, добавлять в избранное и бронировать автомобиль. Функционирование в приложении могут осуществлять два вида пользователей: авторизованный и неавторизованный. Неавторизованному пользователю доступна только функция просмотра списка автомобилей. Авторизованный пользователь имеет более обширный функционал, такой как просмотр списка автомобилей, вход в личный кабинет, просмотр истории бронирований, рекомендаций и, непосредственно, бронирование. Для авторизации пользователю необходимо создать аккаунт или воспользоваться авторизацией через социальную сеть.

Для создания гибридного мобильного приложения используется фреймворк React Native. React— JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React Native — это JS-фреймворк для создания нативно отображаемых iOS- и Android-приложений на основе библиотеки React. Основным преимуществом данной библиотеки является кроссплатформенность и нативность.

В качестве библиотеки, реализующей backend данного приложения, используется Node.js. Это серверная реализация языка программирования JavaScript, основанная на движке V8. Предназначена для создания масштабируемых распределённых сетевых приложений, таких как веб-сервер. В качестве базы данных используется MSSQL.



Список использованных источников:

1. Marc H. Node.js in Action / Marc Harter, Mike Cantelon, Nathan Rajlich // Manning Publications, – 2017. – 371 с.
2. Информационные технологии и управление : материалы 49 науч. конф. аспирантов, магистрантов и студентов, Минск, 6–10 мая 2013 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – 103 с.
3. Nader Dabit. React Native in Action / Nader Dabit. – Manning Publications Company, 2019. – 300 с.

МЕТОДИКА ПРОЕКТИРОВАНИЯ КОМБИНАЦИОННОГО УСТРОЙСТВА ВСЕВОЗМОЖНЫХ ПЕРЕСТАНОВОК

Кохновский С.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Иванюк А.А. – д.т.н., профессор

В данной работе рассмотрено комбинационное устройство для создания всевозможных отображений N входных сигналов на N выходов. Предлагается синтез схемы такого устройства для произвольного $N \geq 2$.

При создании различных систем встаёт задача обеспечения тайны их внутренней конструкции. Такая задача порождена необходимостью защиты авторских прав. Злоумышленники прибегают к обратному проектированию систем, чтобы заполучить технологии их разработки. Для усложнения обратного проектирования системы можно использовать запутывающие устройства, которые значительно повышают сложность такого проектирования [1]. В работе рассмотрен подход к проектированию такого запутывающего устройства. Каждое устройство имеет N входных, N выходных сигналов и ключ, состоящий из набора M бит, оно может создавать всевозможные отображения входных сигналов на выходы в зависимости от подаваемого ключа.

Всевозможное количество ключей для устройства будет равно 2^M , а количество возможных выходных комбинаций – $M!$. Тогда будет справедливо следующее соотношение

$$M \geq \lceil N! \rceil,$$

где M – число блоков, при которых возможно составление всевозможных комбинаций, а N – число этих комбинаций.

Некоторым комбинациям путей будет соответствовать несколько ключей. Тогда $2^M - M!$ комбинаций будут повторными, что может негативно сказаться на эффективности работы такого устройства.

Простейшее комбинационное устройство (рисунок 1 а) состоит из блоков, каждый из которых принимает 2 входных сигнала, 2 выходных и бит ключа, который указывает на порядок следования сигналов. При подаче 0 сигналы не изменяют направления, такое направление будет называться прямым, а при подаче 1 сигналы меняют пути, что является противоположным направлением.

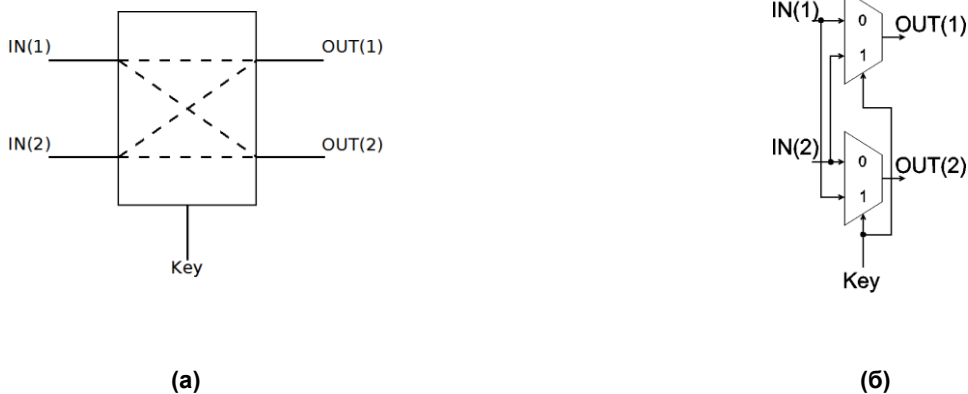


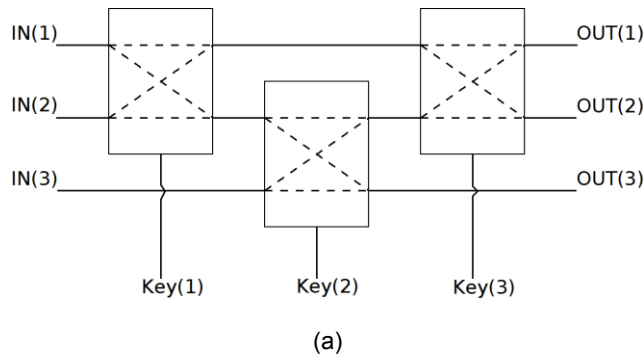
Рисунок 1. Блок (а), внутреннее устройство блока (б)

Каждый блок состоит из мультиплексоров [2], осуществляющих смену направления сигналов по вышеописанной схеме (рисунок 1 б).

Простейшим устройством является устройство с двумя входами и выходами, то есть $N = 2$, блок, как определялось ранее. Такому устройству надо будет один бит, чтобы определять направление сигналов, откуда $M = 1$.

Пусть будет поставлена задача построения устройства при $N = 3$ (рисунок 2 а), блок будет иметь входные сигналы $IN(1)$, $IN(2)$, $IN(3)$ и выходные $OUT(1)$, $OUT(2)$, $OUT(3)$. При проектировании такого устройства будем использовать 3 блока, тогда минимально возможное $M = 3$. Первый блок будет соединять пути $IN(1)$ и $IN(2)$, второй – $IN(2)$ и $IN(3)$, а третий снова соединит пути $IN(1)$ и $IN(2)$. Первый блок «перемешивает» пути $IN(1)$ и $IN(2)$, тогда выходы $OUT(2)$ и $OUT(1)$ будут достижимы из $IN(1)$ и $IN(2)$. Блок №2 обеспечит достижимость $OUT(3)$ из $IN(1)$ и $IN(2)$, также $OUT(2)$ из $IN(3)$. Можно заметить, что недостижимым остался только выход $IN(1)$ из входа $OUT(3)$, чтобы это компенсировать и есть третий блок. Таким образом, блоки до центрального включительно в комплексе соединяют вход $IN(1)$ со всеми остальными выходами, а блок с номером 2, расположенный в центре, являющийся первым для входа $OUT(3)$, вместе с последующими блоками соединяют все выходы для входа $OUT(3)$. Все остальные входы были соединены с выходами первой половины блоков или второй.

Пусть будет рассмотрена матрица для описания устройства при $N = 3$ (рисунок 2 б). Строка будет обозначать блок, соединяющий 2 пути, а столбцы обозначают соединяемые пути. Откуда следует, что в строке может быть только 2 единицы, обозначающие, что данные пути соединяются блоком. Верхняя строка будет обозначать блок-1, вторая – блок-2 и так далее. Столбцы в строке могут переставляться местами, что не влияет на корректность работы устройства. При таких перестановках будут получены матричные представления схем устройств, которые отличаются порядком соединения путей, однако другими характеристиками они отличаться не будут. Строки произвольно переставляться не могут.



	OUT(1)	OUT(2)	OUT(3)
Key(1)	1	1	0
Key(2)	0	1	1
Key(3)	1	1	0

(б)

Рисунок 2. Комбинационное устройство с тремя входами (а), матрица, описывающая комбинационное устройство с тремя входами (б)

Следуя логике построения устройства при $N = 3$, можно обобщить на случай с произвольно заданным количеством N входных и выходных путей (рисунок 3). Будет последовательно соединяться первый и второй пути, второй и третий, ... , $N-1$ и N (всего $N-1$ блок). Эта последовательность гарантировано ставит на выходную позицию N необходимый входной сигнал. Таким образом, из первого входа будут достижимы N выходов, из второго, в худшем случае, будет перенаправлен сигнал в первый после их обмена, то же произойдёт с входными сигналами 3, 4, ..., N . Теперь будет обеспечена достижимость N -ым входом 1-го выхода путём добавления блоков, последовательно соединяющих пути $N-1$ и $N-2$, $N-2$ и $N-3$, ..., 3 и 2, 2 и 1. После построения этой последовательности своё место займёт сигнал, который должен стоять на первом выходе. Далее будут пересекаться таким же образом остальные пути, чтобы расставить оставшиеся сигналы по нужным позициям. Пересекутся пути с 2-го блока до $N-1$, потом с $N-2$ по 3-й и так далее. Тогда для устройства с N выходными и N выходными путями необходимо $M = N - 1 + N - 2 + N - 3 + \dots + 2 + 1$ блоков, которые нуждаются в M -битном ключе. После преобразования получим

$$M = \frac{N}{2} * (N - 1).$$

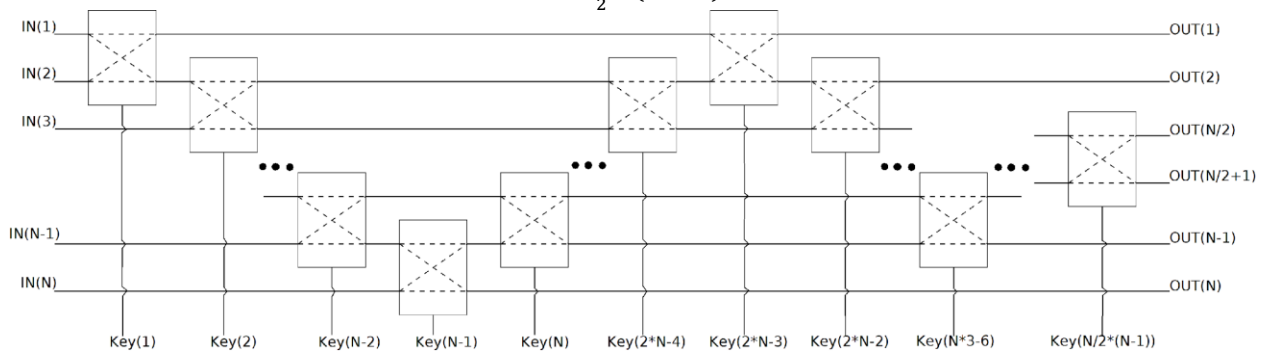


Рисунок 3. Комбинационное устройство с N входными сигналами

Рассмотренная схема устройства строится достаточно просто, но такая схема минимальной не будет. Ранее была выведена формула связи M и N , из неё следует возможность построения устройства с минимальным размером ключа $M = [N!]$. Возможны и другие конфигурации устройств, которые будут давать необходимый результат.

Таким образом, была разработана схема комбинационного устройства, осуществляющего отображение входных сигналов на выходы согласно подаваемому ключу.

Список использованных источников:

1. Privacy protection of VLSI circuits through high level transformation based obfuscation [Electronic resource]. – Mode of access: https://www.researchgate.net/publication/319135767_PRIVACY_PROTECTION_OF_VLSI_CIRCUITS_THROUGH_HIGH_LEVEL_TRANSFORMATION_BASED_OBFUSCATION. – Date of access: 18.03.2019.
2. Угрюмов, Е.П. Цифровая схемотехника: учеб. пособие для вузов/ Е. П. Угрюмов. – СПб. : БХВ-Петербург, 2010. – 816 с.