

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ

**VIII Международной молодежной научной
конференции**

**«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»**

Томск, 26–30 мая 2021 г.

Под общей редакцией И.С. Шмырина

Томск
Издательство Томского государственного университета
2021

All implementations also use 4 flip-flops and dedicated multiplexers F7 MUX, F8 MUX [7] (16 and 8, respectively).

Conclusion

This article presents research results on different substitution representation techniques for FPGA implementation of a reconfigurable finite-state machine based on 4-bit substitutions. Various substitution representation techniques are used, namely table representation (memory, substitution table), ARX-formula, butterfly structure, system of Boolean functions (disjunctive/algebraic normal forms). Nevertheless, all designs, except the one, where substitutions are described as RAM memory, have roughly the same utilization and timing parameters. However, when the fixed substitutions are designed as RAM memory, the R-FSM implementation requires significantly more LUTs and has a slightly lower maximum clock frequency.

REFERENCES

1. *Koster M., Teich J.* (Self-)reconfigurable finite state machines: theory and implementation // Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition – 2002 – P. 559–566.
2. *Glaser J., Damm M., Haase J., Grimm C.* TR-FSM: Transition-based Reconfigurable finite state machine // ACM Transactions on Reconfigurable Technology and Systems (TRETS). – 2011. – V. 4. – № 3. – Article №23.
3. *Das N., Aruna Priya P.* FPGA Implementation of Reconfigurable Finite State Machine with Input Multiplexing Architecture Using Hungarian Method // International Journal of Reconfigurable Computing. – 2018. – V. 2018. – Article ID 6831901 – 15 p.
4. *Trenkaev V.N.* Reconfigurable finite-state machine based on substitutions // *Prikladnaya Diskretnaya Matematika. Prilozhenie.* – 2019. – № 12. – P. 192–193.
5. *Trenkaev V.N.* Zakrevskij's cipher based on reconfigurable FSM // *Prikladnaya Diskretnaya Matematika.* – 2010. – № 3(9). – P. 69–76.
6. *Harris D.M., Harris S.L.* Digital Design and Computer Architecture. Second edition. – Morgan Kaufmann, 2013. – 720 p.
7. Series FPGAs Configurable Logic Block (UG474) – URL: https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf
8. *Komissarov S.M.* On algorithmic implementation of 16-bit S-boxes with ARX and Butterfly structures // *Prikl. Diskr. Mat. Suppl.* – 2019 – 12 – P. 101–107.
9. *Fomin D.B.* New classes of 8-bit permutations based on a butterfly structure // *Mat. Vopr. Kriptogr.* – 2019. – V. 10. – № 2. – P. 169–180.
10. *Ashenden P.J.* The designer's guide to VHDL. 3rd ed. – Elsevier Inc., 2008.
11. Vivado Design Suite User Guide : Synthesis (UG901) – URL: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_2/ug901-vivado-synthesis.pdf

DOI: 10.17223/978-5-907442-42-9-2021-36

ЧТЕНИЕ ДАННЫХ С ЭКСПЕРИМЕНТАЛЬНОЙ УСТАНОВКИ

Безходарнов Н.И., Самохина С.И.

*Томский государственный университет
nblaaa@mail.ru*

Введение

В одной из лабораторий Томского кардиологического центра ведётся работа по снятию показаний с нагруженной сердечной мышцы. Для этого в лаборатории имеется оборудование, предназначенное для экспериментов с сердечной мышцей. Именно для этой установки была поставлена задача разработать информационную систему, в которой выполняется снятие показаний при проведении эксперимента, их оцифровка и математическая обработка. Установка для экспериментов состоит из считывающего устройства, модификаторов сигнала и преобразователя аналогового сигнала в цифровой. Для считывания данных необходимо связаться с преобразователем. Далее полученная кривая аппроксимируется с помощью метода сетки, выполняется анализ полученных данных и визуализация.

1. Постановка задачи

Рассматривается экспериментальная установка, предназначенная для чтения показателей напряжённости сердечных мышц. Чтобы получить с этой установки данные, требуется понять, как она передаёт выходные сигналы, и разработать модуль, предназначенный для чтения данных. На выходе должна быть получена кардиологическая кривая, которая в дальнейшем подвергается обработке, а именно – аппроксимации двумя компонентами экспоненциального вида. Задачей данной работы является получение с экспериментальной установки кардиологической кривой и её двухкомпонентное разложение [1–3].

Основной целью данного проекта является разработка информационной системы, реализующей чтение данных с установки и разложение кардиологической кривой в функцию следующего вида:

$$y = a_1 e^{-b_1(x-c_1)^2} + a_2 e^{-b_2(x-c_2)^2}. \quad (1)$$

Для достижения данной цели были сформулированы следующие задачи:

1. Изучение экспериментальной установки.
2. Разработка модуля чтения данных с установки.
3. Реализация модуля аппроксимации функции для реализации алгоритма.
4. Объединения всех модулей в информационную систему.

2. Функциональные возможности информационной системы

В качестве средств реализации выбрана среда разработки Qt Creator с использованием библиотек Qt [4]. Данная среда является удобной в разработке графического интерфейса, а также способна обеспечить кроссплатформенность. Библиотеки Qt предоставляют широкий спектр возможностей для разработки различного рода приложений.

Функционал разрабатываемой программы включает в себя следующее: чтение кардиологической кривой из устройства, получающее данные при проведении эксперимента, или из файлов, содержащих результаты прошлых экспериментов; отображение графиков на основе полученных данных; нахождение разложения кардиологической кривой, её построение и вывод подробной информации о результатах. Для реализации данного функционала были разработаны модули чтения данных с установки и написан алгоритм, позволяющий найти аппроксимирующую кривую к кардиологической.

3. Описание алгоритма получения данных с экспериментальной установки

Для реализации модуля чтения данных, прежде всего, требовалось понять, как устроен вывод сигнала в экспериментальной установке. Установка состоит из устройства, считывающего показания напряжённости, множества модификаторов сигнала и преобразователя аналогового сигнала в цифровой. Для получения сигнала достаточно установить связь с этим преобразователем и получить от него данные в числовом виде. Экспериментальная установка представляет собой устройство модели DT9802, которое преобразовывает аналоговый сигнал в числовой вид. Данное устройство было разработано компанией Measurement Computing. Также данной компанией была разработана библиотека Universal Library [5], которая позволяет взаимодействовать с подобными преобразователями. Эта библиотека и была использована при разработке модуля. Для полного взаимодействия с устройством необходимо осуществить следующие действия:

1. Получить список устройств.
2. Подключиться к нужному устройству.
3. Выставить все необходимые параметры.
4. Запустить процесс чтения данных.
5. Получить данные с устройства.
6. Остановить процесс чтения данных.
7. Отключиться от устройства.

Получение списка устройств осуществляется функцией `olDaEnumBoards`, которая принимает функцию пополнения списка на 1 элемент и указатель на сам список. В случае отсутствия подключенных устройств эта функция возвращает ошибку.

Подключение к устройству осуществляется функцией `olDaInitialize`, которая принимает на вход название устройства и структуру для работы с драйвером. Пока программа не подключится к устройству, дальнейшая работа с ним невозможна. При подключении требуется подготовить устройство к чтению данных, а именно: выставить режим работы `OL_DF_CONTINUOUS` для непрерывного чтения данных командой `olDataSetDataFlow`, выставить режим работы буферов `OL_WRP_MULTIPLE` для возможности их повторного использования командой `olDataSetWrapMode` и выставить частоту 1000 Гц командой `olDaSetClockFrequency`.

Далее, необходимо выставить все параметры устройства, а именно: канал (`olDaSetChannelListEntry`), фильтр (`olDaSetChannelFilter`), границы (`olDaSetRange`) и множитель (`olDaSetGainListEntry`). Получить всевозможные вариации данных параметров можно с помощью функции `olDaEnumSSCaps`, в которой требуется указать необходимый тип параметра, функцию для пополнения списка на 1 элемент и указатель на список. Затем, получив и выбрав из предоставленных вариантов нужные, требуется убедиться в том, что все параметры были выставлены верно. Данная процедура осуществляется функцией `olDaConfig`. После этого уже можно начать считывать данные.

Для запуска чтения данных прежде всего нужно подать на устройство дескриптор окна, т.к. передача данных осуществляется посредством системных сигналов, отправляемых приложению. Осуществляется это командой `olDaSetWndHandle`. Далее требуется выделить на устройстве память для хранения прочитанных данных, чтобы их можно было передать компьютеру. Выделением памяти занимается функция `olDmCallocBuffer`, а после неё требуется вызвать функцию `olDaPutBuffer`, чтобы привести созданный буфер в состояние готовности к записи. После этого можно запустить чтение данных с помощью функции `olDaStart`.

Передача данных осуществляется посредством системных сигналов. Для того, чтобы получить эти сигналы, нужно использовать функцию `nativeEvent`, являющуюся частью библиотеки Qt. Т.к. сигналы бывают разные, то сначала требуется убедиться, что мы получили нужный. При получении сигнала приходит его идентификатор, по которому можно определить тип сообщения. Для считывания данных с устройства нужно получить сигнал с идентификатором `OLDA_WM_BUFFER_DONE`, в этом случае можно воспользоваться буфером, возвращаемым функцией `olDaGetBuffer`, затем узнать количество валидных элементов функцией `olDmGetValidSamples` и получить указатель на массив данных функцией `olDmGetBufferPtr`. Далее данные считываются с устройства и сохраняются в программе для дальнейшей обработки.

После того, как все данные были получены, нужно остановить чтение. Для этого достаточно вызвать функцию `olDaStop`, затем циклом получить все буферы функцией `olDaGetBuffer` и освободить их функцией `olDmFreeBuffer`.

Остаётся только отключиться от устройства с помощью следующих 2 функций: `olDaReleaseDASS` (освобождает устройство) и `olDaTerminate` (отключает программу от драйвера).

Таким образом осуществляется взаимодействие с устройством. Описанная выше концепция работы позволяет получить валидные данные, которые могут в дальнейшем быть подвергнуты обработке.

4. Описание алгоритма аппроксимации

Алгоритм, предложенный в [6], решает задачу аппроксимации функции определённого вида. Начальными данными для него является кардиологическая кривая, которая задаётся с помощью последовательности точек на плоскости. Ставится задача получения из такого набора точек функции вида (1).

Для решения этой задачи используется метод сетки, который заключается в следующем: задаётся начальное приближение и некоторый шаг для отступа, затем вычисляются значения функции в точках, отстоящих от текущей на заданный шаг, причём отступ происходит во все возможные стороны. Из всех вычисленных значений выбирается минимальное и запоминаются координаты точки, при которых это значение достигается. Далее данные действия повторяются до тех пор, пока в найденной точке функция не достигнет требуемого минимума.

5. Интерфейс программы

Графический интерфейс реализован в виде окна, содержащего кнопки, текстовые поля и инструменты для построения графиков. При запуске программы появляется одно главное окно. Для начала работы требуется загрузить данные о кардиологической кривой. Это можно сделать, нажав кнопку «Построить график», предварительно выбрав в списке выше то, откуда считать данные. Есть возможность получать данные либо из файлов, либо с экспериментальной установки. В главном окне появится специальное поле, где отобразится график, с которым можно взаимодействовать с помощью указателя мыши. Для удобства задания промежутка аппроксимации на курсоре мыши в поле с графиками отображается чёрная вертикальная линия, помогающая ориентироваться. После задания промежутка активируется кнопка «Начать аппроксимацию», нажав которую начнутся вычисления. Во время вычислений главное окно станет неактивным, и на нём появится окно с просьбой подождать. После вычислений отображаются результаты аппроксимации (рис. 1).

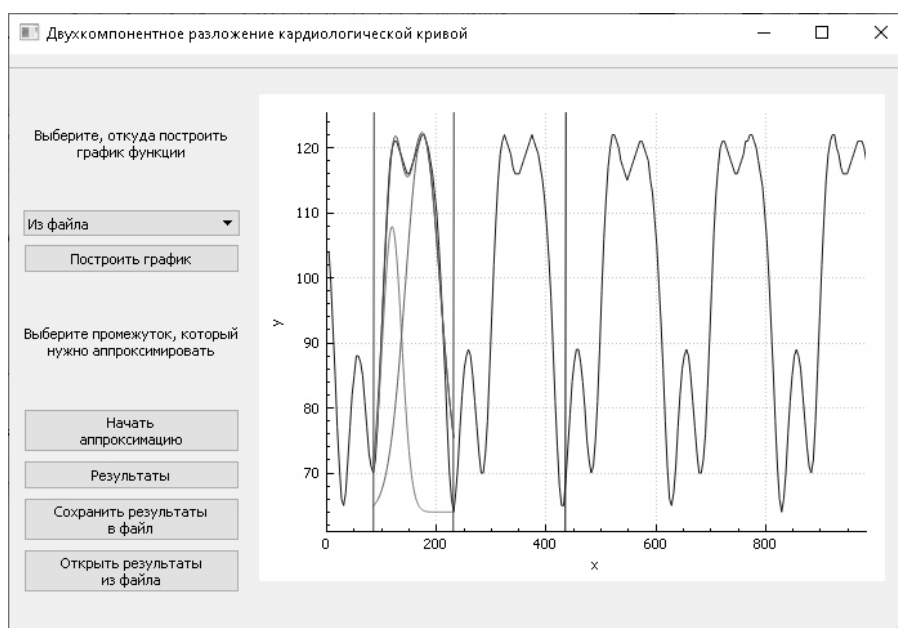


Рис. 1. Главное окно программы после вычислений

Также после вычислений станет активной кнопка «Результаты», нажав которую, можно открыть отдельное окно с различной информацией о найденной функции: график функции, её компонент и её первых двух производных (рис. 2), таблица значений функции и её производных, аналитический вид и другое. Можно сохранить результаты вычислений в файл, чтобы позже загрузить их в программу без повторного пересчёта.

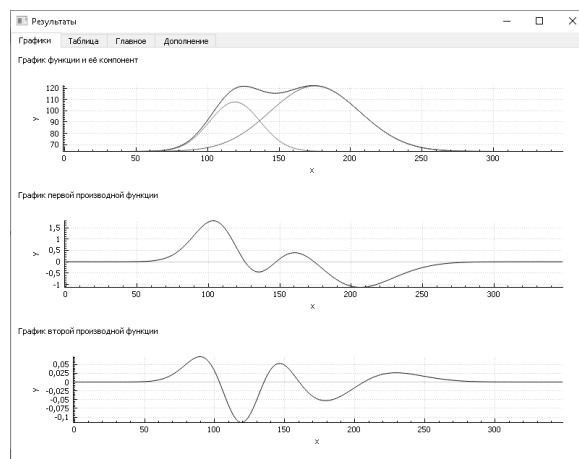


Рис. 2. Окно информации о результатах

В результате реализована информационная система с удобным графическим интерфейсом, которая решает поставленную задачу.

Заключение

В данной работе создана информационная система, которая осуществляет чтение данных с экспериментальной установки, сохраняет данные в виде кардиологической кривой и её разложение в экспоненциальную функцию. Данные с устройства считываются корректно и целостно сохраняются в файл, аппроксимация находится достаточно близкая к оригинальной функции, результаты выводятся на экран наглядно и информативно. Интерфейс разработан интерактивный и понятный. Алгоритм аппроксимации протестирован на множестве входных данных, и получившиеся результаты оказались достаточно корректными.

На данный момент система находится на этапе внедрения в производственный процесс проведения эксперимента с сердечной мышцей. Надеемся, что в будущем данная информационная система окажет немалое влияние в нашей жизни и поможет сделать её лучше и безопаснее.

ЛИТЕРАТУРА

1. Богомаз С.А. Исследование действия кардиологических веществ на пулы кальция с помощью анализа компонентов структуры сокращения миокарда: дис. ... канд. мед.наук. – Томск, 1991. – 184 с.
2. Тарасенко В.Ф., Лаптев Б.И. Способ компонентного анализа сокращения миокарда // Сборник трудов молодых ученых ТНЦ АМН. – 1991. – С. 44–47.
3. Самохина С.И., Шишкин М.М. Программный комплекс моделирования двухкомпонентного разложения кардиологической кривой // Математическое и программное обеспечение информационных, технических и экономических систем : материалы VI Междунар. молодежной науч. конф. Томск, 24–26 мая 2018 г. – Томск: Издательский Дом ТГУ, 2018. – С. 83–88.
4. Qt [Электронный ресурс]: Qt Reference Pages – URL: <https://doc.qt.io/qt-5/reference-overview.html>
5. Data Translation, Inc. Dtx-EZ™ Getting Started Manual [Текст]: руководство пользователя / Data Translation, Inc. – Marlboro.: Data Translation, Inc., 2006. – 234 с.
6. Безходарнов Н.И., Самохина С.И. Двухкомпонентное разложение кардиологической кривой // Математическое и программное обеспечение информационных, технических и экономических систем : материалы Международной научной конференции, Томск, 28–30 мая 2020 г. – Томск: Изд-во Том. ун-та, 2020. – С. 86–90.