



## МОДЕЛИРОВАНИЕ ВСТРАИВАЕМОЙ СИСТЕМЫ НА ПРОГРАММИРУЕМОЙ ЛОГИЧЕСКОЙ ИНТЕГРАЛЬНОЙ СХЕМЕ

**Болдырев А. В.**

*Федеральное государственное бюджетное образовательное учреждение высшего образования «Донской государственный технический университет», Ростов-на-Дону, Российская Федерация, e-mail: avb49@aanet.ru*

В статье рассматривается процесс моделирования встраиваемой системы на программируемой логической интегральной схеме учебного стенда с использованием программного обеспечения Intel/Altera. Цель исследования – создание модели встраиваемой системы в учебных условиях на основе комплексного цикла проектирования системы на программируемой микросхеме. В рамках исследования описан полный цикл проектирования – от настройки аппаратной части до реализации программного обеспечения. В качестве процессорного ядра применяется экономичный вариант Nios II, размещенный на программируемой микросхеме Altera серии Cyclone. Конфигурирование вычислительной системы выполняется в среде SOPC Builder с формированием переключательной структуры Avalon. Аппаратная часть разрабатывается на языке описания аппаратуры Verilog в среде Quartus II, где также задаются параметры микросхемы, генерируется конфигурационный файл и настраиваются выводы микросхемы. В состав системы входят периферийные устройства: клавиатура и жидкокристаллический дисплей, обеспечивающие ввод данных и отображение результатов. Программная часть создается на языке C/C++ с помощью инструментов Nios II среды разработки Eclipse и компилируется в исполняемый файл. Верификация проекта проводится на примере приложения «Калькулятор»: конфигурационный файл загружается в память стенда с помощью специальной утилиты, после чего тестируется взаимодействие компонентов. Методика может быть применена в учебных лабораториях вузов для освоения принципов проектирования встраиваемых систем на программируемой логической интегральной схеме.

**Ключевые слова:** моделирование, учебный стенд, Intel/Altera, Verilog, Quartus II, SOPC Builder

## MODELING OF AN EMBEDDED SYSTEM ON A PROGRAMMABLE LOGIC INTEGRATED CIRCUIT

**Boldyrev A. V.**

*Federal State Budgetary Educational Institution of Higher Education “Don State Technical University” Rostov-on-Don, Russian Federation, e-mail: avb49@aanet.ru*

The article examines the process of modeling an embedded system on a programmable logic integrated circuit of a training stand using Intel/Altera software. The aim of the research is to create a model of an embedded system in educational conditions based on a comprehensive system design cycle on a programmable chip. The research describes the full design cycle – from configuring the hardware to implementing the software. An economical version of the Nios II processor core, deployed on an Altera Cyclone series programmable chip, is used as the processing unit. The computing system is configured in the SOPC Builder environment, which generates the Avalon switching structure. The hardware component is developed using the Verilog hardware description language in the Quartus II environment, where the chip parameters are set, the configuration file is generated, and the chip pins are configured. The system includes peripheral devices – a keyboard and a liquid-crystal display – that enable data input and output. The software component is created in C/C++ using the Nios II tools in the Eclipse development environment and compiled into an executable file. Project verification is carried out using a “Calculator” application as an example: the configuration file is loaded into the stand’s memory via a dedicated utility, after which the interaction of components is tested. The proposed methodology can be applied in university teaching labs to master the principles of designing embedded systems on programmable logic integrated circuit.

**Keywords:** modeling, training stand, Intel/Altera, Verilog, Quartus II, SOPC Builder

### Введение

Моделирование и исследование встраиваемых систем имеют критическое значение в современной технике и науке по ряду причин, связанных с их ролью в технологиях, экономике, образовании и безопасности [1–3].

Встраиваемые системы часто работают в условиях ограниченных ресурсов – вычислительной мощности, памяти, энерго-

потребления. Моделирование позволяет заранее оценить, как система будет вести себя при различных нагрузках, и оптимизировать ее архитектуру.

Многие встраиваемые системы используются в критически важных приложениях – медицинских устройствах, аэрокосмических системах, автомобильной электронике, системах безопасности. Ошибка в таких системах может иметь серьезные

последствия, включая угрозу жизни людей. Моделирование и верификация на ранних этапах разработки помогают выявить потенциальные проблемы до физической реализации, что значительно повышает надежность и безопасность систем.

Моделирование позволяет проводить виртуальное прототипирование и тестирование до создания физического прототипа. Это сокращает время и стоимость разработки, так как многие ошибки и недочеты можно исправить на этапе моделирования.

Встраиваемые системы – это сочетание аппаратных и программных компонентов (HW/SW-проектирование). Моделирование позволяет интегрировать и анализировать взаимодействие между аппаратной и программной частями, что особенно важно при работе со сложными системами с многопроцессорной архитектурой или распределенными вычислениями [4–6].

Моделирование позволяет быстро адаптировать системы к новым технологиям, например, интегрировать машинное обучение или искусственный интеллект в встраиваемые системы. Это особенно актуально в эпоху Интернета вещей (IoT), где встраиваемые системы играют ключевую роль в обмене данными между устройствами.

В образовательных учреждениях моделирование встраиваемых систем используется для обучения студентов принципам проектирования, программирования и отладки. Виртуальные модели и симуляторы позволяют студентам экспериментировать, не рискуя повредить дорогостоящее оборудование.

Современные технологии проектирования встраиваемых систем активно используют программируемые логические интегральные схемы (ПЛИС) – они дают широкие возможности для реализации сложных вычислительных структур с гибкой настройкой под конкретные задачи [7; 8]. Особую роль ПЛИС играют в образовательном процессе: они позволяют студентам на практике изучить принципы построения цифровых систем, отработать навыки работы с языками описания аппаратуры и освоить методики интеграции аппаратных и программных компонентов.

Таким образом, моделирование встраиваемых систем на ПЛИС – это не просто этап разработки, а стратегический инструмент, который позволяет создавать более надежные, эффективные и инновационные решения в условиях постоянно растущих требований к технологиям.

**Цель исследования** – создание модели встраиваемой системы в учебных условиях на основе комплексного цикла проектирования системы на ПЛИС.

## Материалы и методы исследования

Типичная встраиваемая система на ПЛИС состоит из одного или нескольких процессоров, соединенных с памятью и другими устройствами с помощью шинной структуры [9, с. 389]. Главной особенностью создания учебных встраиваемых систем является сочетание модулей, образованных с помощью языка описания аппаратуры, например Verilog, и процессора с программируемым ядром, оптимизированного для работы на ПЛИС учебного стенда.

Для создания прототипа встраиваемой системы использовался учебный стенд SDK-6.1/E, хорошо известный в российских вузах [10; 11]. Стенд оснащен основной и вспомогательной ПЛИС фирмы Intel/Altera, а также подсистемой памяти, необходимой для построения полнофункционального софт-процессора [12] на основе ядра Nios II. Применительно к возможностям основной ПЛИС стенда (Altera Cyclone EP1C6)<sup>1</sup> целесообразно использовать «экономичный» вариант ядра Nios II/e, который является основой 32-битного RISC-процессора и использует не более 700 логических элементов ПЛИС, а также всего два блока памяти по 4К.

На рис. 1 приведен пример учебной встраиваемой системы, состоящей из клавиатуры, жидкокристаллического дисплея (ЖКД), их драйверов и вычислительного устройства с программным ядром Nios II/e.

Данные и выполняемые операции вводятся с клавиатуры. Драйвер клавиатуры подготавливает параллельные данные для чтения процессором. Процессор читает эту информацию через свой интерфейс, выполняет определенные операции и передает результат на интерфейс ЖКД для отображения. В состоянии готовности драйвер ЖКД принимает данные из своего параллельного порта и отображает их на дисплее ЖКД.

Клавиатура и дисплей являются физическими устройствами, которые присоединяются к выводам ПЛИС.

Для моделирования встраиваемой системы используются такие программные средства, как среда проектирования Quartus II компании Intel/Altera с инструментом для конфигурирования системы на ПЛИС SOPC Builder и шинной структурой Avalon<sup>2</sup>, а также язык описания аппаратуры Verilog [13].

<sup>1</sup> Cyclone FPGA Family [Электронный ресурс]. URL: <https://www.chipfind.ru/datasheet/pdf/altera/ep1c6.pdf> (дата обращения: 22.01.2026).

<sup>2</sup> Quartus II Web Edition (Win) [Электронный ресурс]. URL: <https://soltau.ru/index.php/plis/item/360-install-quartus/> (дата обращения: 22.01.2026).

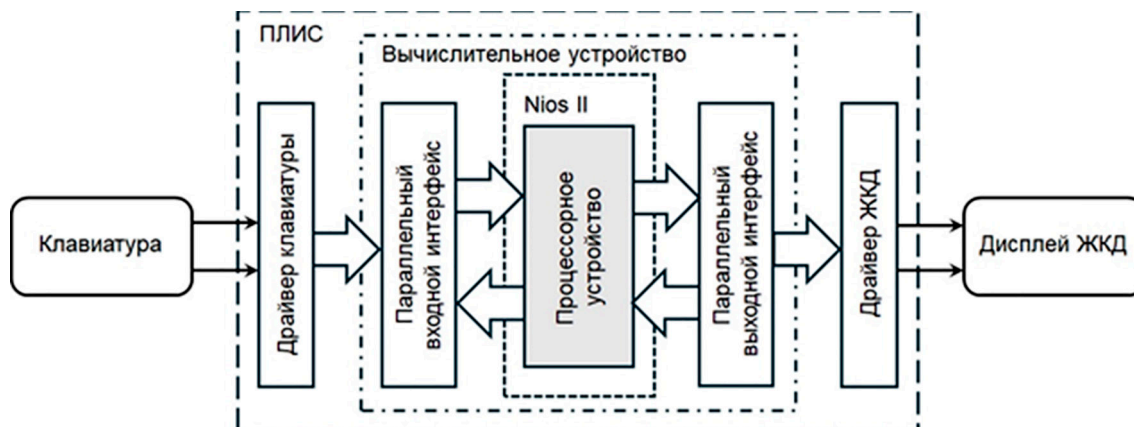


Рис. 1. Пример встраиваемой системы  
Примечание: составлен автором по результатам данного исследования

Методическими и справочными материалами исследования являются руководство пользователя стенда SDK-6.1/E и официальная документация по Quartus II и Nios II.

Перечисленные материалы образуют замкнутый цикл разработки модели: от аппаратного проектирования (Verilog, SOPC Builder) до программного кода (C/C++) и физической реализации на стенде, что позволяет отработать навыки работы с инструментами (Quartus II, Nios II), понять интеграцию НW/SW во встраиваемых системах, получить опыт верификации и отладки на реальном оборудовании.

В ходе учебного моделирования применен комплексный методический подход, объединяющий инструментальные, проектные и экспериментальные методы для проектирования и верификации встраиваемой системы на ПЛИС [14].

Для построения целостной структуры системы, включающей процессор, память, периферию и интерфейсы, использовался метод системного проектирования.

Для описания логики работы периферийных устройств (драйверов клавиатуры и ЖКД) использовался метод аппаратного проектирования на основе HDL с применением готовых модулей, интегрируемых во встраиваемую систему.

Метод программного проектирования позволил реализовать алгоритмическую часть системы на уровне программного обеспечения на C/C++ для процессора Nios II с использованием для компиляции инструментов Nios II среды разработки Eclipse и последующей отладки программной логики.

С целью минимизации ошибок перед финальной загрузкой в стенд осуществлялась пошаговая отладка на каждом этапе раз-

работки модели, состоявшая из проверки HDL-кода на синтаксис и семантику, тестирования драйверов периферии в изолированном режиме, пошагового выполнения программного обеспечения в среде Eclipse и мониторинга сигналов на выводах ПЛИС.

Для подтверждения работоспособности модели встраиваемой системы была проведена экспериментальная проверка путем загрузки разработанных модулей в память и ПЛИС стенда SDK-6.1/E. Тестирование функционала осуществлялось с помощью приложения «Калькулятор» [9, с. 436].

### Результаты исследования и их обсуждение

Учебная модель, включающая драйверы клавиатуры и ЖКД [15], определяется как проект пакета Quartus II компании Intel/Altera, а вычислительное устройство становится файлом проекта в программе SOPC Builder, являющейся составной частью пакета.

Создание и конфигурирование проекта встраиваемой системы выполняется через меню New в ряде страниц Quartus II, в которых задаются название и место расположения проекта, выбирается семейство ПЛИС (Cyclone) и конкретная микросхема (EP1C6T144C8), используемая в стенде SDK-6.1/E. Так как для стенда требуется конфигурационный файл с расширением \*.rbf, необходимо отметить опцию включения его создания при компиляции проекта. Для этого в меню Quartus II следует открыть командой Assignments/Device окно настроек, в котором нажать кнопку Device and Pin Options, а затем выбрать в открывшемся окне вкладку Programming Files и проверить, отмечен ли галочкой пункт Raw Binary File (.rbf). Во избежа-

ние выхода из строя внешних устройств или БИС на плате стенда в связи с тем, что по умолчанию неиспользованные выводы ПЛИС соединяются с общей шиной, в этом же окне необходимо выбрать вкладку Unused Pins и установить As input tristated. Для создания файла, содержащего описание структурной схемы выбирается File/New/Design Files, а в таблице – Block Diagram/Schematic File. В результате открывается окно графического редактора с загруженным в него bdf-файлом, который необходимо сохранить, задав имя и установив флажок Add file to current project.

Конфигурирование вычислительного устройства осуществляется в программе SOPC Builder, в которой устройство определяется как файл системы на программируемом кристалле (СнПК). Программа, вызываемая из меню Tools, автоматизирует задачу объединения элементов аппаратуры в большую систему, в том числе за счет создания переключательной структуры Avalon.

При создании файла СнПК задается имя системы, выбирается язык HDL (Verilog), определяются семейство ПЛИС (Cyclone) и частота тактового генератора (40 МГц для стенда SDK-6.1/Е), после чего выбираются и добавляются в систему необходимые модули из библиотечного списка. Процедура проводится в окне бланка программы SOPC Builder и сопровождается серией окон, позволяющих выбрать желаемые опции.

Первым в окно компонентов бланка устанавливается процессор Nios II с отказом от опции отладчика No Debugger для упрощения проекта. Размещение процессора автоматически вызывает размещение главной шины структуры Avalon для памяти команд ROM и главной шины для памяти данных RAM.

Добавление на следующих двух шагах обоих типов памяти On-Chip Memory, размещаемой внутри ПЛИС, сопровождается окном спецификации, в котором выбираются тип памяти (ROM и RAM), ее размер (32 bits) и объем (4К). После появления новых модулей в бланке СнПК необходимо дифференцировать имена типов памяти, подсоединение шин структуры Avalon и установить адреса компонентов так, как это показано на рис. 2.

При добавлении параллельных интерфейсов ввода-вывода PIO в окне спецификации на каждый из них в зависимости от назначения указывается тип (входной или выходной) и разрядность (1-битный или 8-битный). Интерфейс PIO, используемый для входа KeyReleased от клавиатуры, является 1-битным входным портом, опра-

шиваемым процессором. Поэтому в одном из конфигурационных окон он определяется как входной порт прерывания IRQ. Аппаратура всех интерфейсов уже является частью переключательной структуры Avalon.

Дополнительные настройки процессора на последнем шаге конфигурирования СнПК сохраняются без изменений. Таким образом, в систему на программируемом кристалле добавляются процессорное ядро Nios II/e, два модуля памяти On-Chip Memory, два модуля PIO для клавиатуры и три модуля PIO для дисплея. В результате полное содержимое вычислительного устройства приобретает вид, показанный на рис. 2.

По окончании конфигурирования для создания кода описания аппаратуры осуществляется генерирование системы (кнопка Generate на рис. 2). При отсутствии ошибок работа с программой SOPC Builder завершается, в противном случае вносятся исправления и генерация системы повторяется. С успешным завершением создания вычислительного устройства формируется символ для данной СнПК – NIOS (рис. 3).

Полученные файлы вручную добавляются в проект встраиваемой системы, как и другие файлы, описывающие логику работы системы. Размещение их в окне графического редактора наряду с драйверами клавиатуры и ЖКД, другими компонентами, а также выполнение соединений ничем не отличается от известных методик. Для выполнения назначений входных и выходных сигналов проекта выводам корпуса реальной ПЛИС Cyclone используется карта ее выводов, задействованных в работе стенда SDK-6.1/Е, а сама процедура назначений осуществляется с помощью графического редактора Pin Planner.

Полная структурная схема встраиваемой системы, соответствующая учебному примеру (рис. 1), приведена на рис. 3.

Построение программной части является последним этапом проектирования встраиваемой системы. Эта фаза компилирует программу на языке C/C++, помещаемую в память ROM проекта СнПК. Для создания программы используются инструменты Nios II Software Build Tools for Eclipse из меню Tools. В мастере создания приложения для конфигурирования пакета поддержки платы указывается файл описания СнПК, выбирается процессор, для которого создается приложение, а также название проекта. В результате формируется исполнимый elf-файл, из которого перед загрузкой в ПЛИС выделяется машинный код.

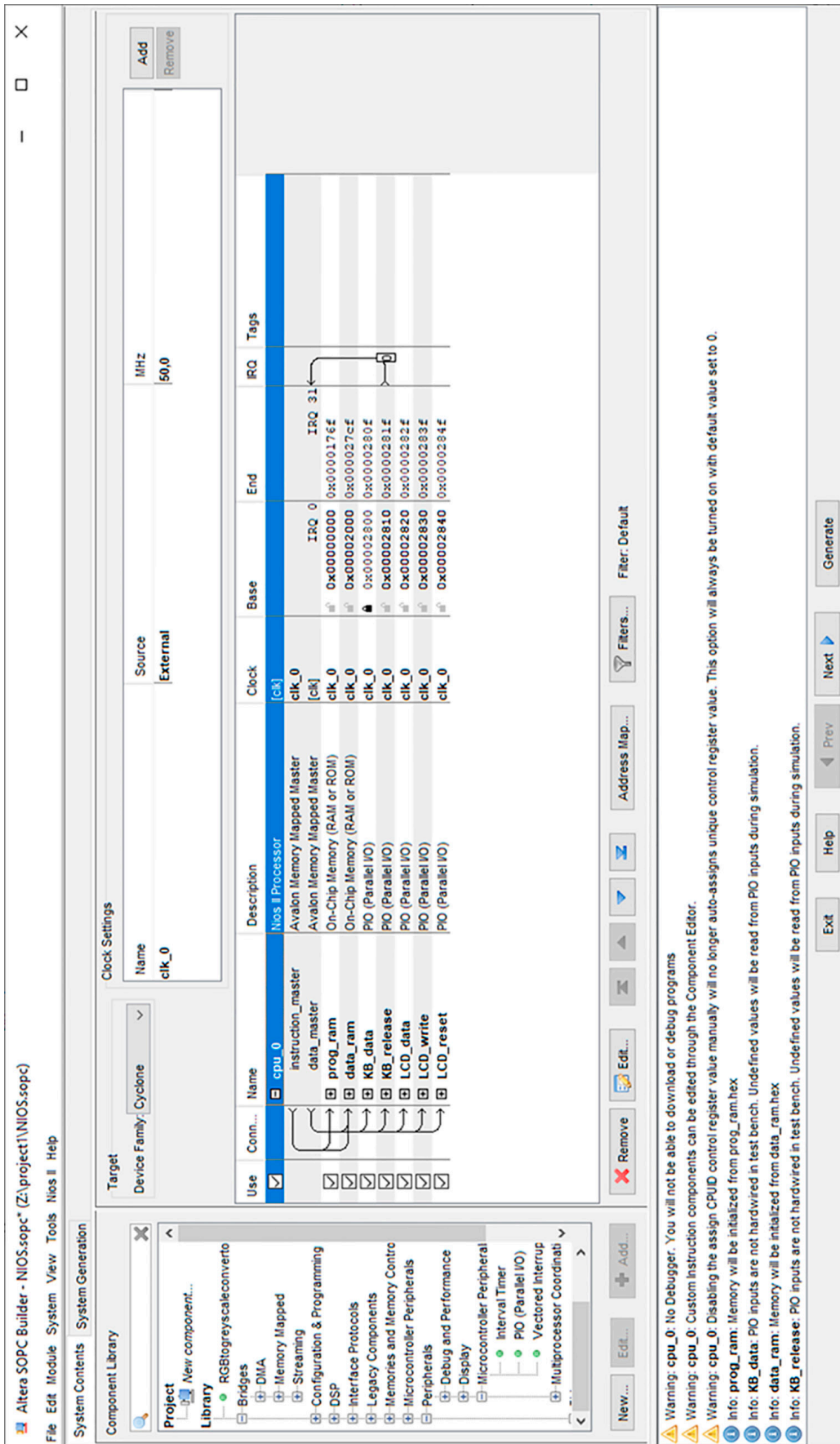


Рис. 2. Конфигурирование вычислительного устройства  
Примечание: составлен автором по результатам данного исследования

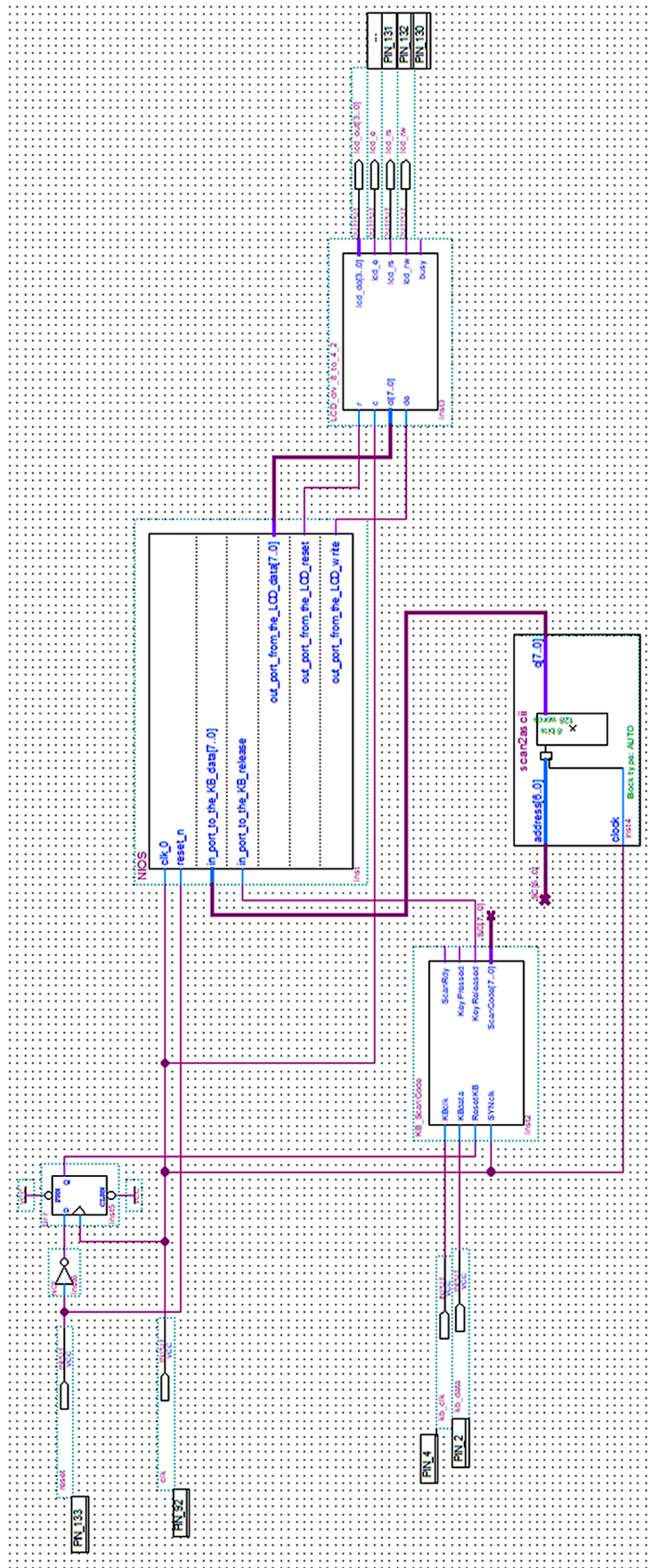


Рис. 3. Схема учебной модели встраиваемой системы  
Примечание: составлен автором по результатам данного исследования

Для верификации проекта, после завершения его создания и компиляции выполняется синтез gbf-файла, содержащий в том числе машинный код для процессора Nios II. С помощью утилиты sdk61fw.exe [11] файл проекта загружается в любую страницу Flash-памяти стенда, не используемую для хранения конфигурационных файлов, данных или программ (согласно инструкции стенда SDK-6.1/E). По окончании загрузки к стенду подключается клавиатура по модифицированному интерфейсу PS/2.

В качестве примера функционирования учебной модели встраиваемой системы на стенде SDK-6.1/E была реализована работа приложения «Калькулятор». Результаты верификации подтверждают корректную работу модели в условиях ограниченного в ресурсах учебного стенда.

### Заключение

В ходе исследования решены поставленные задачи: разработана, реализована и верифицирована модель встраиваемой системы на ПЛИС в условиях учебного стенда.

Отработан полный цикл проектирования от спецификации архитектуры до загрузки конфигурационного файла в реальное оборудование. Продемонстрирована интеграция аппаратной (Verilog) и программной (C/C++) частей системы. Создана работоспособная система на базе конфигурируемого процессора Nios II/e. Подтверждена эффективность методических решений.

Исследование подтвердило практическую реализуемость встраиваемых систем на ПЛИС в учебных условиях, образовательную ценность методики для формирования инженерных компетенций, потенциал масштабирования решения для промышленных прототипов.

Полученные результаты могут служить основой для дальнейших научных работ в области ПЛИС и встраиваемых систем, модернизации учебных курсов по цифровой логике и микропроцессорной технике, разработки специализированных аппаратно-программных комплексов.

Несмотря на ограничения исследования (привязанность к экосистеме Intel/Altera, упрощенность периферии и отсутствие операционной системы), материалы статьи могут быть полезны для освоения цикла разработки от схемы до исполняемого кода, а также понимания роли ПЛИС как оптимальной платформы для прототипирования процессорных систем.

Результаты работы представляют интерес для студентов, изучающих проектирование на ПЛИС, преподавателей, разрабатывающих учебно-методические комплексы, и инженеров, начинающих работу с ПЛИС и Nios II.

### Список литературы

1. Lee E. A., Seshia S. A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach. 2nd ed. MIT Press, 2017. 568 с. ISBN 978-0-262-53381-2.
2. Пенской А. В. Архитектурное документирование встроенных систем с многоуровневой конфигурацией // Известия вузов. Приборостроение. 2015. № 7. С. 527–532. URL: <https://elibrary.ru/item.asp?id=23815546> (дата обращения: 22.01.2026). DOI: 10.17586/0021-3454-2015-58-7-527-532.
3. Аржаев В. И., Скворцов А. В. Особенности применения объектно-ориентированного проектирования встроенных систем жесткого реального времени // Программные продукты и системы. 2016. № 1. С. 56–59. URL: <https://swsys.ru/index.php?page=article&id=4110> (дата обращения: 22.01.2026). DOI: 10.15827/0236-235X.113.056-059.
4. Платунов А. Е. Теоретические и методологические основы высокоуровневого проектирования встраиваемых вычислительных систем. СПб.: Университет ИТМО, 2010. 477 с. [Электронный ресурс]. URL: <https://search.rsl.ru/ru/record/01004917432> (дата обращения: 22.01.2026).
5. Alten J. P., Fürst S. Model-based Design of Embedded Systems with Matlab/Simulink // ATZ Elektron Worldw. 2016. Vol. 11. P. 60–63. URL: <https://link.springer.com/article/10.1007/s38314-016-0019-5> (дата обращения: 09.02.2026). DOI: 10.1007/s38314-016-0019-5.
6. Оскин Д. А., Громашева О. С., Дьяченко М. Е. Модельно-ориентированный подход для автоматизации генерации программного С-кода для встраиваемых систем с использованием модели Matlab/Simulink // Современные наукоемкие технологии. 2018. № 10. С. 92–97. URL: <https://top-technologies.ru/ru/article/view?id=37201> (дата обращения: 09.02.2026). DOI: 10.17513/spno.37201.
7. Строгонов А. В., Городков П. С. Особенности проектирования устройств цифровой обработки сигналов в базе ПЛИС серии 5578 // Известия вузов. Электроника. 2017. Т. 22. № 3. С. 256–265. URL: [http://ivuz-e.ru/issues/3-2017/osobennosti\\_proektirovaniya\\_ustroystv\\_tsifrovoy\\_obrabotki\\_signalov\\_v\\_bazise\\_plis\\_serii\\_5578/](http://ivuz-e.ru/issues/3-2017/osobennosti_proektirovaniya_ustroystv_tsifrovoy_obrabotki_signalov_v_bazise_plis_serii_5578/) (дата обращения: 09.02.2026). DOI: 10.24151/1561-5405-2017-22-3-256-265.
8. Лютерович О. В. Особенности проектирования компонентов на ПЛИС и их применение во встраиваемых системах // Reviews of Modern Science. 2023. № 4. URL: <https://ojs.scipub.de/index.php/RMS/article/view/2323> (дата обращения: 22.01.2026).
9. Наваби З. Проектирование встраиваемых систем на ПЛИС / пер. с англ. Соловьева В. В. М.: ДМК Пресс, 2016. 464 с. ISBN 978-5-97060-174-7.
10. Степаненко Д. Р., Болдырев А. В. Обзор учебных стендов для обучения программированию ПЛИС // Инновации и инжиниринг в формировании инвестиционной привлекательности региона: сборник научных трудов II открытого международного научно-практического форума. Ростов-на-Дону: Издательство ДГТУ-ПРИНТ, 2017. С. 446–451. URL: <https://elibrary.ru/item.asp?id=32329091> (дата обращения: 07.02.2026).
11. Ключев А. О., Платунов А. Е., Дергачев А. М. Опыт использования лабораторных стендов SDK в учебном процессе // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 1. С. 184–187.

URL: <https://ntv.ifmo.ru/file/article/18451.pdf> (дата обращения: 22.01.2026). DOI: 10.17586/2226-1494-2019-19-1-184-187.

12. Тарасов И. Е., Потехин Д. С., Платонова О. В. Перспективы применения софт-процессоров в системах на кристалле на базе программируемых логических интегральных схем // Russian Technological Journal. 2022. Т. 10 № 3. С. 24–33. URL: <https://www.rtfj-mirea.ru/jour/article/view/519> (дата обращения: 17.01.2026). DOI: 10.32362/2500-316X-2022-10-3-24-33.

13. Емец С. Verilog – инструмент разработки цифровых электронных схем // Компоненты и технологии. 2001. № 4 (13). С. 76–79. URL: <https://elibrary.ru/item.asp?edn=mtxawx> (дата обращения: 07.02.2026).

14. Ляпунцова Е. В., Ахмад А. Особенности процесса автоматизированного проектирования встраиваемых систем и возможность их реализации в программном обеспечении проектирования микропроцессорных систем // Известия ТулГУ. Технические науки. 2024. № 1. С. 368–369. URL: <https://scinetwork.ru/articles/7736> (дата обращения: 07.02.2026). DOI: 10.24412/2071-6168-2024-1-368-369.

15. Болдырев А. В. Интерфейс дисплея для учебного стенда на основе программируемой логической интегральной схемы // Научное обозрение. Технические науки. 2025. № 5. С. 12–18. URL: <https://science-engineering.ru/ru/article/view?id=1521> (дата обращения: 09.02.2026). DOI: 10.17513/srts.1521.

**Конфликт интересов:** Авторы заявляют об отсутствии конфликта интересов.

**Conflict of interest:** The authors declare that there is no conflict of interest.