



ОТКРЫТАЯ ПЛАТФОРМА НА ОСНОВЕ ПРОЦЕССОРНОГО ЯДРА RISC-V ДЛЯ ВСТРАИВАЕМЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ НА ОСНОВЕ ПЛИС

OPEN PLATFORM BASED ON PROCESSOR CORE WITH RISC-V ARCHITECTURE FOR EMBEDDED FPGA-BASED CONTROL SYSTEMS

УДК 004.383

ЛИВЕНЦЕВ ЕВГЕНИЙ ВАСИЛЬЕВИЧ¹

liventsev@gmail.com

LIVENTSEV EVGENII V.¹

liventsev@gmail.com

СИЛАНТЬЕВ АЛЕКСАНДР МИХАЙЛОВИЧ¹

SILANTIEV ALEXANDER M.¹

ПАВЛОВ АНТОН НИКОЛАЕВИЧ²

PAVLOV ANTON N.²

¹ НИУ МИЭТ

¹ National Research University of Electronic Technology — MIET

² ФГУ ФНИЦ НИИСИ РАН

² SRISA RAS

В данной работе представлена открытая платформа для встраиваемых информационно-управляющих систем на основе ПЛИС, базирующаяся на микропроцессорном ядре с архитектурой RISC-V (подмножество RV32IM).

Ключевые слова: системы на ПЛИС; свободное ПО; открытая платформа; RISC-V.

This article presents open platform for embedded FPGA-based control systems built on processor core with RISC-V architecture (RV32IM).

Keywords: FPGA-based systems; open software; open platform; RISC-V.

ПОТРЕБНОСТЬ В СОЗДАНИИ

В настоящее время существует большое множество реализаций архитектуры RISC-V в виде микроконтроллерных вычислительных ядер. Большинство из них снабжены примерами программ и ограниченными наборами тестов. Но, как показывает практика, конечный пользователь склонен выбирать не конкретный контроллер, а набор возможностей, обеспеченный совокупностью аппаратной и программной частей.

В нашем понимании термин «платформа» означает не только вычислительное ядро, но и инфраструктуру для решения широкого круга задач, встающих перед разработчиком при попытке применить это ядро в своем проекте. В отличие от вычислительного ядра, снабженного демо-проектом, платформа должна упрощать ряд типовых задач. В первую очередь платформа должна:

- обеспечить возможность расширять аппаратную часть пользовательскими сложно-функциональными блоками (СФ-блоками);
- обеспечить поддержку аппаратной части, в том числе пользовательских СФ-блоков в ПО;
- предоставить возможность моделирования работы аппаратной и программной частей.

ОБЗОР ПЛАТФОРМ НА ОСНОВЕ ВЫЧИСЛИТЕЛЬНЫХ ЯДЕР С АРХИТЕКТУРОЙ RISC-V

Если пользоваться таким определением, то можно выделить две наиболее развитые в данный момент платформы на основе вычислительных ядер с архитектурой RISC-V:

- Pulpino, от группы разработчиков из Высшей технической школы Цюриха;

- SiFive, от создателей архитектуры RISC-V из университета Беркли.

Обе эти платформы пользуются поддержкой внешних компаний. Рассмотрим их подробнее.

Pulpino [1] — открытый проект одноядерной микроконтроллерной системы. Проект включает в себя два ядра: RISKY с четырехстадийным конвейером и поддержкой наборов команд RV32ICMF и zero-risky — более компактное ядро с двухстадийным конвейером с поддержкой наборов команд RV32CIME.

Pulpino содержит сценарии сборки для популярных отладочных плат ZYBO и Zedboard, а также сценарии создания программного обеспечения, написанные на языке Python.

Основным недостатком платформы, на наш взгляд, является то, что упомянутые сценарии уникальны для этого проекта и не позволяют пользователю комфортно добавлять в систему собственные модули.

SiFive Freedom [2] — это открытый проект, включающий генератор RocketChip, написанный на Chisel HDL. Но даже при наличии сценариев сборки и примеров программного обеспечения, применять SiFive крайне тяжело из-за специфики маршрута проектирования аппаратуры с помощью Chisel HDL.

Несмотря на то что эти два проекта сильны и развиты, в настоящее время актуальным является создание платформы на основе RISC-V ядра, более ориентированной на удобство применения. Так как вычислительное ядро и платформа создается для запуска не ней программ, то, на наш взгляд, платформа должна обеспечивать комфортную работу прежде всего программисту.

Своей целью мы ставим создание платформы, позволяющей специализировать микросистему, включить в нее необходимые



пользовательские СФ-блоки и обеспечить при этом удобство написания ПО для полученной системы.

Таким образом, основными компонентами, которые должна включать разрабатываемая платформа, являются:

- конфигуратор микросистемы;
- набор основных контроллеров периферии;
- библиотеки ПО для работы с периферией;
- гибкая программная модель.

Помимо этих компонент необходимо обеспечить возможность автоматизированной сборки сконфигурированной системы и запуска тестов.

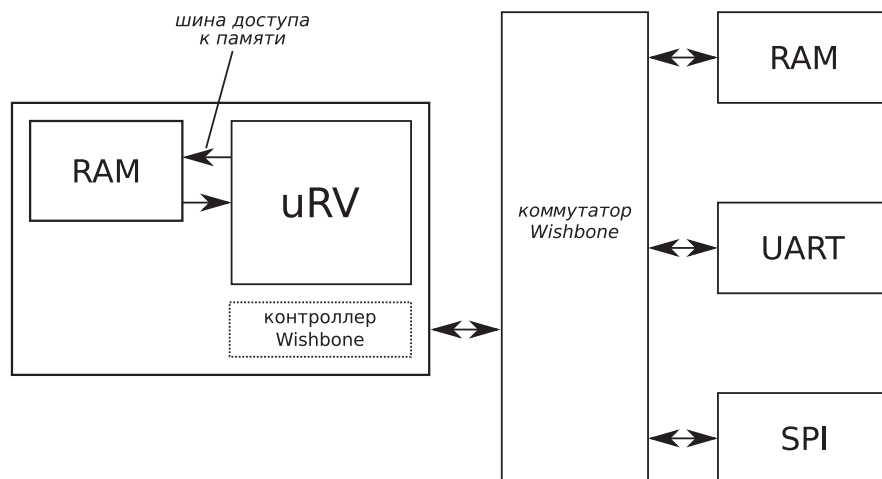


Рис. 1. Структура прототипа платформы

ПРОТОТИП ПЛАТФОРМЫ

Достижение такой цели требует ряда шагов. Ниже описан первый технический результат на пути к созданию платформы — прототип, выполненный в виде одноядерной микроконтроллерной системы, рабочее название «Муравейник» (Anthill).

Структурно микросистема не обладает высокой сложностью и состоит из модифицированного микроконтроллера uRV [3] от CERN и нескольких контроллеров периферийных интерфейсов (см. рис. 1). Модификации ядра касались доработок в реализации контроллера прерываний для адаптации к текущим черновикам спецификаций и расширению разрядности системного таймера.

Несмотря на простую структуру, на прототипе были отработаны несколько вариантов маршрута проектирования, выбраны и опробованы вспомогательные инструменты для проектирования. В результате этой работы были заложены основы аппаратного проектирования для платформы.

В качестве инструмента для автоматизации процесса сборки был выбран инструмент FuseSoC [4]. Этот инструмент позволяет составлять описания СФ-блоков, вести контроль версий и контроль зависимостей частей микросистемы, а также обеспечивает автоматизированную сборку проекта по маршрутам, параметры которых может определять пользователь. Таким образом FuseSoC позволяет строить модульные системы, вести совместную работу над проектом и дорабатывать отдельные узлы и СФ-блоки независимо от остальной системы.

В процессе работы над прототипом были сформированы файлы описания СФ-блоков и проведены доработки во FuseSoC: реализована поддержка IP-блоков Xilinx при моделировании в xsim из пакета Vivado. В результате этих работ налажена автоматизированная сборка и моделирование проекта для целевой платы Digilent Artu с ПЛИС Xilinx Artix 35T. При этом существует возможность быстро перенести проект на другие ПЛИС Xilinx.

В процессе разработки прототипа было принято решение использовать QEMU [5] в качестве системного симулятора для проверки ПО. QEMU позволяет проводить моделирования с высокой скоростью за счет механизма динамической двоичной трансляции. Немаловажным свойством QEMU является возможность расширения за счет добавления Си-моделей аппаратных блоков.

ПРИМЕНЕНИЕ ПРОТОТИПА ПЛАТФОРМЫ

Прототип платформы был успешно применен в трех проектах:

- в качестве контроллера устройства измерения скорости;

- в качестве вспомогательного коммуникационного ядра устройства автоматизированной калибровки ГУН;
- в качестве конфигурационного ядра, управляющего ЦОС-трактом системы ближней радиолокации.

Каждая из этих работ внесла существенный вклад в развитие прототипа.

В процессе работы над контроллером устройства измерения скорости были отработаны основные принципы и механизмы совместной работы группы разработчиков над проектом, базирующимся на прототипе платформы. Были отлажены механизмы автоматизированной сборки, выработаны принципы работы со внешними репозиториями программного и аппаратного обеспечения, принципы ведения контроля версий. Также в ходе этой работы в проект были включены несколько уникальных аппаратных модулей, что позволило оценить возможности прототипа платформы по расширению пользовательскими СФ-блоками.

Кроме работы с самой платформой на этом этапе были выявлены и устранены проблемы с контроллером прерываний микропроцессорного ядра uRV и расширен системный таймер.

При применении в качестве вспомогательного коммуникационного ядра в устройстве автоматизированной калибровки ГУН были отработаны принципы портируемости, воспроизводимости результатов. В этом проекте хорошо проявила себя система

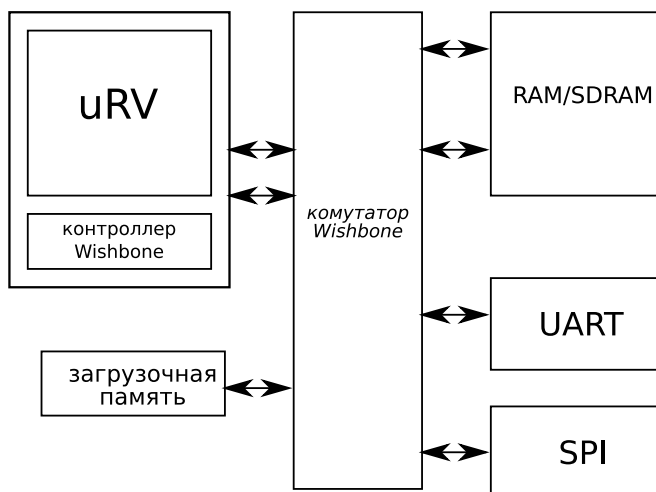


Рис. 2. Развитие структуры аппаратной части платформы



автоматизированной сборки. Используя эту систему с минимальными правками, удалось реализовать проект на другой ПЛИС семейства Xilinx Artix.

В третьем проекте были проверены наработки, полученные в ходе выполнения первых двух проектов.

ДАЛЬНЕЙШАЯ РАБОТА НАД ПРОТОТИПОМ

Для того чтобы продолжить формировать платформу, необходимо доработать прототип в аппаратной и программной частях.

В первую очередь, для ядра uRV необходимо реализовать возможность загрузки инструкций по системной шине и включить в систему контроллер SDRAM. Эти доработки являются критическими с точки зрения запуска на платформе сложного ПО.

Также требуется провести работу по переработке реализации верхнего файла иерархии. В данный момент верхний уровень реализован на языке VHDL с использованием библиотек. Наш опыт подсказывает нам, что проектирование на VHDL для конечного пользователя затруднительно. Чтобы упростить процесс интеграции пользовательских СФ-блоков, было принято решение перевести верхний уровень иерархии на SystemVerilog (см. рис. 2).

Отдельной задачей является поиск ядра микроконтроллерного класса с архитектурой RISC-V с поддержкой аппаратной отладки с помощью интерфейса JTAG. Возможность отладки ПО, запущенного на вычислительном ядре, — одно из важнейших условий развития платформы. Рассматривается как возможность доработки ядра uRV, так и переход на другое вычислительное ядро, например, RISKY [6] или SCR1 [7].

Все это в комплексе позволит существенно развить программную часть платформы.

Разработку программной части планируется разбить на несколько этапов:

- запуск операционной системы FreeRTOS [8];

- запуск загрузчика barebox [9];

- составление программной модели расширяемой платформы. Запуск FreeRTOS продемонстрирует работоспособность контроллера прерываний и системного таймера.

Запуск загрузчика barebox является подготовительным этапом для запуска ОС с ядром Linux. На этом этапе можно обеспечить платформу набором стандартных драйверов и продемонстрировать работоспособность блоков периферии.

ЛИТЕРАТУРА

1. PULP Parallel Ultra Low Power [Электронный ресурс] // URL: <http://www.pulp-platform.org/> (дата обращения 20.12.2017).
2. SiFive, Inc. [Электронный ресурс] // URL: <https://www.sifive.com/> (дата обращения 20.12.2017).
3. uRV RISC-V core [Электронный ресурс] // URL: <https://github.com/twlostow/urv-core> (дата обращения 20.12.2017).
4. FuseSoC is a Package Manager and a Set of Build Tools for FPGA/ASIC Development [Электронный ресурс] // URL: <https://github.com/olofk/fusesoc> (дата обращения 20.12.2017).
5. QEMU the FAST! Processor Emulator [Электронный ресурс] // URL: <https://www.qemu.org/> (дата обращения 20.12.2017).
6. A Tiny Virtual Processor with a RISC-inspired Instruction Set [Электронный ресурс] // URL: <https://github.com/saxbophone/risky> (дата обращения 20.12.2017).
7. SCR1 is an Open-source RISC-V Compatible MCU Core [Электронный ресурс] // URL: <https://github.com/syntacore/scr1> (дата обращения 20.12.2017).
8. The FreeRTOS™ Kernel [Электронный ресурс] // URL: <https://www.freertos.org> (дата обращения 20.12.2017).
9. barebox is a Bootloader Designed for Embedded Systems [Электронный ресурс] // URL: <http://www.barebox.org> (дата обращения 20.12.2017).

КНИГИ ИЗДАТЕЛЬСТВА "ТЕХНОСФЕРА"



Цена 1090 руб.

ВНЕДРЕНИЕ И РАЗВИТИЕ ИНДУСТРИИ 4.0. ОСНОВЫ, МОДЕЛИРОВАНИЕ И ПРИМЕРЫ ИЗ ПРАКТИКИ

Под ред. Армина Рота

Издание осуществлено при финансовой поддержке ПАО «Ростелеком»

Перевод с немецкого под общ. ред. А.В. Кострова

В книге представлены концепция, основные парадигмы развития, структура технологии «Индустрия 4.0». В отличие от широко распространенного технократического прикладного метода изучения издание позволяет выделить и целостно описать уровни стратегического, тактического и оперативного управления. Ключевым элементом при этом является процессная модель, описывающая действия на стратегическом и оперативном уровнях, а примеры практического применения «Индустрии 4.0» в различных отраслях промышленности наряду с мнениями известных экспертов в области науки и производства вызовут интерес не только у новичков отрасли, но и у научных сотрудников, инженерно-технических работников и руководителей высшего и среднего звена.

Благодаря обширному глоссарию издание станет ценным справочным пособием по использованию основных положений и лучших практик «Индустрии 4.0».

М: ТЕХНОСФЕРА, 2017. — 294 с.
ISBN 978-5-94836-482-7

КАК ЗАКАЗАТЬ НАШИ КНИГИ?

✉ 125319, Москва, а/я 91; ☎ +7 (495) 234-0110; ☎ +7 (495) 956-3346; knigi@technosphera.ru, sales@technosphera.ru