**УДК 004.231.3**

**Особенности современных подходов к верификации RISC-V ядер**

**Modern RISC-V verification approaches**

**Чусов С. А.1**

*1 Научно-исследовательская лаборатория энергоэффективных Систем на Кристалле*

*124498, НИУ МИЭТ, г. Москва, г. Зеленоград, пл. Шокина, д.1. chusserge@mail.ru*

**Аннотация**

Определены потенциальные сложности, с которыми может столкнуться инженер в процессе верификации процессорных ядер, в том числе на архитектуре RISC-V. Проанализированы современные подходы к верификации RISC-V ядер. Выделены их особенности, определена применимость для различных этапов проектирования.

**Ключевые слова:** функциональная верификация; симуляция; формальная верификация; RISC-V; процессор; обзор; анализ.

**Abstract**

Potential problems that an engineer may encounter during CPU verification, including RISC-V specific problems architecture, have been presented. Modern approaches to the RISC-V cores verification have been analyzed. Their features and applicability for various development stages have been stated.

**Keywords:** functional verification; simulation; formal verification; RISC-V; CPU; review; analysis.

В настоящее время популярность набирает архитектура набора команд (АНК) RISC-V [1]. В 2015 году был создан международный фонд архитектуры. Уже с 2018 года он работает в партнёрстве с The Linux Foundation. Основная отличительная черта архитектуры RISC-V – ее открытость. Так как АНК может быть использована бесплатно, то уже сейчас вокруг RISC-V сформировалось сообщество из десятков тысяч разработчиков, готовых к сотрудничеству и обмену опытом. Еще одной особенностью RISC-V является модульность. Базовый набор содержит 49 инструкций, при этом в спецификациях присутствует несколько десятков расширений для аппаратной поддержки дополнительного функционала. Также инженерам предоставлена возможность создания нестандартных команд и их интеграция в архитектуру.

Основным цифровым блоком систем на кристалле в большинстве случаев является процессор, выполняющий программную обработку цифровых данных. Сфера верификации процессорных ядер является одной из самых актуальных составляющих индустрии проектирования и верификации сложно-функциональных блоков. В ходе работы были определены сложности, с которыми может столкнуться инженер в ходе функциональной верификации процессорных ядер. Неполный список включает себя проверку всех реализуемых ядром инструкций, проверку их сочетаний, приводящих к микроархитектурным конфликтам, проверку уровня доступа к архитектурным регистрам, проверку корректности обработки синхронных и асинхронных событий. В том числе было разобрано влияние модульности RISC-V на процесс функциональной верификации ядер на основе этой АНК.

Был проведен анализ современных подходов к функциональной верификации RISC-V ядер, где каждый подход реализует один из типов верификации: формальный или же на основе симуляции. Были определены области применимости каждого из типов, а также конкретных подходов каждого типа. Так, к примеру, формальная верификация не может быть использована для проверки асинхронных событий, а тестирование при помощи симуляции с самопроверкой не является достаточным, однако его использование оправдано на начальных этапах проектирования. В ходе работы были подробно рассмотрены несколько уровней функциональной верификации на основе симуляции, отличающихся сложностью и охватом проверяемого функционала. Сюда, прежде всего, вошли тестирование с самопроверкой и различные подходы, использующие сравнение с эталонной моделью.

В ходе работы проанализированы современные интерфейсы для верификации RISC-V ядер, такие как RVFI [2] и RVVI [3] и определены причины, по которым возникла необходимость в их создании и использовании. Также были определены преимущества данных интерфейсов и представлены примеры их использования в различных подходах к верификации.

В заключении автором была дана субъективная оценка текущего состояния области верификации RISC-V ядер и вектора ее развития.

***Литература***

1. RISC-V: Open standard Instruction Set Architecture (ISA) [Электронный ресурс] // URL: https://riscv.org/about (дата обращения: 10.09.23)

2. Интерфейс RVFI [Электронный ресурс] // URL: https://github.com/SymbioticEDA/riscv-formal/blob/master/docs/rvfi.md (дата обращения: 10.09.23)

3. Интерфейс RVVI [Электронный ресурс] // URL: https://github.com/SymbioticEDA/riscv-formal/blob/master/docs/rvfi.md (дата обращения: 10.09.23)

Nowadays, the RISC-V [1] instruction set architecture (ISA) is gaining popularity. The RISC-V International Foundation was created in 2015. It has been working in partnership with The Linux Foundation since 2018. RISC-V is an open architecture. The tens of thousands community has already formed, ready to collaborate and share experience. Another feature of RISC-V is extensibility. The basic set contains 49 instructions, while the specifications contain several dozen extensions for support of the additional functionality. Engineers are also given the opportunity to create custom instructions and integrate them.

The main part of systems on a chip in most cases is a processor unit that performs software processing of digital data. The verification of processor cores is one of the most relevant components of the digital design industry. Difficulties that an engineer may encounter during functional verification of processor cores were presented in this paper. An incomplete list includes checking all instructions implementation, their sequences that lead to microarchitectural conflicts, checking the level of access to architectural registers, and checking the correct processing of synchronous and asynchronous events. In particular, the influence of extensibility on the process of RISC-V functional verification was analyzed.

An analysis of modern approaches to the RISC-V functional verification was presented. Each approach implements one type of verification: formal or simulation-based. The areas of applicability of each type, as well as specific approaches of each type, were discussed. For example, formal verification cannot be used to verify asynchronous events, and self-testing simulation testing is not sufficient, but can be used in the early stages of development. In this paper, several levels of functional verification based on simulation were examined in detail, differing in complexity and coverage of the functionality, including self-testing and various approaches using comparison with a reference model.

In this paper, modern interfaces for verifying RISC-V cores, such as RVFI [2] and RVVI [3], were analyzed and the reasons of their creation were presented. The advantages of these interfaces were discussed and examples of their use in various verification approaches were presented.

In conclusion, subjective assessment of the current state and development vector of RISC-V verification field was given be the autor.

***References***

1. RISC-V: Open standard Instruction Set Architecture (ISA) at https://riscv.org/about (accessed 10 September 2023)

2. RVFI Interface at https://github.com/SymbioticEDA/riscv-formal/blob/master/docs/rvfi.md (accessed 10 September 2023)

3. RVVI Interface at https://github.com/SymbioticEDA/riscv-formal/blob/master/docs/rvfi.md (accessed 10 September 2023)