

Верификация устройства доступа в оперативную память процессора «Эльбрус-8С2»Н.А. Рослов^{1,2}¹Московский физико-технический институт (государственный университет)²АО «МЦСТ»

Микропроцессоры семейства «Эльбрус» отличаются сложной многоуровневой структурой подсистемы памяти, позволяющей собирать многопроцессорные комплексы и призванной обеспечить максимальную загрузку имеющихся ресурсов.

Устройство доступа в память (Memory Access Unit, MAU) является внешним блоком каждого из восьми ядер микропроцессора «Эльбрус-8С2» и обеспечивает взаимодействие ядра с общим L3 кэшем. Каждое процессорное ядро формирует запросы по чтению данных из памяти и запросы по записи данных в память. После прохождения запросов через L2 кэш они попадают в MAU, обрабатываются и выдаются далее в общий L3 кэш. После прихода из L3 кэша ответа на запрос по чтению в MAU формируется ответ в L2 кэш [1]. Сложность организации подсистемы памяти микропроцессоров с архитектурой «Эльбрус» нередко приводит к ошибкам в ней, поэтому необходима тщательная верификация всех компонентов подсистемы. Одним из подходов к проверке функционирования блоков, входящих в состав микропроцессора, является их автономная верификация, позволяющая проверять отдельно выделенный блок, начиная с самых ранних стадий маршрута проектирования [2].

Для автономной верификации MAU была реализована тестовая система, состоящая из следующих компонентов:

1. Тестовое окружение, выполняющее функции генерации тестовых воздействий и эмуляции работы L2 и L3 кэшей, с которыми взаимодействует MAU. Тестовое окружение реализовано на языке SystemVerilog с использованием универсальной методологии верификации UVM. Работа с данными в тестовом окружении выполняется на уровне транзакций.
2. Эталонная модель MAU, моделирующая поведение устройства на уровне транзакций. Отличительной особенностью модели является реализация с помощью высокоуровневых структур данных на языке C++, что позволило упростить процесс ее создания, а также избежать дублирования ошибок с низкоуровневым RTL-описанием устройства.
3. Интерфейс между тестовым окружением и эталонной моделью, выполняющий передачу и преобразование пакетов данных (транзакций) — адаптер модели [3].

В тестовое окружение также входят тестовые сценарии, по которым оно проверяет функционирование MAU в различных режимах, генерируя по заданным алгоритмам тестовые воздействия. При появлении сигналов на интерфейсе между тестируемым устройством и тестовым окружением (тестового воздействия или реакции устройства) они преобразуются в транзакции и передаются в эталонную модель через адаптер модели. В каждом такте работы тестового окружения вызывается функция модели, эмулирующая работу внутренних компонентов MAU и формирующая в итоге эталонные реакции. В конце каждого такта в случае поступления пакета данных от тестового окружения эти данные записываются во внутренние структуры модели для дальнейшего использования в ее работе, либо сравниваются с реакциями устройства, пришедшими в течении этого такта. Сравнение транзакций производится на основе простого равенства их транзакций, но в части случаев применяются более сложные алгоритмы, учитывающие значимости отдельных полей или режимы работы устройства. На основании результатов сравнения выносится решение о корректности функционирования RTL-описания модели.

В ходе верификации с использованием описанной выше тестовой системы был обнаружен и исправлен ряд ошибок в RTL-описании устройства и его документации.

Литература

1. *Kostenko V. O., Kozhin A.S., Polyakov N.Y., Slesarev M.V., Tikhorskiy V.V., Sakhin Y.K.* Elbrus-8C: The Latest Yield from MCST and MIPT Collaboration // Proc. Of International Conference of Engineering and Telecommunication (EnT), 2015. P. 67-68.
2. *М.В.Петроченков, И.А.Стотланд.* Методика автономной верификации устройств подсистемы памяти многоядерных микропроцессоров. // Вопросы радиоэлектроники. 2016. Сер.ЭВТ. Вып.3. С.42-47.

3. *И.А.Стотланд, А.А.Лагутин.* Применение эталонных событийных моделей для автономной верификации модулей микропроцессоров. // Вопросы радиоэлектроники. 2014. Сер.ЭВТ. Вып.3. С.17-27.