

К.т.н. И.А. Стотланд, В.Н. Куцевол (ЗАО «МЦСТ»),

к.т.н. А.Н. Мешков (ЗАО «МЦСТ», ПАО «ИНЭУМ им. И.С. Брука»)

I. Stotland, V. Kutsevol, A. Meshkov

ПРОБЛЕМЫ ФУНКЦИОНАЛЬНОЙ ВЕРИФИКАЦИИ КЭШ-ПАМЯТИ ВТОРОГО УРОВНЯ МИКРОПРОЦЕССОРОВ С АРХИТЕКТУРОЙ «ЭЛЬБРУС»

PROBLEMS OF FUNCTIONAL VERIFICATION OF ELBRUS MICROPROCESSOR L2-CACHE

Рассмотрены основные проблемы функциональной верификации кэш-памяти второго уровня, и выделены преимущества применения их автономной верификации. Предложен подход к функциональной верификации кэш-памяти второго уровня микропроцессоров с архитектурой «Эльбрус», основанный на адаптации программной модели подсистемы памяти. Приведены структура тестовой системы и принципы организации взаимодействия компонентов тестовой системы и модели подсистемы памяти. Описана разработанная модель подсистемы памяти микропроцессоров «Эльбрус». Рассмотрены варианты применения предложенного подхода.

The main methods of functional verification of L2-cache are considered in this paper. The benefits of using stand-alone verification of L2-cache are marked out. The novel approach to functional verification of L2-cache of Elbrus architecture microprocessors is proposed. The memory subsystem model of Elbrus microprocessor and the testbench for L2-cache verification are described. The methods of communication of testbenches and memory subsystem models are presented.

Ключевые слова: кэш-память второго уровня, автономная верификация, модель подсистемы памяти, протокол когерентности, архитектура «Эльбрус».

Keywords: L2-cache, stand-alone verification, memory subsystem model, cache coherence protocol, microprocessor «Elbrus».

Введение

Наращивание производительности микропроцессорных систем приводит к усложнению процессорных ядер и увеличению их числа на кристалле. В результате наиболее критичным местом становится подсистема памяти микропроцессора, обеспечивающая доступ процессорных ядер к данным и поддерживающая когерентность системы.

Современные микропроцессоры с архитектурой «Эльбрус» имеют многоядерную структуру и сложную иерархическую подсистему памяти, одним из наиболее сложных компонентов которой является кэш-память второго уровня (L2-кэш). L2-кэш обслуживает заявки от кэш-памяти первого уровня (L1-кэша), устройства трансляции адресов (Translation Lookaside Unit, TLU), устройства асинхронной предподкачки массивов (Array Access Unit, AAU), а также запросы на подкачку кода от буфера команд (Instruction Buffer, IB) [1].

В общей схеме обеспечения доступа к данным и поддержке их согласованности L2-кэш занимает одну из ключевых позиций, и к корректности его функционирования предъявляются высокие требования. На каждом этапе проектирования микропроцессоров и их компонентов применяют различные методы и средства функциональной верификации. Применительно к кэш-памяти широко распространена проверка их работы в составе микропроцессора – системная верификация. Однако большое число абонентов L2-кэша, прозрачность с программной точки зрения и высокая комбинаторика внутренних состояний существенно усложняют создание направленных тестовых сценариев для достижения критических ситуаций его работы. Кроме того, системная верификация может проводиться только при наличии всех компонентов микропроцессора и требует значительных вычислительных и временных ресурсов для моделирования. В связи с этим целесообразно применять для проверки L2-кэша также методы автономной (модульной) верификации.

В статье предложен подход к автономной верификации кэш-памяти второго уровня, входящей в состав современных многоядерных микропроцессоров с архитектурой «Эль-

брус».

1. Обзор существующих методик

Объектом функциональной верификации являются логические модели уровня регистровых передач (RTL-модели), написанные на одном из применяемых при проектировании языков описания аппаратуры (HDL-языки). Варианты функциональной верификации можно разделить на два основных класса: формальную верификацию и динамическую верификацию, основанную на имитационном тестировании.

Формальные методы предполагают исследование формальной модели аппаратуры в статическом состоянии. Центральной проблемой при этом является возможный «комбинаторный взрыв» в пространстве состояний при увеличении сложности модели, что существенно ограничивает применение формальных методов к RTL-моделям сложных промышленных разработок. В [2, 3] предлагаются подходы к решению подобных проблем, однако в этих и тематически близких работах по формальным методам, как правило, не рассматривается реализация кэш-памяти, а основное внимание уделяется спецификации поддерживаемого ей протокола когерентности.

Методы динамической верификации дают возможность проверить не только спецификацию, но и конкретную реализацию устройства путем анализа его RTL-модели. В случае сложного промышленного устройства обычно применяют динамическую верификацию на основе тестовой системы, имитирующей окружение, в котором ему предстоит функционировать.

Одним из принятых подходов к проверке реакций исследуемой модели является применение ее готового или одновременно разрабатываемого эталона для сравнения результатов тестирования. В зависимости от сложности устройства и полноты спецификации используют эталонные модели разного уровня абстракции: от потактово-точных до событийных уровня транзакций (в [4–6] рассмотрены вопросы построения таких моделей и тестовых систем на их основе). В то же время, ввиду того что L2-кэш представляет со-

бой устройство с большой неопределенностью переходов между внутренними состояниями, реализация для него подобного подхода может повлечь за собой длительный и трудоёмкий процесс.

Для решения проблемы при верификации микропроцессора «Эльбрус-4С» использовался метод декомпозиции: L2-кэш разбивался на более простые подмодули, для которых были реализованы эталонные программные модели с потактовой точностью [7] и тестовые системы. Это позволило обнаруживать ошибки в подмодулях, но не дало возможность проверить работу L2-кэша в целом. Возможный вариант такой проверки рассмотрен в [8], где автономная верификация L2-кэша выполняется на основе модели уровня транзакций и тестовой системы, написанных на языке SystemC. Однако, ввиду того что SystemC не входит в общий маршрут проектирования микропроцессоров с архитектурой «Эльбрус», потребовалось разработать иной подход. Его суть заключается в том, чтобы в автономном режиме верифицировать реализацию кэш-памяти второго уровня без эталонных моделей с применением тестовой системы, которая позволила бы определить реакцию RTL-модели L2-кэша на весь спектр воздействий, возможных в рамках данной архитектуры.

2. Структура тестовой системы

Внешние интерфейсы L2-кэша можно разделить на три основные группы: интерфейсы с устройствами home-ядра, к которому принадлежит L2-кэш (L1-кэш, регистровый файл, IB, TLU, AAU); интерфейсы с подсистемой памяти многоядерной системы; дополнительные интерфейсы диагностики и обратные связи. В свою очередь, входные воздействия на L2-кэш можно разделить на первичные запросы от home-ядра системы и когерентные запросы/ответы от модуля доступа к памяти (Memory Access Unit, MAU), обеспечивающего взаимодействие ядра с общей для всех ядер подсистемой памяти. В связи с этим целесообразно разделить генератор воздействий на два независимых модуля: глобальный генератор первичных запросов и имитатор общей подсистемы памяти.

Глобальный генератор первичных запросов (Global Generator) вырабатывает высокоуровневые воздействия от home-ядра – обращения по записи (store) и чтению (load) – сериализует их в агентах устройств home-ядра и подает на верифицируемую RTL-модель в виде последовательности интерфейсных сигналов. При этом можно следовать как направленным, так и псевдослучайным тестовым сценариям.

Для проверки реализации протокола когерентности тестовая система должна имитировать работу не только окружения L2-кэша, но и всей подсистемы памяти многоядерного микропроцессора, а также моделировать работу памяти нижнего уровня. С этой целью была сформирована структура, представленная на рис. 1. В ее составе используется программная модель (Memory Unit), включающая ранее разработанную в рамках эльбрусовских проектов программную модель подсистемы памяти и подключаемая через адаптер модели (Memory Unit Adapter). Основными функциями адаптера являются десериализация запросов в память, выдаваемых RTL-моделью L2-кэша, и преобразование форматов запросов от L2-кэша в формат пакетов, принятый в модели. Кроме того, он выполняет «подклеивание» заявок на обращение в память, находящихся в очереди на выдачу L2-кэша.

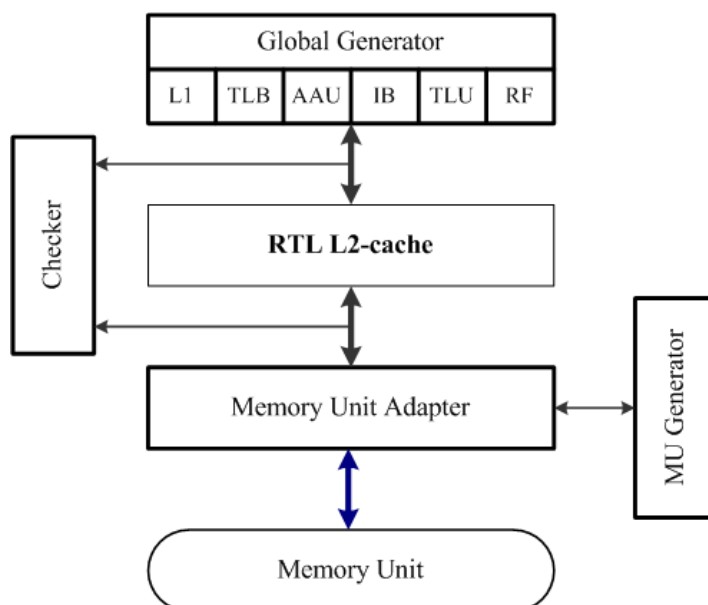


Рис. 1. Структура тестовой системы для верификации кэш-памяти второго уровня

Чтобы сформировать близкий к реальному модельный трафик системы по поддерж-

ке протокола когерентности, был разработан генератор запросов к памяти от других ядер (MU generator), который может выдавать псевдослучайные и направленные воздействия. Для синхронизации работы с глобальным генератором первичных запросов он интегрирован в тестовую систему и имеет интерфейс с адаптером модели, который «на лету» передаёт запросы в модель.

Система не только верифицирует реализацию протокола когерентности в L2-кэше, но и включает набор встроенных средств (Checker) для проверки правильности формирования запросов и ответов в MAU, выдачи ответов в RF и устройства home-ядра.

3. Модель подсистемы памяти

В модель подсистемы памяти (Memory Subsystem Model), структура которой изображена на рис. 2, включены компоненты, соответствующие основным ее устройствам: модули оперативной памяти (Memory), системный коммутатор (System Commutator), кэши L2 и L3, устройство доступа в память (MAU). Кроме того, в состав модели входит генератор запросов (Request Generator), передающий запросы в L2-кэш со стороны процессорных ядер [9]. Модель соединяется с тестовой системой посредством модуля L2Wrapper, встраиваемого вместо одного из L2-кэшей.

Структура связующего компонента L2Wrapper приведена на рис. 3. Объект Converter в нем осуществляет преобразования между транзакциями L2-кэша и пакетными интерфейсами тестовой системы. Чтобы синхронизировать тестовую систему и функциональную модель подсистемы памяти, в L2Wrapper введены выходные очереди (Output Queue). Интерфейс динамического формирования исходных запросов (Initial Requests) позволяет создать трафик запросов от модуля Request Generator в тестируемый L2-кэш, который имитирует его работу в составе многопроцессорной системы. Кроме того, модель предоставляет тестовой системе интерфейсы для управления и контроля (Management). В них входит метод clk, инициирующий модельный псевдотакт.

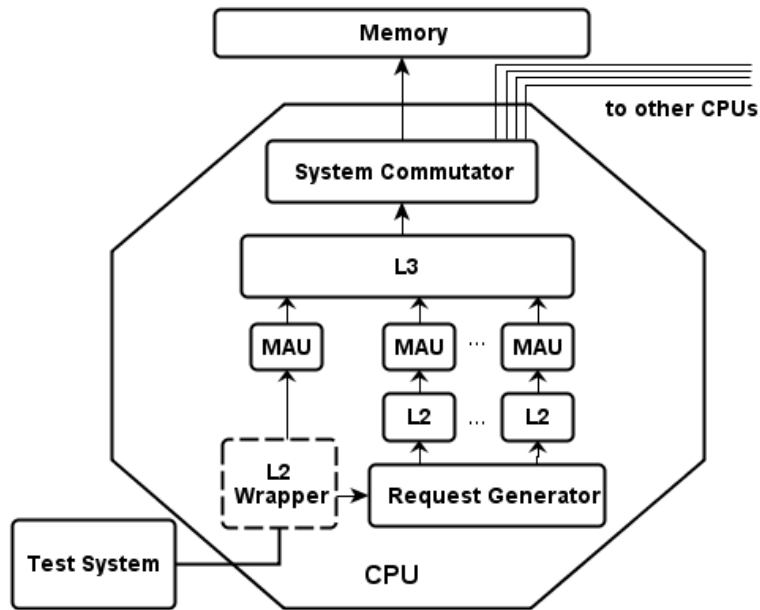


Рис. 2. Архитектура модели подсистемы памяти (Memory Subsystem Model)

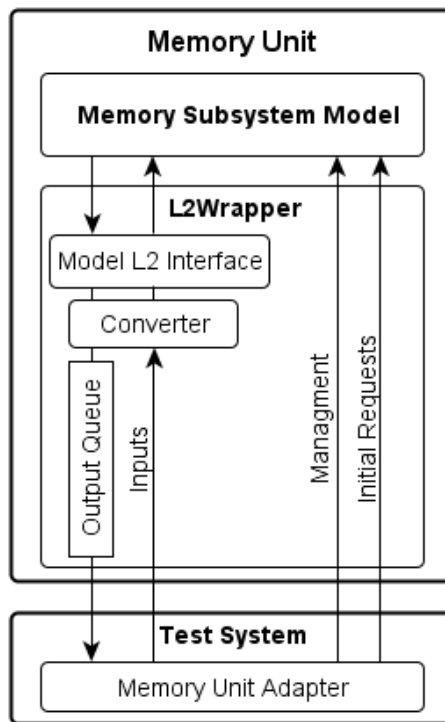


Рис. 3. Организация взаимодействия модели подсистемы памяти и тестовой системы

Каждое устройство в составе модели обладает набором состояний, переход между которыми осуществляется при приеме и передаче сообщений в рамках работы протокола когерентности. Недопустимые переходы приводят к ошибке. Это свойство используется

для проверки корректности ответов от тестируемого устройства. Кроме того, на основе поступивших в модель исходных запросов в оперативную память создается множество эталонных значений возможных состояний оперативной памяти. С его помощью производится контроль обобщенного состояния памяти (состояние оперативной памяти и всех кэшей процессоров), позволяющий выявлять повреждение данных.

Заключение

При постановке данного проекта была предпринята попытка создать систему тестирования, лишенную двух концептуальных недостатков: излишнего дробления спроектированного устройства на более мелкие составляющие и необходимости формирования сложного и трудозатратного потактово-точного эталона всего кэша. Вместо этого путем внедрения уже нашедшей применение модели подсистемы памяти в нее было заложено знание о взаимодействующих с кэшем объектах. Это позволило сконцентрировать усилия на расширении набора уже встроенных в эту подсистему проверок и генерируемых воздействий.

Вместе с тем, созданную тестовую систему удалось сделать достаточно гибкой – предусмотрена возможность задавать как псевдослучайные, так и направленные тестовые последовательности для разных реализаций L2-кэшей.

Практическим результатом работы стала адаптация системы применительно к ряду L2-кэшей в составе микропроцессоров, ранее прошедших верификацию, и создание регрессионных тестов, а также подготовка к верификации нового поколения микропроцессора семейства «Эльбрус». Кроме того, предложенный подход может быть применен при верификации кэш-памяти третьего уровня и других компонентов подсистемы памяти микропроцессора.

Литература

1. Знаменский Д.В. Проблема управления выходным потоком данных кэш-памяти

системы на кристалле «Эльбрус-2S». – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3, с. 48-62.

2. Chen Xi. Verification of hierarchical cache coherence protocols for futuristic processors. PhD thesis, The University of Utah, 2008.

3. Xin Lai, Cong Liu, Zhiying Wang. Applying formal verification to a cache coherence protocol in TLS. // Proc. Of Computer Modeling and Simulation (EMS), nov. 2011. P. 329-334.

4. Chupilko M., Kamkin A. Runtime verification based on executable models: on-the-fly matching of timed traces. // Proc. Of Model-base testing Workshop (MBT-2013), 2013. P. 67-81.

5. Мешков А.Н., Рыжов М.П., Шмелёв В.А. Развитие средств верификации микропроцессора «Эльбрус-2S». – «Вопросы радиоэлектроники», сер. ЭВТ, 2014, вып. 3, с. 5-17.

6. Стотланд И.А. Метод динамической верификации модулей системного обмена микропроцессорных вычислительных комплексов. – Научно-технический вестник Поволжья, 2012, № 4, с. 191-196.

7. Шмелёв В.А., Стотланд И.А. Автономная верификация микропроцессоров на основе эталонных моделей разного уровня абстракции: Сб. научн. тр. Всероссийской научно-технической конференции «Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)», 2012, №1, с. 435-440.

8. Designing, verifying and building an advanced L2 cache sub-system using SystemC / Tessier T., Lin H., Ringoen D., Hickey E., Anderson S. // Proc. Of Design and Verification Conference (DV-CON), feb. 2012. P. 1-8.

9. Куцевол В.Н., Мешков А.Н., Петроченков М.В. Методология верификации протокола когерентности микропроцессора «Эльбрус-2S». – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3, с. 107-117.