электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.052.4

Инструмент верификации протокола когерентности памяти

Буренков В.С.

Аспирант, кафедра «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана, г. Москва, Россия

Научный руководитель: Иванов С.Р.,к.т.н., доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана, г. Москва, Россия

МГТУ им. Н.Э. Баумана

VanSBuren@mail.ru

Широко применяемые методы поиска ошибок в устройствах, реализующих протоколы когерентности памяти, основанные на моделировании со случайными воздействиями, не обеспечивают 100%-полноты верификации. Некоторые ошибки могут проявиться лишь при возникновении длинных последовательностей событий, таких как кэш-промахи и прием сообщений различными частями системы. Поскольку число таких последовательностей является комбинаторным, вероятность их возникновения во время моделирования со случайными воздействиями резко уменьшается при увеличении их длины [1].

На проведение верификации, в ходе которой анализируются все достижимые состояния верифицируемой системы, нацелены формальные методы.

В ходе формальной верификации протокола проверяется соответствие абстрактной модели протокола его спецификации, то есть набору свойств, которым должен отвечать протокол.

Естественной моделью протоколов когерентности является конечный автомат.

Для верификации протоколов когерентности памяти может быть применен формальный метод model checking [2]. Модель верифицируемой системы, соответствующая данному методу (структура Крипке), согласуется с естественной моделью протокола когерентности. Данный метод может быть полностью автоматизирован, его использование предполагает удобный способ выражения

требований к протоколам когерентности на языке формальной логики. Помимо этого, существует множество алгоритмов, позволяющих бороться с проблемой «взрыва числа состояний» при исследовании модели методом model checking.

Для решения поставленной задачи верификации протокола когерентности «Эльбрус-2S», был разработан инструмент, в основе которого находится пакет Spin (Simple Promela Interpreter) [3], в котором реализованы алгоритмы model checking. Основная цель использования пакета Spin (проверка корректности взаимодействующих параллельных асинхронных процессов) сочетается с верификацией протоколов когерентности памяти, поскольку процессоры системы, протокол когерентности которой проверяется, работают параллельно и асинхронно по отношению друг к другу.

На рисунке 1 представлена диаграмма компонентов разработанного инструмента, показывающая, из каких частей состоит инструмент верификации протокола когерентности памяти, а также как эти части зависят друг от друга. Интерфейсы показывают, какая информация предоставляется одними компонентами и используется другими.

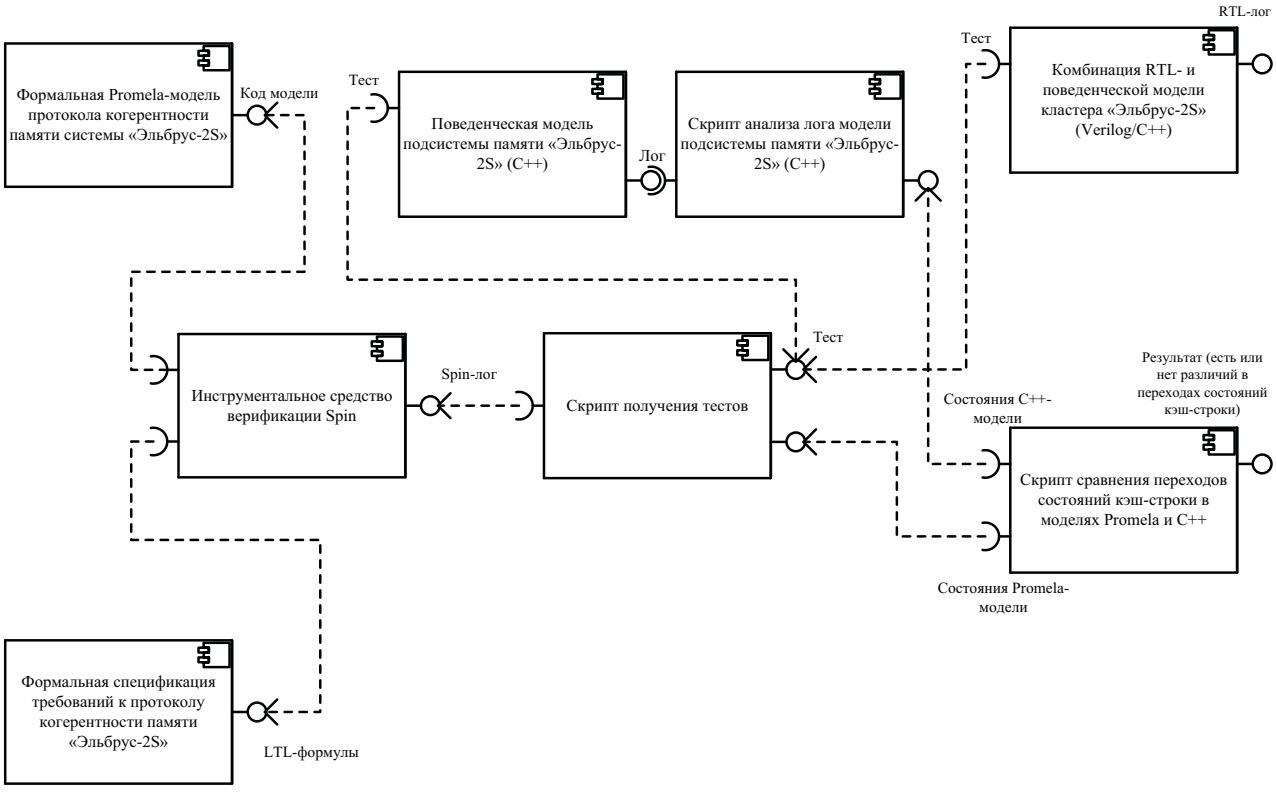

Рис. 1. Диаграмма компонентов инструмента верификации протокола когерентности памяти

Автор описал на входном языке Promela пакета Spin две модели протокола когерентности, используемого в системе на кристалле «Эльбрус-2S». В моделях отражены

N процессоров (которые считаются одноядерными), где значение *N* при исследованиях было равным 3 и 4. Процессоры (и значимые функции различных устройств, участвующих в реализации протокола когерентности) описываются идентичными процессами. В моделях представлена одна кэш-строка, поскольку протокол когерентности памяти отвечает за состояния одной строки памяти и не затрагивает взаимодействие между доступами к различным кэш-блокам. Таким образом, все процессоры обращаются в память только по одному адресу. Разработанные модели отличаются друг от друга полнотой описания протокола, а, значит, и сложностью. В первой модели вводится ограничение – пока не завершится операция от одного процессора, другие процессоры не могут выдавать новые инструкции. Во второй модели такое ограничение снято, что соответствует модели памяти «Эльбрус».

К протоколу были предъявлены следующие требования:

- кэш-строка никогда не может находиться в двух или более кэшах в состоянии Modified;
- кэш-строка никогда не может находиться в двух или более кэшах в состоянии Owned;
- никогда не может быть ситуации, в которой в одном из кэшей строка находится в состоянии Modified, а в каком-либо другом – в состоянии Shared или Owned.

Формальная проверка этих свойств не обнаружила контрпримеров.

Результаты верификации показаны в таблицах 1-3. Верификация осуществлялась на сервере Intel Xeon E5520 с тактовой частотой 2,27 ГГц и объемом оперативной памяти 64 Гб.

Таблица 1

Результаты верификации модели 1 при $N = 3$

Оптимизация Spin	Количество исследованных состояний модели	Объем используемой памяти	Время верификации
Отсутствует	10^6	100 Мб	3 с
collapse	10^6	60 Мб	6 с
минимальный автомат	10^6	22 Мб	22 с
hash compact	10^6	47 Мб	3 с
bitstate	10^6	12 Мб	3 с

Таблица 2

Результаты верификации модели 1 при $N = 4$

Оптимизация Spin	Количество исследованных состояний модели	Объем используемой памяти	Время верификации

Отсутствует	$3 \cdot 10^8$	42 Гб	18 мин
collapse	$3 \cdot 10^8$	20 Гб	30 мин
минимальный автомат	$3 \cdot 10^8$	1 Гб	110 мин
hash compact	$3 \cdot 10^8$	17 Гб	13 мин
bitstate	$\sim 10^8$	407 Мб	4 мин

Таблица 3

Результаты верификации модели 2 при $N = 3$

Оптимизация Spin	Количество исследованных состояний модели	Объем используемой памяти	Время верификации
Отсутствует	$2 \cdot 10^7$	2800 Мб	1 мин
collapse	$2 \cdot 10^7$	1500 Мб	2 мин
минимальный автомат	$2 \cdot 10^7$	240 Мб	7 мин
hash compact	$2 \cdot 10^7$	1200 Мб	50 с
bitstate	$\sim 10^7$	120 Мб	11 с

Для второго варианта модели четырехпроцессорной системы удалось получить только результат с использованием bitstate-хэширования, при котором исследованным (за несколько часов) оказалось множество состояний с мощностью порядка 10^9.

Таким образом, инструментальное средство Spin может применяться для верификации промышленных протоколов когерентности памяти, однако при этом существует ограничение на число узлов процессорной системы, отражаемых в модели, равное 3 или 4 в зависимости от сложности модели. В [4] данный параметр ограничивается аналогичными значениями (при этом речь идет не только о Spin, а о методе model checking вообще).

Это значение, видимо, является пределом применимости метода model checking. Промышленность же обусловливает необходимость верификации для N = 8, 16 и так далее. Для верификации более сложных систем возможно использование других приемов (например, абстракции) совместно с model checking, что является предметом дальнейших исследований.

Список литературы

1. K.L. McMillan. Symbolic Model Checking: An Approach to the State Explosion Problem, Ph.D. Dissertation. // Carnegie Mellon University, 1992.
2. E.M. Clarke, O. Grumberg, D. Peled. Model Checking. // MIT Press, 1999 – 314 pp.
3. G.J. Holzmann. The Spin Model Checker: Primer and Reference Manual. – Addison-Wesley, 2003 – 608 pp.
4. C. Chou, P. K. Mannava, S. Park. A Simple Method for Parameterized Verification of Cache

Coherence Protocols. // Formal Methods in Computer-Aided Design. Lecture Notes in Computer Science – 2004 – Vol. 3312, pp. 382-398.