

О генерации системных тестов для подсистемы памяти микропроцессора*П.В. Фролов^{1,2}*¹ ЗАО «МЦСТ»² Институт электронных управляющих машин им. И.С. Брука

В настоящее время в процессе системной верификации микропроцессоров, разрабатываемых в ЗАО «МЦСТ», основную роль играют генераторы псевдослучайных ассемблерных тестов, что в принципе соответствует общемировой практике [1][2]. Особой задачей является тестирование подсистемы памяти микропроцессора, включающей систему трансляции адресов, кэшей различных уровней, многочисленных буферов и коммутаторов. Основными недостатками применяемых на данный момент генераторов псевдослучайных тестов, направленных на верификацию подсистемы памяти, является отсутствие критериев достаточности и точной оценки достигнутого покрытия в какой-либо функциональной метрике, а также, ввиду отсутствия единого подхода к разработке генераторов, сложность их поддержки и модификации для процессоров разных версий.

Существующие программные решения [3][4][5] ограничиваются подсистемами трансляции адресов и кэшами первого/второго уровня, в то время как возникают задачи по направленной генерации тестов на обнаружение ошибок во внеядерных устройствах подсистемы памяти (справочник многопроцессорной системы, буфер кэширования DMA-операций, системный коммутатор и т.д.). Применение автономной верификации к отдельным устройствам подсистемы памяти является недостаточным: анализ найденных в verilog-описании ошибок позволяет выделить подкласс ошибок подсистемы памяти, связанных с прохождением запроса в системе через определённую последовательность структурных блоков отдельных устройств подсистемы. Таким образом, необходимо средство генерации ассемблерных тестов, обеспечивающих требуемый маршрут запросов.

Для достижения этой цели предлагается метод представления верифицируемой системы в виде сети моделей отдельных узлов, связь между которыми осуществляется через обмен сообщениями. Узлом-источником запросов является исполняющее устройство процессора: при исполнении команды генерируется запрос с определёнными атрибутами (адрес, идентификатор адресного пространства, код операции и т.д.). В зависимости от атрибутов определяется узел-получатель запроса, который при его получении переадресует его следующему узлу и так далее, до окончательного исполнения операции. Таким образом, каждый узел устройства определяется функцией $f:(src, attr, X) \rightarrow (dst, attr', X')$, где src - узел, из которого приходит запрос с атрибутами $attr$, dst - узел, в который запрос перенаправляется,

с возможно изменившимися атрибутами $attr'$. X и X' - состояния узла перед и после получения запроса. Подобное представление позволяет достаточно гибко конфигурировать состав, топологию сети узлов и само описание узлов, определяя таким образом требуемую степень детализации модели системы. То есть, первые системные тесты, а значит и отчёты о найденных в тестируемом процессоре ошибках, можно получить уже при достаточно простом описании. В то же время, дальнейшая детализация позволит более направленно генерировать тесты для покрытия особых ситуаций.

Метрику функционального покрытия модели в данном представлении можно определить разными способами: покрытие состояний отдельного узла (устройства) сети, покрытие множества запросов для конкретного ребра (множества рёбер) сети, покрытие множества атрибутов входящих/исходящих сообщений для конкретного узла, покрытие множества последовательностей состояний, запросов и т. д. Таким образом, для генерации системного теста необходимо построить последовательность запросов для конкретного узла или ребра сети и определить приводящую к ней последовательность ассемблерных команд; атрибуты команд могут быть заданы в виде множеств допустимых значений, в то время как конкретные значения получаются путём псевдослучайного выбора из заданного множества.

В настоящее время идёт разработка прототипа системы генерации тестов для системы кэшей второго и третьего уровней проектируемого в ЗАО «МЦСТ» микропроцессора «Эльбрус-4С+».

Литература

1. *A. Adir [et al.]*. Genesys-Pro: Innovations in Test Program Generation for Functional Processor Verification. // IEEE Des. Test 21, 2 (March 2004). – 2004 – pp.84-93.
2. *F. Corno [et al.]*. Automatic Test Program Generation: A Case Study. // IEEE Des. Test 21, 2 (March 2004) – 2004 – pp.102-109.
3. *A. Adir [et al.]*. DeepTrans – a model-based approach to functional verification of address translation mechanisms. Microprocessor Test and Verification: Common Challenges and Solutions, 2003. Proceedings. 4th International Workshop. – 2003 – pp.3-6.
4. *Д.Н. Воробьев, А.С. Камкин*. Генерация тестовых программ для подсистемы управления памятью микропроцессора. // Труды Института системного программирования РАН. – 2009 г. – Т. 17. – С. 119-132.
5. *Е.В. Корныхин*. Генерация тестовых данных для системного функционального тестирования микропроцессоров с учетом кэширования и трансляции адресов. // Труды Института Системного Программирования. – 2009 г. – Т. 17. – С. 145-160.