

А. А. АНТОНОВ, С. В. БЫКОВСКИЙ, П. В. КУСТАРЕВ

МОНИТОР ВРЕМЕННЫХ ОГРАНИЧЕНИЙ ДЛЯ СИСТЕМ НА КРИСТАЛЛЕ

Предложен метод формального описания внутренних временных ограничений вычислительных систем на уровне отдельных коммуникационных операций (транзакций), ориентированный на снижение сложности аппаратного исполнения соответствующих встроенных средств мониторинга и диагностики. На базе предложенного подхода спроектировано IP-ядро монитора временных ограничений для систем на кристалле с топологией „общая шина“, приведены результаты его экспериментальной реализации на ПЛИС.

Ключевые слова: *монитор ограничений, система на кристалле, реальное время, LTL, PSL, RTL, SoC, TLM.*

Введение. Соблюдение временных ограничений для процессов протекающих в сложных вычислительных системах, предназначенных для управления объектами и обработки данных в режиме реального времени, является ключевой задачей разработчика таких систем. Планирование вычислительного процесса на этапе проектирования не гарантирует фактического исполнения заданных временных ограничений во время работы. Причины нарушений могут быть как внутренними, так и внешними. Возможны ошибки в выборе архитектуры, в реализации функциональных блоков или неточное представление разработчика об окружении, в котором будет эксплуатироваться устройство. Поэтому возникает необходимость внедрения механизмов мониторинга, отслеживающих нарушения динамики вычислительного процесса.

Реализация этого подхода в системах на кристалле имеет свои особенности: возможность широкой интеграции механизмов мониторинга на аппаратном уровне существенно расширяет функционал, но видоизменяет и усложняет проектирование. Именно этой проблеме посвящена данная статья.

Обзор работ. Современные технологии проектирования предполагают разработку систем на кристалле одновременно на нескольких уровнях: системном, программном, на уровне коммуникационных процессов (Transaction Level) и регистровых пересылок (register transfer level, RTL).

В работе [1] выполнен обзор методов контроля временных ограничений на программном уровне для систем, построенных на базе программно-управляемых процессоров. Приводится описание таких методов, как Hybrid, ZM4, Trams. Но поскольку в современных системах на кристалле программно-управляемый процессор не всегда является центральным или даже обязательным компонентом системы, то программная или аппаратно-программная реализация мониторинга временных ограничений будет ограниченной по охвату системы или вообще невозможной.

„Доступность“ разработчику RTL-уровня — уникальное свойство технологии систем на кристалле, позволяющее на уровне регистровых пересылок наблюдать за поведением всех элементов системы. В настоящее время для формализации временных ограничений на RTL-уровне широко используется математический аппарат темпоральной логики (LTL, CTL). На базе этого аппарата разработаны язык PSL (Property Specification Language) и расширение SVA (SystemVerilog Assertion) для языка SystemVerilog. Отметим, что существуют и другие методы формального описания временных ограничений, например, изложенные в статье [2].

Для проверки ограничений, заданных с помощью указанных средств, используется два основных подхода: Symbolic Model Verification (SMV) и Bound Model Checking (BMC).

Из-за вычислительной сложности они неудобны для полноценной реализации в аппаратных мониторах, поэтому актуальна задача выделения такого подмножества средств проверки ограничений, которое допускает эффективный перенос и использование на аппаратной платформе.

В работах [3, 4] авторы описывают подход, позволяющий с помощью описания ограничений и свойств системы на языке PSL автоматически создавать (с помощью компилятора MBAC) аппаратные блоки контроля ограничений системы на RTL-уровне. Основным недостатком такого подхода — необходимость анализа большого объема данных, так как результаты наблюдения включают в себя маркеры изменения шин или даже отдельных электрических сигналов. В таких условиях трудно быстро локализовать проблемное место, помимо того необходим большой объем встроенной памяти системы.

В работе [5] предлагается использовать иерархический подход к формальной верификации вычислительных систем. Приведена идея проверки ограничений сначала на уровне транзакций, т.е. отдельных операций обмена данными между устройствами, для локализации места сбоя, а затем на RTL-уровне — для установления их причин. Однако в статье не описаны даже примерные варианты аппаратных мониторов транзакций.

Авторами настоящей статьи развивается и доводится до уровня конкретной реализации идея, предложенная в работе [5]. Предлагается метод формального описания временных ограничений, обеспечивающий максимально простую организацию блока проверки заданных требований. Приводится описание экспериментального аппаратного монитора транзакций, способного обнаруживать нарушения временных ограничений в реальном масштабе времени.

Метод описания временных ограничений систем на кристалле на уровне транзакций. Временные ограничения на события (ev), происходящие в системе, опишем с помощью только одного типа выражений

$$ev1 \xrightarrow{ev3} ev2, \quad (1)$$

которое следует читать так: событие $ev2$ должно произойти после наступления события $ev1$, но до $ev3$. Задав ограничения в такой форме, возможно однозначно определить правила, определяющие условия актуализации ограничения, его нарушения и выполнения.

Под актуализацией ограничения понимается внесение его в список наблюдения. Предполагается, что по умолчанию ограничения не контролируются до тех пор, пока не актуализирована определенная причинно-следственная связь. Только когда возникла такая причина, начинается ожидание следствия. В выражении (1):

- актуализация ограничения происходит в момент фиксации события $ev1$;
- нарушение ограничения происходит в момент фиксации $ev3$, если не наступило событие $ev2$;
- выполнение ограничения происходит в момент наступления события $ev2$, если до этого не произошло $ev3$.

В рамках рассматриваемого подхода множество событий предлагается разделить на два типа: транзакции обмена данными по шине и временные метки. Для формализации ограничений требуется составить списки: контролируемых событий-транзакций, возможных временных меток, связей (ограничений) между событиями.

Предположим, что для системы заданы ограничения:

$$ev1 \xrightarrow{ev3} ev2, \quad (2)$$

$$ev2 \xrightarrow{ev4=10 \text{ мс}} ev3, \quad (3)$$

согласно которым событие $ev2$ должно произойти после события $ev1$, а событие $ev3$ — в течение 10 мс после $ev2$. При этом не задаются какие-либо ограничения относительно времени

между событиями $ev1$ и $ev2$. Для данного примера списки формального описания представлены в таблице.

id	События-транзакции		
1	Write	Addr1	Data1
2	Read	Addr2	Data2
3	Write	Addr3	Data3

id	Список временных меток
4	10 мс

id	Список связей между событиями		
1	1	3	2
2	2	4	3

События-транзакции задаются шаблонами обмена по шине: адрес, слово данных, тип операции (чтение или запись). Если по шине возможно передавать данные в потоковом режиме, то можно фиксировать только первое слово в транзакции, используемое в качестве маркера, по которому монитор будет выделять определенный тип событий.

Для упрощения обработки и однозначной идентификации различных событий производится их сквозная нумерация (номера — id). Временные события являются порожаемыми. Отсчет временных промежутков запускается по факту фиксации событий-транзакций. В приведенном примере временное событие $ev4$ вводится в систему в момент прихода $ev2$.

Реализация монитора временных ограничений для систем на кристалле. Для описанной модели ограничений был разработан аппаратный монитор временных ограничений на уровне транзакций (рис. 1). Блок фиксации транзакций между устройствами в системе Bus Event Catcher анализирует входные сигналы шины, сопоставляет их с шаблоном контролируемых транзакций, хранящихся в памяти, и на выходе формирует значение id соответствующего события. Блок Event Handler принимает на вход id произошедших событий и помещает их в очередь Event FIFO. По необходимости блок может вводить в систему временное событие (метку), отдавая соответствующую команду блоку Time Event Generator. Блок Time Event Generator содержит набор таймеров, которые командой от Event Handler обновляются и потом декрементируются импульсами тактового сигнала. При достижении соответствующим таймером нуля генерируется временное событие $time_ev_id$.

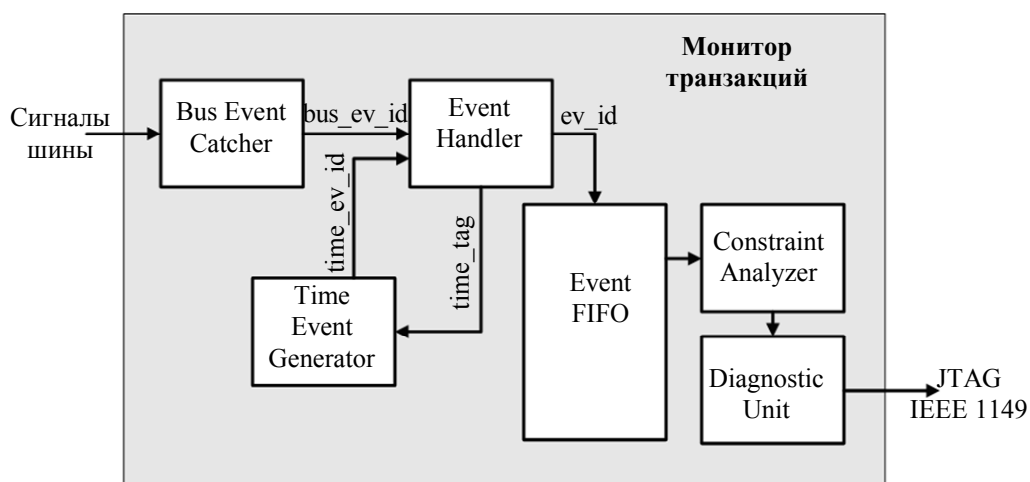


Рис. 1

Все возникающие в системе события-транзакции и временные события помещаются в порядке их возникновения в очередь Event FIFO для дальнейшей обработки (в очередь заносятся только id событий, это существенно упрощает дальнейшую обработку, так как для контроля временных ограничений не требуется сохранять и сверять время их поступления). Блок Constraint Analyzer включает память для списка всех ограничений и списка актуализированных ограничений. В процессе работы он выбирает по одному событию из очереди, анализирует список актуализированных ограничений, удаляет из него выполненные ограничения и

затем проверяет оставшиеся. Информация об обнаруженном нарушении ограничения помещается в диагностическую память блока Diagnostic Unit. В завершение анализируются все заданные ограничения и их часть, при необходимости, переносится в список актуализированных. Доступ к информации о нарушениях ограничений предоставляет контроллер диагностического интерфейса Diagnostic Unit. Конфигурация блоков монитора и съем результатов наблюдения за системой осуществляются с помощью интерфейса JTAG.

Разработанный аппаратный монитор прошел апробацию в составе экспериментальной системы на кристалле (рис. 2). Основные функции этой системы — настройка, сбор, обработка данных с „интеллектуальных“ датчиков (интерфейс I2C) и отправка результатов измерений потребителю (интерфейс CAN). В системе велось наблюдение за 11 событиями (типами транзакций). Прототип системы был реализован на ПЛИС Stratix IV GX EP4SGX230KF40C2N фирмы Altera. Функциональные параметры монитора: число используемых для реализации просмотрных таблиц ПЛИС — 1091, частота системного тактового сигнала, на котором работает разработанный монитор — 40 МГц, частота системной шины — 10 МГц, время обработки одного ограничения, которое определяет время принятия решения о нарушении наблюдаемого ограничения, равно $4T_{bus}$ (T_{bus} — время передачи слова данных по системной шине).

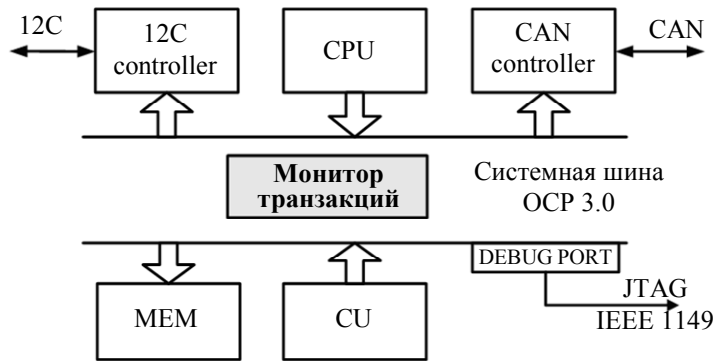


Рис. 2

Заключение. Авторами предложен метод формального описания временных ограничений для систем на кристалле на уровне транзакций. Метод обеспечивает эффективную аппаратную реализацию конечного аппаратного блока монитора.

На базе описанного метода разработан аппаратный монитор контроля временных ограничений для систем на кристалле с топологией „общая шина“. Монитор обеспечивает проверку системы в реальных условиях функционирования, с учетом всех факторов внешнего окружения; он характеризуется высокой скоростью процесса верификации требований (недостижимой при отладке модели системы в симуляторе) и возможностью контроля временных ограничений при эксплуатации системы.

СПИСОК ЛИТЕРАТУРЫ

1. Asadi N., Saadatmand M., Sjodin M. Run-Time Monitoring of Timing Constraints: A Survey of Methods and Tools // ICSEA 2013: The 8th Intern. Conf. on Software Engineering Advances. 2013. P. 391—401.
2. Aloysius K., Liu M., Liu G. Efficient Run-Time Monitoring of Timing Constraints // Proc. of the 3rd IEEE Real-Time Technology and Applications Symposium (RTAS '97). IEEE Computer Society. Washington, 1997. P. 252—257.
3. Boule M., Chenard J., Zilic Z. Adding Debug Enhancements to Assertion Checkers for Hardware Emulation and Silicon Debug // Computer Design. Intern. Conf. ICCD 2006. 2007. P. 294—299.
4. Boule M., Chenard J., Zilic Z. Assertion Checkers in Verification, Silicon Debug and In-Field Diagnosis // ISQED '07. The 8th Intern. Symp. on Quality Electronic Design. 2007. P. 613—620.

5. Kong Z., Bian J., Zhao Y., Deng S. Hierarchical Formal Verification Methodology based on Transactions // Workshop on RTL and High Level Testing. 2009. P. 1—4.

- Александр Александрович Антонов** — *Сведения об авторах*
— студент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: antonov.alex.alex@gmail.com
- Сергей Вячеславович Быковский** — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники; E-mail: bsv.serg@gmail.com
- Павел Валерьевич Кустарев** — канд. техн. наук, доцент; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра вычислительной техники;
E-mail: kustarev@lmt.ifmo.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
23.12.13 г.

УДК 004.896

В. И. ПОЛЯКОВ, В. И. СКОРУБСКИЙ

ИСПОЛЬЗОВАНИЕ МНОГОЗНАЧНОЙ ЛОГИКИ ПРИ ПРОЕКТИРОВАНИИ ФУНКЦИОНАЛЬНЫХ СХЕМ

Рассмотрены приложения многозначной логики к задачам анализа цифровых схем (в том числе функционального и временного моделирования и тестирования). Рассматриваются особенности применения булевой алгебры к вычислениям над собственными подмножествами.

Ключевые слова: моделирование цифровых схем, временное моделирование, тестирование схем, трюичное тестирование.

Введение. Из многочисленных способов кодирования информации особый интерес представляет многозначная (более двух значений переменных) логика как один из опытов расширения границ осознания и формального описания логических связей реального мира. В работе [1] многозначность определена как способ отображения различных смысловых оттенков информации в рассуждениях. Таким образом, возникло направление многозначной логики, в котором работали математики, экономисты, философы, заинтересованные в повышении качества передачи информации в рассуждениях. Многозначные признаки информации — различные уровни дискретности (0, 1, 2) — в измерениях (мало, больше, среднее значение, много), ощущениях и представлениях требуют оценки, распознавания и принятия решений, несмотря на нечеткость значений. Таким образом, многозначная логика приобретает содержание в практических приложениях.

Временное моделирование цифровых схем. В САПР цифровых схем моделирование и тестирование работоспособности — обязательный этап, это единственный доступный при проектировании метод проверки качества синтеза схем с учетом реальных условий их работы.

Применяемые двузначные дискретные сигналы вследствие емкостной нагрузки могут быть представлены как трехзначные или четырехзначные на переходах, когда формируются значения на выходах логических элементов.