

АППАРАТНОЕ УСКОРЕНИЕ ЦИФРОВОГО МОДЕЛИРОВАНИЯ

В. С. Зайцев

ВВЕДЕНИЕ

При автоматизированном проектировании цифровых ИС значительную часть времени занимает выполнение моделирования их HDL моделей. Это обусловлено тем, что для обеспечения достаточной полноты контроля необходимо проверить их поведение на большом числе тестов.

Во многих случаях выполнить это за разумное время без использования специальных программно-аппаратных систем [1] практически не возможно. Ниже описывается результат разработки такой системы.

ПРЕДПОСЫЛКИ ВЫПОЛНЕНИЯ РАЗРАБОТКИ

Время, затрачиваемое на обработку теста в программной среде моделирования, всегда значительно больше времени обработки этого же теста в готовой ИС. На этом основаны все попытки использования в системах моделирования различных прототипов ИС. Такие системы обычно включают специальный процессор, эмулирующий поведение создаваемой микросхемы, и дополнительное ПО для его использования.

Дальнейшим развитием этой идеи стало применение в них вместо спецпроцессора ПЛИС прототипа. Синтез такого прототипа ИС при наличии ее HDL модели не является на данный момент проблемой. А вот систем верификации для работы ПЛИС макета совместно с уже имеющимся тестовым окружением HDL модели на данный момент не так уж и много. Примерами таких решений являются системы, предлагаемые фирмами Altera и Synopsys. Вариант, предлагаемый фирмой Altera [1] включает ПО для моделирования HDL моделей (Riviera-PRO), ПО для прошивки прототипа (Active-HDL), и специальную плату (например, ALDEC HES5) с каналом ее подключения к компьютеру. Аналогичный набор предлагает и фирма Synopsys: плату для прототипирования (Synopsys HAPS) и ПО для прошивки FPGA и моделирования (Synopsys Virtualizer) [2]. Основным недостатком этих решений является то, что их стоимость даже без учета стоимости соответствующей лицензии составляет около 800 000\$. Это не позволяет использовать их при проектировании небольших, но требующих тщательной отладки ИС.

ОСНОВНЫЕ ПРОБЛЕМЫ РЕАЛИЗАЦИИ

Аппаратно-программная система моделирования должна помимо ускорения обработки обеспечивать интерактивный режим работы, иными словами, моделирование должно проходить в стандартной среде верификации без отличий в процессах запуска и обработки тестов с аппаратным ПЛИС эмулятором от программной обработки HDL описания.

В стандартной среде программного моделирования имитируется параллельный характер обработки информации, присущий аппаратуре. Это означает, что при каждом переключении синхросигнала последовательно обрабатывают все активные в данный момент времени операторы HDL описания и обработка на данном такте считается завершенной лишь после обработки всех таких операторов. Понятно, что при таком подходе модельное время одного такта обработки всегда больше его реального значения в аппаратуре и зависит от общего числа элементов в ИС, структуры их соединения, стиля и языка описания модели. Узким местом реализации в такой системе интерактивного режима отслеживания обработки является производительность компьютера. При использовании же ПЛИС прототипа модельное и реальное время обработки одного такта могут совпадать. Однако, для использования этой возможности необходимо решение задачи быстрой пересылки данных, включающей выбор типа адаптера, и формата передаваемых через него сообщений.

ОПИСАНИЕ НАЙДЕННОГО РЕШЕНИЯ

При создании описываемой системы моделирования была выбрана стандартная среда моделирования, поддерживающая язык SystemC [3], доступная по цене плата для прототипирования, адаптер одного из возможных интерфейсов, для подключения FPGA платы к компьютеру [4].

Для выполнения моделирования в ПЛИС макете ИС размещаются следующие компоненты: модуль верхнего уровня, включающий приёмник, передатчик; блок основной схемы – блок формирования общего сброса, блок формирования общего синхросигнала, блок формирования управляемого синхросигнала, драйвер, монитор, кодер, декодер (рис. 1).

Алгоритм работы основного блока включает:

1. установку начального значения на входах аппаратного эмулятора (реализуется вызовом соответствующей функции);
2. ожидание изменения активных сигналов (синхросигналов или сигналов данных, активизирующих асинхронную обработку);

3. при изменении активного сигнала (сигналов) запоминание состояния всех входов и фиксация активного сигнала (сигналов);
4. вызов функции, которая посылает команды и данные ПЛИС эмулятору, т.е. отсылает состояние входов, вид события и ожидает ответ;
5. приём данных от эмулятора о новом состоянии всех выходов и внутренних сигналов, после чего они назначаются соответствующим выходам и внутренним сигналам модели;
6. переход к ожиданию следующего события (возврат на пункт 2).

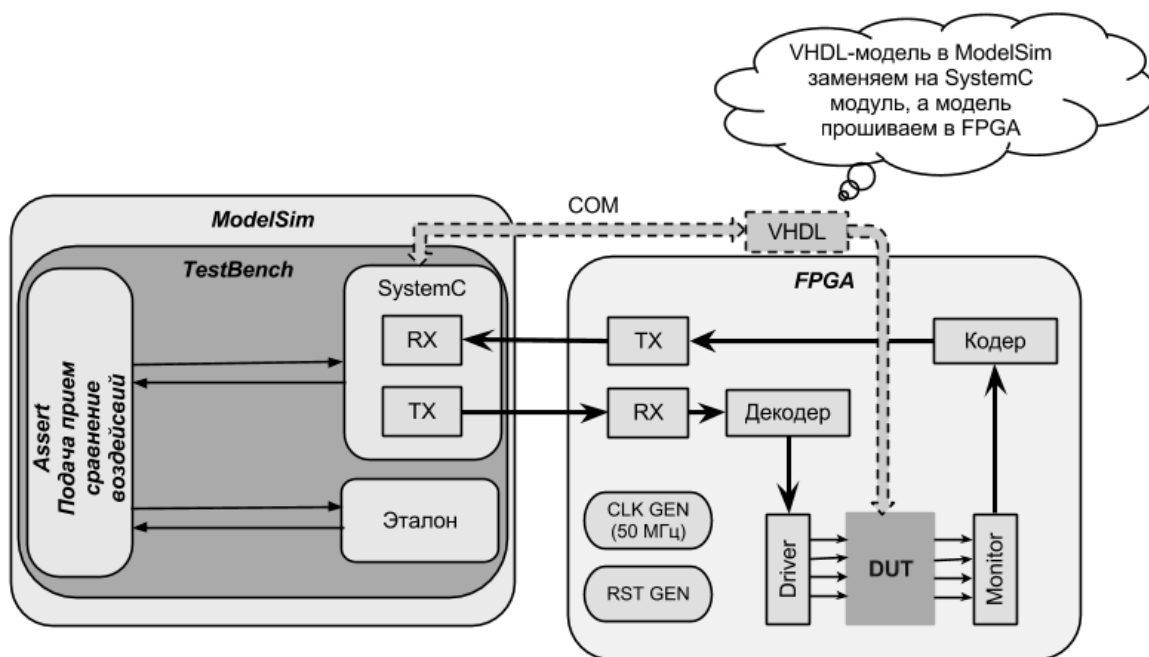


Рис. 1. Структурная схема системы аппаратного ускорения моделирования

Таким образом, пересылка данных между ПЛИС макетом и основным персональным компьютером идет в форме обмена сообщениями: запрос-ответ. Так как от формата сообщений существенно зависит трафик общения, для оптимизации этого узкого места в созданной системе были использованы сообщения разной длины, которые помимо данных содержат дополнительные поля для описания события: EventSync, Events, AsyncPortData, SyncPortData (рис. 2).

Длина входного сообщения определяется состоянием флага EventSync. Если этот флаг равен 0, то выполняется событие без изменения синхронных данных относительно предыдущего события. В этом случае поле SyncPortData будет отсутствовать. Если флаг EventSync равен 1, то имеет место событие с предварительным изменением синхронных данных. Размер следующего поля Events определяется по формуле $\log_2(N+M)$, где N – число синхросигналов, M – число асинхронных сигналов. Первыми записываются синхросигналы, затем асинхронные сигналы. Также в полях фиксируется ClockNumber (число синхросигналов),

AsyncPortNumber (число асинхронных сигналов), AsyncPortFirstNumber (порядковый номер первого асинхронного сигнала). Если событие вызвано изменением только одного вывода, то синхросигналы и асинхронные сигналы не нумеруются, а передается номер соответствующего сигнала. Третье поле AsyncPortData принимает значение активного асинхронного сигнала и игнорируется при событии на синхросигнале. Четвертое поле SyncPortData принимает значения синхронных сигналов. Его ширина (SyncPortDataWidth) равна числу цифровых синхронных сигналов.

EventSync	Events	AsyncPortData	SyncPortData
0	$\log_2 (\text{ClockNumber} + \text{AsyncPortNumber})$	0/1	SyncPortDataWidth-1:0
1	$\log_2 (\text{ClockNumber} + \text{AsyncPortNumber})$	0/1	–

Рис. 2. Формат входного сообщения

Выходное сообщение состоит из двух полей: Outputs, его ширина равна числу выходов, и InternalSignals, его ширина равна числу внутренних сигналов, выводимых на временные диаграммы.

ВЫВОДЫ

В рамках описываемого исследования был разработан SystemC модуль, содержащий C++ функции отправки, ожидания и считывания данных через com-порт. На базе микросхемы MAX3232 была реализована схема адаптера для подключения FPGA платы к компьютеру. Создан VHDL модуль для прошивки в FPGA, обеспечивающий прием и передачу данных между платой и компьютером через com-порт. Разработан описанный выше формат обмена сообщениями. Опробование разработанной системы моделирования проводилось в среде QuestaSim версии 10.1. Для компиляции SystemC модуля использовалось средство gcc-4.2.1. Разработанные функции модуля работают в среде Windows.

Литература

1. Aldec [Электронный ресурс]. Электронные данные. Режим доступа: <http://www.aldec.com/en/>.
2. Synopsys [Электронный ресурс]. Электронные данные. Режим доступа: <http://www.synopsys.com/SYSTEMS/FPGABASEDPROTOTYPING/Pages/HAPS.aspx>.
3. Black D. C. and Donovan J., "SYSTEMC: FROM THE GROUND UP" Kluwer Academic Publishers, Boston, 2004.
4. Зайцев В. С. Аппаратное ускорение цифрового моделирования / В.С. Зайцев, В.Я. Степанец // Информационные технологии и системы 2012 (ИТС-2012): материа-

СВОБОДНЫЕ КОЛЕБАНИЯ СЖАТОЙ ДВУХПРОЛЕТНОЙ БАЛКИ В СЛУЧАЕ СМЕЩЕНИЯ ВНУТРЕННЕЙ УПРУГОЙ ОПОРЫ

В. А. Кельин

В процессе строительства мостов опоры могут быть выставлены со смещением. Важно уметь оценивать последствия таких ошибок. В [1] было исследовано влияние смещения внутренней упругой опоры двухпролетной балки (см. рис. 1.) на потерю устойчивости, оно приводит к *локализации* прогиба – вертикальные перемещения балки сосредотачиваются в одном из ее пролетов. Представляет интерес выяснить, как влияет смещение внутренней опоры на свободные колебания балки.

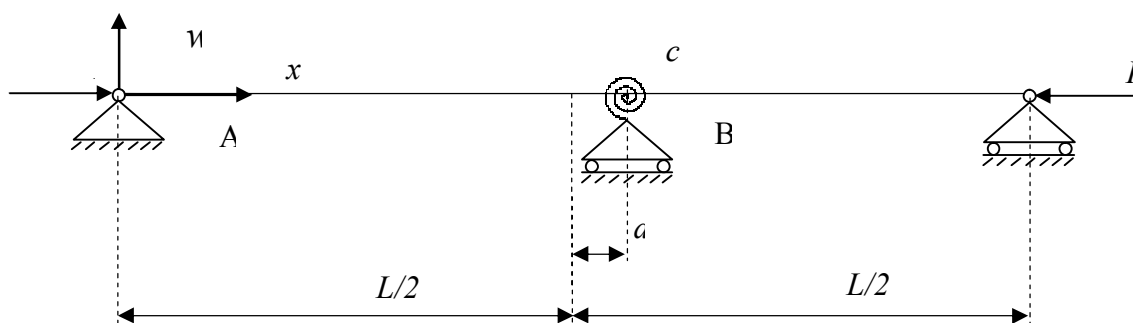


Рис. 1. Схема нагружения балки

Целью работы явилось определение формы свободных колебаний двухпролетной балки в зависимости от величины сжимающей силы и смещения внутренней упругой опоры, а так же качественный анализ полученной зависимости.

В рамках классической механики было получено аналитическое решение для колебаний балки, численными методами определены собственные частоты в зависимости от безразмерного смещения опоры $\delta = d / L$. Формы колебаний для некоторых значений параметра δ приведены на рис. 2.