

АППАРАТНО-ПРОГРАММНОЕ МОДЕЛИРОВАНИЕ ВСТРАИВАЕМЫХ СИСТЕМ

Н.Н. Шарипова, В.Е. Третьяков, Э.А. Завадская

EMBEDDED SYSTEM HW/SW-MODELING

N.N. Sharipova, V.E. Tretjakov, E.A. Zavadskaja

В статье обсуждается использование разработанного авторами VHDL-симулятора, предназначенного для HW/SW-проектирования встроенных систем. На простом примере проиллюстрированы возможности симулятора, методологические аспекты разработки VHDL-описаний поведенческих моделей аппаратуры, обсуждена концепция моделирования, приведены результаты верификации модели.

Ключевые слова: встроенная система, совместное аппаратно-программное проектирование, VHDL-симулятор, имитационное моделирование.

VHDL-simulator for embedded system HW/SW design is represented in this article. Hardware mode! definition methodology, modeling conceptions and verification process are discussed. The content is illustrated by simple example.

Keywords: embedded system, HW/SW-Codesign, VHDL-simulator, simulation.

Введение

Проектирование встроенных систем предполагает совместное HW/SW-проектирование аппаратных блоков встроенного процессора и управляющих алгоритмов специального программного обеспечения (СПО). Необходимым этапом проектирования является верификация поведенческих моделей блоков проектируемого аппаратно-программного комплекса. Для спецификации объектов проекта используется стандартизированный IEEE многоуровневый язык описания аппаратуры VHDL [1], а для моделирования используются разработанные на базе этого языка симуляторы.

Авторами данной статьи разработаны: компьютерная технология HW/SW-проектирования систем [2], работающих под управлением встроенных процессоров; методология описания на VHDL программных и аппаратных блоков встроенных систем и концепция совместного имитационного

моделирования [3]. Разработанная компьютерная технология предполагает: параллельное проектирование аппаратных компонентов и управляющих алгоритмов СПО; возможность стыковки аппаратных и программных блоков на разных (и довольно ранних) этапах проектирования; использование VHDL на всех этапах. Для поддержки этой технологии авторами разработан и реализован VHDL-симулятор [4]. Использование VHDL-симулятора позволяет обрабатывать управляющие алгоритмы СПО в «окружении» аппаратуры, представленной не только на уровне информационных связей, но если нужно, и поведенческих моделей. С другой стороны, проектируемые аппаратные блоки на любом этапе могут быть состыкованы с СПО в единую модель с целью контроля за удовлетворением совместных требований и ограничений.

Существующие в этой области инструментальные системы (такие как, **Silicon**, **Vantage**,

Шарипова Нелли Николаевна - старший научный сотрудник НИИ физики и прикладной математики Уральского государственного университета им. А.М. Горького, г. Екатеринбург; nelly.sharipova@usu.ru

Третьяков Владимир Евгеньевич - д-р техн. наук, профессор, чл.-кор. РАН, президент Уральского государственного университета им. А.М. Горького, г. Екатеринбург; Vladimir.Tretjakov@usu.ru

Завадская Эрна Абрамовна - старший научный сотрудник НИИ физики и прикладной математики Уральского государственного университета им. А.М. Горького, г. Екатеринбург; nelly.sharipova@usu.ru

Sharipova Nelly Nikolaevna - senior scientist of Institute of Physics and Applied Mathematics of Ural State University of A.M. Gorky, Ekaterinburg; nelly.sharipova@usu.ru

Tretjakov Vladimir Evgenjevich - PhD, professor, corresponding member of RAS, president of Ural State University of A.M. Gorky, Ekaterinburg; Vladimir.Tretjakov@usu.ru

Zavadskaja Erna Abramovna - senior scientist of Institute of Physics and Applied Mathematics of Ural State University of A.M. Gorky, Ekaterinburg; nelly.sharipova@usu.ru

Cadence и др.), разработанные крупными зарубежными фирмами (такими как, Mentor Graphics и др.), являются сложными в использовании, дорогостоящими и, в основном, ориентированы на проектирование аппаратуры. Аналогичные отечественные разработки отсутствуют.

Данная статья имеет своей целью ознакомить отечественных разработчиков с возможностями VHDL-симулятора и проиллюстрировать на простом примере его использование для верификации поведенческих моделей аппаратных блоков.

1. Общая характеристика VHDL-симулятора

В качестве языковой основы VHDL-симулятора выбрано подмножество VHDL, необходимое для описания поведенческих моделей аппаратуры уровня СБИС. Подмножество расширено структурированными типами данных и типами данных, необходимыми для представления управляющих алгоритмов СПО, соответствующими операциями и встроенными функциями.

В концепции совместного HW/SW-моделирования реализован принцип «событийного» моделирования. Имитационная модель в этой концепции представляется как совокупность взаимосвязанных и синхронизируемых во времени параллельных процессов. Каждый процесс в общем случае соответствует некоторому блоку моделируемой системы (как аппаратному, так и программному). Процесс может иметь список запуска, т. е. перечисление тех входных сигналов, при изменении значения которых он активизируется. Процесс без списка запуска считается активным с начала моделирования.

В определенный момент модельного времени каждый процесс может находиться в одном из трех состояний: в активной фазе; в состоянии ожидания некоторого события; в завершенном состоянии.

Функционирование модели протекает в дискретном модельном времени, которое имитирует реальное время как неубывающую последовательность отдельных моментов времени. При этом приращение времени на каждом шаге моделирования считается равным интервалу между двумя следующими друг за другом событиями. Под событием понимается одна из следующих ситуаций: сигнал из списка запуска некоторого процесса получил новое значение; выполнено заданное условие активизации какого-либо процесса; для некоторого процесса, находящегося в состоянии ожидания, истекла заданная временная пауза.

Процесс моделирования удобно разделять на циклы прогона модели. В одном цикле параллельно должны быть запущены на выполнение те процессы, для которых в течение предыдущего цикла произошло какое-либо из вышеуказанных событий. Цикл прогона считается завершенным, когда завершены все запущенные в начале цикла процессы.

Для реализации симулятора использована среда C++. Приведем структуру и краткую характеристику основных компонентов симулятора [4]:

- транслятор описаний компонентов модели с подмножества VHDL во внутреннее представление (объектный код модели). Объектный код модели представляет из себя совокупность файлов с текстами на языке C++, а также специальный файл с информацией о структуре модели. Транслятор осуществляет синтаксический и семантический анализ VHDL-текста, декомпозицию поведенческих архитектурных тел и генерацию объектного кода модели;

- компоновщик модели осуществляет: проверку корректности модели по структуре, входам-выходам и переменным связи; окончательную сборку (синтез) модели из заданных VHDL-описаний и типовых элементов, подключаемых из пакетов или библиотек; формирование файла для редактора связей системы C++, получение «.exe»-модуля для модели;

- интерпретатор предназначен для проведения верификации модели во время сеанса моделирования и реализует выполнение действий, заданных на входном языке или специальные отладочные действия;

- планировщик событий - управляющая программа симулятора, которая обеспечивает функционирование имитационной модели во время сеанса моделирования в соответствие с разработанной концепцией моделирования;

- монитор диалога поддерживает многооконный интерфейс с пользователем при проведении сеанса моделирования, обеспечивая следующие возможности: запуск и прекращение сеанса; контроль значений переменных и сигналов; изменение значений переменных и сигналов; контроль активизации процессов; подключение и отключение сигналов; различные остановы и возобновление выполнения; ведение протокола сеанса моделирования;

- блок визуализации результатов моделирования, обеспечивает: просмотр всех промежуточных и окончательных результатов и состояние модели в заданный момент времени; выдачу временных диаграмм указанных сигналов и трассировку процесса моделирования на заданном промежутке времени.

Заметим, что представление каждой модели в виде отдельного «.exe»-модуля, обеспечивает эффективное проведение моделирования и позволяет накапливать модели разрабатываемых устройств, что может значительно упростить проектирование, реализацию и тестирование сложных систем.

2. Пример использования VHDL-симулятора

Кратко рассмотрим методологию описания моделей, включающих аппаратные и программные блоки. При совместном проектировании аппаратные блоки могут быть описаны в виде: структурных

или поведенческих архитектурных тел, процессов, операторов назначения сигнала. Управляющие алгоритмы СПО представляются в виде: поведенческого архитектурного тела, процессов или обрабатывающих процедур. В интерфейс модели, кроме входных и выходных сигналов и параметров настройки, включаются также сигналы и переменные связи аппаратных и программных блоков.

В качестве иллюстрации приведем пример простой модели (устройства с обратной связью), включающей: описание на VHDL поведенческой модели, управляющую программу для верификации модели, результаты имитационного моделирования, полученные с помощью разработанного авторами VHDL-симулятора.

На рис. 1 изображена логическая схема устройства с обратной связью, включающая два типовых блока: схему «ИЛИ-НЕ» и элемент памяти (D-триггер).

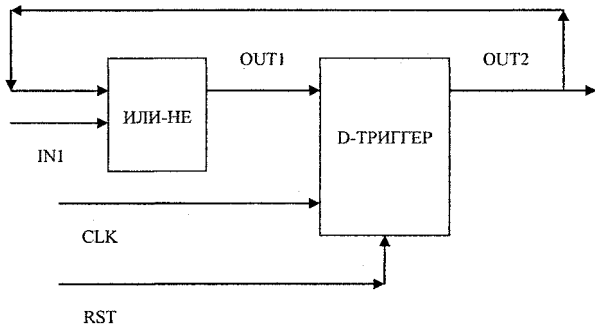


Рис. 1. Логическая схема устройства с обратной связью

Используются следующие обозначения:

IN1 — входной сигнал схемы;

CLK - входной тактирующий сигнал триггера;

OUT 1-выходной сигнал схемы и входной сигнал триггера;

OUT2 - выходной сигнал триггера и входной сигнал схемы;

RST-входной сигнал для сбрасывания триггера в ноль.

Выберем процессную форму поведенческой модели. Типовые блоки представлены следующим образом: оператор назначения сигнала с инерционной задержкой для схемы «ИЛИ-НЕ» и процесс со списком запуска для триггера. Оба блока будут смоделированы как параллельные процессы.

Текст VHDL-описания поведенческой модели для схемы с обратной связью,

—описание интерфейса

entity SCH is

generic (OUT1D: time = 2ns; — задержка в распространении сигнала OUT1

OUT2D: time = 1ns; — задержка в распространении сигнала OUT2

RSD1: time = 1ns; — периоды переключения входного сигнала

RST

RSD2: time = 180ns;

CLKD: time = 3ns;

IN1D: time = 5ns);

Port (OUT2: inout bit;

OUT1: inout bit;

RST: in bit;

CLK: in bit = B'0';

IN1: in bit = B'0');

end SCH;

-- Поведенческое архитектурное тело

architecture TR of SCH is

Begin

-- оператор назначения сигнала для схемы
OUT1<= not (IN1 or OUT2) after OUT1D;

-- процесс для D-триггера

CG: process (CLK,RST,OUT1) is

begin

if RST = B'1' then

OUT2 <= B'0'after OUT2D;

else

if not CLK'STABLE(0) then

if CLK = B'1' then

OUT2 <= OUT1 after OUT2D;

end_if;

end_if;

end_if;

end_process CG;

end TR;

Тестовая программа для запуска и верификации модели:

architecture TEST of S is

signal RU: bit;

-- сигнал «кнопка-ПУСК»

begin

-- запуск модели

RU <= B'1';

RST <= B'1';

-- процесс для генерации входного сигнала RST

G1: process (RST) is

begin

if S <0 then RST <= B'1' after RSD2;

else RST <= B'0' after RSD1;

end_if;

S= 0 - S;

end_process G1;

-- процесс для генерации периодических переключений входного сигнала CLK

G2: process (RU,CLK) is

begin

CLK <= not CLK after CLKD;

end_process G2;

```

— процесс для генерации периодических переключений входного сигнала IN1
G3: process (RU,IN1) is
begin
IN1 <= not IN1 after IN1D;
end_process G3;
end TEST;

```

Архитектурное тело TEST, предназначенное для верификации модели, состоит из трех параллельных процессов, с помощью которых генерируются переключения входных сигналов. Для начального запуска всей модели предназначен сигнал RU (кнопка «ПУСК»), оператор назначения которому интерпретируется, как процесс без списка запуска, и будет запущен в начальный момент времени.

В процессе верификации модели может быть произведен подбор параметров настройки (CLKD, IN1D, OUT1D, OUT2D) и выявлены соотношения задержек, при которых моделируется корректная работа схемы и некоторые неисправности. Мы ограничились случаем, когда сигнал RST, по которому осуществляется сброс триггера, подается один раз в начальный момент времени. Это позволило нагляднее проиллюстрировать подбор остальных параметров, влияющих на работу модели.

В результате процесса верификации установлено, что достаточным условием для моделирова-

ния обратной связи является следующее соотношение задержек:

$$IN1D > OUT1D > OUT2D.$$

Ниже представлены результаты моделирования в виде временных диаграмм симулятора с различными вариантами задержек.

Вариант 1 (наличие обратной связи):

CLKD = 4 ns, IN1D = 3 ns, OUT1D = 2 ns, OUT2D = 1 ns (рис. 2).

Из рис. 2 видно, что цикл прогона для варианта 1 составляет 24 ns.

Продолжительность цикла прогона можно скорректировать с помощью подбора периода переключения сигналов.

Например, для:

CLKD = 5 ns, IN1D = 3 ns - цикл прогона 30 ns;

CLKD = 5 ns, IN1D = 2 ns - цикл прогона 20 ns.

Вариант 2: (моделирование неисправности схемы)

CLKD = 4 ns, IN1D = 2 ns, OUT1D = 2 ns, OUT2D = 1 ns (рис. 3).

Из диаграмм видно, что для данного соотношения задержек IN1D, OUT1D нарушается условие стабильности при передаче сигнала OUT1 и на триггер ничего не поступает. Таким образом, обратная связь не реализуется. Этот вариант можно рассматривать как моделирование неисправности схемы.

Результаты верификации свидетельствуют об адекватности рассмотренной модели процессу функционирования схемы с обратной связью.

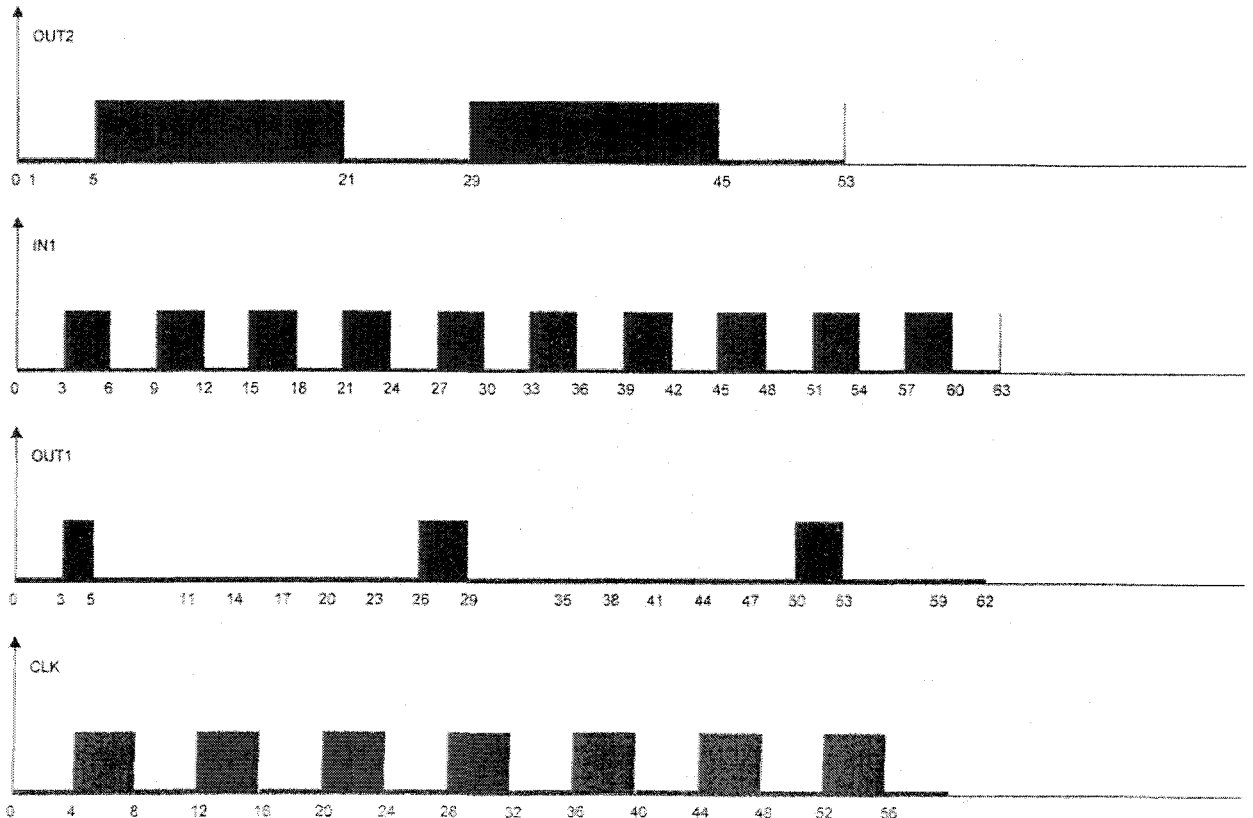


Рис. 2. Временные диаграммы для варианта 1

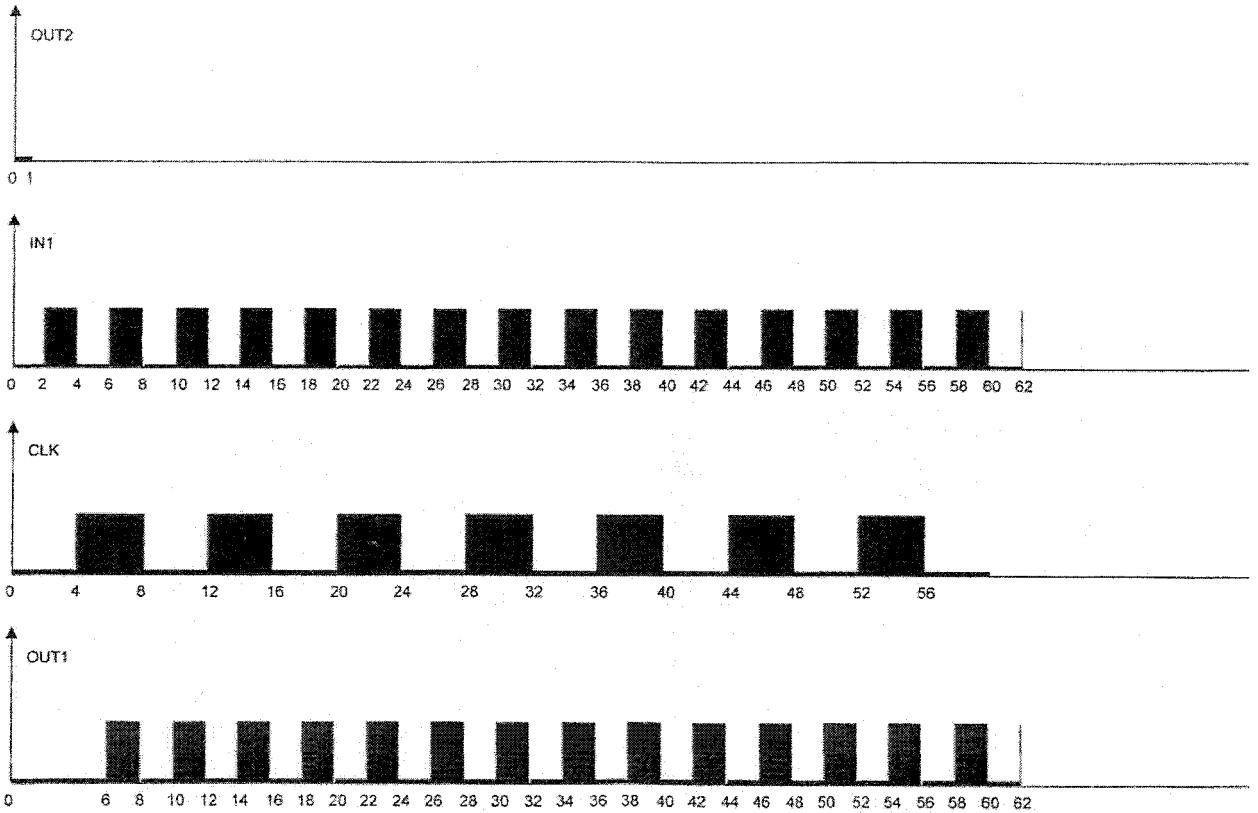


Рис. 3. Временные диаграммы для варианта 2

Заключение

В заключение отметим достоинства предлагаемого симулятора. Во-первых, он содержит минимальный, но достаточный и удобный набор инструментальных средств для большинства этапов HW/SW-проектирования; не требует специальной адаптации и настройки на конкретную область применения, прост в использовании; позволит повысить качество и надежность разрабатываемых изделий, снизить затраты на стендовые испытания и сократить сроки разработки. Во-вторых, это отечественная разработка и его стоимость значительно ниже, чем у известных систем крупных производителей.

Разработки встроенных систем на отечественной элементной базе в нашей стране проводились, в основном, в оборонной и авиакосмической промышленности, опто-электронном приборостроении, в производстве сложной медицинской и др. техники. С начала 90-х годов прошлого века появилась ориентация на использование зарубежных микропроцессоров с зарубежным математическим обеспечением. Этот подход оказался не эффективным и не обеспечивающим надежности разработок. Поэтому в последнее время обозначилось направление на поддержку отечественных разработок элементной и программной базы спецпроцессоров. В таких проектах предлагаемый симулятор может быть реально востребован.

Разработка первой версии VHDL-симулятора проводилась при поддержке РФФИ (2005-2006 годы), № проекта: 05-08-01273. В настоящее время проводится расширение симулятора возможностями объектно-ориентированного моделирования в рамках АВЦП «Развитие научного потенциала высшей школы (2009-2010 годы)» Минобрнауки РФ, № проекта: 2.1.2/3675.

Литература

1. IEEE Standard VHDL Language Reference Manual / Std 1076-1993, revision of IEEE Std. 1076-1987. 1994.
2. Третьяков, В.Е. Проектирование аппаратно-программных комплексов с использованием VHDL-симулятора / В.Е. Третьяков, Н.Н. Шарипова, Э.А. Завадская // Изв. вузов. Приборостроение. - 2004. - Т. 47, №2.-С. 27-30.
3. Шарипова, Н.Н. Моделирование аппаратно-программных комплексов на базе VHDL / Н.Н. Шарипова, В.Е. Третьяков, Э.А. Завадская // Автоматизация и современные технологии. - 2007. — № 6.— С. 22-27.
4. Третьяков, В.Е. VHDL-симулятор как средство верификации поведенческих моделей блоков встроенных систем / В.Е. Третьяков, Н.Н. Шарипова, Э.А. Завадская // Автоматизация и современные технологии. - 2002. — № 1.—С. 7—11.

Поступила в редакцию 18 февраля 2010 г.