

Методы создания прототипов встраиваемых систем

Зыкун Р.В.

Кафедра ПОИТ, ФКСиС

Белорусский государственный университет информатики и радиоэлектроники

Минск, Беларусь

e-mail: lfrontcom@gmail.com

Аннотация — В данной работе рассмотрены методы прототипирования встраиваемых систем. Обозначен метод программного построения VHDL-описаний встраиваемых систем.

Ключевые слова: прототипирование; моделирование; встраиваемые системы; ПЛИС

I. ВВЕДЕНИЕ

Для решения круга прикладных задач, требующих аппаратной реализации могут применяться ASIC, либо ПЛИС. Первое решение предлагает максимальную скорость выполнения, минимальный размер занимаемого физического пространства. Второе – гибкость.

В рыночных условиях, при высоком уровне конкуренции, требование к скорости разработки является критическим. Агрессивная конкуренция оказывает давление на время выхода проекта на рынок. Например, цикл проектирования ASIC может составлять от 9 до 18 месяцев, в то время как «окно» возможностей для внедрения продукта – от 2 до 4 месяцев [1].

При этом цена ошибки, в зависимости от того, на каком этапе она была обнаружена, может стать недопустимой.

Эти факторы привели к спросу на быстрые, эффективные методы проектирования.

II. ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА

Общие требования к встроенным системам: скорость разработки, энергопотребление, занимаемое физическое пространство.

Применение ПЛИС при разработке встроенных систем позволяет получить оптимальный баланс между скоростью и гибкостью, оставляя возможность реконфигурации для внесения изменений, исправления ошибок.

Встроенная система, реализованная на ПЛИС, делает возможным реконфигурацию в целях расширения функционала и исправления причин сбоев, связанных, в том числе, с ошибками аппаратной архитектуры, в процессе сопровождения.

Частичная реконфигурация позволяет системе продолжать работать в то время, когда другие части могут быть перепрограммированы. С одной стороны это позволяет статической части управлять реконфигурацией других частей, что называется «частичная самостоятельная реконфигурация» [2]. С другой стороны – возможность повторного использования конкретной области при внесении изменений в функциональность.

При разработке встроенных систем использование ПЛИС в качестве технологической платформы позволяет производить отладку архитектуры в короткие сроки, сокращая время, затрачиваемое на уточнение требований и создание прототипа.

ПЛИС позволяют использовать описания устройств, предназначенные для производства на интегральных

схемах или системах на кристалле. Перед производством таких микросхем их аппаратные описания, выполненные на языках описания аппаратуры Verilog или VHDL, тестируются, верифицируются и отлаживаются. Этот этап занимает до 70% времени, затраченного на разработку [3].

III. CO-DESIGN

Разработка встроенной системы состоит из шести шагов: анализ, проектирование, реализация, тестирование, внедрение и сопровождение.

Программно-аппаратный Co-Design предлагает одновременное развитие программного обеспечения и аппаратуры в процессе проектирования за счет использования высокого уровня абстракции моделирования на основе ПЛИС прототипов и ASIC эмуляторов [4].

IV. ПРОТОТИПИРОВАНИЕ

Прототип – макет, разработанный с целью проверки пригодности предлагаемых для применения концепций, архитектурных или технологических решений.

Этапы создания прототипа: определение первоначальных требований, разработка первого варианта прототипа, его изучение и определение необходимых изменений и дополнений, переработка и улучшение.

Использование программных эмуляторов обеспечивает высокую прозрачность проектирования при невысокой стоимости. Недостатком является то, что они очень медленные в отношении моделирования больших ASIC.

Для достижения высокой скорости моделирования необходимо использовать методы аппаратной поддержки верификации. Выделяют три категории [5]: использование ускорителей, эмуляция, ПЛИС прототипы.

Использование аппаратных ускорителей направлено на ускорение моделирования ASIC в изоляции, т.е. эта форма проверки не проверяет устройство в контексте системы.

Преимуществом эмуляции является то, что представления интегрированы в систему на уровне среды.

Недостатком – скорость работы на порядок ниже, чем фактическая скорость ASIC.

ПЛИС прототипы позволяют проверить быстродействие архитектуры. Решение заключается в создании аппаратных прототипов ASIC с использованием одного или нескольких ПЛИС.

V. ИНСТРУМЕНТЫ

Хотя для программирования ПЛИС и используется очень мощный VHDL, более эффективным является использование инструментов для быстрого прототипирования при выполнении проектирования системы. Такие инструменты позволяют осуществить проектирование в графической среде на основе различных блоков, которые представляют собой

типичные функции. На основе этого может быть получен VHDL код. Наряду с преимуществом автоматической генерации кода инструменты для быстрого прототипирования позволяют выполнить функциональный тест конструкции, который, как правило, является трудоемкой задачей.

Инструменты быстрого прототипирования, в общем случае, находятся на более высоком уровне, чем компиляторы/синтезаторы. Они предоставляют функциональные возможности, которые улучшают или облегчают функциональное моделирование, позволяя получить результат на ранней стадии проектирования [6][7].

Использование таких инструментов делает возможным создание прототипа относительно функциональных требований уже на начальных этапах разработки без принятия решений относительно аппаратной архитектуры.

Основными компонентами инструментария для прототипирования являются ассемблер, компоновщик, симулятор, отладчик.

Общей задачей таких инструментов является алгоритмизация процесса создания прототипа.

Разработка ассемблера предполагает наличие редактора команд, исполняемых арифметико-логическим устройством, с учетом возможности внесения изменений в формат команд и, как следствие – изменений в описание арифметико-логических блоков и блока декодера.

Справедливо и обратное – проектирование архитектуры, добавление новых блоков и редактирование существующих требует внесения корректировок в ассемблер разрабатываемого проекта.

На основе описаний программных процессоров или библиотеке унифицированных описаний модулей встроенных систем могут быть реализованы инструменты для быстрой разработки.

VI. ПРОГРАММНАЯ КОМПОНОВКА VHDL-ОПИСАНИЙ

Для решения задач создания прототипа предлагается использование метода программного построения унифицированного описания встроенной системы на языке описания аппаратуры с учетом требований к аппаратной реализации, задаваемых входными параметрами, описывающими архитектуру и программное обеспечение разрабатываемого решения.

Для целей структуризации хранения описаний используется библиотека, в которой определены интерфейсы модулей, на основе требований к программной и аппаратной частям.

Цель разрабатываемого метода – его использование при разработке программного инструментария для проектирования набора команд и компоновки описаний архитектуры встроенной системы.

Функциональное назначение метода – поддержка взаимобратной связи разработки архитектуры встроенной системы и процесса проектирования набора команд, используемого в ней. Поставленная задача порождает две подзадачи – создание инструментария для работы с ассемблером и модуля обработки описаний проектируемой системы.

Такой подход предполагает использование формализованного описания проектируемой системы. Описание модулей встроенной системы выполняется на языке описания аппаратуры VHDL. Для организации хранения используется библиотека аппаратных

модулей, структурированная по внешним интерфейсам модулей. Функциональное назначение аппаратной и программной части задается высокоуровневым описанием, построенным на основе требований к системе, с указанием типов операций.

Метод, реализованный в программном обеспечении, позволит вести разработку встроенной системы по принципу «сверху-вниз», уточняя описания системы на каждом последующем шаге.

Так на основании функциональных требований к системе на первом шаге уточняются те функции, которые будут реализованы аппаратно и функции, реализуемые программно. Перемещение между обоими списками будет доступно в любой момент времени.

На втором шаге задаются требования, описывающие конфигурацию аппаратуры. Значения, не заданные пользователем присваиваются как значения по умолчанию. Это позволит прорабатывать конфигурацию с необходимой точностью, в том числе не высокой.

Третий шаг – программирование алгоритмов на заданном ассемблере, который строится на основании функциональных требований к программной части системы.

На основании полученных данных (программных и аппаратных функциональных требований, конфигурации аппаратуры, программных алгоритмов) производится компоновка VHDL-описаний готовой системы.

Степень детализации, с которой пользователь прорабатывает описания системы, определяется выполненной декомпозицией каждого из шагов. Так, если не задана архитектура разрабатываемой системы, то на этапе компоновки будет использована библиотечная универсальная архитектура.

VII. ЗАКЛЮЧЕНИЕ

В настоящий момент изложенный подход прорабатывается. Использование метода, реализованного в программном средстве, предполагается возможным при разработке проектов невысокой сложности, либо в целях отработки решений на начальных этапах проектирования.

[1] EETimes [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.eetimes.com/>.

[2] Design and Reuse [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.design-reuse.com/>

[3] Christopher Lavin. Rapid Prototyping Tools for FPGA – NSF Center for High-Performance Reconfigurable Computing, – 2005 – 4 p

[4] Wayne H. Wolf. Hardware-software Co-Design of Embedded Systems – IEEE, – 1994 – 23 p.

[5] Рубанов, В. В. Автоматизация построения инструментария кросс-разработки программного обеспечения для расширяемых встраиваемых систем / В. В. Рубанов. – М. : Институт системного программирования, 2008. – 208 с.

[6] Sivarama P. Dandamudi. Guide to RISC Processors – Springer Science+Business Media, – 2005 – 387 p.

[7] Xilinx [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.plis.ru/>.