

Мешков А. Н. , Черных С. В. , Куцевол В.Н.

ЗАО «МЦСТ»

Разработка методологии и инструментов создания потактовых программных моделей цифровых устройств

Одним из основных этапов разработки современных микросхем является верификация — процесс определения, соответствует ли реализуемая система заявленной, а также выявление и регистрация ошибок, допущенных в процессе разработки оборудования. По различным данным 50-80% от объема трудозатрат на разработку посвящено верификации [1]. В компании ЗАО “МЦСТ” активно используется автономная верификация составных модулей выпускаемого оборудования. Один из основных методов автономной верификации является сравнение поведения RTL (register transfer level) описания устройства и альтернативной программной модели. Так как оборудование, модель которого необходимо получить, может иметь сложную внутреннюю структуру, актуален вопрос разработки методологии создания моделей и необходимого инструментария.

Потактовые модели — это модели, повторяющие поведение с точностью до тактового сигнала. Такая точность необходима для проверки полного соответствия спецификации верифицируемого устройства.

В реальных устройствах тактовый сигнал приходит во все модули одновременно, и логика осуществляется параллельно. Программная модель, напротив, являясь пользовательским приложением, исполняется последовательно, поэтому требуется реализовать механизмы поддержания псевдопараллельности.

Один из возможных вариантов решения – это создание внутри модулей, входящих в модель, объектов для приема и передачи данных, а также организация цикла работы, состоящего из двух стадий: на первой происходят все чтения из объектов для передачи в объекты для приема, а на второй — выполнение действий, реализующих логику устройства, и запись результата в объект для передачи данных.

К недостатками данного подхода относятся дублирование переменных, разнообразие типов данных, передаваемых между модулями, затрудняющее стандартизацию, и не соответствие механизма реальному (логика заключена внутри модулей).

Другая реализация данного подхода позволяет устранить перечисленные недостатки. Логика выносится в модули, не содержащие данных — action. На первой стадии во всех

action создаются функциональные объекты, принимающие неизменное состояние системы. На второй стадии вызов созданных функциональных объектов изменяет состояние системы.

Данный подход позволяет описать модуль, как совокупность структур данных и связей между ними. Подобное унифицированное описание модели дает возможность создания генераторов кода (например, из визуальной среды).

Для action через переопределение стандартных операторов были разработаны средства создания и композиции, что позволяет создавать модели, не вдаваясь в детали реализации модуля action.

Литература

1. *William K. Lam* , Hardware Design Verification: Simulation and Formal Method-Based Approaches, 2005.