

УДК 662.642: 621.926.7

В.А. Сухомлинов (5 курс, каф. ИУС), А.В. Лебедь (5 курс, каф. ИУС),  
В.П. Котляров, к.т.н., проф.

## АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ТЕСТОВ НА БАЗЕ MSC-ДИАГРАММ

MSC – Message Sequence Charts – диаграммы последовательности сообщений (стандарт ITU z.120) – язык, предназначенный для спецификации и описания поведения связи компонентов системы и их окружения посредством обмена сообщениями. Основное применение MSC находят в областях, где необходимо описывать протокольные взаимодействия (телекоммуникации, распределенные системы и др.). Наиболее широкое распространение MSC получили в области телекоммуникаций (iDEN, TETRA).

Целью данной работы является исследование эффективности применения MSC для автоматической генерации тестов по спецификации системы на примере модели шинного интерфейса и сравнение результатов с тестами, написанными вручную на Verilog. В рассматриваемом примере модель шинного интерфейса разрабатывалась на С для подключения в Verilog-симулятор.

Применительно к модели шинного интерфейса MSC представлены двумя объектами: окружением системы и самой системой. Спецификация системы задает сценарии поведения - последовательности ответов системы на запросы окружения. К спецификации системы, заданной в виде MSC была добавлена оболочка, транслирующая сообщения MSC в вызовы модели. При генерации тестов сообщение, передаваемое от окружения в систему, преобразуется в вызов функции модели на языке С; сообщение, передаваемое от модели, преобразуется в проверку того, что модель вызвала соответствующую функцию. Процесс тестирования сводится к разработке сценариев из MSC диаграмм, которые составляются на основе требований на проект. Далее производится генерация тестов по созданным MSC диаграммам: тесты преобразуются в код, реализованный на языке С, который компилируется вместе с моделью и выполняется тестовый прогон.

Система автоматизированного тестирования состоит из следующих компонентов: MSC диаграммы с макро-подстановками, программа раскрытия макро-подстановок, тестовый генератор, окружение модели и сама модель. MSC диаграммы разрабатываются по спецификациям и являются тестами на проект. Программа раскрытия макро-подстановок выполняет замену макросов на MSC диаграммах на константы, при этом производится увеличение исходного числа диаграмм в несколько раз за счет того, что вместо одного макроса подставляется несколько констант. Таким образом, одна MSC диаграмма преобразуется в большое количество MSC диаграмм. Далее эти диаграммы поступают на вход тестового генератора, который преобразует их в программу, написанную на языке С, выполняющую действия, соответствующие действиям на MSC диаграмме. Эта программа взаимодействует с окружением модели, которое взаимодействует с самой моделью. Окружение модели реализовано на языке С и необходимо для соединения интерфейса модели с интерфейсом сгенерированного кода. Также окружение модели содержит параллельно работающие процессы, которые позволяют отдельно работать и с моделью и со сгенерированным кодом (реализацией MSC диаграмм), что позволяет убрать ненужные сообщения из MSC диаграмм и преобразовать большое количество сообщений, пришедших из модели в одно сообщение на MSC диаграмме.

Тестирование с использованием MSC технологии было проведено на модели размером примерно 50Kloc С кода. Общий объем ручного труда при разработке MSC диаграмм оказался примерно в 4.5 раза меньше объема ручного труда при написании аналогичных тестов на языке Verilog. За счет использования макро-подстановок число

тестов было увеличено примерно в 100 раз, это составляет примерно  $1,4 \cdot 10^4$  тестов и  $4 \cdot 10^5$  сообщений. Время тестового прогона всех тестов составило 2,5 часа на Pentium-III 500. Варьирование параметров в макро-подстановках привело к тому, что в модели было найдено дополнительно 5 ошибок. Общий объем тестового окружения составил примерно 4 тысячи строк С кода.

Таким образом, применение MSC для тестирования в технологических цепочках показывает превосходство над традиционными подходами и позволяет сократить время на разработку тестов и проведения тестирования. В итоге это приводит к сокращению цикла разработки программного продукта заданного качества и имеет экономический смысл.