Московский физико-технический институт

ЗАО «МЦСТ»

Исследование возможности создания системы верификации, управляемой тестовым покрытием

Магистерская диссертация
студента 415 группы ФРТК
Рыжова Михаила Петровича

Научный руководитель
Гурин К.Л.

Москва, 2010

Управляемая покрытием генерация тестов (coverage-driven test generation)

— метод генерации тестовых воздействий, использующий обратную связь по величине тестового покрытия

Преимущества:

- обеспечивает более быстрое достижение требуемого покрытия за счет обратной связи
- предоставляет критерий выбора момента остановки генерации
- метод может быть применен к существующим системам генерации тестов

Требования к системам моделирования

Управляемая покрытием генерация тестов требует от моделирующей системы поддержки

- исполнения тестов по мере их генерации
- возможности измерять покрытие в ходе генерации теста
- восстановления и сохранения состояния модели с высокой частотой

Требования к создаваемой системе

- Быстрое сохранение/восстановление из контрольных точек
- Вычисление покрытия во время генерации тестов
- Использование изменения величины покрытия для управления генерацией тестовых стимулов

Выбор базового программного средства

Рассмотренные варианты:

- Используемые средства моделирования

  Не позволяют производить быстрое восстановление и сохранение состояний

- Другие коммерческие средства

  Заявленная поддержка требуемой функциональности. Высокая стоимость

- Verilator

  Свободное ПО. Удовлетворяет требованиям

Verilator:ограничения и недостатки

Ограничения:

- Неполная поддержка стандарта Verilog

неподдерживаемые конструкции не являются синтезируемыми

- Verilator не позволяет обнаруживать ошибки, связанные с временными характеристиками логических вентилей

задачей данной работы является логическая верификация, анализ временных характеристик проводится другими средствами

- Не моделирует ячейки с тремя состояниями

ячейки с тремя состояниями не используются во внутренних модулях

Недостатки:

- Ресурсоемкость компиляции сгенерированного кода
  перекомпиляция требуется только при изменении RTL-кода
- Не сохраняется логическая структура RTL-кода

Verilator: особенности

- Позволяет транслировать код на языке verilog в C++-модель
- Имеет открытий исходный код
- Откомпилированный код сравним по скорости моделирования с коммерческими пакетами
- Реализация позволяет работать непосредственно с состоянием модели
- Позволяет генерировать временные диаграммы сигналов и измерять покрытие RTL-кода
- Использует циклически-ориентированный подход к моделированию

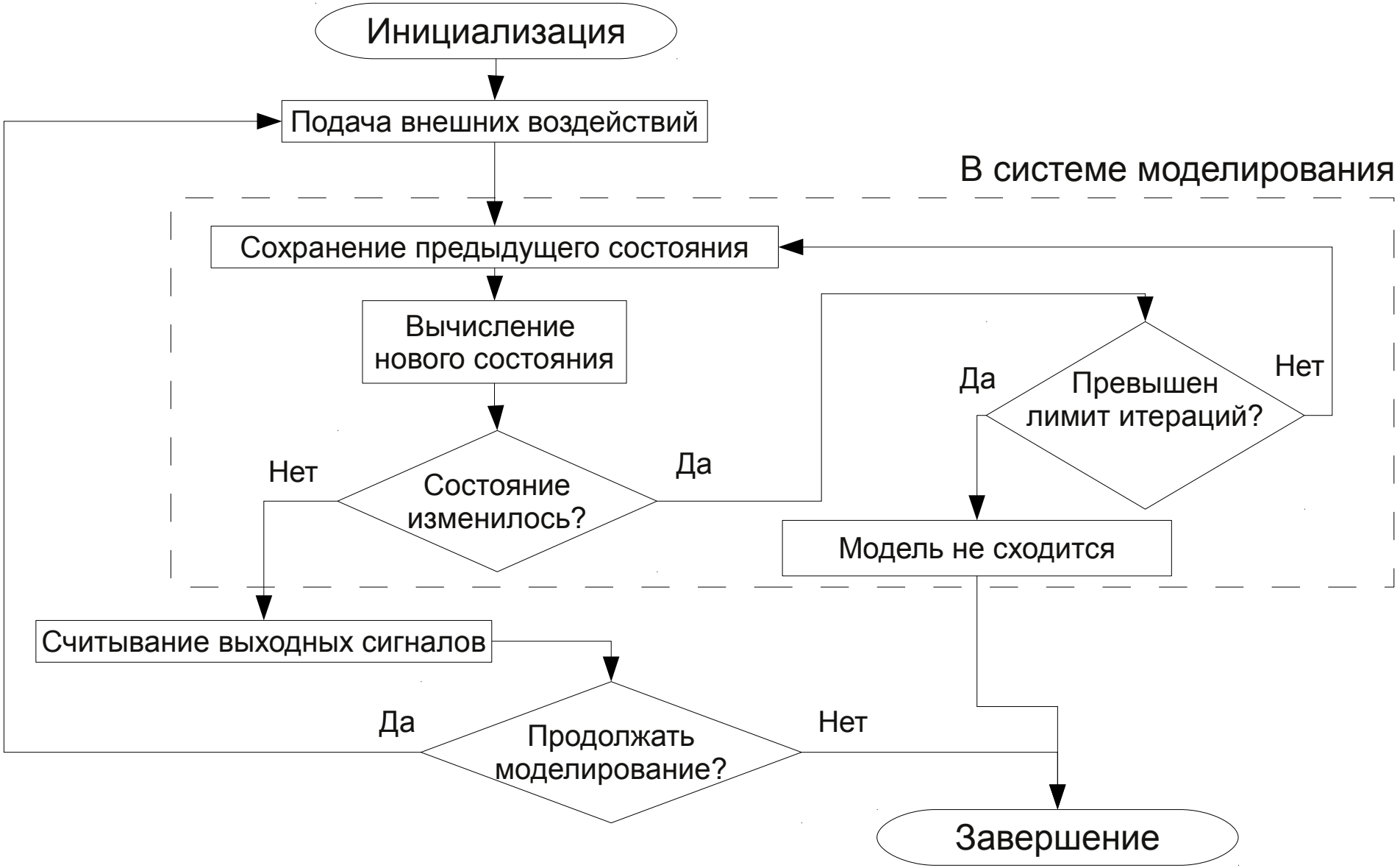
Методы моделирования RTL

Verilator использует **циклически-ориентированный подход,** позволяющий добиться более высокой производительности, но не учитывающий задержки логических элементов

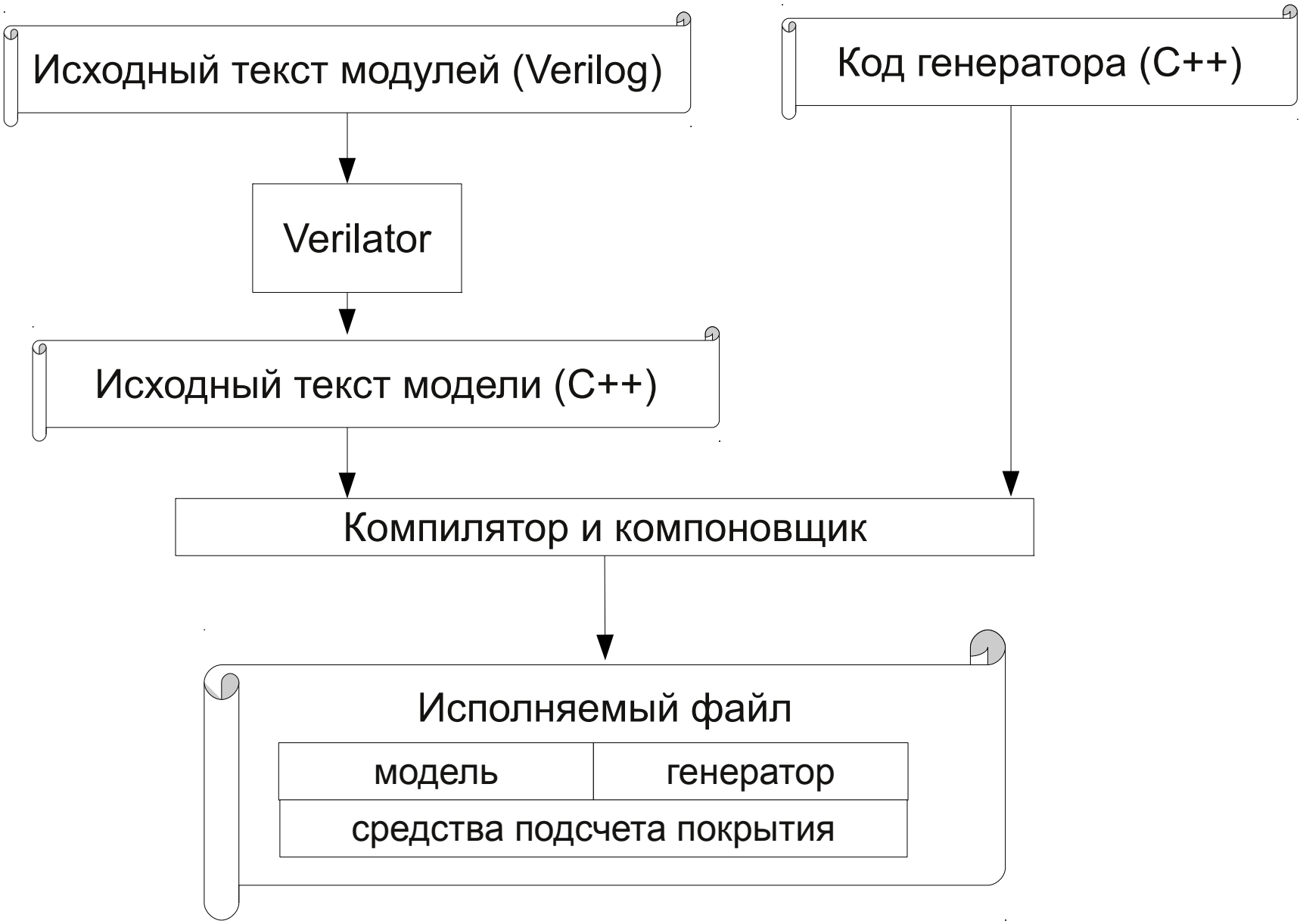
Большинство коммерческих симуляторов используют **событийно-ориентированная модель вычислений**

- Позволяет учитывать сложные временные характеристики
- Обеспечивает высокую гибкость моделирования

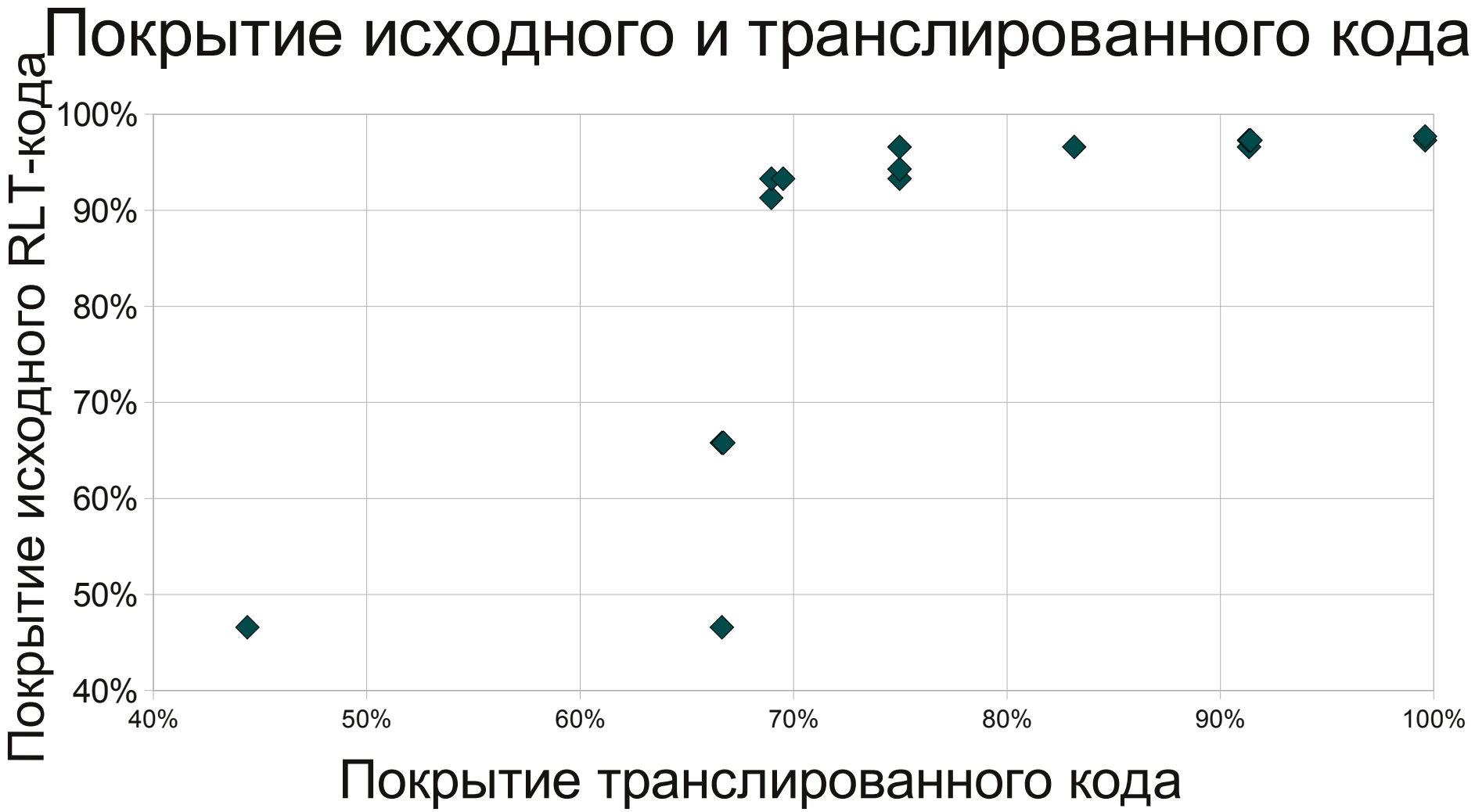
Циклически-ориентированный подход к моделированию RTL



Формирование исполняемого файла



Покрытие транслированного кода как независимая метрика покрытия



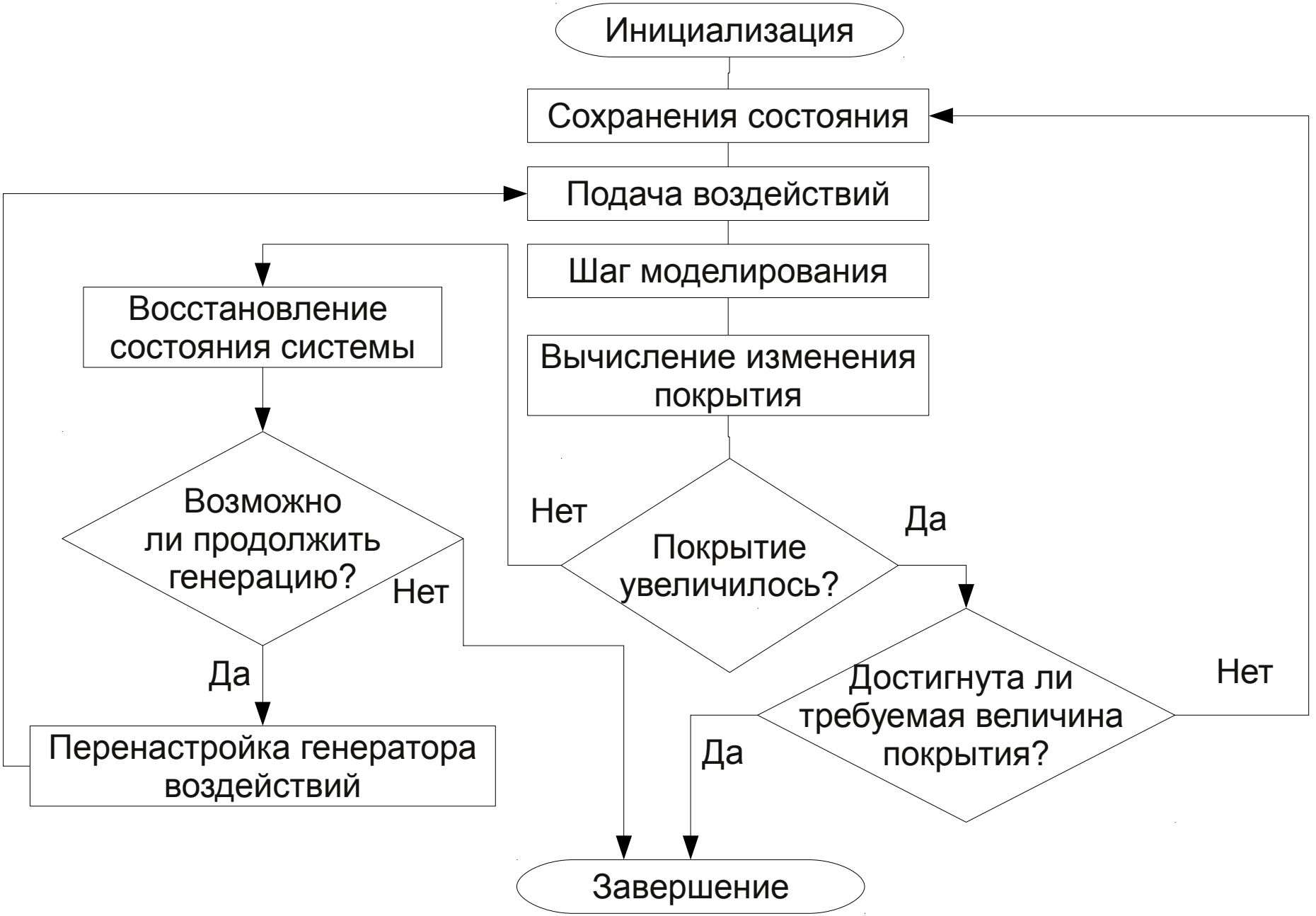
Обеспечение подсчета покрытия в реальном времени

Применяется встроенное средство сбора тестового покрытия компилятора gcc (gcov)

Реализации сбора статистики:

1. Использование сохранения данных в файл и отдельной программы для их анализа
   - Не зависит от версии компилятора
   - Реализация более проста и надежна
2. Считывание собираемой статистики непосредственно из памяти
   - Обеспечивает более высокое быстродействие
   - Требует использования измененной библиотеки libgcov
   - Более сложен в реализации и менее надежен
   - Требуется адаптация реализации к конкретной версии gcc

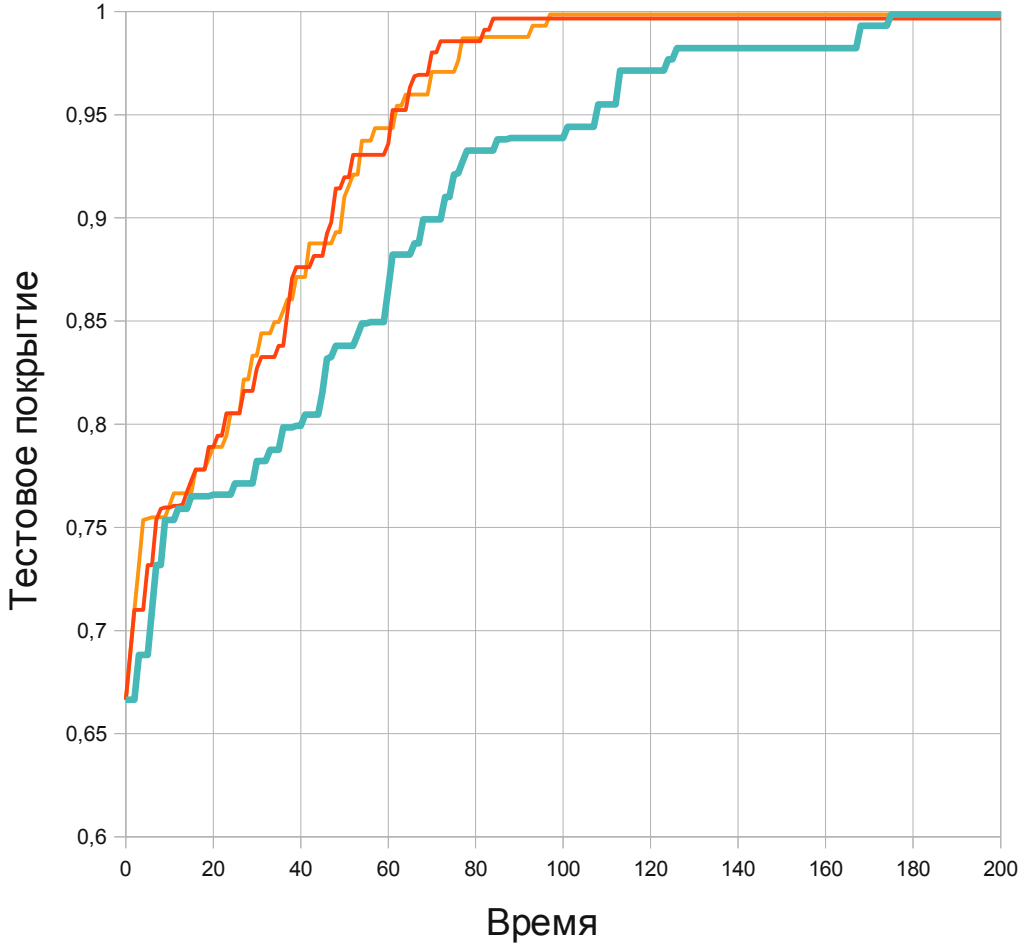
Алгоритм работы генератора



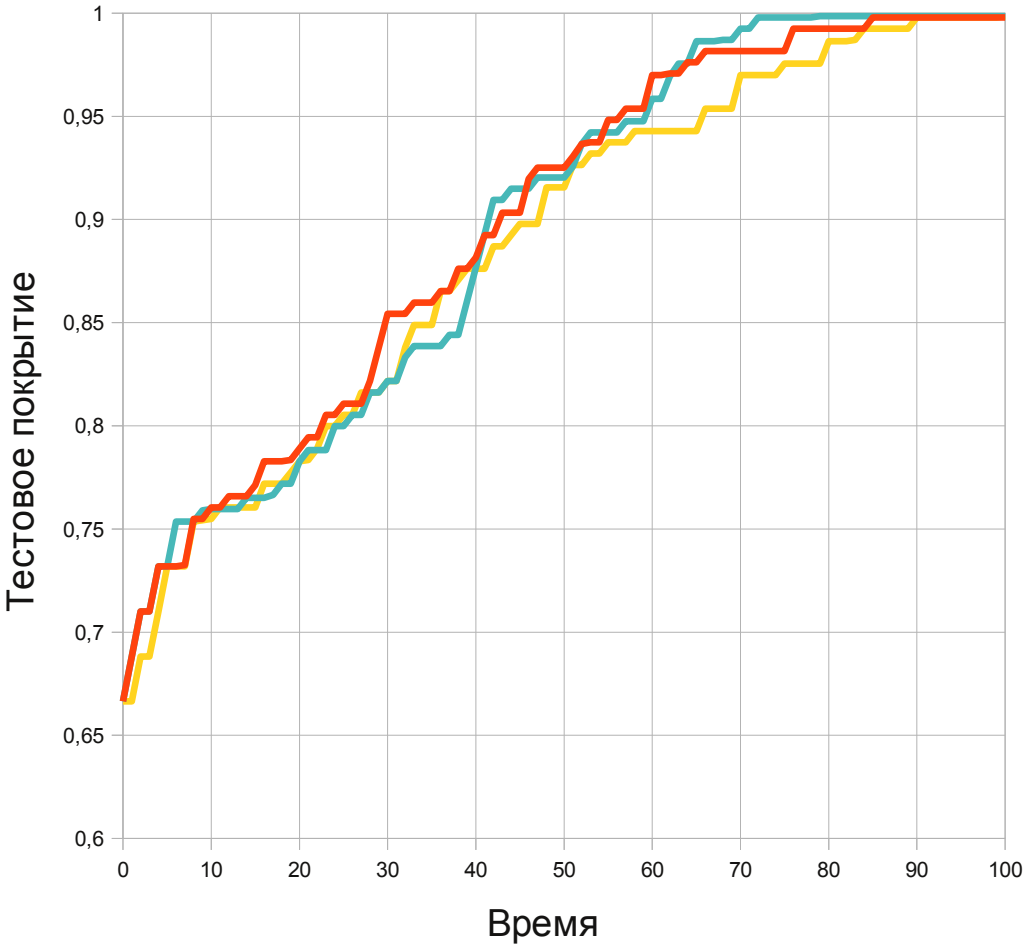
Сравнение с не использующим покрытие генератором

(слайд 1 из 2)

Случайно выбранные параметры генератора



Вручную подобранные параметры генератора



генератор псевдослучайных воздействий, не учитывающий покрытие

управляемый покрытием генератор при различных частотах восстановления

Сравнение с не использующим покрытие генератором

(слайд 2 из 2)

Использование покрытия позволяет

- быстрее достигать требуемого покрытия
- избежать ручного подбора оптимальных параметров при генерации псевдослучайных тестов
- автоматически производить отбор параметров генератора
- использовать изменение покрытия как критерий прекращения генерации

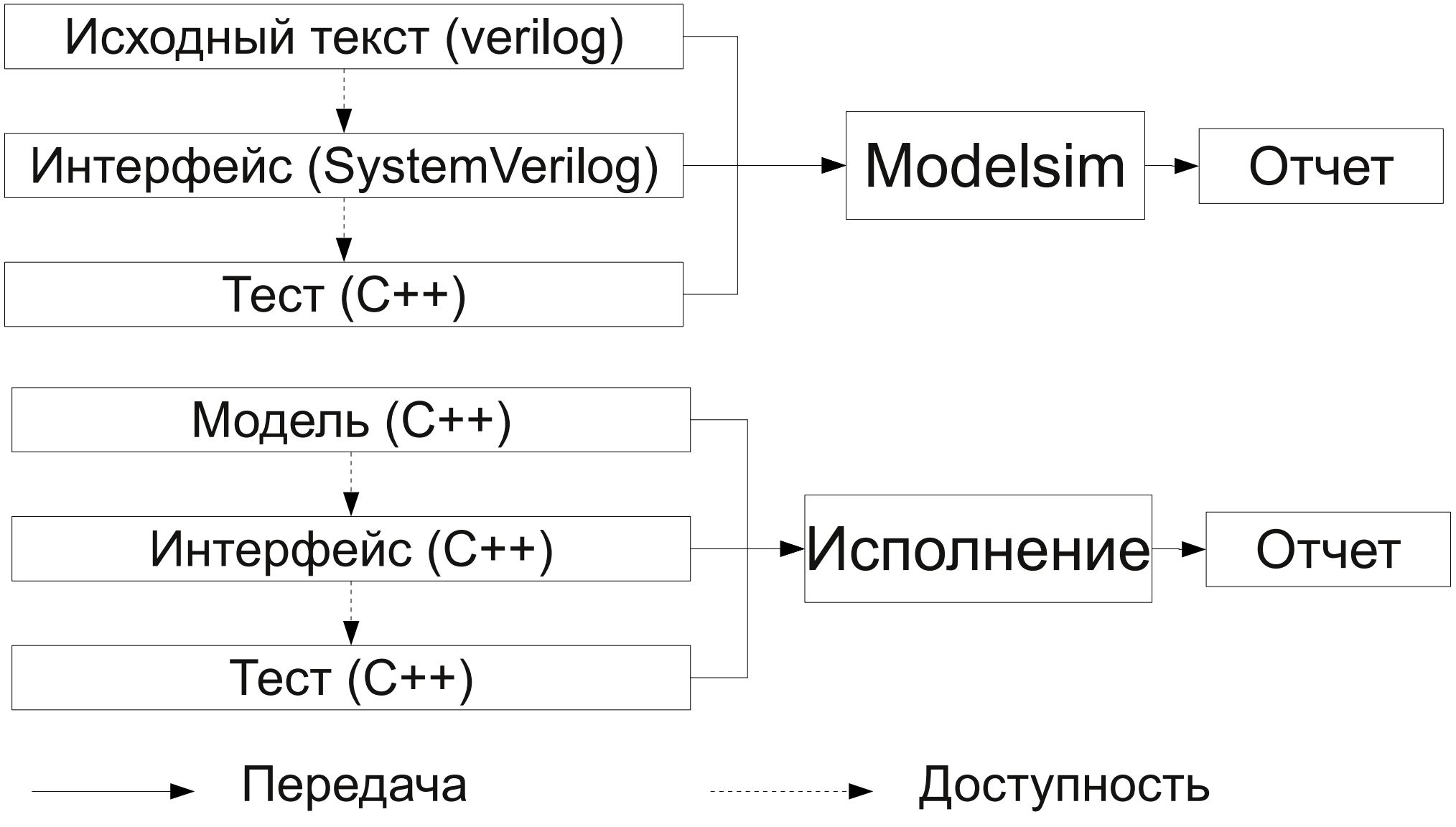
Результаты работы

Создана управляемая покрытием система генерации тестов на примере модуля `data_box`

- Обеспечена работа транслированной в C++ модели модуля `data_box`, в т.ч. в синтезированной в вентильный verilog (gate verilog)
- Обеспечена работа существующего генератора псевдослучайных тестов для данного модуля
- Реализован сбор данных о покрытии во время генерации теста
- Реализовано быстрое сохранение и восстановление контрольных точек
- Реализован механизм управляемой покрытием генерации тестовых воздействий

Спасибо за внимание

Адаптация существующих тестов



Тестовое покрытие

- Является измеримым критерием качества верификации и готовности продукта к выпуску
- Величина покрытия зависит выбора метрики
- Полное покрытие по большинству используемых метрик – необходимое, но не достаточное условие обеспечения качества верификации

Виды метрик тестового покрытия

- Основанные на частоте обнаружения ошибок
- Покрывающие элементы исходного текста
  - Строки
  - Выражения
  - Операторы
  - Ветвления
  - .....
- Покрывающие состояния данных
- Основанные на функциональном покрытии
- Покрывающие состояния и переходы конечных автоматов