

# МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ С ПАРАЛЛЕЛИЗМОМ ПОВЕДЕНИЯ НА УРОВНЕ ТРАНЗАКЦИЙ

Черемисинов Д. И., Черемисинова Л. Д.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: {cher, cld}@newman.bas-net.by

*Описывается объектно-ориентированный подход к моделированию описаний устройств управления с параллелизмом поведения на уровне транзакций. Предлагается метод преобразования описания алгоритма поведения системы управления в модель TLM (Transaction-Level Modeling) уровня транзакций.*

## ВВЕДЕНИЕ

Важным этапом проектирования управляющих систем является этап функциональной верификации, в процессе выполнения которого устанавливается, реализует ли спроектированное устройство желаемое поведение, т. е. работает ли оно согласно установленным в его спецификации требованиям. Самым распространенным подходом к верификации в настоящее время является имитационное моделирование, которое производится в рамках тестовой системы, позволяющей генерировать тестовые последовательности и производить проверку правильности поведения модели тестируемой системы.

Наиболее развитыми подходами и средствами разработки функциональных тестов в настоящее время являются технологии AVM (Advanced Verification Methodology) компании Mentor Graphics и OVM (Open Verification Methodology) компаний Mentor Graphics и Cadence Design Systems на основе языков SystemVerilog или SystemC. При разработке тестовых систем для отладки проектов аппаратуры в рамках этих технологий предполагается: объектно-ориентированный подход, моделирование на уровне транзакций (TLM) и направленная генерация псевдослучайных стимулов (constraint-random generation — генерация на основе ограничений).

В настоящей работе рассматривается задача верификации систем, характеризующихся параллелизмом поведения и непрерывным (и в общем случае бесконечным) обменом сигналами с внешней средой в процессе их функционирования. Спецификация на проектирование систем с параллелизмом поведения задается на языке ПРАЛУ описания параллельных алгоритмов управления. Предлагается модель TLM для описаний устройств с параллелизмом поведения и метод преобразования описания алгоритма на языке ПРАЛУ в модель TLM.

## I. МОДЕЛИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ НА УРОВНЕ ТРАНЗАКЦИЙ

Наиболее популярным языком моделирования уровня TLM является SystemC (стандарт

IEEE 1666), представляющий расширение языка C++. Исполняемая программа, получаемая в результате компиляции модели на языке SystemC, реализует симулятор с интегрированными средствами управления имитацией.

Модель верифицируемого проекта аппаратуры описывается набором последовательных процессов — потоков, которые выполняются одновременно (параллельно). Атомарные операции процессов задаются транзакциями, работающими на общей памяти. Транзакции линеаризуются в процессе их выполнения таким образом, чтобы обеспечивать детерминизм результата выполнения операций над общими данными несколькими параллельными процессами. Синхронизация процессов модели TLM на уровне транзакций осуществляется в SystemC барьерным механизмом. Барьерами являются точки исходного кода, в которых каждый процесс должен приостановиться и подождать достижения барьера всеми параллельными процессами выполняемой группы потока. Изменения значений, запланированные в процессах группы, в общей структуре данных происходят мгновенно после достижения барьера всеми процессами.

## II. TLM МОДЕЛЬ ОПИСАНИЙ СИСТЕМ НА ЯЗЫКЕ ПРАЛУ

Алгоритм управления на ПРАЛУ представляется неупорядоченной совокупностью линейных цепочек  $l_i$  операций языка, каждое из которых открывается метками из множества  $\mu_i$  и заканчивается метками из  $\nu_i$  перехода:  $\mu_i: l_i \rightarrow \nu_i$ . Основными в языке являются операции ожидания  $-k^{in}$  и действия  $> k^{out}$  над двоичными переменными. Первая сводится к ожиданию момента времени, когда конъюнкция  $k^{in}$  примет значение 1, вторая — к присвоению переменным из конъюнкции  $k^{out}$  значений, обращающих ее в 1. Порядок выполнения цепочек алгоритма управления в процессе его реализации определяется множеством запуска, его текущие значения  $N_t$  задают метки тех цепочек, которые могут выполняться одновременно. Если для цепочки  $\mu_i: l_i \rightarrow \nu_i$  выполняется условие  $\mu_i \subseteq N_t$  и реализуется событие

$k^{in}$ , с ожидания которого начинается цепочка  $l_i$ , то она запускается.

В работе используется модель параллелизма операций типа «чередование», в которой одновременность понимается как возможность упорядочивать операции произвольным образом. В такой интерпретации алгоритм на ПРАЛУ обладает свойством линеаризуемости, т.е. результат параллельного выполнения операций алгоритма эквивалентен некоторому последовательному выполнению атомарных операций.

Ключевые понятия TLM модели для описаний на языке ПРАЛУ состоят в следующем:

1. Структурой данных является вектор значений переменных: внешних (входных (условных) и выходных) и внутренних. Каждая компонента вектора задает пару значений одной переменной: текущего и планируемого.

2. Транзакции представляются операциями алгоритма, которые представляются в виде композиций некоторых элементарных операций, выполняемых последовательно. Реализация операции  $-k^{in}$  состоит в выполнении операции приостановки процесса и проверки текущих значений переменных, указанных в конъюнкции  $k^{in}$ . Реализация операции  $> k^{out}$  заключается в становке планируемых значений переменных, указанных в конъюнкции  $k^{out}$ .

3. Цепочки операций языка ПРАЛУ интерпретируются как процессы модели TLM. Вводится понятие ветви, которая является динамическим объектом и задает последовательный процесс, начинающийся с некоторой операции. Образование ветви заключается в занесении первой ее операции в очередь ветвей, готовых для выполнения (ОГ). Прекращение ветви состоит в переносе ее из ОГ в очередь ветвей, ждущих выполнения (ОЖ). Введенные приведенным способом ветви аналогичны потокам в модели TLM на языке SystemC.

4. При моделировании алгоритма управления из ОГ последовательно извлекаются ветви и выполняются до приостановки. Выполнение ветви приостанавливается, когда ее начальный фрагмент  $-k^{in}$  не может выполняться на множестве текущих значений переменных и тогда ветвь переносится в ОЖ. Иначе после ее выполнения в ОЖ вносится ветвь, начинающаяся с операции, которая должна выполняться в этой ветви следующей.

5. Синхронизация параллельно выполняемых ветвей осуществляется с помощью барьерного механизма, который фиксирует такты моделирования и изменения значений переменных. Точки барьера задаются операциями приостановки выполнения всех ветвей. Барьер достигнут, когда ОГ становится пустой, тогда ветви из ОЖ переносятся в ОГ; выгружаются значения выходных переменных; накопленные планируемые значения

пересылаются в текущие; вводятся (если система незамкнута) новые значения входных переменных и запускается выполнение первой ветви из ОГ.

### III. РЕАЛИЗАЦИЯ СИМУЛЯТОРА TLM МОДЕЛЕЙ НА ЯЗЫКЕ ПРАЛУ

Преобразование описания алгоритма на языке ПРАЛУ в модель TLM осуществляется путем представления операций языка в виде композиций предлагаемых элементарных операций, которые выполняются строго последовательно и составляет базис алгоритмического разложения параллельных алгоритмов на языке ПРАЛУ. Набор включает операции приостановки ветви и алгоритма; перехода и запуска заданного множества ветвей; прекращения процесса выполнения ветви по условию; проверки условных переменных; установки выходных переменных; установки и сброса переменных синхронизации; установки и сброса переменных маски.

Основопологающим моментом при моделировании TLM модели алгоритмов на языке ПРАЛУ является соглашение о длительности выполнения операций языка, в частности операции действия. Это существенно влияет на степень соответствия изменений сигналов, производимых симулятором и появляющихся на выходах программной или схемной реализации алгоритма. В настоящей версии программы симулятора принята нулевая длительность операций. В этом случае вычисления по одной ветви выполняются до тех пор, пока для их продолжения не потребуется изменение состояний условных переменных, т.е. пока операция ожидания может выполняться на множестве текущих значений переменных. Достижение барьера при моделировании алгоритма фиксирует такты работы устройства, а полученные изменения значений выходных переменных (отмеченные в векторе планируемых значений) соответствуют изменениям значений сигналов на выходах схемной реализации алгоритма управления при подаче на ее входы значений сигналов, соответствующих значениям в векторе текущих значений.

### IV. ЗАКЛЮЧЕНИЕ

Предложена модель TLM для моделирования описаний систем управления на языке ПРАЛУ, а также метод преобразования этих описаний в программу на промежуточном языке, которая выполняется строго последовательно. Разработаны трансляторы этой программы на языки Verilog и C, результаты компиляции которых на машинный язык представляют собой программы имитационного моделирования заданной системы управления. Исходными данными для программ имитационного моделирования служат тестовые последовательности.