ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ ПЕРЕХОДОВ КОНЕЧНЫХ АВТОМАТОВ ПРИ ПОМОЩИ ЯЗЫКА SYSTEMVERILOG

Пшеничный К.Ю.
Научный руководитель – к.т.н., доц. Хаханова А.В.
Харьковский национальный университет радиоэлектроники
(61166, Харьков, пр. Науки,14, каф. АПВТ, тел. (057) 702-13-26)
e-mail: anna.hahanova@nure.ua, факс (057) 702-13-26

The given work is devoted to finite state machines transition path coverage using functional verification capabilities of SystemVerilog hardware description language. Cover directives method for transition path coverage has been proposed in this paper.

Современные методы верификации цифровых систем, спроектированных при помощи языков описания аппаратуры (HDL), включают в себя генерацию случайных ограниченных тестовых наборов и использование средств анализа полноты покрытий для оценки результатов тестирования. Конечный цифровой автомат (FSM) имеет дискретное число состояний, определенное число условий, которые возбуждают переходы между состояниями, и функции выходов. Несколько условий могут возбуждать один и тот же переход. Покрытие перехода между двумя состояниями при помощи определенного тестового набора показывает, что данный тест учел лишь определенный переход между состояниями, но не учел условие перехода. Покрытие путей предоставляет комплексный анализ, который учитывает условия переходов между состояниями во время верификации. Путь – переход между двумя состояниями под определенным условием. Между двумя состояниями может быть множество путей. Цель исследования – повышение качества тестирования конечных цифровых автоматов за счет использования методов функциональной верификации для анализа полноты покрытий переходов автомата. Задача – разработка функциональной модели анализа полноты покрытий переходов при помощи средств языка SystemVerilog.

Использование конструкции свойств (property) языка SystemVerilog позволяет выразить возможные пути между двумя состояниями для последующей верификации их полноты во время моделирования.

```
property STANDBY_SLEEP_CMD5 ;
   @(posedge clk)
   ((state == STANDBY) |=> ((state == SLEEP) && (cmd == 5) ) ) ;
endproperty
```

Рисунок1.1 – Свойство, описывающее путь из состояния STANDBY в состояние SLEEP

Путь можно выразить вектором <q_0, q_1, r>, где q_0 – начальное состояние пути, q_1 –конечное состояние пути, r – условие перехода. На рис.1.1 представлено свойство, описывающее путь из состояния STANDBY в состояние SLEEP, который активируется, когда сигнал cmd принимает значение 5.

Данное свойство использует оператор импликации (|=>). Предшествующее(antecedent) выражение (слева от оператора) является простым нетемпоральным выражением, проверяющее текущее состояние автомата. Последующее (consequent) выражение (справа от оператора) описывает целевое состояние пути и его условие. Поскольку одно свойство описывает один путь между смежным состояниями, количество свойств, необходимых для описания всех путей, равно количеству путей между смежными состояниями.

Количество свойств будет иметь линейную зависимость, если между смежными состояниями существует всегда один путь, или экспоненциальную, если таковых путей более одного.

Далее необходимо определить верификационный метод для проверки данного свойства. Язык SystemVerilog имеет следующие верификационные директивы: assert, assume и cover[2-3]. Так как целью данного метода является покрытие, то используется директива cover.

```
ARC1: cover property (STANDBY_SLEEP_CMD5);
```

Рисунок 1.2 – Верификация свойства при помощи директивы cover

Научная новизна определяется новым методом тестирования конечных автоматов при помощи мониторинга полноты покрытий путей средствами функциональной верификации языка описания аппаратуры SystemVerilog. Использование данного метода в совокупности с тестирование при помощи случайных тестовых наборов (Constrained Random Testing) позволяет дать процентную оценку покрытия переходов во время верификации.

Список источников:

1. Janick Bergeron. Writing Testbenches: Functional verification of HDL models, 2nd edition. Springer. 2003. С. 80-120.

2. Foster D, Harry D. Assertion-Based Design, 2nd Edition. Springer. 2005. С. 90 – 150.

3. Ben Cohen, SystemVerilog Assertions Handbook, 2nd Edition. Springer. 2010. С.50 – 70.