

АВТОМАТИЗАЦИЯ ДИАГНОСТИРОВАНИЯ HDL-МОДЕЛЕЙ КОНЕЧНЫХ АВТОМАТОВ

Шкиль А. С.,

кандидат технических наук,
Харьковский национальный университет радиоэлектроники,
Харьков, Украина

Кулак Э. Н.,

кандидат технических наук,
Харьковский национальный университет радиоэлектроники,
Харьков, Украина

Гребенюк А. С.,

аспирант,
Харьковский национальный университет радиоэлектроники,
Харьков, Украина

AUTOMATION OF DIAGNOSTICS HDL-MODELS OF A FINITE MACHINES

Shkil A.S.,

Ph.D., Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine

Kulak E.N.,

Ph.D., Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine

Hrebenuk A.S.,

Ph.D. student, Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine

Аннотация.

В статье предложен подход к диагностированию HDL-моделей конечных автоматов с использованием программы ASFTEST в среде Active-HDL. Рассматривается вариант восстановления графа переходов по HDL-модели автомата в форме автоматного шаблона и анализ обхода всех дуг графа для поиска ошибок проектирования.

Ключевые слова: HDL-модель, ошибка проектирования, Active-HDL, автоматный шаблон, TestBench.

Abstract

The paper proposes an approach to the diagnostics of HDL-models of finite state machines using ASFTEST program in Active-HDL environment. Consider option restore the graph of transitions for HDL-model machine in the form of an automaton pattern analysis and bypass all the arcs of the graph to find design errors.

Keywords: hdl-model, design error, active-hdl, automaton pattern, testbench

Верификация HDL-моделей – это один из важных и наиболее затратных по времени этапов автоматизированного проектирования современных цифровых устройств. В процессе верификации идет проверка соответствия полученного кода изначальной спецификации. В данной работе рассматривается верификация HDL-модели конечных автоматов, представленных в виде двухпроцессного автоматного шаблона путем проведения диагностического эксперимента (ДЭ) по обходу всех дуг графа переходов, начиная с начальной вершины.

При проведении верификации HDL-моделей цифровых автоматов тестировщик достаточно часто получает HDL-код «неизвестного происхождения» и спецификацию в словесной (табличной) форме. Строить по спецификации граф, по графу тест (по стратегии обхода всех дуг графа) не имеет смысла, т.к. полученный таким образом тест не га-

рантирует нахождение всех ошибок проектирования, если автомат не принадлежит к «исключительному классу» автоматов.

С другой стороны в современном САПР существуют инструментальные средства восстановления по HDL-коду автоматного шаблона графа переходов и последующего построения теста для данного графа в автоматическом режиме. Если в HDL-коде нет ошибок, то результат моделирования совпадает с эталоном (спецификацией), если не совпадает – то фиксируется наличием ошибки проектирования.

Автоматизация верификации цифровых устройств - гораздо более удобный подход к диагностированию HDL-моделей, особенно в случае большого количества состояний и переходов, а также если имеется код неизвестного происхождения или с подключением различных IP-core. Он позволяет использовать несколько стратегий, совмещая их, или выборочно.

Рассмотрим вариант автоматизации диагностирования с использованием Active-HDL и встроенного инструмента ASFTEST. В качестве исходных данных - HDL-код в виде автоматного шаблона (специальная структура HDL -модели, в которой функции переходов и выходов выделены в отдельные процессы, а назначение нового состояния осуществляется в специальном процессе, связанном с синхронизацией) и спецификация в произвольном виде (путем задания функции выходов). По HDL-коду, с помощью Code2Graphics (в составе Active-HDL), восстанавливается граф переходов, соответствующий коду (с возможными ошибками). Восстановленный граф, представленный в формате ASF, далее используется в качестве входной информации для ASFTEST, которая на его основе восстанавливает HDL-код автомата (что в данном случае неприципиально), создаст файл с информацией о покрытии состояний и путей графа и сгенерирует тестовые наборы в форме TestBench. Запуск ASFTEST осуществляется с помощью командной строки, при помощи параметров которой возможна генерация теста по 3 различным стратегиям, в зависимости от целей моделирования и необходимой полноты теста. ASFTEST также выполняет анализ модели на корректность, генерирует макрокоманды для среды Active-HDL, фиксирует в файле отчета статистическую информацию.

При компиляции *.asf-файла генерируется модель автомата на языке VHDL, но при данном подходе к проведению ДЭ не используется.

В результате работы программы генерируются следующие файлы:

fsm_asft.cov – файл отчета, содержит информацию о покрытии состояний и путей при каждой стратегии, что в дальнейшем значительно упрощает создание таблицы путей

fsm_asft.err – файл отчета об ошибках.

$$D = \bigcap_{v_j=1} M_j - \bigcup_{v_j=0} M_j$$

где M_j – j -я строка матрицы маршрутов обхода графа переходов автомата и ВЭП.

Формула позволяет найти неисправную дугу в графе переходов автомата.

Ошибка в HDL-коде, вероятнее всего, находится во фрагменте кода автоматного шаблона, связанного с вершиной (состоянием) откуда исходит ошибочная дуга. Для нахождения места возникновения ошибки проектирования необходимо возвратиться к HDL-коду автоматного шаблона и выполнить визуальное инспектирование участка кода, реализующего ошибочный фрагмент маршрута обхода графа, вычисленного по формуле выше. После нахождения ошибочного оператора или его фрагмента он корректируется и диагностический эксперимент по проверке корректности HDL-кода повторяется.

fsm_asft.rpt – файл статистики, содержит информацию об обрабатываемом графе.

fsm_gen.vhd – файл описания автомата на языке VHDL.

fsm_tb2.vhd – файл, содержащий сгенерированную по стратегии 2 (обход всех дуг графа переходов автомата) тестовую последовательность на языке VHDL.

fsm_tb2.do – файл с макрокомандами для моделирования теста из файла fsm_tb2.vhd

В результате формируются тестовые последовательности в виде файла на языке VHDL для проверки правильности и верификации проекта с помощью программ моделирования на языках VHDL и Verilog.

В качестве элементарной проверки P_i при проведении ДЭ используется реализация определенного маршрута обхода графа, при этом номер маршрута соответствует номеру элементарной про-

верки. Результат элементарной проверки V_i считается отрицательным (0), если терминальная вершина на этом маршруте достигнута, в противном случае результат элементарной проверки считается положительным.

Для реализации стратегии обхода всех дуг графа строится дерево решений (дерево обхода графа). Результатом проведения ДЭ по обходу графа является вектор экспериментальных прове-

рок (ВЭП) $V = (v_1, v_2, \dots, v_m)$, где m – число терминальных вершин дерева решений.

Место возникновения одиночной ошибки в маршруте обхода графа переходов находится по левой формуле, а кратной – по правой.

$$D = \bigcup_{v_j=1} M_j - \bigcup_{v_j=0} M_j$$

Результаты проведенных ДЭ подтвердили, что используемая стратегия позволяет находить область местонахождения ошибки проектирования в фрагментах HDL-кода для одиночных и кратных ошибок проектирования, то есть значительно сузить область визуального инспектирования кода.

Перечень использованных источников

1. Шкиль А.С. Автоматизация поиска ошибок проектирования в HDL-моделях конечных автоматов / А.С. Шкиль, Г.П.Фастовец, А.С.Сирокурова // АСУ и приборы автоматики. – 2014. – №168. – С. 43-52.

References

1. Shkil A.S. Search automation of design errors in the HDL-models of finite machines / A.S. Shkil, H.P.Phastovets, A.S.Sirokurova // ACS and automation devices. – 2014. – №168. – С. 43-52.