Шевченко Игорь Всеволодович
студент
ФГАОУ ВО «Национальный исследовательский
университет «Московский институт
электронной техники»
г. Москва

ИСПОЛЬЗОВАНИЕ СРЕДЫ MATLAB SIMULINK ДЛЯ ГЕНЕРАЦИИ HDL-КОДА УСТРОЙСТВА

Аннотация: *в данной работе представлена возможность использования высокоуровневых языков проектирования, в частности языка MATLAB, и применение MATLAB Simulink для разработки и реализации на ПЛИС спроектированного устройства посредством генерации HDL-кода устройства из математической модели.*

Ключевые слова: *математическое моделирование, программирование ПЛИС, цифровая обработка сигнала.*

MATLAB – это высокоуровневый язык и интерактивная среда для программирования, численных расчетов и визуализации результатов. С помощью MATLAB можно анализировать данные, разрабатывать алгоритмы, создавать модели и приложения.

Язык, инструментарий и встроенные математические функции позволяют вам исследовать различные подходы и получать решение быстрее, чем с использованием электронных таблиц или традиционных языков программирования, таких как C/C++ или Java.

Simulink – это графическая среда имитационного моделирования, позволяющая при помощи блок-диаграмм в виде направленных графов, строить динамические модели, включая дискретные, непрерывные и гибридные, нелинейные и разрывные системы.

С использованием блоков Simulink и блоков функций на языке MATLAB было проведено поэтапное проектирование блока декодера. Каждый из блоков

представляет собой систему, имеющую внутри себя уникальную структуру и выполняющий определенную функцию или этап декодирования. Иерархический подход к описанию модели позволяет быстро и эффективно ориентироваться в работе устройства, а также производить тестирование и коррекцию блоков с целью точной математической реализации устройства.

По окончанию разработки математического описания устройства, среда Matlab Simulink позволяет создать HDL код модели, в случае синтезируемости компонентов и блоков модели.

Генерация кода происходит в три этапа.

1. Выбор параметров генерации.

2. Проверка синтезируемости.

3. Генерация.

Для проведения всех трех этапов используются функциональные возможности среды MATLAB, где предоставлена возможность поэтапного отслеживания генерации кода и проведения функционального тестирования полученного описания.

После проверки и генерации кода, MATLAB возвращает отчет с предполагаемым количеством комбинационных цифровых устройств. Рис. 1 представлен отчет о проверке и генерации HDL кода в среде MATLAB Simulink. В случае возникновения ошибок и предупреждений, генерация кода приостанавливается и не может быть выполнена до их устранения.

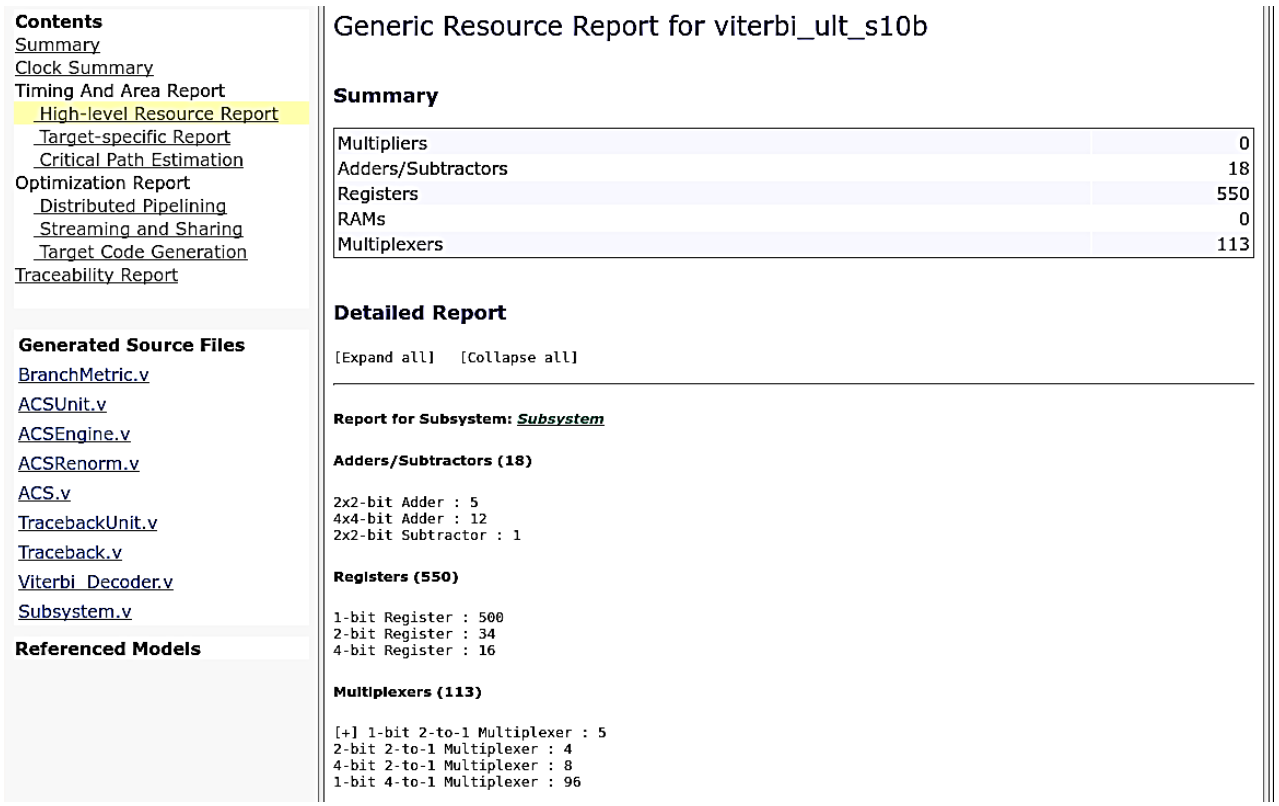

Рис. 1. Пример отчета HDL кодера среды MATLAB

Программирование на ПЛИС в работе осуществляется в среде Vivado Design Suite с использованием Verilog кода, полученного путем генерации из математического описания модели в MATLAB Simulink. Для правильно работы устройства необходимо, чтобы полученный из модели Verilog код функционально соответствовал математической модели, то есть работал точно так же и без ошибок. В следующей главе описан метод проверки сгенерированного HDL кода.

Для того чтобы проверить насколько точно полученное описание модели на языке Verilog соответствует математическому описанию в MATLAB Simulink, следует провести косимуляцию, то есть одновременное воспроизведение работы модуля и синтезированного RTL описание этого модуля. MATLAB имеет возможность такого взаимодействия с симулятором ModelSim компании Mentor Graphics.

На основе тестовой схемы и сгенерированного HDL кода модели, синтезатор составляет описание тестовой схемы (testbench) на языке Verilog и затем загружает полученные Verilog файлы в среду симулятор Modelsim.

Там, с учетом выставленных в MATLAB параметров, происходит синтез и вывод на временную диаграмму входных и выходных последовательностей исследуемого блока декодера. Одновременно, происходит моделирование и подсчет результатов работы декодера в среде MATLAB Simulink. По окончанию вычислений, происходит сравнение выходной значений выходов блоковой модели Simulink и синтезированной в ModelSim схемы.

Таким образом легко проверить, что математическая модель и сгенерированный код не имеют различий в работе.

Исходя из того, что HDL код был сгенерирован верно, то есть устройство функционирует правильно, можно произвести синтез и имплементацию кода в конкретное устройство.

В качестве среды разработки была выбрана среда Vivado Design Suite компании Xilinx. В качестве устройства была выбрана система на кристалле Zynq-7000.

Анализ синтеза и имплементации происходит из сравнения количества задействованных в ПЛИС логических ячеек LUT, а также выполнения требований по рабочей частоте устройства.

На рисунке 2 показан результат синтеза и имплементации устройства в виде занимаемого устройствами объема системы.

Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
FF	607	437200	0.14
LUT	263	218600	0.12
Memory LUT	4	70400	0.01
I/O	21	362	5.80
BUFG	1	32	3.12

Рис. 2. Результаты имплементации HDL кода

Из рисунка видно, что устройство занимает достаточно малое пространство на ПЛИС (не более 1%). Устройство можно проверить на возможность работы на частоте и убедиться в том, что устройство работает без задержек, отследив критические пути и оценив результаты синтеза.

Полученные результаты моделирования и синтеза показывают возможность эффективного использования среды MATLAB Simulink для сквозного проектирования устройств и получения HDL описания модели. Такой подход экономит время на разработку сложных систем и позволяет проводить тестирования устройства, в частности в области цифровой обработки сигнала.

Список литературы

1. Васильев В.В. Математическое и компьютерное моделирование процессов и систем в среде Matlab/Simulink / В.В. Васильев, Л.А. Симак, А.М. Рыбникова // Теория электрической связи: Учебное пособие. – Ульяновск: УлГТУ, 2008.

2. Электронный учебник по курсу «Компьютерная электроника» [Электронный ресурс]. – Режим доступа: http://khaer.com.ua/downloads/ke/rus/common/main.htm (дата обращения: 20.05.2016).

3. Thomas, Donald, and Philip Moorby. The Verilog® Hardware Description Language. – Springer Science & Business Media, 2008.