

ИСПОЛЬЗОВАНИЕ СКРИПТОВ ДЛЯ ЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ И СИНТЕЗА ЛОГИЧЕСКИХ СХЕМ

Бибило П. Н., Романов В. И.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: {bibilo, rom}@newman.bas-net.by

Предлагаются программные средства для формирования скриптов, позволяющих выполнять логическое моделирование VHDL-описаний комбинационной логики, формировать соответствующие системы булевых функций, проводить их логическую оптимизацию и синтезировать логические схемы в различных технологических базисах. Пользователю нужно лишь указать исходное алгоритмическое описание, способ логической оптимизации и целевую библиотеку логических элементов, используемую синтезатором LeonardoSpectrum. В результате выполнения скрипта он может выполнить синтез с использованием глобальной оптимизации и сравнить решение, полученное синтезатором LeonardoSpectrum без предварительной оптимизации.

ВВЕДЕНИЕ

Синтезаторы логических схем заменяют каждую конструкцию языка VHDL соответствующим функциональным описанием [1], после чего дальнейшей оптимизации подвергается комбинационная логика, представленная взаимосвязанными логическими выражениями. Такие выражения задают многоуровневые представления систем булевых функций, описывающих функциональные блоки, входящие в состав проекта цифровой схемы, синтезируемой в том или ином базисе логических элементов FPGA (Field-Programmable Gate Array) или ASIC (Application-Specific Integrated Circuits). Используемая логическая оптимизация является, по сути, локальной, т.е. оптимизации подвергаются части схемы (кластеры), выделяемые из оптимизируемого функционального описания проекта схемы. Глобальная оптимизация для достаточно больших проектов не выполняется, так как размерности оптимизационных задач огромны и достигают сотен входных и выходных переменных и сотен тысяч промежуточных логических переменных. В синтезаторе логических схем LeonardoSpectrum [1] размер кластера может быть установлен как управляющая опция синтеза.

Эксперименты [2] показали, что для функциональных описаний комбинационных схем предварительная глобальная оптимизация, выполняемая с помощью программ совместной минимизации систем булевых функций в классе ДНФ (дизъюнктивных нормальных форм) либо с помощью программ оптимизации многоуровневых BDD (Binary Decision Diagram) представлений на основе разложения Шеннона, может давать значительные выигрыши по площади схем и быстродействию по сравнению с результатами синтеза от исходных неоптимизированных функциональных описаний. Однако для применения программ глобальной оптимизации требуется переход от алгоритмических представлений функ-

ций комбинационных блоков ASIC к представлению функций в виде таблицы истинности либо системы ДНФ, которая может быть подвергнута оптимизации с помощью мощных программ, таких как например, ESPRESSO [3], которая позволяет выполнять совместную и раздельную минимизацию систем булевых функций в классе ДНФ по различным критериям либо программа BDD_Builder [4] минимизации многоуровневых BDD-представлений. Если же функции реализуемой системы являются неполностью определенными (частичными), то переход к двухуровневому (табличному) представлению позволяет проводить логическую оптимизацию с учетом возможности доопределения функций. Однако получение таких форм представления систем булевых функций не предусмотрено в синтезаторах, в частности в синтезаторе LeonardoSpectrum, который ориентирован на синтез не только схем FPGA, но и на синтез схем ASIC.

I. ПРЕДЛАГАЕМЫЙ ПОДХОД

Для исходных алгоритмических VHDL-описаний комбинационных блоков ASIC предлагается использовать логическое моделирование, например в системах ModelSim, Questa (ф. Mentor Graphics), для получения VHDL-описаний систем полностью определенных (либо частичных) булевых функций, задающих поведение этих блоков, и предварительную глобальную логическую оптимизацию с помощью программ, имеющихся в системе CMOSLD [5].

Построение эквивалентных описаний для любого алгоритмического VHDL-описания комбинационной логики, выполнение оптимизации и последующий синтез по оптимизированным описаниям осуществляются с помощью разработанной программы, которая автоматически генерирует соответствующий скрипт, пользователю нужно лишь указать исходное алгоритмическое VHDL-описание, способ оптимизации и целевую библиотеку логических элементов, ис-

пользуемую синтезаторе LeonardoSpectrum. В результате выполнения скрипта он может выполнить синтез с использованием глобальной оптимизации и сравнить решение, полученное синтезатором LeonardoSpectrum без предварительной оптимизации. Такой подход к синтезу эффективен для цепочек (конвейеров) арифметических операций с ограниченным числом входных полюсов.

II. ЭТАПЫ ФОРМИРОВАНИЯ СКРИПТА И ВЫПОЛНЕНИЕ ПРОЕКТНЫХ ПРОЦЕДУР

1. По исходному VHDL-описанию комбинационного блока определяется число n входных полюсов данного блока;
2. Генерируется таблица (текстовый файл IN.TST) всех 2^n элементов булева пространства, построенного над переменными булева вектора входных сигналов;
3. Генерируется тестирующая программа для выполнения моделирования исходного алгоритмического описания на всех 2^n наборах значений входных сигналов;
4. Генерируется скрипт для системы VHDL-моделирования, который позволяет выполнить моделирование исходного VHDL-описания комбинационного блока и построить для каждого набора соответствующую реакцию комбинационного блока – так получается текстовый файл OUT.TST реакций комбинационного блока, например:


```
set PRJ tstb
set PRJ_ARCH behavior
set DUT_NAME tstbp0
vlib ./vsim
vmap work ./vsim
vcom -f z:/BAT/vl6/files.f -source
vsim -novopt work.$PRJ +no_glitch_msg
run -a
quit -f
```
5. Вызывается программа формирования (по файлу IN.TST входных наборов и файлу OUT.TST реакций) SF-описания комбинационного блока в виде таблицы истинности;
6. Вызывается одна из программ глобальной логической оптимизации для SF-описания (ESPRESSO, BDD_Builder и др.), в результате получается минимизированное SF-описание системы булевых функций;
7. Вызывается программа конвертации минимизированного SF-описания в VHDL-описание;
8. Генерируется специальный скрипт для синтезатора LeonardoSpectrum, где указываются опции синтеза и требуемая библиотека логических элементов FPGA или ASIC, например:


```
clean_all;
set encoding Gray;
set modgen_select Smallest;
```

```
set asic_auto_dissolve_limit 500;
set auto_dissolve_limit 500;
read z:/BAT/vl6/optim.vhd;
load_library z:/_Mon/libs/s3lib.syn;
set -hierarchy flatten
set effort standard
optimize -target s3lib -macro -area
-effort standard -hierarchy flatten
report_area -cell_usage
report_delay -num_paths 1
-critical_paths
auto_write z:/BAT/vl6/result.vhd
```

9. Вызывается синтезатор LeonardoSpectrum, который выполняет синтез по подготовленному скрипту.

Запуск скрипта позволяет выполнить синтез схемы с предварительной оптимизацией системы булевых функций комбинационного блока ASIC. Возможно формирование двух видов скриптов – для полностью определенных функций и частичных. Ограничения следующие:

- Исходное VHDL-описание комбинационного блока должно содержать только информационные входные и выходные порты типа *std_logic*, *std_logic_vector*;
- Общее число входных полюсов комбинационного блока n не должно превышать 20.

III. ЗАКЛЮЧЕНИЕ

Замена (свертка) цепочек последовательно выполняемых арифметических операций соответствующими матричными моделями полностью определенных либо частичных булевых функций и оптимизация этих моделей является целесообразной при ограниченных размерах задач синтеза, так как позволяет использовать программы глобальной логической оптимизации и получать лучшие результаты при синтезе функциональных комбинационных блоков ASIC.

1. Бибило, П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum / П. Н. Бибило // – М.: СОЛОН-Пресс, 2005. – 384 с.
2. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов / – Минск: Беларус. наука, 2011. – 279 с.
3. Brayton, K. R. Logic minimization algorithm for VLSI synthesis / K. R. Brayton, G. D. Hachtel, C. T. McMullen, A. L. Sangiovanni-Vincentelli // Boston, e.a.: Kluwer Academic Publishers, 1984. – 193 p.
4. Бибило, П. Н. Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона / П. Н. Бибило, Ю. Ю. Ланкевич // Программная инженерия. – 2017. – № 3. – С. 369–384.
5. Бибило, П. Н. Система логического проектирования функциональных блоков заказных КМОП СВИС с пониженным энергопотреблением / П. Н. Бибило [и др.]. // Микроэлектроника. – 2018. – Т. 46. – № 1. – С. 72–88.