

Архитектура системы синтеза конвейерных логических схем

Бибилло П.Н.; Кириенко Н.А.; Романов В.И.
Объединенный институт проблем информатики НАН Беларуси
Минск, Республика Беларусь
e-mail: rom@newman.bas-net.by

Аннотация—Предлагается архитектура разрабатываемой системы синтеза конвейерных логических схем. Дается краткое описание основных компонент системы и входящих в их состав программных модулей.

Ключевые слова: конвейерные логические схемы; программные комплексы логического проектирования заказных СБИС

В настоящее время широко применяются промышленные системы синтеза логических схем по алгоритмическим описаниям их поведения на языках высокого уровня. Широко распространенным языком проектирования является язык VHDL, а используемым промышленным синтезатором - LeonardoSpectrum [1], который может быть настроен на пользовательскую библиотеку логических элементов. Достижение максимального быстродействия логической схемы из библиотечных элементов может быть обеспечено, если логическая схема функционирует по принципу конвейера, когда все сигналы из выходов одного уровня (блока) конвейера поступают одновременно на вход следующего блока конвейера. Для реализации такого подхода необходима разработка программного комплекса, позволяющего решать задачу синтеза конвейерных логических схем в отечественной элементной базе. Такой подход позволит реализовать схемы на максимально возможной частоте для заданной технологической библиотеки синтеза, в качестве которой выступает библиотека КМОП-элементов, используемая для проектирования заказных СБИС.

Основная задача системы синтеза конвейерных логических схем – это преобразование нерегулярных многоуровневых логических схем в конвейерные структуры путем установки регистров триггеров с общим сигналом синхронизации, как это показано на рис.1. В качестве подзадач выступают разбиение элементов схемы на блоки и определение функций и временных задержек каждого блока.

Архитектура разрабатываемой системы синтеза конвейерных логических схем представлена на рис.2. Система состоит из ряда компонент, каждый из которых предназначен для решения отдельной задачи из технологической цепочки построения конвейера: формирование проекта, построение конвейера тем или иным способом, оформление полученных результатов в требуемой форме.

Формирование рабочего проекта осуществляется на основании описания обрабатываемой схемы на языке VHDL. Система синтеза использует в качестве

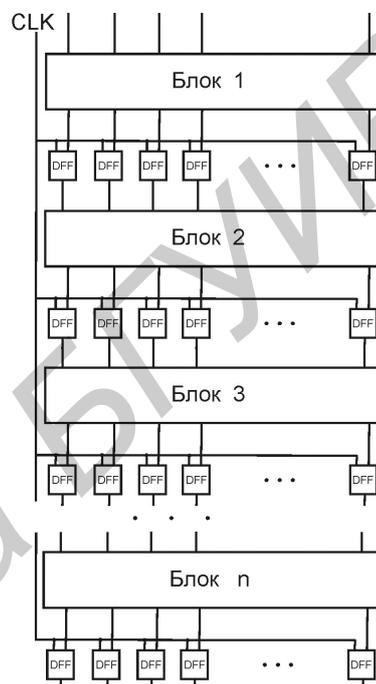


Рис. 1. Схема с конвейерной структурой

внутреннего языка, на котором проводятся преобразования конвейеризации, язык SF [2]. Для получения описания схемы в этом языке используется специальный конвертер, на выходе которого схема представляется в форме иерархического описания, в качестве «листьев» которого в исходном состоянии выступают библиотечные элементы.

В рамках компонента «Формирование проекта» осуществляется определение временных задержек для каждого библиотечного элемента, участвующего в описании конвейерируемой схемы.

Предполагается, что в составе системы синтеза конвейерных логических схем будет реализовано два метода построения конвейера – максимальной и блочной конвейеризации.

Метод максимальной конвейеризации предполагает ранжирование элементов схемы по каскадам (топологическую сортировку) и встраивание регистров триггеров между элементами каждой пары каскадов, а также перед входами и на выходах схемы. Под числом каскадов (уровней) схемы n понимается максимальное число элементов схемы на пути от входного полюса к выходному. На входы элементов, принадлежащих i -му каскаду, поступают выходные сигналы только с элементов j -х каскадов, где $0 < j < i$. Каскад является минимально возможным блоком схемы.

В рамках реализации этого метода задействованы программные модули OneOut и Cascades. Следует отметить, что в целях повышения быстродействия

схемы и нагрузочной способности элементов модуль OneOut выполняет следующее преобразование. Все элементы схемы, имеющие разветвление на выходе,

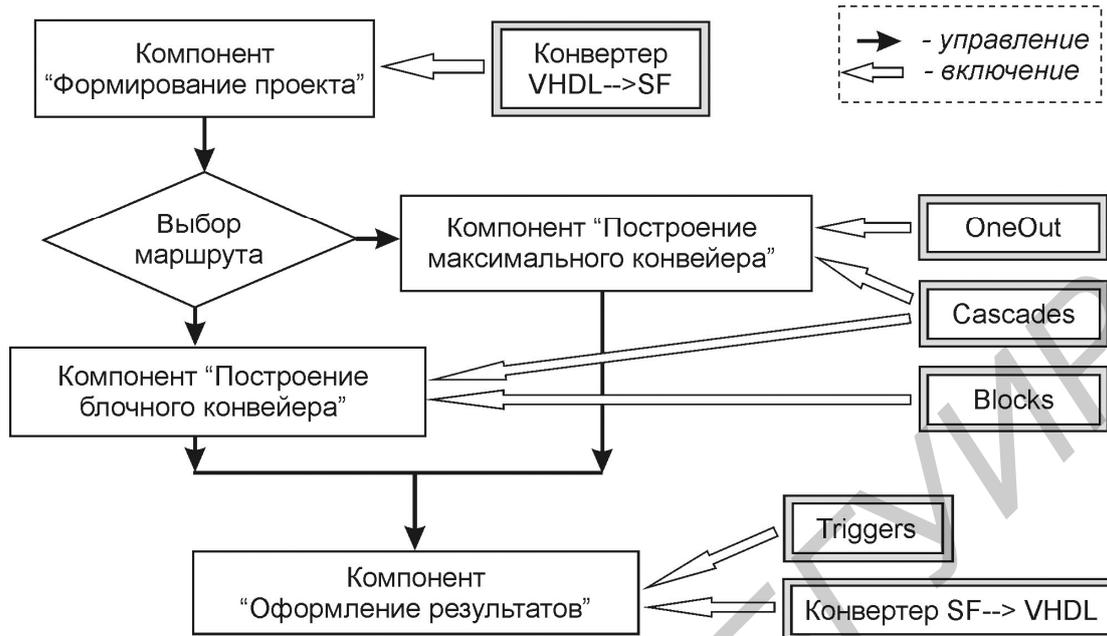


Рис. 2. Архитектура системы

дублируются столько раз, со сколькими элементами в схеме они связаны. Таким образом, мы получаем вариант схемы, в которой каждый элемент связан только с одним элементом на выходе. Далее, для полученного варианта схемы выполняется разбиение элементов на каскады. Информация о каскадах схемы далее может быть использована для встраивания между ними регистров триггеров.

Метод блочной конвейеризации предполагает разбиение элементов схемы на заданное количество блоков с учетом некоторых критериев. Между элементами соседних блоков вставляются триггерные регистры. Число блоков не может быть более числа каскадов в схеме. Как правило, блок объединяет несколько соседних каскадов. Элементы блоков также должны быть топологически отсортированы.

В рамках реализации этого метода задействованы программные модули OneOut, Cascades и Blocks. По сравнению с методом максимальной конвейеризации после получения разбиения элементов на каскады выполняется объединение каскадов в блоки с учетом заданных критериев. Информация о блоках схемы точно так же может быть использована для встраивания между ними регистров триггеров.

На завершающем этапе конвейеризации осуществляется построение описания схемы с триггерами в соответствии с найденными решениями каскадирования или блочного разбиения. Для этого используется программный модуль Triggers и специальный конвертер, преобразующий описание схемы с языка SF в язык VHDL.

Программный модуль Triggers в качестве исходных данных использует структурное описание схемы и информацию о разбиении ее элементов на каскады или блоки. В структуру схемы добавляется столько триггеров заданного типа, сколько имеется связей

между элементами соседних блоков, по одному триггеру на каждую связь.

Конвертер «SF → VHDL» осуществляет перевод только структурной части описания схемы, поскольку собственно компоненты доступны на языке VHDL в форме VHDL-библиотеки КМОП-элементов, используемой для проектирования заказных СБИС.

Отдельный сеанс работы системы синтеза конвейерных схем строится в форме последовательного предъявления пользователю цепочки заполняемых бланков-закладок, на каждом из которых представлены информационные поля и средства управления. Пример одной из закладок представлен на рис.3.

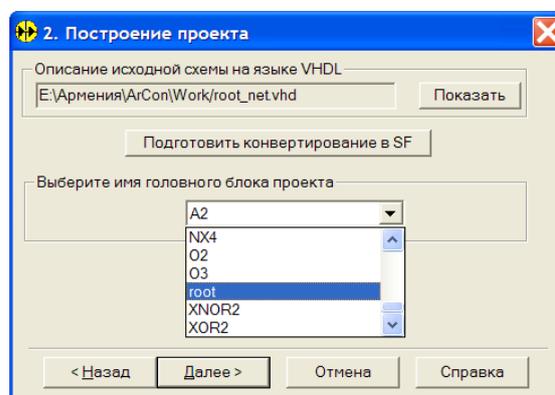


Рис. 3. Закладка одного из цепочки состояний сеанса

[1] Бибило П.Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. - М.: СОЛОН-Пресс, 2005. – 384 с.

[2] Бибило, П.Н. Кремниевая компиляция заказных СБИС / П.Н. Бибило – Минск: Ин-т техн. кибернетики АН Беларуси, 1996. – 268 с.