

ПРОЕКТИРОВАНИЕ СБИС

канд. техн. наук, доцент С.Г. Шматин, студентки гр.113122:
Н.А. Короткевич, Т.А. Перепелкина, А.А. Турыгина

Белорусский национальный технический университет

Проектирование специализированной СБИС (Application-Specific Integrated Circuits – ASIC) начинается с определения базовых функций ее составных частей. Эта стадия важна для выбора соответствующего стиля реализации проекта (design style).

По стилю проектирования и исполнения СБИС делятся на заказные (custom) и полузаказные (semi-custom) делекты (рис.1). Полностью заказные СБИС представляют собой полностью выполненный законченный проект, обеспечивающий максимальную производительность и низкую цену, но только при крупносерийном производстве. Стоимость же разработки и отладки очень высока. Кроме того, полностью заказные СБИС имеют самый большой срок разработки.

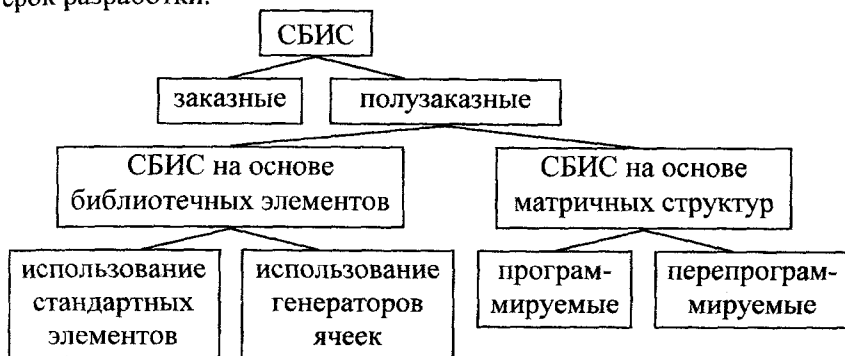


Рис.1. Стили исполнения СБИС

Полузаказные СБИС можно классифицировать на СБИС на основе библиотечных элементов (cell-based design) и БИС на основе матричных структур (array-based design) [1]. При проектировании СБИС на основе библиотечных элементов используют соответствующие библиотеки предварительно разведенных библиотечных компонентов (cells) или специализированные генераторы таких элементов (cell generators). Разработка на базе библиотечных элементов в свою очередь подразумевает либо использование стандартных элементов (standard-cell), либо использование генераторов ячеек (cell generators) для реализации примитивов.

В отличие от БИС на основе библиотечных элементов, полужаказные БИС на основе матричных структур представляют собой предварительно размещенные, но не соединенные базовые логические элементы, расположенные в виде матрицы. К таким БИС относятся соответственно базовые матричные кристаллы, масочные и лазерно программируемые ПЛИС (MPGA, LPGA), а также перепрограммируемые структуры ПЛИС (FPGA на основе технологий SRAM и antifuse).

Традиционно при проектировании специализированных БИС используется нисходящая модель маршрута проектирования (waterfall model). При такой организации маршрута проектирования проект проходит различные фазы, постоянно увеличивая детализацию представления. Нисходящее проектирование подразумевает минимальное взаимодействие между командами разработчиков на различных фазах проекта. Процесс проектирования начинается с разработки технических требований (specification), их последующего анализа, проведения предварительного моделирования с помощью специализированных пакетов или на языке высокого уровня (например, C).

По описанию на уровне RTL с помощью программы логического синтеза формируется список цепей (gate level net list), учитывающий задержки на библиотечных элементах (но, как правило, не учитывающий временные задержки на межсоединениях), который используется для временной верификации проекта (timing verification). Цель временного моделирования – проверить, удовлетворяет ли разрабатываемая БИС заданным временным ограничениям (timing constraints).

На основании данных синтеза топологи (physical design team) разрабатывают и оптимизируют разводку кристалла (floor plan), размещая библиотечные элементы и межсоединения оптимальным образом. После разработки топологии можно повторно выполнить формирование файла задержек и последующее временное моделирование, учитывающее влияние межсоединений. Затем кристалл можно передавать в производство и осуществлять последующее тестирование образцов.

Недостаток этой методологии проектирования: с увеличением сложности проекта увеличивается опасность появления ошибок и затрудняется процесс их поиска.

Переход от модели на функциональном или поведенческом уровне к описанию на уровне регистровых передач осуществляется либо вручную, написанием соответствующего кода на языке опи-

сания аппаратуры, либо с использованием специализированных средств синтеза высокого уровня.

Описание модели на уровне регистровых передач использует компоненты типа сумматоров, перемножителей, регистров, мультиплексоров и т.п., чтобы представить структуру проекта и его межсоединения. Описание на уровне RTL моделируется, как правило, выполняется событийное моделирование с целью верификации функциональности и основных временных характеристик. Верифицированная функциональная модель служит основой для синтеза на уровне логических вентилях.

Системный уровень описания проекта состоит из поведенческого описания в терминах функций, выражений, алгоритмов. На уровне регистровых передач проект представляется совокупностью арифметических и логических узлов, элементов памяти и т.п. Вентильный или логический уровень описывает проект на уровне логических вентилях и триггеров. В этом случае поведение схемы может быть описано системой логических уравнений. Эти логические элементы представляются на кремниевом (топологическом) уровне в виде топологических элементов и межсоединений.

Уровень регистровых передач включает компоненты и межсоединения между ними, для большего количества сложных систем может также включать типовые элементы типа ПЗУ, СБИС. Вентильный (логический) уровень соответствует представлению уровня логического элемента, и набор шаблонов топологических элементов кристалла соответствует геометрическому уровню.

Различные уровни представления проекта различаются типом информации, которую они отображают. Поэтому уровни представления могут быть классифицированы как поведенческий, структурный и физический. В поведенческом представлении описано только функциональное поведение системы и проект представляется как «черный ящик», имеющий зависимость выходного сигнала от входного. Структурное представление детализирует проект, вводя информацию относительно компонентов в системе и их взаимодействия. Детальные физические характеристики компонентов определены в физическом представлении, включая информацию о размещении и трассировке.

Так как поведенческое представление описывает проект в терминах блок-схем и алгоритмов, структурное представление представляет проект в терминах процессоров, блоков памяти и других логических блоков. Точно так же поведенческое представление на уровне межрегистровых пересылок представило бы поток межре-

гистровых пересылок набором поведенческих инструкций, а структурное представление представляет тот же самый поток набором компонентов и связей между ними. На логическом уровне схема может быть представлена булевыми уравнениями или конечными автоматами в поведенческом представлении либо как цепь связанных вентилей и триггеров в структурном представлении. Геометрический уровень представлен как транзисторные функции в поведенческом уровне, как микротранзисторы в структурном представлении, а также как топология, ячейки, кристаллы в физическом представлении.

Технические требования описывают требования к конечному изделию, функциональные возможности и другие требования типа температурного диапазона, потребляемой мощности, требований приемки пользователя и системного испытания. Это ведет к более определенным требованиям на устройство непосредственно в терминах функциональных возможностей, интерфейсов, рабочих режимов, условий эксплуатации, эффективности, отражаемых в техническом задании.

Традиционно для простых проектов ввод проекта выполняется после того, как проект архитектуры более высокого уровня будет закончен. Ввод проекта может быть в форме схемных решений блоков, которые реализуют выбранную архитектуру. Однако с увеличивающейся сложностью проектов соображения относительно системного моделирования и инструментальных средств проверки становятся преобладающими. Системные проектировщики хотят гарантировать, что ПК проектируют качественно и быстро создают рабочую аппаратную модель, моделируют ее взаимодействие с остальной частью системы, осуществляют синтез и формальную верификацию. В существующих методологиях проектирования специализированных интегральных схем, используемых в промышленности, языки описания аппаратуры обычно используются, чтобы описывать проекты на уровне межрегистровых пересылок. Однако в последнее время стала пользоваться популярностью методология «Описал – выполнил – долизал» (Specify-Explore-Refine – SER).

Использованные источники

1. http://www.compitech.ru/html.cgi/arhiv/03_04/stat_144.htm
Стешенко, В. Проектирование СБИС. Стили и этапы проекта // Компоненты и технологии. – 2003. – №4.