ИТМО
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Обзор маршрутов проектирования прикладного программного обеспечения для ПЛИС/ASIC/SoC на основе языков C/C++

Аспирант: Колесников Е.И.

Научный руководитель: д.т.н., профессор Шалыто А. А.

Кафедра компьютерных технологий
Санкт-Петербургского государственного университета
информационных технологий, механики и оптики

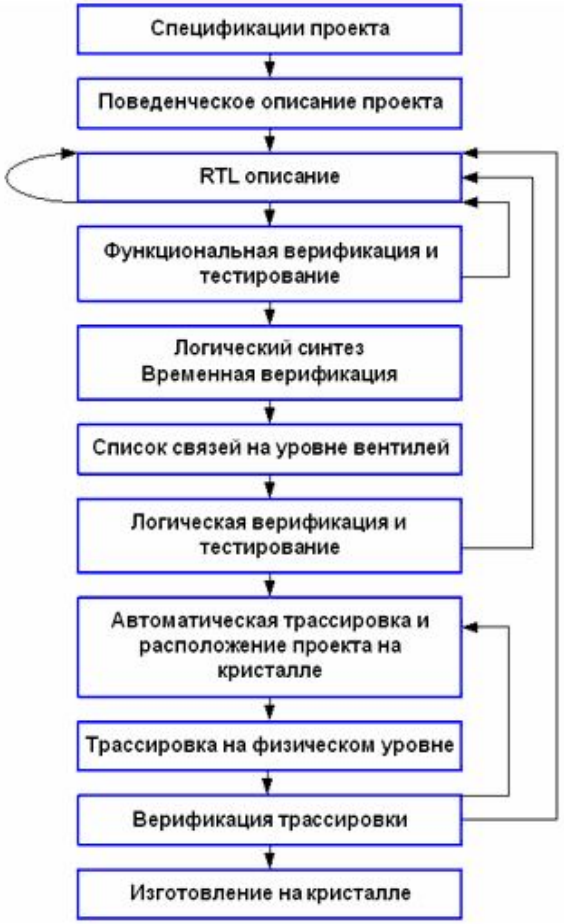
Задачи обзора

- Сравнить маршруты проектирования ПЛИС/ASIC/SoC и показать их преимущества и недостатки
- Рассмотреть и сравнить аппаратные версии языков C/C++ с языками описания аппаратуры – HDL-языками
- Показать целесообразность использования языков C/C++ в маршрутах проектирования ПО для ПЛИС/ASIC/SoC (в отличие от HDL-проектирования)
- Рассмотреть возможность использования автоматного программирования на системном уровне (ESL) проектирования ПЛИС/ASIC/SoC

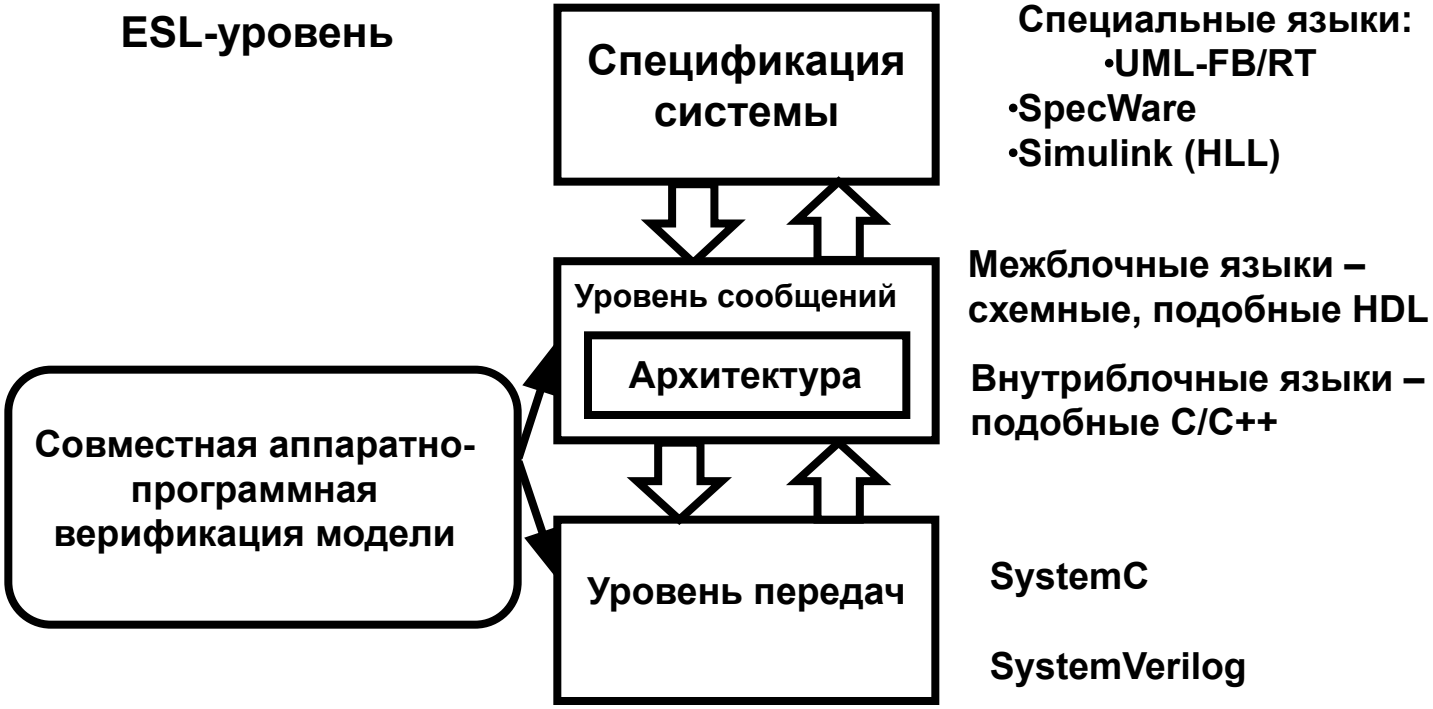
Маршруты проектирования ПЛИС/ASIC/SoC

- Традиционный нисходящий маршрут проектирования
- Маршрут с абстракцией проекта на системном уровне (ESL-проектирование или проектирование «сверху вниз»)

Традиционный маршрут проектирования



Маршрут проектирования с использованием общесистемного уровня



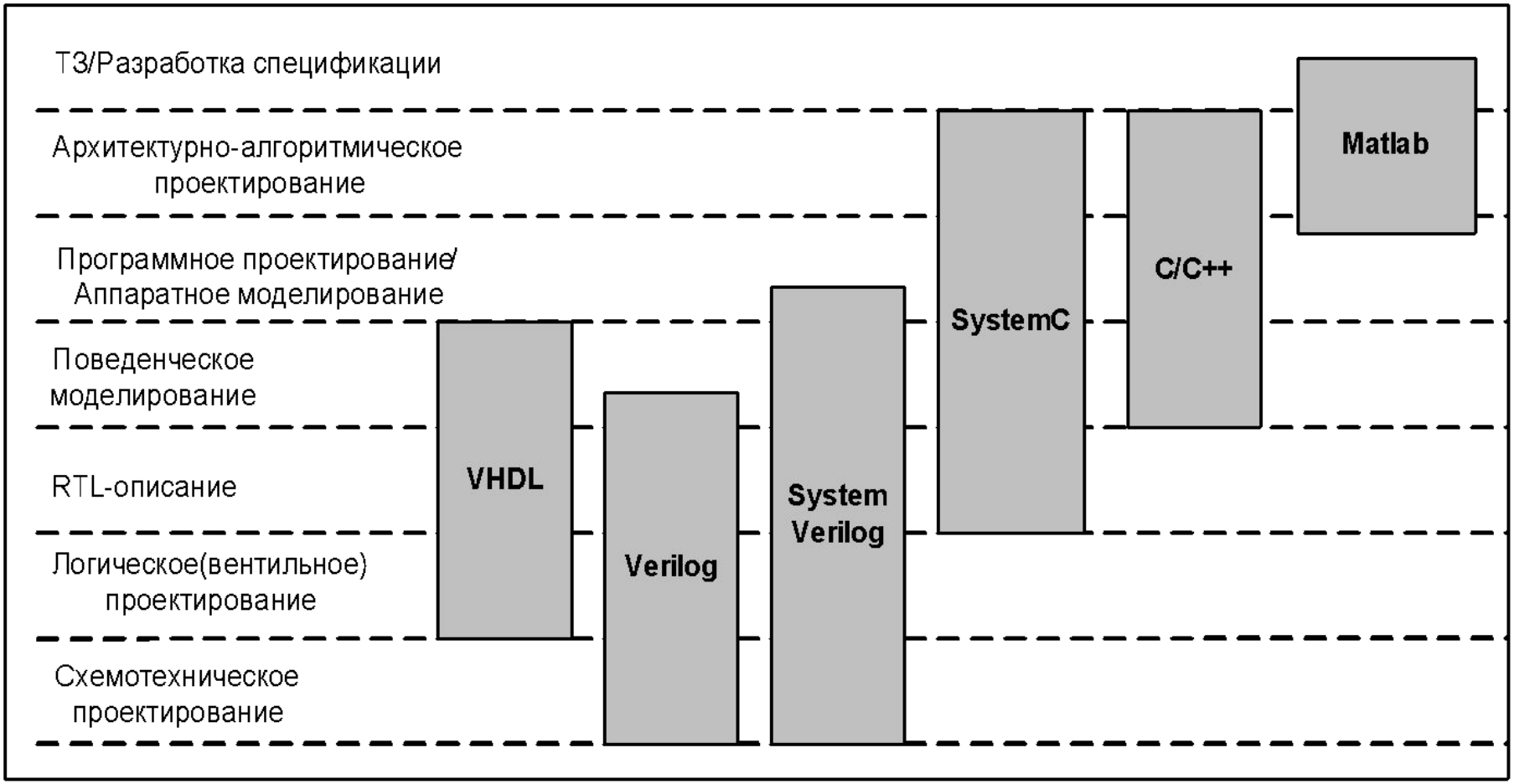
Преимущества автоматического синтеза RTL-описаний из алгоритмов на C/C++

- Устраняется разрыв между алгоритмическим описанием системы и RTL-описанием
- Основная нагрузка по отладке и верификации проекта переносится на более высокий уровень абстракции, а на этап синтеза поступает практически полностью отлаженный проект на системном уровне
- Анализ альтернативных вариантов реализции алгоритмов на C/C++ и внесение возможных изменений выполняется быстро и эффективно
- Описание проекта на C/C++ не привязано к выбору микроархитектуры и конкретной технологии реализации

Направления развития по использованию языков C/C++ для описания архитектуры

- Разработка библиотек на *C/C++*: *SystemC* и *Cynlib*
- Разработка специальных языков, основанных на синтаксисе *ANSI C* – *HandelC и SpecC*
- Разработка языка на базе *Verilog* и *C++* - *SystemVerilog*

Сравнение аппаратных языков C/C++ и HDL-языков по охвату стадий проектирования



Использование автоматного программирования на системном уровне проектирования ПЛИС/ASIC/SoC

Этапы проектирования с использованием автоматного программирования:

1) Использование *UniMod* для проектирования на системном уровне для ПЛИС/ASIC/SoC
2) Генерация *SystemC*-кода из *UniMod* модели
3) Дальнейшая конвертация *SystemC*-кода в *SystemVerilog* в специализированной САПР
4) Реализация на кристалле

Выводы

- Значительными преимуществами обладает маршрут проектирования на общесистемном уровне
- Перспективные аппаратные версии языков C/C++: SystemC и SystemVerilog. Но каждый из этих языков эффективен на своих определенных уровнях проектирования
- Отмеченные недостатки HDL-проектирования показывают целесообразность использования языков C/C++ в маршрутах проектирования ПО для ПЛИС/ASIC/SoC (в отличие от HDL-проектирования)

Заключение

- Внедрение автоматического синтеза ПЛИС/*ASIC*/*SoC* непосредственно из *C*-описаний может расширить круг разработчиков аппаратного обеспечения, повысить их производительность труда, сократив тем самым сроки и стоимость разработки.
- Применение маршрута проектирования с общесистемным уровнем и систем мультиязычного проектирования на основе языков *SystemC*–*SystemVerilog* обеспечивают создание высококачественной системы в кратчайшие сроки.

В дальнейшем планируется разработка методологии применения автоматного программирования на системном уровне (ESL) проектирования ПЛИС/ASIC/SoC.

Спасибо за внимание