

RSCD 2026

Суперкомпьютерные дни в России 2026



Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Учебный курс «Программирование для FPGA»

Евгений Козинов

`evgeny.kozinov@itmm.unn.ru`

Общая информация

❑ Название курса:

- Программирование для FPGA

❑ Направление подготовки / специальность:

- 02.03.02 - Фундаментальная информатика и информационные технологии

❑ Положение курса:

- Курс 4
- Семестр 8

❑ Количество часов:

- Лекции 12
- Практика 12



Цель разрабатываемого учебного курса

- Цели учебного курса «Программирование для FPGA»
 - **Сформировать целостное представление** о методах разработки, описания и тестирования цифровых устройств на FPGA с использованием SystemVerilog
 - **Освоить инструменты разработки и верификации:**
 - Quartus
 - ModelSim
 - Verilator
 - **Расширить представление о стеке разработки:**
 - разработать транслятор на базе ANTLR, преобразующий код с естественного языка в машинные инструкции



Целевая аудитория и требования

□ Целевая аудитория:

- Студенты старших курсов и аспиранты технических и естественно-научных специальностей
- Инженеры ИТ-компаний
- Преподаватели вузов и исследователи

□ Требования к слушателям:

- Базовые навыки разработки программ на языках C, C++, Python
- Знания и навыки в системном программировании, алгоритмах и структурах данных
- Знания основ архитектуры вычислительных систем

Отличительные особенности

- ❑ Курс разработан для направления ФИИТ
 - Акцент сделан на **системное программирование**, а не схемотехнику

- ❑ Лекции строятся на **практико-ориентированном подходе**:
 - Каждая лекция включает теорию и примеры решения задач.
 - Практические занятия закрепляют материал на задачах, приближенных к реальным.
 - Заключительные практики построены на сквозном примере с поэтапным расширением требований.
 - Такой формат практики демонстрирует полный цикл разработки



Инструменты и технологии

❑ В рамках курса используется:

Инструмент / Технология	Назначение
SystemVerilog	Язык описания и верификации аппаратуры
Intel Quartus	Среда проектирования для FPGA
Intel ModelSim	Средство тестирования интегральных схем, интегрированное с Intel Quartus
Verilator	Симулятор с открытым кодом Verilog/SystemVerilog (генератор C++ кода для тестирования)
ANTLR	Генератор парсеров, используется для разработки синтаксических анализаторов и инструментов обработки кода
C/C++	Тестирование интегральных схем и написание трансляторов.
Python	Взаимодействие с FPGA через UART



Структура курса

Тема	Содержание	Часы
Лекция 1. Введение в FPGA.	Содержание лекции: 1. Сравнение парадигм – C++, ASIC, FPGA. 2. NMOS и PMOS транзисторы. Принципы построения логических операций. 3. SystemVerilog – основы синтаксиса. 4. Комбинационная и последовательностная логика. 5. Quartus – основы работы в среде. Примеры проектов: • Комбинационная логика. Лампочка горит если оба выключателя включены. • Последовательностная логика. Мигание лампочки на основе делителя частоты.	2
Практика 1. Комбинационная и последовательностная логика.	1. Комбинационная логика. Реализация сумматора и вычитателя на основе логических операций. Значения считываются со свитчей. Вывод на светодиодах. 2. Последовательностная логика. Значения считываются со свитчей по нажатию кнопки. Вычисления выполняются по нажатию кнопки. Вывод на светодиодах.	2



Структура курса

Тема	Содержание	Часы
Лекция 2. Тестирование интегральных схем	Содержание лекции: 1. Реализация тестовых стендов на основе SystemVerilog. 2. Инструменты запуска тестов: ModelSim, Verilator. Примеры проектов: • Комбинационная логика. Реализации декодера для 7 сегментного индикатора. • Последовательностная логика. Разработка секундомера. Старт и стоп по нажатию кнопки.	2
Практика 2. Тестирование	Тестовые стенды на основе SystemVerilog для сумматора и вычитателя. Проверка корректности сумматора и вычитания в ModelSim и Verilator .	2



Структура курса

Тема	Содержание	Часы
Лекция 3. Конечные автоматы в интегральных схемах	Содержание лекции: 1. Конечные автоматы. Определение и примеры. 2. Примеры реализации конечных автоматов Мура и Миле. 3. Реализация тестов проверки конечного автомата в Verilator и ModelSim. Пример проектов: • Детектирование последовательности 1011 • Умножение 8 битных чисел. Оптимизация вычислений • Автоматы детектирования состояния кнопки • Двухфакторная аутентификация. Объединение нескольких конечных автоматов.	2
Практика 3. Реализация FSM управления светофорами	Реализация светофора на автомобильной дороге. Предполагается, что необходимо разработать устройство управления светофором на пешеходном переходе. Светофоры переключают цвет по таймеру. При необходимости пешеход может переключить светофор по нажатию кнопки.	2



Структура курса

Тема	Содержание	Часы
Лекция 4. Методы работы со статической памятью	Содержание лекции: 1.Реализация ROM и SRAM на SystemVerilog. 2.Чтение данных по адресу. Определение режима чтения/записи по регистру we. 3.Начальная инициализация памяти с использованием readmemb, readmemh. 4.Пример использования IP-блока однопортовой памяти. 5.Пример использования визуального редактора схем Quartus. Пример проекта: • Копирование данных из ROM в SRAM.	2
Практика 4. Вычисление арифметических выражений.	Реализация вычисления арифметических выражений в обратной польской записи. Предполагается, что выражения изначально хранятся в ROM. Для вычислений используется стек на основе SRAM. Для успешной сдачи проекта необходимо предусмотреть обработку ошибок и сигнализацию о состоянии светодиодами.	2



Структура курса

Тема	Содержание	Часы
Лекция 5. Реализация взаимодействия FPGA и PC на основе протокола UART.	Содержание лекции: 1. Основы протокола UART. 2. Реализация передачи данных через пользовательский модуль и IP блок. 3. Реализация Python скрипта чтения и записи данных на FPGA. Пример проекта: • Реализация отладочного модуля чтения/записи данных на FPGA по адресу SRAM.	2
Практика 5. Вычисление арифметических выражений с PC.	Реализация вычисления выражений в обратной польской записи. В дополнении к практике 4 польская запись арифметического выражения передается на интегральную схему через UART протокол.	2



Структура курса

Тема	Содержание	Часы
Лекция 6. Методы построения трансляторов.	Содержание лекции: 1.Основные этапы построения трансляторов. 2.Лексический и синтаксический анализ. 3.Грамматики. 4.Дерево разбора. 5.Алгоритмы построения дерева разбора. Нисходящий и восходящий разбор. 6.Метод построения транслятора с использованием ANTLR.	2
Практика 6. Построение транслятора арифметических выражений.	Реализация вычисления выражений в обратной польской записи. В дополнении к практике 5 реализуется транслятор арифметических выражений в обратную польскую запись в машинных кодах на основе ANTLR. Полученный набор машинных инструкций передается на интегральную схему через UART протокол.	2



Заключение

- ❑ FPGA – востребованная область при проектировании электронных устройств. Поэтому выпускники ИТ-специальностей должны иметь минимальный кругозор в данной области, обеспечивающий быстрый старт в решении реальных прикладных задач.
 - Курс «Программирование для FPGA» направлен на то, чтобы предоставить необходимые для этого базовые знания и навыки.
- ❑ Лекционные материалы и практические работы полностью подготовлены к прочтению
- ❑ Перспективы использования:
 - Спецкурс
 - Повышение квалификации
 - Молодежные школы для студентов и аспирантов



Спасибо за внимание

Евгений Козин

evgeny.kozinov@itmm.unn.ru

Работа выполнена при поддержке
Аналитического Центра при Правительстве Российской
Федерации (договор No 70-2025-000821 от 04.06.2025).

