

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 02.12.2024 г.

Рабочая программа дисциплины

Введение в программирование для программируемых логических
интегральных схем

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Инженерия программного обеспечения

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.08.04 Введение в программирование для программируемых логических интегральных схем относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-4: Способен проектировать программное обеспечение	<p>ПК-4.1: Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</p> <p>ПК-4.2: Знает методы и средства проектирования программного обеспечения</p> <p>ПК-4.4: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения</p>	<p>ПК-4.1: Знает методы использования стандартных IP блоков для реализации интегральных схем.</p> <p>ПК-4.2: Знает основы схемотехники для построения логических элементов. Понимает как из логических элементов реализовать вычислительные блоки. Знает основы SystemVerilog для описания интегральных схем.</p> <p>ПК-4.4: Умеет на основе IP блоков проектировать части интегральных схем. Умеет проектировать модули интегральных схем для решения прикладных задач. Умеет использовать SystemVerilog для описания разработанных модулей.</p>	Проект	Зачёт: Контрольные вопросы

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
--	-------

Общая трудоемкость, з.е.	3
Часов по учебному плану	108
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	12
- занятия семинарского типа (практические занятия / лабораторные работы)	12
- КСР	1
самостоятельная работа	83
Промежуточная аттестация	0 Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	
Л1. Введение в FPGA	6	2	0	2	4
П1. Комбинационная и последовательностная логика	12	0	2	2	10
Л2. Тестирование интегральных схем	6	2	0	2	4
П2. Тестирование интегральных схем	12	0	2	2	10
Л3. Конечные автоматы в интегральных схемах	6	2	0	2	4
П3. Реализация конечного автомата управления светофорами	12	0	2	2	10
Л4. Методы работы со статической памятью	6	2	0	2	4
П4. Вычисление арифметических выражений	12	0	2	2	10
Л5. Реализация взаимодействия FPGA и PC на основе протокола UART	6	2	0	2	4
П5. Вычисление арифметических выражений передаваемых с PC	12	0	2	2	10
Л6. Методы построения трансляторов	5	2	0	2	3
П6. Построение транслятора арифметических выражений для передачи данных на интегральную схему	12	0	2	2	10
Аттестация	0				
КСР	1			1	
Итого	108	12	12	25	83

Содержание разделов и тем дисциплины

Лекция 1 Введение в FPGA.

Сравнение парадигм – C++, ASIC, FPGA. NMOS и PMOS транзисторы. Принципы построения логических операций. SystemVerilog – основы синтаксиса. Комбинационная логика.

Последовательностная логика.

Примеры проектов:

- Комбинационная логика. Лампочка горит если оба выключателя включены.
- Последовательностная логика. Мигание лампочки на основе делителя частоты.

Quartus – основы работы в среде.

Практика 1. Комбинационная и последовательностная логика.

Реализация сумматора и вычитания на основе логических операций. Реализация вычислений с использованием комбинационной и последовательностной логики.

Лекция 2 Тестирование интегральных схем

Реализация тестовых стендов на основе SystemVerilog (testbench).

Примеры проектов:

- Комбинационная логика. Реализации декодера для 7 сегментного индикатора.
- Последовательностная логика. Разработка секундомера. Старт и стоп по нажатию кнопки.

Практика 2. Тестирование интегральных схем

Разработка тестовых стендов на основе SystemVerilog для сумматора и вычитания. Проверка корректности сумматора и вычитания в ModelSim и Verilator.

Лекция 3. Конечные автоматы в интегральных схемах

Конечные автоматы. Определение и примеры. Примеры реализации конечных автоматов Мура и Миле.

Реализация тестов проверки конечного автомата в Verilator и ModelSim.

Примеры проекта:

- Реализация конечного автомата определения состояния кнопки (отпущена, нажата, удерживается)
- Визуальный редактор конечных автоматов в Quartus.

Практика 3. Реализация конечного автомата управления светофорами

Реализация светофора на автомобильной дороге. Переключение по таймеру, а также кнопки у пешехода.

Лекция 4. Методы работы со статической памятью

Реализация ROM и SRAM на SystemVerilog. Чтение данных по адресу. Определение режима чтения/записи по регистру we. Начальная инициализация памяти с использованием readmemb, readmemh. Пример использования IP-блока одно портовой памяти.

Пример проекта:

- Копирование данных из ROM в SRAM.

Пример использования визуального редактора схем Quartus.

Практика 4. Вычисление арифметических выражений.

Реализация вычисления выражений в обратной польской записи. Выражение хранится в ROM. Стек реализуется на основе SRAM.

Лекция 5. Реализация взаимодействия FPGA и PC на основе протокола UART.

Основы протокола UART.

Реализация передачи данных через пользовательский модуль и IP блок.

Реализация Python скрипта чтения и записи данных на FPGA.

Пример проекта:

- Запись и чтение данных на FPGA по адресу SRAM.

Практика 5. Вычисление арифметических выражений передаваемых с РС.

Реализация вычисления выражений в обратной польской записи. Польская запись передается на интегральную схему через UART протокол.

Лекция 6. Методы построения трансляторов.

Основы построения трансляторов. Грамматики. Дерево разбора. Алгоритма построения дерева разбора. Нисходящий и восходящий разбор. Метод построения транслятора с использование ANTLR.

Практика 6. Построение транслятора арифметических выражений для передачи данных на интегральную схему.

Реализация транслятора арифметических выражений в обратно польскую запись в машинных кодах на основе ANTLR.

Кодирование и передача полученного выражения через UART.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

1) Дэвид М. Харрис и Сара Л. Харрис Цифровая схемотехника и архитектура компьютера // Morgan Kaufman © English Edition 2013. – 1662 с.

2) Фрэнк Бруно Программирование FPGA для начинающих / пер. с англ. С. Л. Плехановой; под науч. ред. А. Ю. Романова, Ю. В. Ревича. – М.: ДМК Пресс, 2022. – 304 с.

3) SystemVerilog Tutorial. URL: <https://www.asic-world.com/systemverilog/tutorial.html> (дата обращения 25.11.2023)

4) Pong P. Chu FPGA Prototyping by Verilog Examples: Xilinx Spartan-3 Version // Published by John Wiley & Sons, Inc., Hoboken, New Jersey. 2008. - 521 p.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Проект) для оценки сформированности компетенции ПК-4:

Пример проекта.

Калькулятор арифметических выражений в обратной польской записи на SystemVerilog
Цель проекта: Разработать цифровое устройство на языке SystemVerilog, выполняющее вычисления арифметических выражений, записанных в обратной польской записи (ОПН). Устройство должно поддерживать два режима работы:

Основной режим — чтение выражения из внешней SRAM, пошаговая обработка с тактовой частотой 1 герц, отображение хода вычислений на семисегментных индикаторах и светодиодах.

Дополнительный режим — приём выражения через универсальный асинхронный приёмопередатчик (UART) в машинных кодах, выполнение вычислений и передача хода операций и результата в текстовом виде обратно по UART.

Формат данных

Каждый элемент выражения передаётся как один байт со следующей структурой:

Бит 7 (восьмой бит) — тип данных:

0 — операнд (число),

1 — операция.

Биты 0–6 — значение:

Если тип операнд — 7-битное целое число со знаком в дополнительном коде, диапазон: от -64 до +63 или положительные на Ваше усмотрение.

Если тип операция — код операции:

+ → 1,

- → 2,

* → 3,

/ → 4.

Выражение завершается байтом-маркером: 11111111 (все биты — единицы).

Критерии оценивания (оценочное средство - Проект)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Программа и результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
		не зачтено		зачтено			
<u>Знания</u>	Отсутствие знаний	Уровень знаний ниже	Минимально	Уровень знаний в	Уровень знаний в	Уровень знаний в	Уровень знаний в

	теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	минимальных требований. Имели место грубые ошибки	допустимый уровень знаний. Допущено много негрубых ошибок	объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	объеме, соответствующем программе подготовки. Ошибок нет.	объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»

не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. В чем принципиальное различие между исполнением кода, реализованного на C++ и описанием на SystemVerilog для FPGA?
2. Назовите основные парадигмы разработки цифровых устройств и их ключевые отличия.
3. Каковы главные преимущества и недостатки FPGA по сравнению с ASIC?
4. Как на основе NMOS и PMOS транзисторов можно спроектировать основные логические операции?
5. Дайте определение комбинационной логики. Какой признак является главным? Приведите примеры
6. Какой блок SystemVerilog используется для описания комбинационной логики? Приведите примеры.
7. Какой оператор присвоения (= или <=) используется в always_comb и почему?
8. Дайте определение последовательностной логики. Чем она принципиально отличается от комбинационной? Приведите примеры
9. Какой блок SystemVerilog используется для описания последовательностной логики? Приведите примеры.
10. Перечислите основные операторы SystemVerilog: арифметические, побитовые, логические, редуцирующие, сдвига.
11. Расскажите о методах начальной инициализации сигналов в SystemVerilog? В каких случаях применима секция initial.
12. Приведите пример описания делителя частоты в SystemVerilog. Как выбрать число бит в счетчике
13. Приведите пример описания конечного автомата в SystemVerilog.
14. Перечислите основные этапы создания проекта в Quartus.
15. Что такое RTL и для чего используется данное представление?
16. Для чего используется инструмент Pin Planner? Где взять номера пинов? Приведите примеры
17. Что такое тестовый стенд? Из каких основных компонентов он состоит?
18. В чем концептуальное отличие ModelSim от Verilator? Приведите примеры тестовых стендов
19. Как описать однопортовую SRAM память в SystemVerilog? Напишите пример кода для памяти размером 256x8 бит и примеры ее использования.

20. Опишите методы использования IP блока для работы с однопортовой памятью
21. Что такое UART и какие сигналы используются в этом интерфейсе? Опишите формат передачи одного байта.
22. Как реализовать UART-передатчик на SystemVerilog без IP блоков? Приведите пример реализации
23. Расскажите о методах инициализации SRAM через UART приемник на основе IP блоков.
24. Расскажите о методах разработки скриптов на Python для передачи данных через UART.
25. Для каких задач используется инструмент ANTLR? Приведите примеры
26. Приведите пример грамматики для разбора арифметического выражения.
27. Что такое дерево разбора? Какие алгоритмы построения существуют?
28. Опишите классическую структуру компилятора. Какие этапы проходит исходный код от текста до машинных инструкций?
29. Что из себя представляет стековая архитектура процессора? Чем она отличается от регистровой (например, RISC-V)? Почему для такой архитектуры проще писать транслятор
30. Расскажите о том, как можно представить программу с условиями и циклами в обратной польской записи

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов возможно с незначительными недочетами.
не зачтено	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Дэвид М. Цифровая схемотехника и архитектура компьютера : монография / Дэвид М. Харрис; Сара Л. Харрис. - Москва : ДМК-пресс, 2018. - 792 с. - ISBN 978-5-97060-570-7., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=772995&idb=0>.

Дополнительная литература:

1. Ушенина И. В. Проектирование цифровых устройств на ПЛИС : учебное пособие для вузов / Ушенина И. В. - 2-е изд., стер. - Санкт-Петербург : Лань, 2023. - 408 с. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-47049-5., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=863478&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.
4. Intel Quartus - средство разработки, используется свободно распространяемая версия.
5. ModelSim - средство тестирования - используется бесплатная версия с лицензией GPL
6. Verilator - средства тестирования - предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом (GNU LESSER GENERAL PUBLIC LICENSE)

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Козин Евгений Александрович, кандидат технических наук.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.