

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 27.08.2025

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
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.03.01 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-Опер_1: Осуществляет управление архитектурой изолированной (неинтегрированной) программной системы	<p>ПК-Опер_1.1: Выявляет и согласовывает требования к программной системе с точки зрения архитектуры</p> <p>ПК-Опер_1.2: Осуществляет выбор и моделирование архитектурного решения для реализации программной системы</p> <p>ПК-Опер_1.3: Разрабатывает разделы по архитектуре проектных и эксплуатационных документов программной системы</p> <p>ПК-Опер_1.4: Контролирует реализацию и испытания программной системы с точки зрения архитектуры</p> <p>ПК-Опер_1.5: Осуществляет сопровождение эксплуатации программной системы с точки зрения архитектуры</p>	<p>ПК-Опер_1.1:</p> <p>ПК-ОПЕР_1.1. У-1. Способен выявлять несоответствия требований заказчика к программной системе с точки зрения архитектуры</p> <p>ПК-ОПЕР_1.1. У-2. Способен описывать требования к программной системе с точки зрения архитектуры</p> <p>ПК-ОПЕР_1.1. У-3. Умеет проверять требования на соответствие архитектуре программной системы</p> <p>ПК-ОПЕР_1.1. У-4. Умеет выявлять требования к архитектуре программной системы путем проведения интервью с заинтересованными сторонами</p> <p>ПК-ОПЕР_1.1. У-5. Умеет формулировать архитектурные требования к программной системе</p> <p>ПК-ОПЕР_1.1. З-1. Знает методы управления требованиями</p> <p>ПК-ОПЕР_1.1. З-2. Знает методы моделирования архитектуры программной системы</p> <p>ПК-ОПЕР_1.1. З-3. Знает методы проектирования архитектуры программной</p>	Задания	Зачёт: Контрольные вопросы

		<p>системы</p> <p>ПК-Опер_1.2: ПК-ОПЕР_1.2. У-1. Способен выбрать оптимальное архитектурное решение с учетом особенностей программной системы и принципов её организации ПК-ОПЕР_1.2. У-2. Способен определить архитектуру системы, ее, бизнес-процессов, структуру данных и отдельных компонентов программной системы и методы их интеграции ПК-ОПЕР_1.2. У-3. Способен определить перечень элементов архитектуры, которые должны быть защищены от угроз безопасности информации, связанных с нарушением конфиденциальности, целостности и доступности ПК-ОПЕР_1.2. У-4. Способен моделировать архитектурное решение для изолированной программной системы. ПК-ОПЕР_1.2. У-5. Умеет проектировать бизнес-архитектуру программных систем с применением лучших практик, шаблонов и стилей архитектурного проектирования ПК-ОПЕР_1.2. У-6. Умеет проектировать архитектуру интегрированной программной системы с учетом устойчивости к воздействиям внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) на любую из подсистем и с использованием методов и шаблонов конструктивной (встроенной) безопасности ПК-ОПЕР_1.2. З-1. Знает методы моделирования архитектуры программных систем и критерии сравнения</p>		
--	--	--	--	--

		<p>архитектурных решений ПК-ОПЕР_1.2. 3-2. Знает протоколы взаимодействия программных систем. ПК-ОПЕР_1.2. 3-3. Знает нормативные правовые акты, организационно-распорядительные документы и методические рекомендации, определяющие требования к безопасности программного обеспечения ПК-ОПЕР_1.2. 3-4. Знает методики определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных ПК-ОПЕР_1.2. 3-5. Знает лучшие практики и шаблоны создания конструктивно-безопасных интегрированных информационных систем</p> <p>ПК-Опер_1.3: ПК-ОПЕР_1.3. У-1. Способен описывать технические и организационные меры, обеспечивающие сохранение и восстановление программного обеспечения ПК-ОПЕР_1.3. У-2. Умеет проектировать и моделировать архитектурные элементы программных систем и их взаимосвязи ПК-ОПЕР_1.3. У-3. Умеет формировать технические и организационные меры для защиты программной системы от несанкционированного доступа к элементам конфигурации ПК-ОПЕР_1.3. 3-1. Знает методы моделирования и технического описания архитектуры программных систем ПК-ОПЕР_1.3. 3-2. Знает нормативные правовые акты, организационно-распорядительные документы</p>		
--	--	--	--	--

		<p><i>и методические рекомендации, определяющие требования к безопасности программного обеспечения</i></p> <p><i>ПК-ОПЕР_1.3. 3-3. Знает методики определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных</i></p> <p><i>ПК-Опер_1.4:</i></p> <p><i>ПК-ОПЕР_1.4. У-1. Способен проверять соответствие реализации программной системы выбранному архитектурному решению</i></p> <p><i>ПК-ОПЕР_1.4. У-2. Способен проверять результаты испытаний программной системы на соответствие архитектуре и архитектурным решениям</i></p> <p><i>ПК-ОПЕР_1.4. У-3. Умеет проверять характеристики реализованной программной системы на соответствие архитектурным требованиям</i></p> <p><i>ПК-ОПЕР_1.4. У-4. Умеет формулировать рекомендации по изменению реализованной программной системы для обеспечения соответствия ее архитектурным требованиям</i></p> <p><i>ПК-ОПЕР_1.4. 3-1. Знает способы определения характеристик работающей программной системы</i></p> <p><i>ПК-ОПЕР_1.4. 3-2. Знает методы параметризации архитектуры программных систем</i></p> <p><i>ПК-ОПЕР_1.4. 3-3. Знает основы процесса управления изменениями программных систем</i></p> <p><i>ПК-Опер_1.5:</i></p> <p><i>ПК-ОПЕР_1.5. У-1. Способен проверять запросы на изменения программной системы на реализуемость с</i></p>		
--	--	--	--	--

		<p>точки зрения архитектуры программной системы</p> <p>ПК-ОПЕР_1.5. У-2. Способен согласовывать запросы на изменения программной системы с точки зрения архитектуры</p> <p>ПК-ОПЕР_1.5. У-3. Умеет взаимодействовать с авторами запросов на изменения программной системы для уточнения содержания запросов</p> <p>ПК-ОПЕР_1.5. У-4. Умеет выявлять несоответствия и изменять запросы на изменения программной системы для обеспечения их соответствия выбранной архитектуре</p> <p>ПК-ОПЕР_1.5. З-1. Знает основы процесса управления изменениями программных систем</p> <p>ПК-ОПЕР_1.5. З-2. Знает методы обеспечения устойчивости функционирования программной системы</p> <p>ПК-ОПЕР_1.5. З-3. Знает методы обеспечения надежности архитектуры программной системы</p>		
<p>ПК-ТОП_11: Способен разрабатывать, оптимизировать и отлаживать системное программное обеспечение</p>	<p>ПК-ТОП_11.1: Разрабатывает низкоуровневый код для встраиваемых систем и драйверов</p> <p>ПК-ТОП_11.2: Оптимизирует код под ограниченные ресурсы</p> <p>ПК-ТОП_11.3: Работает с ОС, загрузчиками и аппаратурой</p> <p>ПК-ТОП_11.4: Отлаживает системное ПО без ОС</p>	<p>ПК-ТОП_11.1: ПК-ТОП_11.1. З-1. Знает архитектуру современных процессоров, особенности работы с оборудованием процессора</p> <p>ПК-ТОП_11.1. З-2. Знает принципы взаимодействия ПО с аппаратурой</p> <p>ПК-ТОП_11.1. У-1. Умеет разрабатывать низкоуровневый код для встроенного программного обеспечения и драйверов</p> <p>ПК-ТОП_11.1. У-2. Умеет разрабатывать драйверы для аппаратных устройств</p> <p>ПК-ТОП_11.2:</p>	Задания	<p>Зачёт: Контрольные вопросы</p>

		<p>ПК-ТОП_11.2. 3-1. Знает методы оптимизации (кэш, память, тактовая частота процессора)</p> <p>ПК-ТОП_11.2. 3-2. Знает инструменты профилирования</p> <p>ПК-ТОП_11.5. 3-2. Знает архитектуру компиляторов, оптимизации для выбранных архитектур процессоров</p> <p>ПК-ТОП_11.2. У-1. Умеет анализировать бенчмарки, выявлять узкие места</p> <p>ПК-ТОП_11.2. У-2. Умеет разрабатывать код, оптимизированный для выбранной аппаратной архитектуры</p> <p>ПК-ТОП_11.3:</p> <p>ПК-ТОП_11.3. 3-1. Знает архитектуру ядра Linux</p> <p>ПК-ТОП_11.3. 3-2. Знает принципы работы встраиваемых операционных систем</p> <p>ПК-ТОП_11.3. У-1. Умеет портировать код между различными вариантами загрузчиков и ОС</p> <p>ПК-ТОП_11.3. У-2. Умеет выполнять разработку на стыке программного обеспечения и оборудования</p> <p>ПК-ТОП_11.4:</p> <p>ПК-ТОП_11.4. 3-2. Знает форматы бинарных файлов, работу с памятью</p> <p>ПК-ТОП_11.4. У-1. Умеет отлаживать код через специализированные отладчики системного программного обеспечения</p> <p>ПК-ТОП_11.4. У-2. Умеет анализировать дампы памяти, дизассемблированный код</p>		
ПК-Ф1: Способен планировать и организовывать аналитические	ПК-Ф1.1: Применяет методы машинного обучения и статистического анализа.	ПК-Ф1.1: ПК-Ф1.1. 3-1. Знает основные алгоритмы машинного обучения	Отчет по лабораторным работам	Зачёт: Отчет по лабораторным

<p>работы с использованием технологий больших данных</p>	<p>Знает типы анализа больших данных, виды аналитики; теоретические и прикладные основы анализа больших данных; содержание этапов жизненного цикла больших данных ПК-Ф1.2: Обеспечивает соответствие результатов анализа бизнес-задачам заказчика. Умеет планировать и проводить аналитические работы с использованием технологий больших данных ПК-Ф1.3: Подготавливает отчеты и визуализации для презентации результатов. Умеет проводить анализ больших данных</p>	<p>ПК-Ф1.1. 3-2. Знает методы статистического анализа данных ПК-Ф1.1. 3-3. Знает критерии выбора алгоритмов для различных задач ПК-Ф1.1. У-1. Умеет реализовывать алгоритмы машинного обучения ПК-Ф1.1. У-2. Умеет интерпретировать результаты статистического анализа ПК-Ф1.2: ПК-Ф1.2. 3-1. Знает методы перевода бизнес-требований в аналитические задачи ПК-Ф1.2. 3-2. Знает ключевые бизнес-метрики в предметной области ПК-Ф1.2. 3-3. Знает принципы интерпретации результатов для бизнес-пользователей ПК-Ф1.2. У-1. Умеет адаптировать аналитические модели под бизнес-потребности ПК-Ф1.2. У-2. Умеет оценивать экономический эффект от аналитических решений ПК-Ф1.3: ПК-Ф1.3 3-1. Знает принципы эффективной визуализации данных ПК-Ф1.3 3-2. Знает инструменты создания аналитических отчетов ПК-Ф1.3 3-3. Знает методы сторителлинга на основе данных ПК-Ф1.3 У-1. Умеет выбирать оптимальные типы визуализации ПК-Ф1.3 У-2. Умеет создавать интерактивные дашборды</p>		<p>работам</p>
--	---	---	--	----------------

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	12
- занятия семинарского типа (практические занятия / лабораторные работы)	12
- КСР	1
самостоятельная работа	47
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	6	0	2	2	4
Л2. Тестирование интегральных схем	6	2	0	2	4
П2. Тестирование интегральных схем	6	0	2	2	4
Л3. Конечные автоматы в интегральных схемах	6	2	0	2	4
П3. Реализация конечного автомата управления светофорами	6	0	2	2	4
Л4. Методы работы со статической памятью	6	2	0	2	4
П4. Вычисление арифметических выражений	6	0	2	2	4
Л5. Реализация взаимодействия FPGA и PC на основе протокола UART	6	2	0	2	4
П5. Вычисление арифметических выражений передаваемых с PC	6	0	2	2	4
Л6. Методы построения трансляторов	5	2	0	2	3
П6. Построение транслятора арифметических выражений для передачи данных на интегральную схему	6	0	2	2	4
Аттестация	0				
КСР	1			1	
Итого	72	12	12	25	47

Содержание разделов и тем дисциплины

Цель курса - освоение тестирования и программирования для FPGA устройств.

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.

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.

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.

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.

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.

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.

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 - Задания) для оценки сформированности компетенции ПК-Опер_1:

Практика 1. Комбинационная и последовательностная логика.

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

Практика 2. Тестирование интегральных схем

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

Практика 3. Реализация конечного автомата управления светофорами

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

5.1.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-ТОП_11:

Практика 1. Вычисление арифметических выражений.

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

Практика 2. Вычисление арифметических выражений передаваемых с РС.

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

Практика 3. Построение транслятора арифметических выражений для передачи данных на интегральную схему.

Реализация транслятора арифметических выражений в обратную польскую запись в машинных кодах на основе ANTLR. Кодирование и передача полученного выражения через UART.

Критерии оценивания (оценочное средство - Задания)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами
не зачтено	Выполнено менее половины задания, есть существенные недочеты

5.1.3 Типовые задания (оценочное средство - Отчет по лабораторным работам) для оценки сформированности компетенции ПК-Ф1:

Задание 1. Для лабораторной работы, оцениваемой ПК-ТОП_11, подготовить отчет. В отчете должен быть подробный анализ задачи. Для визуализации результатов использовать методы анализа больших данных.

Критерии оценивания (оценочное средство - Отчет по лабораторным работам)

Оценка	Критерии оценивания
зачтено	Отчет предоставлен преподавателю в срок и в полном объеме
не зачтено	Отчет не предоставлен преподавателю в срок или содержит значительные недочеты

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 теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.

					ошибок		
<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 отдельным и 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 и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми и 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-Опер_1

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, редукционные, 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?

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-ТОП_11

1. Для чего используется инструмент Pin Planner? Где взять номера пинов? Приведите примеры
2. Что такое тестовый стенд? Из каких основных компонентов он состоит?
3. В чем концептуальное отличие ModelSim от Verilator? Приведите примеры тестовых стендов
4. Как описать однопортовую SRAM память в SystemVerilog? Напишите пример кода для памяти размером 256x8 бит и примеры ее использования.
5. Опишите методы использования IP блока для работы с однопортовой памятью
6. Что такое UART и какие сигналы используются в этом интерфейсе? Опишите формат передачи одного байта.
7. Как реализовать UART-передатчик на SystemVerilog без IP блоков? Приведите пример реализации
8. Расскажите о методах инициализации SRAM через UART приемник на основе IP блоков.

9. Расскажите о методах разработки скриптов на Python для передачи данных через UART.
10. Для каких задач используется инструмент ANTLR? Приведите примеры
11. Приведите пример грамматики для разбора арифметического выражения.
12. Что такое дерево разбора? Какие алгоритмы построения существуют?
13. Опишите классическую структуру компилятора. Какие этапы проходит исходный код от текста до машинных инструкций?
14. Что из себя представляет стековая архитектура процессора? Чем она отличается от регистровой (например, RISC-V)? Почему для такой архитектуры проще писать транслятор
15. 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.

5.3.3 Типовые задания (оценочное средство - Отчет по лабораторным работам) для оценки сформированности компетенции ПК-Ф1

Задание 1. Для лабораторной работы, оцениваемой ПК-ТОП_11, подготовить отчет. В отчете должен быть подробный анализ задачи. Для визуализации результатов использовать методы анализа больших данных.

Критерии оценивания (оценочное средство - Отчет по лабораторным работам)

Оценка	Критерии оценивания
зачтено	Обучающийся представил подробный отчет по лабораторной работе
не зачтено	Обучающийся не представил своевременно / представил недостоверный отчет по лабораторной работе

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 25.06.2025, протокол № Протокол №11.