

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных
технологий и компьютерной безопасности
 / П.Ю. Гусев /
И.О. Фамилия
«31» августа 2021 г.

**РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Программирование на Verilog»**

наименование дисциплины (модуля) в соответствии с учебным планом)

Направление подготовки 09.03.01 Информатика и вычислительная техника
код и наименование направления подготовки/специальности

Профиль (специализация) Вычислительные машины, комплексы, системы и сети
название профиля/программы


Квалификация выпускника бакалавр

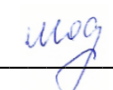
Нормативный период обучения 4 года / 4 года и 11 м.
Очная/очно-заочная/заочная (при наличии)

Форма обучения Очная/Заочная

Год начала подготовки 2021 г.

Автор(ы) программы Зав. кафедрой АВС  В.Ф. Барабанов
должность и подпись

Заведующий кафедрой
Автоматизированных и
вычислительных систем  В.Ф. Барабанов
наименование кафедры, реализующей дисциплину *подпись*

Руководитель ОПОП  С.Л. Подвальный
подпись

Воронеж 2021

ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

изучение и практическое освоение технологии разработки приложений на языке Verilog для моделирования и проектирования изделий электронной техники.

1.2. Задачи освоения дисциплины

к теоретическим задачам относятся освоение современного высокоуровневого языка Verilog в проектировании цифровых систем.

прикладные задачи состоят в приобретении навыков автоматизированного моделирования и проектирования электронных устройств.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на Verilog» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б.1 учебного плана (дисциплина по выбору).

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на Verilog» направлен на формирование следующих компетенций:

ПК-2 - Способен проектировать и разрабатывать компоненты программных комплексов и информационных систем, используя современные технологии программирования и инструментальные средства разработки

ПК-6 - Способен применять современные программные средства при проектировании компонентов технических средств инфокоммуникационных систем.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-2	Знать: возможности современных инструментальных средств разработки программных продуктов и технических средств
	Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения на языке Verilog
	Владеть: методами и средствами проектирования и разработки программного обеспечения и информационных систем на языке Verilog
ПК-6	Знать: возможности современных программных средств для проектирования компонентов технических средств

	Уметь: применять современные программные средства для проектирования компонентов технических средств
	Владеть: методиками проектирования компонентов технических средств с применением современных программных средств

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины «Программирование на Verilog» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий

Очная форма обучения

Вид учебной работы	Всего часов	Семестры			
		8			
Аудиторные занятия (всего)	60	60			
В том числе:					
Лекции	20	20			
Практические занятия (ПЗ)					
Лабораторные работы (ЛР)	40	40			
Самостоятельная работа	93	93			
Курсовой проект(работа) (есть, нет)	+	+			
Часы на контроль	27	27			
Вид промежуточной аттестации (зачет, зачет с оценкой, экзамен)	Экз.	Экз.			
Общая трудоемкость	час	180	180		
	зач. ед.	5	5		

Заочная форма обучения

Вид учебной работы	Всего часов	Семестры			
		10			
Аудиторные занятия (всего)	20	20			
В том числе:					
Лекции	4	4			
Практические занятия (ПЗ)					
Лабораторные работы (ЛР)	16	16			
Самостоятельная работа	147	147			
Курсовой проект(работа) (есть, нет)	+	+			
Часы на контроль	9	9			
Вид промежуточной аттестации (зачет, зачет с оценкой, экзамен)	Экз.	Экз.			

Общая трудоемкость	час	180	180			
	зач. ед.	5	5			

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Обзор системных сред проектирования изделий электронной техники.	Уровни представления и формы абстракции цифровых систем Современные высокоуровневые языки в проектировании СБИС. Использование высокоуровневых языков для программирования ПЛИС. Основные методы работы в САПР Xilinx <u>Самостоятельное изучение.</u> Системные среды проектирования изделий электронной техники (обзор).	6	-	10	30	46
2	Этапы проектирования изделий электронной техники с использованием языка Verilog	Основные этапы процесса подготовки проекта в поведенческой форме на языке Verilog. Базовая структура Verilog –файла. Поведенческая форма проекта Структурная форма проекта	6	-	14	32	52
3	Основные конструкции языка проектирования Verilog	Лексические и Программные элементы языка Verilog. Операции языка Verilog. Условные, операторы цикла и операторы выбора языка Verilog. Примеры реализации цифровых устройств с использованием языка Verilog. <u>Самостоятельное изучение.</u> Пакет ModelSim. Основные режимы и команды.	8	-	16	31	55
Итого			20	-	40	93	153

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Обзор системных сред проектирования изделий электронной техники.	Уровни представления и формы абстракции цифровых систем Современные высокоуровневые языки в проектировании СБИС. <u>Самостоятельное изучение.</u> Системные среды проектирования изделий электронной техники (обзор).	2		4	48	54
2	Этапы проектирования изделий электронной техники с использованием языка Verilog	Основные этапы процесса подготовки проекта в поведенческой форме на языке Verilog. Базовая структура Verilog –файла. Структурная форма проекта	1		4	50	56
3	Основные	Лексические и Программные	1		8	49	61

конструкции языка проектирования Verilog	элементы языка Verilog. Примеры реализации цифровых устройств с использованием языка Verilog. <i>Самостоятельное изучение. Пакет ModelSim. Основные режимы и команды.</i>					
Итого		4		16	147	171

5.2 Перечень лабораторных работ

Лабораторная работа № 1. Изучение интегрированной среды проектирования для ПЛИС ISE WebPACK. Разработка тестов с помощью временных диаграмм в пакете ModelSim.

Лабораторная работа № 2. Основные этапы автоматизированной разработки проекта на языке Verilog. Рассмотрение примеров проектов на языке Verilog.

Лабораторная работа № 3. Разработка иерархической схемы проекта на языке Verilog. Программирование, компиляция проекта на языке Verilog.

Лабораторная работа № 4. Моделирование, синтез на языке Verilog.

Компоновка, монтаж и разводка, временной анализ на языке Verilog.

Лабораторная работа № 5. Разработка проекта в поведенческой форме на языке Verilog.

Лабораторная работа № 6. Разработка проекта в структурной форме на языке Verilog.

Лабораторная работа № 7. Проектирование на структурном уровне с помощью редактора схем.

Лабораторная работа № 8. Отладка устройства с использованием ПЛИС (Spartan 3 Starter Kit ,Virtex 4 Starter Kit).

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 8 семестре для очной формы обучения и в 10 для заочной.

Примерная тематика курсового проекта:

"Разработка прототипа цифрового устройства с использованием отладочного комплекта Spartan 3 Starter Kit"

Целью данной работы является получение навыков разработки, тестирования и создания прототипов реальных цифровых устройств с использованием отладочных плат.

Задание

Разработать модель цифрового устройства на языке Verilog согласно выбранному варианту.

Провести функциональное тестирование разработанной модели с использованием отладочного комплекта Spartan 3 Starter Kit

Краткие теоретические сведения

Выполнение каждого из вариантов заданий требует вывода

информации на 7-сегментные индикаторы. В составе отладочной платы Spartan 3 Starter Kit имеется четыре светодиодных 7-сегментных индикатора. Каждый индикатор также содержит отдельный сегмент для отображения точки.

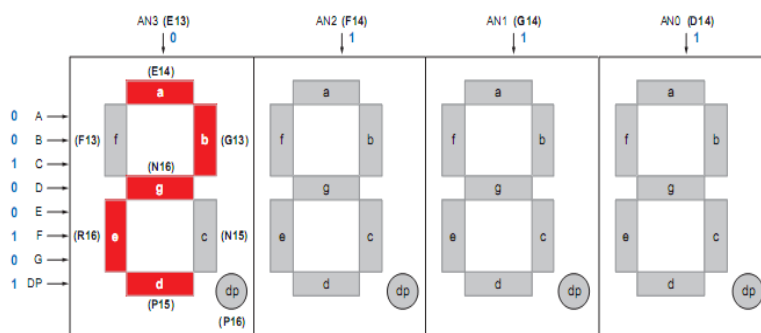


Рисунок 5.1 – Семисегментные индикаторы

В таблице 5.1 представлены выводы ПЛИС, подключенные к сегментам А-Г, и к сегменту DP, отвечающему за отображение точки.

Таблица 5.1 – Соответствие выводов ПЛИС сегментам индикатора

Сегмент	вывод ПЛИС
А	E14
В	G13
С	N15
Д	P15
Е	R16
F	F13
G	N16
DP	P16

В таблице 5.2 представлены выводы ПЛИС, к которым подключены дисплеи AN3 - AN0.

Таблица 5.2– Соответствия выводов ПЛИС выводам, контролирующим включение дисплея

Дисплей	вывод ПЛИС	AN 3	AN 2	AN 1	AN 0
		E13	F14	G1	D1
				4	4

В таблице 5.3 представлены сигналы, которые необходимо подать на сегменты А- G, чтобы отобразить необходимое число от 0 до 9.

Таблица 5.3– Символы дисплея и соответствующие им значения сегментов

Символ	Сегменты	Сегменты	Сегменты	Сегменты	Сегменты	Сегменты	Сегменты
0							
1							
2							

3							
4							
5							
6							
7							
8							
9							

Варианты заданий:

Вариант 1

Разработать устройство, позволяющее считать количество нажатий на кнопку и отображать их на 7-сегментном индикаторе. Предусмотреть возможность сброса количества нажатий.

Вариант 2

Разработать устройство, позволяющее рассчитать сумму двух 4-х битных чисел. Числа задаются с помощью 2-х позиционных переключателей. Результат вывести на 7-сегментный индикатор.

Вариант 3

Разработать устройство, позволяющее подсчитать количество единиц в 8-битном двоичном числе. Число задается с помощью 2-х позиционных переключателей. Результат вывести на 7-сегментный индикатор.

Вариант 4

Разработать устройство, функционирующее следующим образом: после одного нажатия на кнопку светодиодный индикатор включается. Затем после двух нажатий индикатор выключается. Затем после трех нажатий индикатор снова включается и так далее. Количество нажатий кнопки, требуемых для смены состояния светодиода вывести на 7-сегментный индикатор. Предусмотреть возможность сброса количества требуемых нажатий.

Вариант 5

Разработать устройство, реализующее 4-х битный счетчик кода Грея. Счет производится по нажатию на кнопку. Предусмотреть возможность сброса счетчика. Результат вывести в двоичном виде на светодиодные индикаторы а так же в десятичном виде на 7-сегментные индикаторы.

Вариант 6

Разработать устройство, реализующее 8-ми битный циклический сдвиговый регистр. Направление сдвига задается с помощью 2-х позиционного переключателя. Предусмотреть возможность сброса регистра в начальное состояние (00000001). Результат вывести в двоичном виде на

светодиодные индикаторы так же в десятичном виде на 7-сегментные индикаторы.

Вариант 7

Разработать устройство, позволяющее рассчитать произведение двух 3-х битных чисел. Числа задаются с помощью 2-х позиционных переключателей. Результат вывести на 7-сегментные индикаторы.

Вариант 8

Разработать устройство, формирующее 8-битное псевдослучайное число на основе генератора m -последовательности. Каждое новое число генерируется при нажатии на кнопку. Результат вывести в двоичном виде на светодиодные индикаторы а так же в десятичном виде на 7-сегментные индикаторы.

Вариант 9

Разработать устройство, представляющее собой счетчик, позволяющий вести прямой и обратный счет количества нажатий на кнопку (счет ведется от 0 до 9 или от 9 до 0). Режим работы счетчика задается с помощью 2-х позиционного переключателя. Результат вывести на 7-сегментный индикатор.

Вариант 10

Разработать устройство, позволяющее рассчитать сумму двух 5-битных двоичных чисел. Числа задаются с помощью 2-х позиционных переключателей в 2 этапа: по первому нажатию кнопки запоминается первый операнд, по второму нажатию запоминается второй операнд и производится вычисление. Результат выводится на 7-сегментный индикатор.

Вариант 11

Разработать устройство, позволяющее подсчитать количество одновременно нажатых кнопок на отладочной плате (не более 4). Результат вывести на 7-сегментный индикатор.

Вариант 12

Разработать устройство, позволяющее рассчитать сумму по модулю 2 (исключающее ИЛИ) двух 4-х битных чисел. Числа задаются с помощью 2-х позиционных переключателей. Результат вывести в двоичном виде на светодиодные индикаторы а так же в десятичном виде на 7-сегментные индикаторы.

Вариант 13

Разработать устройство позволяющее выполнить операции И, ИЛИ, исключающее ИЛИ с двумя 3-х битными числами. Числа и вид выполняемой операции задаются с помощью 2-х позиционных переключателей. Результат

операции вывести в двоичном виде на светодиодные индикаторы а так же в десятичном виде на 7-сегментные индикаторы.

Вариант 14

Разработать устройство, представляющее собой счетчик с настраиваемым значением инкремента. При каждом нажатии на кнопку значение счетчика увеличивается на 1, 2 3, или 4. Значение инкремента задается с помощью с помощью 2-х позиционных переключателей. Предусмотреть возможность сброса счетчика. Результат вывести на 7-сегментный индикатор.

Вариант 15

Разработать устройство, позволяющее рассчитать разность двух 4-х битных чисел. Числа задаются с помощью 2-х позиционных переключателей. Для отображения знака числа, в случае отрицательного результата, использовать средний сегмент (сегмент g) третьего 7-сегментного индикатора. Результат вывести на 7-сегментный индикатор.

Курсовой проект включают в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-2	Знать: возможности современных инструментальных средств разработки программных продуктов и технических средств	Возможности пакета Xilinx .	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: использовать существующие типовые решения и шаблоны проектирования программного	Базовая структура Verilog – файла.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	обеспечения на языке Verilog			
	Владеть: методами и средствами проектирования и разработки программного обеспечения и информационных систем с использованием языка Verilog	Разработка иерархической схемы проекта на языке Verilog.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-6	Знать: возможности современных программных средств для проектирования компонентов технических средств	Структура пакета САПР	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: применять современные программные средства для проектирования компонентов технических средств	Разработка проекта на языке Verilog	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть: методиками проектирования компонентов технических средств с применением современных программных средств	Разработка тестбенча на Verilog	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 8 семестре для очной формы обучения, 10 семестре для заочной формы обучения по четырехбалльной системе:

- «отлично»;
- «хорошо»;
- «удовлетворительно»;
- «неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии Оценивания	Отлично	Хорошо	Удовл	Неудовл
ПК-2	Знать: возможности современных инструментальных средств разработки программных продуктов и технических средств	Тест	Выполнение теста на 90-100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	Уметь: использовать существующие типовые решения и	Решение стандартных	Задачи решены в полном объеме и	Продемонстрирован верный ход решения	Продемонстрирован верный ход решения в	Задачи не решены

	шаблоны проектирования программного обеспечения на языке VHDL	практических задач	получены верные ответы	всех, но не получен верный ответ во всех задачах	большинстве задач	
	Владеть: методами и средствами проектирования и разработки программного обеспечения и информационных систем на языке VHDL	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-6	Знать: возможности современных программных средств для проектирования компонентов технических средств	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь: применять современные программные средства для проектирования компонентов технических средств	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть: методиками проектирования компонентов технических средств с применением современных программных средств	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

Задание 1

Какого типа значений из перечисленных нет в языке Verilog?

Варианты ответа:

- 1) 1
- 2) Z
- 3) U
- 4) X

Задание 2

Какой тип данных из перечисленных не существует в языке Verilog?

Варианты ответа:

- 1) reg
- 2) real
- 3) integer
- 4) **bool**

Задание 3

Какой переменной раньше всех будет присвоено значение?

```
initial begin
    #2 a <= 1;
    b <= 0;
    #3 c <= 1;
end
```

Варианты ответа:

- 1) Всем переменным сразу
- 2) c
- 3) **b**
- 4) a

Задание 4

Какое ключевое слово не является определителем типа цепи?

Варианты ответа:

- 1) **triorg**
- 2) wire
- 3) wor
- 4) tri

Задание 5

Какой системы счисления нет в языке Verilog?

Варианты ответа:

- 1) **шестеричной**
- 2) шестнадцатеричной
- 3) десятичной
- 4) восьмеричной

Задание 6

Какой оператор является внеблочным присваиванием?

Варианты ответа:

- 1) ==
- 2) **<=**
- 3) =
- 4) :=

Задание 7

Какой оператор является допустимым по отношению к данным типа real?

Варианты ответа:

- 1) &
- 2) **&&**
- 3) <<
- 4) <<<

Задание 8

Какого иерархического объекта нет в языке Verilog?

Варианты ответа:

- 1) модуль
- 2) **макромодуль**
- 3) микромодуль

4) встроенный примитив

Задание 9

Какого логического вентиля нет в языке Verilog?

Варианты ответа:

- 1) dmos
- 2) nmos
- 3) cmos
- 4) pmos

Задание 10

Что обозначает "wire a"?

Варианты ответа:

- 1) целочисленная переменная
- 2) **однобитовый проводник**
- 3) однобитовый сигнал
- 4) однобитовый регистр

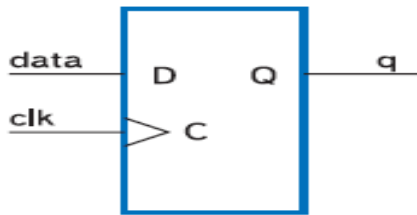
7.2.2 Примерный перечень заданий для решения стандартных задач

7.2.2 Примерный перечень заданий для решения стандартных задач

Составить HDL-модель цифрового устройства (ЦУ) на языке Verilog .

Задание 1

ЦУ: однотактный синхронный D-триггер, тактируемый фронтом сигнала синхронизации.



C	D	Q	Q*
0	-	Q	Q
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

"-" означает любой сигнал, приходящий на элемент

Вариант ответа:

```
`timescale 1ns/1ps
//D-триггер с тактируемым фронтом сигнала синхронизации
module test_L1(
    input D, clk,
    output reg Q
);
always @(posedge clk)
    Q <= D;
endmodule
```

Задание 2

Найдите ошибки в приведённом ниже коде

```
module test_L2(
    input [7:0] x,
    output [7:0] y
);
integer i;
always (x) begin
    for(i=0;i<8;i=i+1) y[i] = ~x[i];
end
```

Вариант ответа:

```

module test_L2(
    input [7:0] x,
    output reg [7:0] y
);

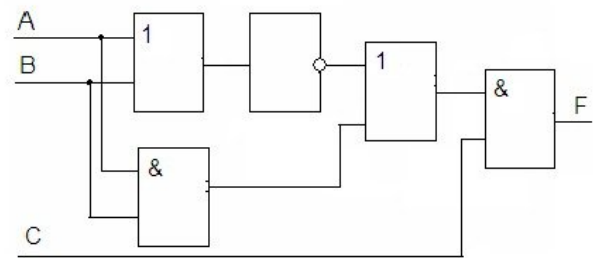
integer i;
always@(x) begin
    for(i=0;i<8;i=i+1) y[i] = ~x[i];
end
endmodule

```

Задание 3

ЦУ: логическая схема с тремя входными сигналами

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

**Вариант ответа:**

```

timescale 1ns / 1ps
//Логическая схема
module test_L3(
    input a, b, c,
    output reg f
);

always@(*) begin
    if((~a & ~b & c) | (a & b & c)) f <= 1'b1;
    else f <= 1'b0;
end
endmodule

```

Задание 4

Найдите ошибки в приведённом ниже коде

```

`timescale 1ns / 1ps
module test_L4(
    input a,
    output reg sel
);

always @( )
case(a)
    2'b00:sel=4'b0001;
    2'b01:sel=4'b0010;
    2'b10:sel=4'b0100;
    default:sel=4'b1000;
endmodule

```

Вариант ответа:

```

`timescale 1ns / 1ps
module test_L4(
    input [1:0]a,
    output reg [3:0]sel

```

```

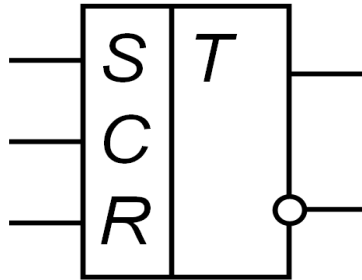
);

always @ (a)
case (a)
2'b00:sel=4'b0001;
2'b01:sel=4'b0010;
2'b10:sel=4'b0100;
default:sel=4'b1000;
endcase
endmodule

```

Задание 5

ЦУ: синхронный RS-триггер.



C	S	R	Q	Q*
0	-	-	0	0
			1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	не определено
1	1	1	1	не определено

"-" означает любой сигнал, приходящий на элемент

Вариант ответа:

```

`timescale 1ns / 1ps
//Синхронный RS-триггер
module test_L5(
    input r, s, clk,
    output reg q, qn
);

initial begin
    q=1'b0;
    qn=1'b1;
end

always @(posedge clk) begin
    if(~s) begin
        if(~r) begin
            q=q;
            qn=qn;
        end
        else begin
            q=1'b0;
            qn=1'b1;
        end
    end
    else begin
        if(~r) begin
            q=1'b1;
            qn=1'b0;
        end
        else begin
            q=1'bx;
            qn=1'bx;
        end
    end
end

```

```
        end
    end
endmodule
```

Задание 6

Найдите ошибки в приведённом ниже коде

```
`timescale 1ns / 1ps
module test_L6(
    int enable, gate, data,
    output q
);

    always @ (enable or data or gate)
    if (enable)
    q = data & gate;
end
```

Вариант ответа:

```
`timescale 1ns / 1ps
module test_L6(
    input enable, gate, data,
    output reg q
);

    always @ (enable or data or gate)
    if (enable)
    q = data & gate;
endmodule
```

Задание 7

Найдите ошибки в приведённом ниже коде

```
`timescale 1ns / 1ps
test_L7(
    input c, d, e, f,
    input [0:1] s,
    output reg pout
);
    always @(c or d or e or f or s) begin
    if (s == 2'b00)
    pout = c;
    else if (s ==2'b01)
    pout = d;
    else if (s ==2'b10)
    pout = e;
    else pout = f;
    and
    endmodule
```

Вариант ответа:

```
`timescale 1ns / 1ps
module test_L7(
    input c, d, e, f,
    input [1:0] s,
    output reg pout
);

    always @(c or d or e or f or s) begin
    if (s == 2'b00)
    pout = c;
    else if (s ==2'b01)
    pout = d;
```



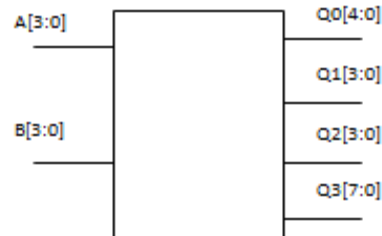
```

else if (s ==2'b10)
pout = e;
else pout = f;
end
endmodule

```

Задание 8

ЦУ: цифровое устройство, реализующее 4 арифметических операции.



Q0	Q1	Q2	Q3
A+B	A-B	A*B	A/B

Вариант ответа:

```

`timescale 1ns / 1ps
//Арифметические операции
module test_L8(
input [3:0] A, B,
output reg [4:0] Q1,
output reg [3:0] Q2, Q3,
output reg [7:0] Q4
);

always @(A or B) begin
Q1 = A+B;
Q2 = A-B;
Q3 = A/2;
Q4 = A*B;
end
endmodule

```

Задание 9

Найдите ошибки в приведённом ниже коде

```

`timescale 1ns / 1ps
module test_S3(
input clk, reset,
output to_zero, [7:0] data
);

reg data_reg;
wire data_plus_one;

always @(posedge clk, posedge reset)
if (reset)
data_reg <= 0;
else if (to_zero)
data_reg <= 0;
else
data_reg <= data_plus_one;
data_plus_one = data_reg + 1'b1;
data = data_reg;

endmodule

```

Вариант ответа:

```

`timescale 1ns / 1ps
module test_S3(

```

```

        input clk, reset, to_zero,
        output [7:0] data
    );

reg data_reg;
wire data_plus_one;

always @(posedge clk, posedge reset)
    begin
        if (reset)
            data_reg <= 0;
        else if (to_zero)
            data_reg <= 0;
        else
            data_reg <= data_plus_one;
    end
assign data_plus_one = data_reg + 1'b1;
assign data = data_reg;

endmodule

```

Задание 10

Найдите ошибки в приведённом ниже коде

```

`timescale 1ns / 1ps
module test_S4(
    input [3:0] data,
    input [1:0] selector,
    reg mux_out
);

always (sel) begin
case(selector)
    2'b00 : mux_out <= data[0];
    2'b01 : mux_out <= data[1];
    2'b10 : mux_out <= data[2];
    2'b11 : mux_out <= data[3];
endcase
endmodule

```

Вариант ответа:

```

`timescale 1ns / 1ps
module test_S4(
    input [3:0] data,
    input [1:0] selector,
    output reg mux_out
);

always @(sel or data) begin
case(selector)
    2'b00 : mux_out = data[0];
    2'b01 : mux_out = data[1];
    2'b10 : mux_out = data[2];
    2'b11 : mux_out = data[3];
endcase
end
endmodule

```

7.2.3 Примерный перечень заданий для решения прикладных задач

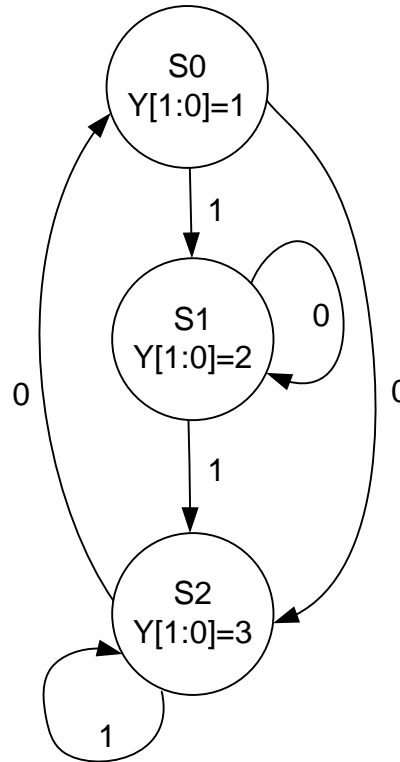
Целью данной работы является получение навыков разработки и тестирования конечных автоматов.

Разработать модель конечного автомата на языках Verilog согласно выбранному варианту.

Варианты заданий:

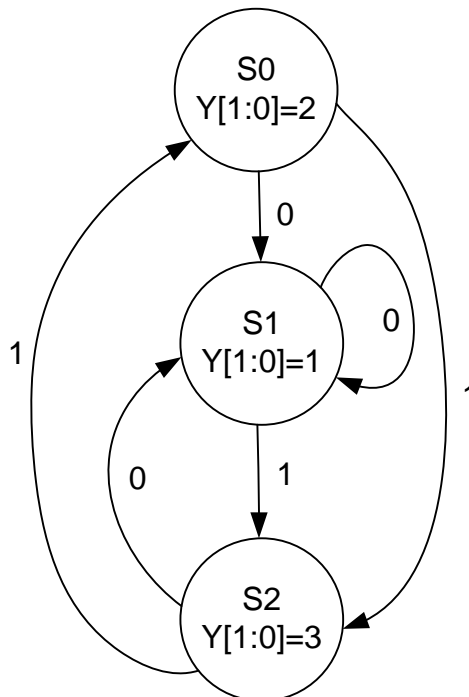
Вариант 1

Начальное состояние: S2



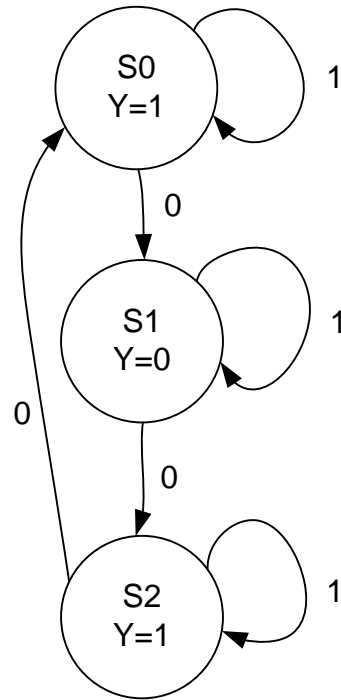
Вариант 2

Начальное состояние: S1



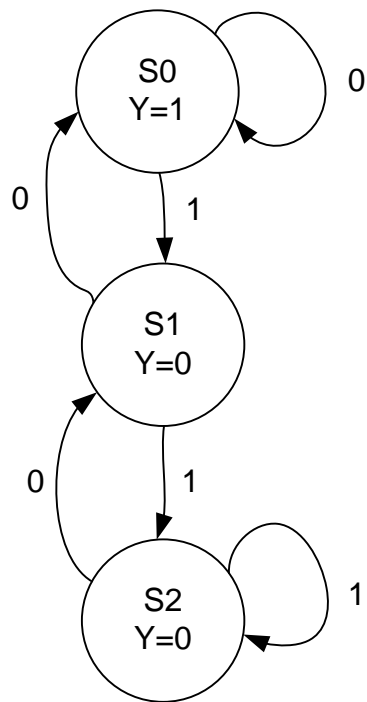
Вариант 3

Начальное состояние: S0



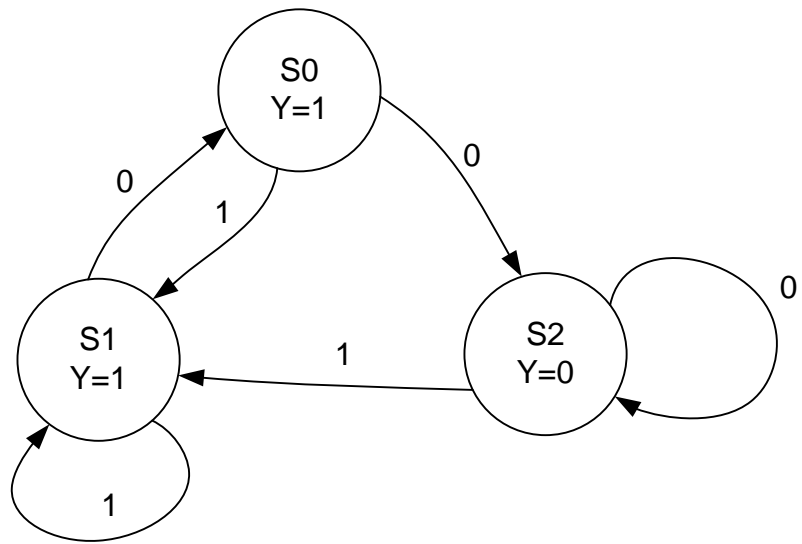
Вариант 4

Начальное состояние: S1



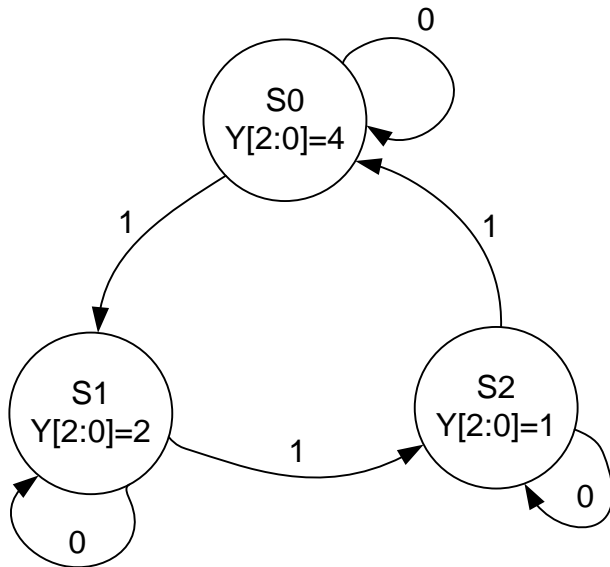
Вариант 5

Начальное состояние: S2



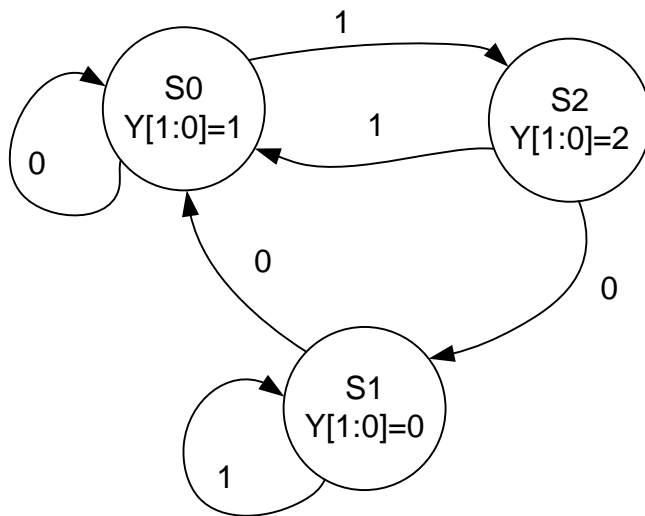
Вариант 6

Начальное состояние: S0



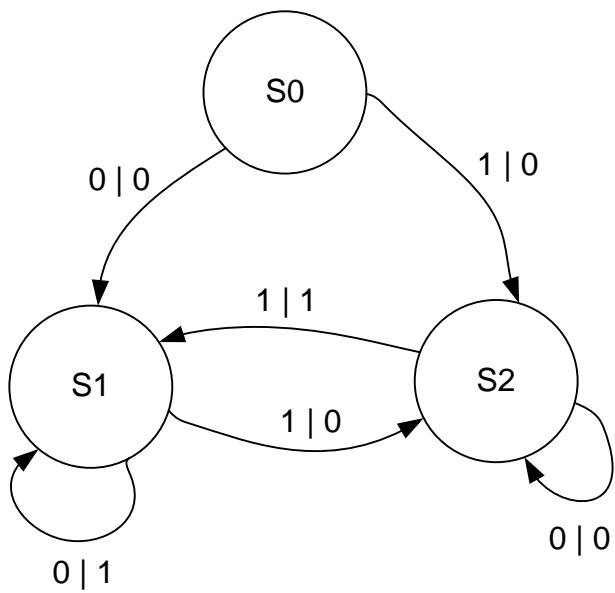
Вариант 7

Начальное состояние: S1



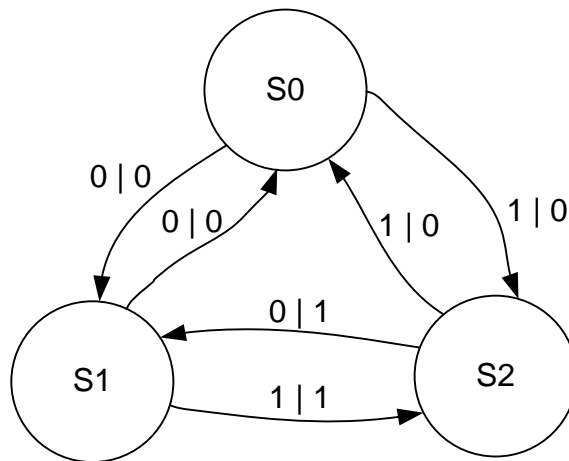
Вариант 8

Начальное состояние: S0



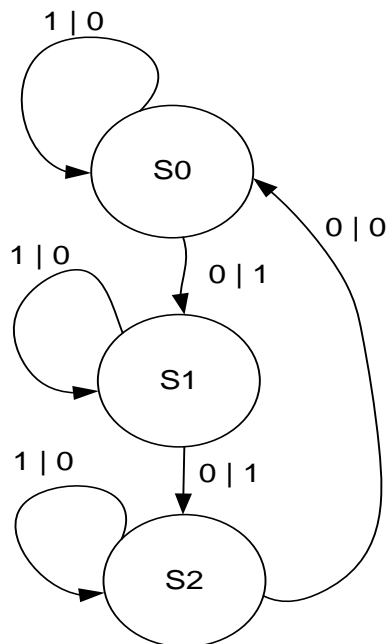
Вариант 9

Начальное состояние: S1



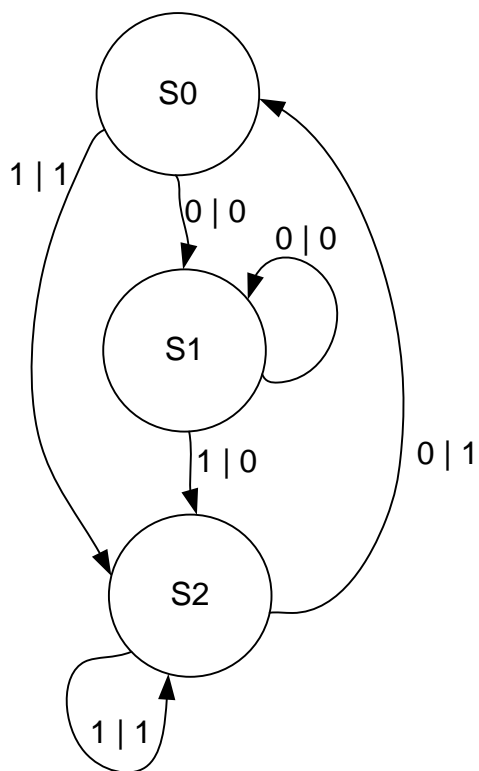
Вариант 10

Начальное состояние: S1



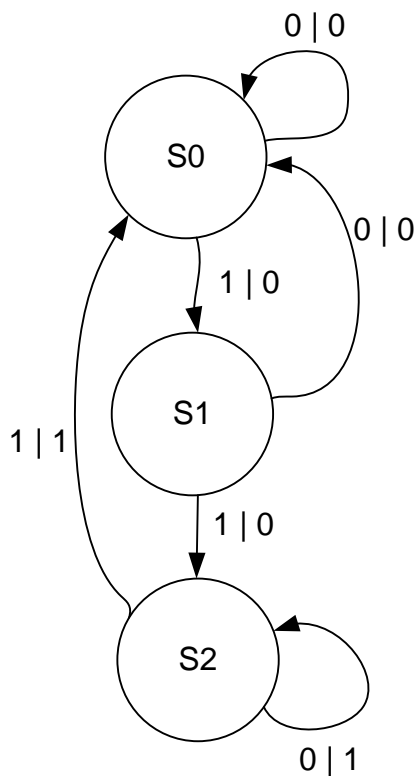
Вариант 11

Начальное состояние: S0



Вариант 12

Начальное состояние: S2



Примеры выполнения заданий:

Код модели автомата Мили на языке Verilog HDL

```

module mealy_fsm
(
  input  clk, x, rst,
  output reg [1:0] y

```

);

```
reg          [1:0]state;
parameter S0 = 0, S1 = 1, S2 = 2, S3 = 3;

always @ (posedge clk or posedge rst) begin
    if (rst)
        state <= S0;
    else
        case (state)
            S0:
                if (x)
                    begin
                        state <= S1;
                    end
                else
                    begin
                        state <= S1;
                    end
            S1:
                if (x)
                    begin
                        state <= S2;
                    end
                else
                    begin
                        state <= S1;
                    end
            S2:
                if (x)
                    begin
                        state <= S3;
                    end
                else
                    begin
                        state <= S1;
                    end
            S3:
                if (x)
                    begin
                        state <= S2;
                    end
                else
                    begin
                        state <= S3;
                    end
        endcase
    end

always @ (state or x)
begin
    case (state)
        S0:
            if (x)
                begin
                    y = 2'b00;
                end
            else
                begin
                    y = 2'b10;
                end
        S1:
            if (x)
```



```

begin
    y = 2'b01;
end
else
begin
    y = 2'b00;
end
S2:
    if (x)
    begin
        y = 2'b10;
    end
    else
    begin
        y = 2'b01;
    end
S3:
    if (x)
    begin
        y = 2'b11;
    end
    else
    begin
        y = 2'b00;
    end
end
endcase
end
endmodule

```

Код модели автомата Мура на языке Verilog HDL

```

module moore_fsm
(
    input  clk, x, rst,
    output reg [1:0] y
);

    reg [1:0]state;
    parameter S0 = 0, S1 = 1, S2 = 2, S3 = 3;

    always @ (state) begin
        case (state)
            S0:
                y = 2'b01;
            S1:
                y = 2'b10;
            S2:
                y = 2'b11;
            S3:
                y = 2'b00;
            default:
                y = 2'b00;
        endcase
    end

    always @ (posedge clk or posedge rst) begin
        if (rst)
            state <= S0;
        else
            case (state)
                S0:
                    state <= S1;

```

```

S1:
    if (x)
        state <= S2;
    else
        state <= S1;
S2:
    if (x)
        state <= S3;
    else
        state <= S1;
S3:
    if (x)
        state <= S2;
    else
        state <= S3;
endcase
end

endmodule

```

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Средства автоматизации проектирования в электронике (обзор).
2. Основные концепции развития современных систем проектирования.
3. Этапы проектирования вычислительных систем.
4. Современные промышленные технические и программные средства.
5. Современные языки проектирования цифровых устройств.
6. Основные методы работы в САПР Xilinx
7. Основные системные возможности ПЛИС FPGA.
8. Структура языка Verilog.
9. Этапы проектирования с использованием языков Verilog.
10. Лексические элементы языка Verilog.
11. Назначение и основные конструкции языка Verilog.
12. Типы данных языка Verilog.
13. Выражения в языке Verilog.
14. Операторы цикла языка Verilog.
15. Операции языка Verilog.
16. Структурная форма проекта языка Verilog.
17. Атрибуты языка Verilog.
18. Иерархические структуры языка Verilog.
19. Поведенческое описание языка Verilog.
20. Системные задачи и функции языка Verilog.
21. Примеры проектов на языке Verilog.
22. Классификация операторов Verilog.
23. Отличительные особенности языков VHDL и Verilog.
24. Условные операторы языка Verilog
25. Операторы выбора языка Verilog
26. Разработка тестов с помощью временных диаграмм в пакете ModelSim.
27. Отладка устройства с использованием ПЛИС (Spartan 3 Starter Kit , Virtex 4 Starter Kit)

7.2.6 Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 5 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 5 баллов. Максимальное количество набранных баллов – 10.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 5 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 5 до 7 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 8 до 9 баллов.

4. Оценка «Отлично» ставится, если студент набрал 10 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Обзор системных сред проектирования изделий электронной техники.	ПК-2, ПК-6	Тест.
2	Этапы проектирования изделий электронной техники с использованием языка Verilog	ПК-2, ПК-6	Защита лабораторных работ.
3	Основные конструкции языка проектирования Verilog	ПК-2, ПК-6	Тест.

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется

оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Барабанов В.Ф., Подвальный С.Л., Гребенникова Н.И., Сафронов В.В. Основы проектирования цифровых устройств на языках VHDL и Verilog: учебное пособие/ В.Ф. , С.Л. Подвальный, Гребенникова Н.И., Сафронов В.В.. – Воронеж: ФГБОУ ВПО «ВГТУ», 2012. –216 с. ,13,5 п.л.
2. Барабанов А.В. Проектирование цифровых устройств на языках VHDL и Verilog: учеб. пособие.- 2015 Воронеж: ФГБОУ ВПО «ВГТУ».
3. Барабанов В.Ф. Основы автоматизации проектирования, тестирования и управления жизненным циклом изделий: учебное пособие/ В.Ф. Барабанов, А.Д. Поваляев, С.Л. Подвальный, С.В. Тюрин. – Воронеж: «Научная книга», 2011. –165 с. (Гриф УМО),11,3 п.л.
4. Строгонов А.В. Основы проектирования микропроцессорных устройств: учебное пособие. 2009, –175 с. – Воронеж: ВГТУ.
5. Барабанов В.Ф., Гребенникова Н.И. Сафронов В.В., Дырда И.Н. и др. № 177-2010, ВГТУ. МУ к выполнению лабораторных работ № 1-4 по дисциплине «АПВС» для студентов специальности 230101 «Вычислительные машины, комплексы, системы и сети» очной и очной сокращенной форм обучения// МУ № 177-2010. – Воронеж: ВГТУ, 2010 – с. 2,8 уч.-изд.л
6. Сперанский Д.В. Моделирование, тестирование и диагностика цифровых устройств : учебное пособие / Сперанский Д.В., Скобцов Ю.А., Скобцов В.Ю.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 529 с. — ISBN 978-5-4497-0551-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/94854.html>
7. Лобач В.Т. Основы проектирования цифровых устройств радиоэлектронных систем : учебное пособие / Лобач В.Т., Потипак М.В.. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2020. — 140 с. — ISBN 978-5-9275-3656-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/115521.html>
8. Головицына М.В. Проектирование радиоэлектронных средств на основе

современных информационных технологий : учебное пособие / Головицына М.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 503 с. — ISBN 978-5-4497-0690-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/97578.html>

9. Организация самостоятельной работы обучающихся : методические указания для студентов, осваивающих основные образовательные программы высшего образования – бакалавриата, специалитета, магистратуры: методические указания / сост. В.Н. Почечихина, И.Н. Крючкова, Е.И. Головина, В.Р. Демидов; ФГБОУ ВО «Воронежский государственный технический университет». – Воронеж, 2020. – 14 с.
10. Методические рекомендации по выполнению курсовых работ и проектов для студентов направлений 09.03.01, 09.04.01 Информатика и вычислительная техника очной и заочной форм обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. Н.И. Гребенникова, В.В. Сафронов, А.М. Нужный, А.В. Барабанов, Воронеж, 2020. 20 с.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Лицензионное ПО:

- Windows Professional 7 Single Upgrade MVL A Each Academic
- Microsoft Office Word 2007
- Microsoft Office Power Point 2007

Свободно распространяемое ПО:

- ISE WebPACK Design Software;

Отечественное ПО:

- Яндекс.Браузер
- Архиватор 7z
- Astra Linux

Ресурс информационно-телекоммуникационной сети «Интернет»:

- Образовательный портал ВГТУ
- <http://www.edu.ru/>
- <https://metanit.com/>

Информационно-справочные системы:

- <http://window.edu.ru>
- <https://wiki.cchgeu.ru/>

Современные профессиональные базы данных:

- <https://proglib.io>
- <https://msdn.microsoft.com/ru-ru/>
<https://docs.microsoft.com/>

Мультимедийные лекционные демонстрации:

Презентация «Современные языки проектирования цифровых устройств».

Презентация «Этапы проектирования цифровых устройств с использованием языка

Verilog».

Презентация «Базовая структура Verilog -файла».

Презентация «Лексические элементы языка Verilog».

Презентация «Программные элементы данных языка Verilog».

Презентация «Операции языка Verilog».

Презентация «Оператор always».

Презентация «Поведенческая форма проекта».
Презентация «Структурная форма проекта».
Презентация «Примеры реализации цифровых устройств с использованием языка Verilog».

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения лекционных занятий необходима аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

Для проведения лабораторных работ необходима лаборатория с ПК, оснащенными программами для проведения лабораторного практикума и обеспечивающими возможность доступа к локальной сети кафедры и Интернет, из следующего перечня:

- 307 (Лаборатория микропроцессорной техники)
- 309 (Лаборатория телекоммуникационных систем)
- 311 (Лаборатория разработки программных систем)
- 320 (Лаборатория общего назначения)
- 322 (Лаборатория распределённых вычислений)
- 324 (Специализированная лаборатория сетевых систем управления (научно-образовательный центр «АТОС»))
- 325 (Лаборатория автоматизации проектирования вычислительных комплексов и сетей)

Лаборатории расположены по адресу: 394066, г. Воронеж, Московский проспект, 179 (учебный корпус №3).

10 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программирование на VHDL» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента (особенности деятельности студента инвалида и лица с ОВЗ, при наличии таких обучающихся)
Лекция	Написание конспекта лекций: кратко, схематично, последовательно

	<p>фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.</p>
Лабораторная работа	<p>Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.</p>
Самостоятельная работа	<p>Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие:</p> <ul style="list-style-type: none"> - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	<p>Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.</p>

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	-------------------------------	--