

**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 /
подпись / *И.О. Фамилия*
«31» августа 2021 г.

**РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Автоматизация проектирования вычислительных систем»**
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 2019 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

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

подготовка специалистов для научно-исследовательской, проектно-конструкторской, производственно-технической и организационно-управленческой работы в области средств вычислительной техники (СВТ), способных разрабатывать и применять в своей деятельности методы и средства автоматизации проектирования и производства вычислительных систем (ВС).

1.2. Задачи освоения дисциплины

- к теоретическим задачам относятся освоение принципов построения и эксплуатации современных САПР СВТ; разработки различных автоматизированных информационных систем;

- прикладные задачи состоят в приобретении навыков автоматизированного конструирования и изготовления СВТ, методов автоматизированного оформления конструкторской документации.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Автоматизация проектирования вычислительных систем» относится к дисциплинам вариативной части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Автоматизация проектирования вычислительных систем» направлен на формирование следующих компетенций:

ПК-1 - Способен обосновывать проектные решения, проверять их корректность и эффективность

ПК-3 - Способен проводить оценку проектных решений и формировать техническое задание на разработку вычислительных систем.

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

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	Знать: методики анализа требований к программному обеспечению
	Уметь: выполнять этапы проектирования программных и технических средств с использованием языка VHDL
	Владеть: методиками проверки корректности и эффективности проектных решений в области разработки программных и технических средств

ПК-3	Знать: методики оценки проектных решений; типовую структуру технического задания на разработку вычислительной системы
	Уметь: осуществлять оценку проектных решений по типовой методике; составлять типовое техническое задание на разработку вычислительной системы
	Владеть: методикой проведения оценки проектных решений; методикой формирования технического задания на разработку вычислительной системы.
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

Общая трудоемкость дисциплины «Автоматизация проектирования вычислительных систем» составляет 4 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		7
Аудиторные занятия (всего)	72	72
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	36	36
Самостоятельная работа	72	72
Виды промежуточной аттестации - зачет с оценкой	+	+
Общая трудоемкость: академические часы	144	144
зач.ед.	4	4

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		9

Аудиторные занятия (всего)	20	20
В том числе:		
Лекции	8	8
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	120	120
Контрольная работа	+	+
Часы на контроль	4	4
Виды промежуточной аттестации - зачет с оценкой	+	+
Общая трудоемкость:		
академические часы	144	144
зач.ед.	4	4

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	Лаб. зан.	СРС	Всего, час
1	Современные средства автоматизации проектирования в электронике	Современные средства автоматизации проектирования. Новые возможности систем автоматизированного проектирования и моделирования (DipTrace, P-CAD, Protel, Altium Designer, и др.) <i>Самостоятельное изучение. Пакет Altium Designer. Основные команды.</i> Основные понятия и этапы процесса проектирования. Внутрисхемное тестирование цифровых электронных компонентов. Эволюция мощных микропроцессоров	6	6	18	30
2	Методы интеграции с конструкторскими и технологическими САПР, средствами инженерного анализа	Интеграция систем моделирования и проектирования (DipTrace, P-CAD, PSPICE, SPECCTRA и др.). Интеграция ERP и PDM систем для построения единой информационной системы. Проблемы схмотехнического моделирования БИС. <i>Самостоятельное изучение. Пакет PSPICE. Основные команды.</i>	6	6	18	30
3	Современные высокоуровневые языки (VHDL и Verilog) в проектировании СБИС. Уровни представления и формы абстракции цифровых систем	Уровни представления и формы абстракции цифровых систем. Современные технологии 3D интеграции. Тенденции развития и основные характеристики ПЛИС. Высокоуровневые языки в проектировании цифровых систем. Этапы проектирования СВТ с использованием языка VHDL. <i>Самостоятельное изучение. Основные команды языка VHDL.</i>	16	16	18	50
4	Основы автоматизации производства и управ	Введение в систему управления жизненным циклом изделия Product Lifecycle Management (PLM).	8	8	18	34

	ления жизненным циклом изделия	Основные методы работы с PDM системой				
Итого			36	36	72	144

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Современные средства автоматизации проектирования в электронике	Современные средства автоматизации проектирования. Новые возможности систем автоматизированного проектирования и моделирования (DipTrace, P-CAD, Protel, Altium Designer, и др.) <i>Самостоятельное изучение. Пакет Altium Designer. Основные команды.</i> Основные понятия и этапы процесса проектирования. Внутрисхемное тестирование цифровых электронных компонентов. Эволюция мощных микропроцессоров	1	4	30	35
2	Методы интеграции с конструкторскими и технологическими САПР, средствами инженерного анализа	Интеграция систем моделирования и проектирования (DipTrace, P-CAD, PSPICE, SPECCTRA и др.). Интеграция ERP и PDM систем для построения единой информационной системы.	1	1	30	32
3	Современные высокоуровневые языки (VHDL и Verilog) в проектировании СБИС. Уровни представления и формы абстракции цифровых систем	Уровни представления и формы абстракции цифровых систем. Тенденции развития и основные характеристики ПЛИС. Высокоуровневые языки в проектировании цифровых систем. Этапы проектирования СВТ с использованием языка VHDL. <i>Самостоятельное изучение. Основные команды языка VHDL.</i>	4	5	30	39
4	Основы автоматизации производства и управления жизненным циклом изделия	Введение в систему управления жизненным циклом изделия Product Lifecycle Management (PLM). Основные методы работы с PDM системой	2	2	30	34
Итого			8	12	120	140

5.2 Перечень лабораторных работ

Автоматизированная разработка электронной схемы и печатных плат в современной системе проектирования (DipTrace, Altium Designer и др.)

Внутрисхемное тестирование цифровых электронных компонентов.

Разработка тестов с помощью временных диаграмм.

Использование интегрированной среды проектирования Xilinx.

Автоматизированная разработка проекта на языке VHDL.

Написание тестбенчей на HDL.

Изучение основ работы в PDM системе (1С: PDM, Teamcenter Siemens)

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

В соответствии с учебным планом освоение дисциплины предусматривает выполнение контрольной работы в 9 семестре заочного обучения.

Вариант контрольной работы: Проектирование делителей частоты. Проектирование цифровых устройств счета времени. Цифровые семисегментные индикаторы.

Плата Spartan 3 Starter Kit имеет четыре семисегментных индикатора, которые контролируются определенным выводом ПЛИС AN0–AN3 (FPGA pin: E13, F14, G14, D14), как изображено на рисунке 30.

Каждая цифра отображается подсвечиванием определенной комбинации некоторых из восьми сегментов, которые также в свою очередь контролируются определенными выводами ПЛИС.

Схема верного оперирования индикаторами следующая: необходимо подать нужную комбинацию, соответствующую отображаемому числу на шину сегментов и перевести в низкий уровень аноды тех индикаторов, на которых следует отобразить это число.

На рисунке 2.12, на крайнем дисплее изображено число 2. Для работы данного дисплея необходимо на вывод E13 подать сигнал низкого уровня, а остальные дисплеи «погасить» путем подачи на выводы F14, G14, D14 сигналы высокого уровня. Далее для отображения числа 2, как показано на рисунке 2, на выводы, соответствующие сегментам a, b, g, e, d будут поданы сигналы низкого уровня, а на остальные – высокого. Сегмент dp отвечает за отображение точки, который также управляется сигналом низкого уровня.

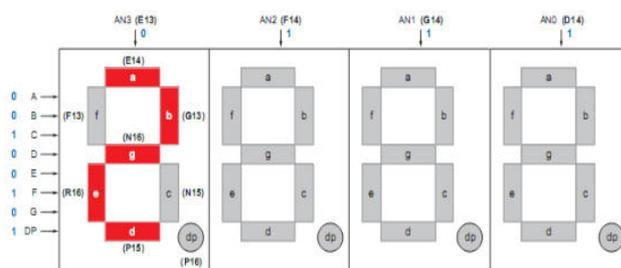


Рисунок 2.12 – Семисегментные индикаторы

В таблице 2.5 перечислены FPGA соединения, соответствующие сегментам a – g, dp.

Таблица 2.5 – Соответствия FPGA pin сегментам индикатора

Сегмент	FPGA pin
A	E14
B	G13

C	N15
D	P15
E	R16
F	F13
G	N16
DP	P16

В таблице 2.6 перечислены FPGA pin, которым соответствуют дисплеи AN3 – AN0.

Таблица 2.6 – Соответствия FPGA pin дисплеям

Дисплей	AN3	AN2	AN1	AN0
FPGA pin	E13	F14	G14	D14

В следующей таблице перечислено, какие сигналы необходимо подать на сегменты a– g, чтобы получить необходимое число от 0 до F.

Таблица 2.7 – Символы дисплея и соответствующие значения сегментов

Символ	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0

7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

В задании необходимо будет отображать необходимые данные одновременно на четырех индикаторах. Для этого необходимо произвести деление исходной частоты (50 МГц). Глаз человека начинает воспринимать изображение на частоте 400 Гц [45]. Временная диаграмма отображения данных на четырех дисплеях индикатора представлена на рисунке 2.13.

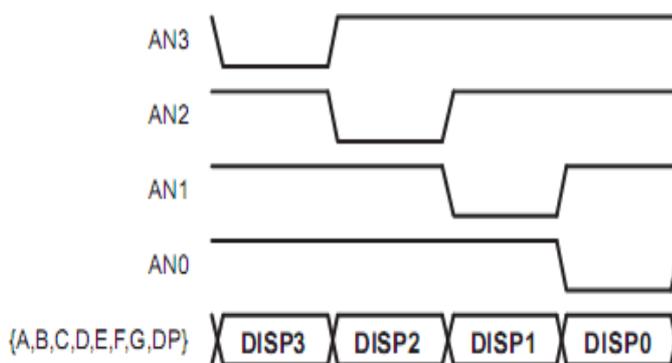


Рисунок 2.13 – Отображение данных на 4 дисплеях

Таблица 2.8 – Варианты заданий

Вариант	Задание
1	Пусть начальное положение переключателя будет 0 – при переключении в 1 обнулить счетчик и начать счет, при переключении обратно в 0 – счет прекратить. Во время счета отображать результат. Начальная установка индикаторов – все нули. Счет произвести в формате “0.000”, то есть с точностью

	до одной миллисекунды.
2	Синтезировать секундомер в формате “00.00”, с правой стороны от точки минуты, с левой секунды. Секундомер должен запускаться и останавливаться с кнопки.
3	Синтезировать секундомер в формате “00.00”, с правой стороны от точки секунды, с левой доли секунды. Секундомер должен запускаться и останавливаться с кнопки.
4	Синтезировать секундомер в формате “00.00”, с правой стороны от точки доли секунды, с левой – секунды. Секундомер должен запускаться и останавливаться с кнопки. Добавить будильник, т.е. при значении времени, жестко фиксированном в коде, остановить счет и моргать всеми светодиодами с частотой полсекунды.
5	Синтезировать секундомер в формате “00.00”, с правой стороны от точки секунды, с левой доли секунды. Секундомер должен запускаться и останавливаться с кнопки. Добавить будильник, т.е. при значении времени, настроенным пользователем, остановить счет и моргать всеми светодиодами с частотой полсекунды.
6	Синтезировать часы: считающие минуты и секунды. Реализовать полную настройку часов, т.е. установка минут, установка часов, настройка будильника.

Требования к отчету

Отчет должен содержать следующую информацию по работе:

- название работы, цель и задание, согласно варианту;
- описание разработанного алгоритма работы дешифратора / мультиплексора / счетчика/устройства счёта времени (словесно графически);
- код программы с комментариями;
- рисунок с временными диаграммами устройств (смоделированное устройство);
- словесное описание работы устройств на плате Spartan 3 Starter Kit.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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-1	Знать: методики анализа требований к программному обеспечению	Знать возможности системы автоматизированного проектирования Xilinx	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: выполнять этапы проектирования программных и технических средств с использованием языка VHDL	Этапы проектирования СВТ с использованием языка VHDL	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть: методиками проверки корректности и эффективности проектных решений в области разработки программных и технических средств	Структурная верификация проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-3	Знать: методики оценки проектных решений; типовую структуру технического задания на разработку вычислительной системы	Разработка общих требований к проектируемой системе	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: осуществлять оценку проектных решений по типовой методике; составлять типовое техническое задание на разработку вычислительной системы	Составление подробного ТЗ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть: методикой проведения оценки проектных решений; методикой формирования технического задания на разработку вычислительной системы.	Верификация проекта	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	Структура пакета САПР	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	компонентов технических средств			
	Уметь: применять современные программные средства для проектирования компонентов технических средств	Разработка проекта на языке VHDL	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 VHDL	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 7 семестре для очной формы обучения, 9 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-1	Знать: методика анализа требований к программному обеспечению	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь: выполнять этапы проектирования программных и технических средств с использованием языка VHDL	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 проверки корректности и эффективности проектных решений в области разработки программных и технических 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
ПК-3	Знать: методика оценки проектных решений; типовую струк-	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов

	туру технического задания на разработку вычислительной системы					
	Уметь: осуществлять оценку проектных решений по типовой методике; составлять типовое техническое задание на разработку вычислительной системы	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: методикой проведения оценки проектных решений; методикой формирования технического задания на разработку вычислительной системы.	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

Задание1 .

Для чего предназначен код на языке VHDL?

```
entity <entity_name> is
generic (
  <generic_name> : <type> := <value>;
  <other generics>...
);
port (
  <port_name> : <mode> <type>;
  <other ports>...
);
end <entity_name>;
```

Варианты ответа:

- а) Объявление сущности
- б) Объявление компонента
- в) Архитектура проекта
- г) Экземпляр компонента

Задание2 .

Тип Units используется, для задания...

Варианты ответа:

- а) Неопределенного сигнала
- б) **Временных промежутков, переменных**
- в) Для задания напряжения, подаваемого на контроллер
- г) Аналогового сигнала

Задание3 .

Допустима ли такая запись?

```
while I=I loop
  <statement>;
  <statement>;
end loop;
```

Варианты ответа:

- а) **Нет, из-за ошибки синтаксиса**
- б) Нет, из-за наличия других конструкций
- в) Да, но есть более подходящие конструкции
- г) Да,

Задание4 .

Сколько разных значений может принимать объект типа std_logic?

Варианты ответа:

- а) 2
- б) 5
- в) 7

г) 9

Задание 5 .

Что описывает следующий VHDL код?

```
process(<clock>)
begin
  if (<clock>'event and <clock> = '1') then
    if (<reset> = '1') then
      <output> <= "0000";
    else
      case <input> is
        when "00" => <output> <= "0001";
        when "01" => <output> <= "0010";
        when "10" => <output> <= "0100";
        when "11" => <output> <= "1000";
        when others => <output> <= "0000";
      end case;
    end if;
  end if;
end process;
```

Варианты ответа:

- а) Синхронный d-триггер
- б) Синхронный счетчик с направлением счета
- в) Сдвиговый регистр
- г) **4-х битный декодер**

Задание 6 .

Какие значения будут принимать объекты представленных классов?

```
constant gnd : bit := '0';
variable y : bit;
signal z : std_logic;
```

Варианты ответа:

- а) gnd = 0, y = 1, z = 1
- б) **gnd = 0, y = 0, z = 0**
- в) gnd = 0, y = 0, z = 1
- г) gnd = 0, y = 1, z = 0

Задание 7 .

Что произойдет с сигналом Rst?

```
signal Rst : std_logic;
```

```
Rst <= 0;
```

Варианты ответа:

- а) Сигнал будет сравниваться с нулем
- б) **Сигнал примет значение логического нуля**
- в) Сигнал примет значение меньше нуля
- г) Сигнал примет значение логической единицы

Задание 8 .

Что описывает следующий VHDL код?

```
process (<selector>,<input1>,<input2>,<input3>,<input4>)
begin
  case <selector> is
    when "00" => <output> <= <input1>;
    when "01" => <output> <= <input2>;
    when "10" => <output> <= <input3>;
    when "11" => <output> <= <input4>;
    when others => <output> <= <input1>;
  end case;
end process;
```

Варианты ответа:

- а) Мультиплексор
- б) Компаратор
- в) Умножитель
- г) Двухнаправленный двух битный регистр

Задание 9 .

Что из перечисленного не является оператором в VHDL ?

Варианты ответа:

- а) loop
- б) exit
- в) if
- г) **for**

Задание 10 .

Что описывает следующий VHDL код?

```
process (<clock>)
begin
  if <clock>'event and <clock>='1' then
    if <reset>='1' then
      <output> <= '0';
    elsif <clock_enable>='1' then
      <output> <= <input>;
    end if;
  end if;
end process;
```

Варианты ответа:

- а) **Синхронный d-триггер**
- б) Синхронный счетчик с направлением счета
- в) Сдвиговый регистр
- г) 4-х битный декодер

7.2.2 Примерный перечень заданий для решения стандартных задач

Синтезировать мультиплексор согласно своему варианту на языке VHDL.

№	Количество информационных входов	Комбинации на информационных входах
1	3	"0001"; "1011"; "0111";
2	2	"100"; "101";
3	5	"11101"; "10101"; "11010"; "11111"; "10011";
4	8	"0100101"; "1011001"; "1011110"; "1001101"; "1101101"; "1001001"; "1110001"; "0100111";
5	3	"001"; "111"; "101";
6	6	"10101"; "10001"; "10010"; "11011"; "10111"; "01001";
7	5	"110110"; "100101"; "101110"; "111111"; "000000";
8	4	"11101001"; "10100101"; "10110110"; "11000011";
9	7	"111"; "101"; "100"; "001"; "010"; "011"; "110";
10	8	"000"; "111"; "101"; "100"; "001"; "010"; "011"; "110";

Варианты ответов:

```

1.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity multiplex is
port (in1: in bit_vector(0 to 1);
      out1: out bit_vector(0 to 3)
);
end multiplex;

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is
when "00"=>out1<="0001";
when "01"=>out1<="1011";
when "10"=>out1<="0111";
when others=>out1<="0000";

```

```
end case;
end process;
end Behavioral;
```

2.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity multiplex is
    port (in1: in STD_LOGIC;
          out1: out bit_vector(0 to 2)
        );
end multiplex;
```

```
architecture Behavioral of multiplex is
begin
    process(in1)
    begin
        case in1 is
            when "0"=>out1<="100";
            when "1"=>out1<="101";
        end case;
    end process;
end Behavioral;
```

3.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity multiplex is
    port (in1: in bit_vector(0 to 2);
          out1: out bit_vector(0 to 4)
        );
end multiplex;
```

```
architecture Behavioral of multiplex is
begin
    process(in1)
    begin
        case in1 is
            when "000"=>out1<="11101";
            when "001"=>out1<="10101";
            when "010"=>out1<="11010";
            when "011"=>out1<="11111";
            when "100"=>out1<="10011";
            when others=>out1<="00000";
        end case;
    end process;
end Behavioral;
```

4.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```

entity multiplex is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 6)
);
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is
when "000"=>out1<="0100101";
when "001"=>out1<="1011001";
when "010"=>out1<="1011110";
when "011"=>out1<="1001101";
when "100"=>out1<="1101101";
when "101"=>out1<="1001001";
when "110"=>out1<="1110001";
when "111"=>out1<="0100111";
end case;
end process;
end Behavioral;

```

5.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is
port (in1: in bit_vector(0 to 1);
      out1: out bit_vector(0 to 2)
);
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is "101";
when "00"=>out1<="001";
when "01"=>out1<="111";
when "10"=>out1<="101";
when others=>out1<="000";
end case;
end process;
end Behavioral;

```

6.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 4)
      );
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is
when "000"=>out1<="10101";
when "001"=>out1<="10001";
when "010"=>out1<="10010";
when "011"=>out1<="11011";
when "100"=>out1<="10111";
when "101"=>out1<="01001";
when others=>out1<="00000";
end case;
end process;
end Behavioral;

```

7.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 5)
      );
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is
when "000"=>out1<="110110";
when "001"=>out1<="100101";
when "010"=>out1<="101110";
when "011"=>out1<="111111";
when "100"=>out1<="000000";
when others=>out1<="000001";
end case;
end process;
end Behavioral;

```

8.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is
port (in1: in bit_vector(0 to 1);
      out1: out bit_vector(0 to 7)
);
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is
when "00"=>out1<= "11101001";
when "01"=>out1<="10100101";
when "10"=>out1<= "10110110";
when "11"=>out1<= "11000011";
end case;
end process;
end Behavioral;

```

9.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 2)
);
end multiplex;

```

```

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is 7
when "001"=>out1<= "111";
when "010"=>out1<="101";
when "010"=>out1<= "100";
when "011"=>out1<= "001";
when "100"=>out1<= "010";
when "101"=>out1<= "011";
when "110"=>out1<= "110";
when others=>out1<= "000";
end case;
end process;
end Behavioral;

```

10.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```

entity multiplex is

```

```

port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 2)
);

end multiplex;

architecture Behavioral of multiplex is
begin
process(in1)
begin
case in1 is 8
  when "000"=>out1<= "000";
  when "001"=>out1<= "111";
  when "010"=>out1<="101";
  when "010"=>out1<= "100";
  when "011"=>out1<= "001";
  when "100"=>out1<= "010";
  when "101"=>out1<= "011";
  when "110"=>out1<= "110";
  when others=>out1<= "000";
end case;
end process;
end Behavioral

```

7.2.3 Примерный перечень заданий для решения прикладных задач

Синтезировать дешифратор согласно своему варианту.

Вариант	Дешифратор
1	3 в 8
2	1 в 2
3	2 в 3
4	3 в 7
5	3 в 6
6	3 в 5
7	4 в 7
8	4 в 5
9	4 в 6
10	4 в 8

Варианты ответов:

1.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 7)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "001"=>out1<= "00000001";
when "010"=>out1<= "00000010";
when "010"=>out1<= "00000100";
when "011"=>out1<= "00001000";
when "100"=>out1<= "00010000";
when "101"=>out1<= "00100000";
when "110"=>out1<= "01000000";
when "111"=>out1<= "10000000";
end case;
end process;
end Behavioral;

```

2.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in STD_LOGIC;
      out1: out bit_vector(0 to 1)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "0"=>out1<= "01";
when "1"=>out1<= "10";
end case;
end process;
end Behavioral;

```

3.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is

```

```

port (in1: in bit_vector(0 to 1);
      out1: out bit_vector(0 to 2)
      );
end deshifrator;

```

architecture Behavioral of deshifrator is

```

begin
process(in1)
begin
case in1 is
  2 B 3
  when "00"=>out1<= "001";
  when "01"=>out1<= "010";
  when "10"=>out1<= "100";
  when others=>out1<= "000";
end case;
end process;
end Behavioral;

```

4.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

entity deshifrator is

```

port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 6)
      );
end deshifrator;

```

architecture Behavioral of deshifrator is

```

begin
process(in1)
begin
case in1 is
  when "000"=>out1<= "0000001";
  when "001"=>out1<= "0000010";
  when "010"=>out1<= "0000100";
  when "011"=>out1<= "0001000";
  when "100"=>out1<= "0010000";
  when "101"=>out1<= "0100000";
  when "110"=>out1<= "1000000";
  when others=>out1<= "0000000";
end case;
end process;
end Behavioral;

```

5.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

entity deshifrator is

```

port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 5)
      );

```

```

        );
end deshifrador;

architecture Behavioral of deshifrador is
begin
process(in1)
begin
case in1 is
    when "000"=>out1<= "000001";
    when "001"=>out1<= "000010";
    when "010"=>out1<= "000100";
    when "011"=>out1<= "001000";
    when "100"=>out1<= "010000";
    when "101"=>out1<= "100000";
    when others=>out1<= "000000";
end case;
end process;
end Behavioral;

```

6.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity deshifrador is
port (in1: in bit_vector(0 to 2);
      out1: out bit_vector(0 to 4)
      );
end deshifrador;

architecture Behavioral of deshifrador is
begin
process(in1)
begin
case in1 is      3 B 5
    when "000"=>out1<= "00001";
    when "001"=>out1<= "00010";
    when "010"=>out1<= "00100";
    when "011"=>out1<= "01000";
    when "100"=>out1<= "10000";
    when others=>out1<= "00000";
end case;
end process;
end Behavioral;

```

```

7.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in bit_vector(0 to 3);
      out1: out bit_vector(0 to 6)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "0000"=>out1<= "0000001";
when "0001"=>out1<= "0000010";
when "0010"=>out1<= "0000100";
when "0011"=>out1<= "0001000";
when "0100"=>out1<= "0010000";
when "0101"=>out1<= "0100000";
when "0110"=>out1<= "1000000";
when others=>out1<= "0000000";
end case;
end process;
end Behavioral;

```

```

8.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in bit_vector(0 to 3);
      out1: out bit_vector(0 to 4)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "0000"=>out1<= "00001";
when "0001"=>out1<= "00010";
when "0010"=>out1<= "00100";
when "0011"=>out1<= "01000";
when "0100"=>out1<= "10000";
when others=>out1<= "00000";
end case;
end process;
end Behavioral;

```

```

9.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in bit_vector(0 to 3);
      out1: out bit_vector(0 to 5)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "0000"=>out1<= "000001";
when "0001"=>out1<= "000010";
when "0010"=>out1<= "000100";
when "0011"=>out1<= "001000";
when "0100"=>out1<= "010000";
when "0101"=>out1<= "100000";
when others=>out1<= "000000";
end case;
end process;
end Behavioral;

```

```

10.
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity deshifrator is
port (in1: in bit_vector(0 to 3);
      out1: out bit_vector(0 to 7)
      );
end deshifrator;

architecture Behavioral of deshifrator is
begin
process(in1)
begin
case in1 is
when "0000"=>out1<= "00000001";
when "0001"=>out1<= "00000010";
when "0010"=>out1<= "00000100";
when "0011"=>out1<= "00001000";
when "0100"=>out1<= "00010000";
when "0101"=>out1<= "00100000";
when "0111"=>out1<= "01000000";
when "1000"=>out1<= "10000000";
when others=>out1<= "00000000";
end case;
end process;
end Behavioral;

```

end process;
end Behavioral;

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Обзор средств автоматизации проектирования в электронике (DipTrace, Altium Designer и др.).
2. Основные концепции развития современных систем проектирования.
3. Современное состояние электронной промышленности в России.
4. Структура и виды обеспечения САПР.
5. Классификационные характеристики САПР.
6. Интеграция пакетов САПР и конвертация форматов данных.
7. Классификация и область определения ВС.
8. Проблемы при преобразовании файлов систем проектирования (форматы DXF, PDIF и др.).
9. Эволюция микропроцессоров и ПК.
10. Типовой технологический маршрут изготовления СВТ.
11. Интеграция систем DipTrace, P-CAD, PSPICE, SPECCTRA и др..
12. Классификация и система обозначений интегральных схем (ИС).
13. Современные языки проектирования цифровых устройств.
14. Основные конструкции языка VHDL.
15. Этапы проектирования цифровых устройств с использованием языка VHDL.
16. Примеры проектов на языке VHDL.
17. Написание тестбенчей на языке HDL.
18. Структура и функциональные возможности системы Teamcenter Siemens
19. Структура и функциональные возможности системы Xilinx.
20. Структура и функциональные возможности системы Dip Trace.
21. Структура и функциональные возможности системы Altium Designer.
22. Тенденции развития ПЛИС. Структура ПЛИС
23. Перспективы расширения систем проектирования.
24. Интеграция систем на базе CALS- технологий
25. Универсальная справочная система электронных компонентов
26. Назначение и структура PLM систем.
27. Структура и функциональные возможности системы PLM Siemens.

7.2.5 Примерный перечень вопросов для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет с оценкой 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	Наименование оценочного средства
1	Современные средства автоматизации проектирования в электронике	ПК-1, ПК-3, ПК- 6	Тест.
2	Методы интеграции с конструкторскими и технологическими САПР, средствами инженерного анализа	ПК-1, ПК-3, ПК- 6	Тест.
3	Современные высокоуровневые языки (VHDL и Verilog) в проектировании СБИС. Уровни представления и формы абстракции цифровых систем	ПК-1, ПК-3, ПК- 6	Защита лабораторных работ.
4	Основы автоматизации производства и управления жизненным циклом изделия	ПК-1, ПК-3, ПК- 6	Защита лабораторных работ.

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.

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., Автоматизация проектирования электронных средств: Учеб. Пособие. Воронеж: гос. тех. ун-т, 2004. -224 с. (14.1 п.л.)
2. 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
3. Автоматизация конструкторско- технологического проектирования электронных средств: Учеб. пособие/ Муратов А.В., Барабанов В.Ф., Подвальный С.Л., Болгов А.Т., Невежин Е.В. Воронеж: Воронеж. гос. техн. ун-т, 2006. -200 с. (Гриф УМО, 12.5 п.л.).
 4. Барабанов В.Ф., Подвальный С.Л., Луконин Ю.А. Компьютерная графика в САД системах: Учеб. пособие- Ростов-на-Дону: Издательский центр ДГТУ, 2007, -214 с. (Гриф УМО, 13,4 п.л.).
 5. Барабанов А.В. Проектирование цифровых устройств на языках VHDL и Verilog: учеб. пособие.- 2015 Воронеж: ФГБОУ ВПО «ВГТУ».
 6. 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
 7. Архитектуры и топологии многопроцессорных вычислительных систем [Электронный ресурс] А.В. Богданов [и др.].— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 135 с.— Режим доступа: <http://www.iprbookshop.ru/52189.html>.— ЭБС «IPRbooks».
 8. Высокоуровневое проектирование встраиваемых систем [Электронный ресурс] Платунов А.Е., Постников Н.П.— Электрон. текстовые данные.— СПб.: Университет ИТМО, 2013.— 174 с.— Режим доступа: <http://www.iprbookshop.ru/66425.html>.— ЭБС «IPRbooks»
 9. Методические рекомендации по выполнению контрольных работ для бакалавров направления 09.03.01 профиля «Вычислительные машины, комплексы, системы и сети», магистров профиля 09.04.01 Информатика и вычислительная техника, программа: Распределенные автоматизированные системы очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. А.М. Нужный, Ю.С. Акинина, Н.И. Гребенникова. Воронеж: Изд-во ВГТУ, 2020. – 8с.
 10. 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

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
- NX Academic Perpetual License
- Teamcenter Unified Academic Perpetual License
- Tecnomatix Manufacturing Acad Perpetual License
- DipTrace 2.XX Standard Edition
- Altium Designer Custom Board Implementation, Perpetual EDU License

Свободно распространяемое ПО:

- Microsoft Visual Studio Community Edition
- ISE WebPACK Design Software

Отечественное ПО:

- Яндекс.Браузер
- Архиватор 7z
- Astra Linux
- 1С:PDM Управление инженерными данными

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 среды проектирования электронной техники».

Презентация «Тенденции развития компьютерных технологий».

Презентация «Управление деятельностью предприятия с использованием возможностей системы mySAP Business Suite».

Презентация «Этапы проектирования изделий электронной техники с использованием языка VHDL».

Презентация «Проектирование изделий электронной техники с использованием языка Verilog».

Презентация «Введение в написание тестбенчей на HDL».

Презентация «Основы автоматизации производства и управления жизненным цик-

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 «Автоматизация проектирования вычислительных систем» читаются лекции, проводятся лабораторные работы.

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.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно 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.
Лабораторная	Лабораторные работы позволяют научиться применять теоретические

работа	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.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <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	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 зачетом с оценкой, зачетом с оценкой три дня эффективнее всего использовать для повторения и систематизации материала.

Лист регистрации изменений

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
1	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины. Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем.	31.08.2020	
2	Внесены изменения в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем, учебной литературы, необходимой для освоения дисциплины.	31.08.2021	