

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

**«УТВЕРЖДАЮ»**  
Декан факультета радиотехники и  
электроники  /В.А. Небольсин/  
« 25 » ноября 2022 г.

**РАБОЧАЯ ПРОГРАММА**  
дисциплины (модуля)

**«Современные системы проектирования РЭС»**

Направление подготовки (специальность) 11.03.01 «Радиотехника»

Профиль (специализация) «Радиотехнические средства передачи,  
приема и обработки сигналов»

Квалификация выпускника бакалавр

Срок освоения образовательной программы 4 года / 4 года 11 месяцев

Форма обучения Очная / заочная

Год начала подготовки 2023

Автор программы  Д.А. Максимов /

Заведующий кафедрой  
радиотехники  /А.В. Останков/

Руководитель ОПОП  /А.В. Останков/

Воронеж 2022

# 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

## 1.1. Цели дисциплины

Цель дисциплины «Современные системы проектирования РЭС» состоит в приобретении студентами знаний основ языков описания аппаратуры VHDL и VerilogHDL и применении данных языков в процессе проектирования радиоэлектронных устройств различного назначения на современном уровне.

## 1.2. Задачи освоения дисциплины

Для достижения цели ставятся задачи:

- изучения основных синтаксических конструкций языков описания аппаратуры (HDL-языков);
- изучения возможностей пакетов прикладных программ (ППП) синтеза и моделирования проектов радиоэлектронных устройств на ПЛИС на основе текстового программного описания алгоритмов;
- использования ППП для реализации разработанных проектов в ПЛИС.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Современные системы проектирования РЭС» относится к дисциплинам обязательной части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Современные системы проектирования РЭС» направлен на формирование следующих компетенций:

ОПК-5-Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-5	знать принципы реализации алгоритмов в параллельном виде и параллельного программирования, синтаксические конструкции, алфавит, типы данных, операции и выражения, операторы HDL-языков (VHDL, VerilogHDL), методику применения HDL-языков для синтеза и моделирования радиоэлектронных устройств на инструментальном персональном компьютере
	уметь разрабатывать алгоритмы обработки сигналов в

	<p>параллельном виде, использовать пакеты прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных программ свободного доступа для автоматизированной разработки проектов радиоэлектронных устройств на ПЛИС, в том числе написания программ на HDL-языках, синтеза программ на уровень логических вентилях, размещения и трассировки в кристалле, моделирования на функциональном поведенческом уровне и с учетом временных задержек реальных кристаллов</p>
	<p>владеть практическими навыками программирования на языках VHDL и VerilogHDL, в том числе написания программ для синтеза в кристалл и моделирования во временной области радиоэлектронных устройств с использованием пакетов прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных программ свободного доступа</p>

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Современные системы проектирования РЭС» составляет 3 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры
		8
<b>Аудиторные занятия (всего)</b>	60	60
В том числе:		
Лекции	24	24
Лабораторные работы (ЛР)	36	36
<b>Самостоятельная работа</b>	48	48
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость академические часы	108	108
з.е.	3	3

**заочная форма обучения**

Виды учебной работы	Всего часов	Семестры
		15
<b>Аудиторные занятия (всего)</b>	20	20
В том числе:		
Лекции	8	8
Лабораторные работы (ЛР)	12	12
<b>Самостоятельная работа</b>	84	84
Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость академические часы	108	108
з.е.	3	3

## 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. зап.	СРС	Всего, Час
1	Введение. Архитектура и особенности микросхем ПЛИС на примере конкретных семейств. Понятие о HDL-языках.	Классификация цифровых микросхем. Место микросхем программируемой логики в общем процессе проектирования. Основные производители микросхем ПЛИС. Типы архитектур ПЛИС. Понятие о языках описания аппаратуры (HDL-языках) VHDL и VerilogHDL. Парадигма программист – схемотехник. HDL-программа с точки зрения программиста и схемотехника.	4	6	8	18
2	Языки VHDL и VerilogHDL. Применение языков для синтеза и моделирования. Структура проекта на VHDL и VerilogHDL.	HDL-языки для синтеза и моделирования. Понятие о параллельных алгоритмах и процессах. Принципы моделирования параллельных процессов на последовательной инструментальной ЭВМ. Понятие о процессе синтеза схем по текстовому формализованному описанию (Синтезатор ↔ Компилятор). Первая HDL-«программа». Entity и архитектурные тела VHDL. Проектный модуль VerilogHDL. Стили описания проектов. Структура и поведение.	4	6	8	18
3	Структурные и поведенческие архитектурные тела VHDL и проектные модули VerilogHDL.	Методика структурного описания проектов. Методика поведенческого описания проектов. Смешанное описание. Типы и подтипы данных, физические типы языка VHDL. Сигналы, переменные и константы, атрибуты сигналов языка VHDL. Типы языка Verilog. Типы reg и wire. Оператор Process языка VHDL. Оператор always языка VerilogHDL.	4	6	8	18

4	Сигналы и переменные в VHDL. Блокирующее и неблокирующее присваивания Verilog HDL. Последовательные операторы VHDL и Verilog HDL.	Понятие сигнала в VHDL. Блокирующее и неблокирующее присваивания Verilog. Место и область видимости сигналов в программном модуле. Отличия переменных языков VHDL и Verilog. Соединения языка Verilog. Сходства и отличия переменных в сравнении с процедурными языками программирования. Операторы присваивания VHDL и Verilog. Модели задержек transport и inertial VHDL. Модели и синтаксис задержек в Verilog. Операторы условия (if) и выбора (case) VHDL и Verilog. Операторы повторения (loop) VHDL и цикла Verilog. Операторы ожидания (wait) VHDL и Verilog.	4	6	8	18
5	Параллельные операторы VHDL. Процедуры и функции языка VHDL	Параллельное сигнальное присваивание VHDL. Непрерывное присваивание Verilog. Операторы Process и Block VHDL. Оператор инициализации Verilog. Оператор fork Verilog. Структурное описание. Оператор вхождения VHDL. Операторы размножения (Generate) VHDL и Verilog. Оператор конкретизации Verilog. Функции и процедуры. Их параллельный и последовательный вызов. Пакеты (package). Особенности функций и процедур VHDL. Особенности функций Verilog.	4	6	8	18
6	Описание на VHDL и Verilog типовых цифровых устройств. Методика моделирования цифровых устройств, описанных с использованием HDL-языков	Примеры описания комбинационных схем на VHDL и Verilog. Логика, мультиплексоры, шифраторы, дешифраторы. Уровневые (прозрачные) защелки (Latch) и их «ошибочное» появление при описании комбинационной логики. Примеры описания последовательностных схем на VHDL и Verilog. Триггеры (регистры), счетчики, память. Описание конечных автоматов (FSM). Синхронные и асинхронные схемы. Схемы синхронизации. Реализация арифметических функций на VHDL и Verilog. Testbench VHDL (Test fixture Verilog) и методика его создания с помощью языковых конструкций. Имитация периодических и однократных воздействий. Использование процедур VHDL для описания поведения сигналов. Операторы вывода VHDL и Verilog и их применение при моделировании. Оператор проверки Assert языка VHDL.	4	6	8	18
<b>Итого</b>			<b>24</b>	<b>36</b>	<b>48</b>	<b>108</b>

### заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зап.	СРС	Всего, Час
1	Введение. Архитектура и особенности микросхем ПЛИС на примере конкретных семейств. Понятие о HDL-языках.	Классификация цифровых микросхем. Место микросхем программируемой логики в общем процессе проектирования. Основные производители микросхем ПЛИС. Типы архитектур ПЛИС. Понятие о языках описания аппаратуры (HDL-языках) VHDL и Verilog HDL. Парадигма программист – схемотехник. HDL-программа с точки зрения программиста и схемотехника.	2	2	14	18
2	Языки VHDL и Verilog HDL. Применение языков для синтеза и моделирования. Структура проекта на VHDL и Verilog HDL.	HDL-языки для синтеза и моделирования. Понятие о параллельных алгоритмах и процессах. Принципы моделирования параллельных процессов на последовательной инструментальной ЭВМ. Понятие о процессе синтеза схем по текстовому формализованному описанию (Синтезатор ↔ Компилятор).	2	2	14	18

		Первая HDL-«программа». Entity и архитектурные тела VHDL. Проектный модуль VerilogHDL. Стили описания проектов. Структура и поведение.				
3	Структурные и поведенческие архитектурные тела VHDL и проектные модули VerilogHDL.	Структура и поведение. Методика структурного описания проектов. Методика поведенческого описания проектов. Смешанное описание. Типы и подтипы данных, физические типы языка VHDL. Сигналы, переменные и константы, атрибуты сигналов языка VHDL. Оператор Process языка VHDL. Типы языка Verilog. Типы reg и wire. Оператор always языка VerilogHDL.	2	2	14	18
4	Сигналы и переменные в VHDL. Блокирующее и неблокирующее присваивания VerilogHDL. Последовательные операторы VHDL и Verilog HDL.	Понятие сигнала в VHDL. Блокирующее и неблокирующее присваивания Verilog. Место и область видимости сигналов в программном модуле. Отличия переменных языков VHDL и Verilog. Соединения языка Verilog. Сходства и отличия переменных в сравнении с процедурными языками программирования. Операторы присваивания VHDL и Verilog. Модели задержек transport и inertial VHDL. Модели и синтаксис задержек в Verilog. Операторы условия (if) и выбора (case) VHDL и Verilog. Операторы повторения (loop) VHDL и цикла Verilog. Операторы ожидания (wait) VHDL и Verilog.	2	2	14	18
5	Параллельные операторы VHDL. Процедуры и функции языка VHDL	Параллельное сигнальное присваивание VHDL. Непрерывное присваивание Verilog. Операторы Process и Block VHDL. Оператор инициализации Verilog. Оператор fork Verilog. Структурное описание. Оператор вхождения VHDL. Оператор размножения (Generate) VHDL и Verilog. Оператор конкретизации Verilog. Функции и процедуры. Их параллельный и последовательный вызов. Пакеты (package). Особенности функций и процедур VHDL. Особенности функций Verilog.	-	2	14	16
6	Описание на VHDL и Verilog типовых цифровых устройств. Методика моделирования цифровых устройств, описанных с использованием HDL-языков	Примеры описания комбинационных схем на VHDL и Verilog. Логика, мультиплексоры, шифраторы, дешифраторы. Уровневые (прозрачные) защелки (Latch) и их «ошибочное» появление при описании комбинационной логики. Примеры описания последовательностных схем на VHDL и Verilog. Триггеры (регистры), счетчики, память. Описание конечных автоматов (FSM). Синхронные и асинхронные схемы. Схемы синхронизации. Реализация арифметических функций на VHDL и Verilog. Testbench VHDL (Testfixture Verilog) и методика его создания с помощью языковых конструкций. Имитация периодических и однократных воздействий. Использование процедур VHDL для описания поведения сигналов. Операторы вывода VHDL и Verilog и их применение при моделировании. Оператор проверки Assert языка VHDL.	-	2	14	16
<b>зачет</b>						<b>4</b>
<b>Итого</b>			<b>8</b>	<b>12</b>	<b>84</b>	<b>108</b>

## 5.2. Перечень лабораторных работ

1. Ознакомительная. Проектирование и моделирование двоичного реверсивного счетчика с загрузкой. Дополнительное задание: проектирование двоично-десятичного счетчика.

2. Разработка модуля передатчика (трансммитера), входящего в состав асинхронного приемопередатчика UART.

3. Разработка модуля приемника (ресивера), входящего в состав асинхронного приемопередатчика UART. Совместная проверка передатчика и приемника UART.

4. Построение синтезируемого описания и моделирования модуля асинхронной памяти FIFO

5. Разработка контроллера клавиатуры

6. Разработка простейшего арифметико-логического устройства (АЛУ)

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

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

## 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»;

«неаттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Неаттестован
ОПК-5	Знать -принципы реализации алгоритмов в параллельном виде и параллельного программирования, синтаксические конструкции, алфавит, типы данных, операторы HDL-языков (VHDL, VerilogHDL), методику применения HDL-языков для синтеза и моделирования радиоэлектронных устройств на инструментальном персональном компьютере	Знание учебного материала и готовность к его обсуждению и применению в рамках выполнения лабораторных работ	Выполнение работ в срок, предусмотренный в рабочей программе	Невыполнение работ в срок, предусмотренный в рабочей программе
	уметь -использовать пакеты прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных	Знание учебного материала и готовность к его обсуждению и применению в	Выполнение работ в срок, предусмотренный в рабочей программе	Невыполнение работ в срок, предусмотренный в рабочей программе

	программ свободного доступа для автоматизированной разработки проектов радиоэлектронных устройств на ПЛИС, в том числе написания программ на HDL-языках, синтеза программ на уровень логических вентилях, размещения и трассировки в кристалле, моделирования на функциональном поведенческом уровне и с учетом временных задержек реальных кристаллов	рамках выполнения лабораторных работ		
	владеть -практическими навыками программирования на языках VHDL и VerilogHDL, в том числе написания программ для синтеза в кристалл и моделирования во временной области радиоэлектронных устройств с использованием пакетов прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных программ свободного доступа	Знание учебного материала и готовность к его обсуждению и применению в рамках выполнения лабораторных работ	Выполнение работ в срок, предусмотренный в рабочей программе	Невыполнение работ в срок, предусмотренный в рабочей программе

### 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 двухбалльной системе:

«зачтено»

«незачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Незачтено
ОПК-5	Знать -принципы реализации алгоритмов в параллельном виде и параллельного программирования, синтаксические конструкции, алфавит, типы данных, операторы HDL-языков (VHDL, VerilogHDL), методику применения HDL-языков для синтеза и моделирования радиоэлектронных устройств на инструментальном персональном компьютере	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь -использовать пакеты прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных программ свободного доступа для автоматизированной разработки проектов радиоэлектронных устройств на ПЛИС, в том числе написания программ на HDL-языках, синтеза программ на уровень логических вентилях, размещения и трассировки в кристалле, моделирования на	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

функциональном поведенческом уровне и с учетом временных задержек реальных кристаллов			
владеть -практическими навыками программирования на языках VHDL и VerilogHDL, в том числе написания программ для синтеза в кристалл и моделирования во временной области радиоэлектронных устройств с использованием пакетов прикладных программ автоматизированного проектирования и моделирования из набора пакетов прикладных программ свободного доступа	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи решены

## 7.2. Примерный перечень оценочных средств (типичные 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

#### 1. Основные операции над данными языка VHDL.

##### Арифметические

- + ---- Сложение
- ---- Вычитание
- \* ---- Умножение
- / ---- Деление
- mod ---- Деление по модулю
- \*\* ---- Возведение в степень (например 2\*\*8 возвращает 256)

##### Логические

- NOT ---- Отрицание
- AND ---- Логическое И
- OR ---- Логическое ИЛИ
- = ---- Равенство
- /= ---- Не равенство
- < ---- Меньше
- <= ---- Меньше или равно
- > ---- Больше
- >= ---- Больше или равно

##### Битовые

- NOT ---- Побитная инверсия
- AND ---- Побитное логическое умножение
- OR ---- Побитное логическое сложение
- XOR ---- Побитная функция неравнозначности
- XNOR ---- Побитная функция равнозначности

## 2. Основные операции над данными языка Verilog HDL.

### Арифметические

+ // Сложение  
- // Вычитание  
\* // Умножение  
/ // Деление  
% // Деление по модулю  
\*\* // Возведение в степень

### Логические

! // Отрицание  
&& // Логическое И  
|| // Логическое ИЛИ  
== // Равенство  
!= // Неравенство  
< // Меньше  
<= // Меньше или равно  
> // Больше  
>= // Больше или равно

### Битовые

~ // Побитная инверсия  
& // Побитное логическое умножение  
| // Побитное логическое сложение  
^ // Побитная функция неравнозначности  
~^ // Побитная функция равнозначности

## 3. Условные операторы языка VHDL. Основные формы.

### Оператор if

```
if <condition> then <statement>  
elsif <condition> then <statement>  
else <statement>  
end if;
```

### Оператор case

```
case (<2-bit select>) is  
when "00" => <statement>;  
when "01" => <statement>;  
when "10" => <statement>;  
when "11" => <statement>;  
when others => <statement>;  
end case;
```

### Оператор select

```
with <choice_expression> select  
<name> <= <expression> when <choices>,  
      <expression> when <choices>,  
      <expression> when others;
```

### **Оператор условного присваивания**

```
<name> <= <expression> when <condition> else  
    <expression> when <condition> else  
    <expression>;
```

## 4. Условные операторы языка Verilog HDL. Основные формы.

### **Оператор if**

```
if (<condition>) begin <statement>; end  
else if (<condition>) begin <statement> end  
else begin <statement>; end
```

### **Оператор case**

```
case (<2-bit select>)  
2'b00 : begin <statement>; end  
2'b01 : begin <statement>; end  
2'b10 : begin <statement>; end  
2'b11 : begin <statement>; end  
default: begin <statement>; end  
endcase
```

### **Оператор непрерывного условного присваивания**

```
assign <output> = <1-bit_select> ? <input1> : <input0>;
```

## 5. Операторы повторения и размножения языка VHDL.

### **Оператор повторения в форме loop**

```
for <name> in <lower_limit> to <upper_limit> loop  
    <statement>;  
    <statement>;  
end loop;
```

### **Оператор повторения в форме while**

```
while <condition> loop  
    <statement>;  
    <statement>;  
end loop;
```

### **Оператор размножения generate**

```
for <name> in <lower_limit> to <upper_limit> generate  
    begin  
        <statement>;  
        <statement>;  
    end generate;
```

## 6. Операторы повторения и размножения языка Verilog HDL.

### **Оператор повторения в форме loop**

```
integer <var>;  
for (<var> = <initial_value>; <var> <= <final_value>; <var>=<var>+1)  
begin  
    <statement>;  
end
```

### **Оператор повторения в форме repeat**

```
repeat (<value>) begin  
    <statements>;  
end
```

### **Оператор повторения в форме while**

```
while (<condition>) begin  
    <statement>;  
end
```

### **Оператор повторения в форме forever**

```
forever begin  
    <statement>;  
end
```

### **Оператор размножения generate в форме loop**

```
genvar <var>;  
generate  
for (<var>=0; <var> < <limit>; <var>=<var>+1)  
begin: <label>  
    <instantiation>  
end  
endgenerate
```

### **Оператор размножения generate в форме case**

```
generate  
case (<constant_expression>)  
    <value>: begin: <label_1>  
        <code>  
    end  
    <value>: begin: <label_2>  
        <code>  
    end  
    default: begin: <label_3>  
        <code>  
    end  
endcase  
endgenerate
```

### 7.2.2. Примерный перечень заданий для решения стандартных задач

1. Описать на языке VHDL синтезируемую модель регистрового дешифратора 3 → 8 с разрешением дешифрирования.

#### Модель дешифратора на языке VHDL

```
process(clk)
begin
  if (clk'event and clk = '1') then
    if (reset = '1') then
      q <= b"00000000";
    elsif e='1' then
      case a is
        when "000" => q <= "00000001";
        when "001" => q <= "00000010";
        when "010" => q <= "00000100";
        when "011" => q <= "00001000";
        when "100" => q <= "00010000";
        when "101" => q <= "00100000";
        when "110" => q <= "01000000";
        when "111" => q <= "10000000";
        when others => q <= "00000000";
      end case;
    else q <= "00000000";
    end if;
  end if;
end process;
```

2. Описать на языке Verilog HDL синтезируемую модель регистрового дешифратора 3 → 8 с разрешением дешифрирования.

#### Модель регистрового дешифратора на языке Verilog HDL

```
reg [7:0] q;
wire [2:0] a;
always @(posedge clk)
  if (reset)
    q <= 8'h00;
  else if (e)
    case (<input>)
      3'b000 : q <= 8'b00000001;
      3'b001 : q <= 8'b00000010;
      3'b010 : q <= 8'b00000100;
      3'b011 : q <= 8'b00001000;
      3'b100 : q <= 8'b00010000;
      3'b101 : q <= 8'b00100000;
      3'b110 : q <= 8'b01000000;
      3'b111 : q <= 8'b10000000;
      default : q <= 8'b00000000;
    endcase
```

```
else q <= 8'b00000000;
```

3. Описать на языке VHDL модель комбинационного мультиплексора  $8 \rightarrow 1$  с разрешением мультиплексирования.

**Модель комбинационного мультиплексора на языке VHDL**

```
process (sel, e, input1, input2, input3, input4, input5, input6, input7, input8)
begin
if e='1' then
  case sel is
    when "000" => q <= input1;
    when "001" => q <= input2;
    when "010" => q <= input3;
    when "011" => q <= input4;
    when "100" => q <= input5;
    when "101" => q <= input6;
    when "110" => q <= input7;
    when "111" => q <= input8;
    when others => q <= input1;
  end case;
else q <= input1;
end if;
end process;
```

4. Описать на языке Verilog HDL модель комбинационного мультиплексора  $8 \rightarrow 1$  с разрешением мультиплексирования.

**Модель комбинационного мультиплексора на языке Verilog HDL**

```
always @(sel, e, input1, input2, input3, input4, input5, input6, input7, input8)
if (e)
  case (sel)
    3'b000: q = input1;
    3'b001: q = input2;
    3'b010: q = input3;
    3'b011: q = input4;
    3'b100: q = input5;
    3'b101: q = input6;
    3'b110: q = input7;
    3'b111: q = input8;
  endcase
else q = input1;
```

5. Описать на языке VHDL модель T-триггера, переключаемого по спадающему фронту, с разрешением тактирования и синхронным сбросом.

**Модель T-триггера на языке VHDL**

```
process (clk)
begin
if clk'event and clk='0' then
  if reset='1' then
    q <= '0';
  elsif clk_enable='1' then
```

```

    q <= not (q);
end if;
end if;
end process;

```

6. Описать на языке Verilog HDL модель Т-триггера, переключаемого по спадающему фронту, с разрешением тактирования и синхронным сбросом.

#### Модель Т-триггера на языке Verilog HDL

```

always @(negedge clk)
if (reset)
begin
q <= 1'b0;
end
else if (<clock_enable>) begin
q <= ~q;
end

```

### 7.2.3. Примерный перечень заданий для решения прикладных задач

1. Подготовить файл тестовых воздействий и выполнить моделирование синтезируемой модели регистрового дешифратора 3 → 8 с разрешением дешифрирования

#### Тестовое воздействие

```

ENTITY decoder_tb IS
END decoder_tb;

ARCHITECTURE behavior OF decoder_tb IS

COMPONENT decoder
PORT(
    clk : IN std_logic;
    reset : IN std_logic;
    e : IN std_logic;
    a : IN std_logic_vector(2 downto 0);
    q : OUT std_logic_vector(7 downto 0)
);
END COMPONENT;

--Inputs
signal clk : std_logic := '0';
signal reset : std_logic := '1';
signal e : std_logic := '0';
signal a : std_logic_vector(2 downto 0) := (others => '0');

--Outputs
signal q : std_logic_vector(7 downto 0);

-- Clock period definitions
constant clk_period : time := 25 ns;

BEGIN

```



```

        input7 : IN std_logic;
        input8 : IN std_logic;
        q : OUT std_logic);
    END COMPONENT;
--Inputs
    signal e : std_logic := '0';
    signal sel : std_logic_vector(2 downto 0) := (others => '0');
    signal input1 : std_logic := '0';
    signal input2 : std_logic := '1';
    signal input3 : std_logic := '0';
    signal input4 : std_logic := '1';
    signal input5 : std_logic := '0';
    signal input6 : std_logic := '1';
    signal input7 : std_logic := '0';
    signal input8 : std_logic := '1';

    --Outputs
    signal q : std_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)
    uut: multiplex PORT MAP (
        e => e,
        sel => sel,
        input1 => input1,
        input2 => input2,
        input3 => input3,
        input4 => input4,
        input5 => input5,
        input6 => input6,
        input7 => input7,
        input8 => input8,
        q => q);
    -- Stimulus process
    stim_proc: process
    begin
        wait for 250 ns;
        sel <= b"001";
        wait for 100 ns;
        sel <= b"010";
        wait for 100 ns;
        sel <= b"011";
        wait for 100 ns;
        sel <= b"100";
        wait for 100 ns;
        sel <= b"101";
        wait for 100 ns;
        sel <= b"110";
        wait for 100 ns;
        sel <= b"111";
        wait for 100 ns;
        sel <= b"000";
        wait;
    end process;

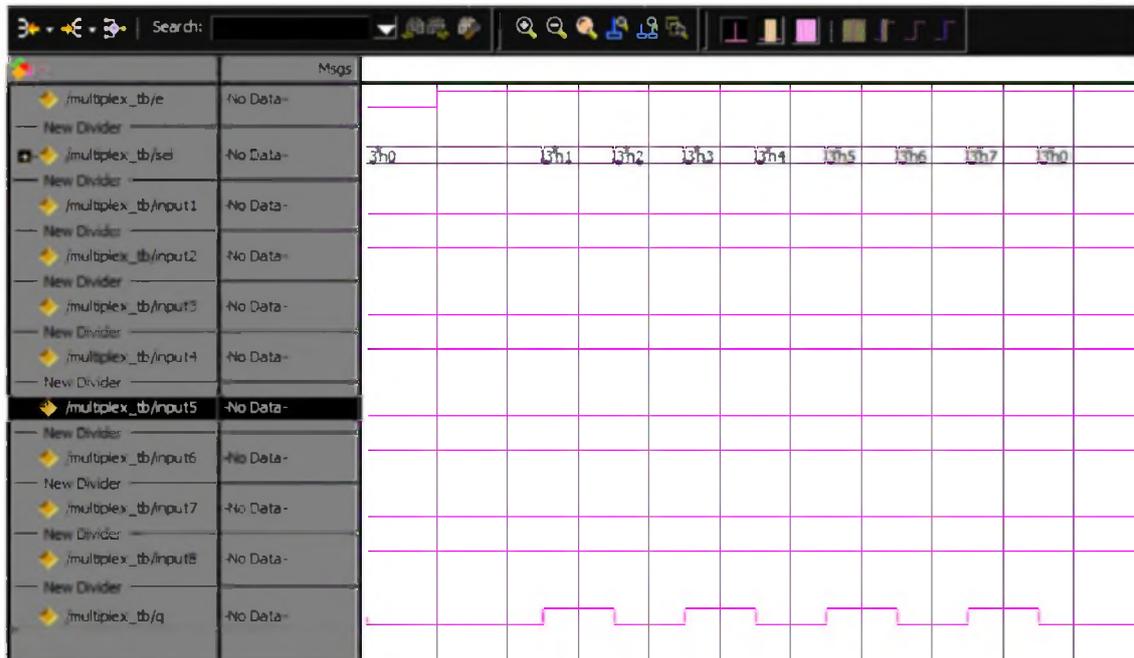
```

```

end process;
e <= '1' after 100 ns;
END;

```

### Временная диаграмма работы мультиплексора



3. Подготовить файл тестовых воздействий и выполнить моделирование синтезируемой модели Т-триггера, переключаемого по спадающему фронту, с разрешением тактирования и синхронным сбросом.

#### Тестовое воздействие

```

ENTITY T_flip_flop_tb IS
END T_flip_flop_tb;

```

```

ARCHITECTURE behavior OF T_flip_flop_tb IS
-- Component Declaration for the Unit Under Test (UUT)

```

```

COMPONENT T_flip_flop
  PORT(reset : IN std_logic;
        clk : IN std_logic;
        clk_enable : IN std_logic;
        q : OUT std_logic);
END COMPONENT;

```

```

--Inputs
signal reset : std_logic := '1';
signal clk : std_logic := '0';
signal clk_enable : std_logic := '0';

```

```

--Outputs
signal q : std_logic;

```

```

-- Clock period definitions

```

```

constant clk_period : time := 20 ns;

BEGIN
-- Instantiate the Unit Under Test (UUT)
  uut: T_flip_flop PORT MAP (reset => reset,
                             clk => clk,
                             clk_enable => clk_enable,
                             q => q);

-- Clock process definitions
  clk_process :process
  begin
    clk <= '0';
    wait for clk_period/2;
    clk <= '1';
    wait for clk_period/2;
  end process;

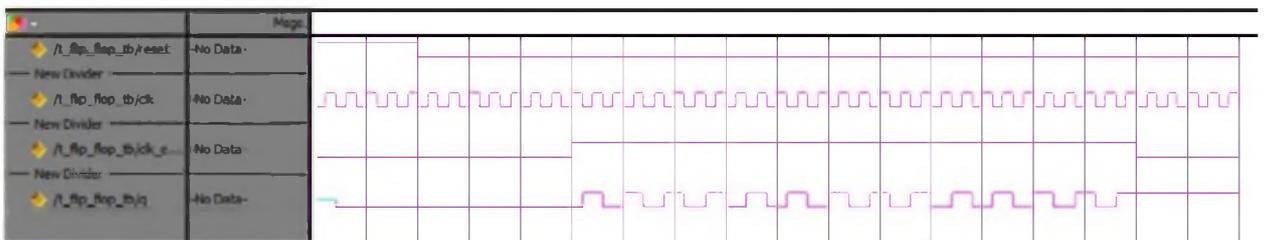
  reset <= '0' after 100 ns;

-- Stimulus process
  stim_proc: process
  begin
    wait for 250 ns;
    clk_enable <= '1';
    wait for 550 ns;
    clk_enable <= '0';
  end process;

END;

```

### Временная диаграмма работы Т-триггера



4. Подготовить файл тестовых воздействий и выполнить моделирование синтезируемой модели двоичного реверсивного счетчика разрядности 8, переключаемого по нарастающему фронту.

#### Тестовое воздействие

```

ENTITY reverse_counter_tb IS
END reverse_counter_tb;

```

## ARCHITECTURE behavior OF reverse\_counter\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

```
COMPONENT reverse_counter
PORT(reset : IN std_logic;
      clk : IN std_logic;
      direction : IN std_logic;
      q : OUT std_logic_vector(7 downto 0));
END COMPONENT;
```

--Inputs

```
signal reset : std_logic := '1';
signal clk : std_logic := '0';
signal direction : std_logic := '1';
```

--Outputs

```
signal q : std_logic_vector(7 downto 0);
```

-- Clock period definitions

```
constant clk_period : time := 20 ns;
```

BEGIN

-- Instantiate the Unit Under Test (UUT)

```
 uut: reverse_counter PORT MAP (
    reset => reset,
    clk => clk,
    direction => direction,
    q => q);
```

-- Clock process definitions

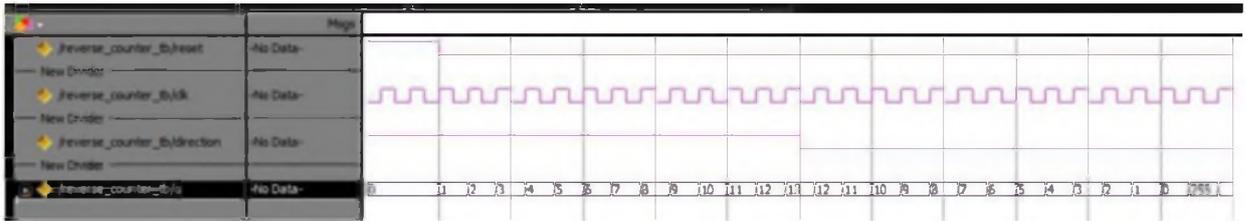
```
clk_process :process
begin
    clk <= '0';
    wait for clk_period/2;
    clk <= '1';
    wait for clk_period/2;
end process;
```

```
reset <= '0' after 50 ns;
```

-- Stimulus process

```
stim_proc: process
begin
    wait for 300 ns;
    direction <= '0';
    wait for 600 ns;
    direction <= '1';
    wait;
end process; END;
```

**Временная диаграмма работы реверсивного счетчика**



5. Подготовить файл тестовых воздействий и выполнить моделирование синтезируемой модели делителя частоты на 12 со скважностью на выходе 50 %.

### Тестовое воздействие

```
ENTITY delitel_12_tb IS
END delitel_12_tb;
```

```
ARCHITECTURE behavior OF delitel_12_tb IS
```

```
-- Component Declaration for the Unit Under Test (UUT)
```

```
COMPONENT delitel_12
PORT(
  reset : IN std_logic;
  clk : IN std_logic;
  clk_12 : OUT std_logic
);
END COMPONENT;
```

```
--Inputs
```

```
signal reset : std_logic := '1';
signal clk : std_logic := '0';
```

```
--Outputs
```

```
signal clk_12 : std_logic;
```

```
-- Clock period definitions
```

```
constant clk_period : time := 20 ns;
```

```
BEGIN
```

```
-- Instantiate the Unit Under Test (UUT)
```

```
 uut: delitel_12 PORT MAP (
   reset => reset,
   clk => clk,
   clk_12 => clk_12);
```

```
-- Clock process definitions
```

```
clk_process :process
```

```
begin
```

```
  clk <= '0';
```

```
  wait for clk_period/2;
```

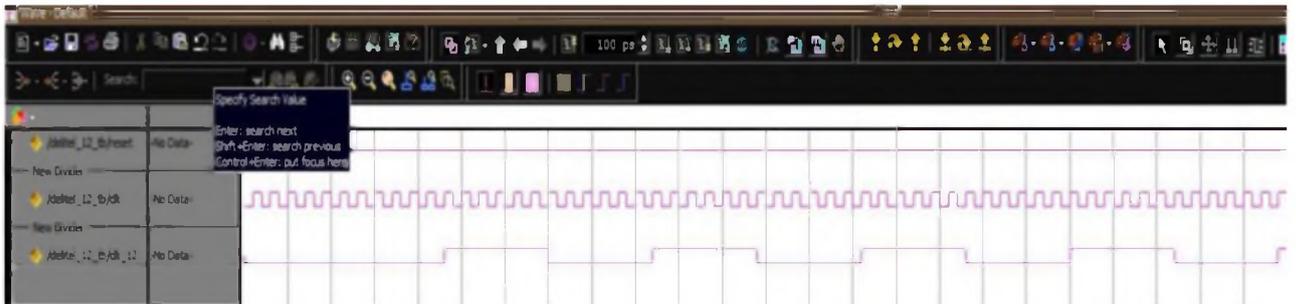
```
  clk <= '1';
```

```
wait for clk_period/2;  
end process;
```

```
reset <= '0' after 100 ns;
```

```
END;
```

### Временная диаграмма работы делителя частоты на 12



6. Подготовить файл тестовых воздействий и выполнить моделирование синтезируемой модели реверсивного сдвигового регистра разрядности 8, тактируемого по нарастающему фронту.

#### Тестовое воздействие

```
ENTITY shift_reg_tb IS  
END shift_reg_tb;
```

```
ARCHITECTURE behavior OF shift_reg_tb IS
```

```
-- Component Declaration for the Unit Under Test (UUT)
```

```
COMPONENT shift_reg  
PORT(  
    reset : IN std_logic;  
    clk : IN std_logic;  
    direction : IN std_logic;  
    s_out : OUT std_logic_vector(7 downto 0)  
);  
END COMPONENT;
```

```
--Inputs
```

```
signal reset : std_logic := '1';  
signal clk : std_logic := '0';  
signal direction : std_logic := '0';
```

```
--Outputs
```

```
signal s_out : std_logic_vector(7 downto 0);
```

```
-- Clock period definitions
```

```
constant clk_period : time := 20 ns;
```

```
BEGIN
```

```

-- Instantiate the Unit Under Test (UUT)
 uut: shift_reg PORT MAP (
     reset => reset,
     clk => clk,
     direction => direction,
     s_out => s_out
 );

-- Clock process definitions
 clk_process :process
 begin
     clk <= '0';
     wait for clk_period/2;
     clk <= '1';
     wait for clk_period/2;
 end process;

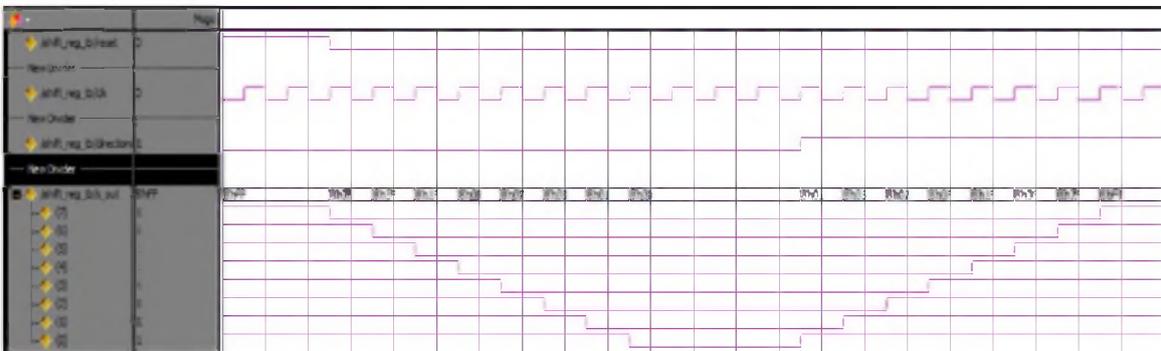
reset <= '0' after 50 ns;

direction <= '1' after 270 ns;

END;

```

### Временная диаграмма работы реверсивного сдвигового регистра



#### 7.2.4. Примерный перечень вопросов для подготовки к зачету

1. Классификация цифровых микросхем. Место микросхем программируемой логики в общем процессе проектирования.
2. Основные производители микросхем ПЛИС. Типы архитектур ПЛИС. Понятие о языках описания аппаратуры (HDL-языках) VHDL и Verilog.
3. HDL-языки для синтеза и моделирования. Понятие о параллельных алгоритмах и процессах. Принципы моделирования параллельных процессов на последовательной инструментальной ЭВМ. Понятие о процессе синтеза схем по текстовому формализованному описанию (Синтезатор ↔ Компилятор).
4. Первая HDL-«программа». Entity и архитектурные тела VHDL.

Проектный модуль Verilog. Стили описания проектов. Структура и поведение.

5. Структура и поведение. Методика структурного описания проектов. Методика поведенческого описания проектов. Смешанное описание.

6. Типы и подтипы данных, физические типы языка VHDL. Сигналы, переменные и константы, атрибуты сигналов в VHDL.

7. Типы языка Verilog. Типы reg и wire.

8. Понятие сигнала и переменной в VHDL. Переменная в Verilog. Блокирующее и неблокирующее присваивания в Verilog.

7. Оператор Process языка VHDL. Оператор always языка Verilog.

8. Отличия переменных в языках VHDL и Verilog. Соединения языка Verilog. Сходства и отличия переменных в сравнении с процедурными языками программирования.

9. Место и области видимости сигналов и переменных в программном модуле языков VHDL и Verilog. Препроцессор языка Verilog.

10. Операторы присваивания VHDL и Verilog. Модели и синтаксис задержек transport и inertial в VHDL. Модели и синтаксис задержек в Verilog.

11. Операторы условия (if) и выбора (case) в VHDL и Verilog. Отличие операторов if и case HDL-языков от аналогичных операторов процедурных языков программирования.

12. Операторы повторения (loop) VHDL и цикла Verilog. Операторы ожидания (wait) VHDL и Verilog.

13. Параллельное сигнальное присваивание VHDL. Непрерывное присваивание Verilog.

14. Операторы Process и Block VHDL. Оператор инициализации Verilog. Оператор fork Verilog.

15. Структурное описание проекта. Оператор вхождения компонента VHDL. Оператор конкретизации Verilog.

16. Оператор размножения generate VHDL и Verilog. Связь с операторами вхождения компонента VHDL и конкретизации Verilog.

17. Функции и процедуры VHDL. Их параллельный и последовательный вызов. Особенности функций Verilog.

18. Примеры описания комбинационных схем на VHDL и Verilog. Логика, мультиплексоры, шифраторы, дешифраторы. Уровневые (прозрачные) защелки (Latch) и их «ошибочное» появление при описании комбинационной логики.

19. Примеры описания последовательных схем на VHDL и Verilog. Триггеры (регистры), счетчики, память. Описание конечных автоматов (FSM). Синхронные и асинхронные схемы. Схемы синхронизации.

20. Реализация арифметических функций на VHDL и Verilog.

21. Testbench VHDL (Test fixture Verilog) и методика его создания с помощью языковых конструкций. Имитация периодических и однократных воздействий.

22. Использование процедур VHDL для описания поведения сигналов. Операторы вывода VHDL и Verilog и их применение при моделировании. Оператор проверки Assert языка VHDL.

### 7.2.5. Примерный перечень вопросов для подготовки к экзамену Непредусмотрен учебным планом

### 7.2.6. Методика выставления оценки при проведении промежуточной аттестации

*Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.*

*1. Оценка «незачтено» ставится в случае, если студент набрал менее 6 баллов*

*2. Оценка «зачтено» ставится в случае, если студент набрал от 6 до 20 баллов*

### 7.2.7. Паспорт оценочных материалов

№п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение. Архитектура и особенности микросхем ПЛИС на примере конкретных семейств	ОПК-5	Тест, защита лабораторных работ
2	Языки VHDL и Verilog. VHDL и Verilog для синтеза и моделирования. Структура проекта на VHDL. Структура проекта на Verilog.	ОПК-5	Тест, защита лабораторных работ
3	Структурные и поведенческие архитектурные тела VHDL и проектные модули Verilog.	ОПК-5	Тест, защита лабораторных работ
4	Сигналы и переменные в VHDL. Блокирующее и неблокирующее присваивание в Verilog. Последовательные операторы VHDL и Verilog.	ОПК-5	Тест, защита лабораторных работ
5	Параллельные операторы VHDL и Verilog. Процедуры и функции языка VHDL. Функции языка Verilog.	ОПК-5	Тест, защита лабораторных работ
6	Описание на VHDL и Verilog типовых цифровых устройств. Методика моделирования цифровых устройств, описанных на VHDL и Verilog.	ОПК-5	Тест, 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 методик и 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

№ пп	Авторы, составители, год издания	Заглавие	Вид издания	Обеспеченность
<b>1. Основная литература</b>				
1	<u>Бибило</u> П.Н., 2017 г.	VHDL. Эффективное использование при проектировании цифровых систем [Электронный ресурс] / П. Н. Бибило, Н. А. Авдеев. - VHDL. Эффективное использование при проектировании цифровых систем ; 2021-05-25. - Москва : СОЛОН-ПРЕСС, 2017. - 342 с. - Гарантированный срок размещения в ЭБС до 25.05.2021 (автопродлонгация). - ISBN 5-98003-293-2. URL: <a href="http://www.iprbookshop.ru/90406.html">http://www.iprbookshop.ru/90406.html</a>	Эл.	1,0
2	<u>Бибило</u> П.Н., 2016 г.	Основы языка VHDL [Электронный ресурс] / П. Н. Бибило. - Основы	Эл.	1,0

		языка VHDL ; 2021-05-25. - Москва : СОЛОН-Р, 2016. - 200 с. - Гарантированный срок размещения в ЭБС до 25.05.2021 (автопродлонгация). - ISBN 5-93455-056-X.URL: <a href="http://www.iprbookshop.ru/90427.html">http://www.iprbookshop.ru/ 90427.html</a>		
3	Соловьев В.В., 2020	Язык Verilog в проектировании встраиваемых систем на FPGA. –М.: Горячая линия – Телеком, 2020. – 440 с.	Электр.	
2. Дополнительная литература				
1	Суворова Е.А., Шейнин Ю.Б., 2003 г.	Проектирование цифровых систем на VHDL. – СПб: БХВ- Петербург, 2003. – 576 с.	Электр.	-
2	Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П., 2002 г.	Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.	Электр.	-
3	Поляков А.К., 2010	Языки VHDL и Verilog в проектировании цифровой аппаратуры. – М.: СОЛОН-Пресс, 2010. – 320 с.	Электр.	
3. Методическая литература				
1	Максимов Д.А., 2011 г.	Методические указания для выполнения лабораторных работ № 1-3 по дисциплине "Современные системы проектирования РЭС" для студентов спец. 200700 "Радиотехника" очной формы обучения	Электр.	

2	Максимов Д.А., 2012 г.	Методические указания для выполнения лабораторных работ № 4-6 по дисциплине "Современные системы проектирования РЭУС" для студентов спец. 200700 "Радиотехника" очной формы обучения	Электр.	-
---	---------------------------	--	---------	---

**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:**

-Программы автоматизированного проектирования и моделирования радиоэлектронных устройств из набора пакетов прикладных программ свободного доступа.

- Электронный ресурс <https://opensource.org/>

## **9.МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Дисплейный класс с необходимым оборудованием.

## **10.МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Современные системы проектирования РЭС».

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.

Контроль усвоения материала дисциплины производится тестированием, проверкой выполнения лабораторных работ, защитой лабораторных работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать 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 работа	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.
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"> <li>- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;</li> <li>- выполнение домашних заданий и расчетов;</li> <li>- работа над темами для самостоятельного изучения;</li> <li>- участие в работе студенческих научных конференций, олимпиад;</li> <li>- подготовка к промежуточной аттестации.</li> </ul>
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 три дня эффективнее всего использовать для повторения и систематизации материала.