

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)



УТВЕРЖДАЮ

Заведующий кафедрой

физики полупроводников и микроэлектроники

(Е.Н.Бормонтов)

31.08.2024

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.ДВ.04.01 Архитектура и программирование ПЛИС

1. Код и наименование направления подготовки/специальности:  
03.04.03 Радиофизика
2. Профиль подготовки: Интегральная элементная база телекоммуникационных технологий
3. Квалификация (степень) выпускника: магистр
4. Форма образования: очная
5. Кафедра, отвечающая за реализацию дисциплины: физики полупроводников и микроэлектроники
6. Составители программы: Богатилов Евгений Васильевич,  
кандидат физико-математических наук, доцент
7. Рекомендована: НМС физического факультета протокол №6 от 26.06.2024
8. Учебный год: 2025-2026 Семестр: 3
9. Цели и задачи учебной дисциплины: целью освоения учебной дисциплины является формирование специальных знаний о применении языка Verilog для разработки цифровых устройств на базе ПЛИС.  
Задачи учебной дисциплины:
  - изучение номенклатуры современных семейств ПЛИС;
  - изучение основ архитектуры популярных семейств ПЛИС;
  - изучение синтаксиса языка Verilog;

- изучение метода проектирования конечных автоматов с использованием языка Verilog;
- изучение методов функциональной верификации Verilog-описаний цифровых устройств на базе ПЛИС;
- приобретение умений и навыков проектирования цифровых устройств на базе современных семейств ПЛИС с помощью САПР.

**10. Место учебной дисциплины в структуре ОПОП:** дисциплина относится к части, формируемой участниками образовательных отношений (вариативная) блока Б1, дисциплины по выбору.

В результате прохождения данной дисциплины обучающийся должен приобрести знания, умения, навыки общепрофессиональных компетенций, необходимые для обеспечения следующих трудовых функций:

- G/03.7 Математическое и компьютерное моделирование составных частей радиоэлектронных средств (профессиональный стандарт 06.048).

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:**

Компетенции		Индикаторы		Планируемые результаты обучения
Код	Наименование компетенции	Код(ы)	Наименование индикатора(ов)	
ПК-1	Способен принимать участие в разработке и научных исследованиях систем связи и телекоммуникаций	ПК-1.1	ПК-1.1 Владеет фундаментальными знаниями в области систем связи и телекоммуникаций	<p><i>Знать:</i></p> <ul style="list-style-type: none"> <li>- архитектуру современных семейств ПЛИС;</li> <li>- области применения ПЛИС в задачах радиофизики.</li> </ul>
		ПК-1.3	Проводит анализ известных технических решений отдельных блоков систем связи, телекоммуникаций и радионавигации	<p><i>Знать:</i></p> <ul style="list-style-type: none"> <li>- принципы разработки HDL-описаний цифровых устройств.</li> </ul> <p><i>Уметь:</i></p> <ul style="list-style-type: none"> <li>- применять язык Verilog для создания синтезируемых HDL-описаний цифровых устройств по техническому заданию.</li> <li>- разрабатывать методику тестирования и проводить функциональную верификацию HDL-описания.</li> </ul> <p><i>Владеть:</i></p> <ul style="list-style-type: none"> <li>- навыками конфигурирования ПЛИС и тестирования законченного цифрового устройства на базе ПЛИС.</li> </ul>

**12. Объем дисциплины в зачетных единицах/часах — 2 / 72.**

Форма промежуточной аттестации – зачет.

### 13. Виды учебной работы:

Вид учебной работы		Трудоемкость (часы)	
		Всего	По семестрам 2 сем.
Аудиторные занятия,		38	38
в том числе:	лекции	26	26
	практические занятия	12	12
Самостоятельная работа		34	34
Итого:		72	72

#### 13.1. Содержание дисциплины:

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>Лекции</b>		
1.1	Основы архитектуры ПЛИС	Классификация ПЛИС: PLA, CPLD, FPGA. Области применения ПЛИС. Мелко-, средне-, крупномодульные архитектуры. Устройства на основе мультиплексоров, устройства на основе таблиц соответствия. Секции и логические ячейки. Конфигурируемые логические блоки CLB. Блоки логических массивов LAB. Распределенное ОЗУ и сдвиговые регистры. Схемы ускоренного переноса. Встроенные блоки ОЗУ. Встроенные умножители, сумматоры, блоки умножения с накоплением. Аппаратные и программные встроенные микропроцессорные ядра. Дерево синхронизации и диспетчеры синхронизации. Ввод-вывод общего назначения. Гигабитные приемопередатчики.
1.2	Основы языка Verilog	Структура Verilog-проекта. Модули. Сигналы типа reg и wire, значения сигналов 'x' и 'z'. Типы портов. Аппаратное представление двунаправленного порта. Правила задания типов сигналов при подключении к портам. Типы данных в Verilog. Арифметические, логические, побитовые и другие операции в языке Verilog. Деление операторов на параллельные и последовательные. Оператор assign. Использование тернарного оператора. Оператор always. Виды условий запуска в операторе always. Последовательные операторы. Операторы блокирующего и неблокирующего присваивания. Операторы ветвления и выбора. Условия появления защелок. Операторы цикла. Структурное и поведенческое описание устройства. Оператор подключения экземпляра модуля.
1.3	Логический синтез и верификация HDL-описания	Понятие синтезруемости HDL-описания. Операция логического синтеза. Статический временной анализ. Понятие функциональной верификации HDL-описания, принципы построения тестбенчей средствами языка Verilog. Средства языка SystemVerilog для верификации. Организация тестового окружения. Параллельные тестовые окружения. Утверждения и последовательности. Функциональное покрытие.
1.4	Параметризуемые модели в Verilog	Параметры и локальные параметры. Переопределение параметров. Макроопределение `define. Символ #. Макроопределение `defparams. Генерация однотипных компонентов с помощью цикла generate и индексной переменной genvar. Условная генерация компонентов.

1.5	Согласование тактовых доменов	Понятие гонки сигналов. Понятие метастабильности триггеров. Схемы для экспериментального наблюдения состояний метастабильности триггеров. Устранение метастабильных состояний с помощью сдвиговых регистров.
1.6	Использование конечных автоматов в Verilog	Конечные автоматы Мура и конечные автоматы Мили. Представление конечных автоматов в виде графов и таблиц переходов. Реализация структуры конечного автомата в Verilog: логика переходов, регистр состояния, логика формирования выходных сигналов. Примеры применения конечных автоматов.
1.7	Конвейерная обработка данных на Verilog	Базовые идеи конвейеризации. Конвейерные идеализмы: одинаковая задержка стадий, унифицированные типы запросов, независимость запросов. Синхронизация стадий конвейера с переменными задержками. Сравнение комбинационной, многотактной и конвейерной реализаций на примере арифметического блока.
<b>Практические занятия</b>		
2.1	Основы архитектуры ПЛИС	
2.2	Основы языка Verilog	Занятие № 1. Разработка модели комбинационного устройства в Quartus.
2.3	Логический синтез и верификация HDL-описания	Занятие № 2. Функциональная верификация комбинационного устройства с помощью Modelsim.
2.4	Параметризуемые модели в Verilog	Занятие № 3. Разработка и верификация параметризованной модели таймера.
2.5	Согласование тактовых доменов	Занятие № 4. Реализация SPI-интерфейса для ведомого устройства
2.6	Использование конечных автоматов в Verilog	Занятие № 5. Реализация SPI-интерфейса ведущего устройства в виде конечного автомата на Verilog
2.7	Конвейерная обработка данных на Verilog	Занятие № 6. Конвейерная реализация арифметических действий

### 13.2. Темы (разделы) дисциплины и виды занятий:

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)			
		Лекции	Практические	Самостоятельная работа	Всего
1	Основы архитектуры ПЛИС	8	-	6	14
2	Основы языка Verilog	6	2	6	14
3	Логический синтез и верификация HDL-описания	4	2	6	12
4	Параметризуемые модели в Verilog	2	2	4	8
5	Согласование тактовых доменов	2	2	4	8
6	Использование конечных автоматов в Verilog	2	2	4	8
7	Конвейерная обработка данных на Verilog	2	2	4	8
	Итого:	26	12	34	72
	Итого по курсу				72

### 14. Методические указания для обучающихся по освоению дисциплины:

Изучение дисциплины «Архитектура и программирование ПЛИС» предусматривает осуществление учебной деятельности состоящей из двух частей: обучения студентов преподавателем и самостоятельной учебной деятельности студентов по изучению дисциплины.

Метод преподавания – проблемный, форма обучения – групповая, форма общения – интерактивная. Обязательное посещение практических занятий и текущих аттестаций.

Подготовка к лекциям является одним из видов самостоятельной работы студентов. Студентам, чтобы хорошо овладеть учебным материалом, необходимо выработать навыки правильной и планомерной работы. Перед началом лекционных занятий надо просмотреть все, что было сделано в предыдущий раз. Это позволит сосредоточить внимание и восстановить в памяти уже имеющиеся знания по данному предмету. Кроме того, такой метод поможет лучше запомнить как старое, так и новое, углубит понимание того и другого, так как при этом устанавливаются связи нового со старым, что является не только обязательным, но и основным условием глубокого овладения материалом. Чем детальнее изучаемое ассоциируется с известным ранее, тем прочнее сохраняется в памяти и быстрее вспомнится, когда требуется.

Следует помнить, что через лекцию передается не только систематизированный теоретический материал, но и постигается методика научного исследования и умение самостоятельно работать, анализировать различного рода явления.

Записывать на лекции необходимо главное, не стремиться зафиксировать все слово в слово. Выбрать же главное без понимания предмета невозможно. Наличие собственного конспекта лекций позволяет еще раз ознакомиться, продумать, разобраться в новом материале, так как недостаточно хорошо понятые во время лекции положения могут быть восстановлены в памяти, сопоставлены с другими, додуманы, дополнены, уяснены и расширены с помощью учебной литературы. Записи являются пособиями для повторения, дают возможность охватить содержание лекции и всего курса в целом.

При этом хорошо овладеть содержанием лекции – это:

- знать тему;
  - понимать значение и важность ее в данном курсе;
  - четко представлять план;
  - уметь выделить основное, главное;
  - усвоить значение примеров и иллюстраций;
  - связать вновь полученные сведения о предмете или явлении с уже имеющимися;
  - представлять возможность и необходимость применения полученных сведений.
- Существует несколько общих правил работы на лекции:
- лекции по каждому предмету записывать удобнее в отдельных тетрадях, оставляя широкие поля для пометок;
  - к прослушиванию лекций следует готовиться, что позволит в процессе лекции отделить главное от второстепенного;
  - лекции необходимо записывать с самого начала, так как оно часто бывает ключом ко всей теме;
  - так как дословно записать лекцию невозможно, то необходимо в конспекте отражать: формулы, определения, схемы, трудные места, мысли, примеры, факты и положения от которых зависит понимание главного, новое и незнакомое, неопубликованные данные, материал отсутствующий в учебниках и т.п.;
  - записывать надо сжато;
  - во время лекции важно непрерывно сохранять рабочую установку, умственную активность.

Самостоятельная работа студентов наряду с аудиторной представляет одну из форм учебного процесса и является существенной ее частью. Самостоятельная работа предназначена для формирования навыков самостоятельной работы как вообще,

так и в учебной, научной деятельности. Она обеспечивает формирование и развитие способности принимать на себя ответственность, самостоятельно решать проблему, находить конструктивные решения, выход из кризисной ситуации и т.д.

Самостоятельная работа формирует самостоятельность не только как совокупность умений и навыков, но и как черту характера, играющую существенную роль в структуре личности современного специалиста высшей квалификации. Она воспитывает самостоятельность как черту характера. Никакие знания, полученные на уровне пассивного восприятия, не ставшие объектом собственной умственной или практической работы, не могут считаться подлинным достоянием человека.

Давая возможность расширять и обогащать знания, умения по индивидуальным направлениям, самостоятельная работа студента позволяет создать разносторонних специалистов. В процессе самостоятельной работы развивают творческие возможности обучающегося, при этом самостоятельная работа завершает задачи всех видов учебной работы.

Самостоятельная работа - это планируемая работа студентов, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Преподаватель, ведущий занятия, организует, направляет самостоятельную работу студентов и оказывает им необходимую помощь. Однако самостоятельность студентов должна превышать объем работы, контролируемой преподавателем работы, и иметь в своей основе индивидуальную мотивацию обучающегося по получению знаний, необходимых и достаточных для будущей профессиональной деятельности в избранной сфере. Преподаватель при необходимости может оказывать содействие в выработке и коррекции данной мотивации, лежащей в основе построения самостоятельной деятельности студента по изучению дисциплины, получению необходимых знаний и навыков.

Основой самостоятельной работы служит научно-теоретический курс, комплекс полученных студентом знаний. Основной, наиболее экономичной формой получения и усвоения информации, теоретических знаний в вузе является лекция, позволяющая воспринять значительную сумму основных знаний и потому способствующая повышению продуктивности всех других форм учебного труда.

Результат обучения и самостоятельной работы студента предполагает наличие следующих составляющих: понимание методологических основ построения изучаемых знаний; выделение главных структур учебного курса; формирование средств выражения в данной области; построение методик решения задач и ориентации в проблемах (ситуациях).

Самостоятельная работа студента при изучении курса «Архитектура и программирование ПЛИС» включает в себя: подготовку и участие в изучении теоретической части курса, подготовку к практическим занятиям, подготовку к зачету.

Самостоятельная работа студента при изучении дисциплины «Архитектура и программирование ПЛИС» включает в себя:

изучение теоретической части курса	- 12 часов;
подготовка к практическим занятиям	- 12 часов;
подготовка к зачету	- 10 часа;

Итого - 34 часа.

## **15. Перечень основной и дополнительной литературы, ресурсов Интернет, необходимых для освоения дисциплины:**

а) основная литература:

№ п/п	Источник
-------	----------

1	Угрюмов Е. П. Цифровая схемотехника : [учебное пособие для студ. вузов] / Е.П. Угрюмов .— 3-е изд. — Санкт-Петербург : БХВ-Петербург, 2010 .— 797 с.
---	--

б) дополнительная литература:

№ п/п	Источник
2	Дональд, Т. Логическое проектирование и верификация систем на SystemVerilog : монография / Дональд Т. Москва : ДМК-пресс, 2019. - 384 с. <a href="https://www.studentlibrary.ru/book/ISBN9785970606193.html">https://www.studentlibrary.ru/book/ISBN9785970606193.html</a>
3	Харрис Д. М. Цифровая схемотехника и архитектура компьютера = Digital design and computer architecture : пер. с англ. / Дэвид М. Харрис, Сара Л. Харрис .— Москва : ДМК Пресс, 2017 .— 791 с.
4	Поляков А. К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры / А. К. Поляков .— М. : Солон-пресс, 2003 .— 313 с.
5	Тарасов И. Е. Разработка цифровых устройств на основе ПЛИС Xilinx с применением языка VHDL / И.Е. Тарасов .— М. : Горячая линия-Телеком, 2005 .— 252 с.
6	Зотов В. Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы XILINX / В.Ю. Зотов .— М. : Горячая линия-Телеком, 2006 .— 519 с.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
7	Intel® FPGA Development Tools Support <URL: <a href="https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/support.html">https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/support.html</a> >
8	МАРСОХОД Open Source Hardware Project <URL: <a href="https://marsohod.org">https://marsohod.org</a> >

**16. Учебно-методическое обеспечение для организации самостоятельной работы:**

№ п/п	Источник
1	Язык Verilog и проектирование цифровых устройств на ПЛИС : учебно-методическое пособие / Е.В. Богатиков, А.Н. Шебанов .— Воронеж : Издательский дом ВГУ, 2018 .— 60 с.

**17. Информационные технологии, используемые для реализации учебного процесса по дисциплине:**

№ п/п	Источник
1	<a href="http://www.lib.vsu.ru">http://www.lib.vsu.ru</a> – ЗНБ ВГУ
2	<a href="https://edu.vsu.ru">https://edu.vsu.ru</a> – Образовательный портал "Электронный университет ВГУ"
3	Федеральный портал «Российское образование» <a href="http://www.edu.ru">http://www.edu.ru</a>

**18. Материально-техническое обеспечение дисциплины:**

Лекционные занятия проводятся в мультимедийном кабинете кафедры ФППИМЭ, оснащённым стационарным мультимедийным проектором AcerX125H – 1 шт., ноутбук emachines e510 – 1 шт., экран, с лицензионным программным обеспечением Microsoft Windows 7, Windows 10 договор 3010-15/207-19 от 30.04.2019; ПО Kaspersky Endpoint Security, договор 3010-07/04-20 от 27.01.2020.

Для проведения практических занятий необходима лаборатория со следующим оборудованием: отладочные комплекты ПЛИС Altera MAX II - 8 шт., компьютеры Lenovo

V520-15IKL - 8 шт., цифровые осциллографы UTD2025CL - 6 шт., функциональные генераторы UTG2025A - 3 шт., источники питания QJ1503C – 3 шт., мультиметры цифровые DM3058E – 3 шт., телевизор LED 48” – 1 шт.

Аудитория для самостоятельной работы студентов оснащена сервером на базе 2-х процессоров Xeon E5-2620 v3. – 1 шт., компьютеры HP ProDesk 400 G6 SFF – 9 шт., компьютеры Pentium Dual Core - 2 шт. , подключенные к сети Интернет с обеспечением доступа к электронной информационно-образовательной среде ВГУ и лицензионным программным обеспечением: Microsoft Windows 7, Windows 10, договор 3010-15/207-19 от 30.04.2019; ПО Kaspersky Endpoint Security, договор 3010-07/04-20 от 27.01.2020.

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется через образовательный портал "Электронный университет ВГУ".

## 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины	Компетенция	Индикаторы достижения компетенции	Оценочные средства
1	Основы архитектуры ПЛИС	ПК-1	ПК-1.1	Тестовые задания, перечень вопросов
2	Основы языка Verilog	ПК-1	ПК-1.3	Тестовые задания, перечень вопросов
3	Логический синтез и верификация HDL-описания	ПК-1	ПК-1.1	Тестовые задания, перечень вопросов
4	Параметризуемые модели в Verilog	ПК-1	ПК-1.3	Тестовые задания, перечень вопросов
5	Согласование тактовых доменов	ПК-1	ПК-1.1	Тестовые задания, перечень вопросов
6	Использование конечных автоматов в Verilog	ПК-1	ПК-1.1	Тестовые задания, перечень вопросов
7	Конвейерная обработка данных на Verilog	ПК-1	ПК-1.3	Тестовые задания, перечень вопросов
Промежуточная аттестация: форма контроля – зачет				Вопросы к зачету

## 20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

### 20.1 Текущий контроль успеваемости

Текущий контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: тестовые задания, результаты выполнения практических заданий, ответы на вопросы.

#### Примеры тестовых заданий

1. При помощи какого оператора языка Verilog можно изменять значение сигнала типа wire?

Варианты ответа:



- A) assign
- B) always
- C) initial
- D) case
- E) while

Правильный ответ: A.

2. Какие из операторов языка Verilog являются параллельными?

Варианты ответа:

- A) assign
- B) always
- C) case
- D) if
- E) for

Правильный ответ: A, B.

3. Какие операторы языка Verilog являются последовательными?

Варианты ответа:

- A) case
- B) for
- C) always
- D) assign
- E) initial

Правильный ответ: A, B.

4. Какая языковая конструкция Verilog не является синтезабельной?

Варианты ответа:

- A) always #(5) a=~b;
- B) assign a=~b;
- C) always @(negedge b) a<=~c;
- D) assign a=(b<c)?0:1;
- E) assign a=(b>c)?1'b1:1'bz;

Правильный ответ: A.

5. Какое утверждение относительно портов модуля тестбенча является верным?

Варианты ответа:

- A) модуль тестбенча не содержит портов
- B) в модуле тестбенча не должно быть входов, но должны быть выходы
- C) в модуле тестбенча не должно быть выходов, но должны быть входы
- D) модуль тестбенча должен иметь и входы, и выходы
- E) модуль тестбенча должен иметь только двунаправленные порты

Правильный ответ: A.

6. Найдите неверное утверждение про оператор initial.

Варианты ответа:

- A) оператор initial должен располагаться перед операторами assign и always
- B) оператор initial используется при написании тестбенчей
- C) оператор initial относится к параллельным операторам
- D) оператор initial запускается на выполнение один раз
- E) внутри оператора initial можно настроить сохранение результатов и про-должительность моделирования

Правильный ответ: A.

7. Выберите, какому элементу соответствует следующий код на языке Verilog:

```
module nemo(a,b,c);
  input a,b;
  output reg c;
  always @(negedge a) c<=b;
endmodule
```

Варианты ответа:

- A) D-триггер, тактируемый по заднему фронту
- B) D-триггер, тактируемый по переднему фронту
- C) мультиплексор
- D) сдвиговый регистр
- E) двунаправленный буфер

Правильный ответ: A.

8. Выберите, какому элементу соответствует следующий код на языке Verilog:

```
module nemo (a, b, c);
  input a, b;
  output c;
  assign c =a ? b : 1'bz;
endmodule
```

Варианты ответа:

- A) буфер с z-состоянием
- B) компаратор
- C) RS-триггер
- D) мультиплексор
- E) приоритетный шифратор

Правильный ответ: A.

9. Выберите, какому элементу соответствует следующий код на языке Verilog:

```
module nemo (a, b, c);
  input [1:0] a;
  input [3:0] b;
  output c;
  reg d;
  always @(a)
  begin
    case (a)
      2'b00 : d<=b[0];
      2'b01 : d<=b[1];
      2'b10 : d<=b[2];
      2'b11 : d<=b[3];
    endcase
  end
  assign c = d;
endmodule
```

Варианты ответа:

- A) мультиплексор
- B) триггер
- C) сдвиговый регистр
- D) компаратор
- E) тождественность

Правильный ответ: A.

10. Выберите, какому элементу соответствует следующий код на языке Verilog:

```
module nemo (a, b, c);
  input a, b;
  output [7:0] c;
  reg [7:0] d;
  always @ (posedge a) d<={d[6:0],b};
  assign c=d;
endmodule
```

Варианты ответа:

- A) сдвиговый регистр
- B) JK-триггер
- C) двунаправленный порт
- D) мультиплексор
- E) шифратор

Правильный ответ: A.

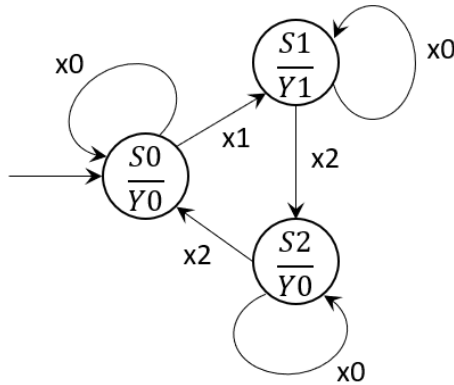
11. Какие из нижеперечисленных описаний портов не могут быть правильными или достаточно полными (сигналы a, b имеют тип wire)?

Варианты ответа:

- A) module nemo(input a, output b);
- B) module nemo(a, b);
  - input a;
  - output b;
- C) module nemo(a, b);
  - wire a;
  - wire b;
- D) module nemo(input wire a, output wire b);
- E) module nemo(a, b);

Правильный ответ: C, E.

12. Составьте совмещенную таблицу переходов и выходов конечного автомата, соответствующую следующему графу конечного автомата:



Решение:

Заголовочный столбец таблицы должен содержать все входные состояния автомата, а заголовочная строка – все внутренние состояния автомата. На пересечении строк и столбцов должны находиться состояния автомата, в которые он перейдет из текущего состояния (указанного в соответствующей ячейке заголовочной строки) под действием текущего входного состояния (указанного в соответствующей ячейке заголовочного столбца). Вместе с внутренними состояниями автомата указываются соответствующие выходные состояния. Для тех переходов, которым не соответствуют ребра графов, указываются прочерки.

Ответ:

X\S	S0	S1	S2
X0	S0/Y0	S1/Y1	S2/Y0
X1	S1/Y1	-	-
X2	-	S2/Y0	S0/Y0

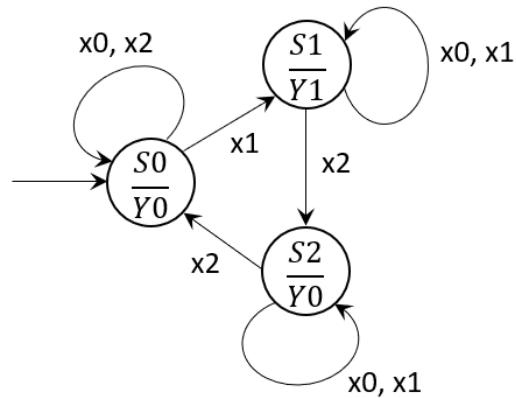
13. Изобразите граф конечного автомата, соответствующий следующей таблице переходов и выходов конечного автомата:

X\S	S0	S1	S2
X0	S0/Y0	S1/Y1	S2/Y0
X1	S1/Y1	S1/Y1	S2/Y0
X2	S0/Y0	S2/Y0	S0/Y0

Решение:

Заголовочный столбец таблицы содержит все входные состояния автомата, а заголовочная строка – все внутренние состояния автомата. На пересечении строк и столбцов находятся состояния автомата, в которые он перейдет из текущего состояния (указанного в соответствующей ячейке заголовочной строки) под действием текущего входного состояния (указанного в соответствующей ячейке заголовочного столбца). Вместе с внутренними состояниями автомата указываются соответствующие выходные состояния. Для тех переходов, которым не соответствуют ребра графов, указываются прочерки.

Ответ:



14. Составьте совмещенную таблицу переходов и выходов конечного автомата, соответствующую следующему описанию на языке Verilog:

```

module FSM (output reg Y0, Y1, input X0, X1, X2, X3, X4, clk, rst_n);
  reg [1:0] state, next;
  always @(posedge clk or negedge rst_n)
    if (!rst_n) state <= 2'b00;
    else state <= next;
  always @*
  begin
    next = 2'bx;
    case (state)
      2'b00: begin
        if (X0) next = 2'b01;
        if (X1) next = 2'b00;
        Y0 = 1'b1;
        Y1 = 1'b0;
      end
      2'b01: begin
        if (X3) next = 2'b10;
        if (X2) next = 2'b01;
        if (X4) next = 2'b11;
        Y0 = 1'b0;
        Y1 = 1'b1;
      end
      2'b10: begin
        if (X3) next = 2'b11;
        if (X4) next = 2'b10;
        Y0 = 1'b0;
        Y1 = 1'b1;
      end
      2'b11: begin
        if (X1) next = 2'b00;
        if (X0) next = 2'b01;
        Y0 = 1'b1;
        Y1 = 1'b0;
      end
    endcase
  end
endmodule
Решение:

```

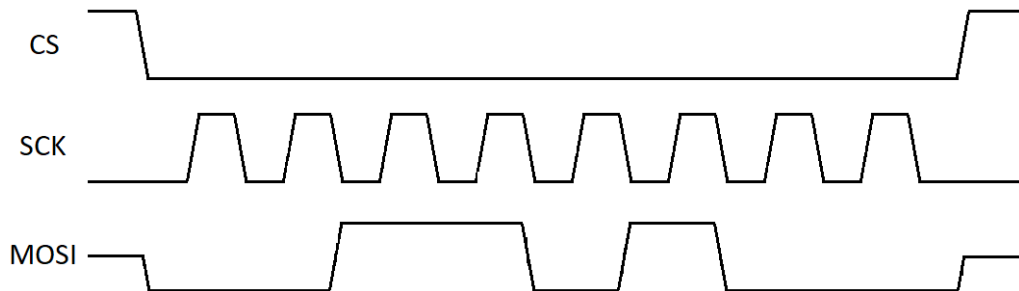
Заголовочный столбец таблицы должен содержать все входные состояния автомата, а заголовочная строка – все внутренние состояния автомата. На пересечении строк и столбцов должны находиться состояния автомата, в которые он перейдет из текущего состояния (указанного в соответствующей ячейке заголовочной строки) под действием текущего входного состояния (указанного в соответствующей ячейке заголовочного столбца). Вместе с внутренними состояниями автомата указываются соответствующие выходные состояния. Для тех переходов, которым не соответствуют ребра графов, указываются прочерки.

Количество внутренних состояний автомата определяется оператором case. В данном случае это 4 состояния. Количество входных и выходных состояний определяется строкой-заголовком модуля.

Ответ:

X\S	S0	S1	S2	S3
X0	S1/Y1	-	-	S1/Y1
X1	S0/Y0	-	-	S0/Y0
X2	-	S1/Y1	-	-
X3	-	S2/Y1	S3/Y0	-
X4	-	S3/Y0	S2/Y1	-

15. Передача данных по интерфейсу SPI в режиме Mode0 от ведущего к ведомому имеет следующую спецификацию: первый принимаемый бит игнорируется, второй бит означает порядок передаваемых данных (0 - первым передается старший бит числа, 1 - первым передается младший бит числа), следующие биты - передаваемое число. Передаче какого числа соответствует приведенная временная диаграмма? (ответ указать в десятичном формате)

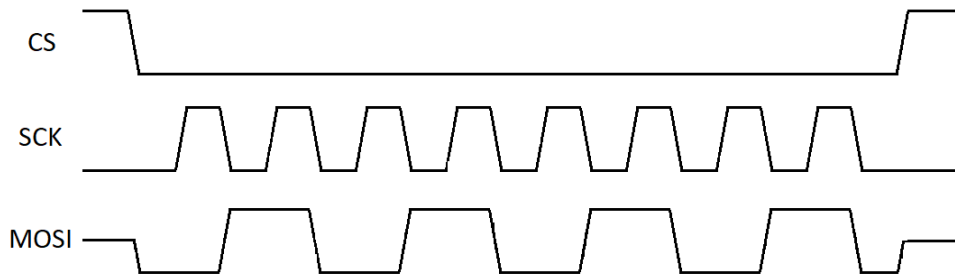


Решение:

При передаче данных по интерфейсу SPI в режиме Mode0 от ведущего к ведомому необходимо считывать данные сигнала MOSI по переднему фронту сигнала SCK. На приведенной диаграмме передается следующая последовательность бит: 0,0,1,1,0,1,0,0 (слева – первый передаваемый бит). Первый передаваемый бит согласно условию игнорируется. Равенство нулю второго передаваемого бита означает, что в последующих передаваемых битах сначала идет старший бит передаваемого числа. Таким образом, в двоичном виде было передано число  $110100_2$ , что соответствует числу  $52_{10}$  в десятичной системе счисления.

Ответ: 52.

16. Передача данных по интерфейсу SPI в режиме Mode0 от ведущего к ведомому имеет следующую спецификацию: первый принимаемый бит игнорируется, второй бит означает порядок передаваемых данных (0 - первым передается старший бит числа, 1 - первым передается младший бит числа), следующие биты - передаваемое число. Передаче какого числа соответствует приведенная временная диаграмма? (ответ записать в десятичном)



Решение:

При передаче данных по интерфейсу SPI в режиме Mode0 от ведущего к ведомому необходимо считывать данные сигнала MOSI по переднему фронту сигнала SCK. На приведенной диаграмме передается следующая последовательность бит: 0, 1, 0, 1, 0, 1, 0, 1 (слева – первый передаваемый бит). Первый передаваемый бит согласно условию игнорируется. Равенство единице второго передаваемого бита означает, что в последующих передаваемых битах сначала идет младший бит передаваемого числа. Таким образом, в двоичном виде было передано число  $101010_2$ , что соответствует числу  $42_{10}$  в десятичной системе счисления.

Ответ: 42.

17. Какая временная диаграмма соответствует следующему коду на языке Verilog?

```

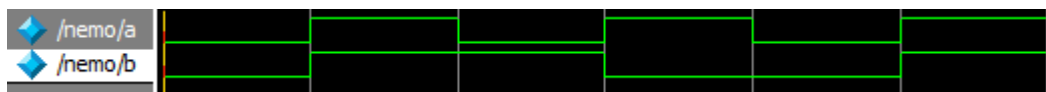
`timescale 1ns/1ns
module nemo();
  reg a=0;
  reg b=0;
  initial
  begin
    #1 a<=1;
    #1 a<=0;
    #1 a<=1;
    #1 a<=0;
    #1 a<=1;
    #1 $stop;
  end
  always @(posedge a) b<=!b;
endmodule

```

Решение:

В приведенной Verilog-модели сигнал «а» изменяется в операторе инициализации initial на каждом шаге моделирования, который составляет 1 нс. Изменения производятся 5 раз, после чего моделирование останавливается системной функцией \$stop. Начальное значение сигнала «а», задаваемое при объявлении сигнала, равно 0. Сигнал «b» изменяется оператором always по переднему фронту сигнала «а» на инвертированное значение сигнала «а». Начальное значение сигнала «b» равно 0.

Ответ:



18. Какая из временных диаграмм соответствует приведенному коду на Verilog?

```

`timescale 1ns/1ps
module nemo();
  reg a=0;
  reg b=0;

```

```

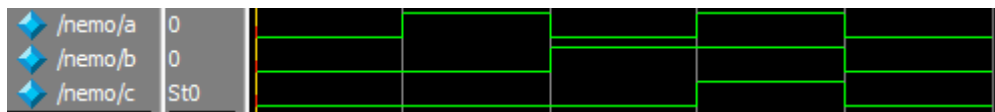
wire c;
initial
begin
    #5 $stop;
end
always #1 a <= !a;
always #2 b <= !b;
assign c = a & b;
endmodule

```

Решение:

В приведенной Verilog-модели сигнал «а» инвертируется в операторе процесса always через 1 нс, а сигнал «b» - через 2 нс. Начальные значения этих сигналов равны 0. Сигнал «с» изменяется оператором непрерывного присваивания assign и представляет собой результат логической операции «И» над сигналами «а» и «b».

Ответ:



### Примерный перечень вопросов для текущего контроля успеваемости

1. Из каких элементов состоит LUT?
2. Логическую функцию какого числа переменных может реализовать LUT, имеющий 64 регистра?
3. Какое устройство в составе логических элементов ПЛИС служит для реализации последовательностных схем?
4. Как осуществляется компиляция программного кода в Icarus-Verilog ?
5. Как называется программа-симулятор в Icarus-Verilog ?
6. Для чего используется файл с расширением .vcd ?
7. В чем заключается отличие параллельных операторов от последовательных?
8. Какими могут быть направленности портов в Verilog и в чем заключаются их отличия?
9. Какие правила следует применять при назначении типов портов и сигналов, подключаемых к портам?
10. Как реализуется двунаправленный порт в Verilog?
11. Что такое z-состояние?
12. Что означает запись 4'b10xz ?
13. Что означает параметр time\_precision директивы компилятора `timescale?
14. Какие требования накладываются на порты тестбенча?
15. Каким образом в тестбенче можно задать продолжительность моделирования?
16. Можно ли реализовать изменение входных сигналов при помощи оператора initial?
17. Можно ли использовать сигнал типа wire в правой части оператора присваивания внутри оператора процесса always?
18. Каким образом можно реализовать ветвление в операторе assign?
19. В чем заключаются отличия flip-flop триггеров от триггеров-защелок?
20. Какие условия запуска процесса always являются синтезабельными, а какие – нет?
21. В чем заключается отличие обычных мультиплексоров от мультиплексоров-селекторов?
22. Мультиплексор какой размерности может быть реализован при помощи четырехвходового LUT?
23. Являются ли операторы цикла синтезабельными?



24. Что такое подтягивающий резистор и для чего он используется?
25. Почему сигналы, подаваемые на входы ПЛИС обычно имеют в качестве активного уровня логический «0»?
26. Как включаются внутренние подтягивающие резисторы в Quartus?
27. Какую направленность будут иметь неиспользуемые входы/выходы ПЛИС?

## 20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине – *зачет*. Оценка за освоение дисциплины определяется ведущим дисциплину преподавателем как экспертом.

### Перечень вопросов к зачету

1. Классификация и области применения ПЛИС.
2. Мелко-, средне-, крупномодульные архитектуры.
3. Устройства на основе мультиплексоров, устройства на основе таблиц соответствия.
4. Секции и логические ячейки.
5. Конфигурируемые логические блоки CLB.
6. Блоки логических массивов LAB.
7. Распределенное ОЗУ и сдвиговые регистры.
8. Схемы ускоренного переноса.
9. Встроенные блоки ОЗУ. Встроенные умножители, сумматоры, блоки умножения с накоплением.
10. Аппаратные и программные встроенные микропроцессорные ядра.
11. Дерево синхронизации и диспетчеры синхронизации.
12. Ввод-вывод общего назначения.
13. Гигабитные приемопередатчики.
14. Структура Verilog-проекта. Модули.
15. Сигналы типа `reg` и `wire`, значения сигналов 'x' и 'z'.
16. Типы портов. Аппаратное представление двунаправленного порта.
17. Правила задания типов сигналов при подключении к портам.
18. Типы данных в Verilog. Арифметические, логические, побитовые и другие операции в языке Verilog.
19. Деление операторов на параллельные и последовательные.
20. Оператор `assign`. Использование тернарного оператора.
21. Оператор `always`. Виды условий запуска в операторе `always`.
22. Последовательные операторы.
23. Структурное и поведенческое описание устройства. Оператор подключения экземпляра модуля.
24. Понятие синтезруемости HDL-описания. Операция логического синтеза.
25. Статический временной анализ.
26. Понятие функциональной верификации HDL-описания, принципы построения тест-бенчей средствами языка Verilog.
27. Средства языка SystemVerilog для верификации. Организация тестового окружения.
28. Параллельные тестовые окружения.
29. Утверждения и последовательности.
30. Функциональное покрытие.
31. Параметры и локальные параметры. Переопределение параметров.

32. Генерация однотипных компонентов с помощью цикла generate и индексной переменной genvar. Условная генерация компонентов.
33. Понятие гонки сигналов. Понятие метастабильности триггеров.
34. Схемы для экспериментального наблюдения состояний метастабильности триггеров.
35. Устранение метастабильных состояний с помощью сдвиговых регистров.
36. Конечные автоматы Мура и конечные автоматы Мили. Представление конечных автоматов в виде графов и таблиц переходов.
37. Реализация структуры конечного автомата в Verilog: логика переходов, регистр состояния, логика формирования выходных сигналов. .
38. Базовые идеи конвейеризации. Конвейерные идеализмы.
39. Синхронизация стадий конвейера с переменными задержками.

Для текущего контроля успеваемости используется устный опрос, отчеты о ходе выполнения практических работ.

Каждому обучающемуся задаются вопросы по всем разделам спецкурса.

Оценка освоения компетенций обучающимися во время прохождения спецкурса осуществляется по следующим критериям:

- уровень профессиональной подготовки;
- ответы на контрольные вопросы.

Уровень профессионализма (профессиональные знания, умения, навыки и компетенции) оценивается по следующим показателям:

- умение формулировать цели исследований;
- адекватное применение физико-математического аппарата для решения поставленных задач;
- адекватная рефлексия выполняемой научно-практической деятельности.

На основании выполнения обучающимся программы спецкурса и с учетом критериев оценки итогов освоения спецкурса выставляется: «зачтено»/«не зачтено».

Критерии оценки работы обучающихся, которые соотносятся с уровнями сформированности компетенций:

- оценка «зачтено» выставляется при полном соответствии работы обучающихся всем вышеуказанным показателям: компетенции сформированы полностью, проявляются и используются систематически и в полном объеме. Данный уровень обязателен для всех осваивающих ООП;
- оценка «не зачтено» выставляется в случае несоответствия работы обучающегося всем требуемым показателям, неорганизованности, безответственности и низкого качества работы при выполнении заданий, предусмотренных программой спецкурса.