**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Проектирование цифровых систем

Digital System Design

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 042918

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Цели: обучение проектированию интегральных схем (ИС) на примере программируемых логических интегральных схем (ПЛИС).

Задачи: изучение среды проектирования Xilinx Vivado либо Intel Quartus, языка уровня RTL, платы для прототипирования с ПЛИС, системы уровня HLS; построение комбинационных, последовательных, конвейерных ИС на этих языках в этих средах.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

* Знание английского языка в достаточной для чтения технической литературы степени.
* Умение программировать на любом языке. Например, в задаче на реализацию умножения двух целых чисел при помощи сумматора, не должен вызывать вопросов итеративный алгоритм умножения двоичных целых чисел «в столбик». Цель курса – научиться техническим особенностям реализации алгоритмов при помощи интегральной синхронной цифровой схемы без процессора.

**1.3. Перечень результатов обучения (learning outcomes)**

Результатами обучения являются:

* понимание и практическое умение реализовывать несложные задачи на языке описания цифровых интегральных схем уровня RTL (VHDL или Verilog):
* задача на комбинационную логику;
* на итерационный алгоритм (итеративное умножение, деление, порождение разнообразных временнЫх диаграмм посредством счетчиков);
* на конвейерное вычисление некоторого арифметического выражения с поддержкой протокола типа AXI Stream;
* умение использовать несинтезируемые конструкции для реализации тестовой обертки для каждой из задач;
* отладка и тестирование задач в симуляторе (таком как встроенный симулятор САПР Xilinx Vivado или бесплатная версия Modelsim в Altera Quartus prime lite);
* умение выполнить синтез, трассировку задачи для ПЛИС на частоту порядка 250-300МГц и, при необходимости, поправить решение так, чтобы все задержки укладывались в эту частоту (на примере задачи с реализацией конвейера, в САПР Vivado или Quartus);
* умение использовать такие вспомогательные средства, как автоматическое порождение схем до/после синтеза и трассировки;
* понимание того, как взаимодействуют отдельные компоненты в системе на кристалле (на примере протоколов семейства AXI: AXI Stream, AXI4 memory mapped и стандартных блоков библиотеки Xilinx для работы с этими протоколами);
* в частности, понимание того, что процессор – одна из таких компонент и того, как работать с другими блоками системы на кристалле из программы, работающей на процессоре;
* умение встроить решение своей задачи в систему, собранную из стандартных блоков, которая позволит протестировать блок не только в симуляторе, но и на отладочной плате с ПЛИС (на примере интеграции задачи с конвейером в систему на кристалле Xilinx ZYNQ при помощи DMA, блоков очередей, блоков преобразования ширины шины и тестирования работы конвейера из программы на процессоре);
* умение использовать внутрисхемный анализатор для наблюдения за значениями сигналов в ПЛИС;
* умение реализовать более сложную задачу (предполагающую не только распараллеленные конвейерные вычисления, но также и нетривиальное использование внутренних памятей – в частности, таковы многие задачи обработки видео) при помощи средства высокоуровневого синтеза (на примере Vivado HLS) и встроить решение в тестовое окружение для тестирования на ПЛИС;
* умение пользоваться недорогими логическими анализаторами и бесплатной программой pulseview для анализа низкоскоростной коммуникации между компонентами печатной платы по таким протоколам как i2c, SPI, UART.

Компетенции, формируемые дисциплиной:

* ОПК-1 – способен находить, формулировать и решать актуальные проблемы фундаментальной и прикладной информатики и информационных технологий;
* ПКА-1 – способен находить, формулировать и решать актуальные проблемы фундаментальной и прикладной информатики и информационных технологий;
* ПКП-1 – способен демонстрировать фундаментальные знания математических и естественных наук, программирования и информационных технологий;
* ПКП-4 – способен преподавать дисциплины математики и информатики в различных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения;
* ПКП-6 – способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;
* ПКП-7 – способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности;
* ПКП-8 – способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений.

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Активные и интерактивные формы учебных занятий — семинары (15 ак.ч.), лекции, предполагающие обсуждение материала с преподавателем (5 ак.ч.).

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 1 | 15 | 15 | 2 |  |  |  |  |  | 2 |  |  |  | 38 |  | 36 |  | 20 | 3 |
|  | 2-25 | 2-25 | 2-25 |  |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 15 | 15 | 2 |  |  |  |  |  | 2 |  |  |  | 38 |  | 36 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 1 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование темы (раздела, части)** | **Вид учебных занятий** | **Кол-во часов** |
| 1 | Тема 1. Введение. Язык VHDL . Навыки запуска и работы в САПР. | лекции | 4 |
| семинары | 4 |
| 2 | Тема 2. Задержки в сложной комбинационной логике. Повышение тактовой частоты засчет её конвейеризации. | лекции | 2 |
| семинары | 2 |
| 3 | Тема 3. Протоколы обмена семейства AXI, стандартные блоки поддержки протоколов. | лекции | 3 |
| семинары | 3 |
| 4 | Тема 4. Среда графического проектирования Xilinx Vivado IP Integrator | лекции | 2 |
| семинары | 2 |
| 5 | Тема 5. Система на кристалле ZYNQ, работа с отладочной платой с ZYNQ. | лекции | 2 |
| семинары | 2 |
| 6 | Тема 6. Система высокоуровневого синтеза Vivado HLS | лекции | 2 |
| семинары | 2 |
|  | Самостоятельная работа с использованием  методических материалов |  | 74 |
| 7 | Промежуточная аттестация | экзамен | 2 |
| консультации | 2 |
| самостоятельная работа | 36 |
| **Итого** | | | **144** |

План курса:

**Тема 1. Введение. Язык VHDL. Навыки запуска и работы в САПР**

Проектирование цифровых схем как параллельное программирование с потенциально бесконечным числом независимо работающих устройств. Разные маршруты проектирования, позволяющие придти от чисто программной реализации некоторого алгоритма к специализированному устройству, реализующему этот алгоритм более оптимально в каком-либо смысле. Технологическая библиотека ASIC, FPGA. Асинхронная логика, задержки, последовательные вычисления, конвейеризация – на примере 2-битового сумматора.

Основные конструкции языка VHDL: типы данных, литералы, приведение типов, if, case, сигналы, переменны. Entity, architecture, process на примере описания 2-битового сумматора на VHDL. Вырезки, циклы. Создание экземпляра entity в реализации другого entity. Организация тестирования в симуляции. Generics, generate.

Способы описания регистров в VHDL.

**Тема 2. Задержки в сложной комбинационной логике. Повышение тактовой частоты засчет её конвейеризации**

Пример: вычисления минимума из 8 16-битовых чисел. Следует рассмотреть соответствующую интегральную схему, реализующую этот пример за 1 такт. В такой схеме образуется очень длинный комбинационный путь, растягивающий такт до 20нс. Далее следует рассмотреть конвейерную схему для того же примера, понять, почему в ней задержка получается порядка 4нс, заметить, что есть разница между пропускной способностью (throughput) и задержкой (latency) конвейера.

**Тема 3. Протоколы обмена семейства AXI, стандартные блоки поддержки протоколов**

Протокол AXI Stream: сигналы \*tvalid, \*tdata, \*tready. Библиотека поддержки Xilinx для AXI Stream: FIFO, clock converter, data width converter, switch.

Объединение пяти каналов типа AXI Stream в интерфейс AXI4 memory mapped. AXI Interconnect. Варианты реализации (шина N:1→1:M, полный граф N:M). Поле ID, маршрутизация пакетов согласно адресам общего адресного пространства. Автоматизированное распределение адресов.

Шина конфигурации. Конфигурационные регистры аппаратных блоков.

**Тема 4. Среда графического проектирования Xilinx Vivado IP Integrator**

Автоматизированное порождение IP блока для IP Integrator из кода на VHDL с правильными именами входов/выходов entity верхнего уровня.

Создание проекта в IP Integrator, вставка экземпляров блоков, автоматизированная вставка шинных блоков. Validate design.

**Тема 5. Система на кристалле ZYNQ, работа с отладочной платой с ZYNQ**

Архитектура (процессорная подсистема: процессор ARM, контроллер DDR3, SD Card, интерфейсы к подсистеме ПЛИС, размеры очередей в AXI4 интерфейсах, максимальные частоты).

IP блок DMA. Конфигурационный мастер шины, работа с ним из обычной userspace программы под ОС Linux.

Chipscope.

Сопоставление сигналов верхнего уровня и ножек ПЛИС.

Подключение платы к PC через USB, через ethernet, работа с платой через UART, JTAG, ssh, scp.

**Тема 6. Система высокоуровневого синтеза Vivado HLS**

Автоматическое планирование под целевую тактовую частоту, порождение памятей, интерфейсов семейства AXI. Анализ результатов планирования в графическом интерфейсе. Разбиение алгоритма на параллельно работающие процессы.

Использование всех описанных выше возможностей в коде на С++: типы ap\_int<>, ap\_uint<>, #pragma HLS PIPELINE, UNROLL, DATAFLOW, ARRAY\_PARTITION, INTERFACE, RESOURCE. Ручные преобразования кода на С++, которые требуются для того, чтобы успешно воспользоваться этими прагмами.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать лекции и практические занятия, выполнять задания преподавателей.

**3.1.2 Методическое обеспечение самостоятельной работы**

Списки обязательной и дополнительной литературы, задания (см. п. 3.1.4).

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Контроль производится путем оценки выполнения домашних заданий и выставления баллов за них и оценки прохождения онлайн-курса.

|  |  |
| --- | --- |
| Тема задания | балл |
| Успешное прохождение (зачёт) онлайн курса «Digital Systems: From Logic Gates to Processors» https://www.coursera.org/learn/digital-systems | 3 |
| Последовательная логика ИЛИ Конвейеризация с протоколом обмена (реализация на VHDL или Verilog) | 3 |
| Запуск на отладочной плате с ПЛИС ИЛИ Высокоуровневый синтез (в системе Vivado HLS) | 4 |

По полученному оценка выставляется по следующим критериям.

| Суммарный балл | Оценка ECTS | Оценка СПбГУ за экзамен |
| --- | --- | --- |
| 9-10 | A | отлично |
| 8 | B | хорошо |
| 7 | C | хорошо |
| 6 | D | удовлетворительно |
| 4-5 | E | удовлетворительно |
| менее 4 | F | неудовлетворительно |

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

В процессе обучения каждый обучающийся получает задания, которые необходимо выполнить для положительной оценки по аттестации.

*Задания по темам*

1. комбинационная логика
2. последовательная логика
3. конвейеризация
4. конвейеризация с протоколом обмена
5. запуск задачи на отладочной плате с ПЛИС
6. высокоуровневый синтез

*Примеры заданий:*

**Арифметико-логическое устройство**

Некоторые целочисленные арифметические операции определены для unsigned, из пакета ieee.std\_logic\_arith.all, соответственно, можно включить этот пакет в файл директивой use ieee.std\_logic\_arith.all;

После чего возможно объявлять входы-выходы типа

A : in unsigned(3 downto 0);

B : in unsigned(3 downto 0);

И использовать их в арифметике, например, написать в теле процесса X <= A + B.

Задача: реализовать блок, который в ответ на два 4х битовых беззнаковых целых числа и 2- битового кода операции выдает их сумму, разность либо побитовый OR в зависимости от кода (B”00”, B”01”, B”11”).

Случай, когда код равен B”11”, считается ошибочным. Тогда выходной однобитовый сигнал error вашего блока должен быть равен 1.

**Итеративное умножение**

Блок получает два N-битных беззнаковых целых (N : integer – VHDL generic), перемножает их, используя один сумматор ширины 2\*N (и какое-то колчичество управляющей логики), выдает число ширины 2\*N. В качестве теста хотелось бы видеть полный перебор всех возможных значений входов для небольшого значения N (например, 4). Для проверки в тесте можно пользоваться стандартной операцией “\*” библиотеки целых чисел.

**Кусочек билинейной интерполяции**

Дано grayscale изображение 2х2 пикселя по N бит и целые числа 0 <= x, y < 64 . Надо посчитать яркость пикселя с координатами (x/64, y/64) в этом изображении. (Иными словами, можно сказать, что координаты даны в формате fixpoint 0 знаков до двоичной точки и 6 знаков после). Необходимо реализовать это упражнение в виде конвейера, способного выдавать 1 результат на каждом такте.

**Работа с ПЛИС на отладочной плате**

Запустить результат предыдущего упражнения на отладочной плате.

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем техническом образовании по направлениям «Программная инженерия», «Математическое обеспечение и администрирование информационных систем» и другим смежным направлениям, связанным с информационными технологиями.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Аудитории должны соответствовать требованиям СанПиН 2.2.2/2.4.1340-03 в редакции от 21 июня 2016 года.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

В аудиториях, где проводятся лекционные и практические занятия, необходимо наличие досок и средств письма на них. Для показа слайдов необходим компьютер с установленным программным обеспечением для работы со слайдами в форматах PDF, PPT, PPTX и подключенный к нему мультимедиа проектор с экраном.

Аудитории для проведения практических занятий дополнительно должны быть оснащены компьютером для каждого обучающегося с возможностью загрузки на них студентами файлов с внешних USB-носителей.

**3.3.3 Характеристики специализированного оборудования**

Отладочная плата на базе любого чипа Xilinx ZYNQ, например, Zybo Z7 (https://store.digilentinc.com/zybo-z7-zynq-7000-arm-fpga-soc-development-board/),

отладочная плата на базе SoC altera (ПЛИС+ARM), например, https://rocketboards.org/foswiki/view/Documentation/AtlasSoCDevelopmentPlatform

**3.3.4 Характеристики специализированного программного обеспечения**

На каждом компьютере для обучающихся должны быть установлены следующие программные системы:

Xilinx Vivado WebPack

Intel (Altera) Quartus Prime Lite в комплекте с Modelsim PE Altera edition.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объеме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объеме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. VHDL Cookbook. (https://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf) – ЭР открытого доступа в сети Интернет.
2. AXI4-Stream Infrastructure IP Suite. (https://www.xilinx.com/support/documentation/ip\_documentation/axis\_infrastructure\_ip\_suite/v1\_1/pg085-axi4stream-infrastructure.pdf) – ЭР открытого доступа в сети Интернет.
3. Vivado Design Suite User Guide. High-Level Synthesis. (https://www.xilinx.com/support/documentation/sw\_manuals/xilinx2017\_4/ug902-vivado-high-level-synthesis.pdf) – ЭР открытого доступа в сети Интернет.
4. IEEE Standard VHDL Language Reference Manual. https://edg.uchicago.edu/~tang/VHDLref.pdf – ЭР открытого доступа в сети Интернет.

**3.4.2 Список дополнительной литературы**

Не требуется.

**3.4.3 Перечень иных информационных источников**

1. https://www.xilinx.com/ – ЭР открытого доступа в сети Интернет.

2. http://altera.com/– ЭР открытого доступа в сети Интернет.

**Раздел 4. Разработчики программы**

Луцив Дмитрий Вадимович, старший преподаватель кафедры системного программирования [d.lutsiv@spbu.ru](mailto:d.lutsiv@spbu.ru).

Медведев Олег Валерьевич, инженер-программист, ООО «Ланит-Терком».