

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

УТВЕРЖДАЮ

Декан факультета
ВМК МГУ имени М.В. Ломоносова

академик _____

«___» _____ 201__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование дисциплины (модуля):

«Математические модели и методы синтеза сверхбольших интегральных схем»

Уровень высшего образования:

бакалавриат

Направление подготовки (специальность):

01.03.02 Прикладная математика и информатика

Направленность (профиль) ОПОП:

дисциплина относится к базовой части программы

2-й поток: математические методы обработки информации и принятия решений

Форма обучения:

Очная

Рабочая программа рассмотрена и одобрена
Методической комиссией факультета

(протокол № _____ от «___» _____ 201__ г)

Москва 20__

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки по направлениям 01.03.02, 01.04.02 «Прикладная математика и информатика» в редакции приказа МГУ от 30 декабря 2016 г.

Год (годы) приема на обучение 2011-2018

1. Дисциплина относится к вариативной части ОПОП ВО.

2. Входные требования для освоения дисциплины (модуля): учащиеся должны владеть знаниями по дискретной математике и иметь навыки программирования в объеме, соответствующем программе первого и второго года обучения основных образовательных программ бакалавриата по укрупненным группам направлений и специальностей 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки».

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

Компетенции выпускников, частично формируемые при реализации дисциплины (модуля):

- **ОПК-1.Б** Способность применять и адаптировать существующие математические и компьютерные методы для разработки и реализации алгоритмов решения актуальных задач в области фундаментальной и прикладной математики
- **ОПК-2.Б** Способность применять и модифицировать математические модели, а также интерпретировать полученные математические результаты при решения задач в области профессиональной деятельности
- **ПК-5.Б** Способность определить совокупность математических методов и программных решений для отдельного этапа решения прикладной задачи в рамках заданной схемы

Планируемые результаты обучения по дисциплине (модулю):

Знать:

1. основы технологии производства современных интегральных схем, обобщенный маршрут их проектирования;
2. основные методологии проектирования современных интегральных схем;
3. основные алгоритмы логической оптимизации цифровых интегральных схем;
4. основные алгоритмы привязки логической схемы к технологической библиотеке;
5. основные алгоритмы разбиения графов и гиперграфов;
6. основные алгоритмы размещения элементов и трассировки соединений в интегральной схеме;
7. основы языка описания схем Verilog;
8. базовые методы описания основных блоков цифровой интегральной схемы при помощи языка описания схем Verilog.

Уметь:

1. создавать, модифицировать и применять на практике алгоритмы, решающие различные математические задачи проектирования интегральных схем;
2. создавать описания простых интегральных схем при помощи языка Verilog;
3. использовать инструменты автоматизации проектирования интегральных схем для проектирования простых интегральных схем;
4. прототипировать простые интегральные схемы при помощи программируемых интегральных схем.

Владеть:

1. базовыми навыками проектирования цифровых интегральных схем с использованием языка Verilog;
2. базовыми навыками прототипирования интегральных схем с использованием программируемых интегральных схем.

4. Формат обучения: лекции проводятся с использованием экрана и проектора для демонстрации слайдов презентаций и маркерной или меловой доски, практические занятия проводятся в специально оборудованном компьютерном классе, оборудованном компьютерами с установленными системами автоматизации проектирования интегральных схем и комплектами отладочных плат с программируемыми интегральными схемами.

5. Объем дисциплины (модуля) составляет 4 з.е., в том числе 72 академических часа, отведенных на контактную работу обучающихся с преподавателем, 72 академических часов на самостоятельную работу обучающихся.

6. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий.

Наименование и краткое содержание разделов и тем дисциплины (модуля), Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы			Самостоятельная работа обучающегося, часы <i>(виды самостоятельной работы – эссе, реферат, контрольная работа и пр. – указываются при необходимости)</i>
		Занятия лекционного типа*	Занятия семинарского типа*	Всего	
1. Основные сведения о задаче проектирования цифровых интегральных схем и связанные с ней модели дискретных управляющих систем (общие сведения о проектировании цифровых интегральных схем; уровни абстракции и основные стратегии проектирования цифровых интегральных схем; упрощенный маршрут проектирования современных цифровых интегральных схем. КМОП-схемы; связь между логическим и транзисторным уровнем, понятие о технологической библиотеке).	4	4	0	4	0
2. Основы языка описания схем Verilog и методологи проектирования интегральных схем с его использованием (основные элементы языка	12	12	0	12	0

Verilog, поведенческое и структурное описание интегральных схем, синхронные схемы).					
3. Текущий контроль успеваемости: практическая работа № 1.	12	0	0	0	12
4. Основы работы с системами автоматизации проектирования на примере прототипирования интегральных схем при помощи программируемых логических интегральных схем (основы работы с системой Quartus Prime; проектирование комбинационных и последовательных блоков; проектирование арифметических блоков, блоков памяти и конечных автоматов; временной анализ схем; аппаратная реализация алгоритмов).	36	0	36	36	0
5. Текущий контроль успеваемости: практическая работа № 2.	12	0	0	0	12
6. Задачи логического проектирования интегральных схем и связанные с ними математические модели интегральных схем (двоичные решающие диаграммы (BDD) основные операции над BDD и представление логических схем с использованием BDD; реализация функций алгебры логики в виде дизъюнктивных нормальных форм; модель логических сетей и основные структурные операции над ними; задача привязки логической схемы к технологической библиотеке и основные алгоритмы ее решения).	10	10	0	10	0

7. Текущий контроль успеваемости: практическая работа № 3	12	0	0	0	12
8. Задачи физического проектирования интегральной схемы (задача разбиения интегральной схемы и основные алгоритмы ее решения; задачи размещения элементов интегральной схемы и основные подходы к ее решению; задача трассировки соединений в интегральной схеме).	10	10	0	10	0
Промежуточная аттестация: устный экзамен	36	0	0	0	36
Итого	144	36	36	72	72

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю)

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

Практическая работа № 1

Моделирование комбинационных и последовательных схем на языке Verilog. Симуляция и тестирование схем.

Цель практической работы.

Цель практической работы освоить базовые конструкции языка Verilog. Научиться моделировать простые устройства, научиться проводить симуляцию, тестировать и верифицировать смоделированные устройства.

Практическая работа состоит из нескольких этапов. В рамках практической работы все студенты выполняют одинаковый набор заданий.

При этом устройство, для которого производится моделирование и тестирование у каждого студента свое (свой вариант).

Далее, идет описание этапов практической работы, а после указаны варианты устройств. При этом распределение студентов по вариантам указано в отдельном файле.

Этапы практической работы

Этап 1. Поведенческое (автоматное) моделирование.

Необходимо создать поведенческое (автоматное описание) на языке Verilog устройства для которого дано формальное описание (см. ниже). Формальное описание включает в себя описание входов и выходов моделируемого устройства, а также формальное поведение (функцию или алгоритм, реализуемый устройством).

Этап 2. Логическое (функциональное) моделирование.

По описанию, построенному на предыдущем этапе необходимо создать gate-level описание на языке Verilog. При этом разрешается использовать только следующие стандартные логические элементы языка Verilog: NAND, NOR, AND, OR, XOR, XNOR, BUF, NOT. Элементы единичной задержки (регистры) можно моделировать любым доступным в языке Verilog способом.

Этап 3. Тестирование поведенческого описания.

Необходимо создать тестовый «стенд» (testbench) на языке Verilog, который позволяет протестировать работу смоделированного устройства в разных режимах (сценариях работы). Требуется написать Verilog модель, которая генерирует различные сценарии работы устройства и проверяют корректность работы полученного на первом этапе описания.

Этап 4. Верификация логического описания.

Тестирование логического описания производится при помощи сравнения с поведенческим описанием. На данном этапе требуется построить тестовый «стенд» (testbench), который для заданного набора сценариев симулирует работу и поведенческого и функционального описаний и сравнивает результаты их работы. Если описания эквивалентны, то результаты их работы должны совпадать. На данном этапе требуется написать генератор случайных входных сценариев для проверки описаний.

Варианты моделируемых устройств.

Обязательные входы.

Во всех вариантах предполагается наличие следующих обязательных входов:

1. clock – вход тактового генератора.
2. reset – бит сброса. Когда на указанном входе значение «1», то значение на всех выходах устройства сбрасывается в нулевое значение. Задержка сброса не регламентируется.

enable – бит включения устройства. Когда на данном входе значение «1», то устройство работает, когда значение «0», то считается, что устройство выключено.

Вариант 1. Числа Фибоначчи.

По порядковому номеру вычислить двоичное представление соответствующего члена числового ряда Фибоначчи.

Вход: 3-х битовый провод, на который передается порядковый номер числа Фибоначчи.

Выход: 4-х битовый регистр, в котором сохраняется соответствующее число Фибоначчи.

Вариант 5. Уникальное число в памяти.

Пусть задан массив из 9 ячеек памяти. Каждая ячейка памяти представляет собой 4-х битовый регистр. Предполагается, что массив ячеек памяти заполнен так, что восемь ячеек памяти содержат одинаковые числа, а оставшийся регистр содержит число, отличное от всех остальных. Построить схему, которая находит это число. Если вход схемы не соответствует описанию, то значения на выходах схемы не регламентируются.

Вход: 36-битовый регистр, представляющий массив ячеек памяти.
Выход: 4-х битовый регистр, в котором сохраняется уникальное число.

Практическая работа № 2

Основы проектирования интегральных схем с использованием языка Verilog.

Цель практической работы.

Цель практической работы освоить базовые принципы дискретных устройств при помощи языка Verilog. Практическая работа состоит из нескольких независимых заданий.

Вариант 1.

1. Спроектировать на языке Verilog сумматор с каскадным переносом (размер входных аргументов должен быть

Вариант 2.

1. Спроектировать на языке Verilog дерево мультиплексоров (число слоев дерева должно быть параметром

параметром спроектированного модуля);

2. спроектировать конечный автомат на языке Verilog, который распознает последовательность 10010;
3. создать аппаратную реализацию на языке Verilog следующего модельного алгоритма:

```
count = 0;
i = 0;
while(i < 64){
    if( i % 6 != 0)
        count++;
}
```

спроектированного модуля);

2. спроектировать конечный автомат на языке Verilog, который распознает последовательность 00101;
3. создать аппаратную реализацию на языке Verilog следующего модельного алгоритма:

```
count = 0;
i = 0;
while(i < 128){
    if( i % 7 == 0)
        count++;
}
```

Практическая работа № 3

Аппаратная реализация алгоритмов и структур данных на базе программируемых интегральных схем.

Цель практической работы.

Цель практической работы освоить базовые принципы аппаратной реализации алгоритмов и структур данных с использованием синхронных схем. Научиться проектировать соответствующие операционный и управляющий автоматы. Практическая работа состоит из нескольких этапов. В рамках практической работы все студенты выполняют одинаковый набор заданий. При этом алгоритм, для которого производится моделирование и тестирование, у каждого студента свой (свой вариант). Далее, идет описание этапов практической работы, а после указаны варианты алгоритмов. При этом распределение студентов по вариантам указано в отдельном файле.

Этапы практической работы

Этап 1. Элементы операционного автомата (datapath).

Необходимо создать набор модулей на языке Verilog, которые реализуют основные блоки операционного автомата (datapath). Основными блоками операционного автомата являются регистры, блоки сравнения и блоки выполняющие соответствующие арифметические операции.

Этап 2. Проектирование операционного автомата (datapath).

Требуется спроектировать операционный автомат для алгоритма, заданного в варианте, на основе модулей, созданных на предыдущем этапе. В результате этого этапа должна быть построена блок-схема операционного автомата и определен его интерфейс.

Этап 3. Проектирование управляющего автомата.

Составить диаграмму Мура управляющего автомата, который использует операционный автомат для выполнения основных операций, заданного алгоритма. По диаграмме Мура написать модуль на языке Verilog, реализующий указанный автомат.

Этап 4. Сборка и верификация полученной аппаратной реализации алгоритма.

Написать модуль на языке Verilog, который инстанцирует модуль управляющего автомата и все компоненты операционного автомата и инстанцирует блок памяти соответствующего размера. Указанный блок памяти соответствующим образом инициализируется при помощи .MIF файла. Требуется соответствующим образом соединить все инстанцированные блоки и проверить корректность полученной аппаратной реализации алгоритма при помощи её симуляции с использованием ModelSim. Кроме того, требуется выполнить временной анализ полученной аппаратной реализации, установить максимальную тактовую частоту на которой она будет работать корректно. Дополнительно требуется, чтобы все модули были параметризованными. В качестве основных параметров выступает количество бит для регистров и других основных блоков, а также количество ячеек памяти.

Этап 5 Реализация устройства при помощи программируемых интегральных схем.

Используя Quartus Prime, реализовать спроектированную ранее аппаратную реализацию алгоритма в виде соответствующего файла-

прошивки для программируемой логической интегральной схемы (ПЛИС) Altera DE0-Nano и DE0-CV. При этом во всех вариантах предполагается, что все входы устройства выведены на переключатели и кнопки ПЛИС (сопоставление входов схемы с периферией студент выбирает самостоятельно), а выходы на светодиоды. Для тестирования работы аппаратной реализации алгоритма в качестве сигнала тактового генератора должна использоваться одна из кнопок ПЛИС. Кроме того, должна быть спроектирована и протестирована аппаратная реализация алгоритма, когда сигнал тактового генератора подается через соответствующий PLL блок, выдающий максимально возможную частоту сигнала, при которой спроектированное устройство работает корректно.

Вариант 1. Связанный список.

Создать аппаратную реализацию связанного списка (элементами списка являются бинарные последовательности заданной длины), поддерживающую следующие операции:

4. добавление и удаление элемента из списка;
5. подсчет суммы значений всех элементов списка.

Вариант 1. Ассоциативный массив.

Создать аппаратную реализацию ассоциативного массива (ключами и элементами массива являются бинарные последовательности заданной длины) на базе двоичной кучи, поддерживающую следующие операции:

1. добавление и удаление элемента из массива;
2. поиск элемента по ключу.

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

Зачет учебным планом не предусматривается.

Вопросы к экзамену.

1. Сведения о КМОП технологии: n- и p- каналные транзисторы, стандартные ячейки, схема и топология инвертора, 2-х входового элемента НЕ-И.
2. Маршрут проектирования СБИС.
3. Основы языка Verilog: модули, экземпляры (инстанции) модулей, типы reg и wire.
4. Основы языка Verilog: непрерывное присваивание, блок always, блокирующее и неблокирующее присваивания.
5. Последовательные схемы. Защёлки и триггеры. RS-триггер, D-защёлка, D-триггер.
6. Последовательные схемы. Проектирование конечного автомата.

7. Двухуровневый логический синтез и ДНФ. Основные подходы к двухуровневой оптимизации. Реализация функций алгебры логики в виде ДНФ и ее связь с ПЛИМ. Простые импликанты и избыточные покрытия. Обобщенно-монотонное разложение булевых функций.
8. Эвристический алгоритм двухуровневой оптимизации ESPRESSO. Общее описание идеи и структуры алгоритма. Построение тупикового покрытия по заданному покрытию (процедура IRREDUNDANT).
9. Эвристический алгоритм двухуровневой оптимизации ESPRESSO. Сужение граней заданного покрытия (процедура REDUCE).
10. Эвристический алгоритм двухуровневой оптимизации ESPRESSO. Максимальное расширение граней заданного покрытия (процедура EXPAND).
11. Многоуровневый логический синтез и связанные с ним представления функций. Модель логических сетей. Основные подходы к многоуровневой оптимизации.
12. Алгебраическое деление. Совокупное ядро делителей ДНФ и алгоритм его нахождения.
13. Критерий существования общих нетривиальных делителей для ДНФ (без доказательства). Алгоритмы нахождения общих тривиальных и нетривиальных алгебраических делителей для логических сетей.
14. Использование областей неопределенности (don't care) для упрощения логических сетей.
15. Привязка логической схемы к библиотеке. Использование метода динамического программирования для построения оптимального покрытия при решении задачи привязки логической схемы к библиотеке.
16. Задача разбиения электрической схемы. Примеры формулировок задачи разбиения. Классификация алгоритмов.
17. Алгоритм Кернигана-Лина и его расширения (алгоритм Федуччи-Маттеуса).
18. Задача кластеризации и основные подходы к многоуровневому разбиению интегральной схемы. Алгоритм hMeTiS.
19. Задача размещения модулей СБИС. Классификация алгоритмов
20. Алгоритмы размещения, основанные на разбиении схемы
21. Аналитическое размещение
22. Алгоритм моделирования отжига для решения задачи размещения
23. Задача трассировки соединений. Классификация алгоритмов трассировки. Основные подходы к представлению областей трассировки.
24. Задача глобальной трассировки. MST и SMT деревья. Последовательный алгоритм построения дерева Штейнера.
25. Глобальная трассировка на основе графа смежности областей трассировки.
26. Одновременная глобальная трассировка всех сетей. Сведение задачи глобальной трассировки к задаче целочисленного линейного программирования.

Экзаменационный билет состоит из двух вопросов, например

1. Основы языка Verilog: непрерывное присваивание, блок always, блокирующее и неблокирующее присваивания.
2. Алгоритм Кернигана-Лина и его расширения (алгоритм Федуччи-Маттеуса).

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)				
Оценка РО и соответствующие виды оценочных средств	2	3	4	5
Знания <i>Экзамен, практические работы (и/или тесты, и т.п.)</i>	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения <i>Экзамен, практические работы</i>	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности принципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) <i>Практические работы</i>	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

Соответствие результатов обучения и компетенций, в развитии которых участвует дисциплина (модуль)	
Результаты обучения	Компетенция, с частичным формированием которой связано достижение результата обучения
<p>Знать:</p> <ol style="list-style-type: none"> 1. основные алгоритмы логической оптимизации цифровых интегральных схем; 2. основные алгоритмы привязки логической схемы к технологической библиотеке; 3. основные алгоритмы разбиения графов и гиперграфов; 4. основные алгоритмы размещения элементов и трассировки соединений в интегральной схеме. <p>Уметь:</p> <ol style="list-style-type: none"> 1. создавать, модифицировать и применять на практике алгоритмы, решающие различные математические задачи проектирования интегральных схем. 	ОПК-1.Б
<p>Знать:</p> <ol style="list-style-type: none"> 1. основы технологии производства современных интегральных схем, обобщенный маршрут их проектирования; 2. основные методологии проектирования современных интегральных схем; 3. основы языка описания схем Verilog; 4. базовые методы описания основных блоков цифровой интегральной схемы при помощи языка описания схем Verilog. <p>Уметь:</p>	ОПК-2.Б

<ol style="list-style-type: none"> 1. создавать описания простых интегральных схем при помощи языка Verilog; 2. использовать инструменты автоматизации проектирования интегральных схем для проектирования простых интегральных схем; 3. прототипировать простые интегральные схемы при помощи программируемых интегральных схем. <p>Владеть:</p> <ol style="list-style-type: none"> 1. базовыми навыками проектирования цифровых интегральных схем с использованием языка Verilog. 	
<p>Владеть:</p> <ol style="list-style-type: none"> 1. базовыми навыками прототипирования интегральных схем с использованием программируемых интегральных схем. 	ПК-5.Б

8. Ресурсное обеспечение:

Основная литература:

1. Thomas D., Moorby P. The Verilog hardware description language. – Springer US, 2002. – 386 p.
2. Brayton R.K., Logic Synthesis. — Univ. of California, Berkeley, 2000.
3. Hatchel G.D., Somenzi F. Logic Synthesis and Verification Algorithms. – Kluwer Academic Publishers, 2002.
4. Giovanni De Micheli Synthesis and Optimization of Digital Circuits. – McGraw-Hill Science/Engineering/Math, 1994.
5. Andrew B. Kahng, Jens Lienig, Igor L. Markov, Jin Hую VLSI Physical Design: From Graph Partitioning to Timing Closure, 2011

Дополнительная литература:

1. Harris D.M., Harris S.L. Digital design and computer architecture. 2-nd ed. – Morgan Kaufmann, Elsevier, 2012. – 712 p.
2. Palnitkar S. Verilog HDL: a guide to digital design and synthesis. 2-nd ed. – Prentice Hall PTR, 2003. – 496 p.
3. Bhasker J. Verilog HDL synthesis. A practical primer. Star Galaxy Publishing, 1998. – 216 p.
4. Vahid F. Digital design with RTL design, VHDL, and Verilog. – John Wiley and Sons Publishers, 2011. – 592 p.

5. Ложкин С.А. Лекции по основам кибернетики. — М.: Изд. Отдел ф-та ВМиК МГУ, 2004. — 256 с.
6. Ж.М. Рабаи, А. Чандракасан, Б. Николич Цифровые интегральные схемы. Методология проектирования. – Вильямс, 2007.
7. T. Lengauer. Combinatorial algorithms for integrated circuit layout. Wiley, 1990, 694 p.
8. Naveed Sherwani. Algorithms for VLSI physical design automation. Kluwer academic publishers, 1995, 538p.
9. Г.Г.Казеннов, В.М.Щемелинин. Топологическое проектирование нерегулярных БИС. М., Высшая школа, 1990, 109с.
10. Introduction to Algorithms, T. Cormen, C. Lesierson, R. Rivest, The MIT Press, Second Printing, 1996.
11. Handbook of Algorithms for Physical design Automation, Edited by Charles J. Alpert Dinesh P. Mehta Sachin S. Sapatnekar, 2009.

Информационные справочные системы:

1. <http://www.asic-world.com/>
2. <https://www.edaplayground.com/>

Материально-техническое обеспечение:

1. аудитория с партами, меловой доской, проектором и экраном.
2. компьютерный класс, оборудованный компьютерами с установленной операционной системой Unix и системами автоматизации проектирования интегральных схем Intel(Altera) Quartus Prime и отладочными платами с программируемыми интегральными схемами Intel(Altera) DE0-Nano, DE0-CV или аналогичными.

9. Язык преподавания - русский.

10. Преподаватели:

к.ф.- м.н., доцент М. С. Шуплецов

11. Авторы программы:

к.ф.- м.н., доцент М. С. Шуплецов