

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет «Московский институт электронной техники»



УТВЕРЖДАЮ
Проректор по УР

А.Г. Балашов

«19» октября 2024 г.

ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

«ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ»

Программа разработана в Передовой инженерной школе
«Средства проектирования и производства электронной компонентной базы»

Москва – 2024 г.

1. Цель реализации программы

Цель программы – формирование у слушателей профессиональных компетенций в области функциональной верификации цифровых устройств.

2. Характеристика профессиональной деятельности и (или) квалификации

Научный проект ПИШ, в котором востребованы компетенции, сформированные в результате обучения по программе - "Создание и развитие отечественных перспективных СФ-блоков для проектирования систем на кристалле на архитектуре RISC-V", "Исследование возможностей создания программно-аппаратного комплекса для ускорения верификации цифровых систем на кристалле".

При разработке программы учтено мнение следующих высокотехнологичных компаний – КНС ГРУПП.

Область профессиональной деятельности: верификация цифровых устройств (как отдельных модулей, так и Систем на Кристалле).

Вид экономической деятельности: деятельность в области информации и связи (ОКВЭД) – 26.11.13 производство интегральных электронных схем.

Укрупненная группа специальностей: 09.00.00 «Информатика и вычислительная техника»

Квалификация: новая квалификация не приобретается.

3. Требования к результатам обучения

Формируемая профессиональная компетенция – Способен осуществлять функциональную верификацию и разрабатывать тесты функционального контроля интегральных схем, определена на основании профессионального стандарта 40.019 «Специалист по функциональной верификации и разработке тестов функционального контроля наноразмерных интегральных схем»,

В результате освоения данной программы слушатель должен:

знать: основные подходы к верификации цифровых устройств, принципы работы симуляторов Verilog и System Verilog,

уметь: проверять простейшие цифровые устройства с одиночным тактовым доменом, получать и анализировать функциональное покрытие.

иметь практический опыт: верификации цифровых устройств.

4. Содержание программы

Учебный план

программы повышения квалификации

«ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ»

Категория слушателей – обучающихся по программе бакалавриата, магистратуры, аспирантуры, а также преподаватели и специалисты.

Срок обучения – 80 часов

Форма обучения очная

№	Наименование разделов/модулей	Всего, час	В том числе		
			Аудиторных		СРС
			Лекции	Практические занятия	
1.	Маршрут проектирования. Введение в функциональную верификацию. Симуляторы SystemVerilog. Запуск и советы по отладке.	8	4	2	2
2.	Последовательностная логика. Циклические и случайные воздействия. Рандомизация. Верификация на основе транзакций	8	4	2	2
3.	Функциональная верификация и создание тестовых сценариев. Взаимодействие с устройством при помощи протоколов.	8	4	2	2
4.	Объектно-ориентированное программирование в SystemVerilog. Создание классов, манипуляции с классами, рандомизация классов	8	4	2	2
5.	Верификационное окружение с применением объектно-ориентированного подхода. Основные компоненты и принципы проектирования. Использование пакетов и интерфейсов.	8	4	2	2
6.	Функциональное и кодовое покрытие.	8	4	2	2
7.	SystemVerilog Assertions. White-box тестирование.	8	4	2	2
8.	Verification Intellectual Property. Разбор полного маршрута функциональной верификации. Верификация многосоставных дизайнов.	8	4	2	2
9.	Работа над индивидуальным проектом.	16	0	0	16
	Итоговая аттестация	зачет			
	Всего	80	32	16	32

Календарный учебный график

Календарный учебный график составляется в форме расписания занятий при наборе группы и прилагается к программе повышения квалификации.

Учебно-тематический план программы повышения квалификации «ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ»

№	Наименование разделов/модулей	Всего, час	В том числе		
			Аудиторных		СРС
			Лекции	Практические занятия	
			очная	очная	очная
1.	Раздел 1. Маршрут проектирования. Введение в функциональную верификацию. Симуляторы SystemVerilog. Запуск и советы по отладке.	8	4	2	2
1.1	RTL to GDSII маршрут проектирования..				
1.2	Верификация ее место в RTL to GDSII.				
1.3	Функциональная верификация. Входные, промежуточные и выходные данные.				
1.4	Как работает цифровое устройство.				
1.5	Введение в SystemVerilog.				
1.6	Простейшие концепции функциональной верификации. Генерация входных воздействий. Проверка выходных значений. Верификационное окружение.				
1.7	Время симуляции.				
1.8	Статические процессы SystemVerilog. Генерация входных воздействий.				
1.9	Временные задержки и разрешающая способность (timescale, timeunit, timeprecision).				
1.10	Проверка выходных значений.				
1.11	Совместное выполнение статических процессов SystemVerilog. Методы синхронизации.				
1.12	Советы по отладке.				
2	Раздел 2. Последовательностная логика. Циклические и случайные воздействия. Рандомизация. Верификация на основе транзакций	8	4	2	2
2.1	Управление временем в SystemVerilog (@, wait()).				
2.2	Общее описание основных принципов верификации последовательностных устройств.				
2.3	Регионы выполнения SystemVerilog. Active Event Region. NBA Event region.				
2.4	Циклы: forever, for, while и т.д.				
2.5	Рандомизация с \$random(), \$urandom(),				

№	Наименование разделов/модулей	Всего, час	В том числе		
	\$urandom_range().				
2.6	Инициализация процессов в SystemVerilog. Стабильность рандомизации.				
2.7	Рандомизация с std::randomize(). Ограничения для рандомизации.				
2.8	\$urandom()/\$urandom_range() versus std::randomize().				
2.9	Общие рекомендации по рандомизации.				
2.10	Концепция разделения обязанностей. Генератор, драйвер, монитор, агент, чекер.				
2.11	Уровни абстракции. Верификация на основе транзакций.				
3	Раздел 3. Функциональная верификация и создание тестовых сценариев. Взаимодействие с устройством при помощи протоколов.	8	4	2	2
3.1	Задачи и функции SystemVerilog. Теория и практика.				
3.2	Тестовые сценарии.				
3.3	Верификационный план.				
3.4	Тестовая регрессия, аргументы командной строки.				
3.5	Система на Кристалле. Взаимодействие между блоками СнК.				
3.6	Стандартные протоколы.				
3.7	Введение в AXI-Stream.				
3.8	Проектирование и применение верификационного окружения для устройства с поддержкой AXI-Stream. Обновление верификационного плана				
3.9	Рандомизация транзакций верификационного окружения. Ключевое слово `rand`.				
3.10	SystemVerilog и динамические процессы. Принцип работы и "хитрости".				
3.11	Концепция master/slave.				
4	Раздел 4. Объектно-ориентированное программирование в SystemVerilog. Создание классов, манипуляции с классами, рандомизация классов	8	4	2	2
4.1	Введение в ООП.				
4.2	Связь ООП и верификации на основе транзакций.				
4.3	SystemVerilog классы. Введение.				
4.4	Поля и методы классов.				
4.5	Рандомизация классов: rand, rand_mode и т.д.				
4.6	Ограничения для рандомизации.				
4.7	Решатель ограничений. Принцип работы.				
4.8	Классы и стабильность рандомизации.				
4.9	Принципы ООП в SystemVerilog. Виртуальные методы. Upcasting/downcasting.				
4.10	Проблема недостаточного уровня абстракции: переиспользование, расширяемость, конфигурируемость.				

№	Наименование разделов/модулей	Всего, час	В том числе		
5	Раздел 5. Верификационное окружение с применением объектно-ориентированного подхода. Основные компоненты и принципы проектирования. Использование пакетов и интерфейсов.	8	4	2	2
5.1	Формулирование основных концепций для решения проблем недостаточного уровня абстракции.				
5.2	Пошаговое проектирование классов верификационного окружения.				
5.3	Разбор конечной структуры верификационного окружения.				
5.4	SystemVerilog пакеты. Функциональное разделение частей верификационного окружения.				
5.5	SystemVerilog интерфейс и виртуальный интерфейс. Использование виртуального интерфейса в верификационном окружении.				
5.6	Примеры решения проблем недостаточного уровня абстракции при помощи ООП.				
5.7	Управление сложностью.				
6	Раздел 6. Функциональное и кодовое покрытие.	8	4	2	2
6.1	Условие завершения тестового сценария.				
6.2	Проблема условия завершения функциональной верификации.				
6.3	Введение в функциональное и кодовое покрытие.				
6.4	План покрытия. Связь плана покрытия с верификационным планом.				
6.5	Конструкции SystemVerilog для создания модели функционального покрытия и сбора функционального покрытия. Команды симулятора для анализа функционального покрытия.				
6.6	Основные типы кодового покрытия. Команды симулятора для сбора кодового покрытия. Команды симулятора для анализа кодового покрытия.				
6.7	Верификационное окружение с функциональным и кодовым покрытием. "Локализация" модели функционального покрытия в верификационном окружении. Производительность.				
6.8	Тестовые регрессии и тестовые сценарии с функциональным и кодовым покрытием. Обновление верификационного плана и плана покрытия. Объединение результатов покрытия.				
7	Раздел 7. SystemVerilog Assertions. White-box тестирование.	8	4	2	2
7.1	SystemVerilog Assertions (SVA). Роль SVA в функциональной верификации.				
7.2	Типы SVA: immediate и concurrent				
7.3	Immediate SVA. Синтаксис, примеры, области применения. Команды симулятора.				

№	Наименование разделов/модулей	Всего, час	В том числе		
7.4	Введение в concurrent SVA. Области применения. Команды симулятора.				
7.5	Примеры использования concurrent SVA.				
7.6.	Подробный разбор concurrent SVA. `sequence`, `property`, их взаимосвязь.				
7.7	Верификационное окружение с SVA. Производительность. "Локализация" concurrent SVA в тестируемом дизайне и верификационном окружении.				
7.8	Тестовые регрессии и тестовые сценарии с SVA. Обновление верификационного плана и плана покрытия. Объединение результатов покрытия.				
7.9	SVA и white-box тестирование. SystemVerilog `bind`.				
8	Раздел 8. Verification Intellectual Property. Разбор полного маршрута функциональной верификации. Верификация многосоставных дизайнов.	8	4	2	2
8.1	Verification Intellectual Property (VIP). Почему актуально и в каких случаях используется?				
8.2	Введение в APB.				
8.3	Проектирование APB VIP.				
8.4	Прохождение полного маршрута функциональной верификации цифрового устройства с использованием VIP. Верификационный план и план покрытия. Тестовые регрессии. Итоговая документация.				
8.5	Обсуждение порядка работы над индивидуальным проектом.				
9.	Работа над индивидуальным проектом.	16	-	-	16
	Итоговая аттестация	зачет			
	Всего	80	32	16	32

**Учебная программа
повышения квалификации
«ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ»**

Раздел 1. Маршрут проектирования. Введение в функциональную верификацию. Симуляторы SystemVerilog. Запуск и советы по отладке. (8 часов).

Тема 1.1. RTL to GDSII маршрут проектирования.

Тема 1.2. Верификация ее место в RTL to GDSII.

Тема 1.3. Функциональная верификация. Входные, промежуточные и выходные данные.

Тема 1.4. Как работает цифровое устройство.

Тема 1.5. Введение в SystemVerilog.

Тема 1.6. Простейшие концепции функциональной верификации. Генерация входных воздействий. Проверка выходных значений. Верификационное окружение.

Тема 1.7. Время симуляции.

Тема 1.8. Статические процессы SystemVerilog. Генерация входных воздействий.

Тема 1.9. Временные задержки и разрешающая способность (timescale, timeunit, timeprecision).

Тема 1.10. Проверка выходных значений.

Тема 1.11. Совместное выполнение статических процессов SystemVerilog. Методы синхронизации.

Тема 1.12. Советы по отладке.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
1.1	Маршрут проектирования. Введение в функциональную верификацию. Симуляторы SystemVerilog. Запуск и советы по отладке	2
1.2		
1.3		
1.4		
1.5		
1.6		
1.7		
1.8		
1.9		
1.10		
1.11		
1.12		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
1.1	Изучение дополнительной литературы для подготовки к практическим занятиям, выполнение домашнего задания.	2
1.2		
1.3		
1.4		
1.5		
1.6		
1.7		
1.8		
1.9		
1.10		
1.11		
1.12		

Раздел 2. Последовательностная логика. Циклические и случайные воздействия. Рандомизация. Верификация на основе транзакций. (8 часов).

Тема 2.1. Управление временем в SystemVerilog (@, wait()).

Тема 2.2. Общее описание основных принципов верификации последовательностных устройств.

Тема 2.3. Регионы выполнения SystemVerilog. Active Event Region. NBA Event region.

Тема 2.4. Циклы: forever, for, while и т.д.

Тема 2.5. Рандомизация с \$random(), \$urandom(), \$urandom_range().

Тема 2.6. Инициализация процессов в SystemVerilog. Стабильность рандомизации.

Тема 2.7. Рандомизация с `std::randomize()`. Ограничения для рандомизации.

Тема 2.8. `$urandom()`/`$urandom_range()` versus `std::randomize()`.

Тема 2.9. Общие рекомендации по рандомизации.

Тема 2.10. Концепция разделения обязанностей. Генератор, драйвер, монитор, агент, чекер.

Тема 2.11. Уровни абстракции. Верификация на основе транзакций.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
2.1	Последовательностная логика. Циклические и случайные воздействия. Рандомизация. Верификация на основе транзакций.	2
2.2		
2.3		
2.4		
2.5		
2.6		
2.7		
2.8		
2.9		
2.10.		
2.11		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
2.1	Изучение дополнительной литературы для подготовки к практическим занятиям, выполнение домашнего задания.	2
2.2		
2.3		
2.4		
2.5		
2.6		
2.7		
2.8		
2.9		
2.10		
2.11		

Раздел 3. Функциональная верификация и создание тестовых сценариев. Взаимодействие с устройством при помощи протоколов. (8 часов).

Тема 3.1. Задачи и функции SystemVerilog. Теория и практика.

Тема 3.2. Тестовые сценарии.

Тема 3.3. Верификационный план.

Тема 3.4. Тестовая регрессия, аргументы командной строки.

Тема 3.5. Система на Кристалле. Взаимодействие между блоками СнК.

Тема 3.6. Стандартные протоколы.

Тема 3.7. Введение в AXI-Stream.

Тема 3.8. Проектирование и применение верификационного окружения для устройства с поддержкой AXI-Stream. Обновление верификационного плана

Тема 3.9. Рандомизация транзакций верификационного окружения. Ключевое слово `rand`.

Тема 3.10. SystemVerilog и динамические процессы. Принцип работы и "хитрости".

Тема 3.11. Концепция master/slave.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
3.1	Тестовые сценарии и верификационный план. Тестовые регрессии. Аргументы командной строки. Система на Кристалле. Стандартные протоколы.	2
3.2		
3.3		
3.4		
3.5		
3.6		
3.7		
3.8		
3.9		
3.10		
3.11		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
3.1	Изучение дополнительной литературы для подготовки к практическим занятиям, прохождение теоретических тестов, выполнение домашнего задания.	2
3.2		
3.3		
3.4		
3.5		
3.6		
3.7		
3.8		
3.9		
3.10		
3.11		

Раздел 4. Объектно-ориентированное программирование в SystemVerilog. Создание классов, манипуляции с классами, рандомизация классов. (8 часов).

Тема 4.1. Введение в ООП.

Тема 4.2. Связь ООП и верификации на основе транзакций.

Тема 4.3. SystemVerilog классы. Введение.

Тема 4.4. Поля и методы классов.

Тема 4.5. Рандомизация классов: rand, rand_mode и т.д.

Тема 4.6. Ограничения для рандомизации.

Тема 4.7. Решатель ограничений. Принцип работы.

Тема 4.8. Классы и стабильность рандомизации.

Тема 4.9. Принципы ООП в SystemVerilog. Виртуальные методы. Upcating/downcasting.

Тема 4.10. Проблема недостаточного уровня абстракции: переиспользование, расширяемость, конфигурируемость.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
4.1	Объектно-ориентированное программирование в SystemVerilog. Создание классов, манипуляции с классами, рандомизация классов.	2
4.2		
4.3		
4.4		
4.5		
4.6		
4.7		
4.8		
4.9		
4.10		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
4.1	Изучение дополнительной литературы для подготовки к практическим занятиям, прохождение теоретических тестов, выполнение домашнего задания.	2
4.2		
4.3		
4.4		
4.5		
4.6		
4.7		
4.8		
4.9		
4.10		

Раздел 5. Верификационное окружение с применением объектно-ориентированного подхода. Основные компоненты и принципы проектирования. Использование пакетов и интерфейсов. (8 часов).

Тема 5.1. Формулирование основных концепций для решения проблем недостаточного уровня абстракции.

Тема 5.2. Пошаговое проектирование классов верификационного окружения.

Тема 5.3. Разбор конечной структуры верификационного окружения.

Тема 5.4. SystemVerilog пакеты. Функциональное разделение частей верификационного окружения.

Тема 5.5. SystemVerilog интерфейс и виртуальный интерфейс. Использование виртуального интерфейса в верификационном окружении.

Тема 5.6. Примеры решения проблем недостаточного уровня абстракции при помощи ООП.

Тема 5.7. Управление сложностью.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
5.1	Верификационное окружение с применением объектно-ориентированного подхода. Основные компоненты и принципы проектирования. Использование пакетов и интерфейсов.	2
5.2		
5.3		
5.4		
5.5		
5.6		
5.7		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
5.1	Изучение дополнительной литературы для подготовки к практическим занятиям, прохождение теоретических тестов, выполнение домашнего задания.	2
5.2		
5.3		
5.4		
5.5		
5.6		
5.7		

Раздел 6. Функциональное и кодовое покрытие. План покрытия. (8 часов).

Тема 6.1. Условие завершения тестового сценария.

Тема 6.2. Проблема условия завершения функциональной верификации.

Тема 6.3. Введение в функциональное и кодовое покрытие.

Тема 6.4. План покрытия. Связь плана покрытия с верификационным планом.

Тема 6.5. Конструкции SystemVerilog для создания модели функционального покрытия и сбора функционального покрытия. Команды симулятора для анализа функционального покрытия.

Тема 6.6. Основные типы кодового покрытия. Команды симулятора для сбора кодового покрытия. Команды симулятора для анализа кодового покрытия.

Тема 6.7. Верификационное окружение с функциональным и кодовым покрытием. "Локализация" модели функционального покрытия в верификационном окружении. Производительность.

Тема 6.8. Тестовые регрессии и тестовые сценарии с функциональным и кодовым покрытием. Обновление верификационного плана и плана покрытия. Объединение результатов покрытия.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
6.1	Функциональное и кодовое покрытие. План покрытия.	2
6.2		
6.3		
6.4		
6.5		
6.6		

6.7		
6.8		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
6.1	Изучение дополнительной литературы для подготовки к практическим занятиям, прохождение теоретических тестов, выполнение домашнего задания.	2
6.2		
6.3		
6.4		
6.5		
6.6		
6.7		
6.8		

Раздел 7. SystemVerilog Assertions. White-box тестирование. (8 часов).

Тема 7.1. SystemVerilog Assertions (SVA). Роль SVA в функциональной верификации.

Тема 7.2. Типы SVA: immediate и concurrent

Тема 7.3. Immediate SVA. Синтаксис, примеры, области применения. Команды симулятора.

Тема 7.4. Введение в concurrent SVA. Области применения. Команды симулятора.

Тема 7.5. Примеры использования concurrent SVA.

Тема 7.6. Подробный разбор concurrent SVA. `sequence`, `property`, их взаимосвязь.

Тема 7.7. Верификационное окружение с SVA. Производительность. "Локализация" concurrent SVA в тестируемом дизайне и верификационном окружении.

Тема 7.8. Тестовые регрессии и тестовые сценарии с SVA. Обновление верификационного плана и плана покрытия. Объединение результатов покрытия.

Тема 7.9. SVA и white-box тестирование. SystemVerilog `bind`.

Перечень практических занятий

Номер темы	Наименование практического занятия	Кол-во часов
7.1	SystemVerilog Assertions. White-box тестирование.	2
7.2		
7.3		
7.4		
7.5		
7.6		
7.7		
7.8		
7.9		

Самостоятельная работа

Номер темы	Вид самостоятельной работы	Кол-во часов
7.1	Изучение дополнительной литературы для подготовки к практическим занятиям, прохождение теоретических тестов, выполнение домашнего задания.	2
7.2		
7.3		