# Лабораторная работа №1.

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

### *Цель лабораторной работы.*

Цель лабораторной работы освоить базовые конструкции языка Verilog. Научиться моделировать простые устройства, научиться проводить симуляцию, тестировать и верифицировать смоделированные устройства, а также реализовывать указанные устройства при помощи программируемых интегральных схем (ПЛИС).

Лабораторная работа состоит из нескольких этапов. В рамках лабораторной 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 лабораторной работы, а после указаны варианты устройств. При этом распределение студентов по вариантам указано в отдельном файле.

### *Этапы лабораторной 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.

**Этап 5. Реализация устройства при помощи программируемых интегральных схем.**

Используя Quartus II, реализовать моделируемое устройство в виде соответствующего файла-прошивки для ПЛИС Altera DE0-Nano. 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. При этом, если периферийных элементов не хватает для того, чтобы подключить все входы устройства, то часть служебных входов можно забить соответствующими константами.

При этом предполагается, что реализуемое устройство должно представлять собой синхронную схему, которая управляется единым тактовым генератором, который представляет собой соответствующим образом сконфигурированный PLL блок. Тактовая частота должна быть подобрана таким образом, чтобы моделируемое устройство проходило все этапы временного анализа и корректно работало на заданной частоте.

***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.
3. 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 (если на входе 0, то и на выходе ).

**Вариант 3. Элементарное арифметическое логическое устройство.**

Схема получает на вход два целых числа, и код операции. Далее схема выполняет соответствующую операцию, и результат поступает на выход схемы. Устройство поддерживает следующие операции: сложение, вычитание. Устройство должно корректно обрабатывать возникающие переполнения.

Вход: два 2-х разрядных провода, на которые подаются значения операндов и одноразрядный провод op, на который подается код операции (0 – сложение, 1 - вычитание).

Выход: 3-х разрядный регистр выходного операнда и однобитовый регистр переполнения (overflow).

**Вариант 4. Код Грея.**

Схема получает на вход двоичное число (лексикографический порядок) и кодирует его в коде Грея. Закодированное число поступает на выход.

Вход: 4-х битовый провод.

Выход: 4-х битовый регистр.

**Вариант 5. Блок вычисления квадрата числа.**

По целому числу, записанному в двоичной системе исчисления, вычисляет квадрат этого числа и записывается в двоичной системе исчисления.

Вход: 3-х битовый провод.

Выход: 6-ти разрядный регистр.

**Вариант 7. Вектор коэффициентов полинома Жегалкина.**

Входной вектор, представляющий вектор значений булевой функции от двух переменных при лексикографическом упорядочивании наборов, необходимо перевести указанный вектор в вектор коэффициентов полинома Жегалкина при лексикографическом упорядочивании коэффициентов.

Вход: 4-х битовый провод, на который подается вектор значений булевой функции.

Выход 4-х битовый регистр для хранения результирующего вектора коэффициентов полинома Жегалкина.