

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»

**ЯЗЫК VERILOG И ПРОЕКТИРОВАНИЕ  
ЦИФРОВЫХ УСТРОЙСТВ НА ПЛИС**

Учебно-методическое пособие

Составители:  
Е. В. Богатилов, А. Н. Шебанов

Воронеж  
Издательский дом ВГУ  
2018

## СОДЕРЖАНИЕ

1. Введение .....	4
2. Архитектура ПЛИС .....	4
3. «Hello, world!» на Verilog.....	8
4. Структура Verilog-проекта .....	11
5. Разработка тестбенча.....	17
6. Параллельные операторы Verilog .....	22
7. Последовательные операторы Verilog.....	26
7.1. Операторы присваивания в Verilog .....	26
7.2. Операторы ветвления и выбора в Verilog .....	31
7.3. Операторы цикла в Verilog .....	36
8. Работа в среде Quartus.....	38
8.1. Пример отладочной платы для ПЛИС .....	38
8.2. Создание нового проекта в Quartus .....	41
8.3. Создание шаблона тестбенча и функциональная верификация .....	42
8.4. Программирование ПЛИС.....	45
Список литературы.....	52
Приложение Справка по языку Verilog.....	53

(Complex Programmable Logic Devices) и FPGA (Field Programmable Gate Arrays). CPLD (исторически произошедшие от программируемых логических матриц) имеют более простую периферию по сравнению с FPGA. Как правило, она состоит из нескольких тактовых входов (на которые подаются тактовые импульсы с внешнего кварцевого генератора), входов программирования, входов питания, а также большого количества (обычно более 100) двунаправленных портов общего назначения, которые могут быть запрограммированы как цифровые входы, либо цифровые выходы. Более сложные FPGA (которые являются эволюцией базовых матричных кристаллов) могут иметь в составе периферии сложные законченные функциональные блоки, например, трансиверы (от слов transmitter — передатчик и receiver — приёмник), поддерживающие такие протоколы как Gigabit Ethernet.

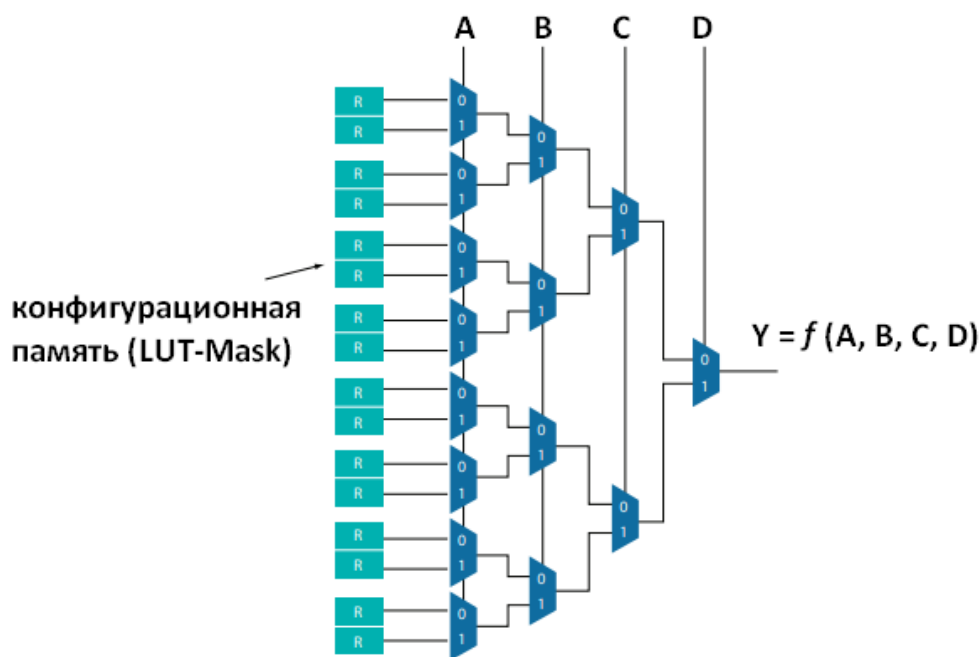


Рис.3. Структура LUT с четырьмя входами

Еще одно отличие между различными классами ПЛИС – место хранения конфигурационной информации. В одних ПЛИС информация о

конфигурации устройства (т.е. о межсоединениях и функциях, выполняемых LUT) хранится в энергонезависимой памяти самой ПЛИС и при включении электропитания загружается в более быстродействующую память SRAM. В других ПЛИС требуется использование внешних энергонезависимых модулей памяти.

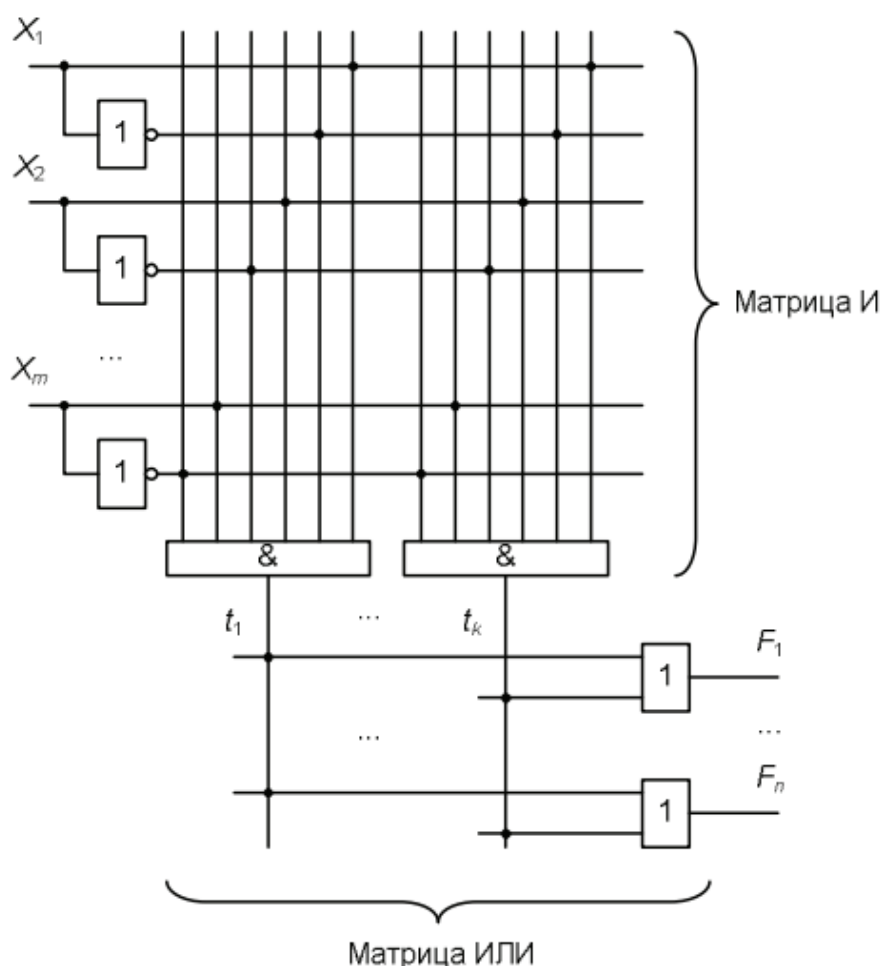


Рис. 4. Структура программируемой логической матрицы AND-OR (PLA)

Перед разработчиками цифровых устройств часто встает вопрос выбора элементной базы. В частности, вопрос выбора между микроконтроллерами и ПЛИС. Принципиальное отличие ПЛИС от микроконтроллеров заключается в том, что структуру цифрового устройства на основе ПЛИС целиком определяет разработчик, программируя соединения в матрице

логических элементов. Если внутреннее устройство микроконтроллеров обеспечивает универсальность при решении различных задач, то устройство ПЛИС обеспечивает максимальную эффективность при решении конкретных требуемых задач. Особенно привлекательно выглядят ПЛИС в тех областях, где необходима параллельная работа большого количества относительно простых цифровых устройств. Если в микроконтроллерах количество одновременно выполняемых задач (без заметного уменьшения производительности) определяется количеством ядер, то в ПЛИС оно определяется только количеством логических элементов ПЛИС и сложностью задач, решаемых цифровыми устройствами.

### **Контрольные вопросы**

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?

### **3. «Hello, world!» на Verilog**

Знакомство с Verilog можно начинать, используя простой и быстрый пакет Icarus-Verilog (является свободно распространяемым). Создадим и проведем моделирование генератора импульсов.

Этап 1. Создание кода.

Icarus-Verilog не содержит редактора кода, поэтому необходимо воспользоваться каким-либо универсальным текстовым редактором. Желательно, распознающим синтаксис языка Verilog (например, Notepad++).

Создадим текстовый файл gen.v следующего содержания (необходимо помнить, что язык Verilog является регистрозависимым):

#### Пример 1.

```
`timescale 1ns/10ps
module gen();
reg signal=0; // Объявляем и инициализируем сигнал
initial // блок задания служебных параметров
begin
    $dumpfile("out.vcd"); // Задаем файл для сохранения
    $dumpvars; // сохраняем все сигналы
    #10 // Задаем продолжительность моделирования
    $finish; // останавливаем моделирование
end
always #(1) signal <= ~signal; // блок, реализующий
                                //инвертирование сигнала
                                //каждую 1 нс
endmodule
```

#### Этап 2. Компиляция проекта

В командной строке Windows необходимо запустить компилятор, указав ему местоположение откомпилированного файла и файла с кодом. Пусть, например, компилятор Icarus находится в папке c:\iverilog\bin, а файл с кодом – в папке c:\iverilog\my. Тогда для запуска компилятора выполняем две команды:

```
cd c:\iverilog\my
c:\iverilog\bin\iverilog -o gencompiled gen.v
```

Первая команда делает текущей папку с проектом. Это необходимо, чтобы при сохранении файла с результатами моделирования out.vcd он был помещен именно в папку проекта. Вторая команда запускает компилятор. Если код содержит ошибки, сообщения о них будут указаны в командной строке. Если ошибок не обнаружено, в папке c:\iverilog\my появится файл с именем gencompiled.

### Этап 3. Симуляция проекта

Не изменяя текущей папки, вводим в командной строке команду запуска симулятора:

```
c:\iverilog\bin\vvp gencompiled
```

В текущем каталоге (которым должен быть каталог проекта `c:\iverilog\my`) должен появиться dump-файл `out.vcd`.

### Этап 4. Просмотр результатов

Для просмотра результатов (они сохранены в dump-файле) в графическом виде может быть использована утилита GTKWave, которая входит в состав Icarus-Verilog. На рис. 5. приведен результат моделирования.

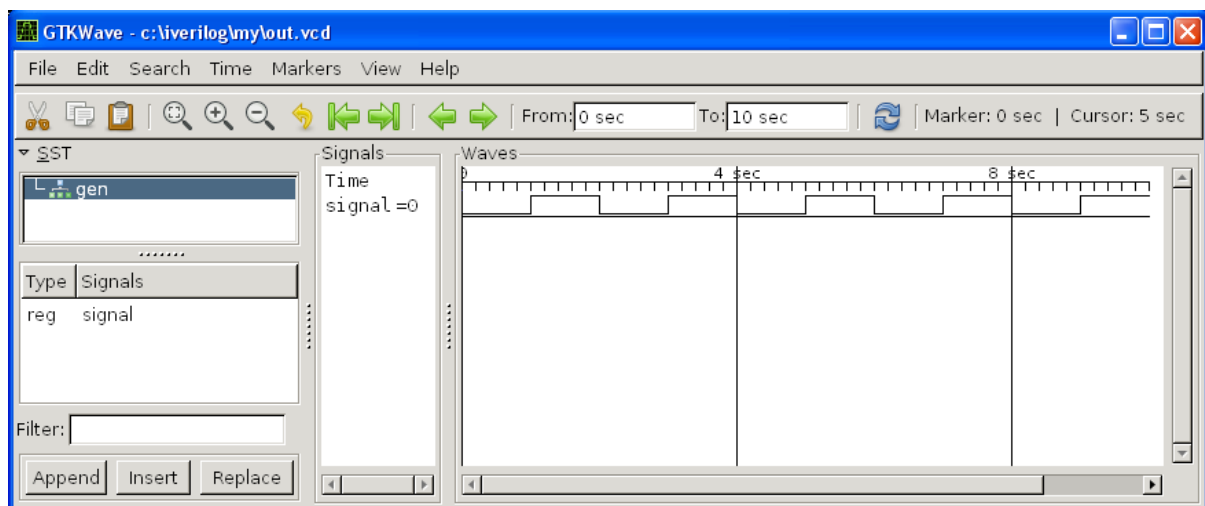


Рис.5. Просмотр результатов моделирования в GTKWave.

### Контрольные вопросы

1. Является ли Verilog регистронезависимым языком ?
2. Как осуществляется компиляция программного кода в Icarus-Verilog ?
3. Как называется программа-симулятор в Icarus-Verilog ?
4. Для чего используется файл с расширением .vcd ?

### Задание 1.