



Вычислительная и микропроцессорная техника

## М11 ЗНАКОМСТВО СО СРЕДОЙ QUARTUS II. СОЗДАНИЕ ПРОСТЕЙШИХ ЦИФРОВЫХ СХЕМ

ъыъ.рф/ыУеЫ

<a href="#">Цель работы</a>	<a href="#">1</a>
<a href="#">Задание на лабораторную работу</a>	<a href="#">1</a>
<a href="#">Выполнение работы</a>	<a href="#">2</a>
<a href="#">Содержание отчета</a>	<a href="#">3</a>
<a href="#">Краткие сведения о САПР Quartus II 15.0</a>	<a href="#">3</a>
<a href="#">Порядок разработки проекта</a>	<a href="#">4</a>
<a href="#">Создание нового проекта в Quartus II 15.0</a>	<a href="#">5</a>
<a href="#">Графический дизайн в Quartus</a>	<a href="#">5</a>
<a href="#">Работа с симулятором</a>	<a href="#">6</a>
<a href="#">Варианты заданий</a>	<a href="#">7</a>

### Цель работы

Познакомиться с некоторыми возможностями системы автоматизированного проектирования Quartus II, основами языка описания аппаратуры Verilog HDL.

### Задание на лабораторную работу

1. Спроектировать комбинационное цифровое устройство согласно [варианту](#) по таблице его функционирования;
2. Получить логические выражения выходных сигналов устройства;
3. Изобразить логическую схему устройства;
4. Написать код на языке Verilog для данного устройства;
5. Построить логическую схему устройства в редакторе Quartus;
6. Выполнить временную симуляцию спроектированного устройства;
7. Получить и проанализировать RTL-схему устройства;

### Выполнение работы

1. Запустите Quartus II 15.0. При первом запуске в окне о лицензии выберите второй пункт.
2. Подключите ModelSim к Quartus. Для этого откройте настройки САПР (**Tools/Options/EDA Tools Options**, скопируйте содержимое поля **ModelSim-A...** в поле **ModelSim**, закройте окно настроек).

3. Проверьте, вызывает ли нажатие клавиши **Prt Sc** приложение **Снимок экрана**. Если нет - откройте настройки клавиатуры (Меню/Настройки/Клавиатура). Перейдите к комбинациям клавиш. Создайте новую комбинацию:
  - Команда **xfce4-screenshooter**,
  - Клавиша **Prt Sc**.
4. Создайте проект: путь **.../VMT/Lab11/**, название **Lab11**.
5. Получите логические выражения для выходов из таблиц функционирования устройств для заданного варианта. Для этого можно воспользоваться картами Карно.
6. Создайте новый файл типа **Block Diagram / Schematic File**. Сохраните файл под именем **lab11\_sch.bdf**.
7. Соберите логическую схему устройства в созданном файле. Сохраните файл. Сохраните результаты работы (скриншот).
8. Сделайте созданный файл файлом высшего уровня (ПКМ по файлу в Project Navigator - Files, Set is Top-Level Entity) (Ctrl+Shift+J). Выполните анализ и синтез проекта  (Ctrl+K).
9. Произведите функциональную симуляцию созданного файла (Сохраните файл под именем **lab11\_sch\_sim.vwf**). В качестве параметров выберите следующие:
  - а - счетчик, начальное значение 0, период 80 нс.
  - b[1:0] - счетчик, начальное значение 0, период 20 нс.
10. Сохраните результат работы (скриншот временной симуляции с развернутыми шинами *от 0 до 160 нс*). При сохранении используйте наибольший возможный масштаб (для помещения в отчет). Закройте симулятор.
11. Создайте новый файл типа **Verilog HDL file**. Сохраните файл под именем **lab11\_hdl.v**.
12. Опишите проектируемое устройство (модуль) при помощи языка описания аппаратуры *Verilog Hardware Description Language*. *Пример описания устройства*:

```
module lab11_hdl ( //название модуля = название файла
    input a, //объявление входного порта a
    input [1:0] b, //объявление входной шины b
    output c, //объявление выходного порта c
    output [1:0] d //объявление выходной шины d
); //завершение описания портов

assign c = (a & !b[0]) | (!a & b[1]); //присваивание
assign d[0] = b[0] & !a & b[1]; //выходных портов
assign d[1] = !b[0] | !a; //
```

```
endmodule //завершение модуля
```

13. Произведите функциональную симуляцию созданного файла (сохраните файл симуляции под именем *lab11\_hdl\_sim.vwf*). В качестве параметров выберите следующие:

- а - счетчик, начальное значение 0, период 80 нс.
- b[1:0] - счетчик, начальное значение 0, период 20 нс.

14. Сохраните результат работы (скриншот временной симуляции с развернутыми шинами, *от 0 до 160 нс*). При сохранении используйте наибольший возможный масштаб (для помещения в отчет). Закройте симулятор.

15. Получите RTL-схему модуля.

16. Сравните полученные результаты (временные диаграммы и схемы).

## Содержание отчета

- Таблица функционирования устройства;
- Логические выражения для выходов устройства;
- Логическая схема устройства;
- Временные диаграммы функционирования схемы от 0 до 160 нс;
- Листинг модуля на языке Verilog HDL;
- Временные диаграммы функционирования модуля от 0 до 160 нс;
- RTL-схема модуля.

## Краткие сведения о САПР Quartus II 15.0

Quartus – САПР, разработанная компанией Altera (ныне подразделение Intel FPGA). САПР предназначена для проектирования и отладки цифровых устройств для СБИС ПЛ компании Altera. Данный цикл лабораторных работ написан для отладочной платы DE1-SoC, использующей систему на кристалле 5CSEMA5F31C6.

Наиболее часто используемые элементы строки меню САПР:

Меню	Элементы	Описание	Hot-key
File	New...	Создать новый файл	Ctrl-N
	Open...	Открыть существующий файл	Ctrl-O
	New Project Wizard	Создать новый проект с помощью мастера создания проектов	
	Open Project...	Открыть существующий проект	Ctrl-J
	Close Project	Закрыть проект	
	Save	Сохранить файл	Ctrl-S
	Save As...	Сохранить файл как...	
	Save All	Сохранить все файлы	Ctrl-Shift-S

	Create / Update	Создание / Обновление	
<b>View</b>	Utility Windows	Используется для настройки окна САПР	
<b>Project</b>	Set as Top-Level Entity	Выбрать файл объектом высшего уровня. Необходимо для выполнения сборки	Ctrl-Shift-J
<b>Assignments</b>	Device...	Выбрать устройство	
	Settings...	Открыть настройки проекта	Ctrl-Shift-E
	Pin Planner	Открыть планировщик контактов FPGA	Ctrl-Shift-N
	Remove Assignments...	Сбросить назначения	
<b>Processing</b>	Start Compilation	Скомпилировать проект	Ctrl-L
	Start Analysis & Synthesis	Анализировать и синтезировать проект	Ctrl-K
	Start I/O Assignment Analysis	Анализировать назначения входов/выходов	
<b>Tools</b>	Netlist Viewer / RTL Viewer	Просмотреть RTL-схему проекта	
	Netlist Viewer / State Machine Viewer	Просмотреть конечный автомат проекта	
	Programmer	Программатор	
	Options...	Открыть настройки САПР	

## Порядок разработки проекта

1. [Создайте новый проект.](#)
2. Создайте файл(ы).
3. Сохраните файл(ы).
4. Выберите необходимый файл объектом высшего уровня.
5. Произведите анализ и синтез проекта -  (Ctrl+K).
6. [Выполните работу с симулятором.](#)
7. Выберите необходимое устройство (если не было выбрано ранее).
8. Выполните планирование входов/выходов проектируемого устройства.
9. Произведите анализ назначений входов/выходов.
10. Скомпилируйте проект.
11. Запрограммируйте DE1-SoC с помощью программатора.

## Создание нового проекта в Quartus II 15.0

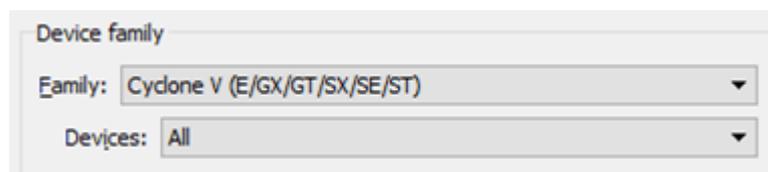
1. Запустите **Quartus II 15.0**.



2. Выберите **New Project Wizard** на приветственной странице САПР или через меню **New - New Project Wizard**.



3. В окне **Introduction** нажмите *Next*.
4. Укажите путь, где будет находиться проект и название проекта в окне **Directory, Name, Top-Level Intity**. Нажмите *Next*. Каждый проект рекомендуется создавать в отдельной папке.
5. В окне **Project Type** установите переключатель  на пункте *Empty Project*, выбрав пустой проект. Нажмите *Next*.
6. В окне **Add Files** нажмите  , чтобы указать пути к файлам, которые вы хотите добавить в новый проект (по умолчанию не требуется). Нажмите *Add*, чтобы добавить файлы. Нажмите *Next*.
7. В окне **Family & Device Settings** выберите Family: *Cyclone V*, Devices: *All*.



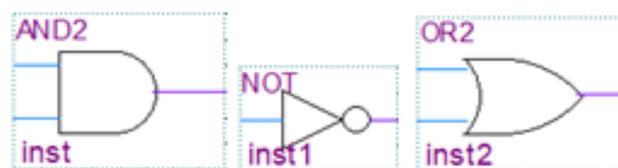
8. В таблице **Available devices** в столбце Name выберите **5CSEMA5F31C6**. Нажмите *Finish*.

## Графический дизайн в Quartus

1. Создайте новый файл типа *Block Diagram / Schematic File*.
2. Создайте необходимые выводы проектируемого устройства при помощи инструмента Pin Tool  , выбирая необходимый тип вывода (input, output, inout). Присвойте выводам названия.



3. Добавьте в проект необходимые элементы при помощи инструмента **Symbol Tool**  , выбирая их в библиотеке элементов. Примеры элементов:



Присвойте элементам названия (по умолчанию не требуется).

Соедините элементы необходимым образом при помощи соединительных линий:

-  – Node Tool – линия из одного провода;
-  – Bus Tool – линия из нескольких проводов одного типа (шина);
-  – Conduit Tool – линия из нескольких проводов разных типов (шина).

Внимание! В одном файле не должно быть элементов с одинаковыми именами. Соединительные линии с одинаковыми именами являются одной и той же соединительной линией.

## Работа с симулятором

1. Создайте новый файл типа *University Program VWF*.
2. В окне **Simulation Waveform Editor** вызовите контекстное меню, нажав ПКМ в левой части рабочего поля. Выберите *Insert Node or Bus*.
3. В окне **Insert Node or Bus** нажмите *Node Finder...*
4. В окне **Node Finder** установите необходимые фильтры поиска (по умолчанию нет) и нажмите *List*.
5. Выберите необходимые сигналы из найденных в левой части окна и скопируйте их в правую часть, нажав  или , если хотите выбрать все (по умолчанию все). Нажмите ОК.
6. В окне **Insert Node or Bus** нажмите ОК.
7. Расположите сигналы в нужном порядке, перетаскивая их мышью.
8. Задайте необходимые параметры симуляции, используя инструменты на верхней панели окна **Simulation Waveform Editor** и выделяя нужные временные промежутки нужных сигналов:
  -  – установить неопределенное состояние на выбранном промежутке;
  -  – установить высокоимпедансное состояние;
  -  – установить логический нуль;
  -  – установить логическую единицу;
  -  – установить состояние «Low»;
  -  – установить состояние «High»;
  -  – инвертировать выбранные промежутки;
  -  – задать счетчик;
  -  – задать тактирующий сигнал;
  -  – задать случайные значения.
9. Измените основание системы счисления необходимых сигналов, щелкнув ПКМ на сигнале и выбрав *Radix* (по умолчанию не требуется).
10. Сохраните файл.

## 11. Запустите функциональную симуляцию при помощи инструмента



Run Functional Simulation

в меню **Simulation**.

### Варианты заданий

Номер варианта выбирается по указанию преподавателя

1	<b>Входы</b>	<b>Выходы</b>	2	<b>Входы</b>	<b>Выходы</b>	3	<b>Входы</b>	<b>Выходы</b>													
	a	b[0]		b[1]	c[0]		c[1]	d	a	b[0]	b[1]	d[0]	d[1]	d[2]	a	b[0]	b[1]	e	f[0]	f[1]	
	0	0		0	1		0	0	0	0	0	1	0	1	0	0	0	1	1	1	1
	0	0		1	0		1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
	0	1		0	0		0	1	0	1	0	0	0	1	0	1	0	0	1	1	1
	0	1		1	0		1	0	0	1	0	0	0	0	1	0	1	0	1	0	0
	1	0		0	1		1	0	1	0	0	0	0	1	1	0	0	0	1	0	0
	1	0		1	0		1	1	1	1	1	1	1	0	0	1	0	1	0	1	0
	1	1		0	0		1	0	1	1	0	0	0	0	0	1	1	0	1	0	0
1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1			

4	<b>Входы</b>	<b>Выходы</b>	5	<b>Входы</b>	<b>Выходы</b>	6	<b>Входы</b>	<b>Выходы</b>													
	a	b[0]		b[1]	f		g[1]	g[0]	a	b[0]	b[1]	g[0]	g[1]	h	a	b[0]	b[1]	h	j[1]	j[0]	
	0	0		0	1		1	0	0	0	0	1	0	0	0	0	0	0	1	1	0
	0	0		1	0		1	0	0	1	0	0	0	0	0	0	1	0	1	1	0
	0	1		0	0		1	0	0	1	0	1	0	1	0	0	1	0	0	1	0
	0	1		1	0		1	0	0	1	0	0	0	0	0	1	0	1	1	1	0
	1	0		0	1		1	1	1	0	0	0	1	1	1	0	0	0	1	1	0
	1	0		1	0		0	1	1	1	0	0	0	0	0	1	0	1	0	1	1
	1	1		0	1		1	0	1	1	0	1	0	1	1	1	1	0	1	1	1
1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1			

7	<b>Входы</b>	<b>Выходы</b>	8	<b>Входы</b>	<b>Выходы</b>	9	<b>Входы</b>	<b>Выходы</b>													
	a	b[0]		b[1]	k		j[0]	j[1]	a	b[0]	b[1]	j[1]	j[0]	k	a	b[0]	b[1]	k[0]	k[1]	l	
	0	0		0	1		1	1	0	0	0	0	1	1	0	0	0	0	1	1	1
	0	0		1	0		0	0	0	0	1	0	0	1	0	0	0	1	1	1	1
	0	1		0	0		1	0	0	1	0	0	1	0	0	0	1	0	0	0	0
	0	1		1	0		0	0	0	1	0	0	0	1	0	0	1	0	1	0	0
	1	0		0	0		0	1	1	0	0	0	0	0	0	0	1	1	1	1	1
	1	0		1	1		1	0	1	0	0	0	0	0	0	1	0	1	1	1	1
	1	1		0	1		1	0	1	1	0	0	0	0	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	1			

10	<b>Входы</b>	<b>Выходы</b>	11	<b>Входы</b>	<b>Выходы</b>	12	<b>Входы</b>	<b>Выходы</b>												
	a	b[0]		b[1]	l[0]		l[1]	m	a	b[0]	b[1]	m	n[1]	n[0]	a	b[0]	b[1]	n[0]	n[1]	o
	0	0		0	0		0	0	0	0	0	1	1	0	0	0	0	0	1	1
0	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	0	

0	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0
0	1	1	0	1	0	0	0	1	1	0	0	0	1	0	1
1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0
1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	0
1	1	1	0	1	0	1	1	1	1	1	1	1	1	0	1

13

Входы			Выходы		
a	b[0]	b[1]	o[0]	o[1]	p
0	0	0	0	1	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1

14

Входы			Выходы		
a	b[0]	b[1]	p	q[1]	q[0]
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	0	0	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	0	1	1

15

Входы			Выходы		
a	b[0]	b[1]	q[0]	q[1]	r
0	0	0	1	1	0
0	0	1	1	1	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	1	1	0

16

Входы			Выходы		
a	b[0]	b[1]	r	s[0]	s[1]
0	0	0	0	0	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

17

Входы			Выходы		
a	b[0]	b[1]	s	t[1]	t[0]
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	0

18

Входы			Выходы		
a	b[0]	b[1]	t	u[0]	u[1]
0	0	0	1	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1

19

Входы			Выходы		
a	b[0]	b[1]	u[1]	u[0]	v
0	0	0	1	1	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	1	1

20

Входы			Выходы		
a	b[0]	b[1]	v[1]	v[0]	w
0	0	0	1	0	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	1	1	1

21

Входы			Выходы		
a	b[0]	b[1]	w	x[0]	x[1]
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	1	0	0