В отчете по лабораторным работам необходимо описать: Функциональное назначение устройства и способ составления таблицы истинности, логических функций, принципиальной схемы, теоретических временных диаграмм, описания на языке Verilog.

Разрабатываемое устройство называется «проект». Каждый проект в САПР Quartus описывают несколько файлов с одинаковыми именами и различными расширениями. Для проектов всех работ создайте рабочий каталог D:\users\845\Lab_Quartus\ . Проектирование в Quartus содержит следующие этапы.

1. Создание нового файла - *File > New.* Для ввода схемы выберите тип файла *Block Diagram / Schematic File* а для ввода описания - *Verilog HDL File,* нажмите *OK.* **Запишите новый файл в рабочий каталог** командой *File / Save As*, введите имя файла. Расширение, для схемы *.bdf, для описания на Verilog *.v.

2. Создание нового проекта. Для каждого разрабатываемого устройства **всегда необходимо создавать** новый проект командой *File > New Project Wizard*. В окне с именем Introduction нажмите *Nex. B*о втором окне (Directory, Name…) в первой строке откройте Рабочий каталог проектов, а, а во второй - имя разработанного файла, нажмите *Finish.* На предложение создать новый каталог отвчайие «нет». Для работы с **ранее разработанным устройством откройте** проект*File > Open Project*, затем, из меню View, или Alt+0, откройте Project Manager, чтобы открыть файл со схемой или с описанием.

3.1. Ввод схемы. Назначение инструментов графического редактора. **1**-Отделение окна схемы от окна программы (при копировании схемы). **2** – выбор элемента, фрагмента схемы, или текста для перемещения, копирования, удаления. Клавиша *Ecs* - переход в этот режим. Правая кнопка мыши открывает контекстное меню доступных функций. **4** - ввод символов элементов, открывает окно *Symbol.* **6** - ввод проводников. **7** - ввод шин из нескольких проводников. **9** - режим неразрывных «резиновых» проводников, при перемещении элементов сохраняются все связи. **10** – вазрывные проводники, разделение схемы на отельные фрагменты для и составления новой схемы из элементов старой, которая сохранена с новым именем (Save As). **11** – масштаб, изменяетсялевой или правой кнопками мыши.

Ввод схемы - это ввод символов элементов и их соединение – логическое, или физическое.

Ввод символа. В окне Symbol (открывается кнопкой 4 или двойным щелчком левой кнопки на свободном месте схемы) откройте *Libraries / primitives*. Логические элементы извлекаются двойным щелчком из папки *logic*, а терминалы ввода-вывода - из папки *pin*. Двойной щелчок по символу терминала открывает окно *Pin Properties* (свойства), в котором необходимо указать имя. Для терминала шины после имени указывают в квадратных скобках индексы старшего и младшего разрядов, разделенные двумя точками. Если в схеме используются отдельные разряды шины, то необходимо к выводу терминала дорисовать шину в виде толстой линии (инстру мент 7), выделит шину, и ввести над ее обозначением все используемые сигналы через запятую. Для копирования элемента установите указатель на символ, нажмите клавишу Ctrl и переместите копию элемента в другое место. Также используют выделение элемента или фрагмента, Ctrl-C и Ctrl-V.

Физическое соединение элементов выполняют проводники и шины. Для соединения двух точек проводником нажмите кнопку инструмента 6, курсор мыши примет вид крестика, поместите курсор на исходную точку, затем, удерживая нажатой левую кнопку мыши, переместите курсор во вторую желаемую точку и отпустите кнопку мыши. Для фиксации положения проводника на схеме необходимо отпустить и вновь нажать левую кнопку. Для редактирования вида цепи, необходимо нажать Esc, выделить цепь на схеме, а затем с помощью контекстно-зависимого меню, вызываемого по правой клавише мыши, выполнить желаемое действие.

Л**огическое соединение - это присваиванием одинаковых имен** проводникам, или шинам, которые должны быть соединены. Необходимо щелчком левой кнопкой мыши выбрать проводник, или шину, и при появлении мигающего курсора ввести имя цепи и нажать ОК. Имя нельзя присвоить выводу элемента, поэтому необходимо к выводу дорисовать отрезок проводника, заканчивающегося на свободном месте схемы, которому можно присвоить имя. Для ввода шины , используется кнопка 7, после имени необходимо указать в квадратных скобках индексы старшего и младшего разрядов, разделенные двумя точками.

3.2. При вводе описания открывается текстовый редактор, который пояснений не требует.

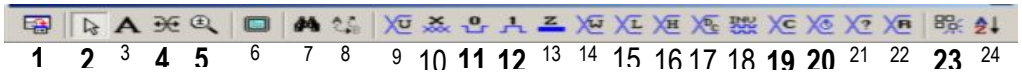
4. Компиляция - *Processing > Start Compilation* предварительная обработка и проверка проекта. **Компиляция выполняется, если был создан проект.** Для создания проекта существующего файла его необходимо добавить в каталог проекта командой Save As, устанавливая флажок Add file to current project. Наличие предупреждений (warnings) не препятствует продолжению работы.

5. Создание символа - *File > Create/Update > Create Symbol Files for Current File*, ОК. Окно со схемой, или описанием должно быть активным. Символы модулей используют в проектах подобно подпрограммам.

6. Создание файла для временных диаграмм *File > New*, выберите *Vector Waveform File*, нажмите ОК. **Запишите файл с именем проекта** и расширением .vwf командой *File > Save As*.

7. Выбор контрольных точек. Двойным щелчком левой копкой в поле имен узлов и шин сигнального редактора откройте окно Insert Node or Bus, нажмите кнопку Node Finder (Регистратор узлов). В следующем окне нажмите кнопку *List,* кнопку с символами «>>», затем ОК, ОК. В строке *Filter* должно быть Pivs all.

8. Ввод тестовых входных сигналов. Выполняется в соответствии с теоретическими временными диаграммами с помощью инструментов сигнального редактора. Номера часто используются инструменты



1- отделение окна диаграмм от окна редактора. **2** – выделение временной диаграммы, или ее фрагмента (перемещением указателя при нажатой левой кнопке) для ввода сигналов 0 или 1 или для инвертирования. **4** – **инверсия ыделенноой части диаграммы**. **5** - изменение маштаба левой или правой кнопклй мыши. **11** или **12** - задание сигнала 0 или 1 для выделенной диаграммы или ее части. **19** - задание сигнала от счетчика на проводнике, или шине. **20** – задание синхросирнала. **23** – включение режима привязки к сетке фронтов формируемых импульсов.

Для подачи тестовых входных сигналов на комбинационную схеу необходимо щелчком левой кнопки по терминалу выделить временную диаграмму проводника (или шины) и подключить сигнал от счетчика – кнопка 19 с буквой «С». В окне *Count Value*, закладка Counting укахите систему счисления Radix -Hexadecimal, закладка *Timing* - установите интервал изменений состояний счетчика Count Every равным 50 нс, при этом период будет 100 нс. Множитель Multiplied By, определяет увеличение периода) по оси времени, для первого проводника он равен 1, для каждого следующего – вдвое больше. Остальные параметры – по умолчанию.

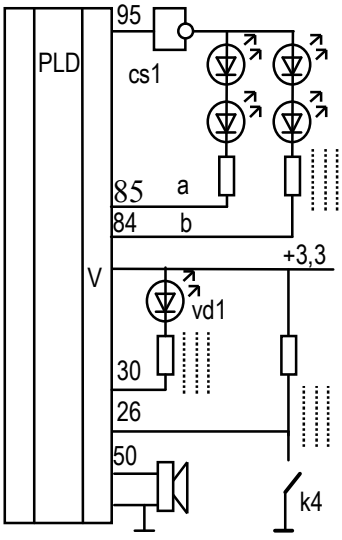
Тестовые входные сигналы для триггеров и счетчиков. Синхросигнал *Clock* включается кнопкой 20, необходимо задать Period – период 50 нс, остальные параметры – по умолчанию. Чтобы нарисоввать управляющие импульсы в произвольные моменты времени необходимо вначале выделить диаграмму, задать исходное значение сигнала (например, 0 кнопкой 11), а затем нажать кнопку 4 и промнверитроватьт выделяемые курсором участки диаграммы для получения необходимых положительных импульсов.

9. Этап. Моделирование устройства запускается командой Processing > Simulator. В окне Processing > Simulator Tool выберите режим моделирования Timing, при котором отображаются временные задержки, или Functional для предварительного ознакомления с работой схемы, без отображения задержек.

Анализ результатов моделирования выполняется по временным диаграммам, которые отображаются в окне **Report.** Это сравнение полученных экспериментальных временных диаграмм с теоретическими и **измерение временных задержек** для оценки быстродействия устройства.

Для измерения временных задержек выберите на временных диаграммах момент начала интервала измерения (например, фронт входного сигнала). Включите привязку к сетке кнопкой 23 Snap to Grid. Установите указатель мыши над выбранным участком диаграмм, и двойным щелчком установите основной маркер (Maser Bar), обозначенный сплошной линией. Отключите привязку к сетке. Двойным щелчком установите штриховой маркер (Time Bar) на конец измеряемого интервала. Задержка определяется как разность координат маркеров. Привязка к сетке позволит точно совмещать фронты логических сигналов с моментами времени, кратными шагу сетки Grid Size, по умолчанию 10 нс. Длительность моделирования устанавливает команда Edit > End Time, по умолчания 1 мкс.

Этап 10. Программирование ПЛИС. 1)Назначение ПЛИС В окне Assidement / Device укажите семейство - Family: MAXII, используемое устройство - Available device: EPM240T100C5. Нажмите кнопку Device and Pin options, на закладке Unused Pins, укажите состояние Input Try Ststes. 2) Включите редактор подключения выводов Assigement/Assigement Editor, установите режим View/Show all Assignable Pin Numbers 3) Установите курсор мыши в нижнюю строку левого столбца таблицы. Двойным щелчком откройте список в троке, выберите Node Finder. В открывшемся окне нажмите List, затем кнопку « >> », затем ОК. 4) Выберите элементы отладочной платы, которые необходимо подключить к проекту в ПЛИС. 5) Для выбранного терминала двойным щелчком откройте окошко с номером контакта (например, Pin_1) и скорректируйте номера контакта. 6) Запишите результат назначения File > Save. Номера контактов должны появиться на схеме. Выполните компиляцию. 6) Подключите отладочную плату к компьютеру, затем включите программатор к USB- разъему. Из меню выберите Tools > Programmer , установите галочки в столбце Program/Configure, нажмите Start.



Светодиоды		Индикатор			
		сегмент		цифра	
№	Pin	№	Pin	№	Pin
vd1	58	a	85	1	95
vd2	57	b	84	2	92
vd3	56	c	83	3	91
vd4	55	d	82	4	80
vd5	54	e	81	5	89
vd6	53	f	78	6	88
vd7	52	g	77	7	87
vd8	51	h	76	8	86

Кнопки: k4-26; k3-27; k2-28; k1-29; clr-44

Подключение элементов к ПЛИС

Основы языка Verilog.

Описание устройства на Verilog позволяет, используя САПР, запрограммировать ПЛИС и получить аппаратную реализацию описанного устройства.

Сигналы, действующие в цифровых схемах, являются переменными (аргументами и функциями) языка Verilog. Они могут принимать значения: ***0, 1, z, x.*** Значение *z* соответствует отключенному состоянию выхода элемента (разрыв цепи), а *x* – неопределенному (любому) значению - 0 или 1. Все сигналы по умолчанию имеют **тип сигнала *wire.*** Этопровод, к которому непрерывно прилагается воздействие от логического элемента - драйвера (driver). Он передает асинхронно (сркзу же после изменения) все изменения сигнала, Входные сигналы имеют тип wire. разрядность сигналов 1 бит, а

Описание комбинационных схем выполняют **параллельные операторы** с ключевым словом ***assign*** – назначать, после которого записываются выражения для операций. Формируют выходной сигнал типа **wire**.

Описание устройств с элементов памяти выполняют последовательные операторы с ключевым словом ***always*** – всегда, которые формируют в**ыходные сигналы** типа **reg**, который необходимо указывать.

Вектор - это параллельный код, который состоит из нескольких разрядов, передается через группу проводников, называемую шиной. При описании вектора перед именем в квадратных скобках через двоеточие указывают индексы старшего, а затем младшего разрядов. Для использования в описании одного элемента вектора (проводника шины), необходимо указать имя вектора и индекс проводника в квадратных скобках.

Арифметические	
Умножение	*
Деление	/
Сложение	+
Вычитание	-
Остаток от деления	%
Логические	
И	**&**
ИЛИ	|
НЕ	~
Искл. ИЛИ	^
И-НЕ	~&
ИЛИ-НЕ	~|
Искл. ИЛИ-НЕ	~^
Сравнения	
Равно?	==
Не равно?	!=
Больше ?	>
Меньше ?	<
Больше или равно ?	>=
Меньше или равно ?	<=
Специальные	
Объединение	{a,b,c}
Условное присваивание	q= x?a:b

Константы имеют краткий или полный формат записи. Краткий формат содержит число в десятичной системе счисления, разрядность определяется компилятором в зависити от значения числа. Полный формат содержит 4 элемента: 1) число, определяющее количество разрядов в константе; 2) одинарная кавычка; 3) буква, определяющая основание системы счисления (b - двоичная, h - 16-ричная d - десятичная); 4) цифры в указанной системе счисления. Для отрицательных чисел указывают знак «минус» в самом начале записи. В двоичной системе допускаются символы z и x. Краткий формат проще, разрядность числа может изменяться. Полный формат используют в описаниях сложных устройств, где требуется строго определить разрядность. Например, краткий формат записи констант 1 и 5 имеет такой же вид 1 и 5, а полный формат в двоичной системе - 1'b1 и 3'b101.

При записи параллельных и последовательных операторов в языке Verilog используются различные операции, с использованием определенных символов.

Арифметические операции могут выполняться над векторами, разрядность которых ограничивается информационной емкостью ПЛИС. Наибольшие аппаратные затраты требуются для выполнения операций умножения и деления.

Логические операции выполняются поразрядно для одноразрядных аргументов и векторов. Если аргументы - многоразрядные векторы, имеющие различную разрядность, то по умолчанию разрядность выравнивается по вектору максимальной разрядности. Вектор меньшей разрядности дополняется слева нулями.

Операции сравнения выполняются над векторами различной разрядности, при этом старшие разряды вектора меньшей разрядности заполняются нулями. Результат операций сравнения - 1 бит, равный 1, если условие выполняется,используется как признак при выполнении условных операторов.

Операция объединения формирует вектор суммарной разрядности, составленный из перечисленных в фигурных скобках векторов. Суммарный вектор может быть записан слева и справа то знака присваивания.

Операция условного присваивания описывает структуру «Если – то – иначе» при использовании поведенческого описания - по алгоритму функционирования, имеет вид: *assign q = x ? a : b;* . В этой записи q – результат , x - условие, згак ? – прмзнак условного оператора, a – ветвь ДА, b – ветвь НЕТ. Выполняется так: Если x = 1, то q = a, иначе q = b. В качестве a и b могут быть использованы одноразрядные сигналы, векторы, или выражения. Условие x должно иметь вид одноразрядного сигнала или выражения в скобках, результат которого имеет разрядность 1 бит.

В описаниях использкют два типа операторов, содержащих отдельные операции.

Параллельные операторы. Каждая операция начинается с ключевого слова **assign** –назначать. Срабатывание оператора (назначение нового значения выходного сигнала) проимходит сразу же после изменения любого входного сигнала. Все сигналы имеют тип wire. Позволяют описывать комбинационные схемы.

Последовательные операторы начинается с ключевого слова **always** –всегда. Позволяют описывать любые схемы – комбинационные и схемы с элементами памяти. Для выходных сигналов должен быть указан тип reg. Срабатывание оператора указывается в списке чувствительности, которыйчинается с символа **@** ,

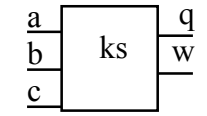
определяющего **«событие»**.В последовательных операторах кроме рассмотренных операций используются операторные скобки begin – tnd, а также операторы if и case.

Оператор условного перехода «if» используется только с последовательным оператором «always», имеет формат **if (** условие **)** ветвь да **; else** ветвь нет **;**

Условие - переменная, или выражение, записывается в круглых скобках. Ветвь ДА - операция, котораявыполняется, если условие истинно. Ветвь НЕТ, выполняется, если результат вычисления условия равен 0, или содержит значения x, z. Заметим, в записи оператора слово then отсутствует.

Оператор варианта case имеет формат:

case (селектор**)** вариант селектора 1 **:** присваивание 1**;** вариант селектора2 **:** присваивание 2**;** default **:** присваивание по умолчанию **; endcase**



$q = a \cdot \bar{b} \cdot c \vee a \cdot b \cdot c$

$\bar{q} = \bar{a} \cdot b \cdot c \vee a \cdot b \cdot c$

Пример 1. Составление описания комбинационной схемы. Вначале необходимо изобразить символ КС с обозначениями всех входных и выходных сигналов, а также логические функции или таблицу истинности.

1 строка - комментарий. определяет функциональное назначение устройства, не влияет на компиляцию, содержит русские буквы, начинается с двух символов «косая черта» (знак деления на клавиатуре).

2. Заголовок описания начинается с ключевого слово ***module*** затем через пробелы имя модуля и перечисление всех входных и выходных сигналов в круглых скобках через запятую. Завершение всех операторов – точка с запятой. **Имя модуля** (а также и идентификаторы сигналов) должно начинаться с латинской буквы, может содержать также цифры и символы подчеркивания, пробелы и русские буквы недопустимы. Регистр имеет значение! Обычно используют нижний регистр. Имя фацла с данным описанием должно в точности совпадать с именем, записанным в заголовке.

3. Описание входных и выходных сигналов. После ключевого слова ***input*** перечисляются входные сигналы. Входные сигналы всегда имеют тип wire. Запись ***input a,b,c;*** назначает по умолчанию одноразрядные сигналы. Запись ***input [3..0] x,y;;*** назначает 4-разрядные сигналы. После ключевого слова ***output*** указывают выходные сигналы. Запись ***output [3..0] z;;*** назначает 4-разрядный выходной сигнал типа wire. Выходные сигналы последовательных операторов должны имеь тип reg Этот тип необходимо указывать в описании: ***output [3..0] q;; reg [3..0] q;***

```
/Задание 11, КС пример 1
module v11_ks1 (a,b,c,q,w);
 input a, b,c;
output   q, w;
assign q=a&~b&c| a&b&c;
assign w= ~a&b&c| a&b&c;
endmodule
```

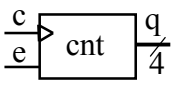
```
/Задание 11, КС пример 2
module v11_ks2 (a,b,c,q,w);
 input a, b,c;   output   q, w;
alays
btgin
 q=a&~b&c| a&b&c;
w= ~a&b&c| a&b&c;
and;
endmodule
```

4. Описание комбинационной схемы параллельными операторами по логическим функция. Необходимо для каждой функции записать параллельный оператор, в котором после ключевого слова *assign* записывается заданная логическая функция, в который используются символы логических операций принятые в языке Verilog. Операция И обозначается знаком амперсанд (&), операция ИЛИ – вертикальной чертой (|), операция НЕ – знаком тильда (~), а операция исключающее ИЛИ – знаком (^).

5. Завершение описания - слово *endmodule*, после которого не должно быть никаких знаков препинания.

Пример 2. Описание комбинационной схемы последовательным оператором. Для выходных сигналов указывается тип reg. Записывается ключевое слово *always,* после которого отсутствует список чувствительности.. В скобках btgin – end записываются выражения (которые могут содержать операторы if, case). Срабатывание оператора (изменение выходных сигналов) будет происходить всегда после изменения любого из входных сигналов, подобно параллельному оператору.

Пример 3. Задано составить описание суммирующего 4-разрядного счетчика с входрм разрешения, который при e = 1 увеличивает выходной код qна 1 по каждому фронту синхросигнала c.



Описание схемы с элементами памяти последовательным оператором, который содержит список чувствительности, указывающий сигналы и события, определяющие срабатывание оператора.

Символ @ обозначает понятию «событие». Запись (posedge c) означает фронт сигнала c (positive edge – положительный край). Срабатывание оператора в данном примере задает строка: «всегда по фроту сигнала c».

```
/Задание 12, счетчик
module v12_cnt (c,e,q);
 input c, e;
output  [3:0] q; reg [3:0] q;
alays @ (postdge c)
if (e)  q=q+1;
endmodule
```

В заголовке указано имя модуля и перечислены все входные и выходные сигналы. Входные сигналы по умолчанию имеют разрядность и бит. Выходной сигнал указан как 4-разрядный вектор типа reg.

Обработку описывает последовательный операторсо списком чувствительности, который читается так. Всегда по фронту сиг8нала с если e = 1, то присваивать q:= q+1.