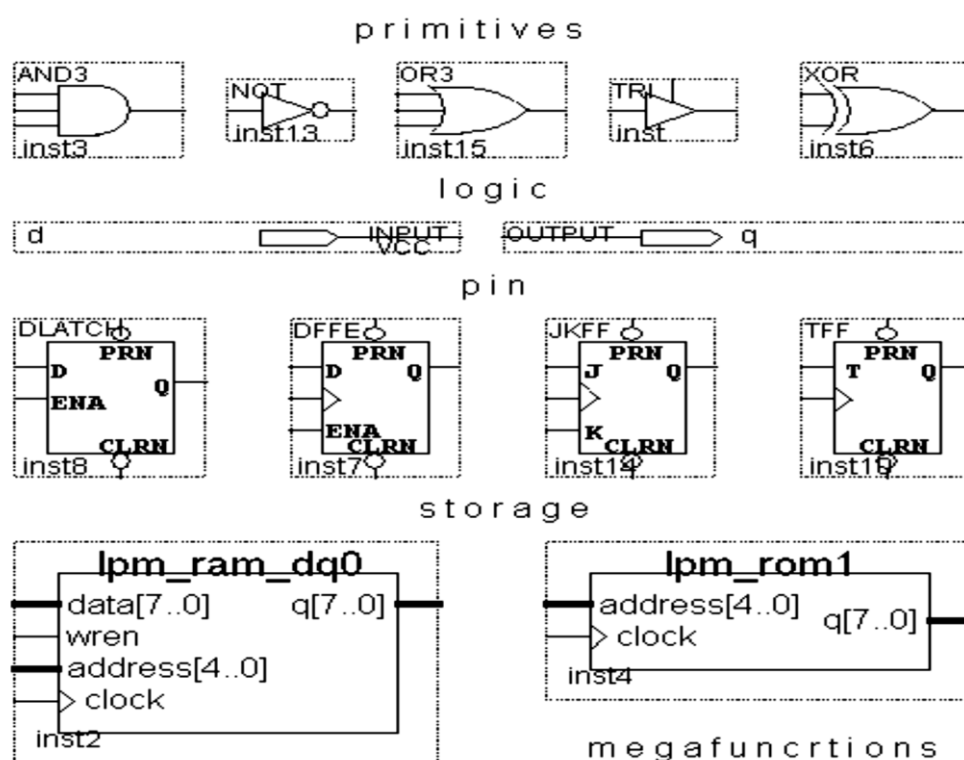


## Схемотехническое проектирование

### Часть 1

Методические указания к лабораторным работам и к практическим занятиям  
по курсам «Электроника и схемотехника», «Проектирование цифровых  
устройств»  
«Схемотехническое проектирование»



Элементы библиотеки системы Quartus

Рязань 2022

Цифровые устройства обработки данных применяются во всех сферах деятельности человека. Для построения цифровых устройств в настоящее время используют интегральные схемы различной степени интеграции. Наиболее перспективными являются программируемые логические интегральные схемы (ПЛИС), применение которых сокращает сроки и стоимость разработки, обеспечивает высокую надежность и производительность, а также низкую стоимость разрабатываемых систем. Изучение цифровых устройств выполняется на примерах проектирования применительно к их реализации в ПЛИС с

использованием САПР Quartus. Такой подход позволяет освоить работу с САПР, а также выполнить моделирование, проверить корректность разработанного проекта, определить параметры устройства. Каждая изучаемая тема содержит теоретический материал в виде отдельных параграфов и задания на экспериментальные исследования.

## § 1.1. Базовые логические элементы цифровых схем

Теоретической основой цифровых устройств является булева алгебра, которая определяет *базис логических элементов*. Базис - набор простых логических функций и соответствующих элементов, с помощью которых можно представить любую сколь угодно сложную логическую функцию. Основой для анализа и синтеза цифровых схем является *булев базис*, содержащий операции И, ИЛИ, НЕ. При производстве интегральных микросхем типа ТТЛ использовался *универсальный базис*, содержащий один элемент И — НЕ (штрих Шеффера). Подобным образом, элемент ИЛИ — НЕ (стрелка Пирса) является *универсальным базисом*.

Кристаллы ПЛИС содержат большой набор различных логических элементов (их называют вентили gates), имеющих от 2 до 12 входов соответствующих расширенному базису, который кроме элементов булева базиса содержит элементы с инверсными входами, а также элемент Иключающее ИЛИ. В таблице истинности для основных логических элементов, используемых для построения цифровых схем, приведены логические функции.

Расширенный базис обладает избыточностью, но позволяет оптимизировать параметры разрабатываемых устройств, повысить быстродействие, упростить схему. Условные графические обозначения соответствуют стандарту, принятому в России, в прямоугольнике обозначают функциональное назначение элемента (для элемента И записывается символ «&», называемый «амперсанд», для ИЛИ – цифра 1, для элемента сумма по модулю 2 –  $\oplus$  символ

Символы, используемые в графических редакторах САПР, соответствуют американскому стандарту. Функциональное назначение элемента и количество входов указывает надпись, and означает - элемент И, or – или, eor (exclusive or) – Иключающее ИЛИ.

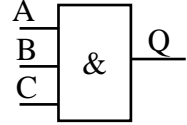
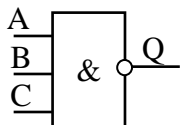
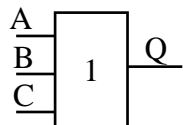
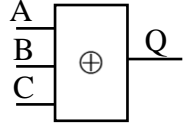
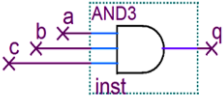
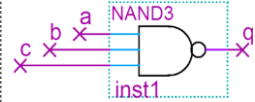
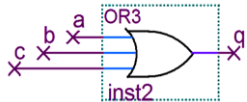
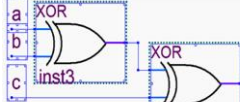
|                                  | И<br>Конъюнкция<br>$Q = A \cdot B \cdot C$  | И-НЕ<br>Штрих Шеффера<br>$Q = \overline{A \cdot B \cdot C}$                          | ИЛИ<br>Дизъюнкция<br>$Q = A \vee B \vee C$  | Искл.ИЛИ<br>Сумма по модулю 2<br>$Q = A \oplus B \oplus C$                            |
|----------------------------------|---|--|---|---|
| Логическая функция               |   |  |   |   |
| Условное графическое обозначение |  |  |  |  |
| Символ в САПР                    |  |  |  |  |

Рис. 1.1. Базовые логические элемент

### Аксиомы для основных логических операций

$$a \cdot 0 = 0; \quad a \cdot 1 = a; \quad a \cdot a = a; \quad a \cdot \bar{a} = 0;$$

$$a \vee 0 = a; \quad a \vee 1 = 1; \quad a \vee a = a; \quad a \vee \bar{a} = 1$$

$$a \oplus 0 = a; \quad a \oplus 1 = \bar{a}; \quad a \oplus a = 0; \quad a \oplus \bar{a} = 1.$$

| N | c | b | a | qand | q NAND | qor | qxor |
|---|---|---|---|------|--------|-----|------|
| 0 | 0 | 0 | 0 | 0    | 1      | 0   | 0    |
| 1 | 0 | 0 | 1 | 0    | 1      | 1   | 1    |
| 2 | 0 | 1 | 0 | 0    | 1      | 1   | 1    |
| 3 | 0 | 1 | 1 | 0    | 1      | 1   | 0    |
| 4 | 1 | 0 | 0 | 0    | 1      | 1   | 1    |
| 5 | 1 | 0 | 1 | 0    | 1      | 1   | 0    |
| 6 | 1 | 1 | 0 | 0    | 1      | 1   | 0    |
| 7 | 1 | 1 | 1 | 1    | 0      | 1   | 1    |

**Таблица истинности** содержит значения выходных сигналов, соответствующие различным комбинациям (наборам) входных сигналов (аргументов). Элементы имеют три входа ( $a, b, c$ ), для которых существует  $2^3 = 8$  вариантов наборов, поэтому таблица истинности имеет 8 строк. Номера наборов аргументов могут быть дополнительно указаны в десятичной, или в 16-ричной системе счисления (столбец N). Это удобно при анализе таблицы. Для выходных сигналов элементов в таблице истинности соответствует определенный столбец. Таблица составлена в соответствии с известными правилами для выходных

сигналов элементов.

Выход элемента И равен 1, если на все входы поданы сигналы 1. Один из входов элемента И будет активным (изменения сигнала на этом входе будут передаваться на выход), если на остальные входы подан сигнал 1. Функцию И реализуют, соединенные последовательно замыкающие контакты нескольких реле. Цепь в этом случае будет замкнута только тогда, когда сработают все реле.

Выход элемента ИЛИ равен 0, если на все входы поданы сигналы 0. Один из входов элемента ИЛИ будет активным, если на остальные входы подан сигнал 0. Функцию ИЛИ реализуют соединенные параллельно замыкающие контакты нескольких реле. Цепь будет замкнута, если сработает хотя бы одно реле.

Выход элемента «Сумма по модулю 2» равен 1, когда число входов, имеющих состояние 1, нечетно. Любой вход элемента «Сумма по модулю 2» всегда является активным, так как любое изменение одного из входных сигналов (с 0 на 1, или с 1 на 0) всегда изменяет

признак четности выходного сигнала на противоположный.

Свойства логических функций характеризуют также аксиомы, определяющие значения функции, если один из аргументов равен  $a$ , а другой принимает значения 0, 1,  $\bar{a}$ . Из аксиомы, описывающей схему сумма по модулю два можно сделать вывод, что любой вход является активным,

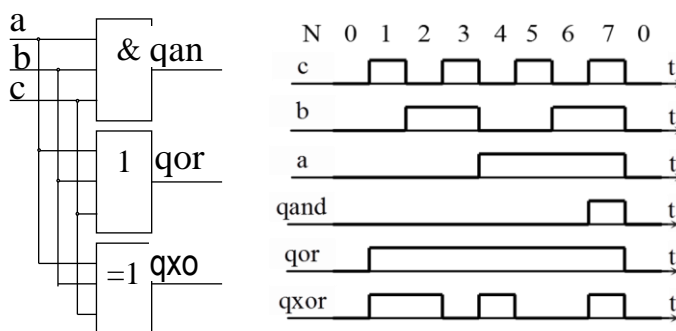


Рис. 1.2. Комбинационная схема, теоретические временные диаграммы

независимо от сигналов на других входах, а на выход можно передать входной сигнал а без изменения, либо с инверсией.

Исследование работы базовых элементов выполняется с использованием комбинационной схемы, содержащей несколько элементов. По таблицам истинности построены теоретические временные диаграммы (рис. 1.2).

В цифровых схемах базовые логические элементы (их называют вентилями - ) выполняют функции стробирования (выделения) сигналов. Вентиль пропускает или не пропускает данные  $d$  в зависимости от управляющего сигнала  $e$ . Сигнал  $e$  (enable - разрешить) называют сигналом разрешения, или стробом.

Многоразрядные схемы стробирования содержат несколько логических элементов, данные  $d$  представлены параллельным кодом, а управление осуществляет одnorазрядный сигнал  $e$ .

В схеме стробирования с элементом И при  $e = 1$  имеем  $q = d$ , при  $e = 0$   $q = 0$ .

В схеме стробирования с элементом ИЛИ при  $e = 1$  имеем  $q = 1$ , при  $e = 0$   $q = d$ .

Схема стробирования с элементом Исключающее ИЛИ – управляемый инвертор.

При  $e = 0$  имеем  $q = d$ , при  $e = 1$   $q = \neg d$  (инверсия  $d$ ).

### **Задание 1.1. Исследование функционирования базовых логических элементов.**

Изучите § 1.1 и Справочный материал, этапы 1 - 9. Разработайте в системе Quartus проект с именем `s11_ks1` по схеме (рис. 1.2). В предлагаемом имени буква  $s$  означает способ ввода исходных данных в виде схемы, второй элемент (цифры 11) – номер задания, затем – буквы и цифры (`ks1`) обозначающие функциональное назначение устройства – комбинационная схема, вариант 1. Для описания на Verilog примем первую букву  $v$ . Предлагаемая система имен файлов, облегчает их поиск и сортировку в каталоге.

При выполнении первой работы вначале создайте рабочий каталог для файлов всех лабораторных работ, например, `D:\users\945\Quartus_LR`. Разработка проекта содержит 10 этапов, смотрите Справочный материал.

**1. Создание нового файла** - *File > New*. Выберите тип Block Diagram File. Запишите созданный файл с именем и расширением `s11_ks1.bdf` в рабочий каталог проекта *File / Save As*.

**2. Создайте новый проект.** Запишите файл в каталог проекта командой *Save As*, при этом флажок *Add file to current project* необходимо установить.

Предупреждения (warnings) не препятствуют продолжению работы.

Другой способ создания проекта *File > New Project Wizard*. В первом окне Introduction –нажмите *Next*. Во втором (Directory, Name, ...) в первой строке, используя кнопку в конце строки, откройте Рабочий каталог проектов, а, а во второй (и в третьей) откройте имя разработанного файла, нажмите *Finish* (рис. 1.3). На предложение создать новый каталог отвечайте «нет». Для работы с ранее разработанным устройством необходимо открыть его проект, а не файл !

**3. Ввод схемы** – это ввод элементов из библиотеки (рис. 1.4) и их соединение (рис. 1.5). Для построения 3-входового элемента сумма по модулю два использовано 2 2-входовых элемента исключающее или.

**4. Компиляция.** Выполняется командой *Processing > Start Compilation* только в том случае, если для файла был создан проект.

**5. Создание символа** *File > Create/Update > Create Symbol Files for Current File*. Окно со схемой, или описанием должно быть активным. Символы создаются для использования в иерархических проектах.

**6. Создание файла для временных диаграмм** с именем проекта. *File > New*, выберите тип файла *Vector Waveform File*, нажмите ОК. Запишите файл *File > Save As*, введите имя проекта и расширение `*.vwf`.

**7. Выбор контрольных точек.** Двойной щелчок левой кнопкой в поле имен узлов и шин сигнального редактора открывает окно *Insert Node or Bus*. Нажмите

кнопку Node Finder (Регистратор узлов). В открывшемся окне (рис. 1. 6) нажмите кнопки List», затем >> и ОК. Будут выбраны все узлы схемы. Впоследствии часть из них легко удаляется.

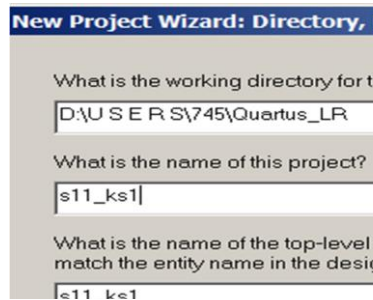


Рис. 1. 3. Создание проекта

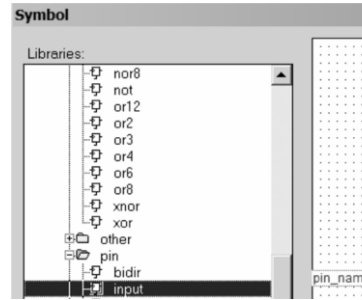


Рис. 1. 4. Ввод элементов

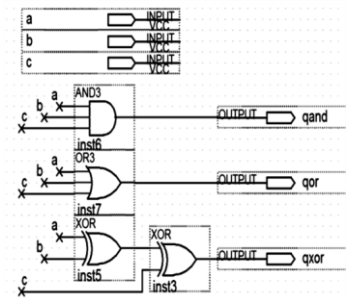


Рис. 1. 5. Результат ввода схемы

**8. Ввод тестовых входных сигналов** выполняется в соответствии с таблицей истинности и теоретическими временными диаграммами (рис. 1.2). Входные сигналы для комбинационных схем подаются от счетчика. Выделите в сигнальном редакторе временную диаграмму для сигнала, период повторения которого выбирается минимальным (например, сигнал с), нажмите кнопку с буквой «С». Первая закладка Counting окна Count Value (рис. 1.7) позволяет задать для счетчика систему счисления Radix, а Вторая (Timing) (рис. 1.8) позволяет задать длительность состояний 0 или 1 выходных импульсов счетчика Count every (установите 50 нс., период составит 100 нс) и множитель для периода Multiplied by (установите 1). Подобным образом создайте диаграммы для сигналов b и a, устанавливая множитель 2 и 4. (рис. 1.12). Введите тестовые сигналы (рис. 1.9).

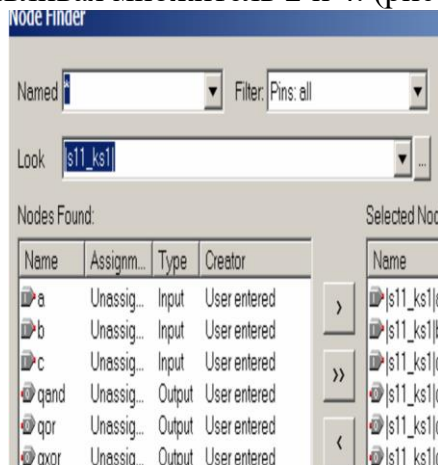


Рис. 1. 6. Выбор контрольных точек

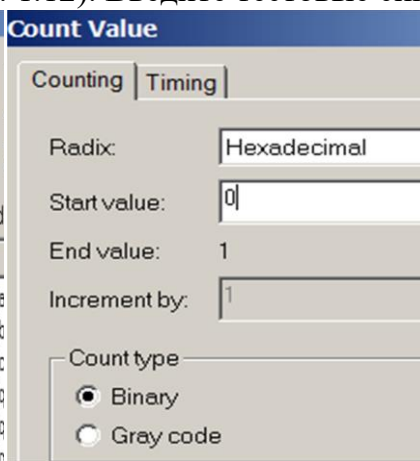


Рис. 1. 7. Настройка режима работы счетчика

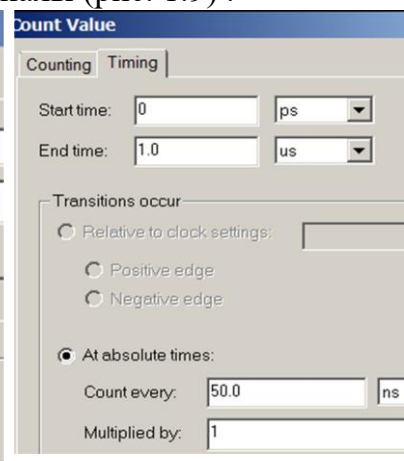


Рис. 1. 8. Настройка параметров счетчика

**9. Моделирование и анализ результатов..** Запуск имитатора выполняет команда Processing > Simulator Tool. Выберите режим моделирования Timing для отображения временных задержек. В строке Simulation input должно быть занесено имя подготовленного файла временных диаграмм, и должна быть включена функция Run simulation until all vector stimuli are used (Выполнять моделирование пока все входные сигналы не будут использованы). Нажми те Start, затем Report. Появится окно Report, в котором будут отображены выходные сигналы (рис. 1.10).

**Анализ результатов** – это описание работы устройства с использованием полученных временных диаграмм, а также сравнение экспериментальных и



теоретических временных диаграмм для определения правильности работы устройства. В отчете приведите результат измерения временных задержек для всех элементов при изменениях входного сигнала из 0 в 1, а также при изменениях из 1 в 0.

в 0. Для измерения временных задержек выберите на временных диаграммах событие для начала интервала измерения (фронт или спад входного сигнала). Включите привязку к сетке кнопкой 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. Штриховых маркеров может быть несколько. Задержка определяется как разность координат маркеров. Для повышения точности измерения необходимо увеличить масштаб (рис. 1.11). В отчете приведите выводы о быстродействии элементов различных типов.

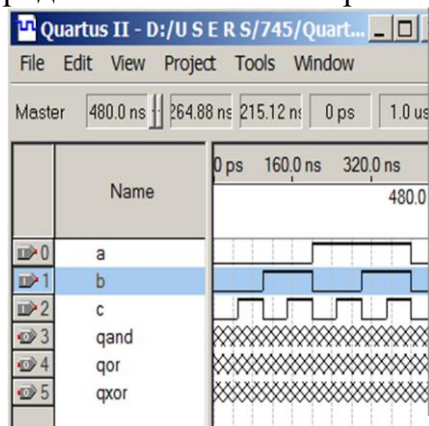


Рис. 1. 9 Исходные диаграммы

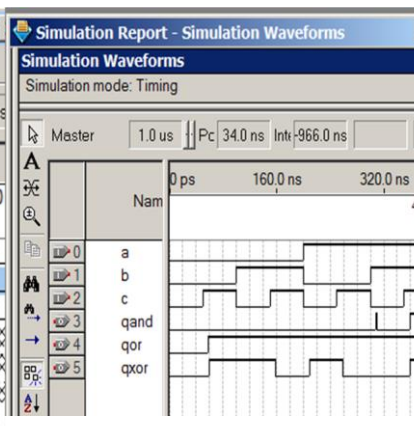


Рис. 1. 10. Результат моделирования Report

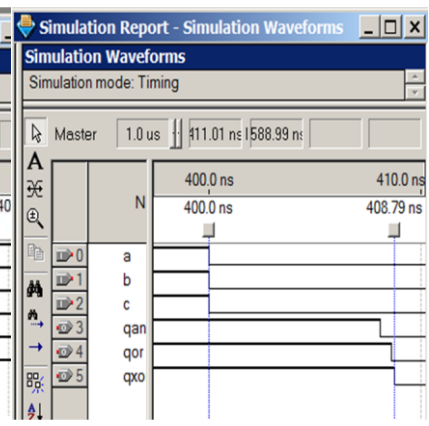
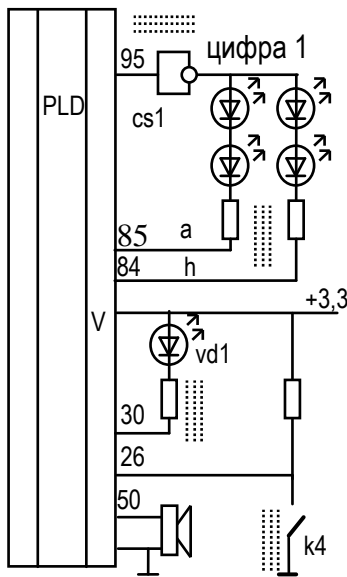


Рис. 1. 11. Измерение временных задержек

**10. Подключение и программирование ПЛИС.** Описание отладочной платы. Верхняя часть платы содержит разъем для подключения источника питания, блок 7-сегментных индикаторов, выключатель питания, транзисторные буферные усилители сигналов на индикатор, выполненные на транзисторах (Q57 – Q50) по схеме с общим эмиттером, резисторами в цепях базы, ограничивающими входной ток (R47 – R40) и резисторами в цепях коллекторов (R57 – R50) для ограничения коллекторных токов. В левой части платы размещены фильтр по питанию (L1, C1), стабилизатор питающего напряжения 3,3 В, кварцевый генератор (U3) на частоту 50 МГц, разъем для подключения программатора с интерфейсом JTAG. В центре платы находится ПЛИС фирмы Altera, структуры CPLD семейства C-M240 типа EPM240T100C5N. К ПЛИС подключены кнопки (K4, K3, K2, K1), акустический излучатель (J2) и светодиоды (D37 – D30) с резисторами (R37 – R30) для ограничения тока. Плата содержит контактные площадки для установки дополнительных разъемов. На плате указаны контакты микросхемы (Pin), к которым подключены элементы. Нумерация контактов ПЛИС выполняется против часовой стрелки от ключа, отмеченного кружком на микросхеме и на плате.

**Схема подключения элементов отладочной платы к ПЛИС** с учетом обозначений приведены на рис. 1.12. Для подачи входных сигналов выбраны три кнопки K4, K3, K2, подключенные к выводам ПЛИС PIN\_26, PIN\_27, PIN\_28 (в соответствии с обозначениями на плате). Для индикации выходных сигналов выбраны три светодиода D37, D36, D35, подключенные к выводам PIN\_58, PIN\_57, PIN\_56.

1. Выполните назначения устройства Assidement> Device. укажите используемую ПЛИС: MAX II, EPM240T100C5 (рис. 1.13).



| Свето-<br>диоды |     | 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

Рис.1.12. Подключение элементов

2. В этом же окне нажмите кнопку Device and Pins option задайте состояние неиспользуемых выводов, закладка Unused Pins, укажите As Input Try Ststes.

3. Подключите выводы.

Assidement>  
Assidement Editor.  
Двойным щелчком выделите сигнал в

#### 4. Выполните компиляцию.

5. Прошивка ПЛИС. Подключите устройство к компьютеру, включите программатор Tools > Programmer > выбрать Program/Configure > Start

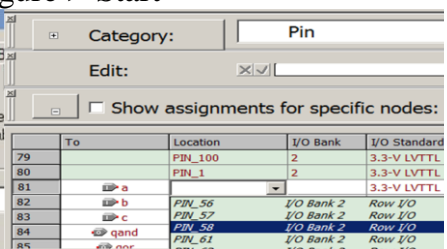
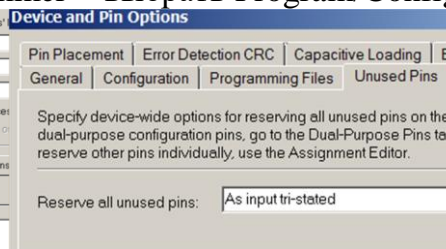
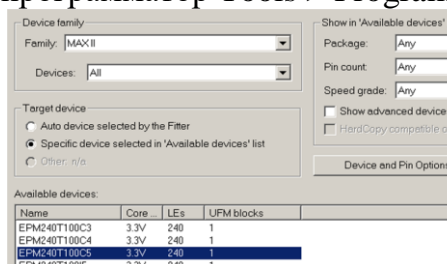


Рис. 1. 13 Назначение

Рис. 1. 14. Назначение

Рис. 1. 15. Назначение

устройства                      неиспользуемых выводов                      выводов  
 Экспериментальная проверка логических элементов с использованием  
 отладочной платы.

Работа схемы И. Проверка правила формирования выходного сигнала. Для подачи  $a=b=c=1$  необходимо кнопки  $k_4, k_3, k_2$  не нажимать. При этом диод D37 не горит, следовательно  $q_{and} = 1$ .

Проверка режима управления передачей сигнала (стробирования). Передача данных от кнопки к4 на диод D37 разрешена, если на остальные входы элемента И подан сигнал 1. Если кнопки к3, к2 не нажаты ( $b=c=1$ ), то при нажатии и отпускании кнопки к4 будет изменяться состояние диода D37.

Работа схемы ИЛИ. кнопки не нажаты, то  $a=b=c=1$  и  $qor=1$ , D36 не горит. Если нажаты все кнопки  $k_4, k_3, k_2$ , то  $a=b=c=0$ . При этом  $qor=0$ , следовательно диод D36 горит.





Рис. 1. 16 Установка аппаратных средств



Рис. 1. 17. Запуск программатора

Проверка режима управления передачей сигнала (стробирования). Передача данных, поданных на вход а разрешена, если на остальные входы подан сигнал 0. Если кнопки к3, к2 нажаты ( $b=c=0$ ), то при нажатии и отпуске кнопки к4 будет изменяться состояние диода D36.

Работа схемы Сумма по модулю 2. Любой вход схемы активен (изменения сигнала передаются на выход) при любых значениях входных сигналов на остальных входах. Изменение состояния любой кнопки вызывает изменение состояния диода D35.

## § 1.2. Синтез комбинационных схем

Комбинационная схема (КС) – это цифровая схема устройства, состоящая только из логических элементов, и не содержащая элементов памяти. Выходные сигналы КС однозначно определяются комбинацией логических входных

сигналов. Они появляются сразу же после подачи входных сигналов с небольшими задержками сигналов в элементах. Значения этих задержек определяет быстродействие элементов.

Синтез КС - это метод проектирования цифровой схемы, обеспечивающий выполнение заданной функции. Рассмотрим основные этапы синтеза КС.

**Этап 1. Изображение символа, определение входных и выходных сигналов, а также функционального назначения.**

Символ КС – прямоугольник, содержащий выводы для подключения входных сигналов (слева), выходных сигналов (слева), а также отдельные поля для обозначения сигналов и функционального назначения схемы.

Таблица 1.2

| № | a | b | c | q | v | w |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 | 0 |
| 7 | 1 | 1 | 1 | 0 | 1 | 1 |

Рис. 1.18. Символ комбинационной схемы ks2

**Этап 2. Составление таблицы истинности** выполняется в соответствии с заданной функцией. Левая часть таблицы отображает входные сигналы, а правая – выходные сигналы. Каждой комбинации входных сигналов соответствуют определенные значения выходных сигналов. Количество строк таблицы истинности равно  $2^n$ , где  $n$  – количество входных сигналов. Двоичные коды, записанные в строках таблицы, называют наборами. Каждой строке соответствует определенный номер. Для комбинационных схем в качестве входных сигналов обычно используют двоичные коды натурального ряда чисел, соответствующие номерам наборов.

**Этап 3. Составление логических функций.** Выполняется по таблице истинности. Логическая функция для каждого выходного сигнала может быть составлена в совершенной дизъюнктивной нормальной форме (СДНФ), или в совершенной конъюнктивной нормальной форме (СКНФ).

**Логическая функция в СДНФ** – это дизъюнкция минтермов (произведений).

**Минтерм** (другое название - конституента единицы) - это произведение всех

входных сигналов, в котором аргументы, равные нулю на данном наборе, записаны с инверсиями. Количество минтермов равно количеству единиц в столбце анализируемого выходного сигнала таблицы истинности. Каждый минтерм соответствует одной строке таблицы истинности, в которой функция равна 1. Запись логической функции в СДНФ используется, если в таблице истинности количество строк, для которых функция равна единице, сравнительно небольшое. Неопределенные значения функции, записанные как x, заменяются нулями.

Для сигнала q (таблица 1.2), столбец которого (после замены значений значений x нулями) содержит всего одну единицу, получим логическую функцию в виде:  $q = \bar{a} \cdot \bar{b} \cdot c$ .

Для сигнала v, столбец которого содержит три единицы, логическая функция содержит три минтерма:

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

**Логическая функция в СКНФ** – конъюнкция всех макстермов (дизъюнкций).

**Макстерм** (другое название - конституента нуля) - это дизъюнкция всех входных сигналов, в которой аргументы, равные единице на данном наборе, записаны с инверсиями. Количество макстермов равно количеству нулей в столбце анализируемого выходного сигнала таблицы истинности. Каждый минтерм соответствует одной строке таблицы истинности, в которой функция равна 0.

Запись логической функции в СКНФ используется, если в таблице истинности количество строк, для которых функция равна нулю, сравнительно небольшое. Неопределенные значения функции, записанные как x, заменяются единицами. Для сигнала w, столбец которого содержит небольшое количество нулей, целесообразно использовать запись логической функции в форме СКНФ:

$$w = (a \vee b \vee c) \cdot (\bar{a} \vee \bar{b} \vee c).$$

**Этап 4. Разработка схемы по логическим функциям.** В графическом редакторе Quartus схема должна содержать три входных (a, b, c) и три выходных (q, v, w) терминала. Для всех входных сигналов необходимы инверторы. Для каждого выходного сигнала используется отдельная КС. В схеме использовано логическое соединение проводников, при котором проводники, имеющие одинаковые имена, считаются соединенными.

**Этап 5. Разработка теоретических временных диаграмм.** Проверку работоспособности устройства выполняют, используя моделирование в САПР, подавая на входы тестовые сигналы, и анализируя выходные сигналы. Для полного тестирования несложных устройств комбинационного типа необходимо выполнить перебор всех возможных комбинаций входных сигналов. Значения выходных сигналов определяется по таблице истинности.

По результатам моделирования можно составить описание работы устройства. Выводы о работоспособности устройства выполняются путем сопоставления теоретических и реальных сигналов и соответствия таблице истинности.

**Задание 1.2. Исследование функционирования ks2.**

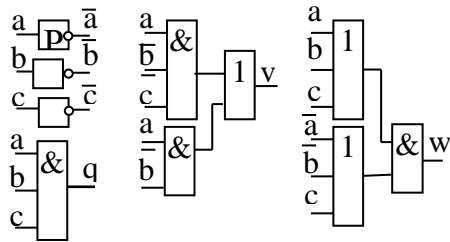


Рис. 1.19. Схема ks2

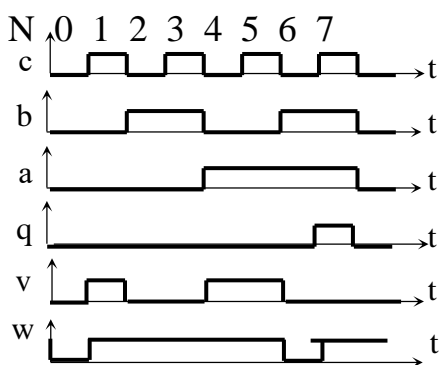


Рис. 1.20. Теоретические временные диаграммы ks2



## Языки описания аппаратных средств (Hardware Description Language - HDL)

1. Предназначены для разработки описания, по которому программируется ПЛИС, а не для разработки вычислительных или моделирующих процедур. Описание на HDL отличается от программы (или кода) на традиционном языке программирования в функциональном назначении.
2. Описывают процессы, происходящие в схеме параллельно во времени.
3. Переменные, с которыми работают HDL-языки, имеют вполне определенный физический смысл, они являются сигналами, которые действуют в цифровой схеме.

**Идентификатор** должен начинаться с буквы или символа подчеркивания, может содержать цифры, знаки подчеркивания и доллара.

**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 проводника из шины (элемента вектора), необходимо указать имя вектора и индекс проводника в квадратных скобках.

Сигналы бывают двух типов: *wire* – провод, или *reg*- регистр.

**Сигнал типа *wire*** моделирует провод, к которому непрерывно прилагается воздействие от логического элемента, который называют драйвер (driver). Входные сигналы имеют тип *wire*. Выходные сигналы комбинационных схем имеют тип *wire*, для их формирования используют параллельные операторы, с ключевым словом *assign* – назначать.

**Сигнал типа *reg*** необходимо назначать для выходных сигналов элементов памяти. Эти сигналы формируют последовательные операторы с ключевым словом *always* - всегда.

**Константы** имеют краткий и полный форматы записи.

**Полный формат записи константы** содержит 4 элемента: 1) число, определяющее количество разрядов в константе; 2) одинарная кавычка; 3) буква, определяющая основание системы счисления (b - двоичная, h - 16-ричная); 4) цифры в указанной системе счисления. Для отрицательных чисел записывают знак «минус» в самом начале записи. В двоичной системе допускаются символы z и x.

**Краткий формат** проще, разрядность числа определяет компилятор. Полный формат используют в описаниях сложных устройств, когда 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.

Обозначения основных функции, принятые в языке Verilog при записи параллельных и последовательных операторов, приведены в таблице.

Арифметические операции могут выполняться над векторами Логические операции выполняются поразрядно для одноразрядных аргументов и векторов.

| Арифметические        |              |
|-----------------------|--------------|
| Умножение             | *            |
| Деление               | /            |
| Сложение              | +            |
| Вычитание             | -            |
| Остаток от деления    | %            |
| Логические            |              |
| <b>И</b>              | <b>&amp;</b> |
| <b>ИЛИ</b>            | <b> </b>     |
| <b>НЕ</b>             | <b>~</b>     |
| <b>Искл. ИЛИ</b>      | <b>^</b>     |
| 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      |

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 и формируют результат (1 бит), равный 1, если условие выполняется. Результат этих операций используется как признак в условных операторах.

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

Состав описания схемы на языке Vtrilog.

**1. Комментарий** может быть началом описания, начинается с двух символов // - знаков деления на клавиатуре, не влияет на компиляцию, содержит русские буквы. Фрагмент в скобках /\* \*/ - также комментарий.

**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 заголовка, и всех последующих строк и операторов – точка с запятой. Имя модуля, или сигнала (идентификатор) должно начинаться с латинской буквы, может содержать также цифры и символы подчеркивания,

пробелы и русские буквы недопустимы Регистр имеет значение! Обычно

используют нижний регистр.

**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, q;` назначает 4-разрядные выходные сигналы одинаковой разрядности типа `wire`.

**4. Параллельные и последовательные операторы.**

**5. Завершение.** Ключевое слово **`endmodule`**

Параллельные операторы описывают комбинационные схемы, начинаются с ключевого слова `assign` – непрерывно назначать.

Последовательные операторы описывают элементы памяти, начинаются с ключевого слова `always`). Выходные сигналы последовательных операторов должны иметь тип `reg`, этот тип необходимо указывать в описании: `output [3..0] q;` `reg [3..0] q;`

Поведенческое. описание выполняется по алгоритму функционирования.

Алгоритм содержащий структуру «Если – то – иначе» описывает оператор

условного присваивания: `assign q = x ? a : b;`. В этой записи `q` – результат, `x` - условие, знак `?` – признак условного оператора, `a` – ветвь ДА, `b` – ветвь НЕТ. Выполняется так: Если `x = 1`, то `q = a`, иначе `q = b`. В качестве `a` и `b` в могут быть использованы одноразрядные сигналы, векторы, или выражения. Условие `x` должно иметь вид одноразрядного сигнала или выражения, результат которого имеет разрядность 1 бит.

Описание комбинационной схемы по логическим функциям.

При проектировании устройств на основе ПЛИС для ввода исходных данных часто используют описания на языке Verilog вместо схем. Для составления описания комбинационной схемы необходимо изобразить ее символ, , отображающий

```
//Описание КС v13_ks2      1
module v13_ks2 (a, b,c,q,v,w); //2
input a, b,c;               //3
output q,v,w;               //4
assign q=a&b&c;              //5
assign v=~a&~b&c | a&~b;     //6
assign w=(a|b|c) & (~a|~b|c); //7
endmodule                   //8
```

входные и выходные сигналы, составить таблицу истинности, записать логические функции.

Описание `v13_ks2` (символ на рисс.

2.1, логические функции в § 1.2).

Начало описания комбинационной схемы, как правило – комментарий, который начинается с двух символов косая черта (знак деления на клавиатуре), допускает русские буквы и не влияет на компиляцию. В

приведенном примере в комментарии указано назначение описания. Номера остальных строк для ссылок записаны как комментарий.

Строка 2 – заголовок, содержащий ключевое слово `module`, за ним следует имя устройства и перечисление всех входных и выходных сигналов в круглых скобках, через запятую. При перечислении сигналов используют обозначения выводов, изображенные на схеме или символе устройства.

Имя устройства должно начинаться с буквы и содержать только латинские буквы, цифры и знак подчеркивания. В имени не допускаются русские буквы и пробелы. Примем систему выбора имен файлов для упрощения поиска в общем каталоге. Для рассматриваемого описания выбрано имя `v13_ks2`, означающее

описание на Verilog, задание 1.3, комбинационная схема 2. Такое же имя должен иметь файл, содержащий это описание.

Строка 3 содержит описание входных сигналов, после ключевого слова `input` перечислены входные сигналы `a`, `b`, `c`. Никаких указаний о типах и разрядности сигналов нет. Эти сигналы по умолчанию будут назначены как одноразрядные, типа `wire`.

Строка 4 содержит описание выходных сигналов, после ключевого слова `output` указаны выходные сигналы `q`, `v`, `w`, которые по умолчанию будут назначены как одноразрядные, типа `wire`.

Строки 5-7 содержат параллельные операторы присваивания для каждого выходного сигнала, которые начинаются с ключевого словом `assign` – непрерывно назначать, записанные по логическим функциям из § 1.2:

$$q = a \cdot b \cdot c; \quad v = \bar{a} \cdot \bar{b} \cdot c \vee a \cdot \bar{b}; \quad w = (a \vee b \vee c) \cdot (\bar{a} \vee \bar{b} \vee c).$$

Правило описания логических функций. Записывается ключевое слово параллельного оператора `assign`, а затем - логическая функция, в которой 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 «^».

Завершением описания является ключевое слово `endmodule`, после которого не должно быть никаких знаков препинания.

### **Задание 1.3. Проект комбинационной схемы по описанию на Verilog.**

Изучите § 1.2, § 1.3, Справочный материал. Этапы выполнения задания.

1. Создание нового файла - *File > New*. Для ввода описания выберите тип файла *Verilog HDL File*, нажмите *OK*. Укажите имя файла, как записано в заголовке описания, `v13_ks2.v`. Расширение (`.v`) определяет язык Verilog.

2. Создание нового проекта – запись файла в каталог проекта командой *Save As*. Флажок *Add file to current project* необходимо установить. Предупреждения (warnings) не препятствуют продолжению работы. Будет создано несколько служебных файлов с тем же именем `v13_ks2` и различными расширениями, необходимых для дальнейшей работы с проектом.

Другой способ создания проекта *File > New Project Wizard*. В первом окне *Introduction* – нажмите *Next*. Во втором (*Directory, Name, ...*) в первой строке, используя кнопку в конце строки, откройте Рабочий каталог проектов, а, а во второй (и в третьей) откройте имя разработанного файла, нажмите *Finish*. На предложение создать новый каталог отвечайте «нет». Для работы с ранее разработанным устройством откройте его проект.

3. Ввод исходных данных - текста описания пояснений не требует.

4. Компиляция - *Processing > Start Compilation* выполняется, если для файла был создан проект.

5. Создание символа - *File > Create/Update > Create Symbol Files for Current File*, *OK*. Окно с описанием должно быть активным. Символы создаются для использования в иерархических проектах.

Выполнение остальных этапов описано в заданиях 1.1 и 1.2. В отчете приведите экспериментальные временные диаграммы, сравните их с теоретическими (рис. 1.20).



## § 1.4. Синтез логического устройства комбинационного типа для поразрядного выполнения логических операций над параллельными кодами.

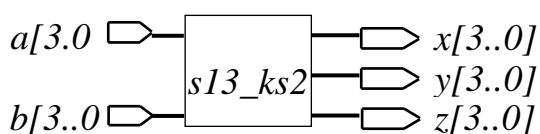


Рис. 1.23 Символ схемы *s14\_ks3*

Рассмотрим пример проектирования комбинационной схемы для поразрядного выполнения логических операций (рис. 1.22), которая имеет 4-разрядные входные сигналы *a* и *b* и 4-разрядные выходные сигналы *x*, *y*, *z*. Заданы логические

функции :  $x = \overline{a \cdot b}$ ;  $y = a \vee \overline{b}$ ;  $z = a \oplus b$ .

Для 4-разрядных сигналов заданные формулы имеют вид:

$$\begin{aligned} x_3 &= \overline{a_3 \cdot b_3}; & x_2 &= \overline{a_2 \cdot b_2}; & x_1 &= \overline{a_1 \cdot b_1}; & x_0 &= \overline{a_0 \cdot b_0}; \\ y_3 &= a_3 \vee \overline{b_3}; & y_2 &= a_2 \vee \overline{b_2}; & y_1 &= a_1 \vee \overline{b_1}; & y_0 &= a_0 \vee \overline{b_0}; \\ z_3 &= a_3 \oplus b_3; & z_2 &= a_2 \oplus b_2; & z_1 &= a_1 \oplus b_1; & z_0 &= a_0 \oplus b_0; \end{aligned}$$

По формулам составлена схема (рис. 1.24.).

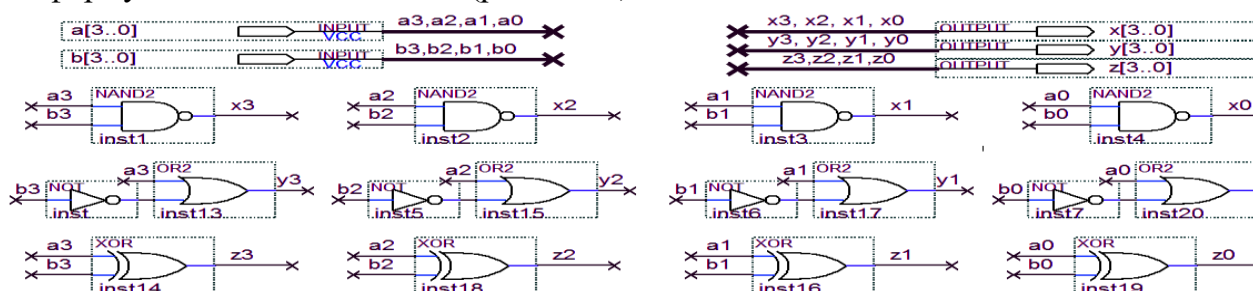


Рис. 1.24. Схема модуля *s13\_ks2*

Схема имеет два 4-разрядных входных терминала и три 4-разрядных выходных терминала. К выводам терминалов подключены толстые линии, отображающие шины. Над обозначениями шин перечислены обозначения проводников, подключенных к данной шине, которые используются в схеме.

Теоретические временные диаграммы (рис. 1.24) предназначены для тестирования разработанного устройства. Для упрощения определения выходных

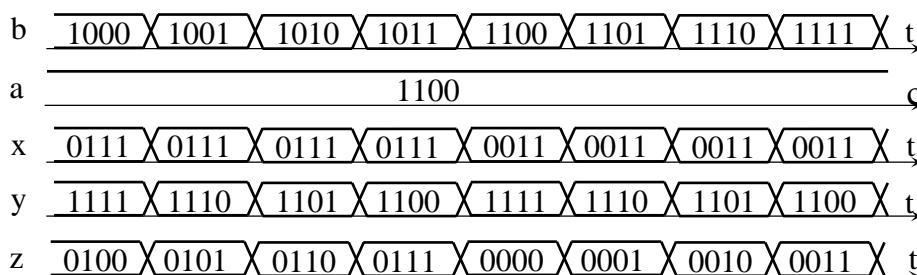


Рис. 1.25 Теоретические временные диаграммы для комбинационной схемы *s14\_ks3*

сигналов на один из входов подан постоянный сигнал (маска). При определении кодов выходных сигналов необходимо использовать правила выполнения для заданных логических

операций (см. § 1).

**Задание 1.4. Проект комбинационной схемы *s14\_ks3* по заданным логическим функциям. Вариант определяется по номеру бригады.**

|         |   |   |   |   |   |   |   |   |   |    |    |    |
|---------|---|---|---|---|---|---|---|---|---|----|----|----|
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---------|---|---|---|---|---|---|---|---|---|----|----|----|

|       |                   |                    |                         |                          |                   |                    |                         |                  |                  |                    |                    |                         |
|-------|-------------------|--------------------|-------------------------|--------------------------|-------------------|--------------------|-------------------------|------------------|------------------|--------------------|--------------------|-------------------------|
| $x =$ | $\bar{a} \cdot b$ | $a \cdot \bar{b}$  | $\bar{a} \cdot \bar{b}$ | $a \cdot b$              | $\bar{a} \cdot b$ | $a \cdot \bar{b}$  | $\bar{a} \cdot \bar{b}$ | $a \cdot b$      | $a \cdot b$      | $\bar{a} \cdot b$  | $a \cdot \bar{b}$  | $\bar{a} \cdot \bar{b}$ |
| $y =$ | $a \vee \bar{b}$  | $\bar{a} \vee b$   | $a \vee b$              | $\bar{a} \vee \bar{b}$   | $a \vee \bar{b}$  | $\bar{a} \vee b$   | $a \vee b$              | $\bar{a} \vee b$ | $\bar{a} \vee b$ | $a \vee \bar{b}$   | $\bar{a} \vee b$   | $a \vee \bar{b}$        |
| $z =$ | $a \oplus b$      | $\bar{a} \oplus b$ | $a \oplus \bar{b}$      | $\bar{a} \oplus \bar{b}$ | $a \oplus b$      | $\bar{a} \oplus b$ | $a \oplus b$            | $a \oplus b$     | $a \oplus b$     | $\bar{a} \oplus b$ | $\bar{a} \oplus b$ | $a \oplus b$            |

Разработайте проект КС для выполнения заданных в таблице логических операций. Выполните этапы 1 – 9 (См. Справочный материал).

В отчете приведите теоретические и экспериментальные временные диаграммы.

### § 1.5. Разработка иерархического проекта комбинационной схемы.

**Иерархический проект** - это сложный проект, составленный из простых модулей, представленных в САПР в виде символов. Подобным образом сложные программы строятся из подпрограмм. Модули иерархического проекта могут быть любой степени сложности, ввод данных при их разработке может быть выполнен по схеме или по описанию. Иерархические проекты будут использоваться в лабораторных работах для одновременного моделирования устройств одинакового функционального назначения, построенных с использованием различных способов ввода исходных данных с целью сравнительного анализа параметров и характеристик.

Для построения схемы проекта верхнего уровня иерархии в графическом редакторе необходимо выполнить следующие действия.

Изобразить символы модулей одинакового функционального назначения, подлежащих сравнению. Заблаговременно необходимо выполнить разработку проектов и создание символов этих модулей.

При этом одноименные входы модулей соединяются параллельно, а выходные сигналы, обозначения которых должны различаться, подлежат сравнению (Рис. 1.19).

**Задание 1.5. Иерархический проект s15\_ks5 по заданной таблице истинности.**

| Вариант 8 | Вариант 9 | Вариант 10 | Вариант 11 | Вариант 12 | Вариант 13 | Вариант 14 |
|-----------|-----------|------------|------------|------------|------------|------------|
| N a b c q | N a b c q | N a b c q  | N a b c q  | N a b c q  | N a b c q  | N a b c q  |
| 0 000 0   | 0 000 0   | 0 000 0    | 0 000 0    | 0 000 0    | 0 000 0    | 0 000 0    |
| 1 000 1   | 1 001 1   | 1 001 1    | 1 001 1    | 1 001 1    | 1 001 0    | 1 001 1    |
| 2 010 0   | 2 010 0   | 2 010 0    | 2 010 0    | 2 010 0    | 2 010 1    | 2 010 0    |
| 3 011 1   | 3 011 1   | 3 011 1    | 3 011 1    | 3 011 1    | 3 011 1    | 3 011 0    |
| 4 100 0   | 4 100 0   | 4 100 0    | 4 100 0    | 4 100 0    | 4 100 0    | 4 100 0    |
| 5 101 1   | 5 101 1   | 5 101 0    | 5 101 1    | 5 101 1    | 5 101 1    | 5 101 1    |
| 6 110 0   | 6 110 0   | 6 110 1    | 6 110 0    | 6 110 0    | 6 110 0    | 6 110 0    |
| 7 111 0   | 7 111 1   | 7 111 0    | 7 111 0    | 7 111 1    | 7 111 0    | 7 111 1    |

| Вариант 1 | Вариант 2 | Вариант 3 | Вариант 4 | Вариант 5 | Вариант 6 | Вариант 7 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| N a b c q | N a b c q | N a b c q | N a b c q | N a b c q | N a b c q | N a b c q |
| 0 000 1   | 0 000 0   | 0 000 1   | 0 000 1   | 0 000 1   | 0 000 1   | 0 000 0   |
| 1 000 0   | 1 001 0   | 1 001 0   | 1 001 1   | 1 001 0   | 1 001 0   | 1 001 1   |
| 2 010 0   | 2 010 0   | 2 010 0   | 2 010 0   | 2 010 0   | 2 010 1   | 2 010 0   |
| 3 011 1   | 3 011 1   | 3 011 1   | 3 011 1   | 3 011 1   | 3 011 1   | 3 011 1   |
| 4 100 0   | 4 100 0   | 4 100 0   | 4 100 0   | 4 100 0   | 4 100 0   | 4 100 0   |
| 5 101 1   | 5 101 1   | 5 101 0   | 5 101 1   | 5 101 1   | 5 101 1   | 5 101 1   |
| 6 110 0   | 6 110 0   | 6 110 1   | 6 110 0   | 6 110 0   | 6 110 0   | 6 110 0   |
| 7 111 0   | 7 111 1   | 7 111 0   | 7 111 0   | 7 111 1   | 7 111 0   | 7 111 1   |

Изучите § 1.2, § 1.5, выполните все этапы синтеза КС, разработайте проект в соответствии с заданной таблицей истинности, вариант определяется по номеру бригады. Порядок разработки иерархического проекта комбинационной схемы ks5 по заданной таблице истинности, которая имеет три входа (a, b, c) и один выход (q).

1. Выполните синтез КС по заданной таблице истинности по методике, описанной в § 1.2, § 1.3, составьте логическую функцию, схему, описание на языке Verilog.

2. Разработайте модуль КС нижнего уровня по схеме. Создайте новый файл типа *Block Diagram / Schematic File*, введите схему, сохраните файл со схемой с именем s15\_ks5.bdf. Выполните этапы 1 - 5, создайте новый проект, выполните компиляцию, и создание символ. Этапы 6 – 9 будут выполняться впоследствии при работе с верхним уровнем проекта.

3. Разработайте модуль КС нижнего уровня по описанию на Verilog. Создайте новый файл типа *Verilog HDL File*. Выполните этапы 1 - 5, введите описание, сохраните файл с описанием с именем v15\_ks5.v, создайте новый проект, выполните компиляцию, и создание символ.

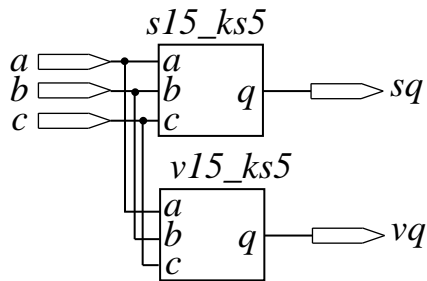


Рис. 1.1.26. Схема sv15\_ks5

4. Разработайте в графическом редакторе проект верхнего уровня иерархии в виде схемы с именем sv15\_ks5.bdf (буквы sv в обозначении указывают на наличие в проекте модулей в виде схемы и в виде описания на Verilog). Для этого необходимо создать графический файл и использовать символы, созданные в предыдущих заданиях.

Введите схему в соответствии с рис. 1.25. Для ввода символов модулей в окне Symbol, которое открывается двойным щелчком, откройте библиотеку Project. Одноименные входы модулей необходимо соединить параллельно для подачи на модули одинаковых сигналов.. Для одноименных выходов модулей выбираются различные имена терминалов, которые для модуля со схемой имеют первую букву s, а для модуля с описанием - первую букву v. Это необходимо для сравнения сигналов, которые при функциональном моделировании должны совпадать, а при временном моделировании могут различаться. Двойной щелчок по символу s15\_ks5 открывает схему, а по символу v15\_ks5 – описание. Выполните этапы 2 - 9, описанные в задании 1.1.

В отчете приведите теоретические и экспериментальные временные диаграммы и выводы по результатам моделирования.

### Контрольные вопросы

1. Для заданного логического элемента (И, ИЛИ, И-НЕ ИЛИ, Исключающее ИЛИ) - приведите символ, таблицу истинности, логическую функцию, теоретические временные диаграммы;
2. Что такое режим стробирования, как он используется?
3. Поясните на примере составление логических функций в СДНФ и в СКНФ.
4. Поясните на примере синтез КС.
5. Изобразите схемы подключения кнопок и светодиодов к ПЛИС на отладочной плате.
6. поясните работу данных элементов.
7. Какие значения может принимать сигнал в языке Verilog?
8. Типы операторов и сигналов в языке Verilog, поясните на примерах.

9.2. Правила записи констант в языке Verilog, поясните на примерах.