0819

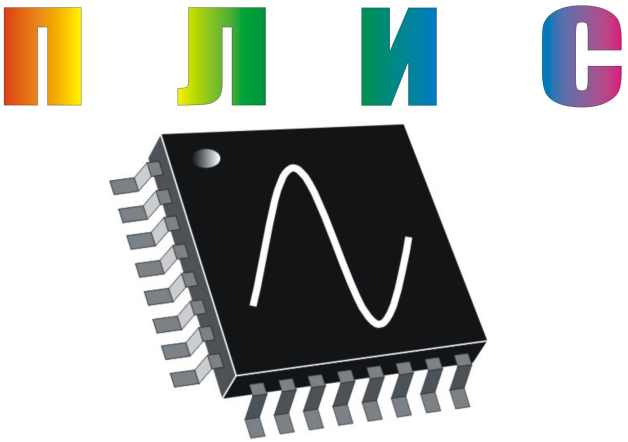
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Ф. УТКИНА

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПЛИС

Методические указания к практическим занятиям

Часть I



Рязань 2020

УДК 621.372 (021)

Современные технологии ПЛИС: методические указания к практическим занятиям. Ч. I / Рязан. гос. радиотехн. ун-т им. В. Ф. Уткина; сост.: А. Ю. Линович. — Рязань, 2020. — 44 с.

Содержат методические указания к практическим занятиям по дисциплине «Современные технологии ПЛИС». Описание каждого практического занятия содержит краткие теоретические сведения, рекомендации по подготовке и выполнению заданий.

Предназначены студентам дневного отделения, обучающимся по направлению 11.04.02 «Инфокоммуникационные технологии и системы связи».

Табл. 8. Ил. 22. Библиогр.: 6 назв.

Программируемая логическая интегральная схема, анализ, синтез, карта Карно, минтерм, макстерм

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета им. В. Ф. Уткина.

Рецензент: кафедра ТОР Рязанского государственного радиотехнического университета им. В. Ф. Уткина (зав. кафедрой профессор В. В. Витязев)

Современные технологии ПЛИС
Часть I

Составитель Л и н о в и ч Александр Юрьевич

Редактор М. Е. Цветкова
Корректор С. В. Макушина
Подписано в печать 05.02.20. Формат бумаги 60x84 1/16.
Бумага газетная. Печать трафаретная. Усл. печ. л. 2,75.
Тираж 50 экз. Заказ
Рязанский государственный радиотехнический университет
им. В. Ф. Уткина.
390005, Рязань, ул. Гагарина, 59/1.
Редакционно-издательский центр РГРТУ.

Практическое занятие № 1

Общие правила языка «Verilog»

Цель работы

Изучить синтаксис языка «Verilog». Научиться и описывать простейшие комбинационные схемы на языке «Verilog».

Теоретическая часть

Языки описания аппаратных средств (Hardware Description Language — HDL) позволяют описать проектируемое устройство, а затем, используя систему автоматизированного проектирования (САПР), запрограммировать ПЛИС и получить аппаратную реализацию этого проекта. Вместо того чтобы разрабатывать схему устройства, и вводить её в графическом редакторе системы проектирования, достаточно ввести ее текстовое описание. Описание проектов на HDL-языках — эффективный способ подготовки исходных данных при проектировании устройств на элементной базе ПЛИС. Языки описания аппаратуры позволяют составить описание цифровой схемы, которое может быть использовано на всех этапах разработки цифровых устройств, при проектировании, моделировании и тестировании аппаратуры.

В современных системах автоматизированного проектирования устройств на ПЛИС в качестве средств описания проектов применяются специализированные и универсальные языки описания аппаратных средств. Специализированные языки позволяют проектировать устройства в ПЛИС определенной фирмы, примером является язык AHDL — Altera HDL, предназначенный для работы с ПЛИС фирмы Altera. Универсальные языки позволяют разрабатывать проекты для ПЛИС различных фирм, это VHDL (Very high-speed 1C Hardware Description Language — Язык для описания быстродействующих аппаратных средств), и Verilog HDL.

Существует несколько отличий языков описания аппаратных средств (hardware description language) от языков программирования.

1) Первое отличие состоит в том, что языки программирования позволяют описать последовательность вычислительных операций для микропроцессора, а HDL-языки предназначены для составления описания взаимосвязей между логическими элементами. Можно получить в ПЛИС аппаратный процессор.

2) Вторая характерная особенность HDL-языков заключается в том, что они описывают преобразования, которые выполняются в логической схеме параллельно. Сигналы в разных точках цифровой схемы могут изменяться одновременно, и эта особенность отображается средствами языка.

Следует отметить, что переменные, с которыми работают HDL-языки, имеют вполне определенный физический смысл, они являются сигналами, которые действуют в цифровой схеме.

Алфавит языка — набор символов, допустимый в описаниях. Этот набор содержит латинские буквы, цифры и специальные символы. Регистр имеет значение! Для ввода описаний обычно устанавливают нижний регистр, и впоследствии его не переключают. Русские буквы можно использовать только в комментариях, которые не влияют на процесс компиляции.

Имя модуля, или сигнала (другое название – идентификатор) должно начинаться с буквы или символа подчеркивания; оно может содержать цифры, знаки подчеркивания и доллара. Имя сигнала выбирает пользователь, в простых схемах обычно используют короткие имена, а в сложных – сочетания букв, определяющие назначение сигнала. В имени сигнала недопустимы пробелы. Если требуется в имени разделить группы букв, имеющие определенные смысловые значения, то используют знак подчеркивания.

Пример имен сигналов: «a, b, c, d, q», пример имени модуля: «log_schema_var1».

Комментарий – строка с пояснениями, начинающаяся с двух символов косая черта «//». Комментарии не влияют на результат компиляции описания. В комментариях можно использовать русские буквы и любые символы. Фрагмент в скобках /* */ - также комментарий. Такие скобки можно использовать при отладке для временного исключения фрагментов из компиляции описания.

Константы в языке Verilog можно записать в различных системах счисления. Полная запись константы содержит четыре элемента. Первый из них — число, определяющее количество разрядов в константе. Второй элемент — одинарная кавычка. Третий элемент – буква, определяющая основание системы счисления: b — для двоичной системы, o — для восьмеричной, d — для десятичной, h для 16-ричной. Четвертым элементом записи являются цифры в указанной системе счисления. Отрицательные числа задаются знаком «минус» перед разрядностью числа в самом начале записи. В двоичной системе допускаются символы z и x (их назначение поясняется далее). Для записи констант

в описаниях сложных устройств отдают предпочтение двоичной системе, которая наглядно отображает разрядность числа.

Краткая запись константы содержит только число. Это соответствует по умолчанию десятичной системе счисления. Запись констант в десятичной системе проще воспринимается и используется при описании простых схем.

Например, запись «1'b1» соответствует единице в двоичной системе, а запись «1» — единице, записанной в десятичной системе. Первая запись, в которой явно указана разрядность, более строгая, ее применяют в сложных описаниях. Рассмотрим запись константы 5. Запись в двоичной системе имеет вид: «3'b101», а в десятичной «5» (3 — колво разрядов, b – основание системы счисления).

Сигналы

Переменные языка Verilog называются сигналами. Они имеют вполне определенный физический смысл, они являются сигналами.

Сигналы могут принимать одно из четырех значений: 0, 1, z, x. Значение «z» формируют элементы с темя состояниями выхода, когда источник сигнала отключен, а значение сигнала «x» означает, что сигнал не определен и может принимать любое значение 0 или 1. Чаще всего значения сигнала — это 0 и 1.

1) Сигналы в схеме бывают двух типов — цепь (wire провод) и регистр (reg- регистр). Сигнал типа цепь - wire - моделирует провод, к которому непрерывно прилагается воздействие от источника сигнала, который называют драйвер (driver). По умолчанию все сигналы в описаниях устанавливаются типа «wire». Выходные сигналы комбинационных схем имеют тип «wire».

Сигнал типа регистр моделирует элемент памяти (триггер или регистр). Тип «reg» необходимо указать для выходных сигналов схем с элементами памяти.

2) Сигналы имеют различную разрядность. Если в описании разрядность не указана, то по умолчанию принимается сигнал в 1 бит.

Так, например, в описаниях полусумматора и одноразрядного сумматора (глава 5) тип и разрядность сигналов не указаны, следовательно, сигналы будут иметь тип «wire» и разрядность 1 бит.

Код, состоящий из нескольких разрядов, который параллельно передается через группу проводников, описывают как вектор или шину.

В описании вектора в квадратных скобках через двоеточие указывают диапазон индексов, нумерующих разряды, или проводники шины. Первый индекс должен соответствовать старшему разряду вектора.

При такой нумерации вес разряда и индекс будут совпадать. Например, описание 4-разрядной шины с именем «d» имеет вид: [3:0] d.

Какой-либо из проводников шины называют также «элемент вектора». Для использования в описании одного проводника шины (элемента вектора), необходимо указать имя вектора и индекс проводника в квадратных скобках. Так, например, запись d[3] означает сигнал третьего проводника шины «d».

По умолчанию вектор — беззнаковое скалярное (scalared) целое, в котором разрешен доступ к отдельным битам. Если перед указанием диапазона записать «signed» — то будем иметь число в дополнительном коде со знаком.

Операторы

В языке Verilog существуют операторы — с одним, двумя и тремя операндами. Окончание всех операторов обозначает разделитель — точка с запятой. Символы пробела, табуляции, возврата каретки транслятор игнорирует, если они не нарушают целостность ключевых слов.

1) Арифметические операторы имеют два аргумента (их называют бинарными), они перечислены в таблице 1.1 в порядке убывания приоритета. Порядок операций можно изменить, используя круглые скобки.

2) В языке Verilog существует различные типы логических операторов.

Таблица 1.1

Арифметические		Поразрядные и свертки		Отношения и сравнения	
Умножение	*	И	&	Равно?	==
Деление	/	ИДИ	|	Не равно?	!=
Сложение	+	НЕ	~	Больше ?	>
Вычитание	-	Искл. ИЛИ	^	Меньше ?	<
Остаток от деления	%	И-НЕ	~&	Больше или равно ?	>=
		ИЛИ-НЕ	~|	Меньше или равно ?	<=
Объединение	{a,b,c}	Искл.ИЛИ-НЕ	~^	Идентично?	===

2.1) Поразрядные операторы содержат два операнда. Входные и выходной сигналы могут быть векторами. Векторы могут иметь различную разрядность, если разрядность одного операнда меньше, чем другого, то недостающие разряды заполняются нулями.

2.2) Операторы свёртки, содержат один операнд, используют те же символы, что и поразрядные операторы (кроме оператора инверсии «~»). Они выполняются над единственным, многоразрядным операндом побитно, шаг за шагом, формируя на выходе одноразрядный результат. Этот тип операторов позволяет формировать признаки данных. Так, например, если d – многоразрядный вектор, то признак p1 = &d будет иметь один бит, равный единице, если все биты входного вектора равны единице, признак p2 = |d будет равен нулю, если все биты вектора равны нулю, а p3 = ^d будет признаком для контроля на четность. Примеры формирования признаков данных содержатся в описаниях арифметических устройств.

2.3) Операторы отношения и сравнения сравнивают два операнда и выдают значение в виде однобитовой логической переменной, которая обычно используется в условных операторах. При выполнении условия выдается «1». Если хотя бы один операнд не определен, то и результат примет неопределенное значение «x». В случае неодинаковой разрядности операндов, более короткий дополняется слева нулями. Использование операторов сравнения показано в описаниях компараторов кодов.

3) Оператор объединения, позволяет из исходных векторов сформировать вектор суммарной разрядности. Исходные векторы записываются в фигурных скобках через запятую. Объединенный вектор может быть записан справа и слева от знака равенства в операторе присваивания.

4) Оператор условного присваивания содержит три операнда (поэтому его называют тернарным), выполняет присваивание выходному сигналу одного из двух значений в зависимости от условия. Запись оператора условного присваивания содержит следующие элементы:

- ключевое слово «assign»;
- имя выходного сигнала;
- знак равенства;
- сигнал, или выражение, определяющие условие;
- знак вопроса;
- ветвь «да»;
- двоеточие;
- ветвь «нет»;
- знак «точка с запятой».

Например, запись assign q = s ? a : b; читается так: «если s =1 то q = a , иначе q = b. В качестве условия «s» в данном операторе может использоваться выражение, формирующее логический сигнал «1» или

«0», а вместо переменных «a» и «b» могут быть операторы, в том числе и другой условный оператор, заключенные в скобки.

Описание логической схемы на языке Verilog

Описание устройства содержит следующие основные элементы.

1. Комментарий. Это пояснение, начинающееся с двух символов «косая черта» (знак деления на клавиатуре), содержащее русские буквы. Комментарий на компиляцию не влияет и может отсутствовать.
2. Заголовок. Вначале это ключевое слово «module», затем через пробел имя модуля и перечисление всех входных и выходных сигналов, заключенное в круглые скобки. Завершение заголовка, и всех последующих строк — разделитель — точка с запятой.
3. Описание входных и выходных сигналов, или портов. После ключевого слова «input» перечисляются входные сигналы, а после ключевого слова «output» - выходные сигналы.
4. Ключевое слово «assign» (или «always») и операторы, описывающие работу устройства. Строк указанных типов может быть несколько.
5. Завершение. Ключевое слово «endmodule», после которого нет никаких знаков препинания.

```
// Пример 1. Комбинационная          //1
// схема ks1                         //2
module p41_ks1 (d,qand,qor,qxor);    //3
input [2:0] d;                       //4
output qand, qor, qxor ;             //5
assign qand=d[2] & d[1] & d[0] ;     //6
assign qor = d[2]  |  d[1]  |  d[0];  //7
assign qxor=d[2]  ^ d[1]  ^ d[0];    //8
endmodule                            //9
```

Порядок выполнения работы

1. Получить задание разработать комбинационную схему. Задание выдаёт преподаватель. Примеры заданий могут быть заданы в форме логических выражений, например:

   $Q = \overline{(A \vee B)} \oplus (C \cdot D)$, $Q = (A \cdot B) \vee \overline{(C \oplus D)}$, $Q = \overline{(A \cdot B) \vee \overline{(C \cdot D)}}$,

   $Q = (A \vee B) \cdot \overline{(C \cdot D)}$, $Q = \overline{(A \cdot B)} \oplus (C \vee D)$, $Q = (A \vee B) \cdot \overline{(C \oplus D)}$,

   $Q = \overline{(A \oplus B) \cdot \overline{(C \vee D)}}$.

2. Разработать текст описания на языке «Verilog».

Практическое занятие № 2

Анализ комбинационных схем

Цель работы

Научиться выполнять анализ комбинационных схем, заданных логическими функциями или логическими уравнениями.

Теоретическая часть

Цифровые логические схемы, не содержащие элементов памяти, называют комбинационными. Выходные сигналы комбинационных схем однозначно определяются комбинацией логических сигналов на их входах в данный момент времени и не зависят от сигналов, которые подавались ранее. Комбинационные схемы используются для построения арифметических и логических устройств, преобразования кодов, переключения цепей передачи данных, формирования логических сигналов и признаков данных. Другой класс схем — это последовательностные схемы, которые содержат элементы памяти. Сигналы на выходе последовательностных схем зависят не только от комбинации входных сигналов в данный момент времени, но и от последовательности сигналов в предшествующие моменты времени.

Анализ логических устройств комбинационного типа неразрывно связан с их применением и предполагает решение для заданной схемы определенных задач. Это изучение работы устройства, определение логических операций, которые производит каждый логический элемент в отдельности, и определение функции, которую выполняет схема в целом, составление таблицы истинности и логических уравнений, разработка сигналов для тестирования устройства, построение теоретических временных диаграмм.

Исходные данные для анализа комбинационной схемы могут быть представлены в одной из форм: в виде принципиальной схемы устройства, таблицы истинности или логических функций.

Результатом анализа является описание функционирования комбинационной схемы, в состав которого входят:

1) таблица истинности;
2) логические функции;
3) схема;
4) временные диаграммы;
5) описания устройства на языке Verilog.

Анализ комбинационной схемы, заданной логическими функциями

Рассмотрим примеры анализа комбинационных схем.

Пусть требуется выполнить анализ комбинационной схемы, предназначенной для моделирования базовых логических элементов И, ИЛИ, сумма по модулю два. Комбинационную схему описывают функции:

$$qand = d2 \cdot d1 \cdot d0;$$

$$qor = d2 \vee d1 \vee d0;$$

$$qxor = d2 \oplus d1 \oplus d0.$$

1) По заданным уравнениям составлена комбинационная схема ks1 (рис. 2.1), содержащая базовые логические элементы, на входы которых подаются одинаковые сигналы.

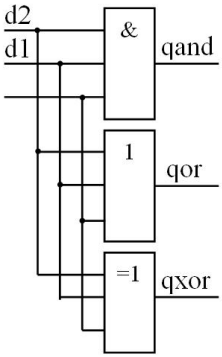


Рис. 2.1. Комбинационная схема ks1

2) По уравнениям составлена таблицы истинности (табл. 2.1). Таблица истинности содержит значения выходных сигналах при различных значениях входных сигналов. Для заданной комбинационной схемы таблица истинности составлена с учетом логики работы элементов. Схема имеет три входа (d2, d1, d0). Для трех переменных существует 23 = 8 вариантов комбинаций входных сигналов, поэтому таблица истинности имеет 8 строк. Комбинации сигналов называют наборами. Номера наборов входных сигналов записаны также в 16-ричной системе счисления в столбце N. Такая нумерация удобна при моделировании.

Комбинационная схема имеет три выхода, каждому из них в таблице истинности соответствует определенный столбец.

Таблица 2.1

N	d2	d1	d0	qand	qor	qxor
0	0	0	0	0	0	0
1	0	0	1	0	1	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	0	1	1
5	1	0	1	0	1	0
6	1	1	0	0	1	0
7	1	1	1	1	1	1

3) Следующая задача анализа – составление временных диаграмм. Важным моментом при составлении временных диаграмм является выбор входных тестовых сигналов, позволяющих выполнить полную проверку правильности функционирования устройства. Этот выбор предполагает получение предсказуемого результата, основанного на знании работы устройства.

При моделировании комбинационной схемы необходимо получить подтверждение того, что все выходные сигналы схемы соответствуют таблице истинности при любых сигналах d2, d1, d0. Формирование сигналов на входах d3..d0 проще всего достигается объединением этих сигналов в шину «d» и подключением к шине кода от двоичного счетчика. За полный цикл счетчика происходит перебор всех возможных комбинаций двоичных кодов.

Теоретические временные диаграммы (рис. 2.2) построены по таблице истинности, они используются при тестировании устройства. Правильность работы определяется сравнением теоретических и экспериментальных диаграмм.

Результатом анализа схемы является описание комбинационной схемы на языке Verilog (пример 2). Первый элемент описания, как правило, – комментарий, который не влияет на компиляцию, допускает русские буквы, начинается с двух символов косая черта (знак деления на клавиатуре). В приведенном примере в комментарии указано назначение описания, он занимает две строки (не уместился в одну строку). Номера остальных строк записаны как комментарии.

```
// Пример 2. Комбинационная          //1
// схема ks1                         //2
module p41_ks1 (d,qand,qor,qxor);    //3
input [2:0] d;                       //4
output qand, qor, qxor ;             //5
assign qand=d[2] & d[1] & d[0] ;     //6
assign qor = d[2]  |  d[1]  |  d[0];  //7
assign qxor=d[2]  ^ d[1]  ^ d[0];    //8
endmodule                            //9
```

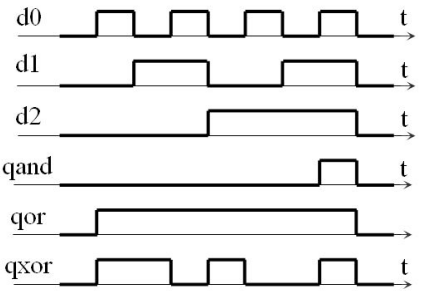


Рис. 2.2. Временные диаграммы, поясняющие работу схемы ks1

Второй элемент описания — заголовок (строка 3), содержит ключевое слово «module», за ним следует имя устройства и перечисление всех входных и выходных сигналов в круглых скобках, через запятую. При записи данной строки используют обозначения выводов, изображенные на схеме или символе устройства. Имя устройства должно содержать только латинские буквы, цифры и знак подчеркивания и начинаться с буквы. В имени не допускаются русские буквы и пробелы. Имя описания «p41_ks1» означает «пример 2». Такое же имя будет иметь файл, содержащий это описание. Выбор имени упрощает поиск файлов в общем каталоге.

Третий элемент описания — назначение входных и выходных сигналов, которые называют также «порты ввода - вывода» (строки 4 и 5). После ключевого слова «input» описывается выходной сигнал, который в данном примере представлен как 3-разрядный вектор, или шина, содержащая несколько проводников для передачи параллельного кода. Перед именем вектора в квадратных скобках через двоеточие указан диапазон индексов.

После слова «output» указаны выходные сигналы. Никаких указаний о типе и разрядности сигнала нет. По умолчанию эти сигналы будут назначены как одноразрядные, типа «wire».

Четвертый элемент описания - параллельные операторы, начинающиеся с обязательного ключевого слова «assign» («назначать»), которые применяются при описаниях комбинационных схем. В данном примере в строках 6 – 8 для каждого выходного сигнала записан параллельный оператор присваивания с ключевым словом «assign».

Операторы записаны по логическим функциям. Для получения строки описания на Verilog необходимо записать логическую функцию, и в этой записи заменить символы логических операций, принятые в алгебре логики, на символы языка Verilog. Индексы переменных в языке Verilog необходимо указывать в квадратных скобках после имени сигнала. Для первого выходного сигнала отрока описания получается после введения следующих замен:

$$qand = d2 \cdot d1 \cdot d0;$$

$$assign\ qand = d[2] \,\&\, d[1] \,\&\, d[0];$$

аналогично для второго и третьего сигналов можно записать:

$$qor = d2 \vee d1 \vee d0; \qquad qxor = d2 \oplus d1 \oplus d0;$$

$$assign\ qor = d[2] \,|\, d[1] \,|\, d[0]; \qquad assign\ qxor = d[2]\hat{}\, d[1]\hat{}\, d[0];$$

Завершение описания – ключевое слово endmodule.

Язык Verilog позволяет записать строки 6 – 8 сокращенно, используя операторы свертки, которые выполняют логическую операцию над многоразрядным вектором бит за битом и выдают результат в виде одного бита.

Запись оператора свертки (пример 3) содержит ключевое слово «assign», затем оператор присваивания значения выходному сигналу, в котором после знака равенства записан символ логической операции, а затем имя вектора. Вместо assign qand=(d[2] & d[1] & d[0] можно записать assign qand= & d. Подобная замена операторов показана в примере 3.

```
// Пример 3. Комбинационная          //1
// схема ks1                         //2
module p42_ks1 (d,qand,qor,qxor);    //3
input [2:0] d;                       //4
output qand, qor, qxor ;             //5
assign qand= &  d;                   //6
assign qor = | d;                    //7
assign qxor=  ^ d;                   //8
endmodule                            //9
```

5) Последним пунктом анализа схем является моделирование устройства, которое позволяет проверить правильность функционирования, а также оценить определенные параметры устройства. Моделирование простых схем имеет, в основном, познавательную ценность. Для сложных схем моделирование позволяет оценить аппаратные затраты и динамические параметры.

Анализ устройства, заданного в виде схемы

Пусть требуется выполнить анализ заданной комбинационной схемы ks2 (рис. 2.3).

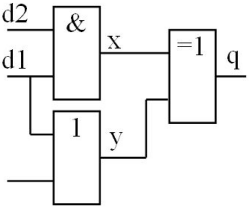


Рис. 2.3. Комбинационная схема ks2

Таблицу истинности можно составить для любой цифровой схемы, состоящей из большого числа логических элементов. Для упрощения составления таблицы целесообразно дополнительно обозначить в схеме внутренние узлы «x» и «y», и добавить для этих узлов столбцы в таблицу истинности. Вначале необходимо заполнить эти столбцы, учитывая работу соответствующих элементов (табл. 2.2).

Таблица 2.2

N	d2	d1	d0	x	y	q
0	0	0	0	0	0	0
1	0	0	1	0	1	1
2	0	1	0	0	1	1
3	0	1	1	0	1	1
4	1	0	0	0	0	0
5	1	0	1	0	1	1
6	1	1	0	1	1	0
7	1	1	1	1	1	0

Сигнал «x » формируется логическим элементом И на входы которого поданы сигналы d2, d1, поэтому, в соответствии с правилом, справедливым для элемента И, получим : x = 1, если d2 = d1 = 1, иначе x = 0.

Сигнал «y» формируется элементом ИЛИ, на входы которого поступают сигналы d1, d0. В соответствии с правилом для указанной операции: y = 0, если d1 = d0 = 0, иначе y = 1.

Используя сигналы «x» и «y» несложно определить выходной сигнал q по правилу для операции «сумма по модулю два»: q = 1, если сумма единиц сигналов «x» и «y» нечётна.

2) По заданной схеме запишем логические функции, для этого необходимо вначале записать уравнения для вспомогательных сигналов «x», «y», а затем — формулу для выходного сигнала «q».

$$x = d2 \cdot d1; \quad y = d1 \vee d0; \quad q = x \oplus y = (d2 \cdot d1) \oplus (d1 \vee d0)\,.$$

Использование скобок гарантирует необходимый порядок выполнения операций.

3) Теоретические временные диаграммы построены с использованием таблицы истинности и логических функций (рис. 2.4).

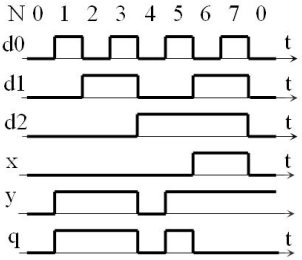


Рис. 2.4. Временные диаграммы, поясняющие работу схемы

4) Описание на языке Verilog рассматриваемой комбинационной схемы (пример 4) выполнено с учетом правил, изложенных в предыдущей главе. Рассмотрим состав этого описания.

```
//Пример 4. Комбинациионная          //1
// схема ks2                         //2
module p43_ks2 (d, q);               //3
input [2:0]d;                        //4
output   q;                          //5
assign q=(d[2]&d[1]) ^(d[1] | d[0]);  //6
endmodule                            //7
```

В данном примере первые две строки — комментарий, следующая строка (строка 3) — заголовок, содержащий имя модуля и перечисление всех входных и выходных переменных. Имя модуля «p43_ks2» означает «пример 4». Такое же имя будет иметь файл, содержащий это описание.

Второй элемент описания — заголовок (строка 3), содержит ключевое слово «module», затем имя устройства, и перечисление всех входных и выходных сигналов в круглых скобках, через запятую. При записи данной строки используют обозначения выводов, изображенные на схеме или символе устройства. Имя устройства должно содержать только латинские буквы, цифры и знак подчеркивания, и начинаться с буквы. В имени не допускаются русские буквы и пробелы.

Третий элемент описания – назначение входных и выходных сигналов, которые называют также «порты ввода - вывода» (строки 4 и 5). После ключевого слова «input» описывается входной сигнал, 3-разрядный вектор, или шина, содержащая три проводника для передачи параллельного кода. Перед именем вектора в квадратных скобках через двоеточие указан диапазон индексов.

После слова «output» указан выходной сигнал. Никаких указаний о типе и разрядности сигнала нет. По умолчанию этот сигнал будет назначен как одноразрядный типа «wire».

Четвертый элемент описания — параллельный оператор, начинающийся с обязательного ключевого слово «assign» (назначать), которые применяются при описаниях комбинационных схем. В данном примере оператор записан по логической функции.

Для получения строки описания на Verilog необходимо записать логическую функцию и в этой записи заменить принятые в алгебре логики символы логических операций на символы языка Verilog. Индексы переменных в языке Verilog необходимо указывать в квадратных скобках после имени сигнала:

$$q = (d2 \cdot d1) \quad \oplus \quad (d1 \vee d0);$$

$$assign\ q = (d[2] \,\&\, d[1])\text{^}\,(d[1] \mid d[0]).$$

Завершение описания — ключевое слово endmodule.

Порядок выполнения работы

1. Получить задание на анализ комбинационной схемы от преподавателя.
2. Выполнить анализ комбинационной схемы.
3. Сдать отчёт о проделанной работе преподавателю.

Практическое занятие № 3

Синтез комбинационных схем

Цель работы

Изучить этапы синтеза комбинационных схем. Научиться выполнять синтез комбинационных схем.

Теоретическая часть

Под синтезом устройств комбинационного типа понимают проектирование и разработку устройства по определенным правилам, обеспечивающим оптимальное решение поставленной задачи. Результатом синтеза являются данные, позволяющие построить схему. Этими данными могут быть логические уравнения или описание на одном из HDL-языков.

Вначале перечислим этапы синтеза логических устройств комбинационного типа.

1. Составление технического задания.
2. Составление таблицы истинности.
3. Запись логических функций.
4. Минимизация логических функций.
5. Разработка схемы.
6. Разработка тестовых сигналов. Построение временных диаграмм.
7. Описание устройства на языке описания аппаратных средств.

Составление технического задания

Началом синтеза является четкая и формулировка задачи, которую должна решать схема. Должен быть однозначно определен закон функционирования. Необходимо определить входные и выходные сигналы, а также изобразить символ логического устройства (рис. 3.1).

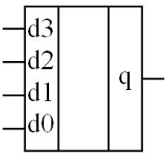


Рис. 3.1. Символ комбинационной схемы

В соответствии с ГОСТ символы логических элементов изображаются в сетке 2,5 мм, входы — слева, выходы — справа, в среднем поле — символическое отображение функционального назначения устройства.

Составление таблицы истинности

Таблица истинности содержит значения выходных сигналов при различных значениях входных сигналов и составляется в соответствии с техническим заданием. В таблице истинности перечисляются все возможные комбинации входных сигналов, называемые наборами, и записывается выходной сигнал, соответствующий каждому набору входных сигналов. Количество строк в таблице равно количеству наборов N и составляет $N=2^n$, где n – число входов.

Максимальное количество различных логических функций Fmax зависит, в свою очередь, от количества наборов N и составляет $Fmax=2^N$. С увеличением количества входных переменных максимальное количество функций существенно возрастает (табл. 3.1).

Таблица 3.1

Количество входов	Количество наборов N	Максимальное количество функций F
1	$2^1 = 2$	$2^2 = 4$
2	$2^2 = 4$	$2^4 = 16$
3	$2^3 = 8$	$2^8 = 256$
4	$2^4 = 16$	$2^{16} = 65\,536$
5	$2^5 = 32$	$2^{32} = 4\,291\,367\,296$

Количество выходов комбинационной схемы m обычно значительно меньше максимального количества функций. Для каждой выходной функции в таблице истинности отводится отдельный столбец.

Таблица может быть задана в техническом задании на проектирование. Например, для комбинационной схемы, приведенной на рис. 3.1 таблица истинности будет содержать 4 строки и 4 столбца выходных сигналов (табл. 3.2).

Таблица 3.2

№ набора	Входные сигналы $d_1\ d_0$	Выходные сигналы $q_3\ q_2\ q_1\ q_0$
0	0 0	0 1 1 0
1	0 1	0 0 1 1
2	1 0	1 1 1 1
3	1 1	1 0 0 0

Составление логических функций

Комбинационная схема может быть описана логическими функциями, представленными в совершенной дизъюнктивной нормальной форме (СДНФ) или в совершенной конъюнктивной нормальной форме (СКНФ). Термин «совершенная» или «каноническая» означает, что функция описывает все возможные комбинации входных сигналов, а термин «нормальная форма» означает, что в логическом выражении, определяющем функцию, последовательно выполняются не более чем две операции типа И, ИЛИ, И — НЕ, ИЛИ — НЕ.

Логическая функция в СДНФ представляет собой дизъюнкцию вспомогательных функций - минтермов.

Минтерм (другое название - конституента единицы) - это произведение всех входных сигналов, записанных без инверсии, или с инверсией, которое равно 1 только на данном наборе аргументов. Каждый минтерм соответствует одной строке таблицы истинности, в которой функция равна 1. Если в таблице истинности количество строк, для которых функция равна единице, сравнительно небольшое, то логическую функцию целесообразно записать в СДНФ. Для этого необходимо выполнить следующие действия.

1) Записать несколько произведений всех аргументов, число которых равно числу единиц логической функции в таблице истинности.
2) Соединить произведения знаками дизъюнкции.
3) Записать под каждым произведением двоичный набор, для которого функция равна единице. Над аргументом, стоящим напротив нуля, необходимо поставить инверсию.

Другая форма записи логической функции — СКНФ — представляет собой конъюнкцию — произведение — вспомогательных функций — макстермов. Макстерм (или конституента нуля) — это дизъюнкция всех входных сигналов, записанных без инверсии, или с инверсией, которая равна 0 только на данном наборе аргументов. Каждый макстерм соответствует строке таблицы истинности, в которой функция равна 0. Если небольшим является количество строк, для которых функция равна нулю, целесообразно выбрать запись в СКНФ.

Для получения логических уравнений в СКНФ записываются дизъюнкции всех аргументов, которые заключаются в скобки и соединяются знаками умножения. Число сомножителей равно числу строк в таблице истинности, для которых функция равна нулю. Под каждой дизъюнкцией записывается двоичный набор, на котором функция равна нулю. Над аргументами, равными единице, в дизъюнкции ставится инверсия.

Получение уравнений по таблице истинности рассмотрим на примере комбинационной схемы (рис. 3.1), для которой задана таблица истинности (табл. 3.2).

Для комбинационной схемы, содержащей 2 входных сигнала, таблица истинности содержит 4 строки для записи всех возможных наборов входных сигналов. Для большей наглядности при тестировании устройства в таблицу добавляют столбец, содержащий номера наборов в десятичной системе счисления.

Максимальное количество логических функций для 2-входовой схемы равно 16, однако заданная схема имеет 4 выходных сигнала (отдельные одноразрядные сигналы, или параллельный код).

Выделяя для каждой функции столбец, запишем выражения в СДНФ

$$q_3 = d_1 \cdot \overline{d_0} \vee d_1 \cdot d_0; \quad q_2 = \overline{d_1} \cdot \overline{d_0} \vee d_1 \cdot \overline{d_0};$$

$$q_1 = \overline{d_1} \cdot \overline{d_0} \vee \overline{d_1} \cdot d_0 \vee d_1 \cdot \overline{d_0}; \quad q_0 = \overline{d_1} \cdot d_0 \vee d_1 \cdot \overline{d_0}.$$

Выражения в СКНФ будут иметь вид:

$$q_3 = (\overline{d_1} \vee \overline{d_0}) \cdot (d_1 \vee \overline{d_0}); \quad q_2 = (d_1 \vee \overline{d_0}) \cdot (\overline{d_1} \vee \overline{d_0});$$

$$q_1 = \overline{d_1} \vee \overline{d_0}; \quad q_0 = (d_1 \vee \overline{d_0}) \cdot (\overline{d_1} \vee d_0).$$

Минимизация логических функций

Логические функции, записанные по таблице истинности, в ряде случаев могут быть упрощены и записаны в минимальной дизъюнктивной нормальной форме (МДНФ) или в минимальной конъюнктивной нормальной форме (МКНФ). Минимизация логических функций позволяет уменьшить аппаратные затраты при построении комбинационной схемы, при этом улучшаются другие параметры проектируемого устройства: повышаются его надежность и быстродействие.

Минимизация может быть выполнена аналитически с использованием теорем и законов алгебры логики. Так, например, логическую функцию q3 можно минимизировать, используя логическую функцию склеивания:

$$q_3 = d_1 \cdot \overline{d_0} \vee d_1 \cdot d_0 = d_1 \cdot (\overline{d_0} \vee d_0) = d_1$$

Аналитические методы минимизации не всегда позволяют определить, является ли полученное уравнение минимальным.

Простым и эффективным методом минимизации логических уравнений является метод карт Карно (другое название — диаграммы Вейча — Карно), позволяющий осуществить операции склеивания и найти минимальные дизъюнктивную, или конъюнктивную формы.

Карта Карно представляет собой прямоугольную таблицу. Для функции n переменных карта Карно состоит из 2n клеток, каждая из которых соответствует определенной строке таблицы истинности и, соответственно, определенному набору входных переменных. В клетки записываются соответствующие значения функции. Если таблица истинности содержит значения для нескольких функций, то для каждой функции строится отдельная карта Карно.

Разметка строк и столбцов карты Карно выполняется таким образом, чтобы для любой клетки было легко определить соответствующую ей комбинацию переменных. Если для обозначения строк таблицы выбрать старший двоичный разряд кода набора, а для обозначения столбцов — младший разряд, то объединение этих значений будет соответствовать коду набора клетки, расположенной на пересечении выбранных строки и столбца. Выбранный порядок нумерации обозначен в верхней левой части таблицы (d1 \ d0). Номера наборов клеток приведены на рис. 3.2.

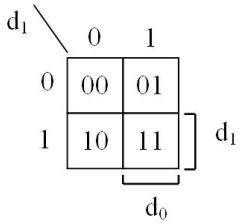


Рис. 3.2. Нумерация клеток в карте Карно для двух переменных

Для отображения значений аргументов, соответствующих клеткам карты Карно также используют квадратные скобки, которые соответствуют значению аргумента, равному 1.

Отсутствие скобки означает равенство аргумента нулю. На приведенном рисунке показана двойная разметка строк и столбцов карты, несмотря на ее избыточность.

Для того, чтобы с помощью карты Карно задать функцию, необходимо в каждую клетку с определенным номером занести значение функции из соответствующей строки таблицы истинности. Клетка, в которой записана «1», отображает минтерм, а в которой «0» — мактерм.

Минимизация логических функций методом карт Карно заключается в выделении на карте контуров, и записи выражений, которые описывают эти контуры.

При поиске минимальной дизъюнктивной нормальной формы уравнения (МДНФ) выделяются контуры, охватывающие соседние единицы карты, нули можно не указывать. Число единиц в правильном контуре должно быть равно степени числа 2, то есть: 2, 4, 8… При этом требуется, чтобы контуры были максимальны по числу охватываемых единиц, а число всех контуров было минимальным. Контуры могут перекрываться. Для контура, содержащего 2 единицы, пропадает (склеиваются) 1 переменная, если контур содержит 4 единицы – пропадает 2 переменных и т. д.

Рассмотрим выделение контуров на картах Карно, составленных для логических функций, заданных таблицей истинности (табл. 3.2).

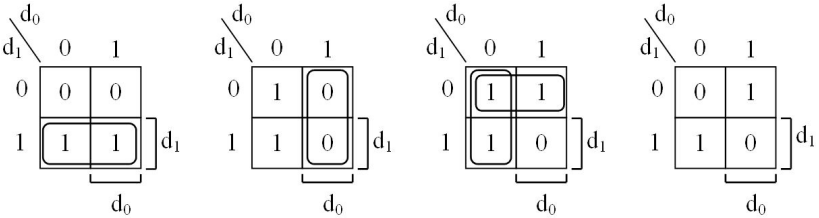


Рис. 3.3. Выделение контуров на картах Карно

Для функции q3 можно выделить контур, охватывающий две единицы, для которого переменная d0 принимает значения 0 и 1, и пропадает, а переменная d1 не изменяется, d1=1. Поэтому, МДНФ для функции q3 имеет вид: $q_3 = d_1$.

Подобную ситуацию имеем для функции q2, пропадает переменная d1, а переменная d0=0.поэтому $q_2 = \overline{d_0}$.

В примере для функции q1 можно выделить два контура, которые перекрываются, что позволяет записать эту функцию в виде: $q_2 = \overline{d_0} \vee \overline{d_1}$.

Для функции q0 выделить контуры, содержащие несколько единиц или нулей, не удается. Форма записи этой функции, полученная по таблице истинности, является минимальной – МДНФ, или МКНФ:

$$q_0 = \overline{d_1} \cdot d_0 \vee d_1 \cdot \overline{d_0}; \quad \text{или } q_0 = (d_1 \vee \overline{d_0}) \cdot (\overline{d_1} \vee d_0).$$

При поиске минимальной конъюнктивной нормальной формы уравнения (МКНФ) функция представляется в карте Карно ее нулями, единицы можно не указывать, выделяются контуры, охватывающие нули. Контуры заменяются элементарными дизъюнкциями, которые соединяются знаками конъюнкции.

Для логических функций трех переменных карта представляет собой таблицу, содержащую 8 клеток (рис. 3.4). Клетки нумеруются числами от 0 до 2^n–1.

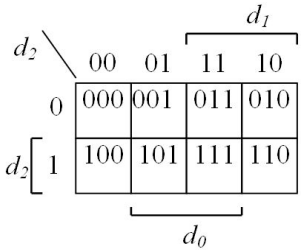


Рис. 3.4. Нумерация клеток в карте Карно для трех переменных

Для обозначения строк таблицы выбран старший двоичный разряд кода набора, а для обозначения столбцов — младшие разряды. Выбранный порядок нумерации обозначен в верхней левой части таблицы (d2\d1d0).

Полученные логические функции в МДНФ, или в МКНФ используют для построения схем.

Для проверки работоспособности полученной схемы проводят ее моделирование, в специально выбранной среде разработки. При моделировании используют специально разработанные тестовые сигналы, позволяющие выполнить полный контроль работоспособности и оценить параметры схемы.

Для нумерации клеток по столбцам используется код Грея, значения которого для соседних клеток различаются только в одном разряде. Это коды 00, 01, 11, 10. Данную последовательность кодов можно получить при обходе по часовой стрелке клеток карты Карно для двух переменных (рис. 3.2).

Важно отметить, что крайние клетки, расположенные в таблице слева и справа являются смежными. Действительно, коды, соответствующие крайнему левому столбцу (00), и крайнему правому столбцу (10) различаются только в одном разряде.

Карта Карно для четырех переменных имеет 16 клеток (рис. 3.5). Строки соответствуют четырем возможным комбинациям двух старших разрядов кода набора d3d2 = 00,01,11 и 10. Аналогично, столбцы помечены комбинациями d1d0. Эти метки отображают информацию, необходимую для определения кода набора, соответствующего каждой клетке карты.

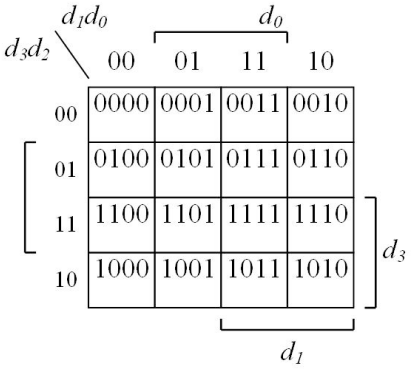


Рис. 3.5. Нумерация клеток в карте Карно для четырёх переменных

Также расставлены квадратные скобки, которые для каждой из четырех переменных определяют области на карте. Каждая область, отмеченная скобкой, — это часть карты, в пределах которой указанная переменная равна 1. Очевидно, что скобки несут ту же самую информационную нагрузку, что и метки, которыми помечены строки и столбцы.

Логическая функция называется «не полностью определенной», если на некоторых наборах она может принимать любое значение — 0 или 1, что обозначают буквой «x». При минимизации для переменной «x» можно выбрать любые значения функции (0 или 1) для увеличения размеров, или количества контуров.

Пример «Синтез формирователя признака числа»

Рассмотрим пример, синтеза комбинационной схемы ks3, которая формирует признак принадлежности числа заданному диапазону.

1. Техническое задание

Задание. Синтезировать комбинационную схему, формирующую одноразрядный признак q в соответствии с условием: q = 1, если d > 8.

Входной сигнал — 4-разрядное двоичное число d.

Выходной сигнал — признак q, содержит 1 разряд.

Заданному устройству соответствует символ (рис. 3.6).

2. Составление таблицы истинности

В рассматриваемом примере имеем логическую функцию четырех переменных, для которой количество наборов равно 16, поэтому таблица истинности также будет содержать 16 строк. Максимальное количество различных функций в данном случае равно 256, но в соответ-

ствии с заданием в устройстве должна быть определена только одна функция — заданный признак числа (табл. 3.3). Таблица дополнена столбцами, содержащими минтермы и макстермы заданной функции. Для каждой строки таблицы, в которой функция равна 1, записан минтерм, а для строки, где функция равна 0, то макстерм.

Таблица 3.3

D_3	D_2	D_1	D_0	q	Минтермы функции q	Макстермы функции q
0	0	0	0	0		$M_0 = d_3 \vee d_2 \vee d_1 \vee d_0$
0	0	0	1	0		$M_1 = d_3 \vee d_2 \vee d_1 \vee \overline{d_0}$
0	0	1	0	0		$M_2 = d_3 \vee d_2 \vee \overline{d_1} \vee d_0$
0	0	1	1	0		$M_3 = d_3 \vee d_2 \vee \overline{d_1} \vee \overline{d_0}$
0	1	0	0	0		$M_4 = d_3 \vee \overline{d_2} \vee d_1 \vee d_0$
0	1	0	1	0		$M_5 = d_3 \vee \overline{d_2} \vee d_1 \vee \overline{d_0}$
0	1	1	0	0		$M_6 = d_3 \vee \overline{d_2} \vee \overline{d_1} \vee d_0$
0	1	1	1	0		$M_7 = d_3 \vee \overline{d_2} \vee \overline{d_1} \vee \overline{d_0}$
1	0	0	0	0		$M_8 = \overline{d_3} \vee d_2 \vee d_1 \vee d_0$
1	0	0	1	1	$m_9 = d_3 \cdot \overline{d_2} \cdot \overline{d_1} \cdot d_0$	
1	0	1	0	1	$m_{10} = d_3 \cdot \overline{d_2} \cdot d_1 \cdot \overline{d_0}$	
1	0	1	1	1	$m_{11} = d_3 \cdot \overline{d_2} \cdot d_1 \cdot d_0$	
1	1	0	0	1	$m_{12} = d_3 \cdot d_2 \cdot \overline{d_1} \cdot \overline{d_0}$	
1	1	0	1	1	$m_{13} = d_3 \cdot d_2 \cdot \overline{d_1} \cdot d_0$	
1	1	1	0	1	$m_{14} = d_3 \cdot d_2 \cdot d_1 \cdot \overline{d_0}$	
1	1	1	1	1	$m_{15} = d_3 \cdot d_2 \cdot d_1 \cdot d_0$	

В данном примере количество строк, в которых функция равна единице, меньше количества строк, в которых функция равна нулю.

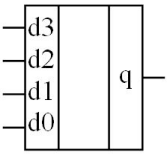


Рис. 3.6. Комбинационная схема ks3

3. Составление логических функций

Логические функции в совершенной дизъюнктивной нормальной форме (СДНФ) представляет собой дизъюнкцию минтермов, число которых равно числу единиц логической функции в таблице истинности. Каждый минтерм (табл. 3.3) является произведением всех аргументов, в котором над аргументом, равным в данном наборе нулю, ставится знак инверсии. Индексы минтермов соответствуют десятичным эквивалентам кодов наборов. Логическая функция для сигнала q в СДНФ имеет вид:

$$q = d_3 \cdot \overline{d_2} \cdot \overline{d_1} \cdot d_0 \vee d_3 \cdot \overline{d_2} \cdot d_1 \cdot \overline{d_0} \vee d_3 \cdot \overline{d_2} \cdot d_1 \cdot d_0 \vee d_3 \cdot d_2 \cdot \overline{d_1} \cdot \overline{d_0} \vee$$
$$\vee d_3 \cdot d_2 \cdot \overline{d_1} \cdot d_0 \vee d_3 \cdot d_2 \cdot d_1 \cdot \overline{d_0} \vee d_3 \cdot d_2 \cdot d_1 \cdot d_0$$

Для получения логических функций в совершенной конъюнктивной нормальной форме (СКНФ) записываются в скобках все макстермы, и соединяются знаками умножения. Число сомножителей равно числу строк в таблице истинности, для которых функция равна нулю.

Для каждого макстерма учитывается двоичный набор, на котором функция равна нулю, над аргументами, равными единице, в дизъюнкции ставится инверсия. Логическая функция для сигнала q в СКНФ имеет вид:

$$q = (d_3 \vee d_2 \vee d_1 \vee d_0) \cdot (d_3 \vee d_2 \vee d_1 \vee \overline{d_0}) \cdot (d_3 \vee d_2 \vee \overline{d_1} \vee d_0) \cdot (d_3 \vee d_2 \vee \overline{d_1} \vee \overline{d_0}) \cdot$$
$$\cdot (d_3 \vee \overline{d_2} \vee d_1 \vee d_0) \cdot (d_3 \vee \overline{d_2} \vee d_1 \vee \overline{d_0}) \cdot (d_3 \vee \overline{d_2} \vee \overline{d_1} \vee d_0) \cdot (d_3 \vee \overline{d_2} \vee \overline{d_1} \vee \overline{d_0}) \cdot$$
$$\cdot (\overline{d_3} \vee d_2 \vee d_1 \vee d_0).$$

Минимизация логических функций

Для минимизации логических функций используют метод карт Карно. Карта Карно для минимизации логической функции «q» должна содержать 16 клеток, что соответствует числу строк в таблице истинности (табл. 3.3). Нумерация клеток по столбцам и по строкам производится в коде Грея (рис. 3.7).

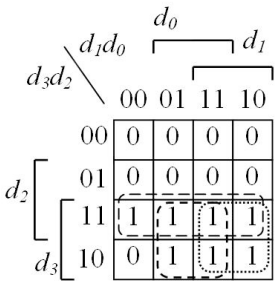


Рис. 3.7. Карты Карно четырёх переменных для функции q

Карта Карно, составленная для рассматриваемого примера, позволяет выделить три контура, каждый из которых сдержит 4 клетки, и записать логическое уравнение в МДНФ:

$$q = d_3d_2 \vee d_3d_1 \vee d_3d_0 = d_3(d_0 \vee d_1 \vee d_2).$$

Разработка схемы

Схема устройства (рис. 3.8) строится по минимизированным логическим уравнениям.

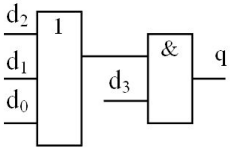


Рис. 3.8. Схема формирования признака

Для формирования выходного сигнала «s» необходимы инверсные значения входных сигналов, для их формирования используются инверторы (элементы NOT). В схеме использовано логическое соединение проводников. Проводники, имеющие одинаковые имена, соединены.

Описание комбинационной схемы на языке Verilog

Наиболее удачным в данном случае является поведенческое описание, в основе которого лежит зависимость выходного сигнала от входных сигналов (пример 5). Укажем содержание строк.

Строки 1, 2 — комментарий, строка 3 — заголовок, в котором после ключевого слова module записано имя модуля (означающее пример 5) и в скобках все входные и выходные сигналы. Далее, в строках 4, 5 все сигналы описаны, вход d — вектор, выход q – по умолчанию одноразрядный сигнал.

```
// Пример 5. Комбинационная схема ks3      //1
module p44_ks3 (d, q);                     //2
input [2:0] d;                             //3
output q ;                                 //4
assign q= d>8;                             //5
endmodule                                  //6
```

Описание поведения, другими словами — работы, предельно краткое (строка 6), в которой предписано назначать выходному сигналу q результат операции сравнения d > 8. Ранее было указано, что входной сигнал d является 4-разрядным вектором, число 8 также будет представлено в виде 4 двоичных разрядов. Компилятор языка Verilog составит схему сравнения двух 4-разрядных чисел. Результат этого сравнения сигнал разрядности 1 бит (1 или 0), который будет назначен сигналу q.

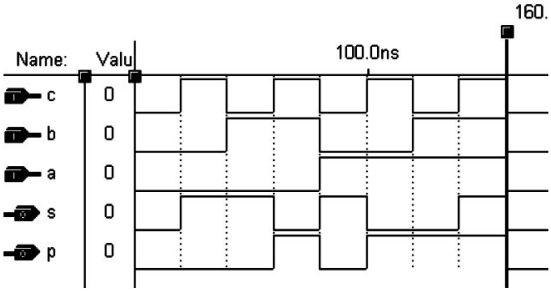


Рис. 3.9. Моделирование сумматора комбинационной схемы ks5 (временные диаграммы)

Описание цифровых устройств на HDL-языках существенно упрощает проектирование.

Порядок выполнения работы

1. Получить задание для синтеза комбинационной схемы от преподавателя.
2. Выполнить синтез комбинационной схемы.
3. Сдать отчёт о проделанной работе преподавателю.

Практическое занятие № 4

Разработка комбинационной схемы на ПЛИС

Цель работы

Научиться создавать логические схемы на ПЛИС и проводить анализ схем с помощью программы «Quartus II».

Теоретическая часть

Для полного тестирования несложных устройств комбинационного типа необходимо выполнить перебор всех возможных комбинаций входных сигналов. Входные сигналы сумматора — слагаемые — равнозначны, поэтому порядок их изменения не имеет значения.

По результатам моделирования (рис. 4.2) можно составить описание работы устройства. Выводы о работоспособности устройства выполняются путем сопоставления теоретических и реальных сигналов и поиска соответствия таблице истинности.

Варианты заданий

Таблица 4.1

Вариант	Логическое уравнение
1	$Q = \overline{(A \vee B)} \oplus (C \cdot D)$
2	$Q = (A \cdot B) \vee \overline{(C \oplus D)}$
3	$Q = \overline{(A \cdot B) \vee \overline{(C \cdot D)}}$
4	$Q = (A \vee B) \cdot \overline{(C \cdot D)}$
5	$Q = \overline{(A \cdot B)} \oplus (C \vee D)$
6	$Q = (A \vee B) \cdot \overline{(C \oplus D)}$
7	$Q = \overline{(A \oplus B) \cdot \overline{(C \vee D)}}$

Порядок выполнения работы

1. Получить разрешение преподавателя или лаборанта на выполнение работы в лаборатории.
2. Включить компьютер. Запустить выполнение программы «Quartus II».

3. Создать рабочую папку, в которой будет размещаться проект (например, «D:\919\Lab4»).
4. Создать проект: File->New Project Wizard. В появившемся диалоговом окне указать путь к рабочей папке («D:\919\Lab4») и строкой ниже — имя проекта (например, «Project_1»). Остальные настройки можно оставить без изменения. В результате, в рабочей папке должен появиться файл проекта («Project_1.qpf») и другие вспомогательные файлы и папки.
5. Создать файл графического описания схемы: File->New-> Design Files->Block Diagram/ Schematic File. Сохранить этот файл File->Save As. При первом сохранении файла графического описания схемы можно дать ему любое имя, по умолчанию файл предлагается назвать именем «Block1.bdf», но имя главного файла проекта должно совпадать с именем проекта: если проект назван «Project_1.qpf», то файлу графического описания следует дать имя «Project_1.bdf».
6. Вызвать навигатор проекта «Project Navigator», воспользовавшись меню MAX+PLUS II-> Hierarchy Display. Открыть закладку «Files»: на ней отображаются все файлы, добавленные в проект. Созданный ранее файл «Project_1.bdf» добавляется в проект автоматически. Для принудительного добавления новых файлов в проект или удаления файлов из проекта следует использовать меню Project-> Add/Remove Files in Project или контекстное меню, которое открывается при нажатии правой кнопки мыши над пиктограммой «Files» в окне навигатора проекта «Project Navigator».
7. Начертить схему проектируемого устройства. Варианты заданий приводятся ниже. Средства создания схемы расположены на специальной панели, которая по умолчанию расположена вдоль левого края окна графического редактора. Названия любого из этих средств можно узнать с помощью всплывающей подсказки, появляющейся при перемещении на них указателя мыши. Пример схемы приводится на рис. 4.1.

   «Symbol Tool» — библиотека готовых модулей. В составе стандартной библиотеки в разделе «primitives» содержатся стандартные логические элементы (подраздел «logic») и входные и выходные контакты (подраздел «pin»), которые потребуются для составления схемы в данной работе.

   «Selection Tool» — позволяет перемещать добавленные в схему элементы с помощью мыши. После того как элементы схемы добавлены в окно редактора, следует переключиться в режим редактирования схемы «Selection Tool» и соединить элементы.

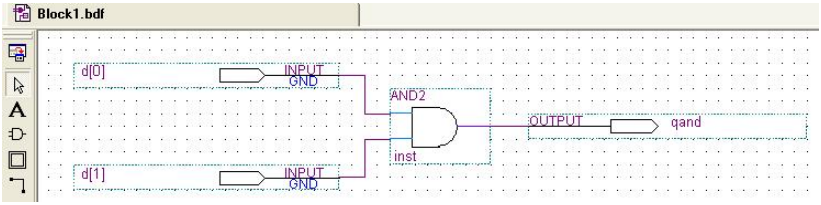


Рис. 4.1. Пример графического описания комбинационной схемы в окне программы «Quartus II»

Нажав клавишу «Ctrl» на клавиатуре и вращая при этом колесо мыши, можно изменять масштаб схемы. Аналогичным образом при выполнении следующих пунктов задания можно менять масштаб временных диаграмм.

8. Изменить названия входных и выходных контактов схемы и указать нулевые начальные значения.
   Свойства элементов схемы задаются в диалоговом окне, появляющемся после двойного щелчка мыши над изображением выбранного элемента. В окне «Pin Properties» следует ввести в первой строке имя входа или выхода, а во второй строке начальное значение «GND» — логический ноль («VCC» — логическая единица).
9. Выполнить компиляцию (проверку правильности и сборку) проекта: Processing->Start Compilation. После компиляции в появившемся окне «Compiler Tool», нажав кнопку «Report», можно получить отчёт, в котором в частности указано число логических элементов «Combinational ALUTs» и количество входных и выходных контактов «Total pins», а рядом в скобках — их доля в общем объёме доступных элементов микросхемы. Зафиксировать эти сведения для отчёта.
10. Построить временные диаграммы, позволяющие проверить правильность работы созданной схемы.
    Создать файл временных диаграмм: File->New-> Verification/Debugging Files->Vector Waveform File. Сохранить этот файл, присвоив ему любое имя (по умолчанию имя файла «Waveform1.vwf»).
    Добавить входные и выходные сигналы: Edit->Insert->Insert Node or Bus или двойным щелчком мыши в левом поле редактора временных диаграмм. В строке «Name» указать имя сигнала, в строке «Type» — тип сигнала («INPUT» — вход, «OUTPUT» — выход),

«Bus width» — ширина шины (число разрядов векторного сигнала).

На вход схемы обычно подают сигнал «Count Value» — сигнал с двоичного счётчика, который обеспечивает полный перебор всех возможных битовых комбинаций. Интерактивная кнопка

находится на панели, расположенной вдоль левого края окна редактора временных диаграмм. После нажатия мышью на эту кнопку открывается диалоговое окно «Count Value», в котором на закладке «Timing» (тактирование) можно указать моменты начала и окончания интервала формирования входного сигнала, а также период дискретизации в строке «Count every». Период дискретизации для всех вариантов равен 100 нс.

11. Выполнить имитационное моделирование проекта: Processing->Start Simulation. Зарисовать для отчёта полученные временные диаграммы. Измерить задержку прохождения сигнала.

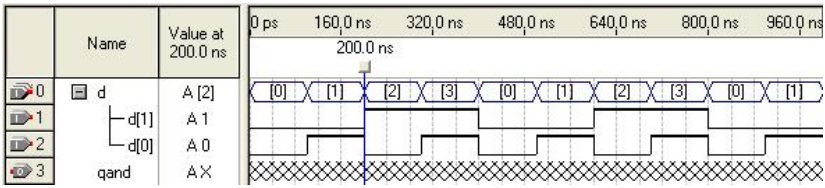


Рис. 4.2. Пример временных диаграмм
в окне программы «Quartus II»

12. Перевести графическое описание разработанной схемы на язык Verilog: File->Create/ Update->Create HDL Design File for Current File->Verilog HDL. Созданный текстовый файл автоматически сохраняется в рабочей папке проекта (например «project_1.v»). Дать модулю другое имя (например «Element1»): чтобы созданный модуль можно было добавить в проект, его имя должно отличаться от имени проекта (имя модуля и имя файла должны совпадать). Сохранить файл текстового описания модуля для отчёта.
13. Добавить созданное устройство в библиотеку готовых модулей: File->Create/Update-> Create Symbol Files for Current File. Созданный файл автоматически сохраняется в рабочей папке (например «Element1.bsf»).
14. Добавить созданный модуль в проект (рис. 4.3).

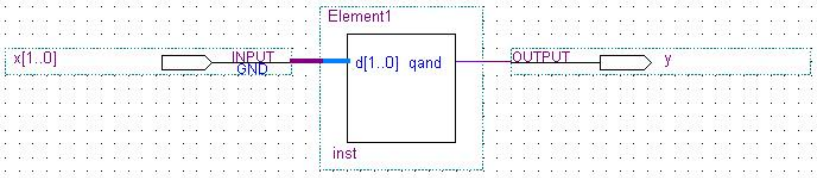


Рис. 4.3. Пример модульного построения схемы
в программе «Quartus II»

15. Построить временные диаграммы, позволяющие проверить правильность работы созданной схемы, и выполнить имитационное моделирование. В диалоговом окне «Count Value» на закладке «Counting» в разделе «Count type» задать последовательность счёта — код Грея («Gray code»).
16. Зарисовать для отчёта полученные временные диаграммы.
17. Результаты работы показать преподавателю.

Содержание отчёта

1. Схема устройства.
2. Текстовое описание, полученное при выполнении п. 12.
3. Временные диаграммы (пп. 11 и 16).
4. Оценки временных задержек и затрат на реализацию (пп. 9 и 11).

Контрольные вопросы

1. Укажите назначение, устройство, сферы применения ПЛИС.
2. Каково назначение языков описания аппаратных средств?
3. Как выполняется анализ комбинационных схем?
4. Назовите известные вам операторы: арифметические, поразрядные, свёртки, отношения и сравнения, условного присваивания.
5. В чём заключается описание на языке «Verilog»: описание по логическим уравнениям, поведенческое описание.
6. В чём различие между параллельными и последовательными операторами? К каким операторам относится оператор условного перехода «if»?
7. Приведите пример использования оператора варианта «case».
8. Как выполняется анализ логических устройств комбинационного типа?
9. Как выполняется анализ комбинационной схемы, заданной логическими функциями?

Практическое занятие № 5

Синтез комбинационных логических схем в программе «Quartus II»

Цель работы

Научиться синтезировать комбинационные логические схемы в программе «Quartus II».

Теоретическая часть

Все системы проектирования устройств на ПЛИС имеют несколько вариантов представления исходных данных проекта. В системе MAX+Plus II предусмотрен ввод исходных данных проекта не только в виде схемы, но и в виде описаний на HDL-языках: на языке Verilog (в виде файла с расширением *.v), на языке VHDL (файл с расширением *.vhd), или на языке Altera HDL (расширение файла *.tdf). Для ввода описания в системе проектирования предназначен текстовый редактор. Проект, введенный в виде описания, компилируется подобно проекту в виде схемы. В обоих случаях результатом является файл для программирования ПЛИС (прошивка).

Для сравнения работы устройств, введённых различными способами, используют иерархические проекты, в которых устройство верхнего уровня иерархии построено из модулей, относящихся к нижнему уровню иерархии. Проект сложного устройства может содержать большое количество уровней иерархии.

Схема иерархического проекта содержит два модуля нижнего уровня иерархии, один из которых (p3) построен по схеме, изображенной на рис. 5.1, а другой (v3) — по текстовому описанию на языке Verilog (рис. 5.2).

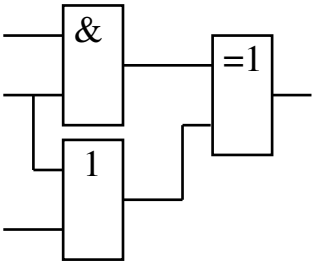


Рис. 5.1. Схема модуля

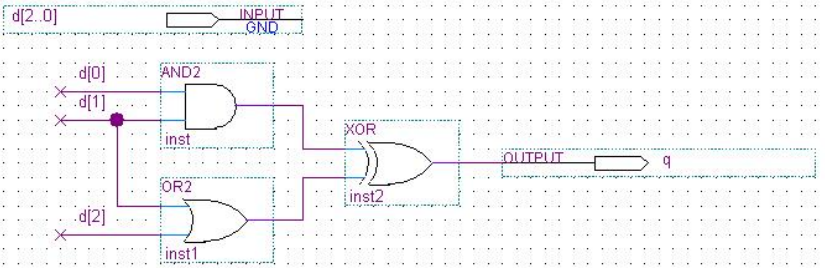


Рис. 5.2. Графическое описание модуля (p3), построенного по схеме, показанной на рис. 5.1

```
// Пример 6. Комбинационная                    // 1
// схема v3                                    // 2
module v3(d, q);                               // 3
input [2:0] d;                                 // 4
output  q;                                     // 5
assign  q = (d[0] & d[1]) ^ (d[2] | d[1]);     // 6
endmodule                                      // 7
```

На входы модулей поданы одинаковые входные сигналы. Моделирование позволяет сравнить выходные сигналы модулей. Для построения иерархического проекта вначале необходимо разработать модули по методике, изложенной в работе 4.

Для разработки первого модуля необходимо создать новый графический файл с именем «s3», ввести схему, изображенную на рис. 5.2, создать символ, то есть выполнить следующие действия.

1. Создание нового файла: «File / New» (Graphic Editor file), «File / Save As» (имя.gdf). Или открытие существующего файла: «File/Open».
2. Ввод элементов схемы (двойной щелчок на схеме — окно «Enter Symbol», библиотека //prim).
3. Соединение элементов схемы.
4. Установка ведущего файла: «File / Project / Set Project To Current File». Команды «File/Open» и «Set Project To Current File» позволяют запускать ранее созданные проекты для продолжения работы.
5. Компиляция: « MAX+plus II/ Compiler».
6. Создание символа: «File/Create Default Symbol».

Более подробно создание модуля рассматривалось в работе 4. Схема в графическом редакторе приведена на рис. 5.2. В результате

буде создан символ устройства, имя которого появится в списке «Symbol Files».

Для разработки второго модуля необходимо создать новый текстовый файл с именем «p3.v», ввести в этот файл описание из приведённого выше примера 6 и создать символ, то есть выполнить следующие действия

1. Создание нового файла: «File / New» (Text Editor File), «File / Save As» (имя.v). Или открытие существующего файла: «File/Open».
2. Ввод описания, или вставка из другого текстового редактора.
3. Установка ведущего файла: «File / Project / Set Project To Current File».
4. Компиляция: « MAX+plus II/ Compiler».
5. Создание символа: «File/Create Default Symbol».

Схема верхнего уровня иерархии (рис. 5.3) содержит ранее созданные символы — два разработанных прежде модуля.

На данной схеме использованы линии различной толщины для изображения одиночных проводников и шин.

В иерархических проектах используют символы модулей с различными способами ввода исходных данных. Двойной щелчок левой кнопкой по символу s3 откроет схему, а по символу p3 откроет пример описания на Verilog.

Варианты заданий

В табл. 5.1 указаны диапазоны четырёхразрядных чисел в шестнадцатеричной форме, в которых проектируемый в лабораторной работе формирователь признаков должен выдать единицу. Для входных величин, не принадлежащих данным диапазонам, на выходе формирователя признака должен появляться ноль.

Таблица 5.1

Вариант	1	2	3	4	5	6	7	8	9	10
Диапазон	0..9	1..8	2..B	3..C	4..D	5..E	6..A	7..C	8..C	5..A

Порядок выполнения работы

1. Получить разрешение преподавателя или лаборанта на выполнение работы в лаборатории.
2. Включить компьютер. Запустить выполнение программы «Quartus II».
3. Создать рабочую папку, в которой будет размещаться проект (например, «D:\919M\Lab5»).

4. Создать проект: File->New Project Wizard. В появившемся диалоговом окне указать путь к рабочей папке («D:\919M\Lab5») и строкой ниже — имя проекта (например, «Project_1»). Остальные настройки можно оставить без изменения. В результате, в рабочей папке должен появиться файл проекта («Project_1.qpf») и другие вспомогательные файлы и папки.
5. Создать файл графического описания схемы: File->New-> Design Files->Block Diagram/ Schematic File. Сохранить этот файл File->Save As. При первом сохранении файла графического описания схемы можно дать ему любое имя, по умолчанию файл предлагается назвать именем «Block1.bdf», но имя главного файла проекта должно совпадать с именем проекта: если проект назван «Project_1.qpf», то файлу графического описания следует дать имя «Project_1.bdf». Имена создаваемых модулей должны совпадать с именами файлов, в которых эти модули описаны.
6. Вызвать навигатор проекта «Project Navigator», воспользовавшись меню MAX+PLUS II -> Hierarchy Display. Открыть закладку «Files»: на ней отображаются все файлы, добавленные в проект. Созданный ранее файл «Project_1.bdf» добавляется в проект автоматически. Для принудительного добавления новых файлов в проект или удаления файлов из проекта следует использовать меню Project-> Add/Remove Files in Project или контекстное меню, которое открывается при нажатии правой кнопки мыши над пиктограммой «Files» в окне навигатора проекта «Project Navigator».
7. Начертить схему проектируемого устройства. Варианты заданий приводятся в табл. 5.1. В каждом варианте требуется создать модуль, формирующий признак сигнала: единица на выходе модуля должна появляться в тех и только тех случаях, когда на двоичный код четырёхразрядной комбинации на входе лежит в заданном в табл. 5.1 диапазоне чисел.

   Средства создания схемы расположены на специальной панели, которая по умолчанию расположена вдоль левого края окна графического редактора. Названия любого из этих средств можно узнать с помощью всплывающей подсказки, появляющейся при перемещении на них указателя мыши. Пример схемы приводится на рис. 5.2.

   «Symbol Tool» — библиотека готовых модулей. В составе стандартной библиотеки в разделе «primitives» содержатся стандартные логические элементы (подраздел «logic») и входные и выходные контакты (подраздел «pin»), которые потребуются для составления схемы в данной лабораторной работе.

«Selection Tool» — позволяет перемещать добавленные в схему элементы с помощью мыши. После того как элементы схемы добавлены в окно редактора, следует переключиться в режим редактирования схемы «Selection Tool» и соединить элементы между собой.

Нажав клавишу «Ctrl» на клавиатуре и вращая при этом колесо мыши, можно изменять масштаб схемы. Аналогичным образом при выполнении следующих пунктов задания можно менять масштаб временных диаграмм.

8. Изменить названия входных и выходных контактов схемы и указать нулевые начальные значения.

   Свойства элементов схемы задаются в диалоговом окне, появляющемся после двойного щелчка мыши над изображением выбранного элемента. В окне «Pin Properties» следует ввести в первой строке имя входа или выхода, а во второй строке начальное значение «GND» — логический ноль («VCC» — логическая единица).
9. Выполнить компиляцию (проверку правильности и сборку) проекта: Processing->Start Compilation. После компиляции в появившемся окне «Compiler Tool», нажав кнопку «Report», можно получить отчёт, в котором в частности указано число логических элементов «Combinational ALUTs» и количество входных и выходных контактов «Total pins», а рядом в скобках — их доля в общем объёме доступных элементов микросхемы. Зафиксировать эти сведения для отчёта.

Создание графического модуля

10. Создать дополнительный файл графического описания. Скопировать в него схему из главного окна проекта.
11. Добавить созданное устройство в библиотеку готовых модулей: File->Create/Update-> Create Symbol Files for Current File. Созданный файл автоматически сохраняется в рабочей папке (например «Element1.bsf»).
12. Добавить созданный модуль в главный файл проекта вместо ранее созданной в нём схемы (рис. 5.3).

Создание текстового модуля

13. Создать дополнительный файл текстового описания (файл с расширением *.v). Ввести текст описания на языке Verilog.
14. Добавить созданное устройство в библиотеку готовых модулей: File->Create/Update-> Create Symbol Files for Current File. Создан-

ный файл автоматически сохраняется в рабочей папке (например «Element1.bsf»).

15. Добавить созданный модуль в проект (рис. 5.3).

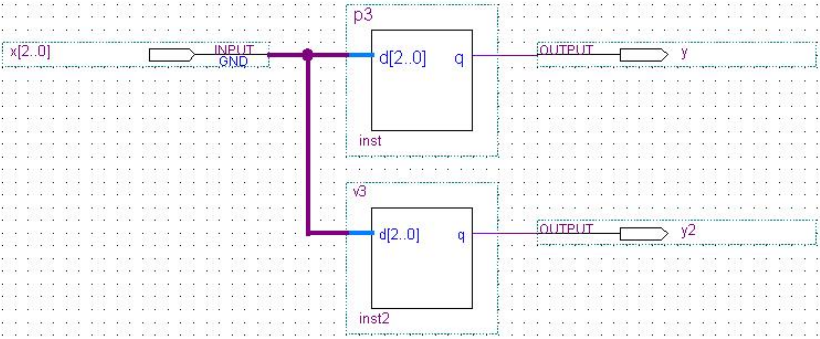


Рис. 5.3. Иерархическая схема, содержащая два модуля

Анализ работы схемы

16. Построить временные диаграммы, позволяющие проверить правильность работы созданной схемы.

Создать файл временных диаграмм: File->New-> Verification/Debugging Files->Vector Waveform File. Сохранить этот файл, присвоив ему любое имя (по умолчанию имя файла «Waveform1.vwf»).

Добавить входные и выходные сигналы: Edit->Insert->Insert Node or Bus или двойным щелчком мыши в левом поле редактора временных диаграмм. В строке «Name» указать имя сигнала, в строке «Type» — тип сигнала («INPUT» — вход, «OUTPUT» — выход), «Bus width» — ширина шины (число разрядов векторного сигнала).

На вход схемы обычно подают сигнал «Count Value» — сигнал с двоичного счётчика, который обеспечивает полный перебор всех возможных битовых комбинаций. Интерактивная кнопка

находится на панели, расположенной вдоль левого края окна редактора временных диаграмм. После нажатия мышью на эту кнопку открывается диалоговое окно «Count Value», в котором на закладке «Timing» (тактирование) можно указать моменты начала и окончания интервала формирования входного сигнала, а также период дискретизации в строке «Count every». Период дискретизации для всех вариантов равен 100 нс.

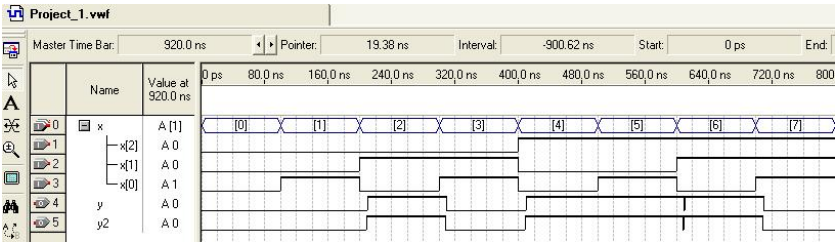


Рис. 5.4. Пример временных диаграмм

17. Построить временные диаграммы, позволяющие проверить правильность работы созданной схемы, и выполнить имитационное моделирование. Сравнить работу двух созданных модулей: графического и текстового.
18. Зарисовать для отчёта полученные временные диаграммы.
19. Результаты работы показать преподавателю.

Содержание отчёта

1. Схема устройства.
2. Графическое и текстовое описания модулей.
3. Временные диаграммы.

Контрольные вопросы

1. Перечислите этапы синтеза комбинационных схем.
2. Приведите пример минимизация логических функций.
3. Опишите по шагам процесс минимизации логических функций методом карт Карно.
4. Приведите пример структурного описания по логическому уравнению.
5. Приведите пример поведенческого описания.
6. Приведите пример описания устройств комбинационного типа на примере мультиплексора.
7. Приведите пример описания устройств комбинационного типа на примере шифратора.
8. Приведите пример описания устройств комбинационного типа на примере АЛУ комбинационного типа.

Практическое занятие № 6

Исследование компараторов кодов и АЛУ комбинационного типа

Цель работы

Научиться выполнять анализ и синтез комбинационных логических компараторов кодов и АЛУ с использованием программы «Quartus II».

Теоретическая часть

Компаратор кода — схемы сравнения кодов чисел. Основными отношениями между двумя двоичными кодами чисел A и B, через которые можно выразить все остальные, являются «равно» (обозначим как E) и «больше» (обозначим как G).

Для формирования признака равенства E чисел A и B необходимо поразрядно выполнить операцию исключающее ИЛИ, и из результатов сравнения отдельных бит сформировать общий результат. Признак E равенства четырехразрядных чисел A и B, принимающий значение 1 при равенстве чисел, можно записать в виде:

$$E = \overline{(A_3 \oplus B_3) \vee (A_2 \oplus B_2) \vee (A_1 \oplus B_1) \vee (A_0 \oplus B_0)}\,.$$

Уравнению соответствует схема (рис. 6.1). Структурное описание на Verilog по логическому уравнению приведено в примере 7, а поведенческое — в примере 8.

```
//Пример 7
module comp_v2 (a, b, e);
input [3:0] a, b;
output  e;
assign e = ~ ( (a[3]^b[3]) | (a[2]^b[2]) | (a[1]^b[1]) | (a[0]^b[0]) );
endmodule

//Пример 8
module comp_v3 (a, b, e);
input [3:0] a, b;
output  e;
assign e = a == b;
endmodule
```

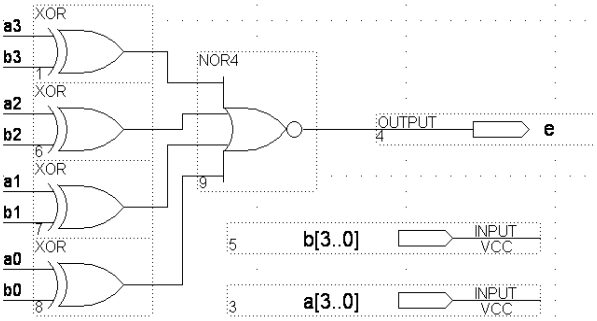


Рис. 6.1. Проект компаратора «comp_v1»

Для формирования признака G необходимо сравнить коды чисел A и B, начиная со старшего разряда. Если числа положительные, и представлены в прямом коде без знака, то признак G будет равен 1, если в старшем разряде числа A записана 1, а числа B записан 0, остальные разряды на результат уже не влияют. Если старшие разряды равны, то значение признака G будут определять следующие разряды:

$$G = A_3 \cdot \overline{B}_3 \vee E_3 \cdot A_2 \cdot \overline{B}_2 \vee E_3 \cdot E_2 \cdot A_1 \cdot \overline{B}_1 \vee E_3 \cdot E_2 \cdot E_1 \cdot A_0 \cdot \overline{B}_0 .$$

Полученное логическое уравнение позволяет составить схему.

АЛУ комбинационного типа

Постановка задачи. АЛУ имеет 4-разрядные входы (a,b) и выход (q); двухразрядный код операции (k); выход переноса (p) (рис. 6.2).

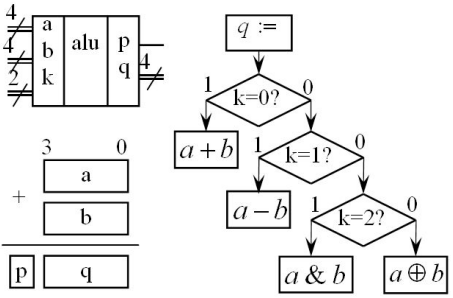


Рис. 6.2. АЛУ комбинационного типа

Пусть заданы коды операций: 0 — суммирование, 1 — вычитание, 2 — поразрядная логическая операция «И», 3 — поразрядная логическая операция «исключающее ИЛИ».

В описании АЛУ (пример 9) строка 2 — заголовок — имя описания и перечисление всех входов и выходов. В строках 3 – 6 описаны: 4-разрядные входные векторы a, b; 2-разрядный вектор кода операции (k); сигнал переноса (p), для которого описание диапазона отсутствует, значит, этот сигнал — одноразрядный; и, наконец, выходной вектор (q).

```
// Пример 9. АЛУ                                    1
module alu (a, b, k, p, q);                         //2
input [3:0] a, b;                                   //3
input [1:0] k ;                                     //4
output p;                                           //5
output [3:0] q;                                     //6
assign  {p,q} = (k == 0) ? (a +b) :
   (k==1)?( a -b) : (k==2) ?  (a & b) : (a ^ b);    //7
endmodule                                           //8
```

Все сигналы по умолчанию имеют тип «wire». С данными сигналами используют только оператор непрерывного назначения с ключевым словом «assign», а также арифметические, логические операторы и оператор условного присваивания.

В строке 7 записан оператор условного присваивания, содержащий вложенные операторы, который выполняет требуемые операции в соответствии с кодом «k». Для улучшения восприятия оператор разделен символами перевода строки, что не влияет на компиляцию.

Сигнал, формируемый оператором присваивания (строка 7), записан как объединение сигналов переноса «p» и выходного вектора «q», они записаны через запятую в фигурных скобках. Результат присваивания — 5-разрядный вектор суммы, или разности, старший бит которого равен «1», если возникает перенос. Работу оператора поясняют рисунки.

Задания к работе

Задание 1. По приведенным формулам составьте схемы преобразования двоичного 4-разрядного двоичного кода «b» в код Грея «g» и обратного преобразования кода «g» в двоичный код «q». Составьте описания преобразователей кодов на языке Verilog. Проведите сравнительное исследование работы преобразователей. Изобразите шаблоны

кодирующих пластин датчика перемещения для двоичного кода и кода Грея, поясните работу преобразователей.

Здание 2. Проведите сравнительный анализ работы компаратора 4-разрядных кодов, формирующего два признака, заданные в табл. 6.1, представленного в виде схемы и описания на Verilog.

Таблица 6.1.

Вариант	1	2	3	4	5	6	7	8	9	10
Признак 1	>=	<	/>	<	>	/>	/<	>=	/<	<=
Признак 2	=	/=	<	=	<=	=	/=	<=	/=	/=

Задание 3. Создайте описание АЛУ для заданных в табл. 6.2 операций. Разработайте тестовые сигналы. Изобразите теоретические временные диаграммы. Опишите работу устройства.

Таблица 6.2 (начало)

Код операции	№ варианта				
	1	2	3	4	5
00	add	add	add	add	add
01	nand	xor	or	and	xor
10	sub	or	xor	or	and
11	xor	nor	nand	nor	or
Флаги	c_out,z	c_out,n	c_out,p	c_out,z	c_out,n

Таблица 6.2 (окончание)

Код операции	№ варианта				
	6	7	8	9	10
00	add	add	add	add	add
01	nand	and	or	xor	and
10	or	or	nor	nxor	nxor
11	xor	nor	xor	or	or
Флаги	c_out,p	c_out,z	c_out,n	c_out,p	c_out,z

Содержание отчёта

1. Схема устройства.
2. Текстовое описание устройства.
3. Временные диаграммы, подтверждающие корректность работы полученных комбинационных схем.

Контрольные вопросы

1. Назовите области применения дешифраторов и шифраторов.
2. Какие функции выполняют компараторы кодов.
3. Как подключить периодическую последовательность на вход моделируемого устройства и задать параметры этой последовательности?
4. Как задать временную диаграмму произвольного импульсного сигнала?
5. Опишите процесс группирования сигналов в шину при моделировании.
6. Как отобразить результаты моделирования в различных системах счисления?
7. Создайте проект по заданию преподавателя.
8. Составьте схему, формирующую признак L «меньше».
9. Для всех устройств и схем, исследованных в работе, поясните функциональное назначение, изобразите схему, опишите принцип работы, назначение всех выводов схемы, особенности применения.
10. Назовите области применения мультиплексоров.
11. Основные элементы структуры модуля описания на Verilog.
12. Типы сигналов,
13. Операторы присваивания «assign», особенности применения.
14. Поясните правила записи всех операторов, использованных в работе.
15. Способы структурного описания схем.
16. Особенности поведенческого описания.
17. Виды логических операторов
18. Синтаксис и применение оператора условного присваивания.
19. Способы описания сумматоров.
20. Параллельные и последовательные операторы.
21. Опишите использование сигналов в виде шин.
22. Поясните принцип построения АЛУ.
23. Поясните выполнение сдвигов: логического, циклического, арифметического.

Библиографический список

1. Кистрин А.В., Никифоров М.Б. Проектирование цифровых устройств: учебник для студентов учреждений среднего профессионального образования. — М.: Академия, 2016. — 282 с.
2. Стешенко В. Плис фирмы «ALTERA». Элементная база, система проектирования и языки описания аппаратуры. — М.: ДМК Пресс, 2016. — 576 с.
3. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы. Курс молодого бойца. — М.: Додэка XXI, ДМК Пресс, 2015. — 408 с.
4. Грэхэм М., Джонсон Г. Конструирование высокоскоростных цифровых устройств. Начальный курс черной магии. — М.: Вильямс, 2015. — 624 с.
5. Бибило П. Основы языка VHDL. — М.: Либроком, 2016. — 328 с.
6. Пухальский Г., Новосельцева Т. Проектирование цифровых устройств: учеб. для вузов. — М.: Лань, 2012. — 896 с.

Содержание