

Работа 2. Арифметические устройства.

§ 2.1. Полусумматор. Полусумматор - устройство для суммирования двух одноразрядных двоичных чисел a и b . Результатами суммирования являются бит суммы данного разряда s и бит переноса в следующий, старший разряд p . Одноразрядный полусумматор используется для суммирования многоразрядных комбинационных суммирующих схем.

Синтез полусумматора – это постановка задачи, изображение символа и таблицы истинности, запись логических функций, составление схемы и описания на Verilog.

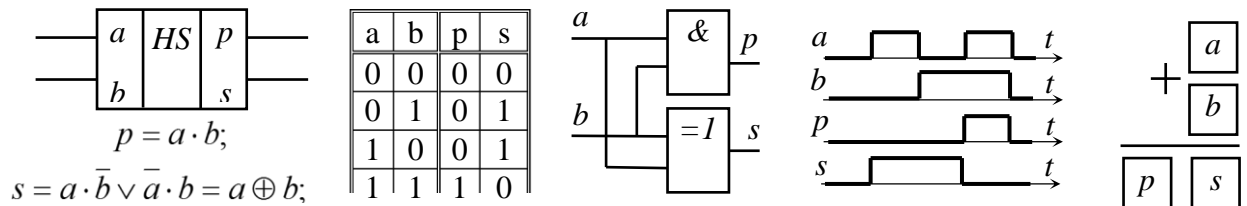


Рис. 2.1. Полусумматор, символ, логические функции, таблица истинности, схема, теоретические временные диаграммы, форматы данных

Символ полусумматора имеет два входа и два выхода. Таблица истинности содержит значения выходных сигналов для всех возможных комбинаций входных сигналов. По таблице истинности составлены логические функции в СДНФ и теоретические временные диаграммы, а по логическим функциям – схема.

```
//поведенческое описание
// полусумматора //2
module v21_hs (a,b, p,s); //3
input a,b; //4
output p,s; //5
assign {p,s}=a+b; //6
endmodule //7
```

Описание полусумматора на Verilog поведенческого типа составлено с учетом алгоритма суммирования двух слагаемых.

Задание 2.1. Иерархический проект полусумматора.

Изучите § 2.1. Для разработки иерархического проекта полусумматора вначале разработайте два модуля нижнего уровня иерархии.

Первый модуль полусумматора с именем $s21_hs.bdf$ создается по схеме (рис. 2.1), а второй - с именем $v21_hs$ - по приведенному описанию на Verilog.

Ведущий модуль верхнего уровня иерархии разрабатывается в графическом редакторе в виде схемы (рис. 2.2) с именем $sv23_hs.bdf$. Буквы sv в обозначении указывают на наличие в проекте модулей в виде схемы и в виде описания на Verilog.

В схеме ведущего модуля одноименные входы модулей нижнего уровня соединены параллельно для подачи на эти модули одинаковых сигналов. Для одноименных выходов модулей, которые имеют различные терминалы,

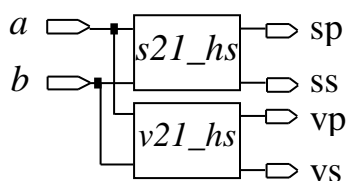


Рис. 2.2 Проект $sv21_hs$

выбираются различные имена, 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 иметь различные временные задержки. При моделировании в САПР

Таблица 2.1

a	b	c	p	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Рис. 2.3. Символ сумматора

двойной щелчок по символу $s2l_hs$ должен открывать схему, а двойной щелчок по символу $v2l_hs$ – описание. Выполните этапы 1 – 9. Выполните сравнение временных задержек.

§ 2.2. Синтез сумматора

Этап 1. определение функционального назначения, входных и выходных сигналов, изображение символа.

Одноразрядный сумматор предназначен для арифметического суммирования трех одноразрядных слагаемых (a, b, c) и формирования суммы для данного разряда (s) и перенос в следующий старший разряд (p). Символ сумматора (рис. 2.3) содержит поля с обозначениями входных

сигналов
(расположенных
слева)
выходных
сигналов
(справа) и

Минтермы и макстермы функций «p» и «s».

Таблица 2.2

a	b	c	p	минтермы функции p	макстермы функции p	s	минтермы функции s	макстермы функции s
0	0	0	0		$M_0 = a \vee b \vee c$	0		$M_0 = a \vee b \vee c$
0	0	1	0		$M_1 = a \vee b \vee \bar{c}$	1	$m_1 = \bar{a} \cdot \bar{b} \cdot c$	
0	1	0	0		$M_2 = a \vee \bar{b} \vee c$	1	$m_2 = \bar{a} \cdot b \cdot \bar{c}$	
0	1	1	1	$m_3 = \bar{a} \cdot b \cdot c$		0		$M_3 = a \vee \bar{b} \vee \bar{c}$
1	0	0	0		$M_4 = \bar{a} \vee b \vee c$	1	$m_4 = a \cdot \bar{b} \cdot \bar{c}$	
1	0	1	1	$m_5 = a \cdot \bar{b} \cdot c$		0		$M_5 = \bar{a} \vee b \vee \bar{c}$
1	1	0	1	$m_6 = a \cdot b \cdot \bar{c}$		0		$M_6 = \bar{a} \vee \bar{b} \vee c$
1	1	1	1	$m_7 = a \cdot b \cdot c$		1	$m_7 = a \cdot b \cdot c$	

функционального назначения..

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.

Для одноразрядного сумматора имеем логическую функцию трех переменных, количество наборов – 8, в количество выходов – 2 (Таблица 2.1). В столбцах, соответствующих входным сигналам, для комбинационных схем записываются двоичные коды натурального ряда чисел. Значение суммы s должно быть равно 1 в строках, для которых количество входных сигналов, равных 1, равно 1 или 3. Заметим, для операции суммирования справедлив переместительный закон, и все слагаемые равнозначны. Значения переноса p равно 1 в строках, для которых количество входных сигналов, равных 1, равно 2 или 3. Значения переноса и суммы образуют двухразрядное двоичное число, равное количеству единиц во входных сигналах.

Этап 3. Составление логических функций. Выполняется по таблице истинности. Для каждого выходного сигнала составляется логическая функция в совершенной дизъюнктивной нормальной форме (СДНФ), или в совершенной конъюнктивной нормальной форме (СКНФ).

Логическая функция в СДНФ – это дизъюнкция минтермов (произведений). Для ее записи необходимо выполнить следующие действия.

- 1) Выделить столбец для выбранной функции, записать несколько произведений всех аргументов, количество которых равно количеству единиц в столбце.
- 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. Будет получена дизъюнкция минтермов

Логическая функция в СКНФ (это конъюнкция всех макстермов - дизъюнкций) необходимо записываются (в скобках) дизъюнкции всех аргументов и соединить их знаками умножения. Количество 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.

В таблице 2.2 приведены минтермы и макстермы функций p и s . Для каждой строки таблицы, в которой функция равна 1 записан минтерм, а для строки, где функция равна 0 – макстерм.

Логические функции в СДНФ для P и S содержат по 4 минтерма.

$$p = \bar{a} \cdot b \cdot c \vee a \cdot \bar{b} \cdot c \vee a \cdot b \cdot \bar{c} \vee a \cdot b \cdot c;$$

$$s = \bar{a} \cdot \bar{b} \cdot c \vee \bar{a} \cdot b \cdot \bar{c} \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot c.$$

Полученные функции могут быть упрощены для получение минимальной дизъюнктивной нормальной формы – МДНФ, выражение для p упрощается, а для s - уже имеет форму МДНФ:

$$p = \bar{a} \cdot b \vee a \cdot c \vee b \cdot c; \quad s = \bar{a} \cdot \bar{b} \cdot c \vee \bar{a} \cdot b \cdot \bar{c} \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot c.$$

Другая форма логических функций – СКНФ - содержит конъюнкцию макстермов.

$$p = (a \vee b \vee c) \cdot (a \vee \bar{b} \vee c) \cdot (a \vee b \vee \bar{c}) \cdot (\bar{a} \vee b \vee c); \quad s = (a \vee b \vee c) \cdot (a \vee \bar{b} \vee \bar{c}) \cdot (\bar{a} \vee b \vee \bar{c}) \cdot (\bar{a} \vee \bar{b} \vee c).$$

Этап 4. Разработка схемы.

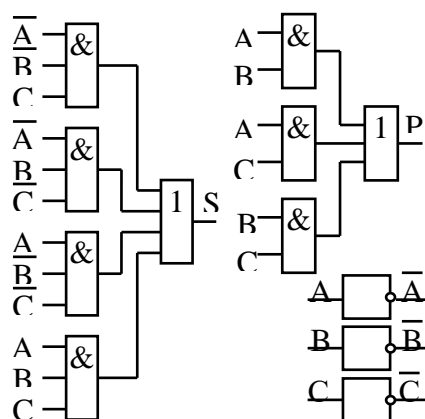


Рис. 2.4. Схема сумматора

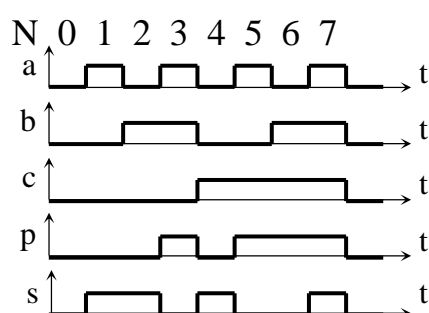


Рис. 2.5 Теоретические временные диаграммы

сигналов и соответствия таблице истинности.

§ 2.3. Описание одноразрядного сумматора на языке Verilog.

1 вариант. Описание по логическим функциям.

Одноразрядный сумматор рассмотрен в § 1.2, где приведены его символ и логические функции

Схема устройства (рис. 2.4) строится по логическим функциям. В соответствии с постановкой задачи схема содержит три входных и два выходных терминала (a,b,c,p,s). 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.

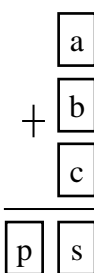
Этап 5. Разработка теоретических временных диаграмм. Проверку

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. Входные сигналы сумматора - слагаемые – равнозначны, поэтому порядок их изменения не имеет значения.

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

```
// описание сумматора по      1
//логическим функциям        2
module v23_sum_var1 (a,b,c,p,s);//3
input  a,b,c;                  //4
output p,s;                    //5
assign p=a&b | a&c | b&c;      //6
assign s= ~a & ~b & c | ~a & b & ~c |
a & b & ~c | a & b & c;        //7
endmodule                      //8
```

```
// поведенческое              1
// описание сумматора         2
module v23_sum_va2 (a,b,c,p,s);//3
input  a,b,c;                  //4
output p,s;                    //5
assign {p,s}=a+b+c;            //6
endmodule                      //7
```



$p = \bar{a} \cdot b \vee a \cdot \bar{b} \cdot c; \quad s = \bar{a} \cdot \bar{b} \cdot c \vee \bar{a} \cdot b \cdot \bar{c} \vee a \cdot \bar{b} \cdot \bar{c} \vee a \cdot b \cdot c$. Рассмотрим первый вариант описания сумматора – по логическим функциям. Выполняется по методике, изложенной в § 2.2, вместо логических функций записаны параллельные операторы с ключевым словом assign, и использованы обозначения логических функций, принятые в Verilog.

Язык Verilog при компиляции воспринимает одинаково пробел, перевод строки и возврат каретки, что дает свободу при форматировании текста описания. Строка 6 разбита на две строки.

2 вариант. поведенческое описание.

В его основу составляет алгоритм функционирования, оно использует более высокий уровень абстракции, короче и проще в восприятии. Поведенческое описание сумматора поясняет представленный рисунок.

В устройстве суммируются три слагаемых, а результат представляют две одноразрядные переменные. Из таблицы истинности следует, что сигналы p и s можно объединить в двухразрядное число, равное сумме входных слагаемых.

Алгоритм функционирования описывает строка 6. Двухразрядное число, полученное объединением сигналов p и s, равно сумме всех входных слагаемых. Использован оператор объединения (конкатенации) сигналов p и s, которые записаны в фигурных скобках через запятую.

Поведенческое описание, основанное на алгоритме, проще в восприятии, целесообразность его использования проявляется особенно ярко при проектировании сложных устройств.

Задание 2.2. Проект сумматора по схеме.

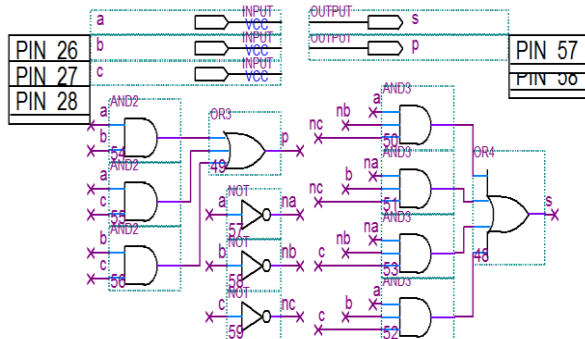


Рис.2.6. Схема сумматора s22_sum

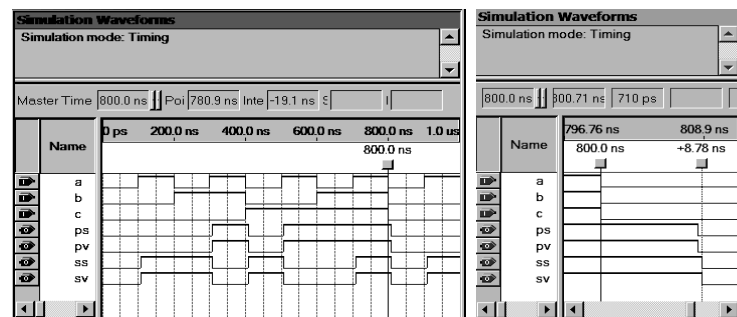


Рис. 2.7.. Измерение временных задержек

Изучите § 2.2. Разработайте проект сумматора по схеме (рис. 2.6) с именем s22_sum, выполните этапы 1-10 по описанию, приведенному в справочном материале. Подключите отладочную плату. Составьте таблицу подключения выводов ПЛИС.

Опишите работу устройства при подаче входных сигналов от кнопок и подключении к выходам светодиодов. Убедитесь, что Формирование выходных сигналов соответствует таблице истинности. Определите ременные задержки для сигналов p и s (рис. 2.7.).

Задание 2.3. Иерархический проект сумматора по описаниям на Verilog

Изучите § 2.2, § 1.5. Разработайте иерархический проект сумматора с именем sv23_sum. Буквы s и v в имени обозначают наличие в проекте двух модулей, разработанных по схеме и по описанию.

Создайте проект верхнего уровня иерархии по схеме, по методике, описанной в

задании 1.5, , содержащий разработанные модули с именами: s22_sum и v23_sum_va2, выполните этапы 1 – 9 и сравнение временных задержек.

§ 2.4. Параллельный сумматор.
Функциональное назначение. Разрабатываемое устройство предназначено для суммирования двух 4-разрядных слагаемых a, b и получения 4-разрядной суммы q и переноса p.
Символ и схема 4-разрядного сумматора содержит один полусумматор и 3 одноразрядных сумматора (рис. 2.8), . Для суммирования слагаемых в младшем разряде используют полусумматор, который формирует сумму данного разряда и перенос в следующий разряд. В остальных разрядах суммируются входные слагаемые и перенос из предыдущего разряда посредством сумматоров.

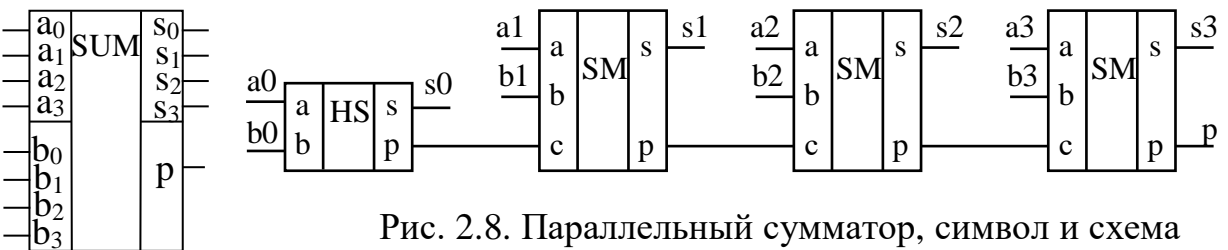


Рис. 2.8. Параллельный сумматор, символ и схема

§ 2.5. Описание параллельного сумматора на языке Verilog.
Первые две строки – комментарий. Третья строка – заголовок, содержит имя описания (указывающее, что это 4-разрядный сумматор) и перечисление всех входных и выходных сигналов. Строка 4 описывает входные сигналы «a» и «b» как 4-разрядные векторы (параллельные коды). Перед именами сигналов в квадратных скобках через двоеточие указан индекс старшего разряда, а затем индекс младшего разряда. Сигналы имеют одинаковую разрядность, поэтому они описаны в одной строке.

Рис. 2.9 Пояснение работы сумматора

```

//Поведенческое описание
//параллельного сумматора
module v24_4sum (a,b,s,p); //3
input [3:0] a,b; //4
output [3:0] s //5
output p; //6
assign {p,s}=a+b; //7
endmodule //8

```

Строка 5

подобным образом описывает выходной сигнал «s». Отдельной строкой описан выходной сигнал «p». По умолчанию он будет назначен как одноразрядный типа «wire». Поведение, или алгоритм

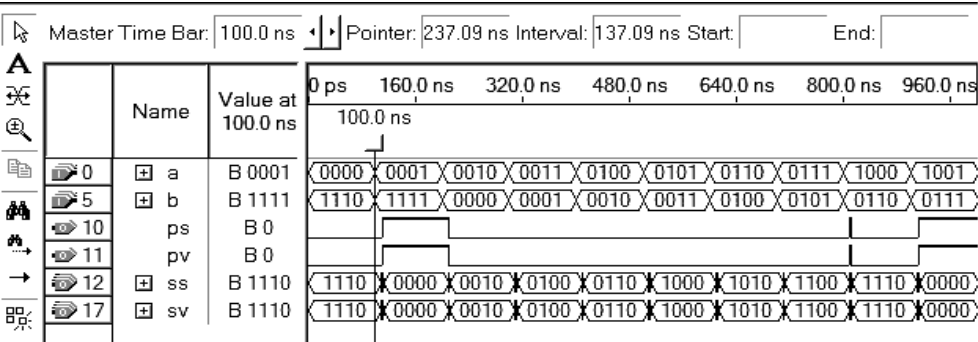


Рис. 2.10. Временные диаграммы 4-разрядного сумматора функционирования описывает строка 7, смысл которой поясняет рис. 2.9.

Теоретические временные диаграммы (рис. 2.10) предназначены для проверки правильности суммирования во всех разрядах и формирования сигнала переноса.

Задание 2.4. Иерархический проект 4-разрядного сумматора

1. Изучите § 2.4. Разработайте проект первого модуля иерархического проекта 4-разрядного сумматора по схеме (рис. 2.8) с именем s24_4sum, используя ранее разработанные модули полусумматора с именем s21_hs и сумматора с именем s22_sum.

Для ввода символов модулей в окне Symbol, которое открывается двойным щелчком, откройте библиотеку Project. Выполните этапы 1-5, создайте символ первого модуля проекта s24_4sum.

2. Разработайте проект второго модуля 4-разрядного сумматора по поведенческому описанию с именем v24_4sum. Выполните этапы 1-5, создайте символ.

3. Разработайте ведущий модуль с именем sv24_4sum по схеме, содержащей модули s24_4sum и v24_4sum, входы которых соединяются параллельно, а выходные сигналы выведены на отдельные терминалы с различными именами. В их обозначения добавляется буква «s» для модуля со схемой, или буква «v» для модуля с описанием на Verilog. Выполните этапы 1 – 9. Сравните временные задержки.

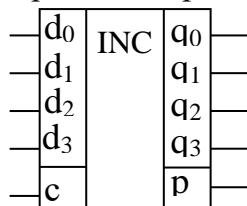


Рис.2.11. Символ инкремента

§ 2.5. Инкрементор

Инкрементор – устройство, выполняющее прибавление единицы к входному коду. Входами инкремента (рис. 2.11) являются 4-разрядный код d[3..0], к которому прибавляется сигнал c (от слова Carry - перенос). При c = 0 инкремента нет. Выходной код обозначен q[3..0], выходной перенос – p. Результат прибавления 1 к n – разрядному коду может иметь n+1 разряд.

В схеме инкремента (рис. 2.12) в каждом разряде используется полусумматор.

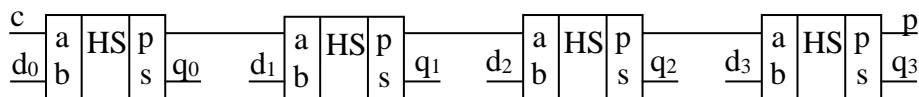


Рис.2.12. Схема инкремента

При описании инкремента на Verilog учитывается, что бит выходного переноса p и 4-разрядный выходной код q образуют 5-разрядное число. Поведенческое описание инкремента на Verilog подобно описанию сумматора. В операторе объединения исходные векторы записываются в фигурных скобках через запятую. Объединенный вектор может быть записан справа и слева от знака равенства в операторе присваивания.

Входные сигналы при составлении теоретических временных диаграмм выбраны таким образом, чтобы были наглядно отображены функции сигналов с и p.

Задание 2.5. Иерархического проекта 4-разрядного инкремента.

1. Создайте символ первого модуля иерархического проекта с именем s25_inc по схеме (рис. 2.12), выполните этапы 1-5.

2. Создайте проект второго модуля иерархического проекта с именем v24_inc.

```
//Инкрементор 1
module v25_inc d,c,q,p);
input [3:0] d; //3
input c; //4
output [3:0] q; //5
output p; //6
assign {p,q} = d+c;//7
endmodule //8
```

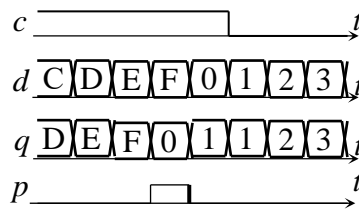


Рис.2.13. Диаграммы инкрементора

3. Создайте проект ведущего модуля иерархического проекта с именем sv24_inc, выполните этапы 1-9 и анализ результатов.

Определите временные задержки, полученные в модулях, введенных различными способами.

Вычислите среднее значение полученных результатов. Это будет **средняя** задержка одного логического элемента.

§ 2.6. Комбинационное устройства сдвига.

По направлению различают сдвиги влево и вправо. При сдвиге влево на один разряд значение числа умножается на 2, а при сдвиге вправо – делится на 2.

По типу (способу выполнения) различают логические, арифметические, и циклические сдвиги. Логические сдвиги используются для умножения или деления на 2 чисел без знака. Арифметический сдвиг вправо для деления на 2 чисел со знаком. Циклические сдвиги, они используются в преобразователях кодов. При логическом сдвиге входного кода d вправо (рис. 2.14) и формировании выходного кода q освобождающийся разряд заполняется нулем, а выдвигаемый разряд является признаком переноса c (от слова Carry) .

Комбинационное устройство логического сдвига вправо в соответствии с рис. 2.11 описывают логические функции: $q_0 = d_1$; $q_1 = d_2$; $q_2 = d_3$; $q_3 = 0$; $c = d_0$.

Схема в САПР (рис. 2.16) строится по приведенным функциям, для однонаправленной передачи сигналов используются буферные элементы.

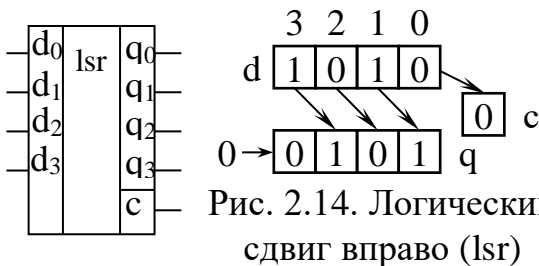


Рис. 2.14. Логический сдвиг вправо (lshr)

```
//Сдвиг v26_lsr
module v25_lsr; //2
input [3:0] d; //3
output [3:0] q; //4
output c; //5
assign {q,c}={d}; //6
endmodule //7
```

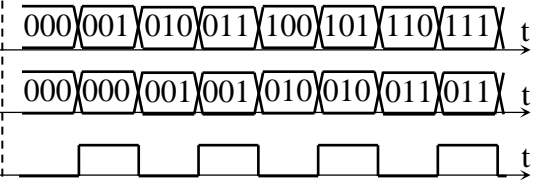


Рис. 2.15. Теоретические временные диаграммы

Описание на Verilog устройства логического сдвига вправо содержит комментарий, заголовок и описание сигналов аналогичные с предыдущим примерам.

Наиболее интересна строка 6. Это поведенческое описание алгоритма

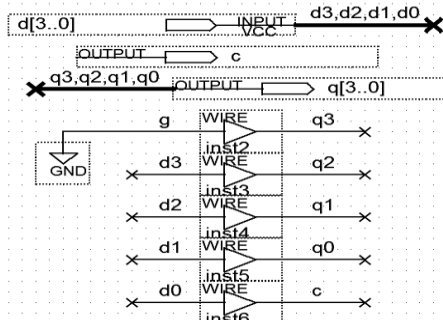


Рис. 2.16. Схема s26_lsr

работы устройства в соответствии с рис. 2.14. В этой строке записано, что выходные сигналы (4-разрядный вектор q и один бит c) объединяются в один 5-разрядный вектор, в котором сигнал c является младшим битом. Этому вектору (его младшим разрядам) присваивается значение 4-разрядного вектора d. В старший бит вектора q, который не указан, по умолчанию будет записан 0.

Теоретические временные диаграммы (рис. 2.15), построенные в соответствии с алгоритмом работы, поясняют деление чисел на 2.

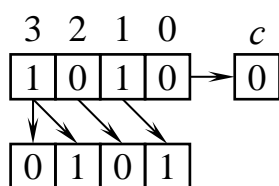


Рис. 2.17.

Арифметический
сдвиг вправо

Комбинационное устройство арифметического сдвига вправо предназначено для деления на 2 чисел со знаком, представленных в дополнительном коде. При арифметическом сдвиге вправо старший знаковый разряд сохраняется и копируется в соседний младший разряд. Эта операция называется расширением знака. Выдвигаемый разряд является признаком переноса c (рис. 2.17).

Задание 2.6. Иерархический проект 4-разрядного устройства логического сдвига вправо.

Разработайте иерархический проект, содержащий модули, введенный по схеме *s26_lsr*, а также введенный по писанию *v26_lsr* и ведущий модуль *sv26_lsr*. При разработке схемы верхнего уровня используйте символы модулей нижнего уровня иерархии. Разработайте теоретические временные диаграммы, выполните моделирование.

По заданию преподавателя разработайте иерархический проект 4-разрядного устройства арифметического сдвига вправо. Выполните моделирование при входных сигналах, соответствующих положительным и отрицательным числам.

§ 2.7. Компараторы кодов

Компаратор кодов – схема сравнения кодов двух чисел, формирующая признаки «больше», «меньше», «равно» и другие. Символ компаратора отображает коды входных чисел и формируемые признаки. Входные сигналы – 4-разрядные коды a и b , a – выходные сигналы – e, g, l , формируемые в соответствии с условиями: $e=1$, если $a = b$; $g = 1$, если $a > b$; $l = 1$, если $a < b$.

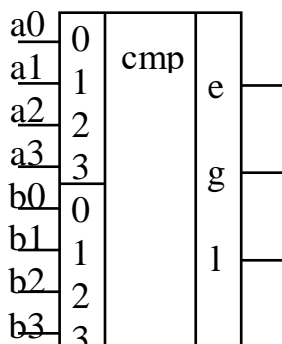


Рис. 2.18. Символ
компаратора кодов

Синтезировать компаратор по общей для комбинационных схем методике, когда по таблице истинности составляются логические функции, затруднительно. Для двух 4-разрядных аргументов таблица будет содержать 8 столбцов входных данных и 256 строк. При увеличении разрядности входных кодов размеры таблицы существенно возрастают.

Необходимо найти способ записи логических функций на основе анализа поразрядного сравнения кодов чисел, начиная со старшего разряда. Достаточно определить признаки «равно» (e) и «больше» (g), используя которые легко

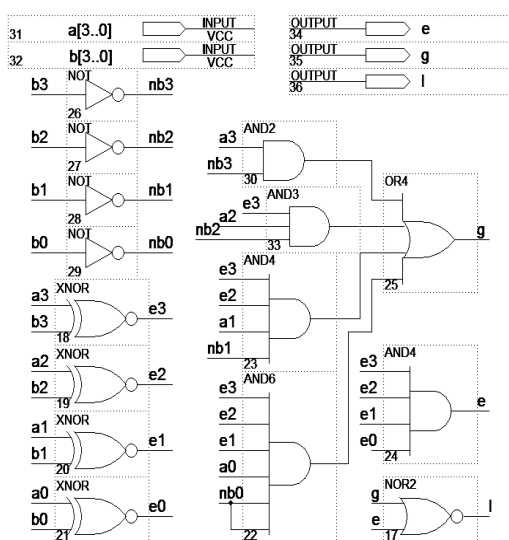


Рис. 2.19. Компаратор кодов

выразить остальные признаки с помощью логических операций.

Признак ($e = 1$) при равенстве отдельных разрядов определяют очевидные формулы:

$$e_3 = \overline{(a_3 \oplus b_3)}; \quad e_2 = \overline{(a_2 \oplus b_2)}; \quad e_1 = \overline{(a_1 \oplus b_1)}; \quad e_0 = \overline{(a_0 \oplus b_0)}.$$

Признак e равенства чисел a и b равен 1, если результаты сравнения всех разрядов равны 1, поэтому: $e = e_3 \cdot e_2 \cdot e_1 \cdot e_0$.

Логическая функция для признака g (больше) записывается следующим образом. Если старший разряд числа a равен 1, а старший разряд числа b равен 0, то $a > b$.

Признак g можно записать в виде $g = a_3 \cdot \overline{b_3} = 1$. Остальные разряды в этом случае на результат уже не влияют.

Если старшие разряды равны, то значение признака g будут определять следующие разряды, логическая функция должна иметь вид: $g = a_3 \cdot \overline{b_3} \vee e_3 \cdot a_2 \cdot \overline{b_2}$.

Рассуждая таким образом получим логическую функцию:

$$g = a_3 \cdot \overline{b_3} \vee e_3 \cdot a_2 \cdot \overline{b_2} \vee e_3 \cdot e_2 \cdot a_1 \cdot \overline{b_1} \vee e_3 \cdot e_2 \cdot e_1 \cdot a_0 \cdot \overline{b_0}.$$

Признак меньше (1) несложно выразить в виде: $l = \overline{e \vee g}$.

По логическим функциям составлена схема (рис. 2.19).

Функциональные возможности языка Verilog позволяют составить компактное и наглядное поведенческое описание компараторов кодов, в которых используются операторы сравнения, формирующие в виде одноразрядных сигналов результаты сравнения параллельных кодов (векторов).

Задание 2.7. Компаратор 4-разрядных кодов. Разработайте иерархический проект компаратора 4-разрядных кодов, формирующего признаки «равно», «больше», «меньше» (рис. 2.18).

1) Создайте проект компаратора кодов по схеме. Введите схему первого модуля с именем `s27_4cmp` в соответствии с рис. 2.19. Создайте символ (этапы 1-5).



Рис. 2.20. Компаратор кодов, временные диаграммы

`sv27_4cmp`, выполните этапы 1-9.

При моделировании компаратора кодов один из входных кодов можно задать от счетчика, а другой задать в виде константы. Для подключения постоянного кода e шине выделите нужную шину в списке сигналов имитатора, нажмите кнопку «Logical States» (кнопка 14 в справочном материале), в строке ввода «Bus State» введите константу, нажмите кнопку «Bus». Опишите результат эксперимента.

§ 2.8.. Матричный перемножитель комбинационного типа

				a3	a2	a1	a0
	×			b3	b2	b1	b0
				s30	s20	s10	s00
+				s31	s21	s11	s01
				s32	s22	s12	s02
				s33	s23	s13	s03
				p7	p6	p5	p4
				p3	p2	p1	p0

Структура матричных умножителей. Основой структуры матричных умножителей являются арифметические выражения, описывающие операцию "умножения столбиком". Перемножение двух целых 4-разрядных двоичных чисел $a[3..0]$, $b[3..0]$ выполняется по известной схеме и дает в результате 8-разрядное произведение $p[7..0]$. Разрядность произведения равна сумме разрядностей

сомножителей.

Слагаемые s_{ij} вида $a_i \cdot b_j$ вырабатываются параллельно во времени конъюнктурами. Их сложение в столбцах, которое можно выполнять разными способами, составляет основную операцию для умножителя и определяет почти целиком время перемножения. Для сложения можно использовать сумматоры sum и полусумматоры hs созданные по описаниям.

Приведенная схема (рис. 2.21) составлена в соответствии с формулами.

Младший разряд произведения получается на выходе конъюнктура: $p_0 = a_0 \cdot b_0$. Для получения первого разряда использован полусумматор, у которого входов в этом случае достаточно. На выходе полусумматора появляется перенос во второй разряд s_2 . Вторым разряд p_2 формируют полусумматор и сумматор, возникает два переноса в третий разряд: s_{31} и s_{32} . От перемены мест слагаемых сумма не изменяется, поэтому порядок подключения входов – произвольный.

Задание 2.8. Разработайте проект матричного перемножителя по приведенной на рис. 2.21 схеме. Модули сумматора (sum) и полусумматора (hs)

<pre> module sum (a,b,c,s,p); input a,b,c; output p,s; assign {p,s}=a+b+c; endmodule </pre>	<pre> module hs (a,b, s,p); input a,b; output s,p; assign {p,s}=a+b; endmodule </pre>	<pre> module v52_mul (a,b,p); input [3..0]a,b; output [7//0] p; assign p=a*b; endmodule </pre>
---	---	--

необходимо построить по приведенным описаниям на `vtrilog`.

Выполните компиляцию. Запишите аппаратные затраты, представленные в окне результата компиляции в строке: - Total logic elements 30 / 240 (13 %). Выполните моделирование, определите временные задержки для наихудшего сочетания входных данных. Рассчитайте максимальную тактовую частоту, при которой обеспечивается корректная работа. Для данного случая приведите в отчете временные диаграммы.

Задание 2.9. Разработайте проект матричного умножителя по описанию на языке Verilog `v29_mul`:

Выполните компиляцию, по результату оцените аппаратные затраты - Total logic elements 33 / 240 (14 %). Выполните моделирование, определите временные задержки для наихудшего сочетания входных данных. Рассчитайте максимальную тактовую частоту, при которой обеспечивается корректная работа. Для данного случая приведите в отчете временные диаграммы.

Задание 2.10. Разработайте проект матричного умножителя 8-разрядных сомножителей по описанию на языке Verilog. Составьте описание `v210_mul`, используйте предыдущее описание, укажите разрядность сомножителей – 8 произведения – 16. Попытка компиляции на установленной ПЛИС выдаст ошибку.

Выберите ПЛИС, обеспечивающую выполнение заданной операции. Выполните компиляцию, по результату оцените аппаратные затраты, сравните с затратами для реализации умножителя 4-разрядных чисел. Выполните моделирование, определите временные задержки для наихудшего сочетания входных данных. Рассчитайте максимальную тактовую частоту, при которой обеспечивается корректная работа. Для данного случая приведите в отчете

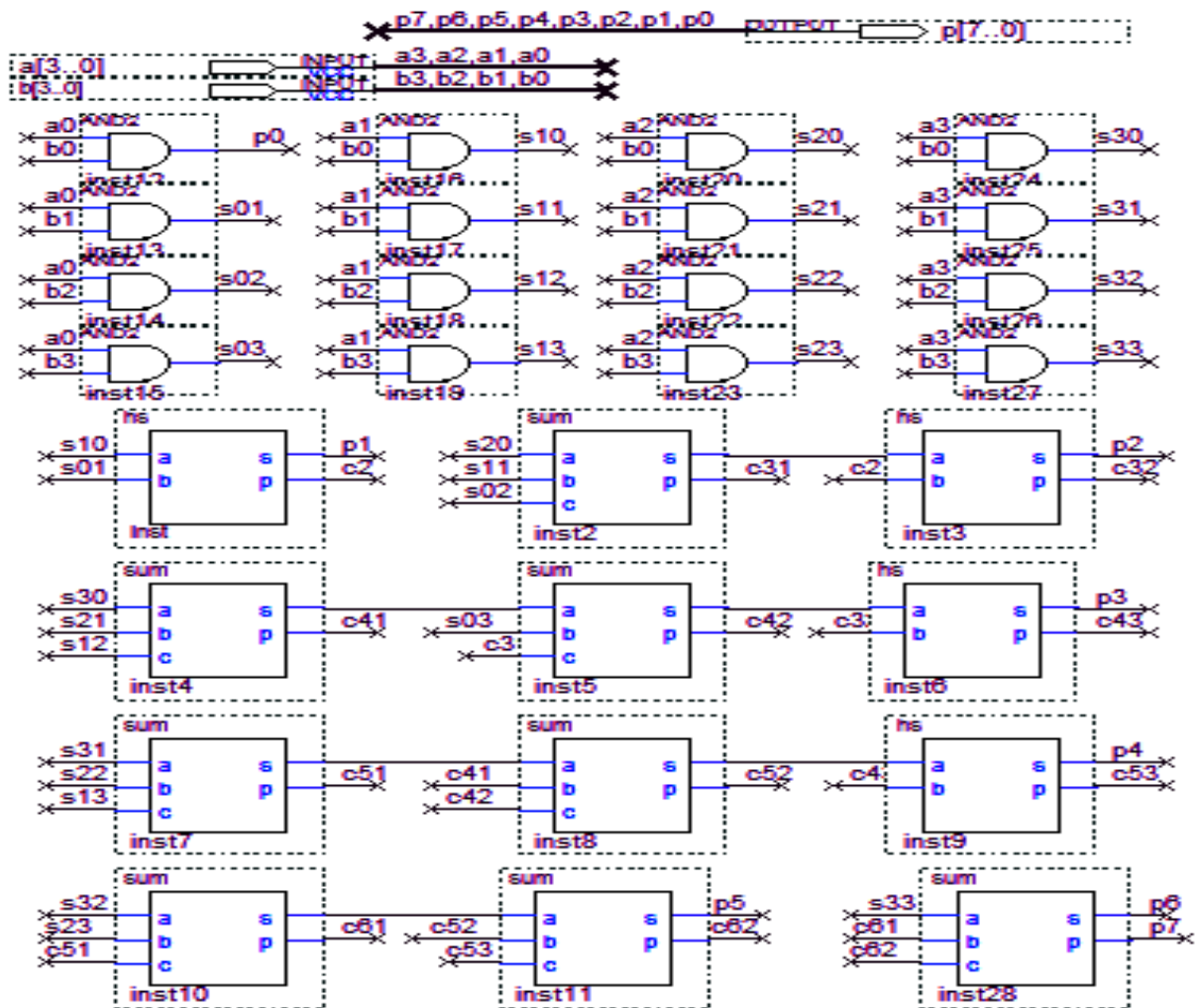


Рис. 2.21. Схема матричного умножителя.

временные диаграммы.

Контрольные вопросы

1. Какие этапы содержит синтез комбинационных схем?
2. Функциональное назначение и символы полусумматора и сумматора.
3. Как составляется таблица истинности для полусумматора и для сумматора?
4. Запишите логические функции, описывающие полусумматор и сумматор.
5. Поясните на примерах составление схем по логическим функциям.
6. Сравните описания сумматора: по логическим функциям и поведенческое. Разработайте
78. схему сумматора, построенного из полусумматоров.

9. Изобразите схему 4-разрядного инкремента, поясните работу схемы.
10. Изобразите схему 4-разрядного сумматора, поясните работу схемы.
11. Как создается иерархический проект, поясните на примере.
12. Укажите типы сдвигов, их функциональное назначение и выполнение.
13. Как изменяется значение числа при сдвиге вправо, или при сдвиге влево?
14. Как создать тестовые входные сигналы для устройств сдвига?
15. Как определить временную задержку сигнала в устройстве?
16. Составьте описание на Verilog 8-разрядного инкремента.
17. Составьте описание на Verilog 8-разрядного сумматора.
18. Составьте описание на Verilog 8-разрядного компаратора кодов.
19. Составьте описание на Verilog 8-разрядного устройства логического сдвига влево.
20. Составьте описание на Verilog 8-разрядного устройства циклического сдвига влево.
21. Определите количество логических элементов в схемах 4-разрядного и 8-разрядного умножителей.
22. Поясните метод построения схемы умножителя.
23. Поясните причину возникновения переходных процессов в умножителе.
24. Как определить максимальную частоту выполнения операций умножения, используя значения временных задержек, полученные экспериментально.1