**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.   
Символ полусумматора имеет два входа и да 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.   
Описание полусумматора на Verilog поведенческого типа составлено с учетом алгоритма суммирования двух слагаемых. ,

//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



*HS*

*a*

*b*

*p*

*s*

*s*

*p*

*a*

*b*

*a*

*p*

*s*

*&*

*=1*

*b*

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | p | s |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

*a*

*b*

*p*

*s*

*t*

*t*

*t*

*t*

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

**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.* Ббуквы 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, 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 двойной щелчок по символу *s21\_hs* должен открывать схему, а двойной щелчок по символу *v21\_hs* – описание. Выполните этапы 1 – 9. Выполните сравнение временных задержек.

Рис. 2.2 Проект *sv21\_hs*

sp

vp

ss

vs

*a*

*b*

*s21\_hs*

*v21\_hs*

**§ 2.2. Синтез сумматора**

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

a

b

c

p

s

sum

Рис. 2.3. Символ  
сумматора

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 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 2n, 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 р 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.

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 |  |  | 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 |  |  |

**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).

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 |

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

Другая форма логических функций – СКНФ - содержит конъюнкцию макстермов.  


**Этап 4. Разработка схемы.**

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.

A

A

С

С

B

B

P

A

B

A

C

B

C

&

1

&

&

A

B

C

A

B

C

A

B

C

A

B

C

1

&

&

&

&

S

Рис. 2.4. Схема сумматора

a

b

p

c

s

t

t

t

t

t

Рис. 2.5 Теоретические  
иременные диаграммы

N 0 1 2 3 4 5 6 7

**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 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 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.

// 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

a

b

s

p

c

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 сумматорп поясняет представленный рисунок.

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

Алгоритм функционирования описывает строка 6. Двухразрядное число, полученное объединением сигналов p и s, равно сумме всех входных слагаемых. Использован оператор объединения (конкатенации) сигналов р 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.**

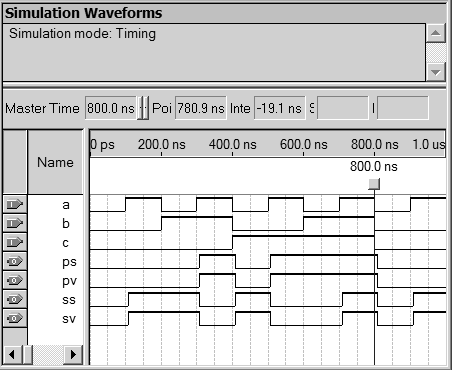
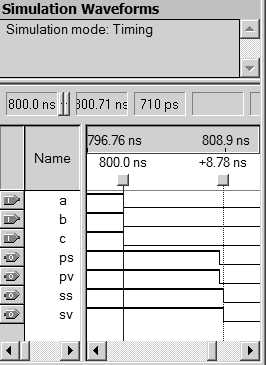


Рис. 2.7.. Измерение временных задержек

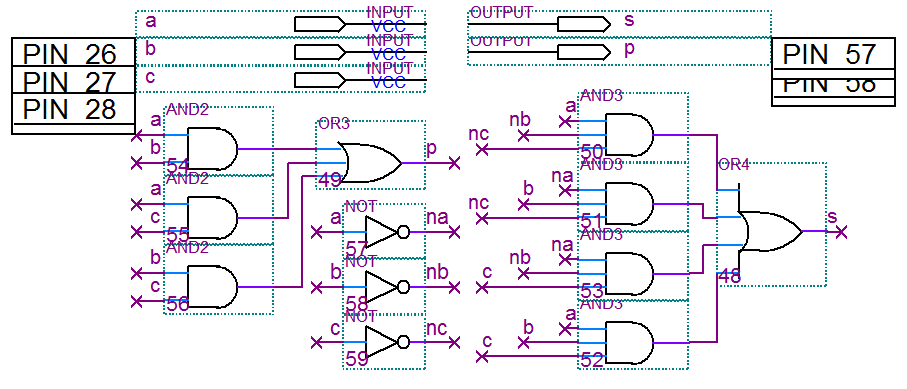


Рис.2.6. Схема сумматора s22\_sum

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 р 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 а, 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.

b0

b3

b1

b2

s0

s3

s1

s2

a0

a3

a1

a2

s3

a3

p

SM

a

s

b

с

p

b3

s2

a2

SM

a

s

b

с

p

b2

s1

a1

SM

a

s

b

с

p

b1

HS

a  
b

s  
p

s0

b0

a0

Рис. 2.8. Параллельный сумматор, символ и схема

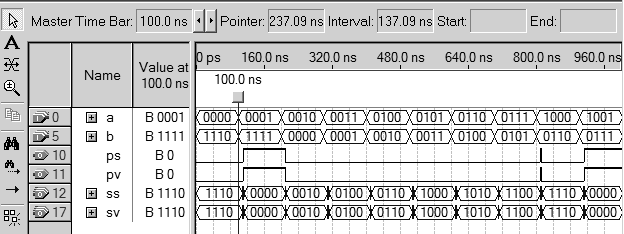
SUM

p

**§ 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.

Строка 5 подобным образом описывает выходчной сигнал «s». Отдельной строкой описан выходной сигнал «р». По умолчанию он будет назначен как одноразрядный типа «wire». Поведение, или алгоритм функционирования описывает строка 7, смысл которой поясняет рис. 2.9.

Рис. 2.10. Временные диаграммы 4-разрядного 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

0

a

b

q

p

3

**+**

Рис. 2.9 Пояснение работы сумматора

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.

**§ 2.5. Инкрементор**

INC

d0

d3

d1

d2

q0

q3

p

q1

q2

Рис.2.11. Символ

инкрементора

с

Инкрементор – устройство, выполняющее прибавление единицы к входному коду. Входами инкрементора (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 инкрементора (рис. 2.12) в каждом разряде используется полусумматор.

HS

a  
b

p  
s

q3

q2

q1

q0

d3

d2

d1

d0

c

p

HS

a  
b

p  
s

HS

a  
b

p  
s

HS

a  
b

p  
s

Рис.2.12. Схема инкрементора

При описании инкрементоора на Verilog учитывается, что бит выходного переноса p и 4-разрядный выходной код q образуют 5-разрядное число. Поведенческое описание инкрементора 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 р.

**Задание 2.5. Иерархического проекта 4-разрядного инкрементора.**

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.

3. Создайте проект ведущего модуля иерархического проекта с именем sv24\_ inc, выполните этапы 1-9 и анализ результатов.

//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

*t*

*t*

*t*

*t*

F

0

D

E

1

1

2

3

*d*

*c*

*q*

*p*

С

D

E

F

0

1

2

3

Рис.2.13. Диаграммы

инкрементора

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 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 с (от слова Carry) .

Комбинационное устройство логического сдвига вправо в соответствии с рис. 2.11 описывают логические функции: 

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

Рис. 2.15. Теоретические временные диаграммы

0000

0011

0100

0111

1000

1011

1100

1111

t

t

t

0000

0000

0011

0011

0100

0100

0111

0111

//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. 2.14. Логический сдвиг вправо (lsr)

1

0

1

0

0

1

0

1

3

2

1

0

0

с

0

d

q

lsr

d0

d3

d1

d2

q0

q3

c

q1

q2

**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 работы устройства в соответствии с рис. 2.14. В этой строке записано, что выходные сигналы (4-разрядный вектор q и один бит с) объединяются в один 5-разрядный вектор, в котором сигнал с 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.

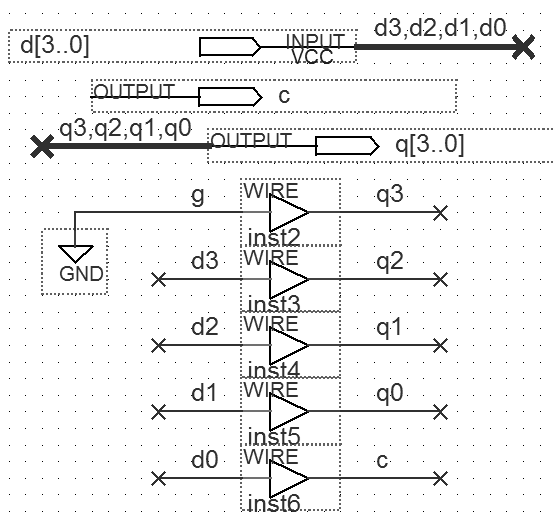


Рис. 2.16. Схема s26\_lsr

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

**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 *с* (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 модулейнижнего уровня иерархии. Разработайте теоретические временные диаграммы, выполните моделирование.

1

0

1

0

0

1

0

1

3

2

1

0

0

*с*

Рис. 2.17. Арифметический  
 сдвиг вправо

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**

0  
1

2

3

0

1

2

3

e  
  
g

l

сmp

a0  
a1

a2

a3

b0

b1

b2

b3

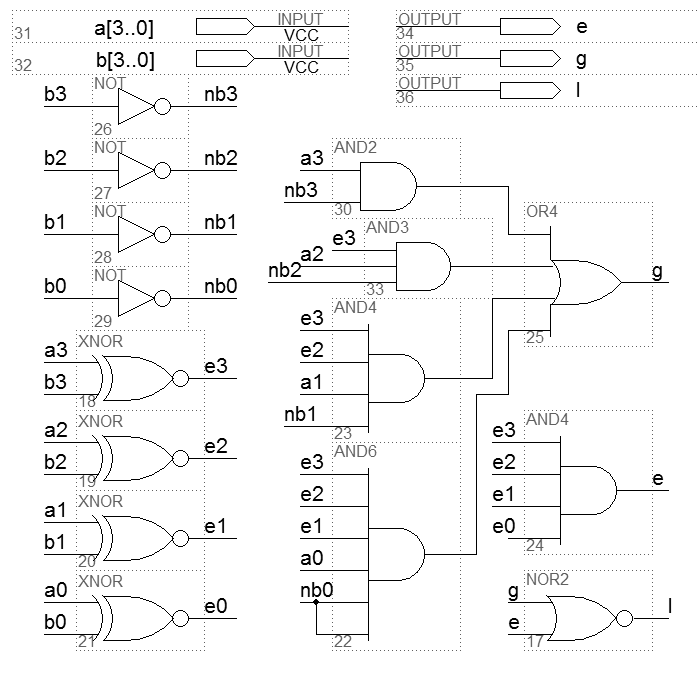
Рис. 2.18. Символ компаратора кодов

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, а* 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.

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» (е) и «больше» (g), используя которые легко выразить остальные признаки с помощью логических операций.  
Признак (е = 1) при равенстве отдельных разрядов определяют очевидные формулы: .   
Признак *e* равенства чисел a и b равен 1, если результаты сравнения всех разрядов равны 1, поэтому: .   
Логическая функция для признака g (больше) записывается следующим образом.   
Если старший разряде числа a равен 1, а старший разряд числа b равен 0, то a > b. Признак g можно записать в виде  Остальные разряды в этом случае на результат уже не влияют.

Рис. 2.19. Компаратор кодов s27\_4cmp



Если старшие разряды равны, то значение признака g будут определять следующие разряды, логическая функция должна иметь вид:   
Рассуждая таким образом получим логическую функцию: .

Признак меньше (l) несложно выразить в виде :. .  
По логическим функциям составлена схема (рис. 2.19).

Функциональные возможности языка Verilog позволяют составить компактное и наглядное поведенческое описание компараторов кодов, в 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).

2) Разработайте второй модуль и символ по описанию v27\_4cmp (этапы 1-5)

module v27\_4cmp

( a, b, e, g, l);

input [3:0]a, b; //3

output e, g, l; //4

assign e = a==b;//5

assign g = a>b; //6

assign l = a<b; //7

endmodule //8

Рис. 2.20. Компаратор кодов, временные диаграммы

0

1

2

3

4

5

6

7

b

a

e

g

l

t

t

t

t

t

4

5

3) Разработайте в графическом редакторе схему ведущего модуля 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 е 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**

**Структура матричных умножителей.** Основой структуры матричных умножителей являются арифметические выражения, описывающие операцию "умножения столбиком". Перемножение двух целых 4-разрядных двоичных чисел а[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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | a3 | a2 | a1 | a0 |
|  |  |  |  | b3 | b2 | b1 | b0 |
|  |  |  |  |  |  |  |  |
|  |  |  |  | s30 | s20 | s10 | s00 |
|  |  |  | s31 | s21 | s11 | s01 |  |
|  |  | s32 | s22 | s12 | s02 |  |  |
|  | s33 | s23 | s13 | s03 |  |  |  |
|  |  |  |  |  |  |  |  |
| р7 | р6 | р5 | р4 | р3 | р2 | р1 | р0 |

Слагаемые sij вида ai\*bj 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 hsсозданные по опаисаниям.

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

Младший разряд произведения получается на выходе конъюнктура: . Для получения первого разряда использован полусумматор, уго входов в этом случае достаточно. На выходе полусумматора появляется перенос во второй разряд с2. Второй разряд р2 формируют полусумматор и сумматор, возникает два переноса в третий разряд: с31 и с32. 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*) необходимо построить по приведенным описаниям на vtrilog.

*module sum (a,b,c,s,p);*

*input a,b,c;*

*output p,s;*

*assign {p,s}=a+b+c;*

*endmodule*

*module hs (a,b, s,p);*

*input a,b;*

*output s,p;*

*assign {p,s}=a+b;*

*endmodule*

module v52\_mul (a,b,p);

input [3..0]a,b;

output [7//0] p;

assign p=a\*b;

endmodule

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 временные диаграммы.

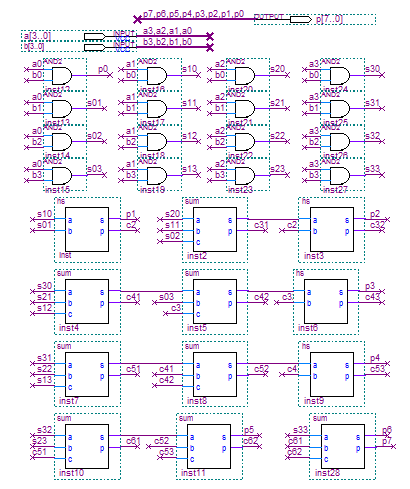


Рис. 2.21. Схема матричного умножителя.

**Контрольные вопросы**

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  78.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21  .  22.  23.  24 | Какие этапы содержит синтез комбинационных схем?  Ффункциональное назначение и символы полусумматора и сумматора.  Как составляется таблица истинности для полусумматора и ддля сумматора?  Запищите ллогические функции, описывающие полусумматор и сумматор.  Поясните на примерах составление схем по логическим функциям.  Сравните описания сумматора: по логическим функциям и поведенческое . Разработайте схему сумматора, построенного из полусумматоров.  Изобразите схему 4-разрядного инкрементора , поясните работу схемы.  Изобразите схему 4-разрядного сумматора, поясните работу схемы.  Как создается иерархический проект, поясните на примере.  Укажите типы сдвигов, их функциональное назначение и выполнение.  Как изменяется значение числа при сдвиге вправо, или при сдвиге влево?  Как создать тестовые входные сигналы для устройств сдвига?  Как определить временную задержку сигнала в устройстве?  Составьте описание на Verilog 8-разрядного инкрементора.  Составьте описание на Verilog 8-разрядного сумматора.  Составьте описание на Verilog 8-разрядного компаратора кодов.  Составьте описание на Verilog 8-разрядного устройства логического сдвига влево.  Составьте описание на Verilog 8-разрядного устройства циклического сдвига влево.  Определите количество логических элементов в схемах 4-разрядного и 8-разрядного перемножителей.  Поясните метод построения схемы перемножителя.  Поясните причину возникновения переходных процессов в перемножителе.  Как определить максимальную частоту выполнения операций умножения, используя значения временных задержек, полученные экспериментально.1 |