

СИНТЕЗ СЧЕТНЫХ СХЕМ

Счетчик — устройство для подсчета числа импульсов, поданных на его вход, цепочка последовательно включенных счетных триггеров, каждый из которых делит частоту повторения входных импульсов на два. Каждый входной импульс изменяет состояние счетчика, которое сохраняется до поступления следующего импульса. Различают суммирующий, вычитающий и реверсивный счетчики. Счетчики обозначают *СТ* (от англ. *counter*).

Каждый триггер счетчика содержит один двоичный разряд двоичного числа, которое однозначно определяет количество импульсов, поступивших на вход. Количество триггеров определяет модуль счетчика. Модуль счетчика M — максимальное количество его состояний. Для n -разрядного счетчика максимальный модуль $M = 2^n$, а период повторения импульсов переполнения содержит 2^n периодов входной частоты. Это означает, что код счетчика может принимать 2^n значений от 0 до $2^n - 1$.

Существуют счетчики с произвольным модулем, который меньше максимального. Для кодирования состояний счетчика используются различные коды, наибольшее распространение получил двоичный код.

Различают асинхронные и синхронные счетчики. *Асинхронный счетчик* — цепочка триггеров, тактовый вход каждого из которых подключен к выходу предыдущего триггера. Сигнал на вход последнего триггера приходит лишь тогда, когда все предыдущие триггеры переключились. В *синхронных счетчиках* тактовые импульсы одновременно подаются на входы «с» всех разрядов, а для управления процессом переключения используются входы J , K , или T . С целью устранения гонок и повышения быстродействия в схемах счетчиков используют цепи ускоренного переноса.

Анализ натурального ряда чисел, записанного в двоичной системе счисления, позволяет построить теоретические временные диаграммы (рис. 12.1) и определить требования к схеме счетчика.

Для построения суммирующего счетчика необходимы триггеры, изменяющие свое состояние на противоположное по каждому входному импульсу. В суммирующем счетчике двоичный код на его выходах увеличивается на единицу с каждым входным импульсом. Если в исходном состоянии на всех триггерах счетчика установлены логи-

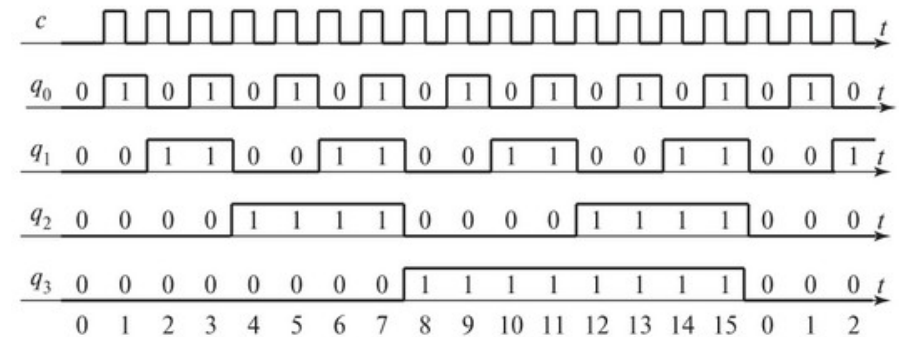


Рис. 12.1. Временные диаграммы работы суммирующего счетчика

ческие нули, то при поступлении импульсов на его вход должна формироваться последовательность кодов 0000, 0001, 0010, 0011, 0100 и т.д.

В суммирующем счетчике каждый триггер изменяет свое состояние на противоположное, когда на него действует спад импульса от предыдущего триггера, если предыдущий триггер переходит из 1 в 0.

Частота на выходе каждого последующего триггера в два раза меньше, чем предыдущего, т.е. каждый триггер делит частоту входного сигнала на два. Это свойство используется в делителях частоты. Частота сигнала на выходе последнего триггера при $n = 4$ меньше частоты входных импульсов в $M = 2^n = 16$ раз. Максимальный код счетчика равен 15, с приходом 16-го импульса происходит переполнение счетчика, все разряды обнуляются и далее процесс повторяется.

В вычитающем счетчике каждый импульс на входе уменьшает на единицу двоичный код на выходах. Процессу вычитания соответствует последовательность двоичных кодов 1111, 1110, 1101, 1100, 1011 и т.д., по которой составлены теоретические временные диаграммы (рис. 12.2).

В вычитающем счетчике каждый триггер изменяет состояние на противоположное по фронту импульса от предыдущего триггера, если предыдущий триггер переходит из 0 в 1.

12.1. Асинхронные счетчики

Асинхронный счетчик — цепочка счетных триггеров, вход каждого из которых подключен к выходу предыдущего триггера. В асинхронных счетчиках под воздействием входного импульса происходит переключение соответствующих разрядов последовательно от разряда

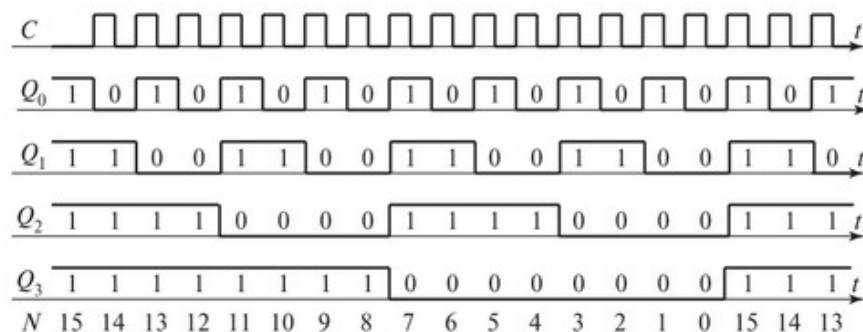


Рис. 12.2. Временные диаграммы работы вычитающего счетчика

обходимо включить последовательно требуемое количество асинхронных триггеров с динамическим управлением (рис. 12.3). При соединении триггеров между собой необходимо учитывать вид сигнала, вызывающего изменение состояния триггера. Для того чтобы триггеры, имеющие прямой динамический вход, изменяли свое состояние по спаду входного сигнала предыдущего триггера, входы триггеров должны быть подключены к инверсным выходам предыдущих триггеров.

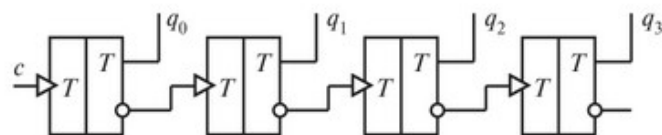


Рис. 12.3. Асинхронный суммирующий счетчик

При поступлении тактовых импульсов на вход C , который называют счетным входом, работа счетчика будет происходить в соответствии с временными диаграммами (см. рис. 12.1), из которых следует, что до прихода первого импульса все триггеры находились в нулевом состоянии. Каждый импульс, поданный на вход c , изменяет выходной сигнал q_0 на противоположный. Каждый спад сигнала q_0 переключает следующий триггер в противоположное состояние. После прихода 16 импульсов все четыре триггера находятся снова в нулевом состоянии. Двоичный код, снимаемый с выходов триггеров, q_3, q_2, q_1, q_0 показывает, сколько импульсов поступило на счетчик.

Описанный счетчик называется асинхронным или последовательным. В нем каждый последующий каскад изменяет состояние после предыдущего. Счетчики с последовательным переносом имеют низкое быстродействие. Максимальное время установления возникает

при переключениях всех разрядов от кода 11...1 к коду 00...0. Сигнал на вход последнего триггера приходит лишь тогда, когда все предыдущие триггеры переключились. Это время пропорционально разрядности счетчика и времени переключения триггеров.

Асинхронный вычитающий счетчик (рис. 12.4) отличается от суммирующего тем, что прямой выход каждого триггера соединен с входом последующего.

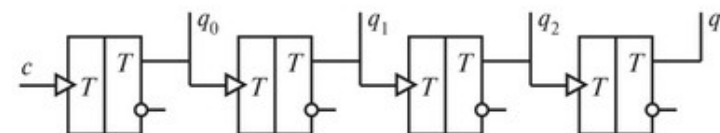


Рис. 12.4. Асинхронный вычитающий счетчик

Реверсивные счетчики могут работать как в режиме сложения, так и в режиме вычитания. Для изменения направления счета необходимо подключать прямой или инверсный выход предыдущего триггера, входящего в счетчик, к входу последующего (рис. 12.5). Реверсивные счетчики изменяют направление счета под воздействием управляющего сигнала, осуществляющего переключение межразрядных связей.

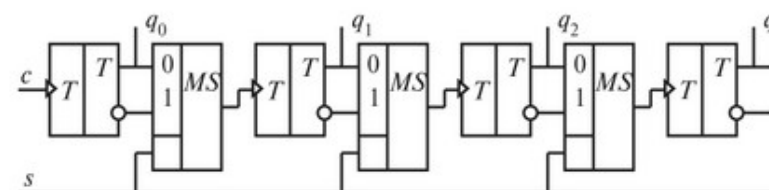


Рис. 12.5. Реверсивный счетчик

Достоинством асинхронных счетчиков является простота схемы: увеличение разрядности производится подключением необходимого числа триггеров. К недостаткам асинхронных счетчиков относятся сравнительно низкое быстродействие и его зависимость от числа разрядов, а также появление промежуточных значений выходных двоичных кодов при последовательном переключении триггеров в новое состояние.

12.2. Синхронные счетчики

В синхронных счетчиках срабатывание всех триггеров происходит по фронту тактовых импульсов, которые одновременно подаются

на входы с всех разрядов, а наличие или отсутствие переключения определяют управляющие входы: j , k или t . Для повышения быстродействия в схемах синхронных счетчиков используют цепи ускоренного переноса (рис. 12.6).

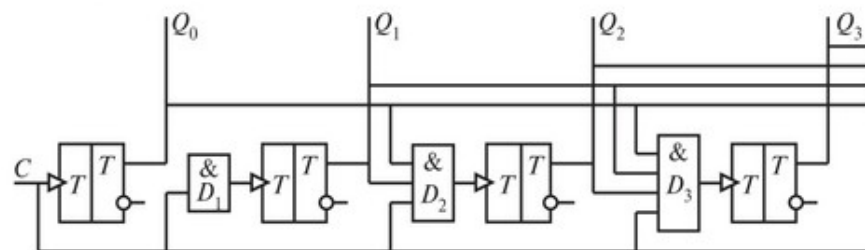


Рис. 12.6. Синхронный счетчик с параллельным переносом

Быстродействие увеличивается в счетчиках со сквозным переносом. Рассматривая пример суммирования двоичных чисел $0011_2 + 1 = 0100_2$, можно сделать вывод, что для получения результата необходимо инвертировать все младшие разряды, равные нулю, и первый разряд, содержащий единицу.

12.3. Многофункциональные счетчики в MAX+plus II

Обычно счетчики имеют несколько входов управления, что расширяет их функциональные возможности.

Многофункциональные счетчики имеют, как правило, режим параллельной загрузки, позволяющий загрузить параллельный код, поданный на дополнительные входы данных, который устанавливает начальное состояние всех триггеров. Режим параллельной загрузкой включается при подаче сигнала 1 на управляющий вход, который может иметь обозначения l , или ldn (от слова Load — загрузка). Данный сигнал поступает на мультиплексоры, переключающие входные цепи триггеров для получения режима счета, или загрузки.

Счетчики, разработанные как библиотечные элементы ПЛИС, являются многофункциональными. Библиотечный элемент «4count» из библиотеки «mf» — 4-разрядный реверсивный счетчик с параллельной загрузкой (рис. 12.7).

Назначение выводов: A, B, C, D — входные данные; QA, QB, QC, QD — выходные данные; CIN — входной перенос; COUT — выходной перенос; LDN (Load Negative) — сигнал, управляющий синхронной параллельной загрузкой входных данных, которая происходит

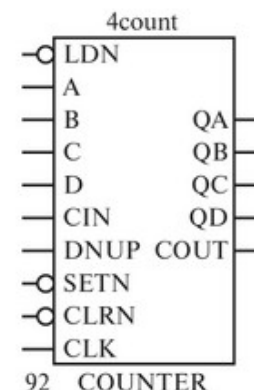


Рис. 12.7. Многофункциональный счетчик на ПЛИС

по фронту синхроимпульса «CLK», при $LDN = 0$; SETN — асинхронная параллельная загрузка; CLRN — асинхронная очистка; DNUP (down-up — вверх-вниз) — направление счета. Работа счетчика при различных комбинациях сигналов описана в Help/Old-Style Macro-functions/Counters. Схема счетчика открывается двойным щелчком по символу (см. рис. 12.7).

Функциональные блоки большой степени интеграции с возможностью изменения параметров по желанию пользователя содержатся в библиотеке параметризованных модулей (Library of Parameterized Modules — LPM).

При вводе символа параметризуемого модуля отображается диалоговое окно настройки, позволяющее выбрать выходы модуля, которые будут отображены и активны, а также настроить параметры сигналов. Подробное рассмотрение библиотечных элементов выполнено в лабораторных работах.

12.4. Счетчики с произвольным модулем счета

Максимальную емкость счетчика определяет модуль $M = 2^n$ который выражается целой степенью числа 2. При использовании счетчика в качестве делителя частоты период повторения импульсов переполнения на выходе счетчика содержит 2^n периодов входного сигнала (рис. 12.8).

Для построения счетчика с произвольным модулем M_1 , используется способ исключения лишних состояний. Счетчики с произвольным модулем используются в качестве делителей частоты, управляемых кодом в частоту, а также для формирования модулированных

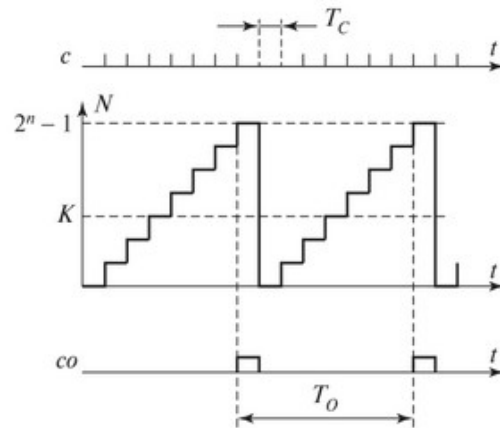


Рис. 12.8. Временные диаграммы, поясняющие работу счетчика

импульсных последовательностей, которые используют для передачи сигналов через каналы связи в устройствах вывода сигналов.

Управляемый делитель частоты с исключением последних состояний счетчика содержит схему, которая формирует импульс принудительного сброса, если код на выходе N достиг определенного значения, заданного входным кодом K . Код счетчика в этом случае будет принимать значения от 0 до K , а период повторения импульсов сброса составит $K + 1$ период синхросигнала.

Исключение в качестве лишних некоторого числа первых состояний достигается путем загрузки начального кода K при переполнении. Код счетчика в этом случае изменяется от значения K до $2^n - 1$, в результате период повторения импульсов переполнения составит $2^n - 1 - K$ периодов входной частоты.

Для построения счетчиков с заданным модулем в ПЛИС хорошо подходят библиотечные многофункциональные счетчики, однако более целесообразным является проектирование подобных устройств с описанием на HDL-языках.

12.5. Особенности описания счетчиков на языке Verilog

Функциональные возможности поведенческого описания устройств на языке Verilog особенно ярко проявляются при описаниях счетчиков и регистров.

Выполним описание 4-разрядного суммирующего счетчика, для которого заданы дополнительные управляющие входы en и r . Вход

он выполняет разрешение счета, если $en = 1$, а при $en = 0$ счетчик должен находиться в режиме хранения. Сигнал r должен выполнять асинхронный сброс, который произойдет сразу же при появлении сигнала $r = 0$, по его отрицательному фронту (спаду).

Заголовок описания (пример 12.1) содержит имя модуля и перечисление всех сигналов. Все входные сигналы, перечисленные после слова «input», по умолчанию будут назначены как одноразрядные типа «wire», выходной сигнал описан как 4-разрядный вектор, для которого указан тип «», что является обязательным при описании устройств с элементами памяти.

```
//Пример 12.1. суммирующий многофункциональный счетчик
module v121_cnt1 (en, c, r, q);
input c, r, en;
output [3:0] q;
reg [3:0] q;
always @ (posedge c or negedge r)
if (~r) q = 4'b0000;
else if (en) q = q + 1;
endmodule
```

В списке чувствительности оператора «always» указано срабатывание по фронту синхросигнала c или по спаду сигнала сброса r .

В качестве условия для оператора *if* записано инверсное значение сигнала r . При $r = 0$ счетчик сбрасывается, а при $r = 1$ и при $en = 1$ выполняется процесс счета.

Пример 12.2 описывает счетчик с параллельной загрузкой и асинхронным сбросом. Для подачи параллельного кода загрузки предусмотрена входная шина d , описанная как 4-разрядный вектор. При $r = 0$ произойдет сброс счетчика, при $r = 1$ и при $l = 1$ — загрузка кода d в счетчик. При $r = 1$ и $l = 0$ устанавливается режим прямого счета, будет выполняться прибавление 1 к коду счетчика по каждому фронту синхросигнала c .

```
//Пример 12.2 счетчик с параллельной загрузкой
module v122_cnt2 (q, d, l, c, r);
input l, c, r;
input [3:0] d;
output [3:0] q;
reg [3:0] q;
always @ (posedge c or negedge r)
if (~r) q = 4'b0000;
else if (l) q = d; else q = q + 1;
endmodule
```


В примере 12.3 описан счетчик с заданным модулем счета. В качестве входного сигнала указан 4-разрядный код k . Оператор *if* выполняет сброс счетчика при достижении равенства $q = k$. Заметим, условие равенства обозначают два знака «равно». При этом модуль счета, определяемый количеством состояний счетчика, с учетом нулевого состояния равен $k + 1$.

```
//Пример 12.3. счетчик с заданным модулем счета
module v123_cnt3 (q, k, c);
input c;
input [3:0] k;
output [3:0] q;
reg [3:0] q;
always @(posedge c)
if (q==k) q = 4'b0;
else q = q + 1;
endmodule
```

12.6. Распределитель импульсов

Распределитель импульсов — схема, содержащая счетчик и дешифратор. Распределители импульсов используют для формирования различных импульсных сигналов с заданными параметрами, в качестве которых могут быть заданы период повторения сигнала и интервалы времени, которым соответствуют единичные и нулевые значения сигнала. Выходной сигнал, или совокупность сигналов, могут быть заданы в виде временных диаграмм.

Пусть задана временная диаграмма (рис. 12.9), в соответствии с которой выходной сигнал y изменяет свое значение в моменты времени, соответствующие фронтам импульсов сигнала синхронизации c . Интервалы времени, для которых сигнал y равен 1 или 0, заданы в виде количества периодов частоты синхронизации. Необходимо разработать распределитель импульсов, формирующий заданный сигнал.

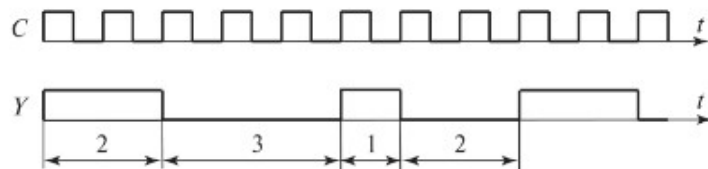


Рис. 12.9. Импульсный сигнал

В соответствии с заданной диаграммой период повторения импульсов равен 8 периодам тактовой частоты, поэтому потребуется

Q_2	Q_1	Q_0	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

счетчик, содержащий 3 триггера. Для синтеза схемы дешифратора составим таблицу истинности, которая содержит 8 строк. Периоды сигнала синхронизации на временной диаграмме необходимо пронумеровать и в таблице истинности учитывать, что заданный сигнал равен единице в нулевом, первом и пятом тактах. По таблице истинности (см. табл. 12.1), составим логическую функцию, которая может быть минимизирована:

$$Y = \bar{Q}_2\bar{Q}_1\bar{Q}_0 \vee \bar{Q}_2\bar{Q}_1Q_0 \vee Q_2\bar{Q}_1Q_0 = \bar{Q}_2\bar{Q}_1 \vee \bar{Q}_1Q_0 = \bar{Q}_1(\bar{Q}_2 \vee Q_0).$$

По уравнениям составляется схема рис. 12.10.

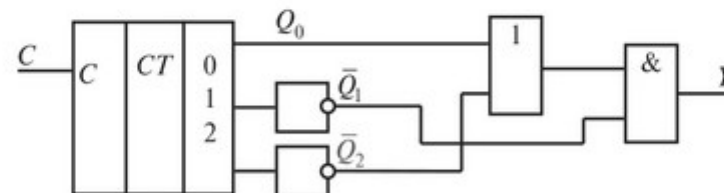


Рис. 12.10. Схема распределителя импульсов

В описаниях распределителей импульсов на языке Verilog можно выделить две части. Одна из них описывает счетчик, который может иметь произвольный модуль счета с использованием последовательных операторов. Вторая часть описывает дешифратор посредством параллельных операторов.