

## Двоичный сумматор.

Таблица истинности логической функции сложения двух одноразрядных двоичных чисел уже была рассмотрена ранее в главе 2, здесь она приведена повторно в табл.3.3.

Таблица 3.3. - Сложение двух одноразрядных чисел.

CIN	A	B	CO	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

В этой таблице CIN– бит входного переноса в этот разряд. В результате сложения получаем сумму S для этого разряда и бит переноса CO в следующий разряд. Первый каскад многоразрядного сумматора, складывающий младшие разряды чисел, не содержит входа переноса CIN. Этот каскад называется **полусумматором**. В результате сложения двух одноразрядных двоичных чисел может появиться число от 0 до 2. Для записи этого результата необходимо два бита. Младший бит суммы называют **полусуммой HS**, а старший бит – **переносом CO в старший разряд**. Ниже символ  $\oplus$

означает логическую операцию XOR - Исключающее ИЛИ .

$$HS = A \oplus B = A \cdot \bar{B} \vee \bar{A} \cdot B;$$

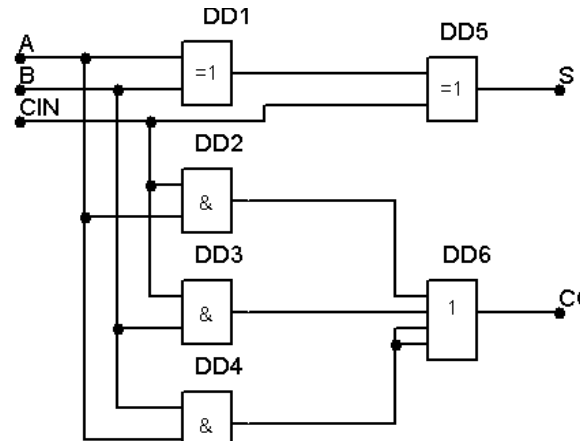
$$CO = A \cdot B;$$

Остальные каскады многоразрядного сумматора имеют вход переноса CIN и называются **полными сумматорами**. Сумма трех одноразрядных чисел может принимать значения от 0 до 3. Поэтому для представления результата достаточно два двоичных разряда S (сумма) и CO – перенос в старший разряд.

$$S = A \oplus B \oplus C_{IN};$$

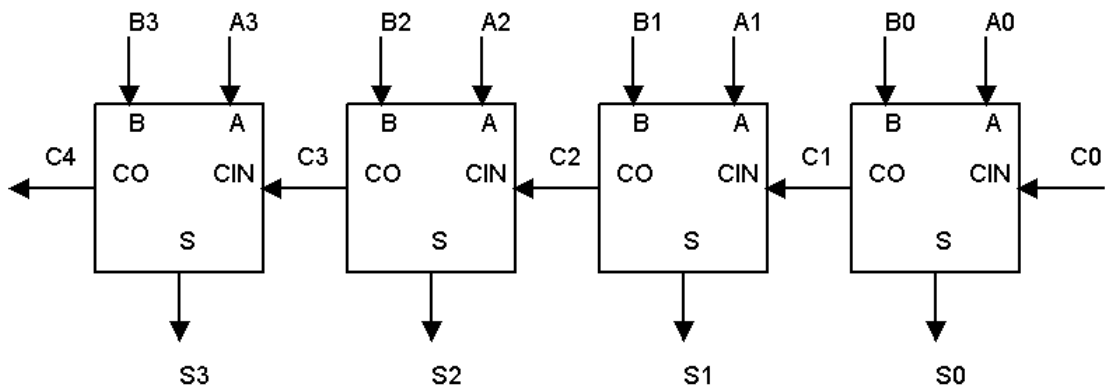
$$CO = A \times B \cup A \&C_{IN} \cup B \&C_{IN};$$

Схема каскада полного одноразрядного сумматора приведена на рисунке 3.16.



**Рис. 3.16.** Схема каскада полного одноразрядного сумматора.

Схема многоразрядного сумматора с последовательным переносом приведена на рисунке 3.16:



**Рис. 3.17.** Многоразрядный сумматор с последовательным переносом.

Сумматор с последовательным переносом обладает малым быстродействием. Время суммирования практически пропорционально числу разрядов сумматора. Значительно меньшими задержками обладают сумматоры с параллельным переносом.

### 3.6.1. Многоразрядный сумматор с параллельным переносом.

Многоразрядный сумматор с параллельным переносом реализует очень простую идею:

1. В каждом двоичном разряде складываются только два входных числа  $A_i$  и  $B_i$ .
2. Сигналы переноса  $C_i$  вырабатываются одновременно и независимо при помощи специальной комбинационной схемы ускорения переноса  $CR_i$  (рис. 3.18).

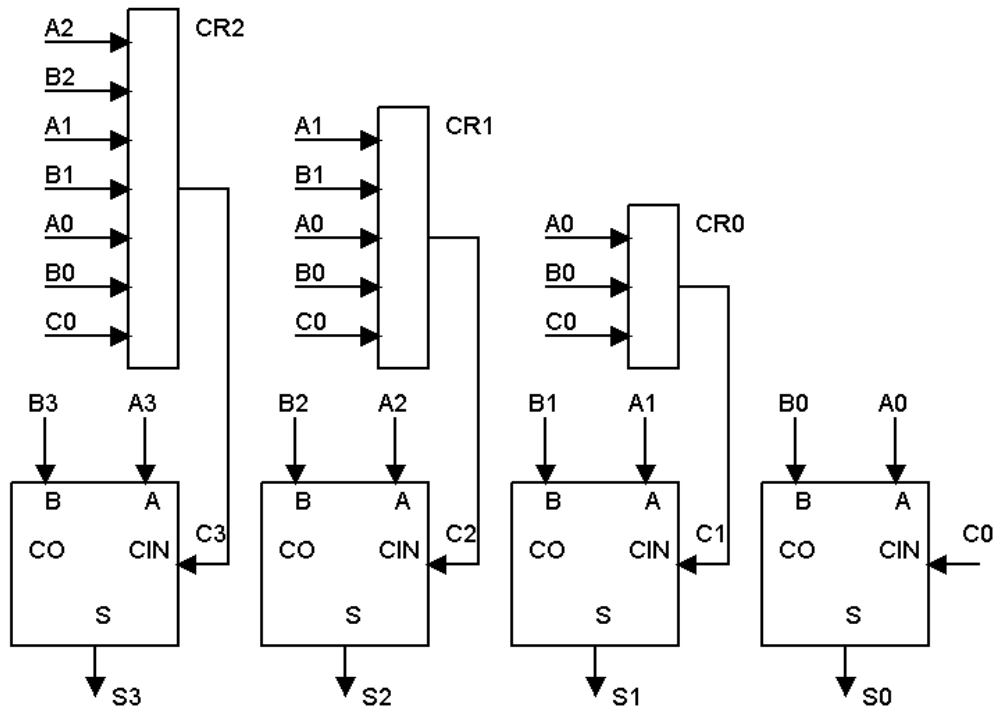


Рис. 3.18. Многоразрядный сумматор с параллельным переносом.

Определим следующие логические функции:

1. Функция генерации переноса  $G_i$ .

Если  $A_i$  и  $B_i$  равны единице, то на выходе разряда сумматора появляется сигнал переноса, не зависящий от сигнала  $CIN_i$ . Поэтому

$$G_i = A_i \cdot B_i;$$

2. Функция распространения  $H_i$ .

$$H_i = A_i \oplus B_i;$$

Если только одна из переменных  $A_i$  или  $B_i$  равна единице, то сигнал переноса на выходе разряда сумматора появляется только при наличии сигнала  $CIN_i$ .

В этой формуле функцию "Исключающее ИЛИ" можно заменить на "ИЛИ", так как при  $A = 1$  и  $B = 1$  сигнал переноса формируется функцией  $G_i$ .

$$H_i = A_i \cup B_i;$$

Рекуррентная функция формирования сигнала переноса на выходе разряда имеет следующий вид

$$C_i = G_i \cup H_i \times C_{(i-1)};$$

Для нулевого разряда сумматора

$$C_1 = G_0 \cup C_0 \times H_0;$$

Для первого разряда

$$C_2 = G_1 \cup C_1 \times H_1 = G_1 \cup G_0 \times H_1 \cup C_0 \times H_1 \times H_0;$$

Эти логические функции позволяют построить схемы формирования сигналов переноса в базисе И – ИЛИ. Однако лучшие результаты получаются в базисе И – ИЛИ – НЕ. Получим

$$C_0 = \overline{A_0 \cdot B_0 \cdot C_0 \cdot H_0};$$

$$C_1 = \overline{A_1 \cdot B_1 \cdot A_0 \cdot B_0 \cdot H_1 \cdot C_0 \cdot H_1 \cdot H_0};$$

$$C_2 = \overline{A_2 \cdot B_2 \cdot A_1 \cdot B_1 \cdot H_2 \cdot A_0 \cdot B_0 \cdot H_2 \cdot H_1 \cdot C_0 \cdot H_2 \cdot H_1 \cdot H_0};$$

Схема трех первых каскадов сумматора с параллельным переносом приведена на рис. 3.19.

Время сложения в сумматоре с параллельным переносом равно времени сложения в сумматоре  $t_{\text{СУМ}}$  и времени формирования сигнала переноса  $t_{\text{CR}}$ . Время  $t_{\text{CR}} = 3t_{\text{ЛЭ}}$  и не зависит от разрядности сумматора.

Однако с ростом числа разрядов в схеме квадратично возрастает число элементов И-НЕ и число их входов. Это приводит к росту нагрузки в схеме и, как следствие, к увеличению задержек сигналов в сумматоре. Рациональная разрядность таких сумматоров 5 – 8.

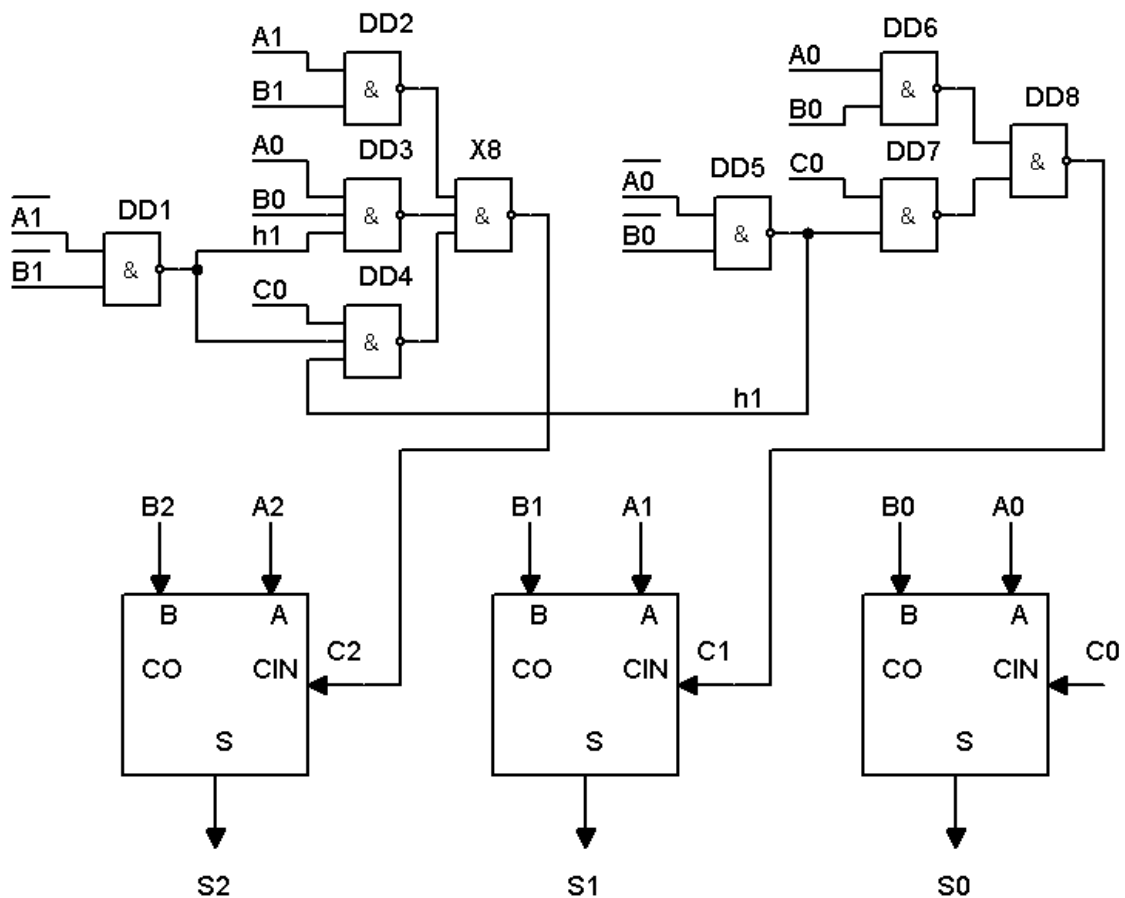


Рис. 3.19. Схема сумматора с параллельным переносом.

### 3.6.3. Описание сумматора на VHDL.

При описании схемы сумматора на VHDL необходимо учитывать, что над типом сигналов `STD_LOGIC_VECTOR` определены **только логические операции** и операции сравнения.. С этими типами данных арифметические операции "+" и "-" **не работают**.

Арифметические операции и функции, позволяющие работать с ними, включены в пакеты `IEEE.STD_LOGIC_ARITH` и `IEEE.STD_LOGIC_NUMERIC`. Поэтому при описании схемы сумматора необходимо подключение подобного пакета.

```
library IEEE; [1][2]
use IEEE.STD_LOGIC_1164.all; `
use IEEE.STD_LOGIC_ARITH.all;
entity add_1 is
    port( A : in STD_LOGIC_VECTOR(7 downto 0);
          B : in STD_LOGIC_VECTOR(7 downto 0);
```

```

        C : out STD_LOGIC_VECTOR(8 downto 0) );
end add_1;

architecture add_1_ar of add_1 is
    signal A1, B1 : SIGNED (7 downto 0); -- сигналы типа SIGNED
    signal C1 :SIGNED (8 downto 0); -- сигналы типа SIGNED
begin
    A1 <= SIGNED(A); -- преобразование типов данных
    B1 <= SIGNED(B); -- преобразование типов данных
    C1 <= ('0'&A1) + ('0'&B1); -- добавление слева к двоичным
        -- числам A1 и B1 нулевых разрядов и
        -- сложение двоичных чисел
    C <= STD_LOGIC_VECTOR(C1); -- обратное преобразование
        -- типов данных
end add_1_ar;

```

## 4. ЦИФРОВЫЕ УЗЛЫ ПОСЛЕДОВАТЕЛЬНОСТНОГО ТИПА.

### 4.1. Определения.

**Последовательностной** называется логическая схема, выходные сигналы которой определяются не только текущими значениями входных сигналов, но и их предыдущими значениями. Это означает, что такие схемы имеют память.

**Состояние последовательностной** схемы – это совокупность состояний ее элементов памяти, значения которых в данный момент времени содержат всю информацию о прошлом, достаточную для описания работы схемы в будущем.

#### Автоматы

**Понятие автомата** является обобщенным математическим представлением поведения последовательностной схемы.

#### Синхронные и асинхронные автоматы..

Если изменение состояний автомата проводится в определенные моменты времени, задаваемые внешним тактовым сигналом (синхросигналом), то мы имеем дело с **синхронным автоматом**. Если оно может происходить в любой момент времени, то имеем дело с **асинхронным автоматом**.

#### Конечные и бесконечные автоматы

По числу возможных состояний автомата различают **конечные (FSM-Finite State Machine) автоматы**, у которых множество состояний конечно и **бесконечные автоматы (ISM-Infinite State Machine)**, у которых бесконечное множество состояний.

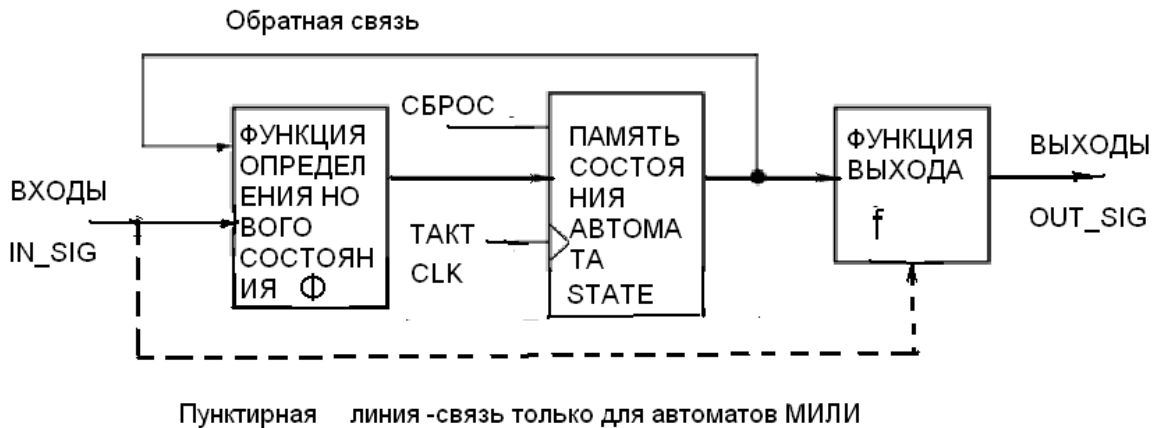
#### Автоматы Мили и Мура

Известны два типа конечных автоматов (FSM-Finite State Machine): **автомат Мура (Moore) и автомат Мили (Mealy)**.

На рисунке представлена блок-схема автомата. В общем случае автомат состоит из комбинационной схемы, вычисляющей новое состояние автомата как функцию значений входных сигналов и состояния памяти автомата, блока памяти, хранящего состояние автомата и блока вычисления значений выходных сигналов.

В автомате Мура выходные сигналы (**out\_sig**) зависят (**f-функция выходов**) только от текущего состояния (**state**) автомата.

В Автомате МИЛИ на выходные сигналы влияет также текущее значение входных сигналов.



Блок-схема автомата

В автомате Мили выходы являются функцией (**f- функция выходов**) как текущего состояния, так и входных сигналов (**in\_sig**), что обозначено пунктирной линией. Состояние автомата изменяется по синхросигналу (**C-clk**). **Φ-функция**, вычисляющая новое состояние автомата (функция переходов, функция возбуждения, функция входов) на базе старого состояния и значений входных сигналов. Ниже в таблице на псевдокоде представлены описания трех частей автомата.

Обобщенное представление описаний трех частей автомата

Автомат Мура	Автомат Мили
<b>out_sig=f (state);</b>	<b>out_sig=f (state, in_sig);</b>
<b>nexstate=Φ (in_sig, state);</b>	<b>nexstate=Φ (in_sig, state);</b>
<b>state = nexstate when posedge of clk;</b>	<b>state = nexstate when posedge of clk;</b>

Классическими средствами описаний конечных автоматов являются **таблицы переходов и таблицы выходов** ( иногда эти две таблицы совмещают в одну- **совмещенную таблицу переходов и выходов**), а также **граф- схемы или диаграммы состояний автоматов**.

Эти средства ниже иллюстрируются на примере простейшего автомата Мура, так называемого D-триггера. Его УГО представлено на рисунке.



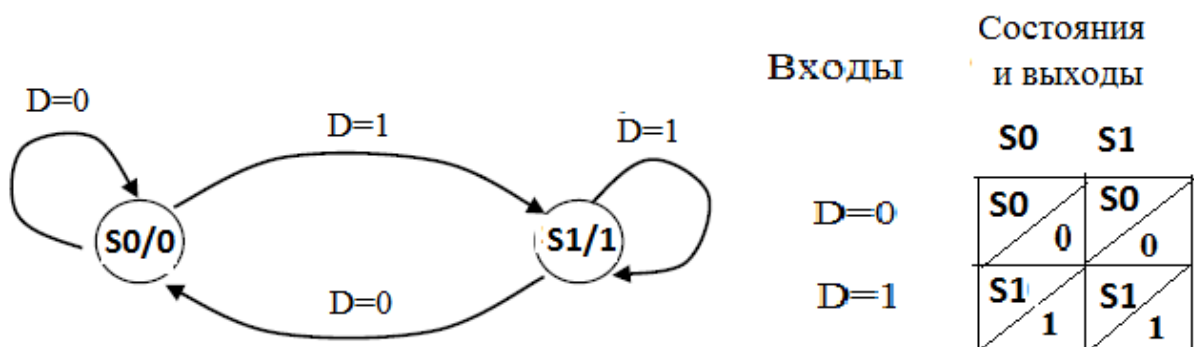


УГО D-триггера, слева- по ЕСКД, справа зарубежное

Этот автомат имеет два состояния : S0 и S1. Его входные сигналы – сигнал данных D и синхросигнал C , выходной сигнал Q. Косая черта у сигнала C (вариант ЕСКД) означает, что активным является фронт сигнала (перепад из 0 в 1).

**Граф-схема автомата Мура**, задающая D – триггер, представлена ниже на рисунке слева. Дуги направленного графа обозначают переходы из одного состояния в другое и помечены значениями входного сигнала D. При соответствующем значении D и наличии фронта синхросигнала C автомат переходит в соответствующее состояние. Состояния автомата представлены вершинами графа (кружочками). Внутри каждой вершины указано состояние автомата ( S0 ,S1) и после косой черты - выходной сигнал, соответствующий этому состоянию.

**Совмещенная таблица переходов и выходов автомата D – триггера** представлена на рисунке справа. Сбоку слева указаны значения входа D. Над таблицей указаны старые состояния автомата. В клетках таблицы указаны новые состояния автомата и после косой черты - выходные сигналы автомата



Граф схема автомата D-триггера и совмещенная таблица переходов и выходов

Другим способом задания работы автомата, в примере D-триггера, является **матрица переходов** (ниже рисунке слева), где в первом столбце

значения входа D, в столбце Qt-1 – состояние триггера на предыдущем такте; Qt – в текущем. Состояние автомата S0 закодировано как 0, состояние S1-как 1.

D	Qt-1	Qt
0	0	0
1	0	1
0	1	0
1	1	1

C	D	Qt-1	Qt
$\uparrow$	D	X	D
1	X	Qt-1	Qt-1
0	X	Qt-1	Qt-1

Матрица переходов автомата D-триггера (слева) и ее обычная форма представления (справа) в справочниках по микросхемам

В отечественных справочниках по микросхемам обычно приводится УГО D-триггера и другая форма матрицы переходов (рисунке справа), в которой представлены входной сигнал D, синхросигнал C и выходной сигнал, совпадающий с состоянием триггера Q. Для обозначения фронта синхросигнала C (перехода из 0 в 1) используется  $\uparrow$ , для обозначения произвольного значения сигнала символ X. Обозначение Qt-1 применяется для столбца таблицы со старыми состояниями триггера и совпадающего с ним выходного сигнала, обозначение Qt- для нового состояния и совпадающего с ним значения выходного сигнала.

Применительно к D-триггеру три компонента описания автомата превращаются в такие упрощенные формулы:

**out\_sig=Qt;** // выходной сигнал равен состоянию D-триггера

**nexstate =D;** // вход D определяет будущее состояние D-триггера

**Qt = nexstate when posedge ( C) else Qt-1;**// состояние D- триггера изменяется только по фронту синхросигнала C.

Еще одним способом описаний автоматов является их описание на HDL.В примере дан один из вариантов HDL- описания D-триггера, в котором три компонента описания автомата в таком простом случае заменены одним оператором установки выходного сигнала Q и тестирующей программы. Состояние автомата изменяется только по фронту синхросигнала C ( на VHDL обозначается как - rising\_edge ( C ), а иначе оно остается старым.

Один из вариантов VHDL- описания D-триггера

```

library IEEE; use IEEE.std_logic_1164.all; [1][2]
entity DFF is
port (D, C: in std_logic; Q: out std_logic);
end DFF;

architecture BEH_ of DFF is begin
Q<=D when rising_edge ( C ) else Q;
end ;

```

## 4.2. Триггеры.

**Триггер** – логический элемент с двумя устойчивыми состояниями. Состояние триггера определяется логическим уровнем выходного сигнала на прямом выходе (лог. "0" или лог. "1"). Переключение триггера из одного состояния в другое происходит под действием входных сигналов.

Триггер содержит один или несколько входов, обозначаемых символами R, S, K, J, T, D, C, V, и выход, обозначаемый символом Q.

Классификация большого числа разработанных к настоящему времени видов триггеров проводится в основном по типу информационных входов и по динамическим характеристикам.

### 1. Классификация по типу информационных входов:

*Типы информационных входов:*

*R* – вход сброса ( $Q = 0$ );

*S* – вход установки ( $Q = 1$ );

*K* – вход сброса триггера ( $Q = 0$ );

*J* – вход установки триггера ( $Q = 1$ );

*T* – счетный вход;

*D* – информационный вход, переключение триггера в состояние, соответствующее логическому уровню на этом входе;

*C* – управляющий (синхронизирующий, стробирующий[*уточнить!*]) вход;

*V* – вход для блокирования триггера с целью сколь-угодно длительного хранения ранее записанной информации.

В соответствии с типами входов, триггеры называют:

*RS* – триггер;

*T* – триггер;

*D* – триггер;

*JK* – триггер;

DV – триггер;  
TV – триггер.

## 2. Классификация по динамическим характеристикам:

По моменту реакции на входной сигнал триггеры подразделяются на:

– *Асинхронные* – изменяют свое состояние непосредственно в момент изменения сигнала на его информационных входах ( другое название таких триггеров- триггер-защелка);

– *Синхронные* – меняют свое состояние лишь в определенные ( тактовые) моменты времени. Моменты изменения состояния триггера соответствуют активному значению сигнала на синхронизирующем входе С.

Синхронные триггеры по виду активного логического сигнала на С- входе триггера подразделяются на:

– управляемые уровнем синхронизирующего сигнала или триггеры с потенциальным управлением;

– управляемые фронтом или спадом синхронизирующего сигнала или триггеры с динамическим управлением;

– двухступенчатые триггеры или триггеры с импульсным управлением.

### 4.2.1 Асинхронный RS-триггер-защелка.

Асинхронным RS триггером называется следующая схема, собранная на двух логических элементах И-НЕ или ИЛИ-НЕ и охваченная положительной обратной связью. Сигналы на входы асинхронного триггера могут приходить в произвольные моменты времени.

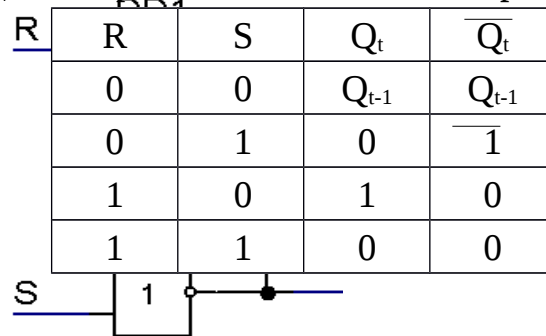


Рис. 4.1 Асинхронный RS-триггер на элементах ИЛИ-НЕ.

Исследуем работу триггера.

1. Пусть на вход S DD2 подается сигнал 1, а на вход R DD1 – сигнал 0.

Тогда на выходе  $\overline{Q}$  появится сигнал 0, а на выходе Q – сигнал 1. Если теперь сигнал 1 с входа S снять, то схема останется в прежнем состоянии за счет цепей обратной связи, при этом на прямом выходе Q остается сигнал 1. Это установка триггера.

2. Пусть на вход R DD1 подается сигнал 1, а на вход S DD2 – сигнал 0.

Тогда на выходе Q появится сигнал 0, на выходе  $\overline{Q}$  – сигнал 1. . Если теперь

сигнал 1 с входа R снять, то схема останется в прежнем состоянии за счет цепей обратной связи, при этом на прямом выходе Q остается сигнал 0. Это сброс триггера.

3. Если сигналы ни на один из входов не подаются, то триггер сохраняет текущее состояние.

4. Логические элементы триггера переключаются последовательно. Реальные логические элементы имеют задержку распространения сигнала ( $T_{pd\_elem}$ ). Поэтому возможны малые промежутки времени, в которых логические элементы находятся в одинаковом состоянии.

5. Время установления RS-триггера равно

$$T_{pd\_rs} = 2 * T_{pd\_elem}$$

Если одновременно подать на входы сигналы  $R = 1$  и  $S = 1$ , то выходы окажутся в состоянии  $Q = \bar{Q} = 0$ . Такая комбинация входных сигналов  $R = 1$  и  $S = 1$  является запрещенной для RS-триггера на элементах ИЛИ-НЕ. Запрет означает лишь то, что если после входной комбинации значений сигналов  $R = 1$  и  $S = 1$  подать комбинацию  $R = 0$  и  $S = 0$ , то в триггере возникают так называемые **гонки** сигналов и состояние, в которое триггер перейдет, определяется соотношением задержек в его элементах. Интересно, что так как при моделировании эти задержки считаются равными, то состояние триггера все время будет меняться, что при моделировании схем без учета задержек может вызвать останов моделирования. А при моделировании с учетом задержек - генерацию последовательностей состояния выходов триггера 00-11-00-11 и т.д..

Временная диаграмма работы RS-триггера элементах ИЛИ-НЕ приведена на рис. 4.2.

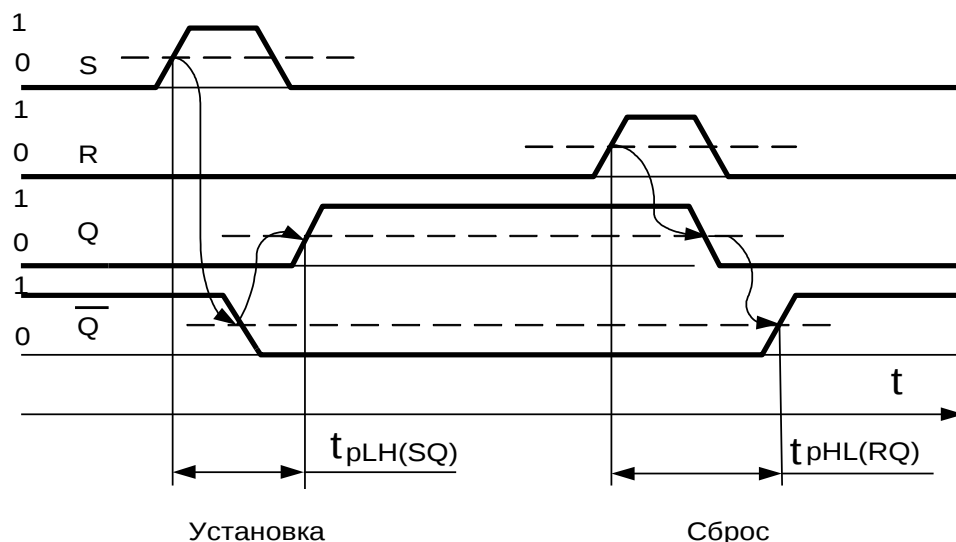


Рис. 4.2. Временная диаграмма RS-триггера-защелки элементах ИЛИ-НЕ.

Важным временным параметром цифровой схемы, как уже отмечалось, является *Время задержки распространения сигнала* ( $T_{pd-propagation\ delay\ time}$ ). Это время, необходимое для изменения выходного сигнала после изменения входного сигнала. На рис. 4.2 обозначены два времени задержки :

$t_{pLH(SQ)}$  – время, необходимое для переключения от низкого уровня сигнала к высокому, и

$t_{pHL (RQ)}$  – время, необходимое для переключения от высокого логического уровня сигнала к низкому. Эти времена задержки могут быть не равны между собой.

Сигналы, поступающие на входы RS-триггера, должны иметь достаточную длительность. Если длительность входных сигналов меньше заданного в справочниках минимального значения  $t_{p(min)}$  (обычно это время близко к времени задержки распространения), то переходный процесс может не дойти до конца, и работа схемы будет нарушена.

#### 4.2.2 Описание асинхронного RS-триггера-защелки на VHDL.

Ниже представлены два варианта описания RS-триггера-защелки.

Первый вариант прямо соответствует схеме рис.4.1, второй иллюстрирует процессный стиль описания триггера и несколько искажает его поведение т.к. допускает одновременное появление одинаковых значений на выходах- или двух 0, или двух 1. [зачем он нужен?]

Первый вариант- описание схемы RS- триггера на элементах ИЛИ-НЕ.

```
library IEEE; [2][1]
use IEEE.STD_LOGIC_1164.all;
entity RS_Latch is port
(S,R : in STD_LOGIC; Q, Qbar : out STD_LOGIC := '0');
end ;

architecture SXEMA of RS_Latch is
begin
Q<= not (S OR Qbar) ;
Qbar<= not (R OR Q);
end ;
```

Второй вариант –описание RS - триггера функциональное

```
library IEEE; [1][2]
use IEEE.STD_LOGIC_1164.all;
entity RS_Tr is
port( S : in STD_LOGIC;
R : in STD_LOGIC;
Q : out STD_LOGIC;
Qbar : out STD_LOGIC );
end RS_Tr;

architecture BEH of RS_Tr is
begin
process (R,S)
begin
if (S ='1') and (R = '0') then
Q <= '1'; Qbar <= '0';
end if;
```

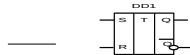
```

if (R = '1') and (S = '0') then
    Q <= '0'; Qbar <= '1';
end if;
end process;
end ;

```

. Переключение триггера происходит при изменении сигналов R или S и при возникновении "разрешенной" комбинации, перечисленной в условных операторах if.

#### 4.2.3 Асинхронный триггер-защелка на элементах И-НЕ.



RS-триггер может быть построен на элементах И-НЕ.

$\overline{R}$	$\overline{S}$	R	S	$Q_t$	$\overline{Q}_t$
1		1	1	$Q_{t-1}$	$\overline{Q}_{t-1}$
1		1	0	0	1
1		0	1	1	0
1		0	0	1	1

a)

б)

в)

a) схема, б) таблица функционирования, в) УГО

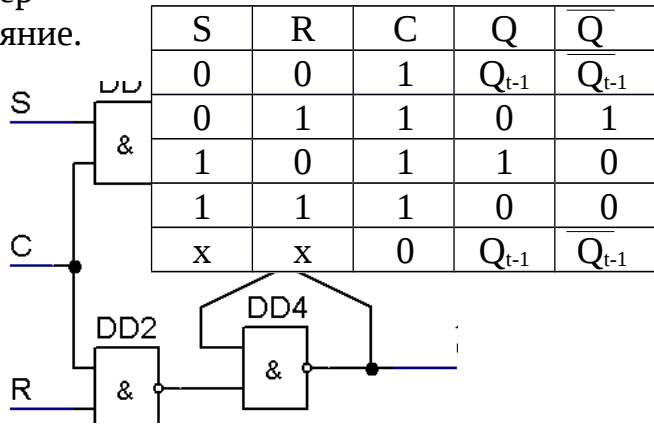
**Рис. 4.3. Асинхронный триггер –защелка на элементах И-НЕ.**

Активными уровнями сигналов R и S на входе являются низкие уровни. Запрещенным состоянием является  $R = 0$  и  $S = 0$ . По такой схеме выпускается отечественная микросхема RS-триггера типа TP2

#### 4.2.4. Синхронный RS-триггер.

Для синхронизации переключений триггера вводится дополнительный сигнал синхронизации  $C$ , активный уровень которого разрешает переключение триггера в соответствии с входными сигналами  $R$  и  $S$ .

Эта схема при  $C = 1$  ведет себя как обычный RS-триггер. При  $C = 0$  триггер сохраняет предыдущее состояние.

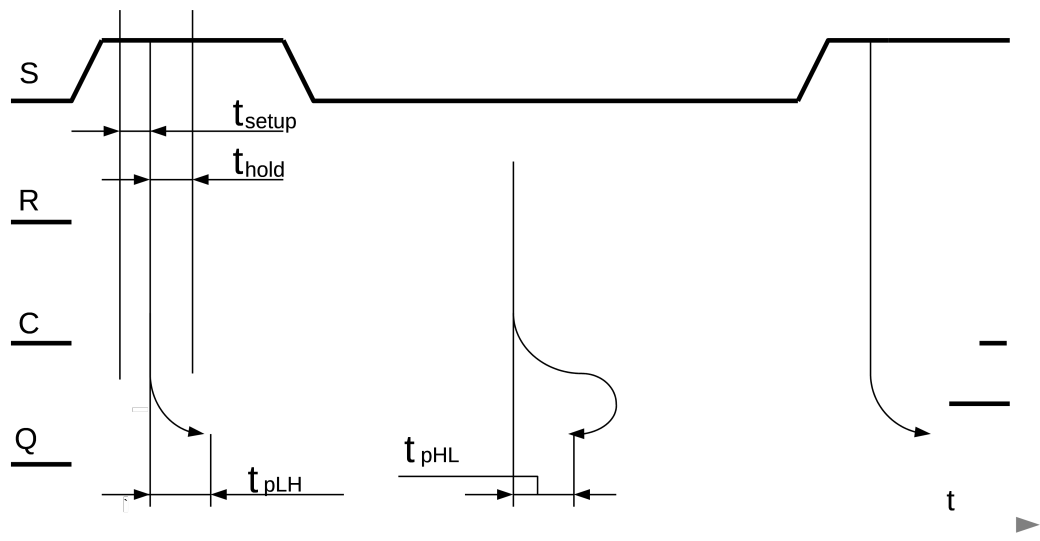


S	R	C	Q	$\overline{Q}$
0	0	1	$Q_{t-1}$	$\overline{Q_{t-1}}$
0	1	1	0	1
1	0	1	1	0
1	1	1	0	0
x	x	0	$Q_{t-1}$	$\overline{Q_{t-1}}$

а) схема б) таблица функционирования

**Рис. 4.4. Синхронный RS-триггер.**

Временная диаграмма работы синхронного RS-триггера показана на рисунке 4.5.



**Рис. 4.5. Временная диаграмма работы синхронного RS-триггера.**

Отсчет моментов времени на временной диаграмме проводится от середины фронтов импульсов. Если сигналы на входах  $S = 1$  и  $R = 0$ , то через время  $t_{pLH}$  на выходе  $Q$  появится сигнал  $Q = 1$ .

Минимально необходимое для правильной работы триггера время упреждения подачи входных сигналов перед появлением синхроимпульса называется **временем предустановки** ( $t_{setup}$ ). Минимальное время, в течение которого эти сигналы еще должны удерживаться после фронта синхросигнала называется **временем удержания** ( $t_{hold}$ ).



#### 4.2.5 Описание синхронного RS-триггера на VHDL.

```
library IEEE; [1][2]
use IEEE.STD_LOGIC_1164.all;
entity RSC_Tr is
    port( S : in STD_LOGIC;
          C : in STD_LOGIC;
          R : in STD_LOGIC;
          Q : out STD_LOGIC;
          Qbar : out STD_LOGIC);
end RSC_Tr;

architecture RSC_Tr_arc of RSC_Tr is
begin
    process(C)
    begin
        if(C = '1') and (R = '1') and (S = '0') then
            Q <= '0'; Qbar <= '1';
        elsif (C = '1') and (R = '0') and (S = '1') then
            Q <= '1'; Qbar <= '0';
        end if;
    end process;
end RSC_Tr_arc;
```

Переключение триггера происходит при появлении сигнала  $C = '1'$ . "Разрешенная" комбинация сигналов  $R$  и  $S$  должна появиться на входах триггера раньше сигнала  $C$ .

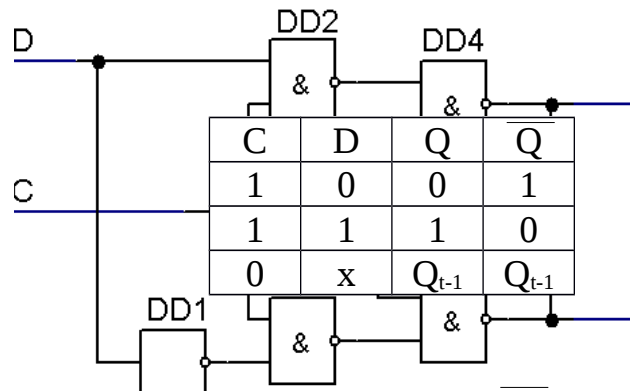
Если на входах триггера "разрешенная" комбинация сигналов не появляется или появляется "не разрешенная" комбинация, то сигнал  $C = '1'$  не изменяет состояние триггера.

#### 4.2.6. D-триггер с управлением уровнем синхросигнала (защелка).

Для хранения одного бита информации, поступающего по отдельной сигнальной линии, применяется D-триггер. Его схема приведена на рис. 4.6.

Когда подан синхросигнал  $C$  высокого уровня, D-триггер "прозрачен", так как сигнал на выходе  $Q$  повторяет сигнал на входе  $D$  с учетом задержки распространения. Если приходит сигнал  $C$  низкого уровня, то D-триггер переходит в режим хранения и сохраняет на выходе последнее значение сигнала  $D$ .

Триггер "прозрачен" все время, пока действует синхросигнал.



а)

б)

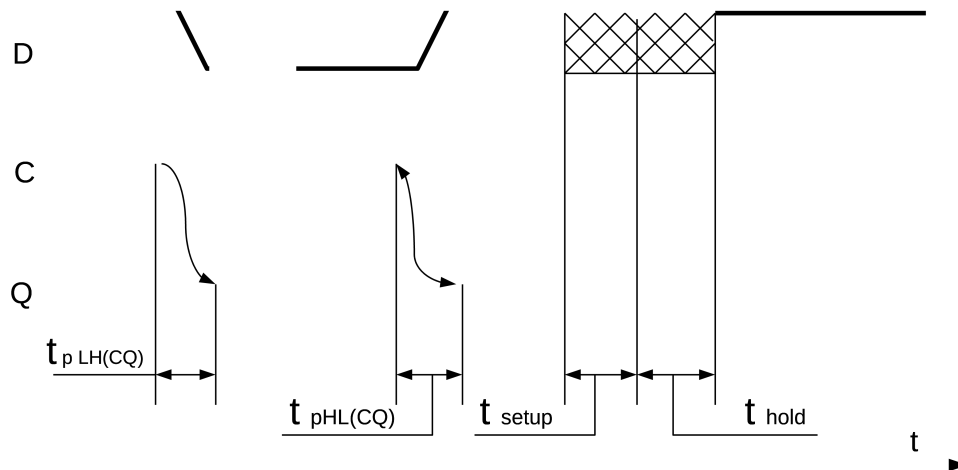
а) схема D-триггера,

б) таблица функционирования.

**Рис. 4.6. D-триггер (D-защелка).**

Временная диаграмма работы D-триггера-защелки представлена на рис. 4.7. Если сигнал на входе  $D = 1$ , то после появления сигнала  $C = 1$  через время  $t_{pLH(CQ)}$  на выходе  $Q$  появится сигнал  $Q = 1$ .

Поведение D-триггера в районе спада сигнала  $C$  имеет особенности. В это время сигнал на входе  $D$  не должен меняться. Этот интервал времени начинается за время установки  $t_{setup}$  до спада сигнала  $C$  и заканчивается через время удержания  $t_{hold}$  после спада сигнала  $C$ . Если сигнал  $D$  изменяется в течение этого интервала, то последующее состояние триггера оказывается непредсказуемым. Это может привести к сбою в работе цифровой схемы.



**Рис. 4.7. Временная диаграмма работы D-триггера (защелки).**

Асинхронные триггеры и триггеры с синхронизацией по уровню синхросигнала (триггеры с потенциальным управлением) являются наиболее простыми. Поэтому они широко применяются в качестве базовых элементов в современных заказных интегральных схемах. Недостатки, присущие этим триггерам, устраняются средствами САПР.

#### 4.2.7. Описание синхронного D-триггера (защелки) на VHDL.

```
library IEEE; [1][2]
use IEEE.STD_LOGIC_1164.all;
entity D_Tr_1 is
    port( C : in STD_LOGIC;
          D : in STD_LOGIC;
          Q : out STD_LOGIC;
          Qbar : out STD_LOGIC );
end D_Tr_1;
architecture D_Tr_1_arc of D_Tr_1 is
begin
    process(C)
    begin
        if (C = '1') then
            Q <= D;
            Qbar <= not D;
        end if;
    end process;
end D_Tr_1_arc;
```

#### 4.2.8. Триггеры с динамическим управлением.

В микросхемах средней степени интеграции широко применяются триггеры с динамическим управлением. Такой триггер изменяет свое состояние только в малой окрестности фронта синхросигнала. Промышленность выпускает различные типы микросхем триггеров с динамическим управлением, схемы которых близки между собой.

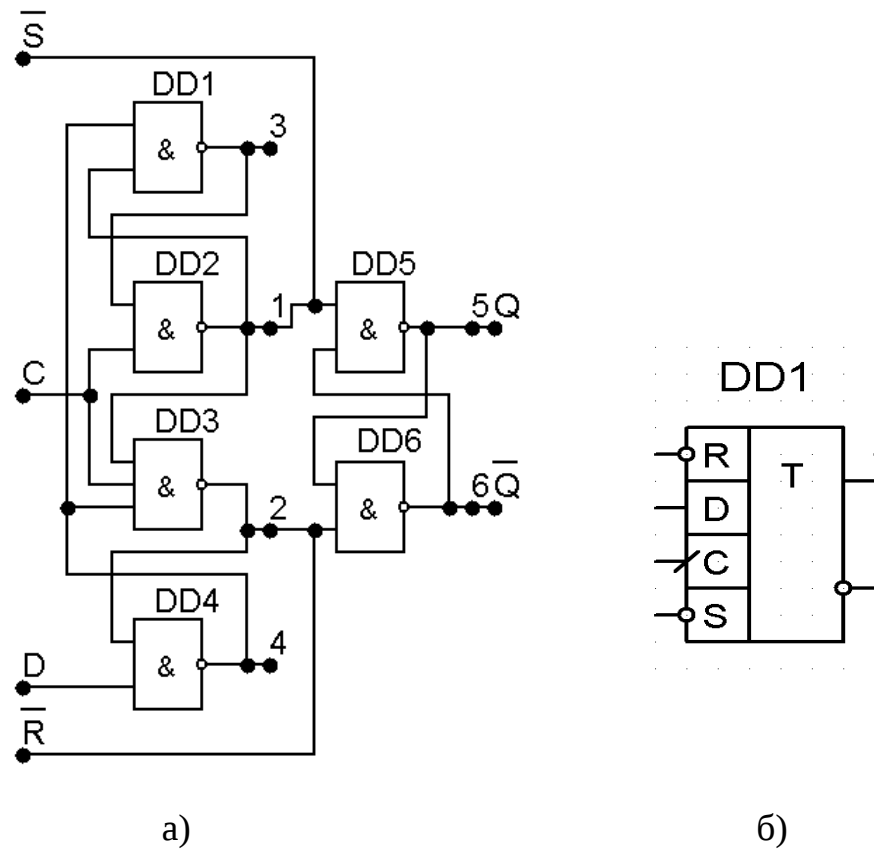
Схема и УГО шестиэлементного D-триггера с динамическим управлением приведены на рис. 4.8, а временная диаграмма его работы – на рис. 4.9.

На временной диаграмме названия графиков совпадают с обозначениями контрольных точек на выход логических элементов.

В этой схеме хранение информации происходит в выходном RS-триггере (элементы DD5 и DD6) с инверсными сигналами на входах.

Управление схемой осуществляют еще два RS-триггера (элементы DD1 и DD2, DD3 и DD4).

Логические элементы DD2 и DD3 управляют состоянием выходного RS-триггера, собранного на логических элементах DD5 и DD6. Если  $C = "0"$ , то логические элементы DD2 и DD3 закрыты и выходной RS-триггер находится в режиме хранения. Это отражено на временной диаграмме, где при изменении сигнала D на входе в моменты времени  $t_1$ ,  $t_2$ ,  $t_4$ ,  $t_5$  и  $t_7$  состояние триггера не изменяется (точки 5 и 6).



а) – схема D-триггера, б) – УГО D-триггера типа ТМ2

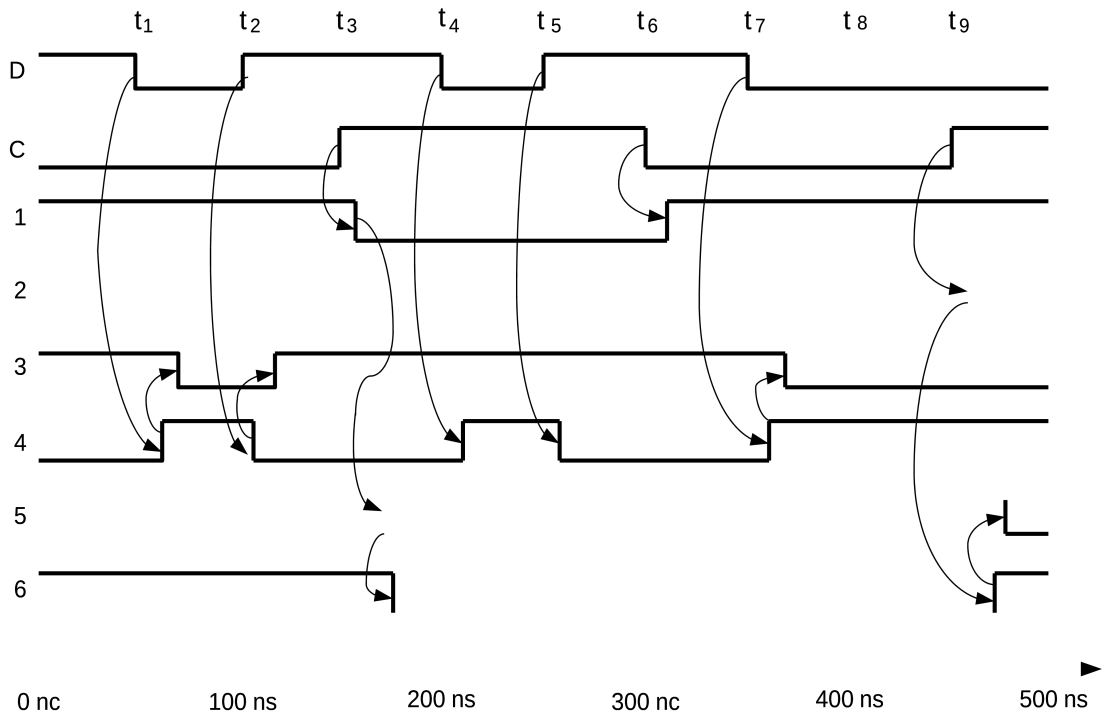
**Рис. 4.8. D-триггер с динамическим управлением.**

В момент времени  $t_3$  ( $D = "1"$ ) происходит переключение триггера. В этот момент времени на входе DD1 (точка 3) – "2", на выходе DD3 (точка 1) – "1", а на выходе DD4 (точка 4) – "0". В результате на входах DD2 образуется "1", а на выходе (точка 1) – "0". В результате DD5 закрывается и на его выходе Q (точка 5) появляется "1". Следом на выходе DD6 Q (точка 6) появляется "0".

В момент времени  $t_9$  по следующему фронту C происходит переключение состояния триггера (точка 5) в состояние "0", соответствующее уровню входного сигнала D в этот момент времени. При этом на входе DD4 (точка 4) – "1", на выходе DD1 (точка 3) – "0", а на выходе DD2 (точка 1) – "1". В результате на входах DD3 образуется единица, а на выходе (точка 2) – ноль. В результате DD6 закрывается и на его выходе Q (точка 6) появляется "1". Следом на выходе DD5 Q (точка 5) появляется "0". В остальные моменты времени, отмеченные на временной диаграмме, триггер не переключается.

Все элементы схемы триггера переключаются последовательно во времени. Время задержки переключения триггера от входа C до выхода равно трем временам задержек в логических элементах  $t_p$ .

$$t_{pCQ} = 3t_p;$$



**Рис. 4.9. Временная диаграмма работы шестиэлементного D-триггера.**

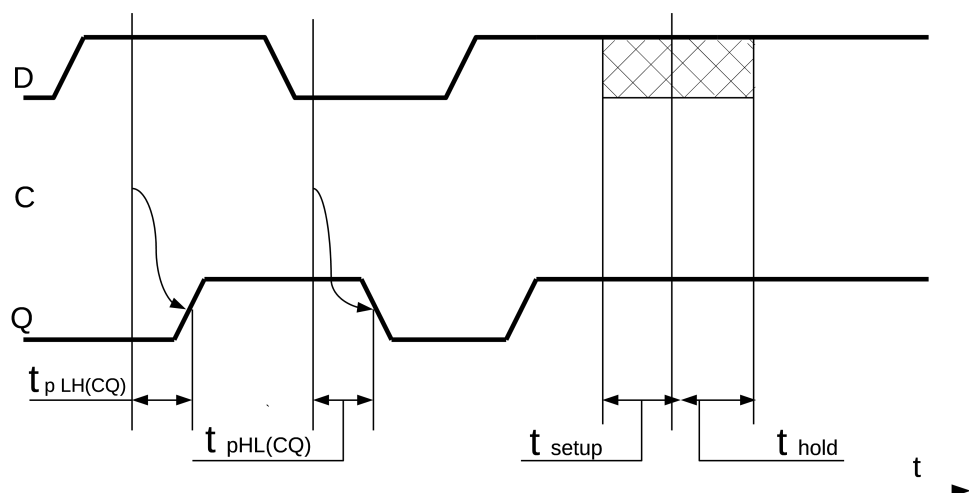
После перехода сигнала C в "0" начинается переход коммутирующих элементов триггера в исходное состояние, что также занимает время

$$t_{pQC} = 3t_p;$$

$$t = 6t_p;$$

Таким образом, общая задержка сигнала в триггере с динамическим управлением равна

Внешняя временная диаграмма работы D-триггера с динамическим управлением представлена на следующем рисунке.



**Рис. 4.10. Временная диаграмма работы D-триггера с динамическим управлением.**

*Замечание.* Сигнал на входе D триггера не должен меняться в окрестности фронта сигнала синхронизации С. В документации на микросхемы триггеров указывается минимальное время предустановки  $t_{\text{setup}}$  и удержания  $t_{\text{hold}}$  сигнала на входе D относительно фронта синхроимпульса С.

#### 4.2.9 Описание D-триггера с динамическим управлением на VHDL.

Приведем описание на VHDL триггера типа ТМ2. Условно-графическое обозначение триггера приведено на рис. 4.8б.

```
library IEEE; use IEEE.std_logic_1164.all;
entity DFFRS_1 is -- описание устройства DFFRS_1
port (D, C, R, S : in std_logic; -- определение входных сигналов
Q, Qbar : out std_logic); -- определение выходных сигналов
end DFFRS_1;

architecture DFFRS_1_arc of DFFRS_1 is -- архитектура DFFRS_1
begin
process (C, R, S) begin
if (S = '0') then -- приоритетная асинхронная установка
Q <= '1' ;
Qbar <= '0' ;
elsif (R = '0') then -- приоритетный асинхронный сброс
Q <= '0' ;
Qbar <= '1' ;
elsif (C'event and C = '1') then -- выделение переднего фронта
Q <= D ; -- назначение Q сигнала, поступившего на вход D
Qbar <= not (D) ; -- назначение Qbar сигнала, поступившего на вход D
end if;
end process;
end DFFRS_1_arc;
```

В описании на VHDL D-триггера типа ТМ2 с асинхронной установкой и сбросом использованы оператор process и условный оператор if. Передний фронт синхроимпульса выделяется при помощи стандартного в VHDL логического выражения (C'event and C = '1'). Триггер переключается при изменении сигналов С, R и S, входящих в список чувствительности оператора process.

Второй вариант VHDL- описания D-триггера и вариант тестирующей программы приведены ниже.

```
library IEEE; use IEEE.std_logic_1164.all;
entity DFF is
port (D, C: in std_logic; Q: out std_logic);
end DFF;

architecture BEH_ of DFF is begin
Q<=D when rising_edge ( C ) else Q;
end ;
```

-- Тестирующая программа для D - триггера.

```
-----
library IEEE; use IEEE.std_logic_1164.all;
```

```

entity DFF_TB is end DFF;
architecture BEH_ of DFF_TB is begin
signal D, C,Q: std_logic;
component DFF port (D, C: in std_logic; Q: out std_logic);
end component ;

begin; --ниже конкретизация модели триггера

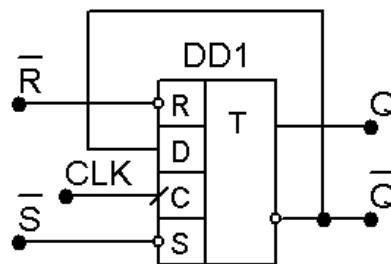
UUT:DFF port map (D, C, Q);
--ниже генератор тактов C
Process begin C<='0';wait for 20 ns; C<='1';wait for 20 ns;
End process; --ниже генератор сигнала D
Process
begin D<='0';wait for 100 ns; D<='1';wait for 100 ns;
End process;

End;

```

#### 4.2.10. Счетный Т-триггер с динамическим управлением.

Т-триггер изменяет свое состояние на противоположное после прихода очередного тактового сигнала. Т-триггер можно построить на основе D-триггера с динамическим управлением ( рис.4.11).

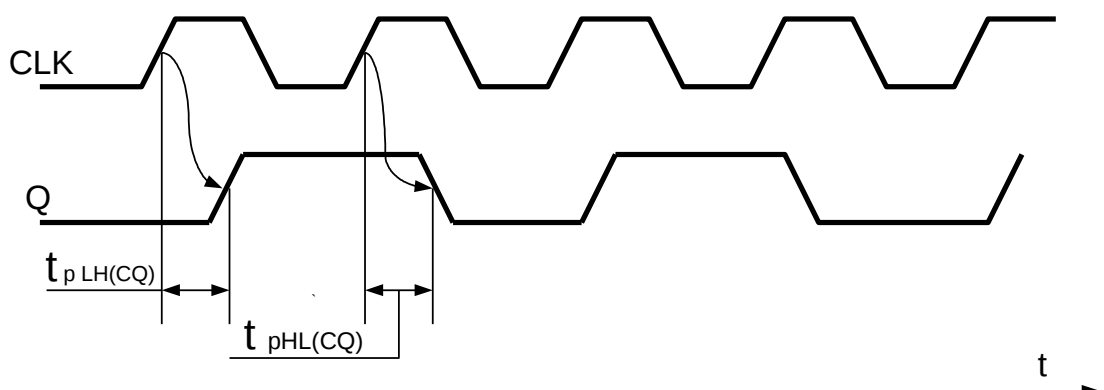


**Рис. 4.11. Схемы счетного триггера на основе стандартных D и JK-триггеров.**

Для реализации счетного триггера вход D соединяют с инверсным выходом Q. Изменение состояния Т - триггера происходит по фронту импульсов синхронизации CLK, при этом при поступлении очередного фронта состояние триггера Q меняется из Q = "1" в состояние Q = "0" и наоборот.

Период выходной последовательности равен двойному периоду сигнала CLK.

Временная диаграмма работы Т-триггера приведена на следующем рисунке



**Рис. 4.12. Временная диаграмма работы Т-триггера.**

#### 4.2.11. Описание счетного Т-триггера на VHDL.

Напомним, что оператор назначения сигнала  $\leq$  VHDL имеет так называемую  $\delta$ -задержку изменения значения сигнала в его левой части сигнала, равную минимальному шагу времени в процессе моделирования. Приведенный ниже неточный вариант описания счетного Т-триггера предполагает, что всегда при подаче питания исходное состояние триггера = "0", что обычно не соответствует действительности для реальных триггеров.

```
library IEEE; [1][2]
use IEEE.STD_LOGIC_1164.all;
entity T_Tr_1 is
    port( C : in STD_LOGIC;
          Q : out STD_LOGIC );
end T_Tr_1;

architecture T_Tr_1_arc of T_Tr_1 is
    signal D,D1: STD_LOGIC := '0';
    begin
    process(C)
        begin
            if (C'event and C = '1') then
                Q <= not D;
                D1 <= not D;
            end if;
        end process;
        D <= D1;
    end T_Tr_1_arc;
```

В VHDL описании Т-триггера в условном операторе if передний фронт синхроимпульса выделяется с помощью логического выражения (C'event and C = '1').

Более реалистичная модель Т-триггера с входным асинхронным сигналом сброса R, предполагающая неопределенное состояние триггера после подачи питания, в связи с чем его надо установить в 0, приведена ниже.

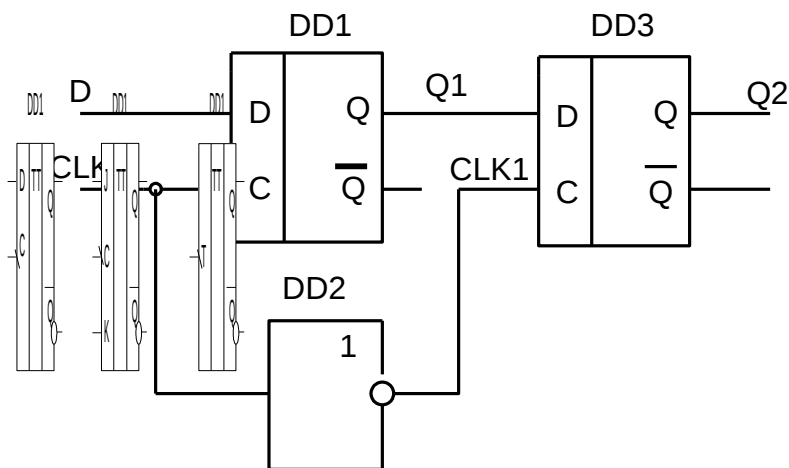
```
library IEEE;use IEEE.STD_LOGIC_1164.all; [1][2]
entity T_Tr_2 is port( C,R : in STD_LOGIC; Q : out STD_LOGIC );
end ;

architecture BEH of T_Tr_2 is
    signal D: STD_LOGIC ';
    begin
    process(C,R) begin
        if R='1' then D<='0';
        elsif (C'event and C = '1') then D <= not D;
        end if;
    end process;
    Q <= D;
end;
```



#### 4.2.12. Двухступенчатый D-триггер, срабатывающий по спаду синхроимпульса.

Схема двухступенчатого D-триггера, построенного на двух последовательно соединенных D-триггерах с синхронизацией по уровню, приведена на следующем рисунке.



а)

б)

а) – схема, б) - УГО

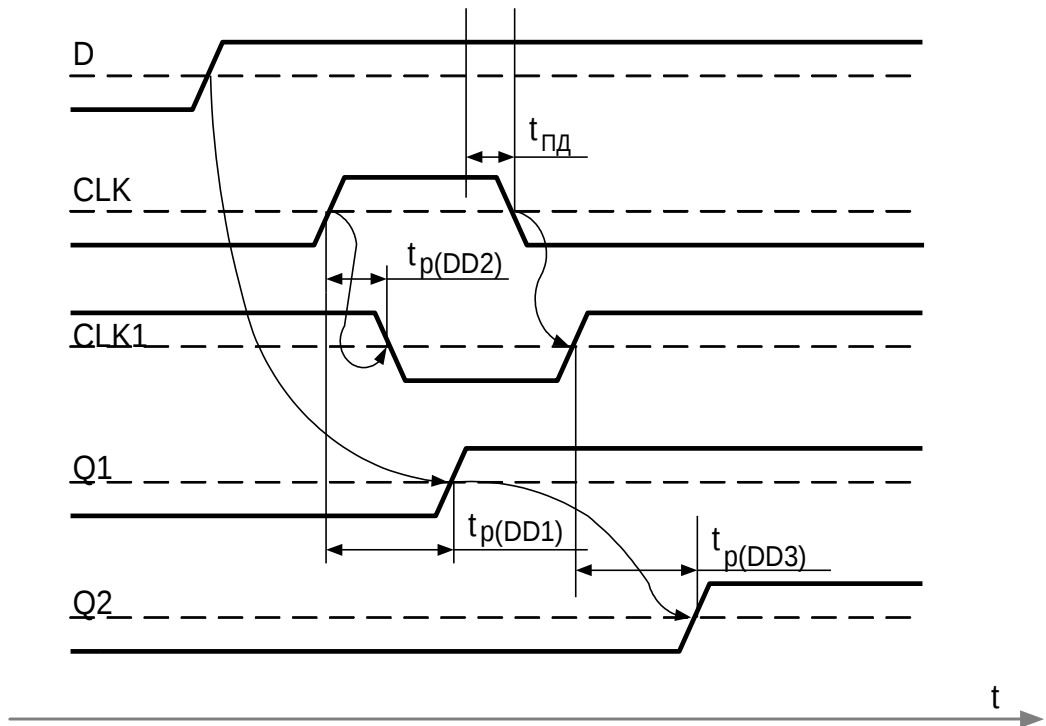
**Рис. 4.14. Двухступенчатый D-триггер и его УГО.**

Первый триггер DD1 является ведущим (Master) [должен быть раздел про идеологию МастерСлайв]. Он изменяет свое состояние в соответствии с входным сигналом D в то время, когда сигнал CLK равен 1. В это время инвертор DD2 подает на вход синхронизации второго ведомого (Slave) триггера DD3 сигнал CLK1, равный 0. Поэтому триггер DD3 закрыт и сохраняет свое предшествующее состояние.

На спаде сигнала CLK триггер DD1 закрывается и фиксирует на выходе Q1 последнее значение сигнал D. С задержкой, равной времени задержки сигнала в инверторе DD2, ведомый триггер DD3 открывается и передает на свой выход Q2 сигнал с выхода Q1.

Временная диаграмма работы двухступенчатого D-триггера представлена на следующем рисунке.





**Рис. 4.15. Временная диаграмма работы двухступенчатого D-триггера.**

Таким образом, схема, собранная из двух последовательно соединенных прозрачных D-триггеров (защелок), является теоретически непрозрачной по входу D.

Интервал времени, предшествующий срезу сигнала CLK ( $t_{\text{пд}}$ ), является опасным. В это время сигнал на входе D не должен изменяться.

Для реальных микросхем в схеме двухступенчатого триггера существует окно прозрачности, равное времени задержки сигнала в инверторе DD2. Однако на практике при правильном выборе временных соотношений вероятность сбоя при работе двухступенчатого триггера является очень малой.