

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**Л.А.Брякин**

**ОСНОВЫ СХЕМОТЕХНИКИ  
ЦИФРОВЫХ УСТРОЙСТВ**

Конспект лекций

ПЕНЗА 2005

УДК 681.3

Первая часть конспекта лекций содержит описание принципов работы и использования триггеров, основных комбинационных и накапливающих узлов цифровых устройств. Приведённые во введении краткие сведения о булевой алгебре и элементной базе облегчают понимание основного материала без привлечения дополнительной литературы.

Конспект лекций подготовлен на кафедре «Вычислительная техника» Пензенского государственного университета и предназначен для студентов специальности 23.01.00 (Вычислительные машины, комплексы, системы и сети) и для студентов других специальностей, осваивающих схемотехнику цифровых устройств.

Составитель: **Л.А.Брякин**

## Содержание

### 1. Введение в схемотехнику

1.1. Предмет дисциплины, её цель и решаемые задачи

1.2. Основные логические функции и основы синтеза схем

1.3. Элементная база ЭВМ

1.3.1. Основные сведения

1.3.2. Физическая реализация логических функций

1.3.3. Основные параметры цифровых элементов и микросхем

1.4. Разновидности электрических схем

1.5. Моделирование электрических схем

Контрольные вопросы

### 2. Схемотехника комбинационных узлов

2.1. Общие сведения

2.2. Дешифраторы

2.3. Шифраторы

2.4. Мультиплексоры

2.5. Демультимплексоры

2.6. Схемы сравнения кодов

2.7. Комбинационные сумматоры

2.8. Арифметико-логические устройства

Контрольные вопросы

### 3. Схемотехника триггеров

3.1. Общие сведения

3.2. Асинхронный RS-триггер

3.2.1. Асинхронный RS-триггер на элементах И-НЕ

3.2.2. Асинхронный RS-триггер на элементах ИЛИ-НЕ

3.3. Синхронные триггеры с потенциальным управлением

3.3.1. Общие сведения

- 3.3.2. Одноступенчатый RS-триггер
  - 3.3.3. Одноступенчатые D- и DV-триггеры
  - 3.3.4. Проблема генерации в одноступенчатом триггере
  - 3.3.5. Двухступенчатые синхронные триггеры
    - 3.3.5.1. Двухступенчатый RS-триггер
    - 3.3.5.2. T-триггер
    - 3.3.5.3. JK-триггер
  - 3.4. Триггеры с динамическим управлением
  - 3.5. Некоторые применения универсальных триггеров
  - 3.6. Основы синтеза триггерных схем
- Контрольные вопросы

- 4. Схемотехника накапливающих узлов
    - 4.1. Общие сведения
    - 4.2. Регистры
      - 4.2.1. Простейшие регистры
      - 4.2.2. Регистры сдвига
      - 4.2.3. Реверсивный сдвигающий регистр
    - 4.3. Счётчики
      - 4.3.1. Общие сведения
      - 4.3.2. Счётчики с последовательным переносом
      - 4.3.3. Счётчики с параллельным и сквозным переносом
      - 4.3.4. Счётчики по произвольному основанию
- Контрольные вопросы

# 1. Введение в схемотехнику

## 1.1. Предмет дисциплины, её цель и решаемые задачи

Под схемотехникой будем понимать дисциплину, которая изучает основы построения аппаратных средств цифровых устройств, в том числе цифровых вычислительных машин (ЦВМ) и персональных компьютеров. В рамках дисциплины рассматриваются принципы работы логических элементов и элементная база цифровых устройств, основные комбинационные и накапливающие узлы, триггеры, принципы организации и функционирования полупроводниковой памяти, принципы работы и организации микропроцессоров, микропроцессорных систем и устройств, входящих в состав этих систем.

Целью дисциплины с учётом сказанного является изучение принципов построения, совместной работы и методов проектирования различных узлов и устройств ЦВМ и микропроцессорных систем.

В результате изучения дисциплины студенты должны:

-знать:

- номенклатуру, характеристики и функциональное назначение интегральных схем, выпускаемых промышленностью для ЦВМ;
- основные методы проектирования схем ЦВМ на основе схем средней степени интеграции, матричных БИС, программируемых логических матриц (ПЛМ), программируемых логических интегральных схем (ПЛИС);
- принципы построения интегральных микросхем памяти и способов построения на их основе устройств памяти большой емкости;
- принципы построения однокристалльных микропроцессорных БИС и СБИС и микропроцессорных устройств на их основе.

-уметь:

- выбирать схемотехническую базу при проектировании различных устройств ЦВМ;
- читать электрические схемы средств вычислительной техники;

- составлять временные диаграммы работы простейших устройств вычислительной техники;
- пользоваться средствами моделирования устройств вычислительной техники.

-иметь навыки экспериментального исследования спроектированных схем.

Поскольку объём предлагаемого конспекта лекций ограничен, основное внимание уделено принципам построения, работы и использования основных узлов ЦВМ: триггеров, комбинационных и накапливающих узлов. Предлагается подкрепить предложенный теоретический материал решением практических задач с привлечением программ моделирования с целью проверки результата решения и более глубокого изучения материала. Моделирование удобно выполнять, используя пакет «Electronics Workbench» [3], который практически не требует времени на изучение собственно программного продукта.

Во введении приводятся основные сведения об алгебре логики, о логических функциях, элементной базе цифровой схемотехники и электрических схемах, используемых для описания цифровых устройств на разных уровнях проектирования. Предложенный во введении материал более подробно можно изучить, используя дополнительную литературу. Если предлагаемый материал вам известен, пропустите его и переходите к следующему разделу.

Полного понимания дисциплины и умения использовать полученные знания для анализа работы цифровых устройств или проектирования новых устройств можно достичь, если подкрепить знания принципов работы основных узлов цифровых устройств, которые описываются в предлагаемом конспекте, выполнением лабораторных работ на универсальных стендах-конструкторах и на компьютерах [4] и проектированием реальных цифровых устройств, решающих практически важные задачи.

## **1.2. Основные логические функции и основы синтеза схем**

Известно, что цифровые устройства оперируют с двоичными переменными, которые представляются обычно двумя чётко различимыми уровнями электрического напряжения: уровнем логического нуля  $U^0$  и уровнем логической единицы  $U^1$ . Для описания функционирования цифрового устройства используется булева алгебра, которая определяет несколько способов задания логических функций как функций двоичных переменных. Основные способы описания логических функций представлены ниже.

Любую сколь угодно сложную логическую функцию можно представить с помощью простейших функций, обладающих функциональной полнотой.

Среди простейших логических функций выделим функцию отрицания, конъюнкцию и дизъюнкцию.

Отрицание или функция НЕ - функция одной двоичной переменной, равная нулю, если переменная равна единице и наоборот:

$$y = \bar{x} \text{ (} y \text{ равно не } x \text{)}$$

Реализуется отрицание с помощью инвертора, условное графическое обозначение которого предложено на рис. 1.1,а. Признаком инвертирования на обозначении инвертора является наличие на линии сигнала кружочка.

Функция конъюнкция (логическое умножение) или функция И - функция двух или большего числа переменных, равная единице только в том случае, когда все входящие в её состав переменные равны единице. Для случая двух переменных  $x_1$  и  $x_2$  конъюнкция запишется в следующем виде:  $y = x_1 \& x_2$ ,  $x_1 \cdot x_2$  или  $x_1 x_2$  ( $y$  равно  $x_1$  и  $x_2$ ). Реализуется конъюнкция с помощью конъюнктора, условное обозначение которого предложено на рис. 1.1,б.

Функция дизъюнкция (логическое сложение) или функция ИЛИ - функция двух или большего числа переменных, равна единице, когда хотя бы одна из переменных равна единице. Для случая двух переменных дизъюнкция запишется в следующем виде:  $y = x_1 \vee x_2$  ( $y$  равно  $x_1$  или  $x_2$ ). Реализуется дизъюнкция с помощью дизъюнктора (рис. 1.1,в).

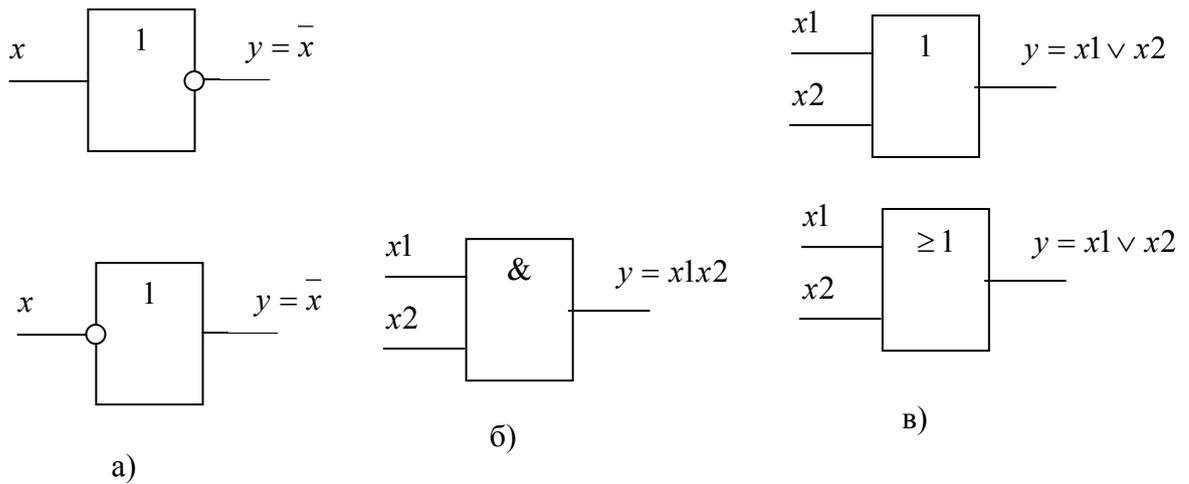
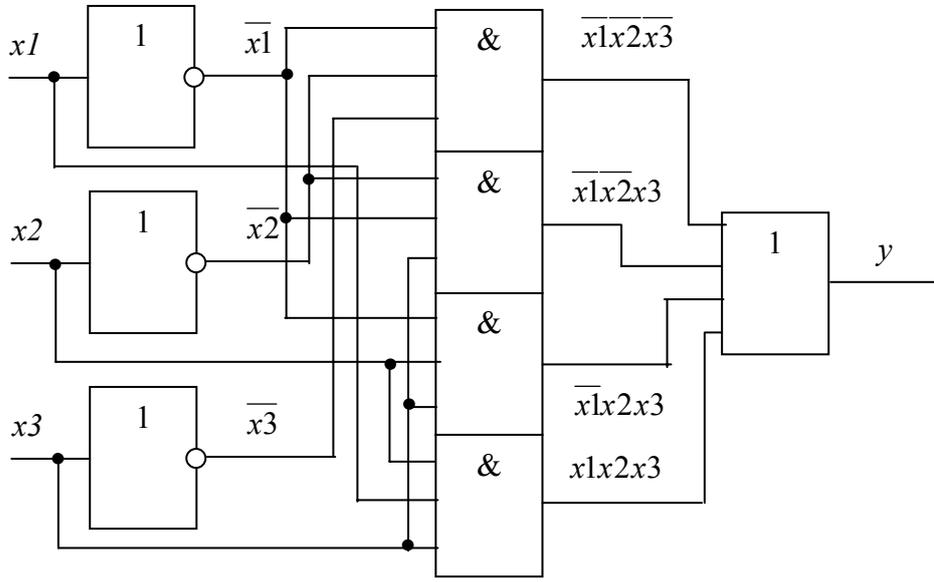


Рис. 1.1. Условные графические обозначения логических элементов на принципиальных или функциональных схемах: а - инвертор; б - конъюнктор; в - дизъюнктор

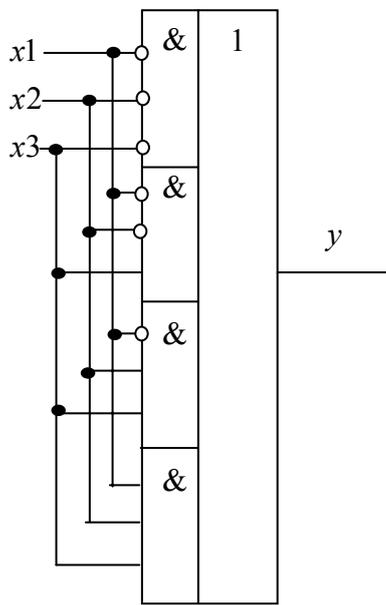
Значения логической функции на всех возможных наборах значений двоичных переменных могут быть заданы табличным или аналитическим способом. Для задания функции может быть использована таблица истинности, форма которой для функции трех переменных  $y=f(x1,x2,x3)$  представлена на рис. 1.2,а.

N	$x1$	$x2$	$x3$	$y$
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

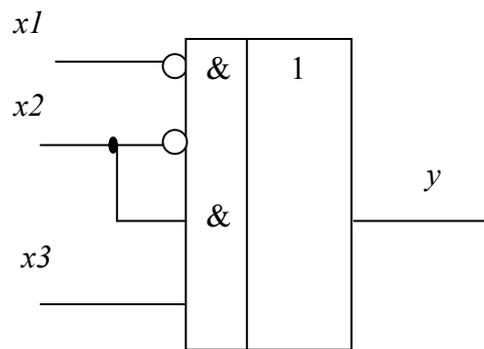
а)



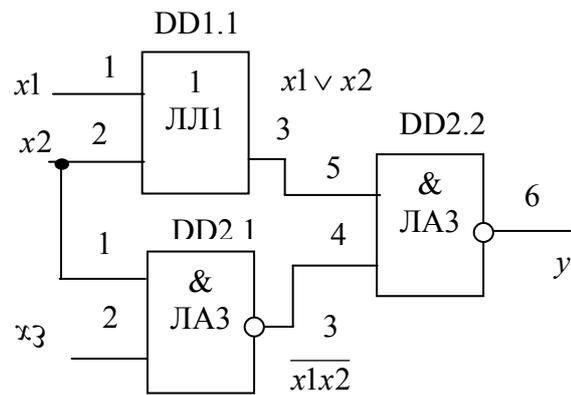
б)



в)



г)



д)

Рис.1.2. Табличный способ задания функции (а) и её реализация в виде электрической схемы (б, в, г, д)

Количество наборов возможных значений переменных определяет число строк таблицы и при  $n$  переменных равно  $2^n$ . Можно заметить, что номер строки  $N$  соответствует десятичному эквиваленту двоичного многоразрядного числа  $x_1x_2x_3$ , где  $x_3$  - младший разряд.

В столбце  $y$  записываются значения функции для соответствующего набора значений переменных. В данном примере столбец  $y$  заполнен произвольным образом, что позволило задать данной таблицей одну из множества возможных логических функций  $n$  переменных. Если на некоторых наборах значение функции произвольно или не определено, то подобные функции называют частично заданными или не полностью определенными. При необходимости эти функции могут быть разумным образом доопределены путём подстановки в соответствующих местах таблицы нуля или единицы вместо пробела  $\square$ .

От табличного способа задания функции легко перейти к аналитическому, используя понятие совершенной дизъюнктивной нормальной формы (СДНФ). СДНФ представляет дизъюнкцию конъюнкций. Каждая конъюнкция представляет конституенту единицы. В состав каждой конъюнкции или конституэнты входят все переменные, взятые с отрицанием или без отрицания в зависимости от того, какому набору значений переменных данная конъюнкция соответствует. Конъюнкция  $\bar{x}_1 \& \bar{x}_2 \& \bar{x}_3$  например, соответствует набору  $N=0$ , т.е. только в том случае, когда  $x_1=0$ ,  $x_2=0$  и  $x_3=0$  данная конъюнкция равна единице. А конъюнкция  $\bar{x}_1 \& x_2 \& x_3$  соответствует набору  $N=3$ , то есть  $x_1=0$ ,  $x_2=1$  и  $x_3=1$ . Причём, количество конъюнкций равно числу наборов, на которых функция равна единице. Тогда СДНФ для функции  $y$  будет следующая:

$$y = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$$

Используя аналитическую форму представления логической функции, легко построить комбинационную схему на инверторах, конъюнкторах и дизъюнкторах, которая реализует эту функцию (рис. 1.2,б). При этом следует реализовать сначала операции над отдельными переменными (отрицание), затем конъюнкции и в последнюю очередь - дизъюнкции. Схема может быть представлена в компактном виде, если использовать дополнительное поле в условном обозначении дизъюнк-

тора и воспользоваться символом отрицания (рис. 1.2,в). На практике редко строится комбинационная схема непосредственно по СДНФ. Во-первых, потому, что необходимо учитывать возможности реальных логических элементов (их число входов и реализуемые ими функции) и соответствующим образом видоизменить аналитическое выражение, чтобы можно было реализовать схему в заданной системе элементов или в заданном базисе. Для преобразования аналитического выражения с этой целью используют правило де Моргана, которое формулируется следующим образом:

$$\overline{z_1 \vee z_2 \vee \dots \vee z_m} = \overline{z_1} \overline{z_2} \dots \overline{z_m},$$

$$\overline{\overline{z_1} \overline{z_2} \dots \overline{z_m}} = z_1 \vee z_2 \vee \dots \vee z_m,$$

где под  $z_i$  понимается отдельная переменная или комбинация переменных. Для рассматриваемого случая применение правила де Моргана с целью исключения функции дизъюнкции позволяет прийти к выражению, требующему для своей реализации только логические элементы, выполняющие функции конъюнкция и отрицание (функция И-НЕ):

$$y = \overline{\overline{x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 x_2 x_3}} = \overline{\overline{(x_1 x_2 x_3)(x_1 x_2 x_3)(x_1 x_2 x_3)(x_1 x_2 x_3)}}$$

Здесь используется тот факт, что двойное отрицание от логической функции равно самой функции.

Во-вторых, прежде чем реализовать комбинационные схемы, обычно упрощают (или минимизируют) аналитическую запись, используя аксиомы алгебры логики. Вот некоторые из них:

$$\begin{aligned} \bar{1} &= 0; & \bar{0} &= 1; & 1x &= x; & 0x &= 0; \\ x &= xx; & x \vee x &= x; & 1 \vee x &= 1; & 0 \vee x &= x; \\ x\bar{x} &= 0; & x \vee \bar{x} &= 1; & \overline{\overline{x}} &= x; & x1 \vee x1x2 &= x1; \\ & & & & & & x1(x1 \vee x2) &= x1; \end{aligned}$$

Кроме отмеченных аксиом, для логических выражений справедливы переместительные, распределительные и сочетательные законы:

$$\begin{aligned}x_1 \vee x_2 &= x_2 \vee x_1; & x_1(x_2x_3) &= (x_1x_2)x_3; \\x_1x_2 &= x_2x_1; & x_1 \vee (x_2 \vee x_3) &= (x_1 \vee x_2) \vee x_3; \\x_1(x_2 \vee x_3) &= x_1x_2 \vee x_1x_3; & x_1 \vee x_2x_3 &= (x_1 \vee x_2)(x_1 \vee x_3)\end{aligned}$$

В процессе минимизации отыскиваются в СДНФ или произвольной исходной аналитической форме конъюнкции, отличающиеся в одной переменной, и к ним применяется правило склеивания:  $x_1 \cdot z \vee \bar{x}_1 \cdot z = (x_1 \vee \bar{x}_1) \cdot z$  или поглощения:  $x_1 \vee x_1 \cdot z = x_1 \cdot (1 \vee z) = x_1$ , где  $z$  - отдельная переменная или логическое выражение. В результате две конъюнкции, связанные дизъюнкцией, заменяются одной, содержащей к тому же меньшее число переменных. Для предложенного примера процесс минимизации выглядит следующим образом:

$$y = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_1x_2x_3 = \bar{x}_1\bar{x}_2(\bar{x}_3 \vee x_3) \vee x_2x_3(\bar{x}_1 \vee x_1) = \bar{x}_1\bar{x}_2 \vee x_2x_3$$

В результате преобразования функция значительно упрощается:

$$y = \bar{x}_1 \cdot \bar{x}_2 \vee x_2 \cdot x_3 \tag{1}$$

Конъюнкция  $\bar{x}_1\bar{x}_2$  в полученном выражении заменяет выражение  $\bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3$  в исходной СДНФ, а конъюнкция  $x_2x_3$  заменяет выражение  $\bar{x}_1x_2x_3 \vee x_1x_2x_3$ . Функциональная схема устройства, реализующего функцию  $y$ , оказывается значительно проще (рис.1.2,г).

При небольшом числе переменных удобно использовать для минимизации диаграммы Вейча, которые представляют разновидность табличного способа задания логической функции. Особенностью диаграммы Вейча является то, что конъюнкции, отличающиеся в одной переменной, оказываются рядом. И если в соседних двух клетках функция равна единице, то возможна операция склеивания по одной переменной, если четыре единицы функции расположены в квадрате  $2 \times 2$  или на одной линии, то возможно склеивание по двум переменным, а если восемь единиц расположены в прямоугольнике  $2 \times 4$ , то возможно склеивание по трем переменным. Диаграмма Вейча для трех переменных представлена на рис. 1.3. Эта

диаграмма заполнена в соответствии с таблицей истинности, представленной на рисунке 1.2,а.

Выполняя операцию склеивания в представленном случае над двумя парами единиц так, как показано на диаграмме, можно упростить или минимизировать выражение. В результате предложенная таблицей истинности функция запишется в следующем виде:

$$y = \overline{x1} \cdot \overline{x2} \vee x2 \cdot x3.$$

Можно заметить, что получено такое же выражение для исходной функции, какое получено при использовании правила склеивания непосредственно к СДНФ.

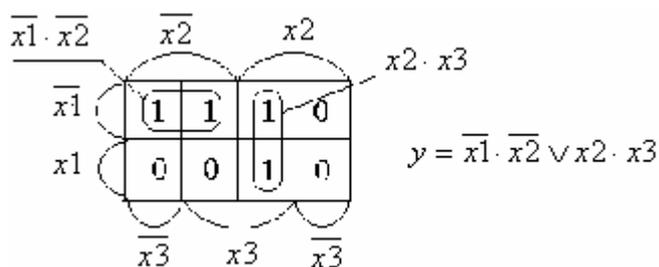


Рис.1.3. Диаграмма Вейча

Диаграмму Вейча следует считать сложной фигурой, у которой крайние боковые стороны могут быть соединены в цилиндр и крайние горизонтальные стороны также могут быть замкнуты в цилиндр.

Аксиомы булевой алгебры дают возможность преобразовывать формулу выходной переменной  $y$  таким образом, чтобы возможна была реализация выражения на определённых логических элементах, выполняющих известные логические функции. Это позволяет перейти от функциональной схемы (рис.1.2,б, рис.1.2, в и рис.1.2,г) к принципиальной схеме устройства, которая предполагает применение конкретных микросхем в составе устройства. Осуществим возможные преобразования для выражения (1). Избавимся от необходимости применения инверторов на каждом входе, используя двойное отрицание:

$$y = \overline{x_1} \cdot \overline{x_2} \vee x_2 \cdot x_3 = \overline{\overline{\overline{x_1} \cdot \overline{x_2}}} \vee x_2 \cdot x_3 = \overline{(x_1 \vee x_2)} \vee x_2 x_3.$$

Дизъюнкция  $(x_1 \vee x_2)$  возникла благодаря применению правила де Моргана.

Применим это же правило для дальнейших преобразований:

$$y = \overline{(x_1 \vee x_2)} \vee x_2 x_3 = \overline{\overline{\overline{(x_1 \vee x_2)} \vee x_2 x_3}} = \overline{(x_1 \vee x_2)} \cdot \overline{(x_2 x_3)}.$$

В результате преобразований получаем выражение:

$$y = \overline{(x_1 \vee x_2)} \cdot \overline{(x_2 x_3)}.$$

Полученное выражение реализуется с использованием двухвходового дизъюнктора (например, из микросхемы К555ЛЛ1) и элементов типа И-НЕ, содержащихся в микросхеме типа К555ЛА3. Принципиальная схема синтезированного устройства предложена на рисунке 1.2,д.

На этой схеме около условных обозначений логических элементов цифрами указаны номера выводов конкретных микросхем. Тип микросхемы представлен сокращённым названием внутри условного обозначения каждого логического элемента. Устройство реализуется на двух микросхемах: DD1 – К555ЛЛ1, DD2 – К555ЛА3. Идентификатор микросхемы строится с помощью двух букв и порядкового номера микросхемы на чертеже. Первая буква D обозначает, что имеем дело с микросхемой, вторая буква D говорит о том, что микросхема цифровая, то есть работает с двоичными сигналами. Если вторая буква A, то микросхема аналоговая, обрабатывающая аналоговые сигналы. Если реализация на микросхемах подразумевается, то первая буква может быть пропущена. Если в корпусе микросхемы реализовано несколько элементов, то их условные обозначения могут располагаться в разных местах чертежа, но их порядковый номер будет один и тот же и соответствовать номеру первого из них в процессе нумерации элементов сверху вниз чертежа и слева направо. Через точку к порядковому номеру добавляется номер элемента в корпусе микросхемы. Из корпуса микросхемы К555ЛА3 в схеме на рисунке 1.2,д используется два элемента из четырёх, а в микросхеме К555ЛЛ1 используется только один из четырёх элементов. Соответственно элементы из

корпуса К55ЛА3 идентифицированы как DD2.1 и DD2.2, а элемент микросхемы назван DD1.1.

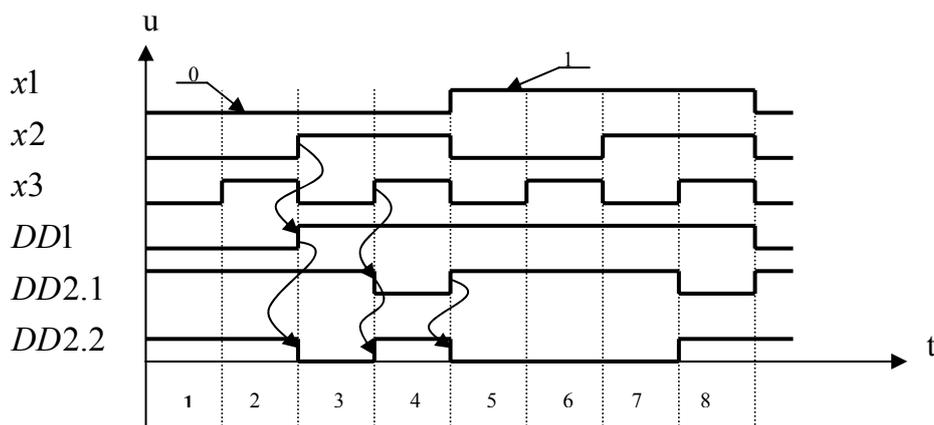
Чтобы микросхемы выполняли логические функции надо подвести питающее напряжение: к выводам 7 микросхем следует подключить минус пятивольтового источника напряжения, а к выводам 14 микросхем следует подключить плюс того же источника. Входные и выходные переменные задаются относительно выводов 7, то есть относительно отрицательного полюса источника напряжения питания.

Синтезированное устройство можно отнести к классу комбинационных узлов, то есть к цифровым узлам, не содержащим запоминающие элементы. Всякое изменение состояний входных сигналов может привести к изменению состояния выходного сигнала с минимальной задержкой, определяемой лишь задержкой в прохождении информации через используемые элементы. Быстродействие определяется возможностями используемых элементов. Для выбранных элементов максимальная задержка в формировании правильного значения выходного сигнала в ответ на изменение входных сигналов не превышает задержки двух элементов (рис.1.2,г) и не более  $(22+15)$ нсек (наносекунд) [5], где 22нсек - задержка элемента К555ЛЛ1, а задержка элемента К555ЛА3 – не более 15нсек.

Булева алгебра позволяет успешно решать задачи синтеза комбинационных схем, выполняющих операции над двоичными числами или двоичными переменными. Некоторые примеры применения булевой алгебры для описания функционирования комбинационных узлов будут предложены ниже. Но булева алгебра позволяет решать и задачи анализа комбинационных схем с целью проверки их работоспособности, понимания принципов работы устройства или с целью упрощения, минимизации получаемых при анализе функций. Исходным материалом при этом может быть принципиальная схема устройства. При известной принципиальной схеме комбинационного узла достаточно путём перебора всех возможных состояний входных сигналов составить таблицу истинности выходных сигналов, что позволяет затем выполнить анализ работы устройства и упростить устройство при необходимости путём минимизации и проектирования новой схемы.

Для описания работы цифровых устройств широко используют временные диаграммы: представленные во времени во взаимосвязи входные и выходные сигналы устройства. При построении временных диаграмм обычно идеализируется форма входных и выходных сигналов, которые изображаются с идеальными фронтами. Временные диаграммы могут строиться с учётом задержек сигналов в реальных элементах или без учёта, то есть в предположении максимального быстродействия, то есть для идеального случая. Будем считать, что уровень напряжения  $U^1$ , которым представляется единица, выше уровня напряжения логического нуля  $U^0$ :  $U^1 \gg U^0$ . В этом случае уровень логической единицы изображается на временных диаграммах выше уровня нуля. Строить временные диаграммы без учёта задержек проще, поэтому при ручном изображении временных диаграмм используют именно этот способ. С помощью стрелок можно на временных диаграммах указать причинно-следственные связи, что облегчает восприятие диаграмм и анализ работы устройства. Наличие взаимосвязанных стрелок позволяет при необходимости быстро оценить и задержку в формировании выходных сигналов.

Опишем поведение разработанной схемы (рис.1.2,д) с помощью временных диаграмм. При этом произвольно зададим входные сигналы на оси времени, перебрав все возможные наборы значений переменных (что не всегда целесообразно). С учётом значений входных сигналов определим выходные сигналы всех трёх элементов устройства (DD1, DD2.1, DD2.2) и стрелками покажем причинно-следственные связи. Результат построения временных диаграмм без учёта задержек предложен на рисунке 1.4.



## Рис.1.4.-Временные диаграммы

Задав входные сигналы, необходимо поделить временную ось на интервалы стабильного состояния этих сигналов. В данном случае это привело к восьми интервалам с номерами от 1 до 8. На первом интервале наблюдается исходное состояние схемы. Входные сигналы имеют при этом низкие уровни, что соответствует логическому нулю. На выходе элемента DD1 наблюдается логический нуль, а на выходах элементов DD2.1 и DD2.2 наблюдаются уровни логической единицы. В момент изменения сигнала  $x_2$  с нуля на единицу (нарастающий фронт) появляется уровень единицы на выходе DD1 в третьем интервале времени, что приводит к появлению нуля на выходе DD2.2 в том же интервале времени. Задержка в формировании спадающего фронта на выходе DD2.2 равна сумме времени задержки элемента DD1 и элемента DD2.2. Аналогично можно рассуждать для других моментов времени.

### **1. 3. Элементная база ЭВМ**

#### **1.3.1. Основные сведения**

ЦВМ состоит из множества элементов, взаимодействие между которыми осуществляется путём обмена сигналами [6]. В общем случае под сигналом можно понимать физический процесс, несущий информацию или представляющий интерес для наблюдателя. Физически сигнал в ЦВМ представляется уровнем электрического напряжения. Различают сигнал аналоговый и цифровой. Под аналоговым понимается сигнал, для которого важным считается конкретное значение напряжения, которое может принимать любую величину в заданном диапазоне значений. Под цифровым сигналом понимают электрический сигнал, для которого справедливым или допустимым являются лишь два значения: 0 и 1. Различают два способа представления цифровой информации. При импульсном представлении за 0 принимается тот сигнал, которому соответствует отсутствие импульса (нулевой уровень напряжения). Двоичная единица в этом случае, представляется одним или множеством импульсов электрического напряжения. Под импульсом

можно понимать быстрое кратковременное изменение электрического напряжения. При потенциальном способе представления двоичной информации нулю и единице соответствуют два чётко различимых уровня напряжения:  $U^0$  и  $U^1$ . Для современных логических элементов обычно  $U^0 \approx 0\text{В}$ ,  $U^1 > 0\text{В}$ . Для элементов серии ТТЛ (транзисторно-транзисторная логика):  $U^1 \geq 2,4\text{В}$ ,  $U^0 \leq 0,4\text{В}$ .

Под элементом понимается наименьшая функционально и конструктивно законченная часть ЦВМ, которая выполняет какую-либо логическую (булеву) функцию (логический элемент) или какое-либо другое преобразование информации. В настоящее время элементы реализуют в составе микросхем. Микросхема — это микроэлектронное изделие с повышенной степенью интеграции. Будем считать, что понятия микросхема и интегральная схема равноценны. В зависимости от используемых сигналов различают элементы и микросхемы аналоговые, аналого-цифровые и цифровые. В зависимости от степени интеграции, то есть от числа элементов в составе микросхемы, различают микросхемы малой, средней, большой и сверхбольшой степени интеграции.

В зависимости от технологии изготовления различают микросхемы, выполненные по интегральной, гибридной или плёночной технологии. Интегральная технология предполагает выполнение всех деталей в одном кристалле полупроводника (чаще всего — кремния). Гибридная технология предполагает применение интегральных элементов в бескорпусном варианте, которые монтируются на изоляторе и соединяются с помощью напылённых проводников. Плёночная технология предполагает применение плёнок даже при изготовлении активных компонентов.

В зависимости от базовой схемы, которая используется при реализации элементов в микросхеме, различают микросхемы, выполненные по следующим технологиям:

1. ДТЛ - диодно-транзисторная логика;
2. ЭСЛ — эмиттерно-связанная логика;
3. ТТЛ — транзисторно-транзисторная логика;
4. ТТЛШ — ТТЛ с диодами Шоттки;

5. МОП (металл- окись- полупроводник);
6. МДП (металл- диэлектрик- полупроводник);
7. КМОП – комплементарные МОП;
8. КМДП - комплементарные МДП;

В зависимости от используемых транзисторов различают микросхемы, выполненные по биполярной технологии и по МДП-технологии.

Цифровые микросхемы выпускают в виде серий элементов или микросхем. Элементы или микросхемы одной серии - это микросхемы, выполненные в одинаковых корпусах, с одинаковыми характеристиками, с одинаковым питанием и предназначенные для совместного применения. В микросхемах одной серии обязательным условием их совместного применения является одинаковый способ представления в них двоичной информации, одинаковые логические уровни.

### **1.3.2. Физическая реализация логических функций**

Любое цифровое устройство можно создать, используя простейшие логические элементы, если выполняемые ими логические выражения являются функционально полными. Сколь угодно сложное устройство можно реализовать на элементах, выполняющих логическую функцию типа И-НЕ или ИЛИ-НЕ. То есть, если имеются логические двухвходовые элементы, выполняющие функцию «конъюнкция-отрицание» ( $y = \overline{x1 \& x2}$ ) или «дизъюнкция-отрицание» ( $y = \overline{x1 \vee x2}$ ), то на них можно построить сколь угодно сложное цифровое устройство. Обратим внимание на способы реализации отмеченных функций. При анализе предлагаемых ниже схем будем считать, что уровень логической единицы выше уровня логического нуля. Для физической реализации выделенных функций в простейшем случае можно использовать транзисторную логику.

При реализации транзисторной логики используют параллельное и последовательное включение транзисторов. Наличие транзисторов приводит к тому, что кроме функции типа «И», «ИЛИ» элементы автоматически выполняют функцию «НЕ». Рассмотрим реализацию транзисторной логики на биполярных транзисторах, поскольку принцип работы элементов на МДП-транзисторах такой же.

Параллельное включение транзисторов предложено на рисунке 1.5а. Только при наличии на всех входах уровня логического нуля транзисторы окажутся закрытыми, на выходе наблюдается уровень логической единицы. То есть элемент реализует функцию «2ИЛИ-НЕ», где цифра 2 указывает на число входов дизъюнктора:

$$y = \overline{x1 \vee x2}.$$

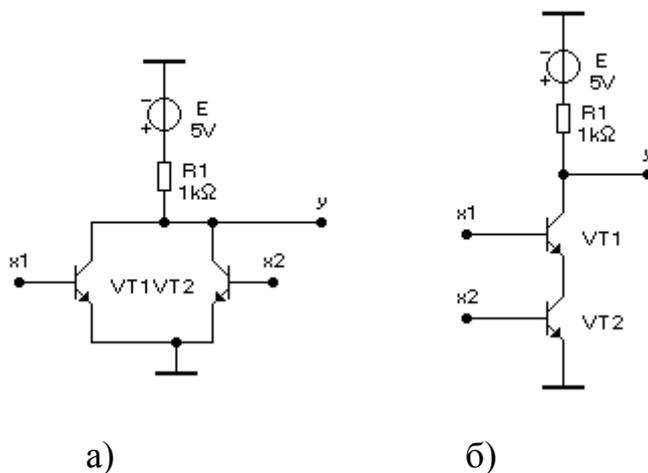


Рис. 1.5. - Схемные решения транзисторной логики

При последовательном включении транзисторов (рис. 1.5,б) только при наличии на всех входах логической единицы на выходе наблюдается нуль. Элемент реализует функцию «2И-НЕ», функция связана с входными переменными следующим образом:

$$y = \overline{x1 \& x2}.$$

### 1.3.3. Основные параметры цифровых элементов и микросхем

Чтобы успешно использовать элементы в составе цифровых устройств необходимо учитывать хотя бы некоторые параметры используемых элементов. Выделим основные параметры, учёт значений которых позволяет создавать работоспособные цифровые устройства и определять их быстродействие и входные и выходные допустимые токи в статике.

1. Логические уровни - это уровни напряжения, которые соответствуют нулю и единице -  $U^1$  и  $U^0$ .

2. Пороговое напряжение - это граница раздела нуля и единицы в бесконечной цепочке элементов.
3. Максимальная величина входного тока определяется при нуле и единице на входе.
4. Максимальный выходной ток элемента определяет его нагрузочную способность. Выходные токи также определяются при уровнях логического нуля и логической единицы на выходе элемента.
5. Нагрузочная способность в пределах серии элементов определяется максимально допустимым количеством элементов (а точнее, входов элементов) той же серии, которые могут быть подключены к выходам элементов.
6. Быстродействие элементов определяется временем задержки информации и продолжительности фронтов сигналов на выходе.

Проектируемое устройство будет работоспособно, если во всех используемых в нём элементах логические уровни одинаковы, если с выходов элементов потребляется допустимый ток. Знание входных и выходных токов элементов позволяет определить входные и выходные токи всего устройства, а знание времени задержки используемых элементов позволяет оценить быстродействие всего устройства.

#### **1.4. Разновидности электрических схем**

Электрическая схема – способ изображения цифрового изделия на плоскости с использованием условных графических обозначений (УГО) элементов и узлов.

Вычислительная техника использует 3 разновидности электрических схем:

- 1) схема электрическая структурная (Э1);
- 2) схема электрическая функциональная (Э2);
- 3) схема электрическая принципиальная (Э3).

На структурных схемах изображают прямоугольниками или другими графическими обозначениями основные функциональные части изделия и основные взаимосвязи между ними. Графическое построение схемы должно давать пред-

ставление о последовательности взаимодействия функциональных частей в изделии. Стрелками на линиях взаимосвязи показывают направление хода процессов, происходящих в изделии. Структурная схема даёт общее представление о функциональном составе (из каких блоков, частей или устройств состоит изделие) изделия и взаимодействии основных блоков между собой.

По функциональной схеме можно понять принцип функционирования изделия, из каких основных узлов состоит изделие. При синтезе цифровых устройств с использованием булевой алгебры, если схема проектируется не на определённой серии элементов, не в определённом базисе элементов создаётся именно функциональная схема. Примерами функциональных схем являются схемы, предложенные на рисунках 1.2,б и 1.2,в. Если же проектируется функциональная схема сложного устройства, то в составе этого устройства могут присутствовать комбинационные и накапливающие узлы, представление которых в виде совокупностей инверторов, конъюнкторов и дизъюнкторов нецелесообразно, потребует слишком много места, усложнит анализ работы устройства. В таком случае на функциональных схемах комбинационные узлы изображаются в виде трапеции, а накапливающие узлы в виде прямоугольников, причём информационные входы и выходы располагаются примыкающими к более длинной стороне, а управляющие сигналы - к короткой стороне. Поток информации на функциональной схеме должен идти сверху вниз или слева направо. Направление передачи информации кодируется стрелкой. Высота изображения узла на схеме равна  $H=10,15\dots\text{мм}$ . Длина изображения  $L=1.5H$ .

Если необходимо особо подчеркнуть разрядность узла, которая существенно больше разрядности других узлов, длина изображения выбирается больше.

Примеры изображений комбинационных узлов показаны на рисунке 1.6. Идентификатором DC представлен дешифратор, а идентификатором SM помечен комбинационный сумматор.

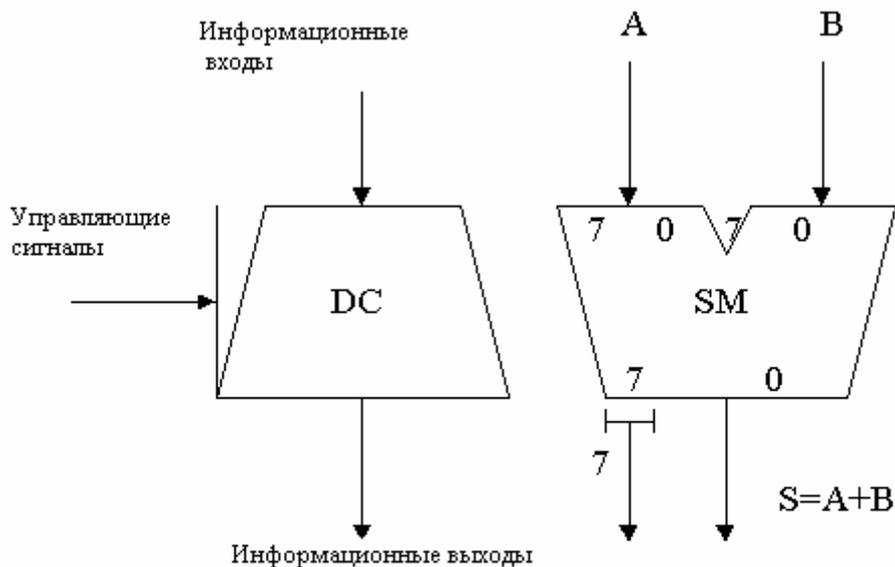


Рис. 1.6.-Условные обозначения комбинационных узлов на функциональных схемах

Пример изображения накапливающего узла на функциональных схемах показан на рисунке 1.7.

Принципиальные схемы используются при изготовлении изделия. Поэтому на них должна быть полная информация обо всех соединениях, разводке питания, используемых деталях, микросхемах. Перечень используемых деталей прилагается к принципиальной схеме как дополнительный документ или предлагается непосредственно на схеме, если деталей немного.

На принципиальных схемах логические элементы и узлы изображаются в виде прямоугольников, имеющих только основное поле, если выполняемая функция достаточно проста (рис. 1.8,а), или имеющих кроме основного дополнительные поля (рис.1.8,б). Входы и выходы располагаются с противоположных длинных сторон условного обозначения.

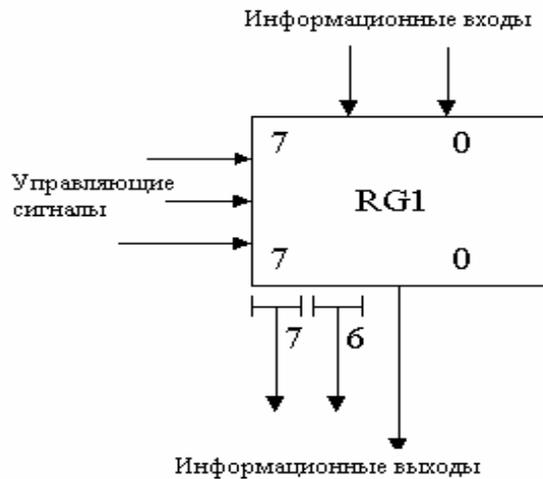


Рис. 1.7.-Пример условного обозначения накапливающего узла на функциональной схеме

Внутри основного поля, напротив верхнего входа, располагается символ выполняемой основной функции, под которым располагают маркировку используемой микросхемы. Входы и выходы помечаются метками (при необходимости).

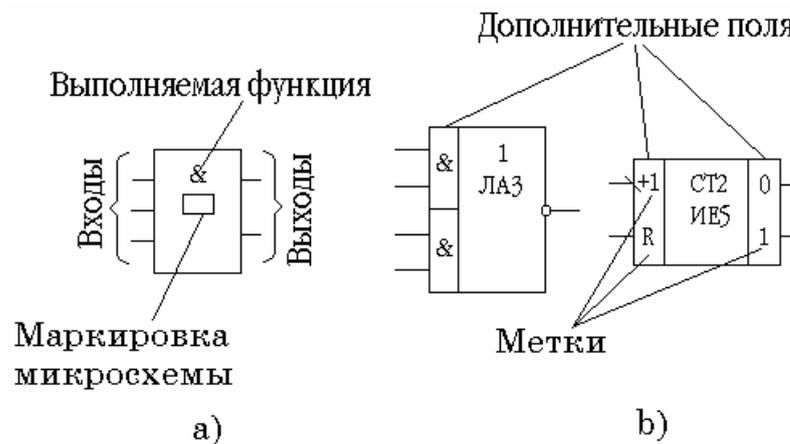


Рис.1.8.-Условные обозначения логических элементов: а - элемент без дополнительных полей; б) - элементы с дополнительными полями

У входов и выходов условных обозначений элементов указывают номера контактов микросхемы, в состав которой входят эти элементы.

На всех электрических схемах обычно не показывают линии питания каждой микросхемы. На принципиальной схеме способ подведения питания к микросхеме показывают с помощью текста или специальных таблиц.

### **1.5. Моделирование электрических схем**

Компьютер открывает прекрасные возможности как при проверке правильности работы синтезированных схем, так и для проектирования схем с заданным алгоритмом работы. Удобно пользоваться для проверки работоспособности схем пользоваться разными версиями программ «Electronics Workbench» [3] и «Multisim 2001», которые практически не требуют обучения при использовании. Четвёртая версия «Electronics Workbench» позволяет с помощью логического преобразователя (Logic converter) определить правильность работы схемы по таблице истинности, которую компьютер составляет автоматически. Результат проверки работоспособности ранее синтезированной схемы (рис. 1.2,б) и использования логического преобразователя предложен на рисунке 1.9.

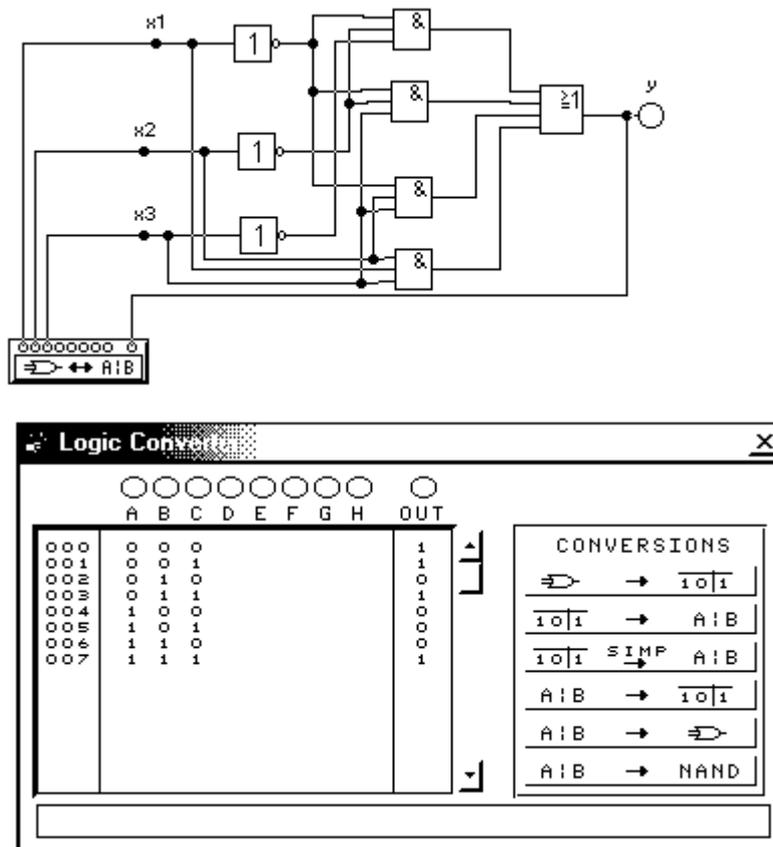


Рис. 1.9. – Результат использования логического преобразователя при моделировании схемы рис. 1.2,б

Сравнивая исходную таблицу истинности (рис. 1.2, а) с полученной при моделировании легко проверить правильность работы схемы.

Возможны и другие способы проверки разработанной схемы. С помощью генератора слов (Word Generator) можно задать сигналы на входах схемы. Поскольку для разработанной схемы были построены временные диаграммы (рис. 1.8), с помощью генератора слов зададим такие же входные сигналы, что и на временных диаграммах рисунка 1.8. Результаты программирования предложены на рисунке 1.10. Генератор позволяет формировать восемь сигналов на шестнадцати временных интервалах. В нашем случае требуется сформировать всего три сигнала. Для проверки правильности работы схемы с помощью логического анализатора (Logic analyzer) можно одновременно наблюдать как входные сигналы, так и выходные. Это позволяет по шагам проанализировать работу схемы и сделать выводы о её работоспособности.

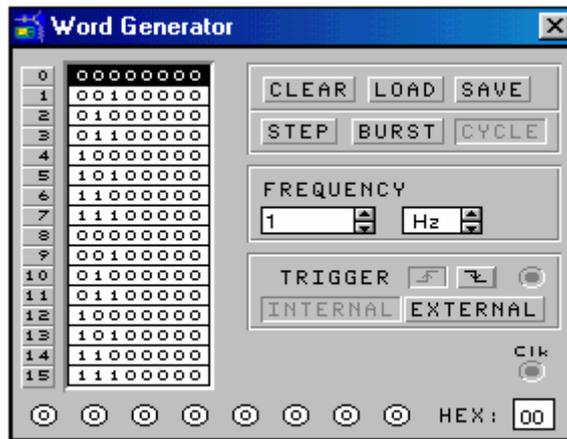


Рис. 1.10. – Панель управления генератором слов

Схема подключения приборов к изучаемой схеме представлена на рисунке 1.11.

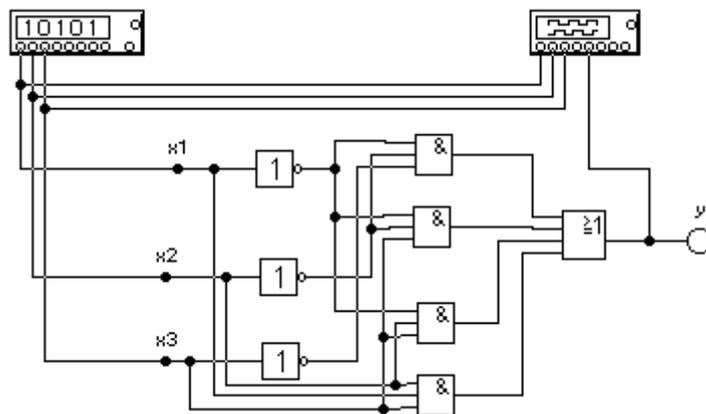


Рис. 1.11. – Схема модели

Временные диаграммы работы схемы предложены на экране логического анализатора (рис. 1.12).

Верхние три сигнала соответствуют входным сигналам схемы ( $x_1$ ,  $x_2$ ,  $x_3$ ). Последний сигнал является выходным сигналом схемы, соответствующим сигналу на выходе DD2.2 рисунка 1.4. На выходном сигнале можно заметить кратковременный паразитный импульс при переходе входных сигналов из состояния 111 в состояние 000. Объясняется это переходными процессами в схеме за счёт ограниченного быстродействия элементов. Когда на входах сигналы уже изменили со-

стояние на нулевые, на выходах инверторов ещё сохраняются нули, что и приводит к ложному кратковременному импульсу на выходе.

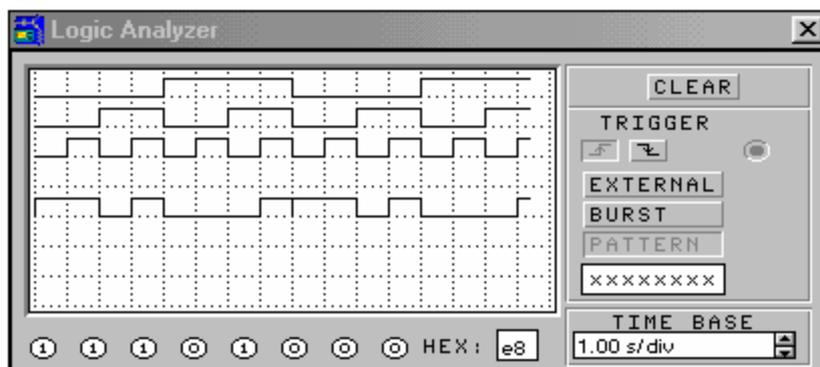


Рис. 1.12. – Результаты моделирования временных диаграмм работы схемы

Программы моделирования позволяют использовать и другие средства проверки работоспособности схем. Например, возможно формирование сигналов с помощью моделей механических ключей (тумблеров), контроль состояний как входных, так и выходных сигналов при этом возможно с помощью моделей логических пробников. В схеме рис. 1.11 на выходе используется для индикации состояния выходного сигнала у логический пробник. В этом случае сигналы задаются путём коммутации ключей и анализа работы схемы для каждой набранной комбинации, то есть в статике.

Возможно формирование сигналов и с помощью моделей независимых генераторов импульсов, а наблюдать результаты работы схемы можно не только с помощью логического анализатора, но и с помощью модели двухканального осциллографа.

## Контрольные вопросы

1. Какими электрическими сигналами представляют двоичные переменные?
2. Какие существуют методы задания булевых функций?
3. При четырёх переменных сколько возможно наборов значений переменных, сколько строк будет содержать таблица истинности?
4. С помощью какого правила можно конъюнкцию заменить дизъюнкцией и наоборот?
5. В чём состоит суть правила склеивания?
6. Как на электрических схемах изображаются инверторы, дизъюнкторы и конъюнкторы?
7. Что такое временные диаграммы?
8. Какова разница между аналоговым и цифровым сигналом?
9. Чем характеризуются микросхемы одной серии?
10. Какие логические функции реализуются при параллельном и последовательном соединении транзисторов?
11. Назовите основные параметры логических элементов.
12. Какие разновидности электрических схем вы знаете? В чём их разница?

## 2. Схемотехника комбинационных узлов

### 2.1. Общие сведения

Под комбинационными понимают узлы, не содержащие элементов памяти. В таких узлах всякое изменение состояния входных сигналов вызывает соответствующее изменение выходных сигналов с минимальной задержкой, зависящей от быстродействия данного узла. В общем случае комбинационный узел содержит несколько входов и несколько выходов. Поведение комбинационного узла описывается функциями алгебры логики (булевыми или логическими функциями). При этом для каждого выхода можно определить соответствующую булеву функцию, которая полностью определяет поведение комбинационного узла по данному выходу без учёта задержки распространения сигнала, то есть для идеального узла. Время задержки сигнала по каждому выходу может быть разным. Время задержки зависит и от того, какой входной сигнал изменением своего состояния привёл к соответствующему изменению данного выходного сигнала.

Поведение комбинационного узла может быть задано таблицей истинности, в которой каждой комбинации значений входных сигналов ставится в соответствие значение каждой выходной переменной. Входные переменные будем обозначать буквой  $x$  с соответствующим индексом, а выходные - буквой  $y$  с соответствующим индексом. По известной таблице истинности можно для каждого выхода составить соответствующую булеву функцию, как это будет продемонстрировано ниже. Количество комбинационных узлов, которые можно синтезировать, огромно даже при достаточно небольшом числе входных переменных. Но практика проектирования средств вычислительной техники показывает, что в большинстве случаев не требуется собственно разработки нового комбинационного узла, а достаточно ограничиться применением известных схемных решений, реализованных в виде микросхем. Достаточно лишь разумно использовать стандартные микросхемы.

Выделим следующие комбинационные узлы, которые реализованы в виде микросхем: дешифраторы и шифраторы, мультиплексоры и демультимплексоры,

компараторы цифровых сигналов, сумматоры, арифметико-логические устройства. Именно эти комбинационные узлы рассмотрим в этом разделе.

В комбинационных узлах можно выделить информационные, адресные и управляющие входы. Управляющие входы помечают меткой E (Enable – разрешение). Информационные и адресные входы помечаются цифровыми метками, которые могут обозначать номер по порядку или вес двоичной переменной, подаваемой на данный вход. Вес указывают или степенью двойки, в которую следует возвести двойку, чтобы получить реальный вес разряда, или действительным значением веса разряда. Примеры двух вариантов обозначения одинаковых входов, которые отличаются весом, показаны на рисунке 2.1.

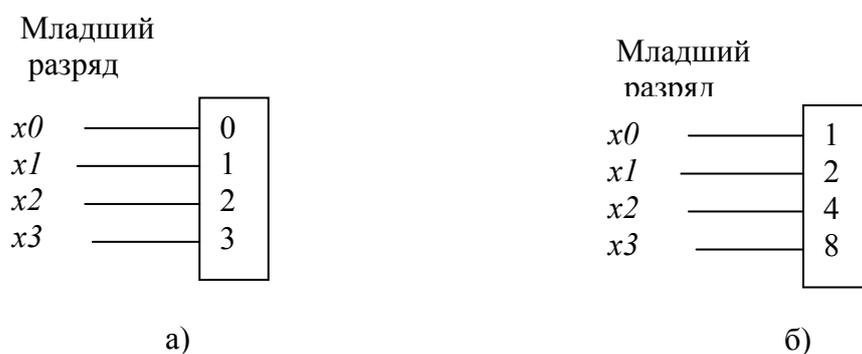


Рис. 2.1. – Варианты обозначения входов с разными весами

Для входов и выходов узла можно определить понятие активного и пассивного сигнала. Под активным понимают тот уровень сигнала, который вызывает определённые действия или представляет интерес. Если активным является уровень единицы, то соответствующий вывод является прямым и изображается так, как показан прямой вход разрешения работы на рисунке 2.2,а. Если активным является нулевой уровень, то соответствующий вывод помечается инвертирующим кружком, как показано на рисунке 2.2,б.

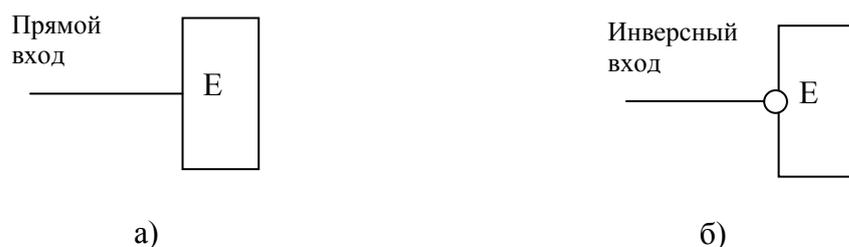


Рис. 2.2.-Примеры изображения входов разрешения

## 2.2. Дешифраторы

Дешифратор - это комбинационный узел или комбинационная схема, имеющая при  $n$  информационных входах до  $2^n$  выходов и осуществляющая преобразование параллельного двоичного числа в унитарный код. Унитарный означает такой способ представления числа, при котором величина числа определяется положением активного сигнала на множестве выходов.

Двоичное число, подаваемое на информационные входы, указывает номер того выхода, на котором наблюдается активный сигнал. То есть входы дешифратора в данном случае являются адресными, определяют адрес выхода с активным сигналом. На остальных выходах наблюдаются в то же время пассивные сигналы. При определении номера выхода надо учитывать тот факт, что входная информация представлена в двоичном виде, а выходы пронумерованы в десятичной системе счисления. Следовательно, для определения номера выхода необходимо с учётом веса каждого информационного разряда осуществить преобразование двоичного числа, подаваемого на информационные входы, в десятичное. Полученное десятичное число и определяет номер выхода, на котором формируется активный сигнал.

Сказанное поясним, рассмотрев дешифратор с числом входов  $n=3$ .

Функционирование этого дешифратора описывается таблицей истинности 2.1.

Анализируя эту таблицу, можно заметить, что на любой комбинации возможных значений входных сигналов только на одном определённом выходе из восьми присутствует единица. Причём, номер этого выхода равен десятичному эквива-

ленту подаваемого на входы  $x_2, x_1, x_0$  двоичного числа в предположении, что  $x_2$  является старшим разрядом, а  $x_0$  – младшим.

Указанный в первом столбце порядковый номер строки является фактически десятичным эквивалентом подаваемого на входы двоичного числа. Действительно, например, в строке с номером №=6 комбинация значений входных сигналов  $x_2, x_1, x_0$  равна 110, что в десятичной системе соответствует числу 6. На указанной строке единичный уровень выходного сигнала будет наблюдаться на выходе  $y_6$ .

Таблица 2.1

№	$x_2$	$x_1$	$x_0$	$y_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

Единичный уровень на выходе является при этом активным, он указывает на значение числа в унитарном представлении. Как будто на оси целочисленных величин, представленной линейным расположением выходов, точкой или штрихом отмечено положение числа. На всех остальных выходах наблюдаются пассивные сигналы, в рассматриваемом случае это логические нули.

Синтезируем дешифратор как комбинационную схему, имеющую 8 выходов. При этом будем считать, что выходные переменные зависят только от входных, что позволяет синтезировать 8 независимых комбинационных схем, каждая из которых формирует один из выходных сигналов:

$$y_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0; \quad y_1 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0; \quad y_2 = \bar{x}_2 \cdot x_1 \cdot \bar{x}_0; \quad y_3 = \bar{x}_2 \cdot x_1 \cdot x_0;$$

$$y_4 = x_2 \cdot \bar{x}_1 \cdot \bar{x}_0; \quad y_5 = x_2 \cdot \bar{x}_1 \cdot x_0; \quad y_6 = x_2 \cdot x_1 \cdot \bar{x}_0; \quad y_7 = x_2 \cdot x_1 \cdot x_0.$$

Эти выражения позволяют построить функциональную схему дешифратора с использованием инверторов для формирования отрицаний входных переменных, и конъюнкторов для реализации функций выходных переменных. Функциональная схема дешифратора для рассмотренного случая предложена на рисунке 2.3. Предложенный дешифратор можно изображать на функциональных схемах двумя способами, предложенными на рисунке 2.4. В условных обозначениях в основном поле на верхней строчке изображается символ выполняемой функции. В нашем случае этим символом является сочетание букв «DC» (Decoder). Подобным же образом дешифратор можно изобразить и на принципиальной схеме, если он реализован в виде микросхемы, добавив у каждого вывода номер контакта микросхемы. В основном поле условного обозначения на принципиальной схеме под символом функции дешифратора ставится тип микросхемы. В названии микросхемы дешифратора после номера серии элементов ставятся буквы «ИД», после которых ставится номер разработки.

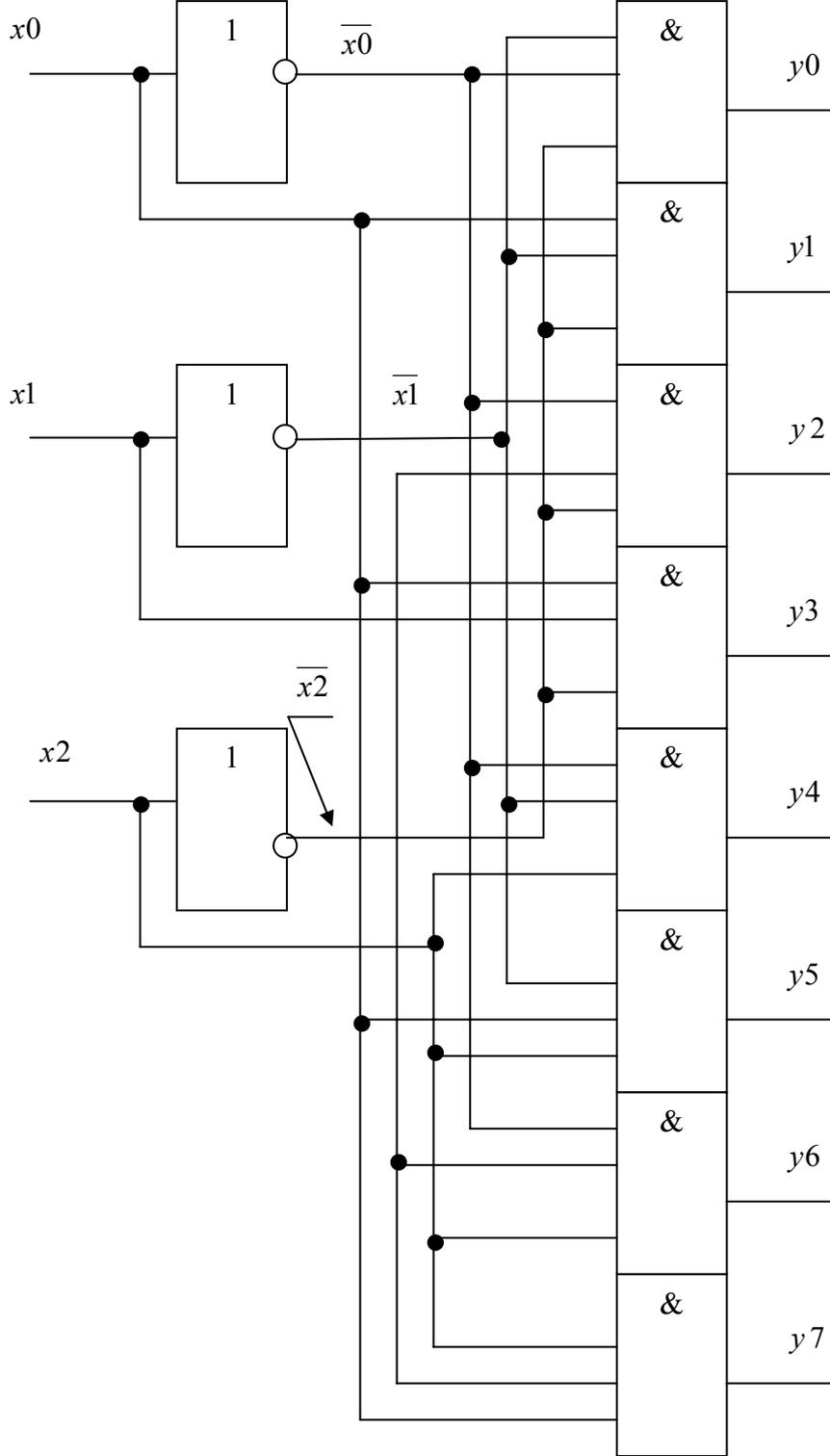


Рис. 2.3. – Функциональная схема дешифратора

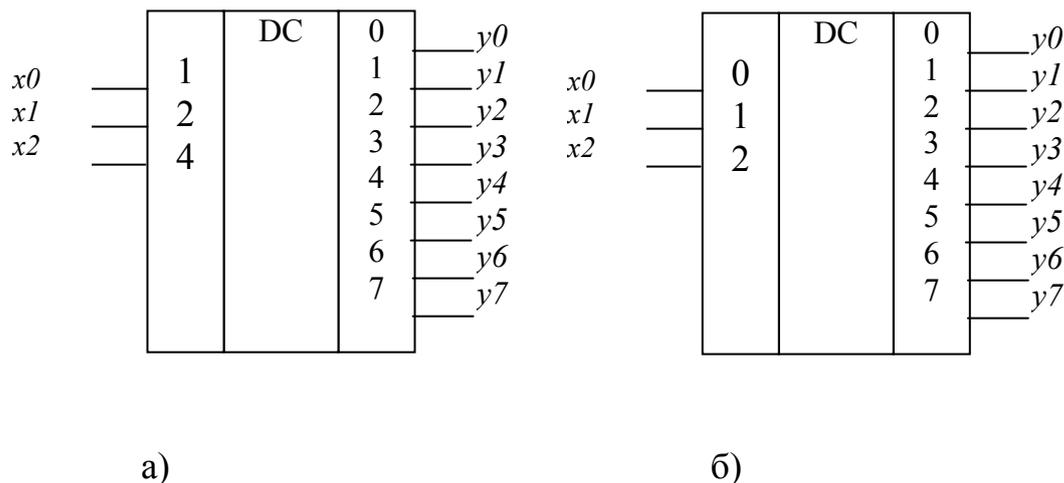


Рис. 2.4. – Возможные способы условного обозначения простейшего дешифратора на функциональной схеме.

Отличие между предложенными вариантами обозначения дешифратора заключается в способе изображения информационных (или адресных, поскольку они определяют номер, то есть адрес выхода) входов. В первом случае входы обозначены своими весами в десятичной системе счисления. Младший разряд входного слова ( $x_0$ ) имеет при этом вес, равный единице, следующий разряд – равный двойке, а самый старший разряд в данном случае имеет вес, равный четырём. Во втором случае входы обозначаются степенью двойки. Чтобы при этом определить вес разряда, необходимо возвести двойку в степень, соответствующую указанному числу. Для входа, обозначенного во втором случае цифрой 2, вес разряда равен  $2^2$ , то есть четырём.

Входы и выходы предложенных дешифраторов являются прямыми, то есть активными сигналами для них являются единицы.

Очень часто современные дешифраторы имеют специальный вход разрешения работы E (Enable). Только при наличии на этом входе активного сигнала дешифратор осуществляет требуемое преобразование, то есть на одном из выходов появится активный сигнал. Если на входе разрешения сигнал пассивный, то на всех выходах дешифратора наблюдается пассивный сигнал. Пусть активным сигналом по входу E является 1, тогда предложенные выше функции, описывающие

поведение дешифратора, следует скорректировать добавлением сигнала E следующим образом:

$$y_0 = E \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0; \quad y_1 = E \cdot \bar{x}_2 \cdot \bar{x}_1 \cdot x_0;$$

$$y_2 = E \cdot \bar{x}_2 \cdot x_1 \cdot \bar{x}_0; \quad y_3 = E \cdot \bar{x}_2 \cdot x_1 \cdot x_0;$$

$$y_4 = E \cdot x_2 \cdot \bar{x}_1 \cdot \bar{x}_0; \quad y_5 = E \cdot x_2 \cdot \bar{x}_1 \cdot x_0;$$

$$y_6 = E \cdot x_2 \cdot x_1 \cdot \bar{x}_0; \quad y_7 = E \cdot x_2 \cdot x_1 \cdot x_0.$$

Используя полученные выражения, можно предложить соответствующее условное изображение дешифратора, у которого активными сигналами на выходах являются единицы (рис. 2.5).

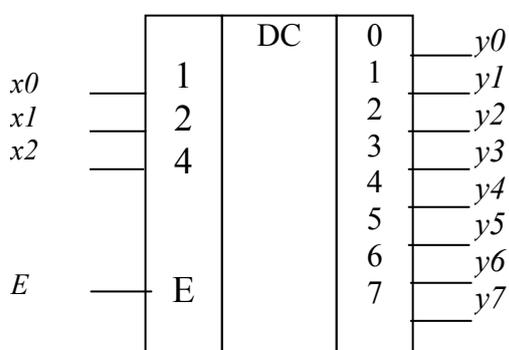


Рис. 2.5. - Условное графическое обозначение дешифратора с активными единичными сигналами на выходах и входе E

На практике на входе E (разрешение работы) может быть реализован конъюнктор с прямыми и инверсными входами, а на выходах активным сигналом также может быть ноль. При этом соответствующие входы и выходы с активными нулями помечаются кружочками инверсии.

Условное изображение дешифратора с активными нулями на выходах и конъюнктором на входе разрешения, предложено на рисунке 2.6. Предложенный на этом рисунке дешифратор соответствует микросхеме дешифратора К555ИД7. Работа дешифратора разрешается только в том случае, когда на входах разрешения будут присутствовать следующие сигналы:

$$E_1=1, E_2=0, E_3=0.$$

То есть по входу разрешения  $E$  реализуется функция:  $E = E1 \cdot \overline{E2} \cdot \overline{E3}$ .

При этом условии в зависимости от сигналов на входах  $x2, x1, x0$  формируется уровень логического нуля на соответствующем выходе.

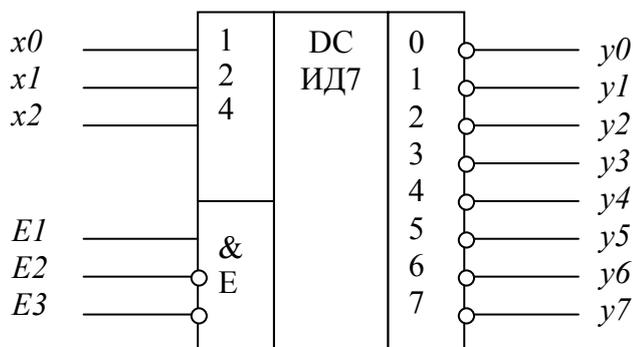


Рис. 2.6. - Условное обозначение дешифратора с активными нулями на выходах и конъюнктом на входе разрешения работы

Рассмотрим способ увеличения количества выходов дешифратора. Пусть в нашем распоряжении имеются полные (число выходов равно  $2^n$  при  $n$  информационных входах) дешифраторы типа  $2 \rightarrow 4$  (два входа – четыре выхода). Необходимо построить дешифратор, который имеет 4 информационных входа и 16 выходов, то есть дешифратор типа  $4 \rightarrow 16$ . Пример построения такого дешифратора и условное обозначение микросхемы, реализующий такой дешифратор, предложены на рисунке 2.7.

В зависимости от состояний сигналов  $x3$  и  $x2$  при наличии на входе разрешения работы  $E$  дешифратора DD1 формируется единица на одном из четырёх выходов этого дешифратора. Это приводит к тому, что только один из выходных дешифраторов будет реагировать на комбинацию сигналов на входах  $x0$  и  $x1$ . Только выбранный дешифратор сформирует единицу на одном из своих выходов, номер которого определяется сигналами  $x0$  и  $x1$ . Например, пусть на входах  $x3x2x1x0$  присутствует число 1011. На входах  $x3x2$  присутствует комбинация 10, что соответствует в десятичном виде числу 2.

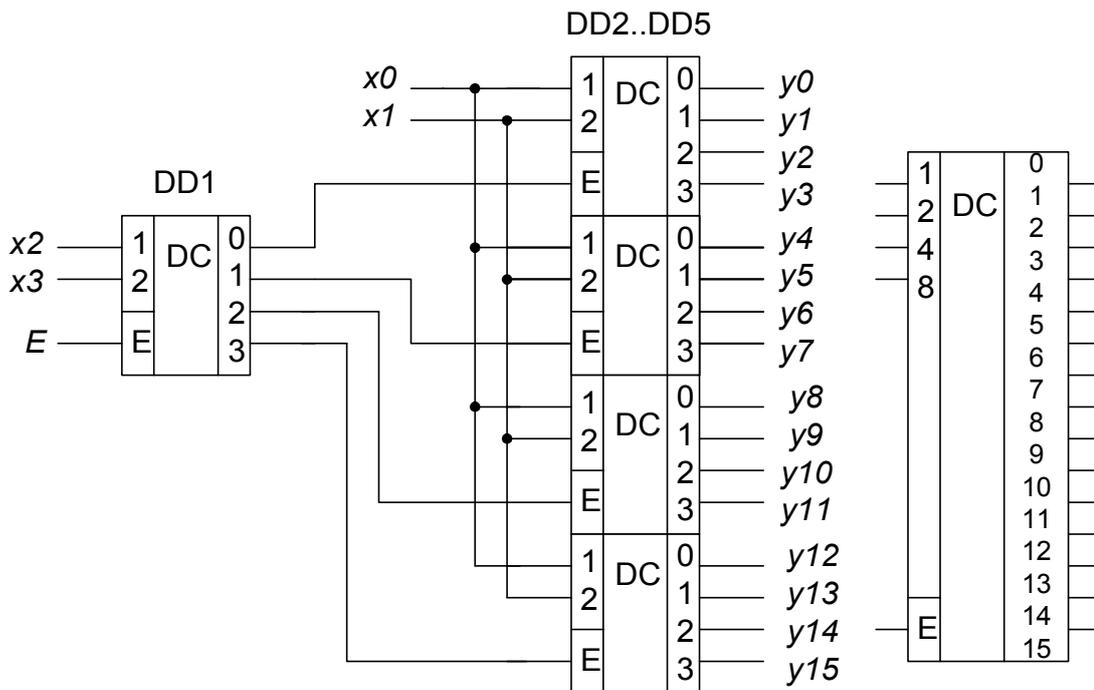


Рис. 2.7. - Способ реализации сложного дешифратора и его условное обозначение

Следовательно, именно на выходе 2 дешифратора DD1 сформируется активный сигнал, равный единице. Только дешифратору DD4, который принимает по входу E активный уровень, будет разрешаться работа. На входах  $x1/x0$  присутствует число 11, что соответствует в десятичном виде числу 3. На третьем выходе выбранного дешифратора DD4 будет формироваться единица, то есть активный сигнал. На остальных выходах выбранного дешифратора будет присутствовать ноль так же, как и на выходах невыбранных дешифраторов DD2, DD3, DD5. То есть только на выходе  $y11$  присутствует активный сигнал. Если перевести заданное двоичное число 1011 в десятичную систему, то получим номер выбранного выхода в десятичной системе: 11. Процедура перевода двоичного числа с учётом весов разрядов предлагается ниже.

$$1011_2 = 2^3 + 2^1 + 2^0 = 11_{10}.$$

Существуют специальные дешифраторы, предназначенные для работы с элементами индикации. В микросхемах ТТЛ существуют дешифраторы, которые способны коммутировать высокое напряжение для управления неоновыми индикаторами. Но в настоящее время широкое применение находят семисегментные индикаторы, выполненные на светодиодах или жидкокристаллических элементах.

Расположение сегментов в таких индикаторах с их буквенными идентификаторами и примеры формирования некоторых десятичных цифр предлагаются на рисунке 2.8.

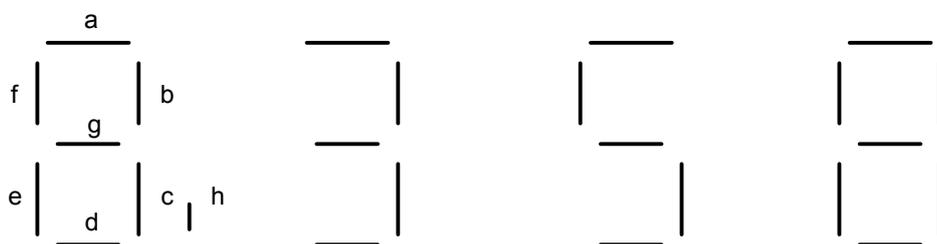


Рис.2.8. – Расположение сегментов в семисегментном индикаторе и варианты формирования символов.

Светодиодные индикаторы различаются способом соединения светодиодов: по схеме с общим анодом и с общим катодом. В зависимости от разновидности эти индикаторы подключаются общим анодом к источнику положительного напряжения величиной  $E=5В$ , а общим катодом к общему проводу. Достоинством светодиодных индикаторов является хорошая видимость в условиях недостаточного внешнего освещения. Для управления подобными индикаторами используют специальные дешифраторы, у которых имеются 4 информационных входа и 7 выходов. Пример подключения светодиодов семисегментного индикатора к специальному дешифратору показан на рисунке 2.9. Предложенный дешифратор способен преобразовать параллельный двоичный код от 0000 до 1001 в соответствующие сигналы управления светодиодами. Заметим, что дешифратор формирует сигналы управления только для сегментов a, b, c, d, e, f, g. Управление запятой h должно осуществляться дополнительными схемами. Резисторы необходимы для ограничения тока управления светодиодами. В некоторых дешифраторах токоограничивающие резисторы располагаются внутри микросхемы.

Предложенная схема работает следующим образом.

При подаче на входы  $x_3, x_2, x_1, x_0$  двоичного эквивалента десятичного разряда дешифратор формирует на соответствующих выходах уровни логического нуля, что вызывает протекание токов через соответствующие светодиоды, которые будут светиться. Пусть на входах присутствует двоичное число, соответ-

вующее десятичной цифре 5. При этом  $x_3x_2x_1x_0=0101$ . Дешифратор сформирует нули на выходах a, c, d, f, g. На светодиодном индикаторе будет светиться цифра 5, как показано на рисунке 2.8.

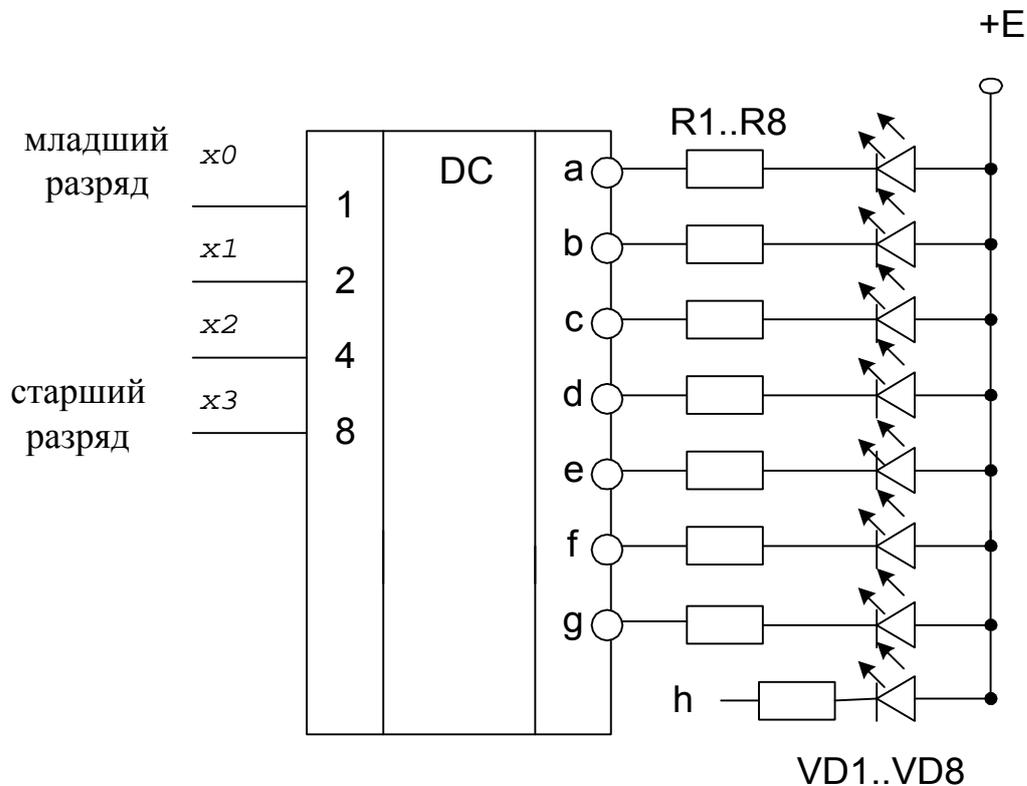


Рис. 2.9. – Схема подключения светодиодов семисегментного индикатора к специальному дешифратору

Если требуется формировать с помощью индикатора более сложные символы, то придётся управлять каждым сегментом индикатора.

Поскольку светодиодный индикатор выполняется как законченное изделие, то на электрических схемах целесообразно использовать его условное графическое обозначение, а не изображать каждый светодиод, как показано на рисунке 2.9. Пример изображения семисегментного индикатора предложен на рисунке 2.10. Звёздочка в символе функции обозначает, что это не логический элемент, буква H является символом индикатора, 7S указывает на число сегментов. Информационные входы обозначены как инвертирующие, то есть активным сигналом на них является нуль. На вход U подаётся питание.

Для управления жидкокристаллическими индикаторами используются аналогичные дешифраторы. Особенностью жидкокристаллических индикаторов является необходимость применения генератора импульсов для обеспечения коммутации полярности подводимого к элементам индикации напряжения. Импульсы с генератора подаются одновременно на общий для индикатора вход питания и в качестве сигнала управления выходными цепями дешифратора. Одновременно в этом случае меняется как состояние питающего сигнала, так и происходит поразрядное инвертирование всех выходных сигналов дешифратора. Такое управление жидкокристаллическим индикатором существенно продляет срок его службы.

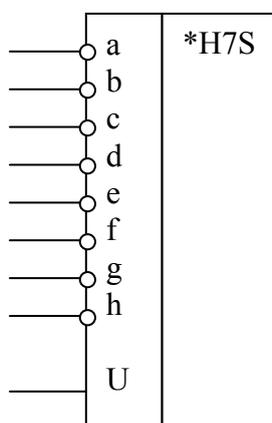


Рис. 2.10. – Условное обозначение семисегментного индикатора

Дешифраторы находят широкое применение. В составе компьютеров, например, дешифраторы позволяют адресоваться к определённому устройству, модулю, с которым в данный момент осуществляется обмен информацией. Чтобы обратиться именно к определённому устройству, достаточно подключить вход разрешения работы этого устройства к определённому выходу дешифратора, а входы дешифратора использовать для задания адреса устройства.

Дешифраторы могут находить и неожиданное применение. Например, дешифратор может быть использован как основной узел цифрового замка. При программировании определённый выход дешифратора через усилитель подключается к электромагниту замка. На лицевую сторону двери ставятся кнопки, которые управляют адресными входами дешифратора и входом разрешения работы. На

адресных входах одновременно нажимаются кнопки, соответствующие выбранному при программировании выходу. Каждая кнопка может переключать вход дешифратора с уровня логического нуля на уровень логической единицы и наоборот. Можно упростить кнопки путём предварительной подачи с помощью резисторов на каждый вход определённого уровня, например, уровня единицы. Тогда нажатие кнопки приводит к подаче на соответствующий вход уровня нуля. Последней нажимают кнопку, которая подаёт на вход разрешения работы дешифратора активный сигнал. Если при этом комбинация нажатых кнопок правильная, то на электромагнит поступает ток, который заставит электромагнит открыть дверь. Если же набрана ошибочная комбинация, то можно включить специальную сирену. Условием включения сирены может быть отсутствие активного сигнала на входе усилителя электромагнита при наличии активного сигнала на входе разрешения дешифратора. То есть достаточно к дешифратору добавить двухвходовой конъюнктор для подключения сирены. Выбор выхода дешифратора с  $n$  адресными входами при программировании можно осуществить, например, с помощью переключателя на  $2^n$  положений.

*(Попробуйте предложить функциональную схему электронного замка на дешифраторе с четырьмя адресными входами и входом разрешения с активным нулём, подключив усилитель электромагнита к выбранному вами выходу с требуемой комбинацией сигналов).*

С помощью дешифратора легко реализовать бегущий огонь на цепи светодиодов или ламп накаливания, которые через усилители подключаются к выходам дешифратора. Если на входы дешифратора подавать импульсы с частотами  $f$ ,  $f/2$ ,  $f/4$ , ..., то активный сигнал поочерёдно будет появляться на выходах  $0$ ,  $1$ ,  $2$ ,  $3$ , ... Временные диаграммы входных сигналов в предложенном случае будут подобны входным сигналам  $x_3$ ,  $x_2$ ,  $x_1$ , показанным на рисунке 1.4.

*(Подумайте, на какой вход дешифратора надо подавать сигналы с максимальной частотой, а на какой вход следует подавать минимальную частоту, чтобы активный сигнал появлялся на выходах в указанном порядке).*

### 2.3. Шифраторы

Шифраторы осуществляют преобразование унитарного кода в параллельный двоичный код, то есть они выполняют операцию, обратную дешифрированию.

При  $n$  выходах простейший полный шифратор должен иметь  $2^n$  входов. Можно предполагать, что активный сигнал поступает только на один из входов в данный момент времени. Обратим внимание на две особенности в работе шифратора.

Во-первых, каждому входу с активным сигналом соответствует определённая комбинация выходных сигналов. Но ведь может быть случай, когда на всех входах присутствуют пассивные сигналы. Но при этом на выходах будет присутствовать какая-то определённая комбинация, которая соответствует наличию активного сигнала на определённом входе. Окажется, что наличие любого сигнала на этом входе приводит к одной и той же комбинации выходных сигналов, что приводит к нарушению логики работы узла. Эта проблема решается двумя способами. Можно исключить соответствующий вход у шифратора и учитывать соответствующее состояние выходных сигналов как исходное или соответствующее активному состоянию сигнала на отсутствующем входе. Такое решение используется в микросхеме К555ИВ3, выполненной по технологии ТТЛ. Другое решение возможно путём добавления специального выхода группового переноса (GS), активный сигнал на котором появляется только в том случае когда хотя бы на один из входов поступает активный сигнал. При таком решении считывание информации с выходов шифратора осуществляется лишь при наличии активного сигнала на выходе группового переноса, что гарантирует однозначное преобразование унитарного кода в параллельный двоичный код.

Во-вторых, одновременно на нескольких входах может присутствовать активный сигнал, что должно приводить к строго определённому коду на выходах. Учитывая такую возможность, создают приоритетные шифраторы, в которых все входы оказываются с разными приоритетами. Тогда при одновременной подаче активных сигналов на несколько входов шифратор формирует на выходах код входа с максимальным приоритетом.

В шифраторах также необходимо учесть возможность расширения числа информационных входов за счёт одновременного использования нескольких шифраторов, включенных определённым образом. При этом целесообразно добавить вход разрешения работы данного шифратора (EI - enable input) и выход разрешения работы для других шифраторов (EO - enable output). С учётом всех замечаний рассмотрим работу приоритетного шифратора КМ555ИБ1 [5], функционирование которого описывается таблицей 2.2, а условное обозначение этого шифратора предложено на рисунке 2.11.

Таблица 2.2

Входы									Выходы				
EI	$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	GS	$y_2$	$y_1$	$y_0$	EO
1	b	b	b	b	b	b	b	b	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	0	1	1	1	1
0	1	1	1	1	1	1	0	b	0	1	1	0	1
0	1	1	1	1	1	0	b	b	0	1	0	1	1
0	1	1	1	1	0	b	b	b	0	1	0	0	1
0	1	1	1	0	b	b	b	b	0	0	1	1	1
0	1	1	0	b	b	b	b	b	0	0	1	0	1
0	1	0	b	b	b	b	b	b	0	0	0	1	1
0	0	b	b	b	b	b	b	b	0	0	0	0	1

Символ b указывает на то, что состояние соответствующего сигнала не имеет значение. Обратите внимание на то, что по всем входам и выходам активным сигналом является уровень логического нуля. Таблица функционирования (истинности) шифратора позволяет при необходимости синтезировать полную схему на элементах типа И-НЕ, ИЛИ-НЕ. Обратим внимание только на особенности формирования выходных сигналов группового переноса GS и сигнала разрешения на выходе EO. Для сигнала группового переноса справедливо:

$$GS = EI \vee \overline{EI} \cdot x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7.$$

Сигнал разрешения на выходе ( $EO$ ) равен нулю только на одном наборе входных переменных. Обратим внимание именно на этот набор, записав конституенту единицы для него:

$$k = \overline{EI} \cdot x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7.$$

Тогда выходной сигнал  $EO$  будет равен:

$$EO = \overline{k} = \overline{\overline{EI} \cdot x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7}.$$

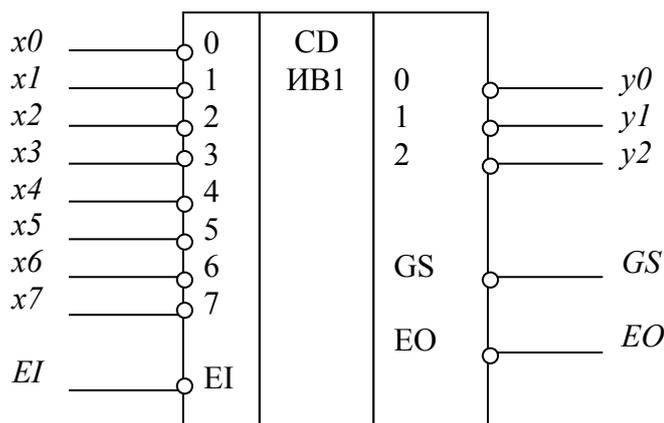


Рис. 2.11. – Условное обозначение шифратора

Шифраторы могут быть использованы для формирования кода нажатой клавиши, то есть при организации клавиатуры. Если количество клавиш не превышает число информационных входов выбранного шифратора, то схема клавиатуры окажется простой и предложена на рисунке 2.12. При необходимости увеличения числа клавиш выход EO следует подключить к входу EI следующей микросхемы, как указано на рисунке.

Клавиатура содержит восемь клавиш SB1-SB8, восемь резисторов R1-R8 и один шифратор на микросхеме КМ555ИВ1. Благодаря тому, что вход EI подключен к общему проводу, на этом входе присутствует логический нуль, разрешается работа микросхемы шифратора. То есть шифратор будет реагировать на подачу нуля на информационные входы.

Работает клавиатура следующим образом.

При отсутствии нажатых клавиш на всех информационных входах присутствуют единичные уровни благодаря тому, что входы микросхемы соединены с источником питания через резисторы. Сопротивления резисторов выбираются равными нескольким килоомам (2КОма). При нажатии, например, на клавишу SB1 подаётся нуль на нулевой вход шифратора, что приводит к формированию на выходе группового переноса GS активного уровня, то есть нуля. В то же время

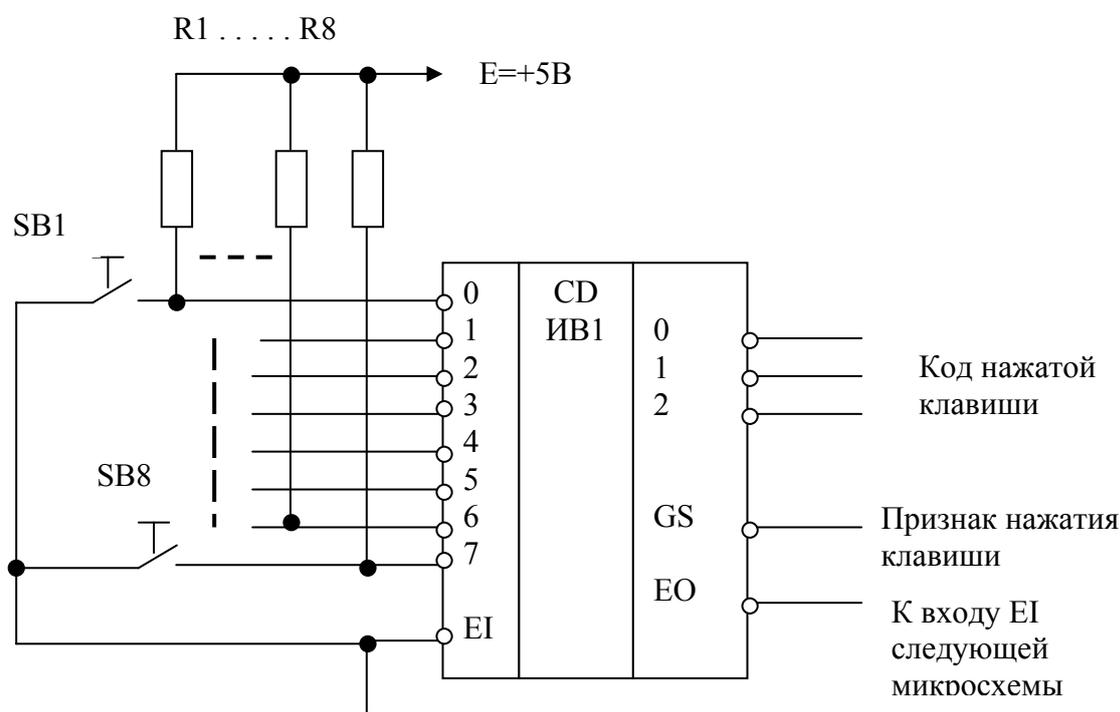


Рис. 2.12. – Способ организации простейшей клавиатуры

на выходе EO формируется пассивный сигнал, равный уровню единицы, что блокирует по входу работу тех микросхем, которые подключены к данной микросхеме. Таким образом реализуется максимальный приоритет у клавиш, подключенных к данному шифратору. На информационных выходах формируется код нажатой клавиши. Если нажать какую-либо другую клавишу, то на информационных выходах соответствующим образом изменится код. У входа с большим номером приоритет выше. То есть, если нажаты, например, одновременно клавиши на входах 4 и 5, на выходах сформируется код  $y_2y_1y_0 = 011$ . Это соответствует коду четвёртого входа. (Обратите внимание на различие в нумерации клавиш и входов). Если предполагается ввод кода нажатой клавиши в микропроцессорную систему,

то признак нажатия клавиши может быть использован как сигнал прерывания работы системы. По этому сигналу система должна принять информацию с клавиатуры.

При использовании нескольких микросхем шифраторов возникнет необходимость разумным образом с помощью дополнительных логических элементов сформировать выходной код, который соответствовал бы нажатой клавише. Признак нажатой клавиши также потребует применения дополнительной логики.

## 2.4. Мультиплексоры

В средствах вычислительной техники при передаче цифровой информации возникает проблема приёма информации с нескольких направлений в разные интервалы времени. Эта задача решается применением мультиплексоров.

Мультиплексор - это электронный коммутатор, осуществляющий передачу сигнала с одного из  $2^n$  входов под управлением  $n$ -разрядного адреса этого входа на единственный выход. Адресные входы обозначают комбинацией символов SED (селекция данных). Кроме адресных и информационных входов мультиплексор может иметь вход разрешения работы  $E$ . Выход мультиплексора может быть организован по схеме с тремя состояниями [6], в этом случае вход разрешения управляет состоянием выходной цепи. Рассмотрим мультиплексор типа  $8 \rightarrow 1$  (восемь входов, один выход), условное обозначение которого предложено на рисунке 2.13. На этом рисунке заданы сигналы на входах селекции и управления. Комбинация сигналов на входах селекции соответствует десятичной цифре 5, то есть мультиплексор передаёт на выход информацию с входа  $x_5$ . Мультиплексор реализует функцию:

$$y = E \cdot (\overline{S_2} \cdot \overline{S_1} \cdot \overline{S_0} \cdot x_0 \vee \overline{S_2} \cdot \overline{S_1} \cdot S_0 \cdot x_1 \vee \overline{S_2} \cdot S_1 \cdot \overline{S_0} \cdot x_2 \vee \overline{S_2} \cdot S_1 \cdot S_0 \cdot x_3 \vee S_2 \cdot \overline{S_1} \cdot \overline{S_0} \cdot x_4 \vee S_2 \cdot \overline{S_1} \cdot S_0 \cdot x_5 \vee S_2 \cdot S_1 \cdot \overline{S_0} \cdot x_6 \vee S_2 \cdot S_1 \cdot S_0 \cdot x_7).$$

При  $E=1$  предложенная функция соответствует таблице истинности 2.3.

*(Осознайте справедливость таблицы, её соответствие функции, которая реализуется мультиплексором).*

Логическую функцию, которую выполняет мультиплексор, можно реализовать по предложенному выше выражению с использованием многовходовых конъюнкторов и восьмивходового дизъюнктора. Если функцию селекции (выбора) входа, с которого передаётся в данный момент информация на выход, реализовать с помощью дешифратора, то мультиплексор легко реализуется по схеме, предложенной на рисунке 2.14.



Рис. 2.13. – Условное обозначение мультиплексора

Таблица 2.3

№	$S_2$	$S_1$	$S_0$	$Y$
0	0	0	0	$x_0$
1	0	0	1	$x_1$
2	0	1	0	$x_2$
3	0	1	1	$x_3$
4	1	0	0	$x_4$
5	1	0	1	$x_5$
6	1	1	0	$x_6$
7	1	1	1	$x_7$

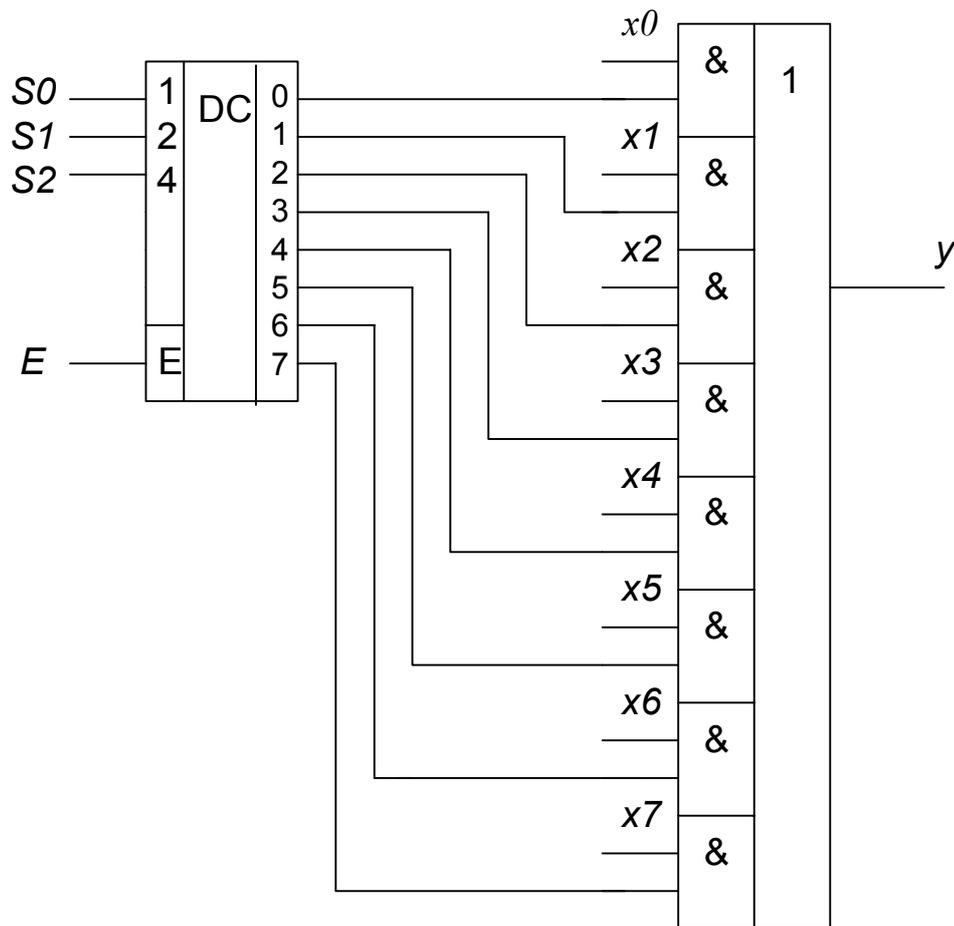


Рис. 2.14. - Функциональная схема мультиплексора

Благодаря применению элемента И-ИЛИ и применению управляющего дешифратора выходной сигнал будет определяться состоянием строго определенного входного сигнала, адрес которого поступает на входы селекции. Например, при  $E=1$  и  $S_2S_1S_0=110$  на шестом выходе дешифратора формируется 1, которая разрешает прохождение на выход переменной с входа  $x_6$ , то есть  $y=x_6$ .

## 2.5. Демultipлексор

Осуществляет операцию, обратную операции мультиплексора, то есть с единственного информационного входа передаёт сигнал на один из  $2^n$  выходов. Адрес этого выхода определяется комбинацией сигналов на адресных входах, которые принимают  $n$ -разрядное двоичное число. В виде микросхемы, специально предназначенной для этого, демultipлексор не реализуется. Демultipлексор реализу-

ется на дешифраторе, имеющем вход разрешения работы  $E$ . Пример реализации демультиплексора на основе дешифратора предлагается на рисунке 2.15.

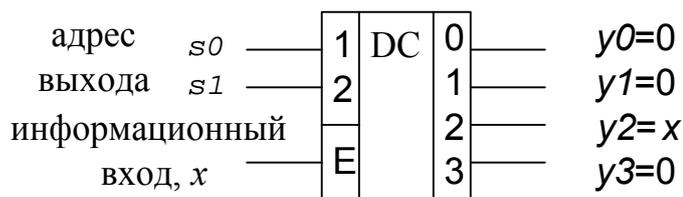


Рис. 2.15. -Демультиплексор на дешифраторе

При реализации демультиплексора на дешифраторе вместо символа функции дешифратора  $DC$  может быть использован символ функции демультиплексора:  $DMX$ . На рисунке 2.15 предполагается, что на входы  $s1s0$  поступает двоичное число 10. При этом вход  $s1$  является старшим разрядом адреса выхода, на который поступает сигнал с входа  $x$ . Для предложенной комбинации адресных сигналов входной сигнал  $x$  проходит на выход  $y2$ .

## 2.6. Схемы сравнения кодов

Схемы сравнения (или компараторы) кодов предназначены для сравнения двух двоичных чисел и формирования результата сравнения в виде цифровых сигналов. Различают сравнение на равенство и на неравенство.

При сравнении на равенство осуществляется поразрядное сравнение двух чисел, что позволяет затем сформировать на выходе всей схемы активный сигнал при равенстве входных чисел. Функционирование схемы по каждому разряду подчиняется таблице истинности, предложенной в таблице 2.4. В этой таблице  $a_i, b_i$  являются входными переменными, одноимёнными разрядами многоразрядных двоичных чисел  $A$  и  $B$ ,  $z_i$  является выходной переменной для разряда с номером  $i$ .

Таблица 2.4

$a_i$	$b_i$	$z_i$
0	0	1
0	1	0
1	0	0
1	1	1

Составим СДНФ выходной функции  $z_i$ , используя эту таблицу:

$$z_i = \overline{a_i} \cdot \overline{b_i} \vee a_i \cdot b_i.$$

Многоразрядные двоичные числа будут равны, если выполняется равенство в каждом разряде. То есть  $z_i = 1$  для каждого разряда. Чтобы сформировать окончательный результат сравнения многоразрядных чисел достаточно выполнить логическое умножение:

$$y = z_1 \cdot z_2 \cdot z_3 \cdots z_n,$$

где  $n$  – число разрядов в сравниваемых числах,  $y$  – результат сравнения.

Только при поразрядном равенстве выходной сигнал  $y$  будет равен логической единице.

В интегральном исполнении, то есть в виде микросхем изготавливают элементы, реализующие функцию «исключающее ИЛИ». Эти элементы реализуют функцию:  $d_i = \overline{a_i} \cdot b_i \vee a_i \cdot \overline{b_i}$ . Можно заметить справедливость равенства:  $d_i = \overline{z_i}$ .

На рисунке 2.16 предложена схема сравнения на равенство, использующая элементы «исключающее ИЛИ».

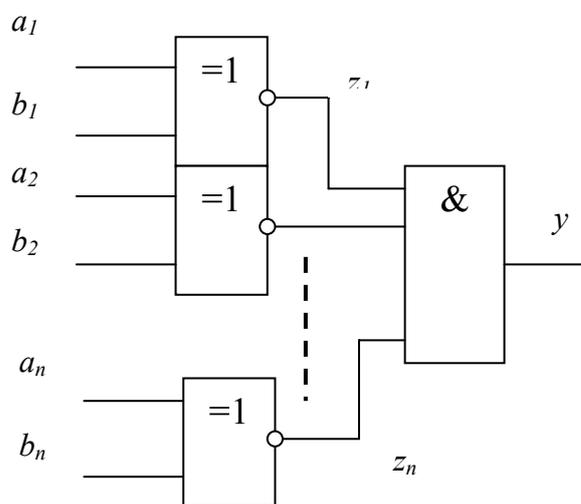


Рис. 2.16. – Схема сравнения на равенство

Схема сравнения на неравенство реализуется в виде микросхемы и имеет кроме информационных входов три входа для стыковки с предыдущей микросхемой и 3 выхода, которые являются выходами всей многоразрядной схемы сравнения или используется для формирования входных сигналов следующей микросхе-

мы. Условное обозначение компаратора четырёхразрядных двоичных чисел, который реализуется в виде микросхемы во многих сериях элементов, предложено на рисунке 2.17.

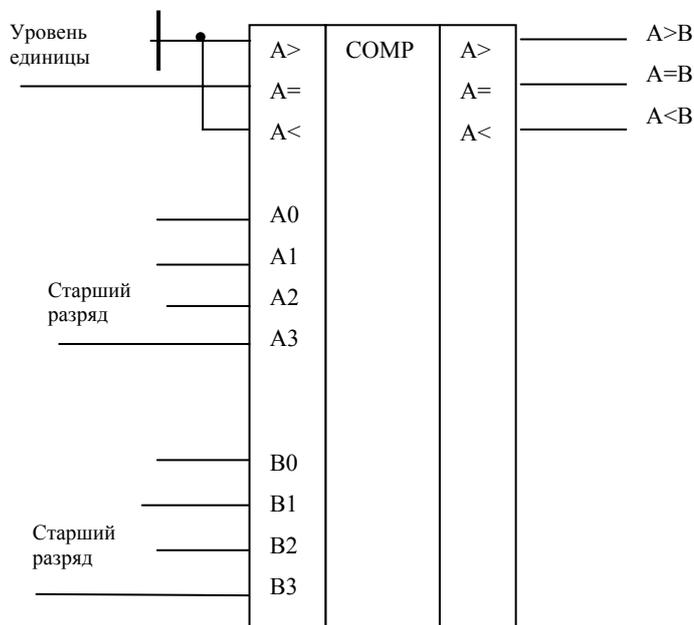


Рис. 2.17. – Условное обозначение четырёхразрядного компаратора кодов

Сравнение многоразрядных чисел выполняют, начиная со старших разрядов входных чисел: А и В. Предполагается равенство предыдущих, отсутствующих старших разрядов, поэтому на вход « А= » подаётся активный сигнал (единица), а на входы « А>, А< » подаётся пассивный сигнал, в нашем случае логический уровень. Если сравниваются числа с разрядностью более четырёх, то выходы компаратора старших разрядов подключаются к одноимённым входам компаратора младших разрядов сравниваемых чисел. Выходами всего многоразрядного компаратора кодов являются выходы компаратора самых младших сравниваемых разрядов.

## 2.7. Комбинационные сумматоры

Комбинационная схема, предназначенная для сложения двоичных чисел, называется сумматором. Сумматор любой разрядности обычно строится на одноразрядных полных сумматорах, которые имеют три входа и два выхода. На входы подаются сигналы соответствующих разрядов суммируемых чисел ( $a_i, b_i$ ) и сигнал

переноса  $p_i$  из предыдущего, младшего разряда с номером  $(i-1)$ . На одном из выходов формируется сумма в данном разряде  $s_i$ , а на другом выходе формируется перенос в следующий разряд  $p_{i+1}$ .

Синтезируем схему одноразрядного сумматора. Для этой цели воспользуемся таблицей истинности, предложенной в таблице 2.5.

Таблица 2.5

$a_i$	$b_i$	$p_i$	$p_{i+1}$	$S_i$
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

Для сигнала переноса  $P_{i+1}$  легко заполнить диаграмму Вейча и выполнить минимизацию. *(Выполните минимизацию самостоятельно или с помощью диаграммы Вейча, или используя склеивание для совершенной дизъюнктивной нормальной формы, записанной с учётом приведённой таблицы истинности.)*

В результате получим функцию переноса:

$$p_{i+1} = a_i b_i \vee b_i p_i \vee p_i a_i.$$

Сумма, как функция трёх переменных  $s_i = f(a_i, b_i, p_i)$ , минимизации не подлежит. Вы можете убедиться в этом, составив диаграмму Вейча для трёх переменных. Но поскольку функция  $p_{i+1}$  легко реализуется, то можно воспользоваться полученным результатом для формирования суммы. Рассматривая сумму  $s_i$  как функцию четырех переменных:  $a_i, b_i, p_i, p_{i+1}$ , легко составить диаграмму Вейча для этой функции как частично заданной и, доопределив её разумным образом, выполнить минимизацию функции. Можно не выполнять минимизации и не анализировать диаграмму Вейча, если обратить внимание на тот факт, что сумма на большинстве наборах значений переменных равна инверсии переноса  $p_{i+1}$ . Только на нулевом наборе требуется сформировать ноль, а на последнем наборе, где все

входные переменные равны единице, необходимо обеспечить единицу на выходе переноса  $p_{i+1}$ . Требуемый ноль формируется с помощью дизъюнкции входных переменных:  $(a_i \vee b_i \vee p_i)$ . Требуемая единица формируется с помощью конъюнкции:  $(a_i \cdot b_i \cdot p_i)$ . В результате функция переноса оказывается следующей:

$$s_i = \overline{p_{i+1}} \cdot (a_i \vee b_i \vee p_i) \vee a_i \cdot b_i \cdot p_i.$$

Полученное выражение можно преобразовать к виду:

$$s_i = \overline{a_i p_{i+1}} \vee \overline{b_i p_{i+1}} \vee \overline{p_i p_{i+1}} \vee a_i b_i p_i.$$

Функциональная схема одноразрядного сумматора, построенная с использованием полученных выражений, предложена на рисунке 2.18.

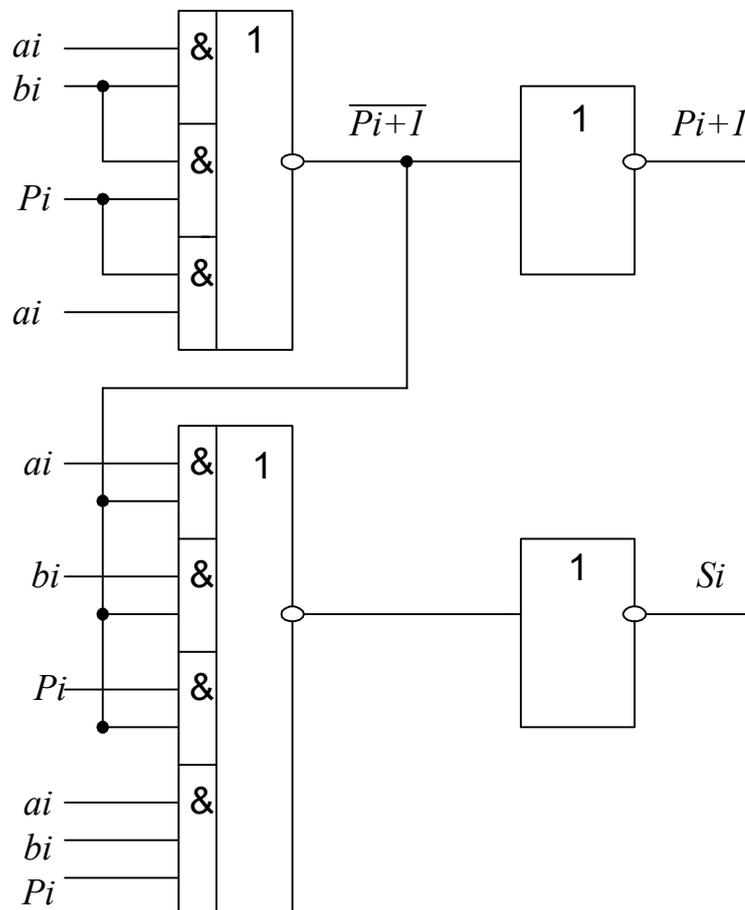


Рис. 2.18. – Схема одноразрядного сумматора

Многоразрядный сумматор на основе одноразрядных полных сумматоров строится по схеме, предложенной на рисунке 2.19.

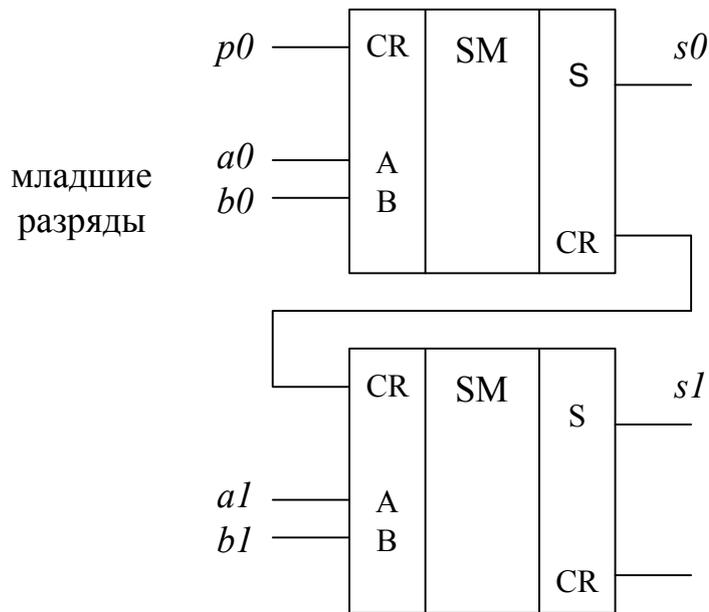


Рис. 2.19. – Схема увеличения разрядности сумматора

В сериях микросхем, выполненных по технологии ТТЛ и КМДП, имеются четырёхразрядные сумматоры, для которых справедливо условное обозначение, предложенное на рисунке 2.20. В элементах ТТЛ этому обозначению соответствует микросхема К555ИМ3.

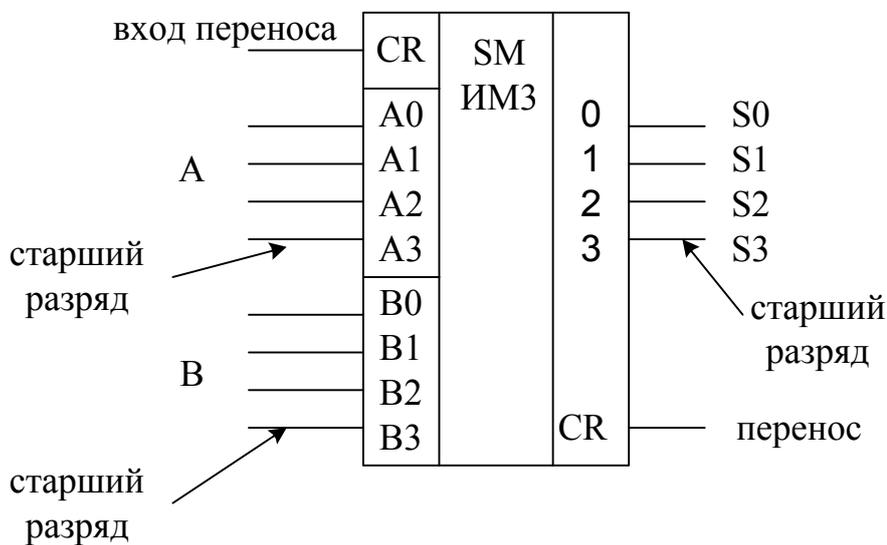


Рис. 2.20. – Условное обозначение четырёхразрядного сумматора

## 2.8. Арифметико-логическое устройство

Арифметико-логическое устройство представляет собой комбинационную схему, которая способна выполнять ряд арифметических и поразрядных логических операций над многоразрядными входными числами. В арифметико-логическом устройстве существуют информационные входы для чисел А и В, вход определения режима работы (М) и входы селекции операций (S3,S2,S1,S0). Вход М определяет, какая будет выполняться операция: арифметическая или логическая. Входы селекции определяют разновидность операций в заданном классе. Символом функции арифметико-логического устройства на схемах является сочетание букв: ALU.

### Контрольные вопросы

1. Какие узлы называют комбинационными?
2. Назовите основные комбинационные узлы, которые рассматривались в данном разделе.
3. Какой стандартный узел имеет при  $n$  входах до  $2^n$  выходов и опишите его функционирование.
4. Какие разновидности входов могут присутствовать у комбинационных узлов и каково их назначение?
5. Что такое мультиплексор и как он функционирует?
6. Что такое шифратор и как его можно использовать для организации клавиатуры?
7. Зачем нужен выход «групповой перенос» в шифраторе?
8. Что такое демультимплексор и каким образом его можно реализовать на дешифраторе?
9. Какие входы и выходы имеются у микросхемы компаратора цифровых сигналов?
10. Объясните принцип построения сложного дешифратора на основе множества простых дешифраторов.
11. Объясните принцип построения сложного мультиплексора на основе множества простых мультиплексоров.

12. Как обозначаются информационные входы у дешифратора и входы селекции у мультиплексора?
13. Зачем нужен вход «Enable» у дешифратора и мультиплексора и как эти узлы будут работать при пассивном сигнале на этом входе?
14. Какие схемы соединения светодиодов в индикаторах вы знаете, как отличаются способы управления ими?
15. Зачем используют последовательно со светодиодами резисторы?

### 3. Триггеры

#### 3.1. Общие сведения

Триггер - электронная схема с двумя устойчивыми состояниями. Схема триггера на электронных лампах была предложена в 1918 году русским учёным М.А.Бонч-Бруевичем. Одно состояние триггера принимается за нулевое, другое состояние принимается за единичное. Обычно триггер строится на основе двух инверторов, выход одного из которых считается прямым выходом триггера, а выход другого инвертора считается инверсным. Состояние прямого выхода определяет состояние триггера. Символом триггера на схемах является буква Т. Простейшая схема триггера на инверторах и условное обозначение триггера на электрических схемах предложено на рисунке 3.1.

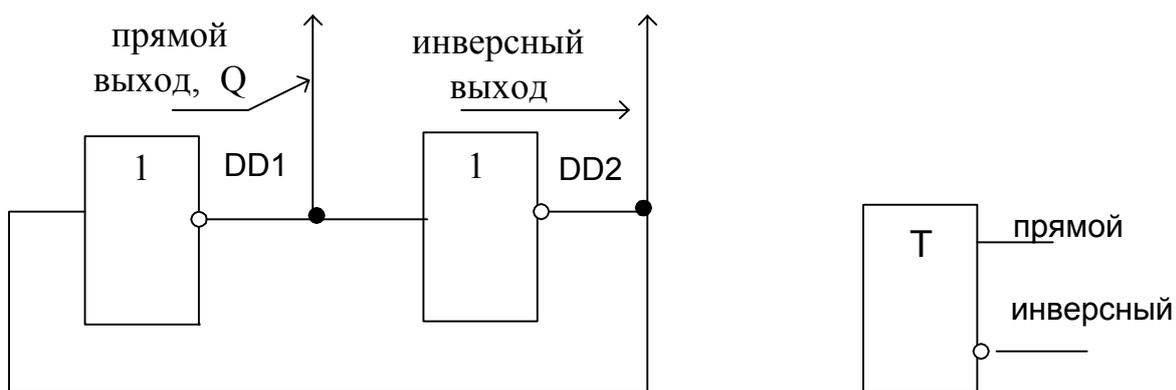


Рис. 3.1. – Простейший триггер и его условное обозначение

Можно заметить, что данная схема действительно способна хранить два состояния. Если на выходе первого инвертора DD1 наблюдается нулевой уровень, то это приводит к появлению на выходе второго инвертора DD2 уровня логической единицы благодаря инвертирующей способности инвертора. Единица с выхода второго инвертора, поступая на вход первого инвертора, поддерживает на его выходе нулевой уровень. В таком состоянии схема может находиться сколь угодно долго, если подаётся питание на инверторы. При включении питания симметричная схема, в которой параметры инверторов одинаковы, с равной вероятностью примет нулевое или единичное исходное состояние.

Поскольку схема не содержит входов, то возможно управление состоянием триггера по выходам. Предположим, что триггер находится в единичном состоянии, то есть на выходе инвертора DD1 присутствует уровень единицы. Для изменения состояния триггера на нулевое достаточно замкнуть прямой выход на общий провод электронным ключом или проволочной перемычкой. В результате на выходе DD1 образуется нулевой уровень благодаря действию перемычки, хотя сам элемент старается сформировать уровень единицы. То есть образуется короткое замыкание на выходе элемента DD1. Образованный нулевой уровень инвертируется элементом DD2. Это приводит к появлению на выходе DD2 спустя время задержки элемента (несколько наносекунд) уровня логической единицы. Реагируя на этот уровень, инвертор DD1 спустя время задержки сформирует на прямом выходе триггера, то есть на своём выходе уровень логического нуля. Состояние короткого замыкания, с которого начинался процесс переключения триггера, закончится. Триггер перешёл в другое состояние за время задержки сигнала в двух инверторах. Кратковременное короткое замыкание на выходе DD1 не приведёт к катастрофе, процесс переключения триггера завершён, перемычку можно снимать, триггер в новом состоянии.

Перевести триггер в противоположное состояние можно и подачей уровня единицы на соответствующий выход. Процесс переключения схемы будет аналогичен рассмотренному случаю. Заметим, что время действия внешнего сигнала для устойчивого переключения триггера в противоположное состояние должно быть не менее суммы времён задержки сигнала в инверторах, образующих триггер.

Поведение триггера описывается матрицей или таблицей переходов. Различают полную и сокращенную таблицу переходов. В полной таблице переходов определяется последующее состояние триггера  $Q(t+1)$  в зависимости от состояний входных информационных сигналов и предыдущего состояния триггера:

$$Q(t+1) = f(Q(t), x_i(t)).$$

Сокращенная таблица переходов не принимает в расчёт предыдущее состояние триггера, поскольку поведение триггера полностью определяется состоянием входных сигналов. У триггера различают информационные входы и входы синхронизации.

ции, помечаемые символом С (от слова Clock). По каждому входу определяют понятие активного сигнала. Активным сигналом может быть уровень (ноль или единица) или фронт сигнала (нарастающий или спадающий). Активный сигнал определяет поведение триггера в следующий момент времени.

Если триггер не имеет входов синхронизации, то его называют асинхронным. В этом случае его поведение однозначно определяется в момент прихода активного сигнала по информационному входу. В зависимости от условного обозначения входа триггер будет изменять своё состояние или под действием уровня входного сигнала или под действием фронта этого сигнала. Сказанное поясняется рисунком 3.2. На этом рисунке символом S помечен информационный вход триггера. Естественно, что каждому условному обозначению триггера будет соответствовать и схемная реализация триггера, то есть внутренняя структура, схема триггера. Информационные входы имеют буквенные идентификаторы, которые соответствуют их назначению.

Если триггер имеет хотя бы один вход синхронизации, то он считается синхронным. У такого триггера имеются информационные входы, приём информации по которым происходит в момент активного синхросигнала. Независимо от этих входов триггер может иметь и другие информационные входы, которые асинхронно или синхронно определяют поведение триггера.

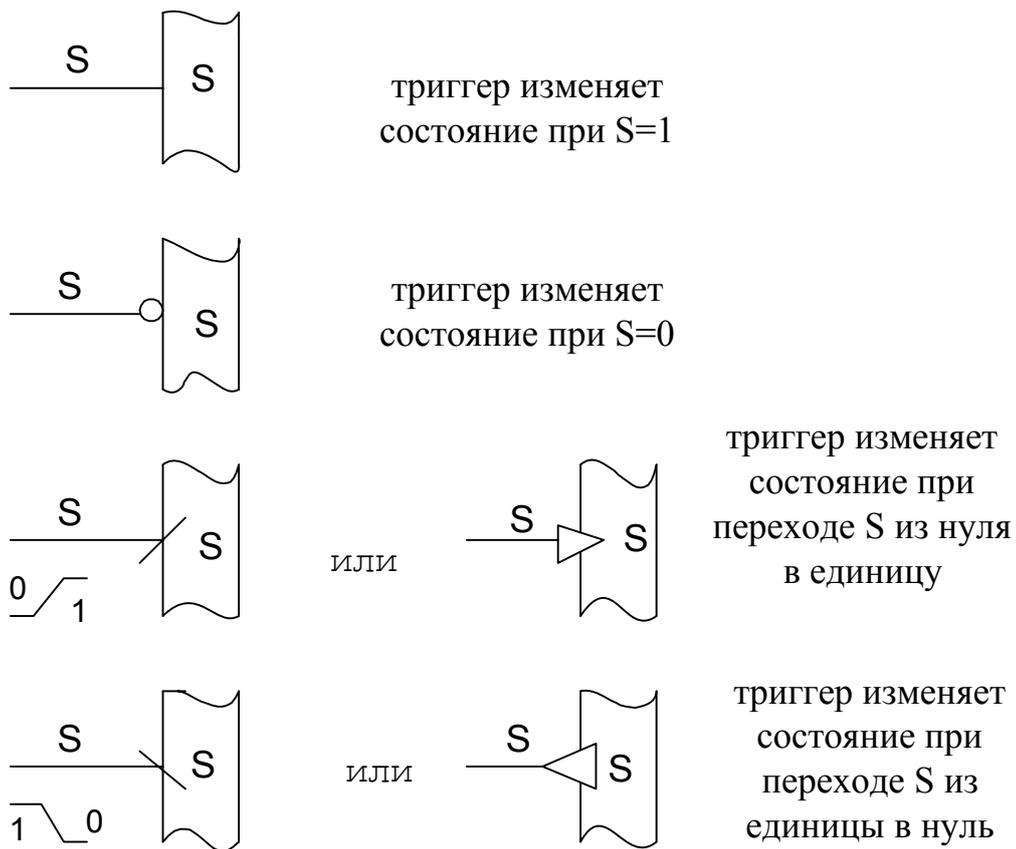
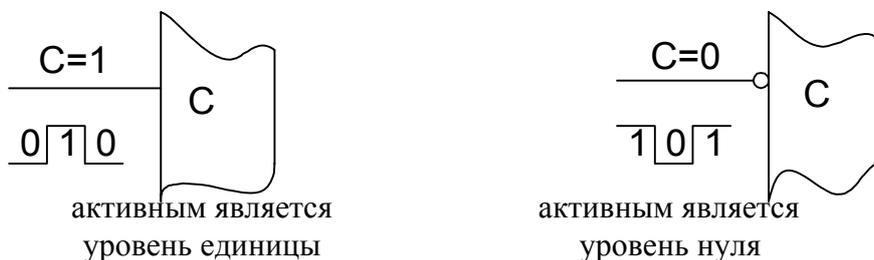
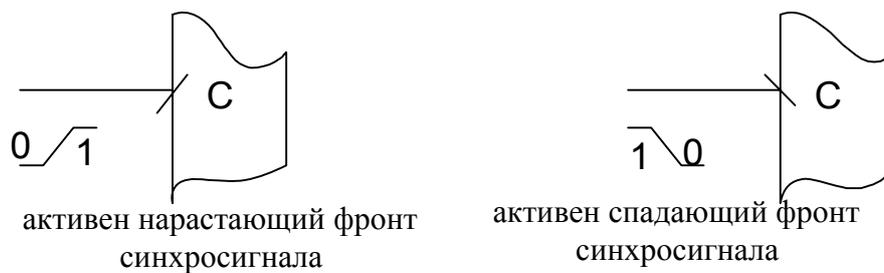


Рис. 3.2. – Примеры условных обозначений входа и условия изменения состояния триггера

В зависимости от того, какой сигнал является активным по входу синхронизации, различают триггеры потенциальные (со статическим управлением) или управляемые по уровню и триггеры с динамическим управлением или срабатывающие по фронту. Условные обозначения входов синхронизации в зависимости от того, какая часть синхросигнала активна, предложены на рисунке 3.3.



*Потенциальное управление*



### *Динамическое управление*

Рис. 3.3. – Условные обозначения входов синхронизации триггеров

В некоторых триггерах под действием определённой комбинации информационных сигналов на прямом и инверсном выходах наблюдаются одинаковые состояния сигналов. В этом случае нарушается логика работы триггера. Такое состояние триггера называется запрещённым и соответствующая комбинация входных сигналов запрещена. При построении временных диаграмм, если поведение триггера анализируется по его условному обозначению, запрещённое состояние триггера можно не изображать или выделять каким-либо способом. Если схема триггера представлена на логических элементах, то временные диаграммы могут быть построены для выходных сигналов всех логических элементов и запрещённое состояние триггера может быть выявлено в этом случае по совпадению в течении длительного времени состояний сигналов на прямом и инверсном выходах.

### **3.2. Асинхронный RS-триггер**

Асинхронный RS-триггер является базовым при создании более сложных триггеров. В простейшем случае асинхронный RS-триггер имеет два входа: S (Set) - вход установки триггера в единичное состояние, R (Reset) - вход установки триггера в нулевое состояние. Активный сигнал по входу S в момент появления заставляет триггер перейти в единичное состояние. Активный сигнал по входу R в момент появления заставляет триггер перейти в нулевое состояние. Рассмотрим некоторые возможные варианты реализации асинхронного RS-триггера на логических элементах, соответствующие схемным решениям условные обозначения триггеров и особенности функционирования асинхронных RS-триггеров. На практике находят широкое применение триггеры на элементах И-НЕ и на элементах ИЛИ-НЕ, которые рассмотрим подробнее.

### 3.2.1. Асинхронный RS-триггер на элементах И-НЕ

В простейшем случае подобный триггер строится на двух элементах 2И-НЕ. Принципиальная схема триггера представлена на рисунке 3.4.

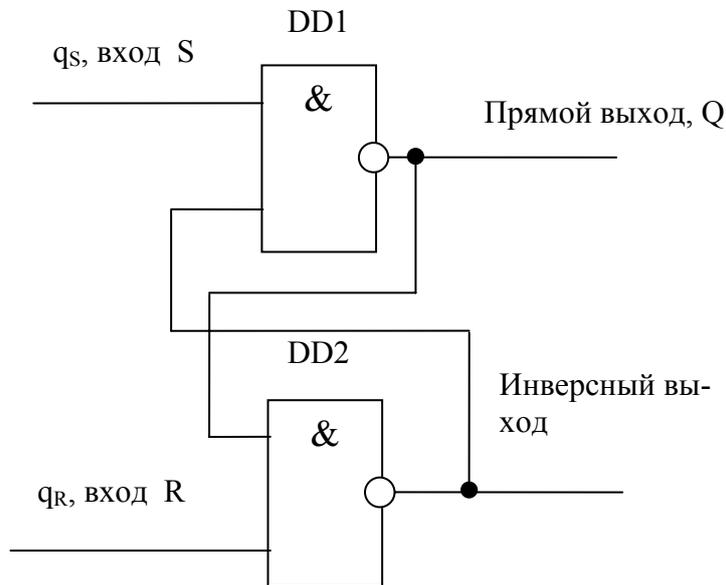


Рис. 3.4. Функциональная схема RS-триггера

Проанализируем работу этой схемы с целью определения активного сигнала. Считаем выход элемента DD1 прямым. По заданному положению прямого выхода определим положения входов установки триггера в ноль (R) и в единицу (S). Если предположить, что единица на верхнем входе, то состояние выходного сигнала DD1 будет зависеть от сигнала с выхода DD2. То есть единица на этом входе не заставляет схему непременно менять своё состояние. Это пассивный уровень сигнала на верхнем входе. Если на верхнем входе логический ноль, то независимо от сигнала с DD2 на выходе DD1 спустя время задержки одного элемента  $t_{зд}$  появится единица, если её до прихода нуля на верхний вход не было. Сформированная единица, поступая на вход DD2 при наличии на нижнем входе DD2 единицы приведёт к появлению нуля на выходе DD2 спустя время задержки  $t_{зд}$ . То есть через время  $2t_{зд}$  состояние сигнала на верхнем входе может быть изменено на противоположное. Триггер к этому моменту перешёл в новое, единичное состояние.

Таким образом, активным сигналом на верхнем входе является 0, этот вход является входом S, поскольку приводит к появлению 1 на выходе Q. Поскольку

схема симметрична, можно предположить, что нижний вход является входом сброса триггера в ноль  $R$ , причём активным сигналом и для этого входа является ноль. Сигналы на этих входах можно обозначить символами  $q_S$ ,  $q_R$ . Но в данной работе часто будем обозначать сигналы на временных диаграммах идентификаторами входов, то есть в данном случае буквами  $S$  и  $R$ . Временные диаграммы работы триггера с учётом задержки сигнала в элементах предложены на рисунке 3.5.

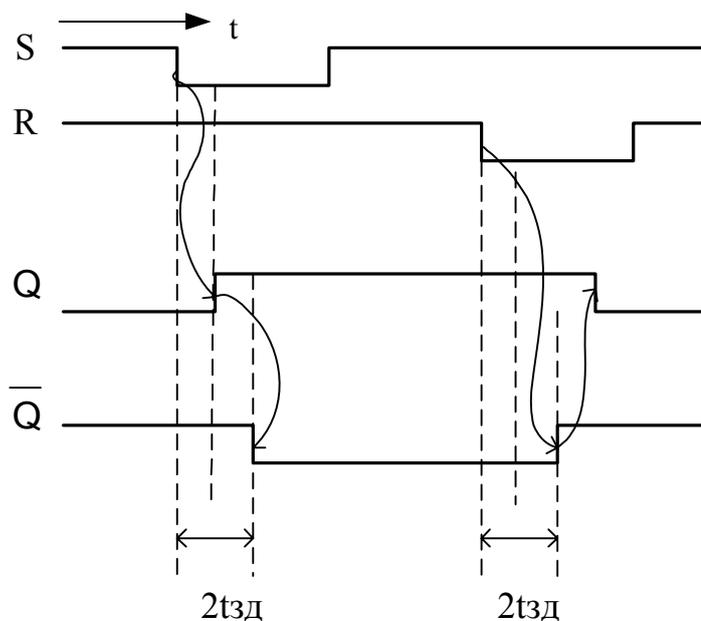


Рис. 3.5. – Временные диаграммы работы RS-триггера с учётом задержки элементов

Если предположить, что на оба входа одновременно подаются активные сигналы, то есть нули, то на обоих выходах формируются одновременно единицы. Триггер оказывается в запрещённом состоянии, которое поддерживается активными сигналами на входах. Если одновременно изменить состояние входных сигналов на пассивное, то в триггере может наблюдаться паразитная генерация с периодом  $2t_{зд}$ , которая прекратится в непредсказуемый момент времени, триггер перейдёт в определённое, но непредсказуемое состояние. На рисунке 3.7 показано возможное поведение сигналов на выходах триггера.

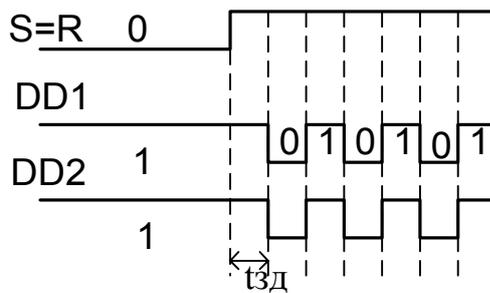


Рис. 3.7. – Возможное поведение триггера при выходе из запрещённого состояния

Построим временные диаграммы работы триггера без учёта задержки элементов, выделив интервал запрещённого состояния штриховкой (рис. 3.8). Исходное состояние триггера выбрано произвольно.

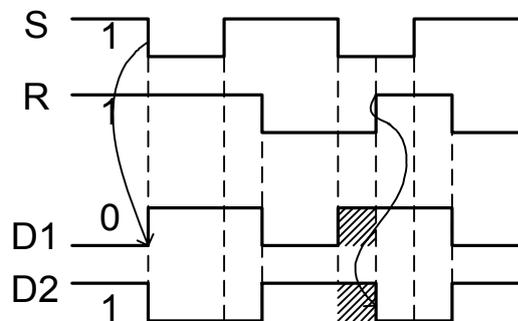


Рис. 3.8. – Временные диаграммы работы триггера без учёта задержек элементов

Если триггер реализован в составе микросхемы или необходимо изобразить его на функциональной схеме без показа внутреннего состава, то условное обозначение RS-триггера должно учитывать, что активными сигналами по входам являются нулевые уровни. Следовательно, на входах необходимо изобразить символы инверсии, как показано на рисунке 3.9.

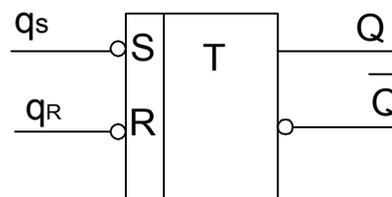


Рис. 3.9. - Условное обозначение асинхронного RS-триггера на электрических схемах

Поведение триггера можно определить сокращённой таблицей переходов (таблица 3.1), в которой сигналы на входах R и S определены для момента времени

$t$ , а состояние триггера определяется для следующего момента времени, который определяют как  $t+1$ .

Таблица 3.1

$q_S(t)$	$q_R(t)$	$Q(t+1)$	Примечание
0	0	-	запрещено
0	1	1	единица
1	0	0	нуль
1	1	$Q(t)$	хранение

Поведение триггера можно описать и матрицей переходов (таблица 3.2), которая определяет, при каких сигналах на входах в момент времени  $t$  происходит переход из исходного состояния триггера в момент времени  $t$  в состояние триггера в момент времени  $t+1$ . Исходное и конечное состояние триггера определено в столбце  $Q(t)-Q(t+1)$ , а значения сигналов на входах в момент времени  $t$  определены в столбцах  $q_S(t)$  и  $q_R(t)$ . Рассмотрим принцип построения матрицы переходов. Для первой строчки, чтобы из нулевого исходного состояния триггер перешёл в нулевое, необходимо, чтобы состояние сигнала  $q_S(t)$  на входе  $S$  было пассивным, а состояние сигнала на входе  $R$  не имеет значения. Это объясняется тем, что при пассивном сигнале на входе  $R$  триггер просто сохранит своё исходное состояние, а при активном сигнале на этом входе происходит запись нуля в триггер. Но при этом в любом случае конечное состояние триггера будет равным нулю, то есть будет равно требуемому значению. Произвольное состояние сигнала помечено в таблице символом  $b$ .

Таблица 3.2

$Q(t)-Q(t+1)$	$q_S(t)$	$q_R(t)$
0-0	1	$b$
0-1	0	1
1-0	1	0
1-1	$b$	1

### 3.2.2. Асинхронный RS-триггер на элементах ИЛИ-НЕ

Схема асинхронного триггера на элементах ИЛИ-НЕ и условное обозначение с учётом активности сигналов на входах показаны на рисунке 3.10.

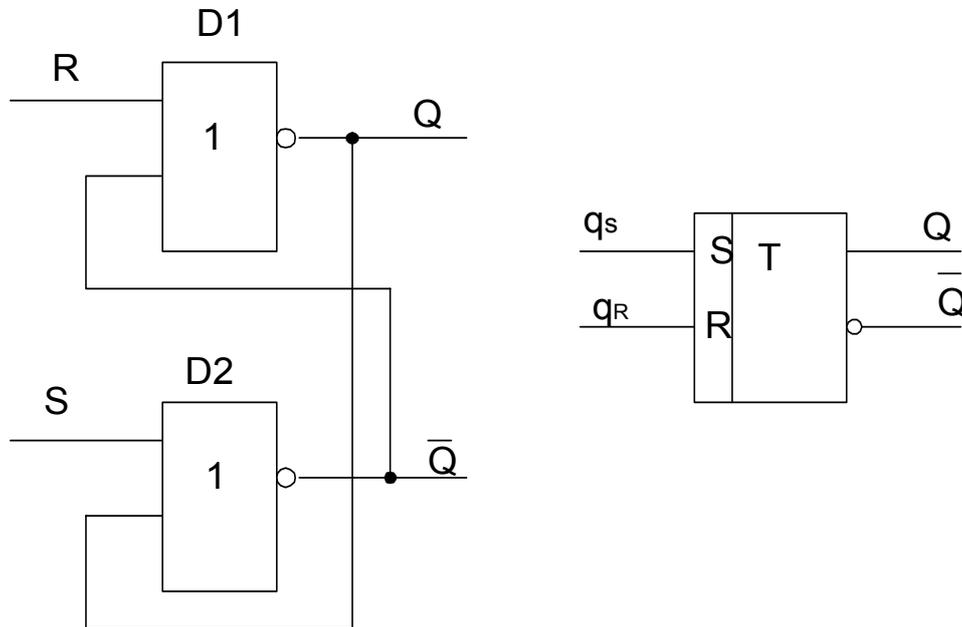


Рис.3.10. – Схема триггера на ИЛИ-НЕ и его условное обозначение

По данной схеме определим активный уровень для входов. Если предположить, что на верхнем входе ноль, то поведение DD1 будет определяться вторым входом, соединённым с DD2, то есть нулевой уровень на данном входе не определяет однозначно состояние триггера. Данный уровень является пассивным. Если на верхнем входе единица, то независимо от состояния сигнала с выхода DD2 на выходе DD1 формируется ноль, который при условии наличия нуля на втором входе DD2 приведёт к появлению единицы на выходе DD2. То есть цепь обратной связи замкнулась, триггер запомнил новое состояние. Поскольку единица на верхнем входе привела к появлению нуля на прямом выходе, то данный вход является входом R. Аналогичные рассуждения позволяют доказать, что вход элемента DD2 является входом S и единица является активным сигналом по этому входу. Поскольку по обоим входам активным сигналом является единица, на условном обозначении триггера оба входа показаны прямыми. Таблица переходов и матрица переходов предложены в таблицах 3.3 и 3.4 соответственно.

Таблица 3.3

$q_s(t)$	$q_R(t)$	$Q(t+1)$	Примечание
0	0	$Q(t)$	хранение
0	1	0	нулевое
1	0	1	единичное
1	1	-	запрещено

Таблица 3.4

$Q(t)-Q(t+1)$	$q_s(t)$	$q_R(t)$
0-0	0	b
0-1	1	0
1-0	0	1
1-1	b	0

Временные диаграммы работы триггера предложены на рисунке 3.11.

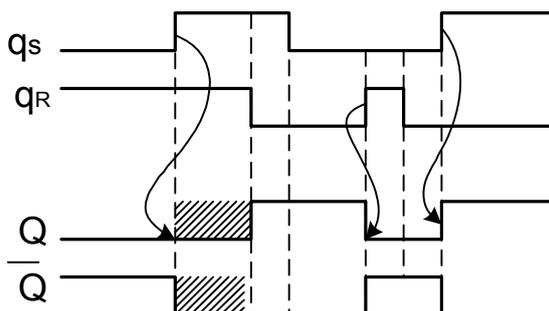


Рис. 3.11. – Временные диаграммы работы асинхронного RS-триггера на ИЛИ-НЕ

*Если необходим триггер с разными активными сигналами по входам, достаточно взять любой из рассмотренных триггеров и добавить на один из входов инвертор, чтобы проинвертировать активность сигнала. Таким образом можно получить триггер, условное обозначение которого предложено на рисунке 3.12.*

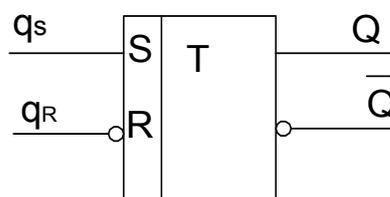


Рис. 3.12. – Вариант условного обозначения триггера с разными уровнями активного сигнала по входам

Пример временных диаграмм поведения этого триггера предложен на рисунке 3.13.

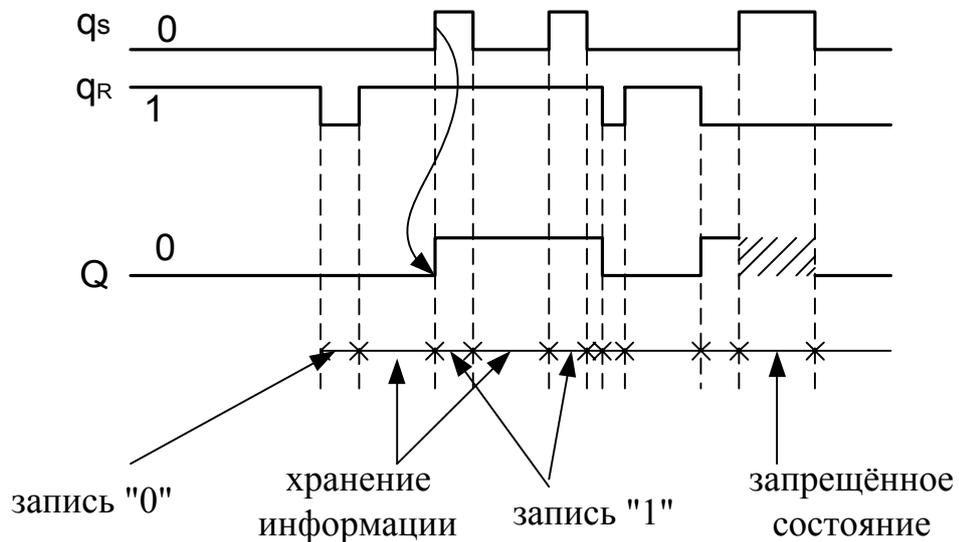


Рис. 3.13. – Временные диаграммы работы триггера

Для предложенного варианта триггера легко записать сокращённую таблицу переходов, представленную в таблице 3.5.

Таблица 3.5

$q_s(t)$	$q_R(t)$	$Q(t+1)$	Примечание
0	0	0	запись нуля
0	1	$Q(t)$	хранение
1	0	-	запрещено
1	1	1	запись единицы

### 3.3. Синхронные триггеры с потенциальным управлением

#### 3.3.1. Общие сведения

Синхронные триггеры принимают информацию при активном уровне синхронизирующего сигнала и в течение всего этого интервала.

Выделим среди данного множества триггеров одноступенчатые и двухступенчатые.

Одноступенчатые триггеры отличаются применением в качестве базового одного асинхронного RS-триггера. В этих триггерах интервал приёма информации является началом интервала выдачи новой информации на выходе. То есть триггер сформирует своё новое состояние в выходных цепях во время появления активного уровня синхросигнала. Всякое изменение состояния информационных входов при активном уровне синхросигнала нежелательно, если требуется установить триггер в строго определённое состояние. На время активного синхросигнала триггер как бы превращается в асинхронный. Если в таком триггере предположить, что входные сигналы зависят от состояния этого же триггера, то при активном синхросигнале в триггере может возникнуть паразитная генерация. Логика работы триггера будет нарушена, схема окажется неработоспособной. Поведение триггера в подобном случае будет рассмотрено ниже.

Чтобы исключить паразитную генерацию используют двухступенчатые триггеры или триггеры с динамическим управлением.

В двухступенчатых триггерах по активному уровню синхросигнала происходит запись информации с информационных входов в первый RS-триггер, а с изменением состояния синхросигнала по пассивному уровню его происходит перепись принятого первым триггером состояния во второй триггер. Выходами двухступенчатого триггера являются выходы второго триггера.

### **3.3.2. Одноступенчатый синхронный RS-триггер**

Рассмотрим два варианта реализации этого триггера: на элементах И-НЕ и на элементах ИЛИ-НЕ. Триггеры на элементах И-НЕ легче реализуются при использовании технологии ТТЛ, а на ИЛИ-НЕ проще реализуются при использовании технологии ЭСЛ или МДП.

Синхронный RS-триггер на элементах И-НЕ строится с использованием асинхронного RS-триггера на И-НЕ, у которого добавляется входная логика на таких же элементах. Схема триггера и соответствующее ей условное обозначение синхронного триггера предложены на рисунке 3.14.

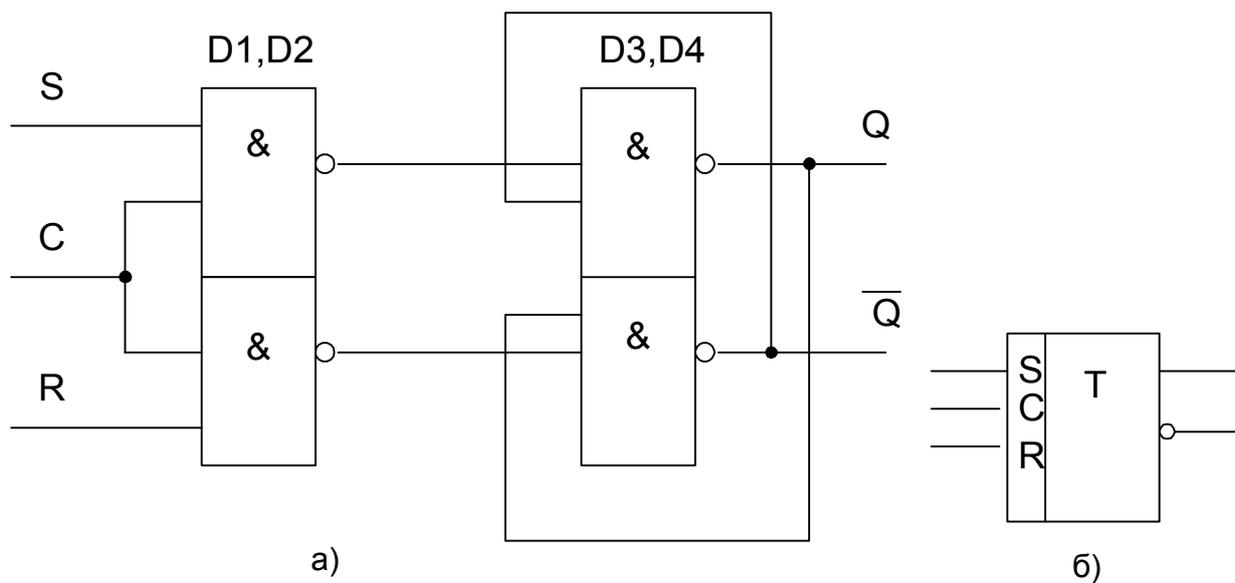


Рис. 3.14. – Схема синхронного триггера на элементах И-НЕ (а) и его условное обозначение (б)

При  $C=0$  на выходах D1, D2 (входная логика) формируются уровни логической единицы, которые воспринимаются триггером D3, D4 как пассивные сигналы. Триггер хранит предыдущее состояние. При  $C=1$  разрешается передача информации с информационных входов S и R на выходы входной логики. При этом поведение триггера определяется таблицей переходов (таблица 3.6).

Таблица 3.6

S(t)	R(t)	Q(t+1)	Примечание
0	0	Q(t)	хранение
0	1	0	запись нуля
1	0	1	запись единицы
1	1	-	запрещено

Проанализируем работу триггера, используя его схему.

Если  $S=R=0$ , то независимо от состояния сигнала синхронизации C на выходах элементов входной логики D1, D2 формируются единичные уровни. При этом триггер D3, D4 оказывается в режиме хранения информации, то есть предыдущего состояния. При  $S=1, R=0$  под действием активного синхросигнала  $C=1$  происходит запись единицы в триггер. Действительно, на выходе D1 под действием заданной комбинации сигналов формируется уровень нуля, а на выходе D2 наблюдается еди-

ница. Под действием нуля с выхода D1 асинхронный триггер на элементах D3, D4 переходит в единичное состояние. При  $S=0, R=1$  под действием активного синхросигнала  $C=1$  происходит запись нуля в триггер. При  $S=1, R=1$  под действием активного синхросигнала  $C=1$  триггер оказывается в запрещённом состоянии. Объясняется это наличием на выходах входной логики D1, D2 одновременно нулей, что приводит к единицам на выходах элементов D3, D4. Логика работы триггера нарушается, на обоих его выходах наблюдается одинаковое состояние. По окончании действия активного синхросигнала в этой ситуации трудно предсказать поведение триггера, поскольку в нём возможно возникновение генерации или состояние триггера оказывается непредсказуемым. На рисунке 3.15 показаны временные диаграммы работы синхронного триггера. На этих диаграммах показано поведение триггера и при запрещённой комбинации входных сигналов, и показана область непредсказуемого состояния триггера.

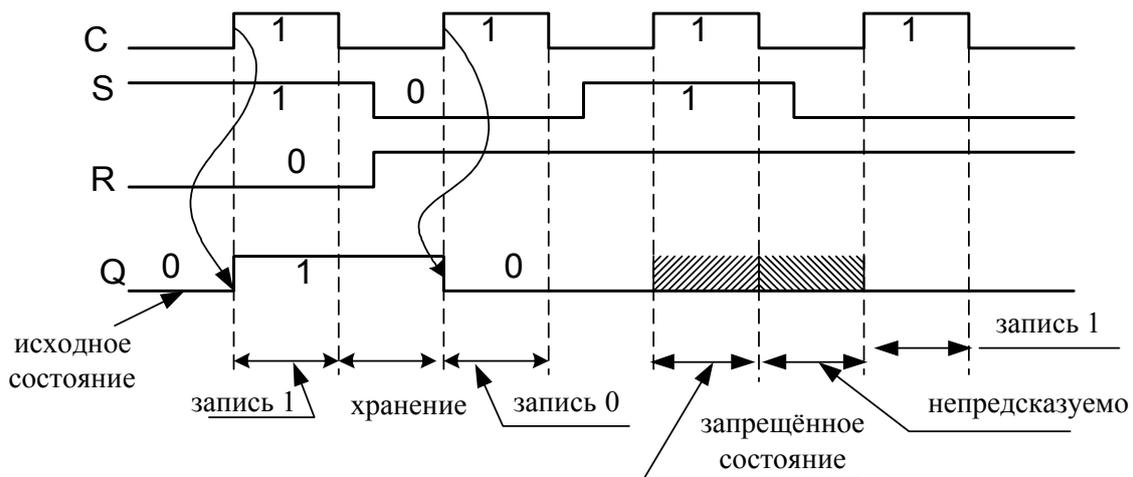


Рис. 3.15. – Временные диаграммы работы синхронного триггера

Определённый интерес представляет и поведение триггера при изменении информационных сигналов во время действия активного синхросигнала. Временные диаграммы поведения триггера для этого случая предложены на рисунке 3.16.

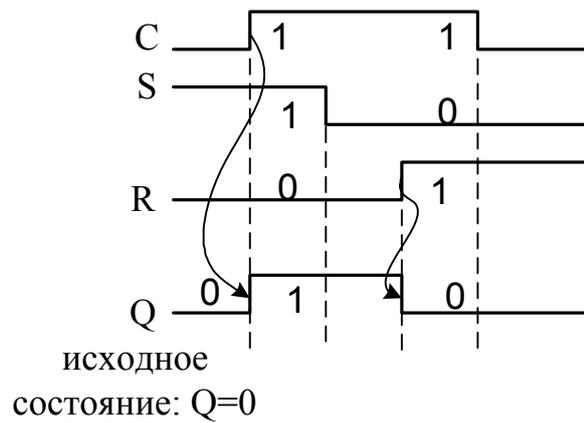


Рис. 3.16. – Временные диаграммы работы триггера при изменении сигналов S и R во время активного синхросигнала

Эти диаграммы показывают, что на время действия синхросигнала триггер превращается как бы в асинхронный. Всякое изменение информационных сигналов при потенциальном управлении приводит к изменению состояния триггера. Чтобы исключить это явление необходимо исключить возможность изменения информационных сигналов при активном уровне синхросигнала.

Схема синхронного RS-триггера на элементах ИЛИ-НЕ и его условное обозначение предложены на рисунке 3.17. Особенностью триггера является то, что активными сигналами по всем входам являются логические нули.

*(Анализ работы схемы читатель в состоянии сделать самостоятельно, используя порядок, предложенный для триггера на элементах И-НЕ).*

Поведение триггера во времени показано на временных диаграммах рисунка 3.18. Можно заметить, что в начальный момент времени триггер оказывается в запрещённом состоянии, после которого поведение триггера, то есть его состояние, непредсказуемо до следующего активного импульса на входе синхронизации при условии соблюдения неравенства на входах S и R: S не равно R.

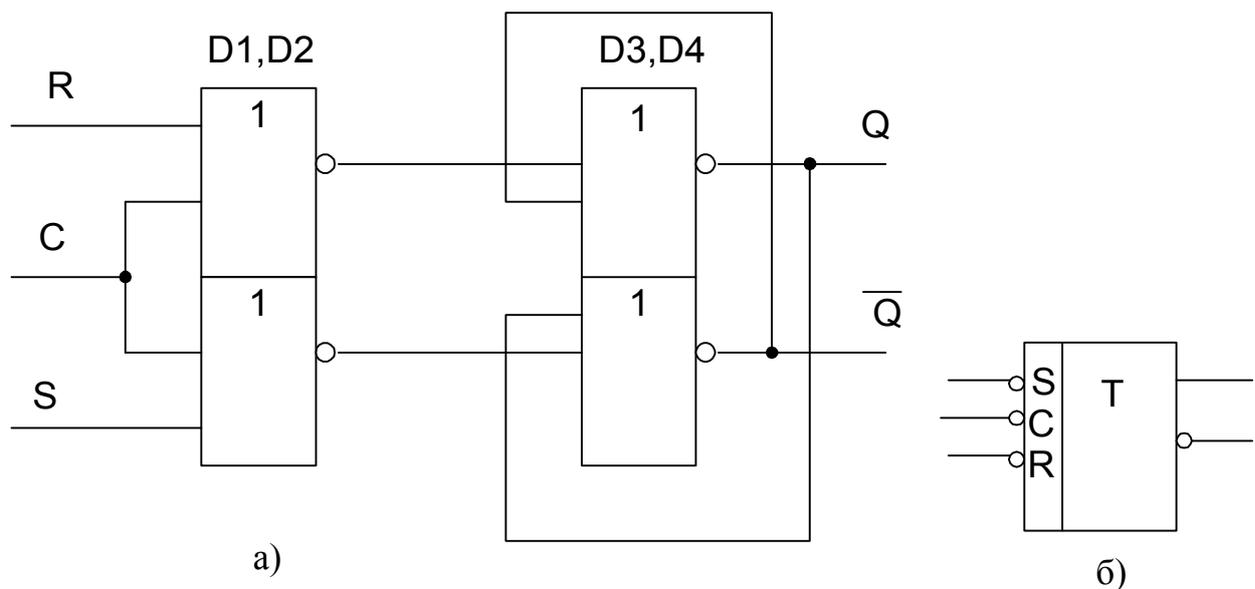


Рис. 3.17. – Схема синхронного RS-триггера на элементах ИЛИ-НЕ (а) и его условное обозначение (б)

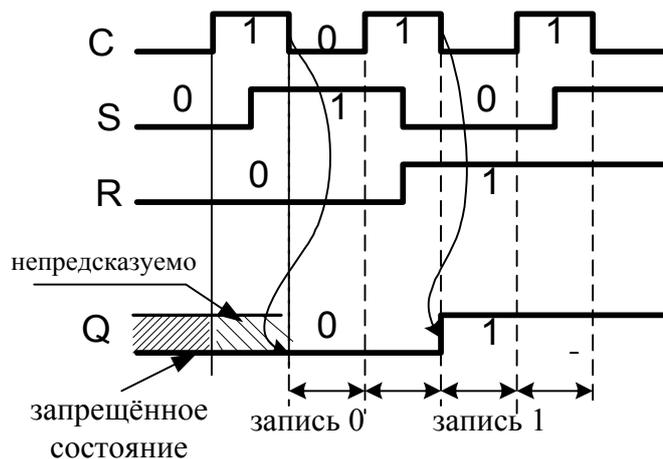


Рис. 3.18. - Временные диаграммы работы триггера на ИЛИ-НЕ

Для обоих рассмотренных триггеров время задержки срабатывания в ответ на фронт входного синхросигнала равно времени задержки трёх логических элементов. *(Таблица и матрица переходов для триггера на ИЛИ-НЕ легко строится читателями самостоятельно, предполагая, что сигнал синхронизации активен).*

### 3.3.3. Синхронный одноступенчатый D-триггер.

По сигналу синхронизации на своём прямом выходе D-триггер повторяет состояние сигнала на входе D (Delay). Триггер может быть построен с использованием синхронного RS-триггера по схеме, предложенной на рисунке 3.19. На этом же рисунке показано и условное обозначение простейшего D-триггера с потенциальным управлением.

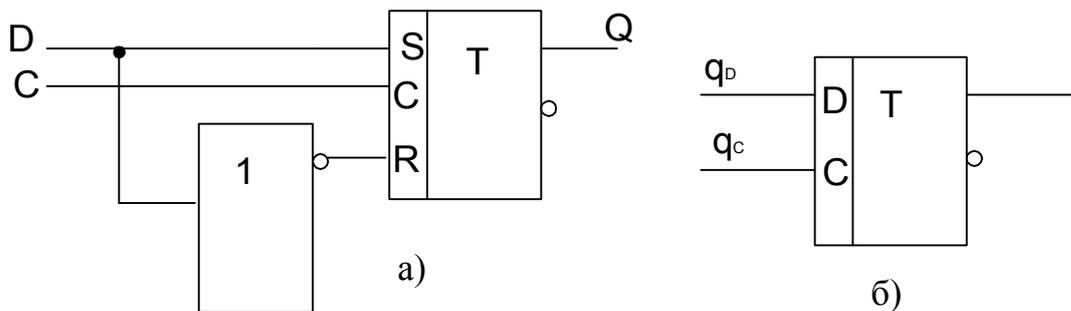


Рис. 3.18. – Схема D-триггера и его условное обозначение

При  $D=0$  на входе  $S$  формируется ноль, а благодаря инвертору на входе  $R$  наблюдается единица. Следовательно, с приходом активного сигнала на синхровход  $C$  будет происходить запись нуля в триггер. То есть на прямом выходе триггера повторяется состояние входного сигнала в момент действия активного синхросигнала. При  $D=1$  происходит запись единицы. Поведение триггера описывается таблицей переходов (таблица 3.7) и матрицей переходов, предложенной в таблице 3.8.

Таблица 3.7

$D(t)$	$Q(t+1)$
0	0
1	1

$Q(t)-Q(t+1)$	$D(t)$
0-0	0
0-1	1
1-0	0
1-1	1

Обратите внимание, что для D-триггера нет запрещённой комбинации сигналов на входах  $D$  и  $C$ .

Определим поведение D-триггера с помощью временных диаграмм, предложенных на рисунке 3.19.

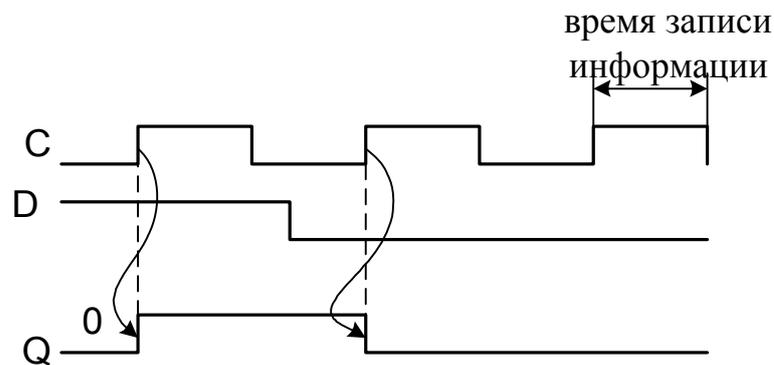


Рис. 3.19. – Временные диаграммы работы D-триггера с потенциальным управлением

При активном синхросигнале нежелательно менять состояние сигнала на информационном входе D. В момент окончания действия активного синхросигнала происходит переход триггера из режима записи в режим хранения принятой информации. Как бы триггер защёлкивается в новом состоянии. Поэтому подобные триггеры иногда называют триггерами-защёлками.

Обратите внимание на тот факт, что при активном синхросигнале изменение состояния сигнала на входе D повторяется на выходе. То есть триггер превращается в повторитель входного сигнала. Этим фактом иногда пользуются для повышения, например, нагрузочной способности схемы.

Недостатком предложенного схемного решения (рис.3.17,а) является наличие лишнего элемента: инвертора. Используя инвертирующие способности элементов входной логики, D-триггер можно реализовать проще. Возможное схемное решение D-триггера предложено на рисунке 3.20.

Входная логика D1, D2 реализует одновременно функцию инвертора в предыдущей схеме. При  $C=0$  триггер хранит информацию, поскольку на выходах D1, D2 присутствуют единицы, что соответствует пассивным сигналам асинхронного триггера D3, D4. При  $C=1$  в триггер записывается состояние сигнала D. Если  $D=0$ , то на выходе D1 формируется единица, а на выходе D2 формируется ноль, что приводит к записи в триггер нуля.

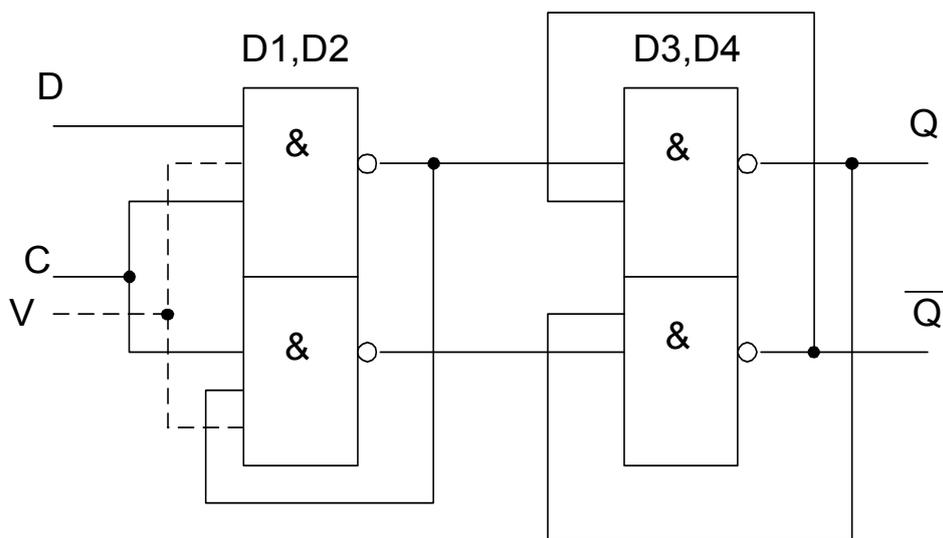


Рис. 3.20. – Схема D-триггера с потенциальным управлением

Если D-триггер снабжён дополнительным входом  $V$ , то образуется DV-триггер. Вход  $V$  является входом разрешения действия синхросигнала на триггер. Условное обозначение и временные диаграммы работы DV-триггера предложены на рисунке 3.21.

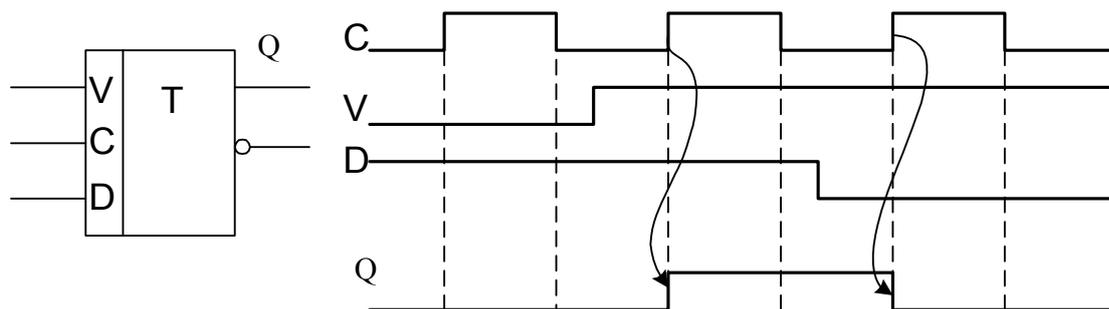


Рис. 3.21. - Условное обозначение и временные диаграммы работы DV-триггера

При  $V=0$  запрещается прохождение импульса синхронизации на триггер, что обеспечивает режим хранения в триггере. Если  $V=1$ , то триггер реагирует на синхросигнал, принимая по его активному уровню информацию с входа  $D$ .

### 3.3.4. Проблема генерации в одноступенчатом триггере

Если при активном синхросигнале меняются входные информационные сигналы, то триггер может изменять своё состояние. Очень часто входные информационные сигналы данного триггера оказываются функцией выходного сигнала этого же триггера. В этом случае в одноступенчатых триггерах возможна паразитная генерация, что приводит к потере работоспособности схемы. Рассмотрим возможный вариант включения одноступенчатого D-триггера, в котором при длительном синхросигнале появится паразитная генерация. Схемное решение и временные диаграммы работы одноступенчатого триггера с потенциальным управлением предложены на рисунке 3.22.

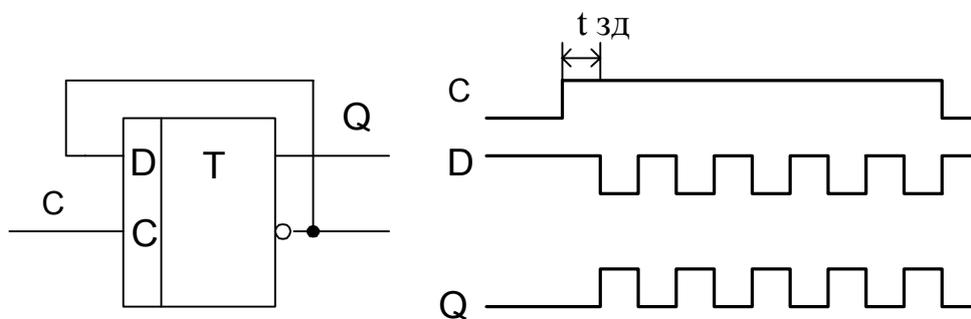


Рис.3.23. – Обратная связь в D триггере и его поведение при активном синхросигнале

Инверсный сигнал на входе D приводит к тому, что с появлением синхросигнала в триггер записывается противоположное состояние. Но если синхросигнал продолжает действовать, то на входе D будет периодически меняться информация, что приводит к периодическому изменению состояния триггера с периодом, равным  $2t_{зд}$ , где  $t_{зд}$  – время задержки срабатывания триггера, то есть изменения его состояния.

Чтобы исключить данное паразитное явление, можно или осуществлять приём информации в течение короткого интервала времени, прилегающего к фронту синхросигнала, или обеспечить стабильность входных информационных сигналов в течение всего интервала времени действия синхросигнала. Первый способ оправдывает применение триггеров с динамическим управлением, а второй оправдывает применение двухступенчатых триггеров.

### 3.3.5. Двухступенчатые синхронные триггеры

Двухступенчатые синхронные триггеры отличаются наличием двух последовательно включённых триггеров. Первый триггер принимает информацию с входов при активном уровне синхросигнала, а второй принимает информацию с первого триггера при пассивном уровне синхросигнала и выдаёт его на выходы всей схемы.

#### 3.3.5.1. Двухступенчатый RS-триггер

Двухступенчатый RS-триггер представлен на рисунке 3.23.

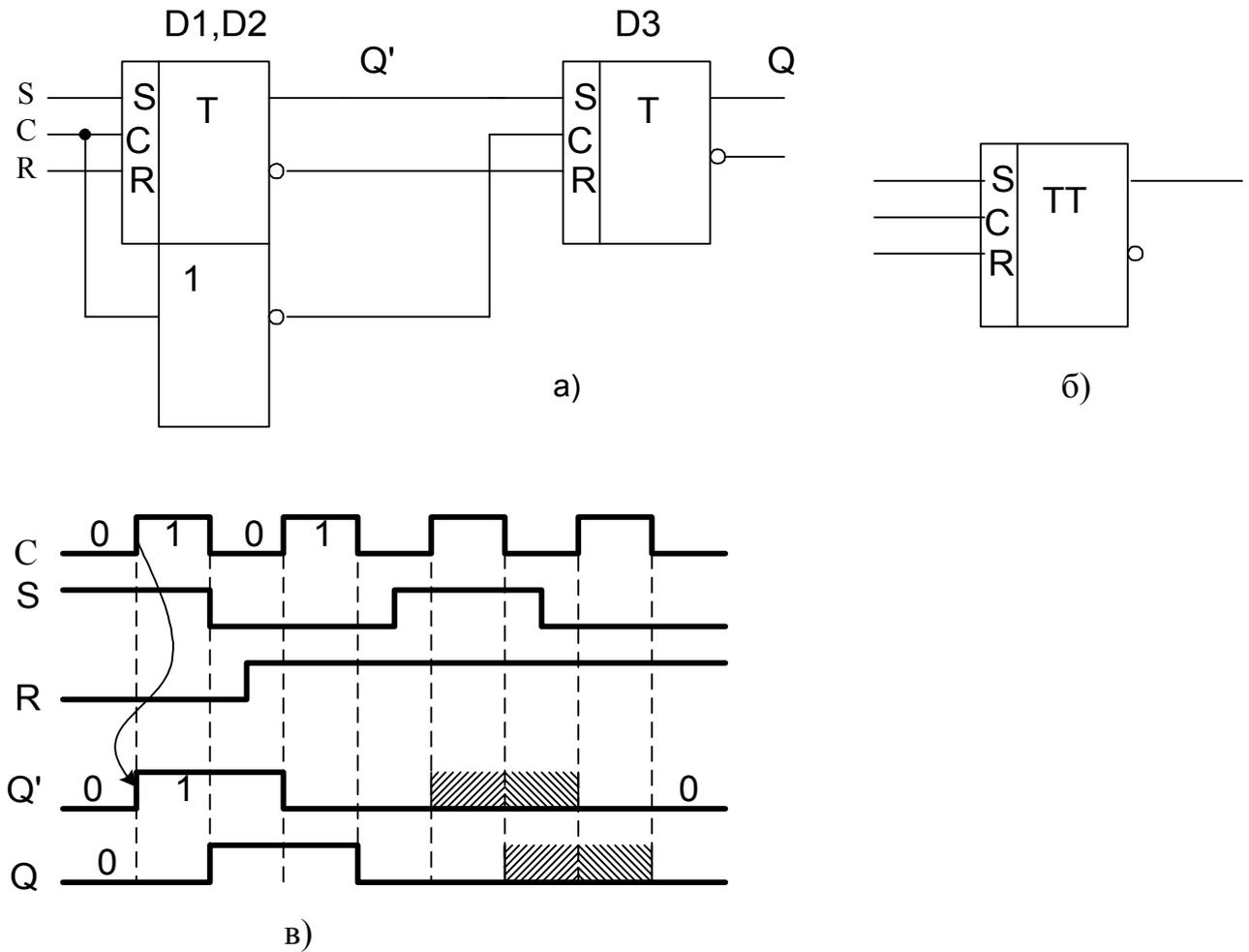


Рис. 3.23. – Схема, условное обозначение и временные диаграммы работы двухступенчатого RS-триггера

#### 3.3.5.2. Т-триггер

Т-триггер или триггер со счётным входом предназначен для подсчёта поступающих на вход импульсов. Название триггера берётся от слова Toggle –

кувыркаться. Т-триггеры бывают асинхронные и синхронные. В синхронных триггерах имеется вход разрешения счёта или управления. Рассмотрим реализацию асинхронного Т-триггера на двухступенчатом RS-триггере (рисунок 3.24).

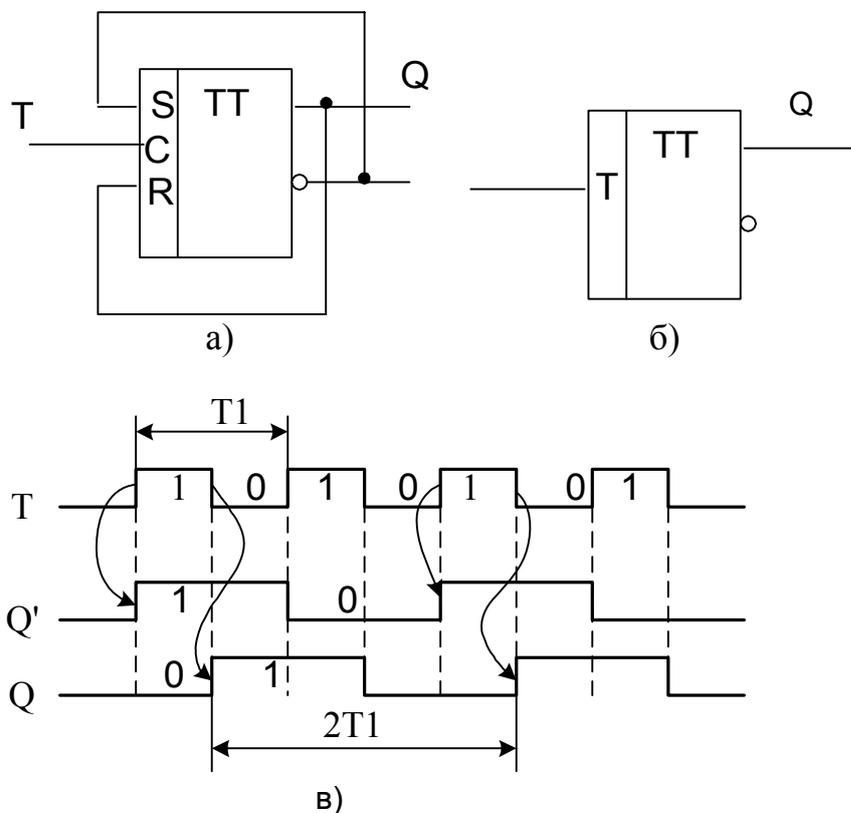


Рис. 3.24. – Асинхронный Т-триггер на основе RS-триггера (а), его условное обозначение (б) и временные диаграммы работы (в)

За счёт перекрёстных обратных связей на входах первой ступени формируется состояние, противоположное состоянию второй выходной ступени двухступенчатого триггера.

При  $C=1$  первая ступень принимает новое состояние так, что в течение всего интервала, пока  $C=1$ , триггеры оказываются в противоположных состояниях. При переходе синхросигнала в ноль новое состояние триггера первой ступени транслируется на выход, то есть переписывается во второй триггер. Оба триггера оказываются в новом одинаковом состоянии, но входная комбинация сигналов оказывается готовой изменить состояние входного триггера.

Т-триггер делит частоту входного сигнала на 2 и может быть использован для формирования симметричного импульсного сигнала, скважность которого равна двум. Напоминаем. Что скважность импульсного сигнала равна отношению дли-

тельности периода к длительности импульса. Т-триггер ведёт себя как сумматор одноразрядных двоичных чисел.

Синхронный Т-триггер отличается наличием на входе Т конъюнктора, один из входов которого выполняет роль управляющего входа Т, а второй является входом синхронизации. Синхронный Т-триггер представлен на рисунке 3.25.

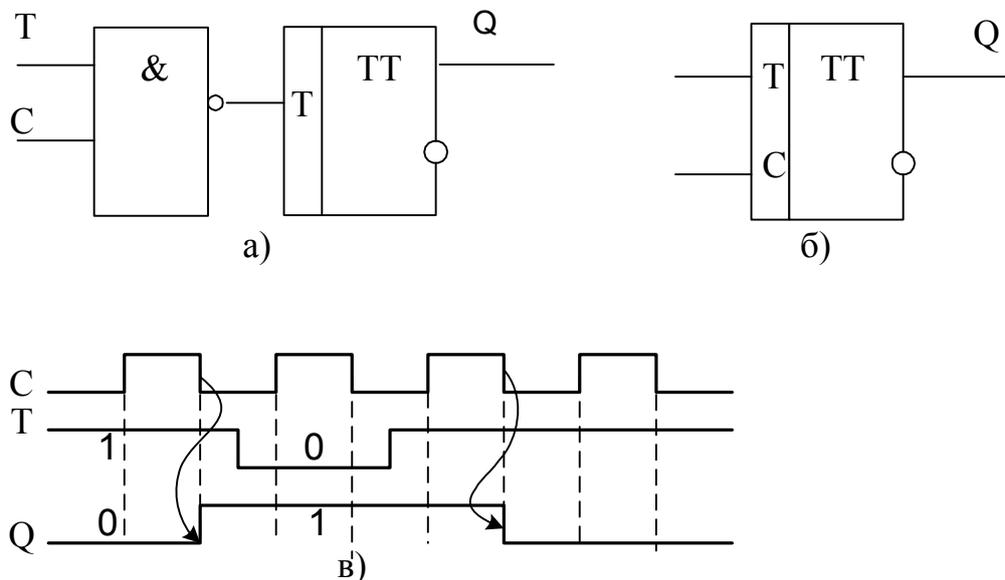


Рис. 3.25.- Схема синхронного Т-триггера (а), его условное обозначение (б) и временные диаграммы работы (в)

### 3.3.5.3. JK-триггер

JK-триггер отличается от двухступенчатого RS-триггера тем, что в нём отсутствует запрещённая комбинация входных сигналов. Своё название триггер берёт от слов Jerk – резкий толчок, внезапное включение и Kill – убивать, отключение. Когда  $J=K=1$ , он превращается в Т-триггер, на остальных комбинациях ведёт себя подобно RS-триггеру. При этом J подобен входу S, а вход K подобен входу R. На рисунке 3.26 показана схема JK-триггера на основе двухступенчатого RS-триггера и его условное обозначение.

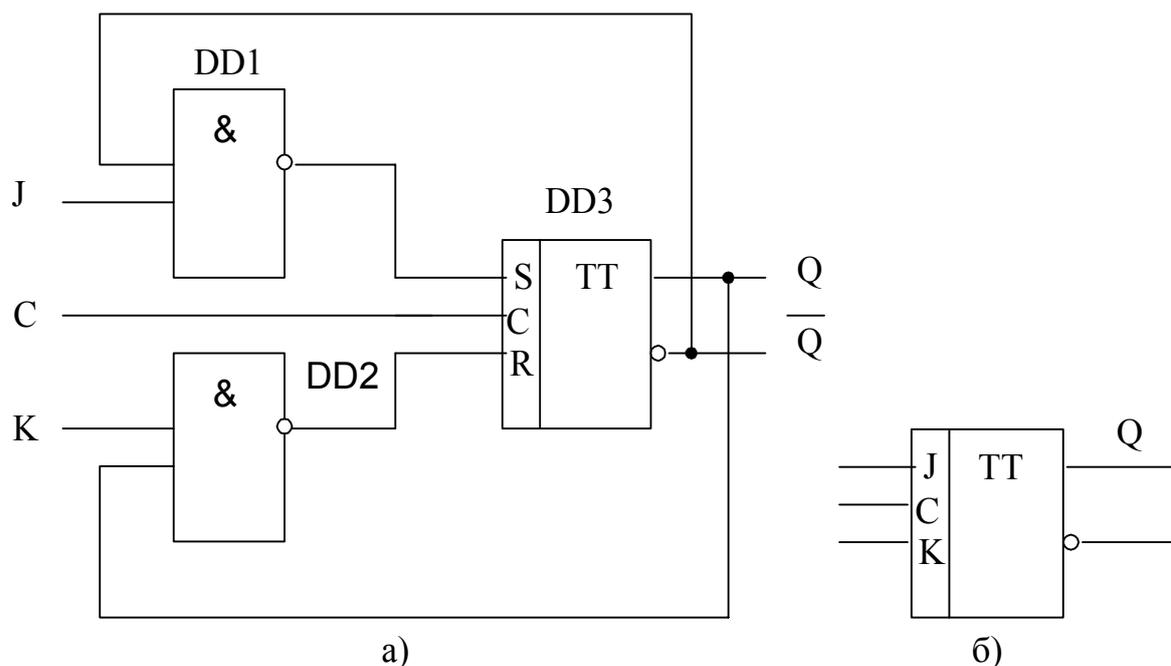


Рис. 3.26. – Схема (а) и условное обозначение (б) двухступенчатого JK-триггера

При реализации входной логики не требуется введения дополнительных конъюнкторов, как это предлагается на рисунке 3.26,а, поскольку достаточно увеличить количество входов у входных конъюнкторов двухступенчатых RS-триггеров.

Поведение JK-триггеров характеризуется таблицей переходов (таблица 3.9) и матрицей переходов (таблица 3.10).

Таблица 3.9

J(t)	K(t)	Q(t+1)	Примечание
0	0	Q(t)	хранение
0	1	0	запись нуля
1	0	1	запись единицы
1	1	$\overline{Q(t)}$	режим Т-триггера

Таблица 3.10

Q(t)-Q(t+1)	J(t)	K(t)
0-0	0	1

0-1	1	b
1-0	b	1
1-1	b	0

Чтобы расширить функциональные возможности JK-триггеров формируют несколько входов J и K. Все входы J и K обычно связаны конъюктивно. Кроме того, триггер снабжают асинхронными входами R и S, которые имеют приоритет в работе. На рисунке 3.27 предлагается возможный вариант JK-триггера серии ТТЛ, реализованный микросхемой К555ТВ1. Предложенный триггер асинхронно может быть уровнями логического нуля установлен в любое состояние, если эти уровни подать на входы S и R. Только при наличии на всех входах j1, j2, j3 триггер воспринимает единицу по входу J. Аналогичная логика имеется и на входе K.

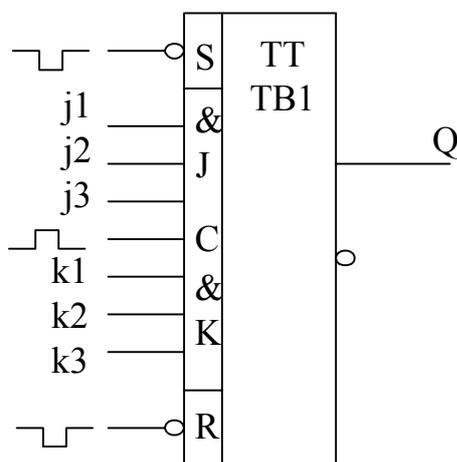


Рис. 3.27. – JK-триггер с расширенными функциональными возможностями

### 3.4. Триггеры с динамическим управлением

Особенностью триггеров с динамическим управлением является то, что они принимают информацию в течение короткого интервала времени вблизи активного фронта синхросигнала. На практике находит широкое применение D-триггеры с динамическим управлением. На рисунке 3.28 предложена схема D-триггера с динамическим управлением, которая реализована в составе микросхемы ТТЛ типа К555ТМ2. Пунктиром показаны цепи асинхронных входов установки триггера.

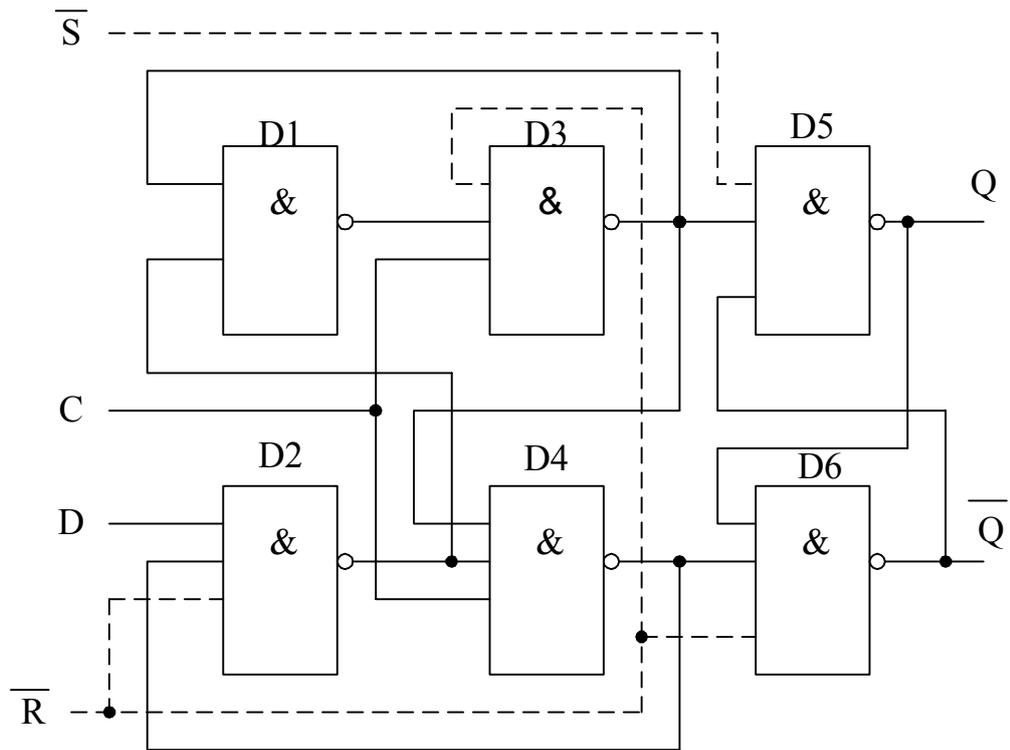


Рис. 3.28. – Схема D-триггера с динамическим управлением

Временные диаграммы работы D-триггера предложены на рисунке 3.29.

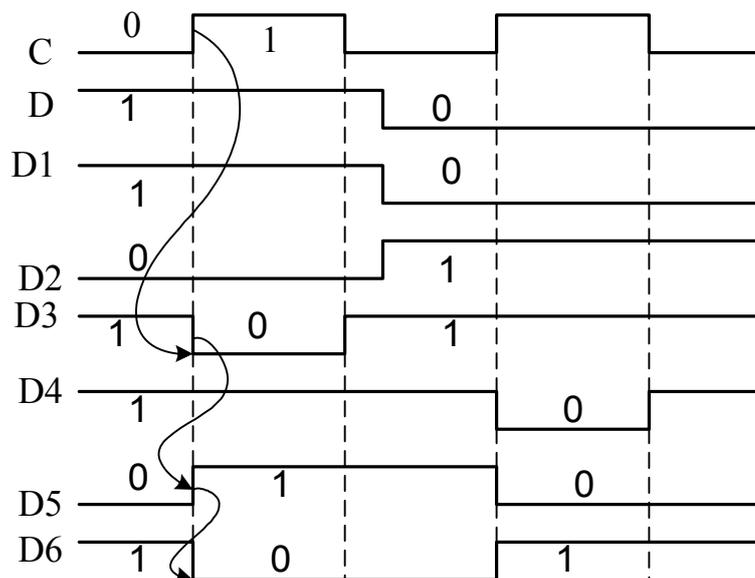


Рис. 3.29. – Временные диаграммы работы D-триггера

Исходное состояние триггера D5, D6 задано и не противоречит комбинации входных сигналов. Триггер хранит предыдущее состояние. При D=1 на выходе D2 формируется ноль, что приводит к появлению единицы на выходе D1. С приходом нарастающего фронта синхросигнала наступает совпадение единиц на входах D3, что приводит к появлению 0 на его выходе спустя время задержки элемента. Этот

ноль поступает на входы D1 и D4, блокируя информационный вход триггера D, то есть спустя время задержки всего одного элемента после фронта синхросигнала изменение сигнала D не вызывает изменение состояния триггера. При этом ноль на выходе D3 переводит выходной сигнал D5 в единицу, а D6 - в ноль. Время задержки триггера от фронта синхросигнала оказывается равным трём временам задержки логических элементов.

Аналогично триггер ведёт себя при  $D=0$ , только в этом случае блокирование информационного сигнала происходит нулём с выхода D4 по входу D2.

Предложенный анализ работы входной логики триггера, выполненной на элементах D1, D2 показывает, что с появлением нарастающего фронта на входе синхронизации спустя время задержки всего одного элемента происходит блокирование информационного входа так, что изменение состояния сигнала после этого не приводит к изменению состояния триггера. Только с приходом следующего нарастающего фронта синхросигнала возможна запись в триггер состояния информационного сигнала.

Условное обозначение предложенного триггера с учётом асинхронных входов представлено на рисунке 3.30.

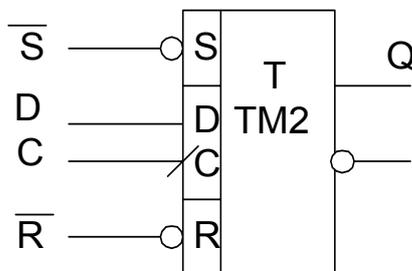


Рис. 3.30. – Условное обозначение триггера микросхемы К555ТМ2

Наличие асинхронных входов расширяет функциональные возможности триггера. При подаче активного сигнала на любой из асинхронных входов блокируется работа синхронного входа D.

### 3.5. Некоторые применения универсальных триггеров

Простейшим применением D-триггеров и JK-триггеров является реализация T-триггеров на их основе.

Способ построения синхронного Т-триггера на основе JK-триггера и временные диаграммы его работы показаны на рисунке 3.31.

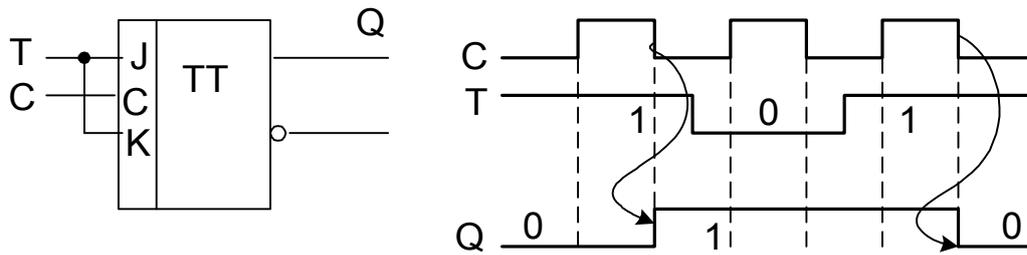


Рис. 3.31. – Т-триггер на базе JK-триггера

При подаче на вход Т уровня логической единицы на входах J и K триггера оказывается единица, JK-триггер с приходом каждого импульса на вход синхронизации будет менять своё состояние на противоположное. Если же на входе Т присутствует ноль, то триггер независимо от состояния синхросигнала будет находиться в режиме хранения предыдущего состояния. Временные диаграммы показывают, что переход в противоположное состояние происходит по спадающему фронту синхросигнала, поскольку именно в этот момент происходит запись в выходной триггер двухступенчатого триггера состояния триггера первой ступени. Если на входы J и K триггера подавать постоянно уровень единицы, то предложенный триггер превращается в асинхронный Т-триггер, у которого вход синхронизации выполняет роль счётного входа.

На основе D-триггера с динамическим управлением легко реализовать асинхронный Т-триггер. Чтобы D-триггер «кувыркался» с приходом каждого импульса на вход синхронизации достаточно поставить переключку между входом D и инверсным выходом триггера, как показано на рисунке 3.32.

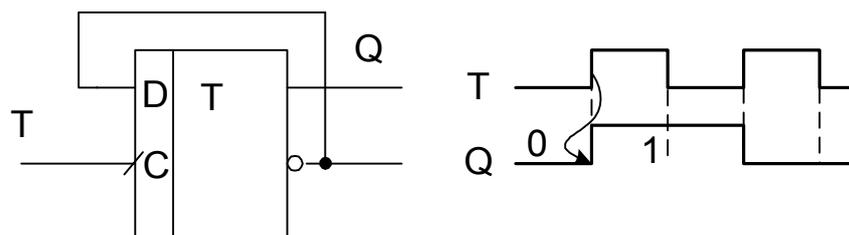


Рис. 3.32. – Асинхронный Т-триггер на базе D-триггера с динамическим управлением и временные диаграммы его работы

Если триггер находится в нулевом состоянии, то с его инверсного выхода на вход D поступает единица. Это приводит к тому, что в момент действия нарастающего фронта синхросигнала в триггер с входа D запишется единица. То есть триггер перешёл в противоположное состояние. В результате на входе D формируется логический ноль, который запишется в триггер с приходом следующего нарастающего фронта синхросигнала.

Если на входе синхронизации рассмотренного Т-триггера используется элемент 2И-НЕ, как показано на рисунке 3.33, то будем иметь дело с синхронным Т-триггером.

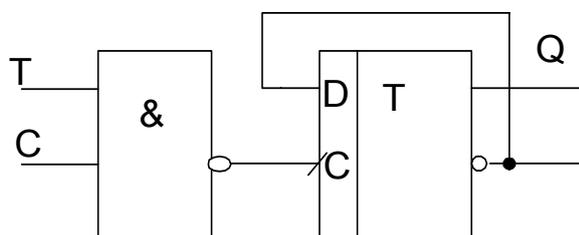


Рис. 3.33. – Синхронный Т-триггер на базе D-триггера

Активным фронтом синхросигнала в этом случае будет спадающий фронт.

*(По примеру временных диаграмм, предложенных на рисунке 3.3, постройте временные диаграммы для сигналов на выходе элемента И-НЕ и на выходе триггера и докажите, что активным сигналом будет спадающий фронт сигнала C на входе элемента И-НЕ)*

### 3.6. Синтез триггерных схем

Поскольку функциональные возможности микросхем триггеров невелики, возникают проблемы их расширения за счёт добавления к триггерам входной логики. Рассмотрим пример синтеза многофункционального триггера на основе D-триггера с динамическим управлением. Синтез сводится к проектированию комбинационных схем КС1 и КС2, которые должны использоваться на входах D и C исходного триггера. Структура проектируемого триггера предложена на рисунке 3.34,а.

Будем понимать под микрооперацией действия, которые выполняются в узле в данном дискретном интервале времени под действием одного управляющего сиг-

нала. Под микрокомандой будем понимать совокупность микроопераций, выполняемых разными узлами в один и тот же интервал времени. Кроме управляющего сигнала  $Y_i$  подразумевается, что на схему действуют импульсы синхронизации. При однофазной синхронизации на всём интервале активного управляющего сигнала действует один синхроимпульс. В более сложном случае при многофазной синхронизации в течение времени действия управляющего сигнала могут действовать 2-3 импульса синхронизации, сдвинутых во времени, которые действуют на разные точки схемы. Взаимное расположение управляющего и синхронизирующего сигналов при однофазной синхронизации показаны на рисунке 3.34,б.

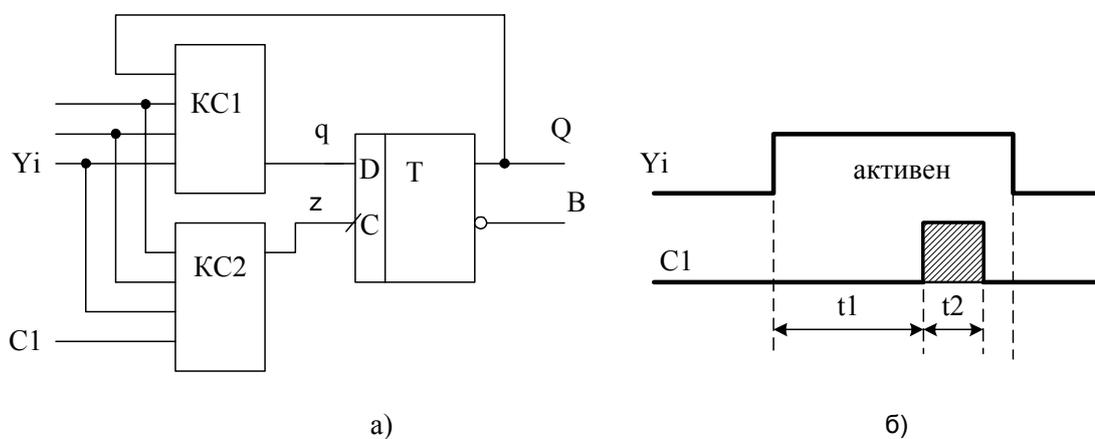


Рис. 3.34. – Структура синтезируемого триггера (а) и временные диаграммы управляющих сигналов (б)

Интервал времени  $t_1$  необходим для того, чтобы за это время под действием управляющего сигнала  $Y_i$  закончились переходные процессы во всех комбинационных схемах, которые участвуют в выполнении данной микрооперации. В данном случае за это время должна подготовиться новая информация на входе D. Сигнал синхронизации  $C_1$ , проходя КС2 с минимальной задержкой, действует на вход синхронизации триггера, заставляя его принять готовую информацию. Причём на активном интервале действия синхросигнала информационный сигнал меняться не должен. В простейшем случае синхросигнал действует непосредственно на вход С триггера, то есть КС2 отсутствует. Рассмотрим вариант синтеза триггера с заданными функциональными возможностями. Предположим, что под действием управляющих сигналов  $Y_1, Y_2, Y_3$ , из которых только один в данный интервал времени

может быть активен, то есть равен единице, выполняются заданные в таблице 3.11 микрооперации.

Таблица 3.11

Управляющий сигнал	Микрооперация	q	z	Примечание
Y1	$Q:=\overline{Q}=B$ (рис.3.34,а)	B	C1	Инвертирование состояния триггера
Y2	$Q:=A$	A	C1	Запись A
Y3	$Q:=1$	1	C1	Запись единицы

Анализируя микрооперации, приходим к выводу, что нужен двухступенчатый триггер или триггер с динамическим управлением, поскольку в момент действия управляющего сигнала Y1 для инвертирования состояния триггера следует на его вход подать сигнал с его же выхода. Выберем с учётом сказанного D-триггер с динамическим управлением (рисунок 3.34,а), для которого необходимо синтезировать комбинационные схемы с выходными сигналами q и z.

Предложенная таблица в процессе проектирования схемы расширяется добавлением значений функций возбуждения триггера, то есть добавлением значений сигналов q и z для каждой строчки таблицы. синтезируем соответствующие схемы. С учётом значений функций возбуждения на входах триггера легко записать булевы функции. Для сигнала q справедливо:

$$q=Y1\&B\vee Y2\&A\vee Y3\&1.$$

Для сигнала z справедливо:

$$z=(Y1\vee Y2\vee Y3)\&C1.$$

Функциональная схема синтезированного триггера предлагается на рисунке 3.35.

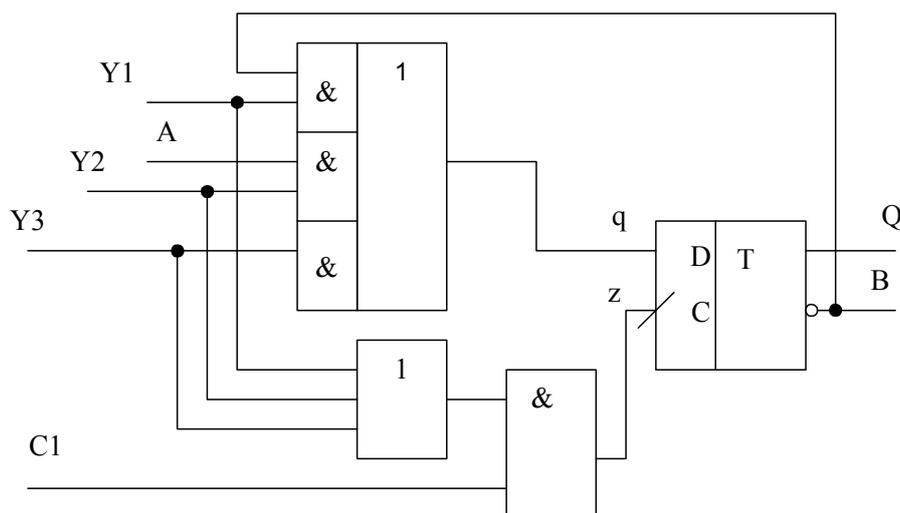


Рис. 3.35. – Функциональная схема синтезированного триггера

Если количество коммутируемых сигналов на входе D велико, то входной сигнал можно сформировать с помощью мультиплексора, на информационные входы которого подаются информационные сигналы по усмотрению проектировщика, а затем синтезируются значения сигналов на входах селекции данного мультиплексора с учётом полученного распределения информационных сигналов.

### Контрольные вопросы

1. Что такое триггер?
2. Какими таблицами можно описать поведение триггера?
3. Какова разница между активным и пассивным сигналами?
4. В чём разница между синхронным и асинхронным триггером?
5. Чем отличаются триггеры с потенциальным и динамическим управлением?
6. На что указывает наклонная черта на входе синхронизации?
7. Какие части импульса могут быть активными?
8. Какая комбинация входных сигналов триггера считается запрещённой?
9. Могут ли элементы триггера выйти из строя при подаче запрещённой комбинации?
10. Какой логический элемент следует использовать на входе триггера, чтобы изменить активность сигнала по этому входу?

11. Какое минимальное число входов у асинхронного RS-триггера и опишите их назначение?
12. На каких элементах строятся асинхронные RS-триггеры и определите для каждого случая запрещённую комбинацию сигналов?
13. Какова минимальная продолжительность активного сигнала на входе асинхронного триггера, чтобы триггер устойчиво перешёл в противоположное состояние?
14. При каких условиях возможна генерация импульсов в триггере?
15. Объясните поведение триггера, предложенное на рисунке 3.7.
16. Объясните различие в условных обозначениях входов RS-триггеров на элементах И-НЕ и ИЛИ-НЕ.
17. Объясните причину использования более сложных двухступенчатых триггеров и триггеров с динамическим управлением вместо простейших одноступенчатых?
18. Что добавляется к асинхронным триггерам для создания синхронных триггеров?
19. Объясните наличие на временных диаграммах (рисунок 3.15) интервала непредсказуемого поведения триггера.
20. Объясните, почему меняется состояние триггера (рисунок 3.16) в момент действия активного синхросигнала?
21. Почему D-триггер (рисунок 3.18) не имеет запрещённого состояния входных сигналов?
22. Объясните работу триггера, предложенного на рисунке 3.20.
23. Могут ли оказаться в запрещённом состоянии D-триггер, предложенный на рисунке 3.30 и JK-триггер, предложенный на рисунке 3.27?
24. Объясните временные диаграммы, предложенные на рисунке 3.23.
25. Импульсы с какого входа считает синхронный T-триггер: импульсы, поступающие на вход T или на вход C?
26. Как соотносятся частоты входных и выходных импульсов для триггера со счётным входом?

27. Как D-триггер с динамическим управлением превратить в T-триггер?

## 4. Накапливающие узлы

### 4.1. Общие сведения

Под накапливающими узлами понимают узлы, которые содержат в своём составе элементы памяти, а именно – триггеры. Можно выделить два вида накапливающих узлов – регистры и счётчики.

Регистр – накапливающий узел, предназначенный для приема, хранения, преобразования и выдачи двоичной информации.

Счётчик - накапливающий узел, предназначенный для подсчёта событий, подсчёта импульсов, поступающих на счётный вход.

Среди множества входных сигналов накапливающего узла можно выделить информационные, управляющие, синхронизирующие. Под действием входных сигналов узел способен выполнять ту микрооперацию, на которую он настроен предварительной подачей определённой комбинации управляющих сигналов. Собственно микрооперация завершается изменением состояния узла синхронно под действием сигнала синхронизации или асинхронно при подаче сигнала на асинхронный вход предварительной установки узла в определённое состояние.

Выполняемые узлом микрооперации могут быть любые, какие требуются для исполнения заданного алгоритма. Поэтому часто счётчик, например, одновременно может принимать параллельно информацию во все разряды, то есть выполнять микрооперацию, характерную для регистра, и наоборот, регистр проектируют таким образом, что он способен выполнять и функцию подсчёта числа импульсов. То есть, в одном узле могут быть реализованы как функции регистра, так и счетчика. Сложные микрооперации могут потребовать привлечения дополнительных комбинационных узлов для реализации микроопераций, на которые не рассчитана используемая микросхема регистра или счётчика.

Управляющие сигналы позволяют выбрать режим работы узла, то есть задают требуемую микрооперацию. Непосредственно микрооперация завершается в момент действия синхросигнала или в момент прихода сигнала на асинхронный

вход узла. Если узел содержит информационные входы параллельного приёма информации, то они могут нумероваться по порядку или с учётом их веса. Выходные цепи узла могут быть организованы по схеме с тремя состояниями [6], что потребует наличия дополнительного входа управления выходными усилителями.

## 4.2. Регистры

### 4.2.1. Простейшие регистры

Простейшие регистры способны выполнять минимальное количество микроопераций типа параллельного приёма информации. Они могут быть построены на любых триггерах. Рассмотрим вариант реализации регистра на D-триггерах. Двухразрядный регистр и его графическое изображение предложены на рисунке 4.1.

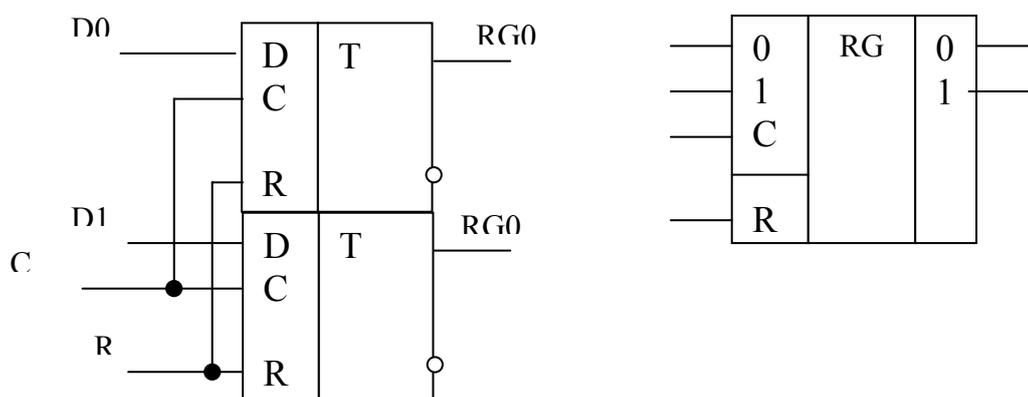


Рис. 4.1. – Схема простейшего регистра и его условное обозначение

Предложенный регистр строится с использованием одноступенчатых триггеров с потенциальным управлением. Если регистр или счётчик реализуется на триггерах с динамическим управлением или двухступенчатых триггерах, то вход синхронизации помечается как динамический, фронт срабатывания по которому соответствует моменту появления новой информации на выходах регистра. Поведение узла может быть описано временными диаграммами, на которых каждый сигнал представлен своей линией. Если узел многоразрядный, то потребуется изображение большого числа линий, что оказывается неудобным при анализе работы. В этом случае возможно групповое изображение входных и выходных информационных сигналов с указанием в шестнадцатеричной или другой системе

состояния узла в текущий момент времени. На рисунке 4.2 предлагается вариант временных диаграмм четырёхразрядного регистра с информационными входами  $D[3/0]$  и выходами  $RG[3/0]$ .

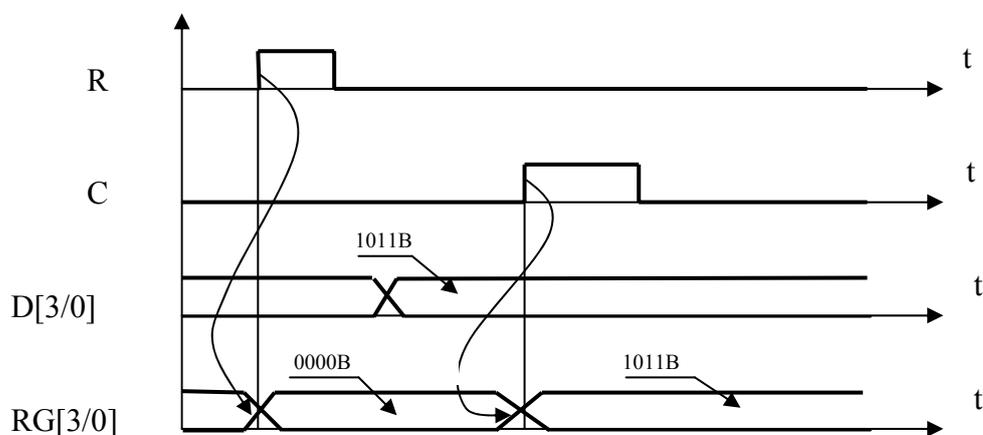


Рис. 4.2. – Временные диаграммы работы четырёхразрядного регистра

По сигналу сброса  $R$  регистр сбрасывается в нулевое состояние, которое в двоичной системе представлено на временных диаграммах числом  $0000B$ . В некоторый момент времени входная информация меняет своё состояние, что выражено двоичным числом  $1011B$ . В момент прихода активного сигнала синхронизации происходит занесение входной информации, в результате чего на выходах регистра формируется то же число  $1011B$ .

#### 4.2.2. Регистры сдвига

Регистры сдвига строятся с использованием двухступенчатых триггеров или триггеров с динамическим управлением. При каждом сдвиге в сторону старших разрядов двоичное число оказывается умноженным на два, а при сдвиге в сторону младших разрядов происходит деление на два. Простейший регистр сдвига осуществляет сдвиг только в одном направлении, которое указывается стрелкой в символе выполняемой регистром функции. Если регистр сдвигает слева направо или сверху вниз, то символ выполняемой функции выглядит следующим образом:  $RG>$ . Если же регистр сдвигает снизу вверх или справа налево, то символ функ-

ции выглядит так:  $RG<$ . На рисунке 4.3 предложена схема простейшего регистра на D-триггерах с динамическим управлением.

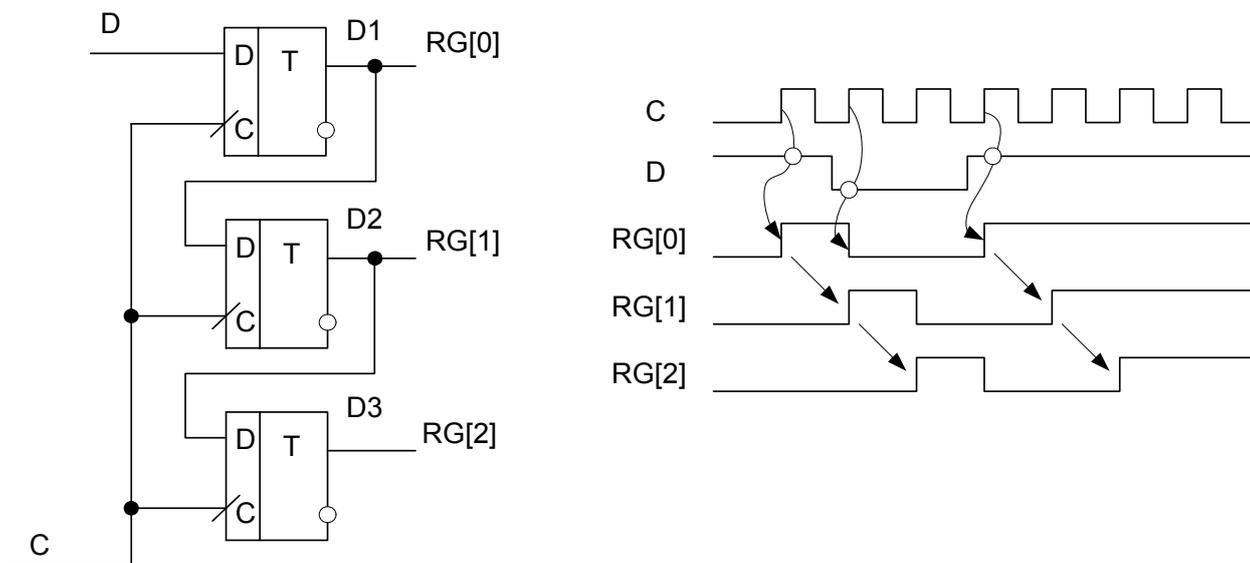


Рис. 4.3. –Схема простейшего сдвигающего регистра и временные диаграммы его работы

На этом же рисунке предложены временные диаграммы его работы.

Условное обозначение четырёхразрядного регистра сдвига, построенного по предложенной схеме, показано на рисунке 4.4.

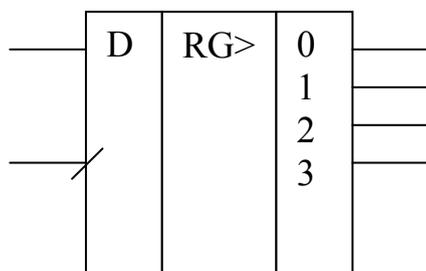


Рис. 4.4. – Условное обозначение сдвигающего регистра

При анализе поведения триггеров в сдвигающем регистре необходимо помнить, что с приходом фронта синхросигнала одновременно меняют свои состояния все разряды регистра. А поскольку временные диаграммы строятся без учёта задержки сигналов в триггерах регистра, необходимо при определении состояния разряда учитывать состояние сигнала на входе D этого разряда непосредственно перед действием активного фронта синхросигнала. Зависимость состояния разряда от конкретного уровня входного сигнала показывается на временных диаграм-

мах кружочком на определённом уровне сигнала, расположенном одновременно на стрелке зависимости сигнала на выходе от фронта синхросигнала.

### 4.2.3. Реверсивный сдвигающий регистр

Реверсивный регистр сдвига позволяет сдвигать хранимую информацию влево или вправо, следовательно, может выполнять 2 микрооперации:

$$RG:=LS(RG, 1),$$

$$RG:=RS(RG,1).$$

Первая микрооперация представляет сдвиг содержимого регистра влево на один разряд, а вторая – сдвиг вправо на один разряд. Если реверсивный регистр сдвига реализуется в виде микросхемы, то обычно предусматривают у него и микрооперацию параллельного занесения информации. То есть такой регистр способен выполнять и микрооперацию приёма информации с информационных входов в параллельном виде:  $RG:=D$ .

Поскольку один и тот же регистр способен выполнять множество микроопераций, требуется формирование у него управляющих входов, состояние сигналов на которых определяет поведение регистра, то есть выполняемую под действием синхросигнала микрооперацию. Количество выполняемых микроопераций определяет число управляющих входов.

Синтезируем один разряд реверсивного регистра сдвига, способного выполнять кроме сдвига микрооперацию параллельного приёма. Предполагаем применение в регистре D-триггеров с динамическим управлением. Для выбора режима работы регистра используем два управляющих сигнала:  $E0$  и  $E1$ . Пусть при  $E0=E1=1$  выполняется микрооперация параллельного приёма, при  $E0=E1=0$  регистр хранит предыдущее состояние, при двух других комбинациях управляющих сигналов реализуются сдвиги в разных направлениях. Заметим, что микрооперация «хранение» может быть реализована или путём запрета записи или путём перезаписи состояния регистра. Запрет записи может быть реализован запретом

действия импульса синхронизации. Выберем для реализации путь перезаписи, повторения состояния регистра.

Синтез сводится к синтезу комбинационной схемы на входе D триггера. Пусть номер разряда  $i$ . Определим функцию возбуждения для информационного входа триггера:

$$d_i = A_i \cdot E1 \cdot E0 \vee RG_{i+1} \cdot \bar{E1} \cdot E0 \vee RG_{i-1} \cdot E1 \cdot \bar{E0} \vee RG_i \cdot \bar{E1} \cdot \bar{E0} ,$$

где  $A_i$  – соответствующий разряд записываемого параллельно в регистр слова  $A$ .

Проверим справедливость предложенного выражения. Если  $E0=E1=1$  функция возбуждения на входе D триггера равна:  $d_i = A_i$ . То есть происходит запись нового числа в регистр по активному сигналу синхронизации. Если  $E0=E1=0$ , то справедливо:  $d_i = RG_i$ . С прямого выхода триггера данного разряда переписывается информация на тот же триггер. В результате триггер не меняет своего состояния. При  $E1 \neq E0$  происходит запись в данный разряд информации с одного из соседних разрядов. Таким образом, предложенное выражение справедливо. Используем на входе D триггера  $i$  – того разряда мультиплексор на четыре входа с сохранением функции управляющих сигналов. Для этого случая схема одного разряда регистра, выполняющего все требуемые микрооперации, предложена на рисунке 4.5.

В сериях элементов ТТЛ имеется микросхема восьмиразрядного регистра типа К555ИР13, которая реализует рассмотренный набор микроопераций и микрооперацию сброса регистра в ноль с помощью асинхронного входа.

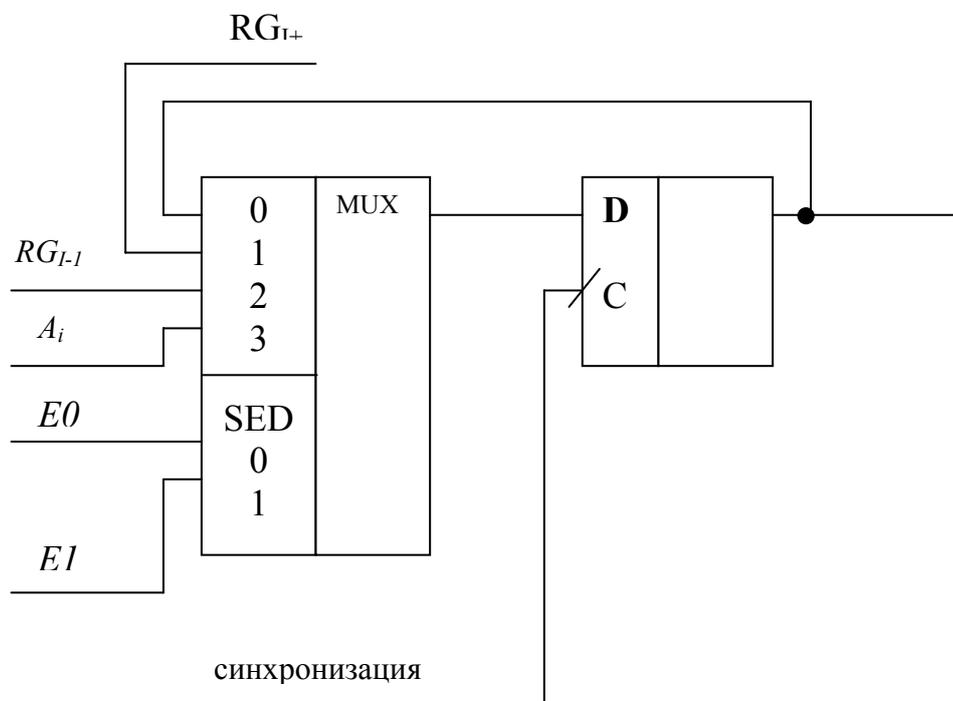


Рис. 4.5. – Схема одного разряда реверсивного регистра сдвига

### 4.3. Счётчики

#### 4.3.1. Общие сведения

В зависимости от направления счёта различают счётчики суммирующие, вычитающие и реверсивные, направление счёта в которых зависит или от управляющего сигнала, или от того, на какой вход: суммирующий или вычитающий, подаются импульсы, которые подвергаются счёту. В последнем случае суммирующий вход помечают символом «+1», а вычитающий вход помечают символом «-1». Если направление счёта зависит от состояния сигнала на управляющем входе, то обычно подвергаются счёту импульсы, подаваемые на вход синхронизации микросхемы.

В виде микросхемы реализуют суммирующие и реверсивные счётчики. То есть вычитающий счётчик как отдельное изделие обычно не реализуется. Объясняется это тем, что вычитающий счётчик можно реализовать на суммирующем, у которого результат счёта снимается с инверсных выходов триггеров или используются инверторы на прямых выходах триггеров.

*(Проверьте этот факт. Для этого составьте таблицу состояний мало-разрядного счётчика при суммировании, а затем повторите эту таблицу, заменяя нули единицами и наоборот. Проанализируйте полученную в результате преобразования таблицу).*

В зависимости от основания счёта различают двоичные счётчики, десятичные и счётчики по произвольному основанию. Используемый символ функции в условном обозначении зависит от основания счёта. В условном обозначении двоичного счётчика функцию обозначают сочетанием «СТ2», в десятичном счётчике функцию обозначают «СТ10», при основании счёта N функцию обозначают «СТN».

В зависимости от организации цепей переноса различают счётчики с последовательным переносом, параллельным, сквозным и смешанным переносом.

Счётчики строятся на Т-триггерах, причём, счётчики с последовательным переносом используют асинхронные Т-триггеры, а остальные счётчики обычно строятся на синхронных Т-триггерах.

#### **4.3.2. Счётчики с последовательным переносом**

Счётчики с последовательным переносом обладают минимальным быстродействием, при изменении состояния многих триггеров в счётчике эти триггеры меняют состояния последовательно. В результате счётчик перебирает во время переключения все в этом случае возможные состояния.

Подобные счётчики удобны тем, что легко реализуются с любой разрядностью на отдельных D-триггерах с динамическим управлением или двухступенчатых или на JK-триггерах. Рассмотрим некоторые возможные схемные решения на

D-триггерах и на JK-триггерах. На рисунке 4.6 предложена схема вычитающего двоичного счётчика на D-триггерах и временные диаграммы его работы.

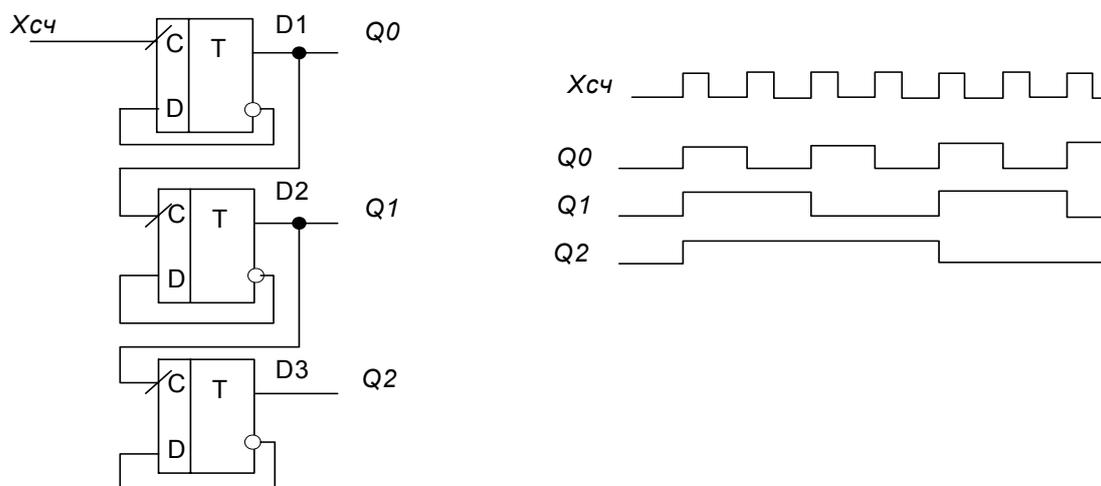


Рис. 4.6. – Вычитающий счётчик на D-триггерах и временные диаграммы его работы

Поскольку перенос последовательный, временные диаграммы могут строиться последовательно. То есть при заданном входном сигнале  $X_{сч}$  сначала строится сигнал на выходе младшего разряда  $Q_0$  на протяжении всей временной оси, затем строится сигнал на выходе  $Q_1$  с учётом того, что на счётный вход триггера  $Q_1$  поступают импульсы с выхода  $Q_0$ . В последнюю очередь строится сигнал старшего разряда  $Q_2$ . Состояние счётчика определяется числом  $Q_2Q_1Q_0$ , где  $Q_2$  – старший разряд. Анализируя временные диаграммы, можно убедиться, что счётчик именно вычитающий. Действительно, в исходном состоянии предполагается состояние 000 (смотрите начало временных диаграмм). После действия первого импульса по его нарастающему фронту переходит в единичное состояние сначала триггер  $Q_0$ , затем под действием нарастающего фронта с выхода  $Q_0$  переходит в единичное состояние триггер  $Q_1$ , затем  $Q_2$ . Счётчик окажется в состоянии 111. следующий импульс меняет состояние только младшего разряда и счётчик окажется в состоянии 110. Таким образом, состояние счётчика с каждым импульсом уменьшается на единицу. Счётчик оказывается вычитающим.

Чтобы построить суммирующий счётчик достаточно к входу синхронизации каждого следующего триггера подключить инверсный выход предыдущего триггера или снять информацию с инверсных выходов триггеров.

На рисунке 4.7 предложена схема суммирующего счётчика на JK-триггерах и временные диаграммы его работы.

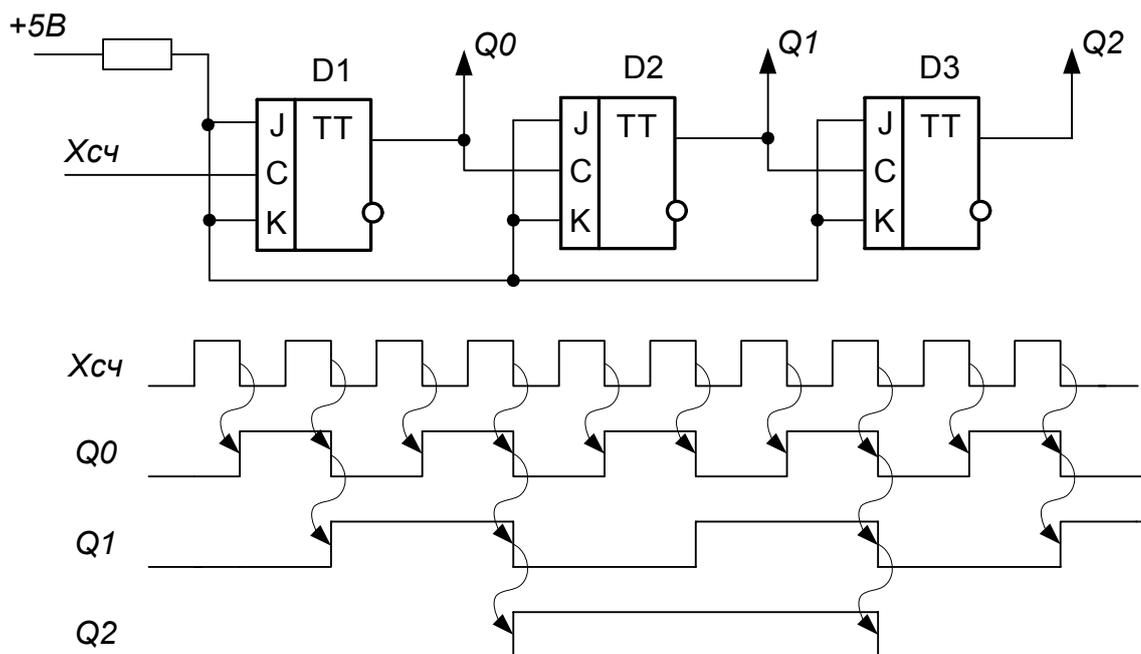


Рис. 4.7. – Суммирующий счётчик на JK-триггерах и временные диаграммы его работы

Предполагается, что JK-триггеры выполнены по технологии ТТЛ. Для того, чтобы превратить JK-триггеры в асинхронные Т-триггеры, на входы J и K следует подать уровень логической единицы. Для ТТЛ уровень единицы формируется подключением входов J и K к источнику напряжения 5В через резистор, как это сделано на схеме. Поскольку триггеры двухступенчатые, выходные триггеры меняют состояния по спадающему фронту сигнала, подаваемого на синхровходы. Учитывая это, можно проверить правильность предложенных диаграмм. Как и в предыдущем случае временные диаграммы строятся сначала для Q0, затем для

Q1, потом для Q2. По временным диаграммам можно убедиться, что счётчик суммирующий.

### 4.3.3. Счётчики с параллельным и сквозным переносом

Анализируя работу суммирующего счётчика, можно заметить, что переключение разряда с номером  $i$  возможно только при наличии в предыдущих разрядах единиц. Если в отмеченном разряде используется синхронный Т-триггер, то функция переноса в данный разряд определится из выражения:

$$T_i = Q_0 \cdot Q_1 \cdot Q_2 \cdot \dots \cdot Q_{(i-1)}.$$

Сигнал переноса может формироваться с помощью многовходового конъюнктора, число входов которого определяется номером разряда, для которого формируется перенос. Если все сигналы переноса формируются таким образом, то при использовании синхронных Т-триггеров можно построить счётчик с параллельным переносом, схема которого предложена на рисунке 4.8.

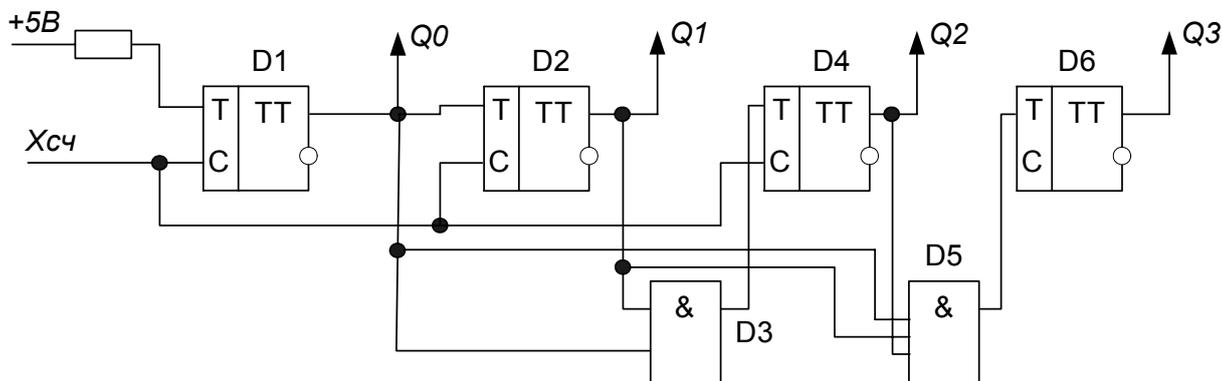


Рис. 4.8. - Счётчик с параллельным переносом на синхронных Т-триггерах

Используемые в схеме синхронные Т-триггеры могут быть реализованы на JK-триггерах или на D-триггерах с динамическим управлением или двухступенчатых.

Чтобы счётчик был вычитающим, следует при применённых Т-триггерах для формирования переноса использовать сигналы с инверсных выходов.

Счётчик с параллельным переносом отличается максимальным быстродействием.



чики синтезировать не надо, поскольку они существуют в виде микросхем. Более того, имеются микросхемы, основание счёта которых зависит от состояния сигнала на входе режима работы. Изменение состояния этого сигнала меняет режим с режима двоичного счёта на режим десятичного счёта.

Необходимость синтеза счётчиков с произвольным основанием ограничивает их применение.

Существует другой подход к проблеме синтеза счётчиков с произвольным основанием: использование цепей обратной связи для приведения счётчика в исходное состояние при достижении основания счёта.

В простейшем случае счётчик с произвольным основанием можно построить, используя микросхему счётчика, у которого имеется асинхронный вход сброса в нулевое состояние. Чтобы счётчик считал до  $N$ , достаточно в момент достижения в процессе счёта состояния  $N$  сформировать с помощью выходного состояния счётчика в этот момент и комбинационной схемы активный сигнал на входе сброса. Как только счётчик в процессе счёта переберёт все состояния от нуля до  $N-1$ , то с приходом очередного импульса на счётный вход счётчик достигает состояния  $N$  и происходит сброс его в нулевое состояние. То есть после состояния  $N-1$  счётчик к приходу следующего импульса окажется в нулевом состоянии и счёт начнётся заново.

Построим счётчик по основанию 6. Этот счётчик должен перебирать состояния от 000 до 101. Как только счётчик окажется в состоянии 6, он должен сброситься в нулевое состояние. Цифре 6 соответствует состояние  $Q_2Q_1Q_0=110$  в двоичной системе счисления. Легко сформировать сигнал сброса  $R$ , используя выходные сигналы счётчика в этом состоянии:

$$R = Q_2 \cdot Q_1 \cdot \bar{Q}_0.$$

Учитывая тот факт, что следующие состояния после основания счёта невозможны, функция сброса может принимать во внимание только те сигналы, которые при достижении основания счёта равны единице. В нашем случае будет справедливо следующее выражение:  $R = Q_2 \cdot Q_1$ .

Схема счётчика по основанию 6 и временные диаграммы его работы предлагаются на рисунке 4.10.

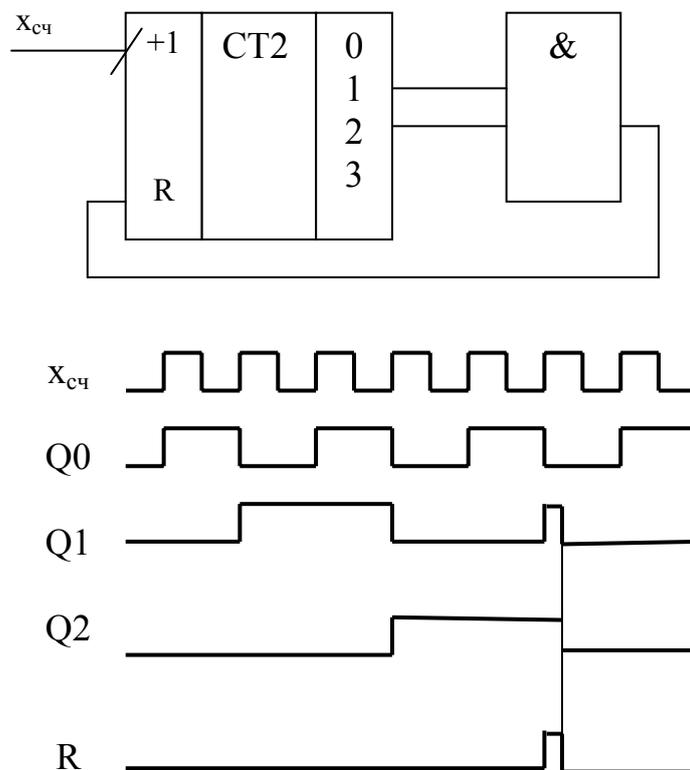


Рис. 4.10. – Счётчик по основанию шесть

Как только счётчик досчитает до шести, на выходе конъюнктора сформируется уровень единицы, который поступит на вход сброса счётчика. Это приведёт к сбросу счётчика в нулевое состояние, что в свою очередь приведёт к окончанию действия уровня единицы на выходе конъюнктора.

## Контрольные вопросы

1. Какие узлы называют накапливающими?
2. Что такое регистр и какие разновидности регистров вы знаете?
3. Что такое счётчик и какие разновидности счётчиков вы знаете?
4. Почему не проектируют микросхемы вычитающих счётчиков?
5. Предложите схему сдвигающего регистра на D-триггерах и на JK-триггерах.
6. Предложите схемы суммирующих и вычитающих счётчиков на D-триггерах и JK-триггерах.
7. Какие D-триггеры можно использовать в сдвигающих регистрах и счётчиках?
8. Какие разновидности цепей переноса вы знаете?
9. Счётчики с каким переносом обладают минимальным и максимальным быстродействием?
10. Как реализуется цепь переноса в счётчике со сквозным переносом?
11. Опишите принцип построения простейших счётчиков по произвольному основанию.
12. В каких счётчиках используются синхронные T-триггеры?
13. Объясните схемное решение одного разряда реверсивного сдвигающего регистра.
14. Опишите принцип организации реверсивного счётчика.

### Рекомендуемая литература

1. Н.П.Сергеев, Н.П.Вашкевич. Основы вычислительной техники. М.,ВШ, 1988.- 312с.
2. Угрюмов Е.П. Цифровая схемотехника. -СПб.: БХВ-Санкт-Петербург, 2001. (учебное пособие для вузов).
3. В.И.Карлащук. Электронная лаборатория на IBM PC. Программа Electronics Workbench и её применение. – М.: Солон-Р, 1999.
4. Л.А.Брякин. Проектирование устройств ЭВМ на микросхемах повышенной интеграции. Лабораторный практикум. РИО ППИ, Пенза. 1991г.
5. В.Л.Шило. Популярныe цифровые микросхемы. М.: Радио и связь, 1987.
6. Л.А.Брякин. Электротехника и электроника: Конспект лекций. – Пенза: Изд-во Пенз. гос. ун-та., 2004. – 156с..