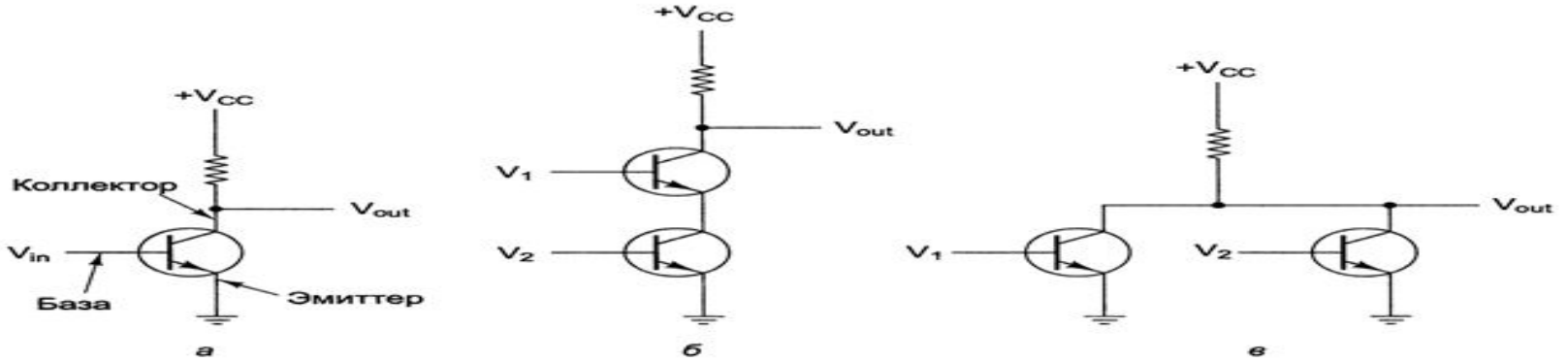
Базовые логические функции



Транзисторный инвертор (*а*); вентиль НЕ И (*б*); вентиль НЕ ИЛИ (*в*)

A	X
0	1
1	0

а

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

б

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

в

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

г

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

д

Значки для изображения пяти основных вентилей.
(элементарные логические функции)
Режимы работы функции для каждого вентиля

Основные понятия алгебры логики

- **Алгебры логики** - раздел математической логики, в котором изучаются логические операции над выражениями, представленными в двоичной форме(истина-ложь, ноль-единица).
- Алгебра логики находят широкое применение при синтезе и анализа схем ЭВМ, так как информация представляется в двоичном виде и реализуется с помощью физических элементов которые могут пропускать или не пропускать ток, иметь на выходе высокий или низкий уровень сигнала (напряжения или тока – ноль или единицу) .

Основные понятия алгебры логики

- *логическая переменная* это такая переменная, которая может принимать одно из двух значений: истинно или ложно (да или нет, единица или ноль), что хорошо согласуется с двоичным представление информации в ЭВМ.
- *логическая константа* это такая постоянная величина, значением которой может быть: истинно или ложно (да или нет, единица или ноль).
- *логическая функция* это такая функция, которая может принимать одно из двух значений (истинно или ложно, да или нет, единица или ноль), в зависимости от текущего значений её аргументов, в качестве которых используются логические переменные.

Логическая функция

- Логическая функция может быть одного (n=1) или нескольких (n >1) аргументов. Значение логической функции определяется комбинацией конкретных значений переменных, от которых она зависит. Комбинация конкретных значений переменных (аргументов функции) называется набором. Количество различных наборов N для «n» переменных определяется как:
- $N= 2^n$.

Задание логической функции

- Зависимость логической функции от переменных может задаваться:
- *Словесным описанием*, как правило, может использоваться в случае сравнительно не сложной логической функции.
- *Таблицой истинности* является универсальным средством задания логической функции. Она включает все наборы для заданного количества переменных, определяющих значение логической функции, с указанием значений, которые принимает функция для каждого набора. В одной таблицы истинности может задаваться несколько логических функций, зависящих от одних и тех же переменных
- *В виде логического выражения*.

.

Таблица истинности

Таблица истинности для нескольких функций y_i и всевозможных

№ п.п.	x_1	x_2	x_3	y_1	y_2	y_3	.....	y_n
0	0	0	0	0	1	1		0
1	0	0	1	1	1	0		1
2	0	1	0	1	1	1		0
3	0	1	1	0	1	0		-
4	1	0	0	1	0	1		0
5	1	0	1	0	0	0		1
6	1	1	0	0	0	1		-
7	1	1	1	1	1	0		1

Логическая функция, называются *«полностью определенная»*, если для неё заданы значения по всем возможным наборам. Функции, называются *«частично определенная»,* если для некоторых наборов значения функции не заданы.

В таблице истинности функции y_1, y_2 являются полностью определенными, а функция y_n - частично определенная (знак «-

Логическое выражение

- Логическим выражением называется комбинация логических переменных и констант, связанных элементарными базовыми логическими функциями (или логическими операциями), которые могут разделяться скобками.
- Например, логическую функцию y_1, определенную выше приведенной таблице истинности, можно представить в виде логического выражения:

$$Y1 = (\overline{X1} * \overline{X2} * X3) + (\overline{X1} * X2 * \overline{X3}) + (X1 * \overline{X2} * \overline{X3}) + (X1 + X2 + X3)$$

- где « ¯ », «+», «*» - знаки базовых логических функций

Булевый базис.

- Набор элементарных логических операций, с помощью которого можно задать любую, сколь угодно сложную логическую функцию, называется «функционально полная система логических функций» или базисом.
- Наиболее распространенный, который в качестве базовых логических функций использует функцию одной переменной « НЕ» (функция отрицания), и две функции двух переменных «И» (конъюнкция или логическое умножения) и «ИЛИ» (дизъюнкция или логическое сложение). Эта система получила название система *Булевых* функций или *Булевый базис*.

Основные законы алгебры Буля

- В алгебре Буля используется следующая **приоритетность выполнения операций:**
- **- сначала рассчитываются значения имеющих место отрицаний и скобок,**
- **- затем выполняются операция И (логическое умножение);**
- **- самый низший приоритет имеет операция ИЛИ (логическая сумма)**.
- При работе с булевыми логическим выражениями используются следующие законы и правила.
- ***Переместительный (коммутативный) закон***. Закон справедлив как для конъюнкции, так и для дизъюнкции.
- $x_1 + x_2 + x_3 + x_4 = x_4 + x_3 + x_2 + x_1$ - от перемены мест логических слагаемых сумма не меняется;
- $x_1 * x_2 * x_3 * x_4 = x_4 * x_3 * x_2 * x_1$ - от перемены мест логических сомножителей их произведение не меняется.

Основные законы алгебры Буля

- ***Сочетательный (ассоциативный) закон***. Закон справедлив как для конъюнкции, так и для дизъюнкции.
- $x_1 + x_2 + x_3 + x_4 = (x_2 + x_3) + x_1 + x_4 = (x_1 + x_4) + (x_2 + x_3)$ - при логическом сложения отдельные слагаемые можно заменить их суммой;
- $x_1 * x_2 * x_3 * x_4 = (x_2 * x_3) * x_1 * x_4 = (x_1 * x_4) * (x_2 * x_3)$ - при логическом умножении отдельные логические сомножители можно заменить их произведением.
- ***Распределительный (дистрибутивный) закон.***
- $(x_1 + x_2) * x_3 = x_1 * x_3 + x_2 * x_3$.
- $(x_1 + x_2) * (x_1 + x_3) = x_1 + x_2 * x_3$.

Правило де Моргана

- $\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$ - отрицание суммы равно произведению отрицаний;

- $\bar{x}_1 \bar{x}_2 = \overline{x_1 x_2}$ -отрицание произведения равно сумме отрицаний. Правило справедливо при любом числе логических операндов.

Операция склеивания

- *Операция склеивания:*
- $x_1 A + A\bar{x}_1 = A$- операция *склеивания* для конъюнкций
- $(\bar{x}i + A)(xi + A)$ =A - операция склеивания для дизъюнкций,

- *Операции с отрицаниями:*
- $\bar{\bar{x}} = x$ - двойное отрицание равносильно отсутствию отрицания;

$$\bar{x} + x = 0$$

$$\bar{x} * x = 0$$

Формы представления логических выражений

- Одну и туже логическую функцию можно представить различными логическими выражениями. Среди множества выражений, которыми представляется логическая функция особое место занимают две формы:
- *совершенная конъюнктивная нормальная форма* (СКНФ),
- *совершенная дизъюнктивная нормальная форма* (СДНФ).
- *Совершенная дизъюнктивная нормальная форма* представляет собой дизъюнкцию простых конъюнкций, где под термином простая *конъюнкция* имеется в виду конъюнкция переменных или их отрицаний.

СДНФ

- ***СДНФ - Совершенная дизъюнктивная нормальная форма*** представляет собой дизъюнкцию простых конъюнкций (сумма произведений).
- В СДНФ простые конъюнкции содержат все переменные в своей прямой или инверсной форме и отражают собой наборы, на которых представляемая функция имеет **единичное значение**. Такие конъюнкции называются конституентами единицы

N стр	x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	0	1	1	1
1	0	0	1	1	0	1
2	0	1	0	0	0	0
3	0	1	1	1	1	1
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	0	1
7	1	1	1	1	0	1

$$y_1 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$$

$$y_2 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2x_3;$$

СКНФ

- ***Совершенная конъюнктивная нормальная форма*** **это конъюнкция простых дизъюнкций (произведение сумм).**
- В СКНФ простые дизъюнкции содержат все переменные в своей прямой или инверсной форме и отражают собой наборы, на которых представляемая функция имеет **нулевое значение**. и представляют

N стр	x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	0	1	1	1
1	0	0	1	1	0	1
2	0	1	0	0	0	0
3	0	1	1	1	1	1
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	0	1
7	1	1	1	1	0	1

$y_1 = (x_1 + \bar{x}_2 + x_3)\ (\bar{x}_1 + x_2 + \bar{x}_3)\ (\bar{x}_1 + x_2 + x_3)$

$y_2 = (x_1 + x_2 + \bar{x}_3)\ (x_1 + \bar{x}_2 + x_3)\ (\bar{x}_1 + x_2 + x_3)\ (\bar{x}_1 + x_2 + \bar{x}_3)$

Минимизация логических выражений

- Учитывая то, что одну и ту же логическую функцию можно представить различными выражениями, перед реализацией функции в виде логической схемой весьма важным является **выбор из всех возможных выражений, соответствующих данной функции, самое «простое».** Решить эту проблему можно за счет использования процедуры **минимизации логического выражения**.

Карты Карно - Вейча

- Карта Карно для «n» логических переменных представляет собой множество квадратов (клеток), объединённых в близкую к квадрату прямоугольную форму. Каждая такая клетка соответствует одному набору логических переменных, причем наборы двух соседних клеток должны отличаться на значение одной переменной (представляются в коде Грея и образуют склеивающиеся наборы).

- Карта Карно задает своего рода таблицу истинности.

- Записываемая функция должна быть представлена в СДНФ(СКНФ). Запись функции в карту осуществляется за счет установки «1»(0 - СКНФ) в те клетки карты, где функция принимает единичное (нулевое) значение.

Этапы минимизации

- Для выполнения минимизации представленной в карте Карно функции необходимо выполнить два этапа:

1) **охватить множество клеток карты Карно контурами;**

2) **записать минимальное выражение для заданной функции в виде дизъюнкции конъюнкций для СДНФ (или конъюнкция дизъюнкций для СКНФ), где каждая конъюнкция (дизъюнкция) соответствует одному из введенных на карте контуров.**

Логические соседи

- ***Логическими соседями*** **являются такие две клетки, наборы которых отличаются только одной переменной - в одной эта переменная должна иметь прямое, в другой - обратное значение.**
- **Для того, чтобы быть логическими соседями, клеткам достаточно быть *геометрическими соседями*.**

Логические соседи

$X_1 X_2$ \ $X_3 X_4$	00	01	11	10
00	1			1
01		1	1	
11				
10	1			1

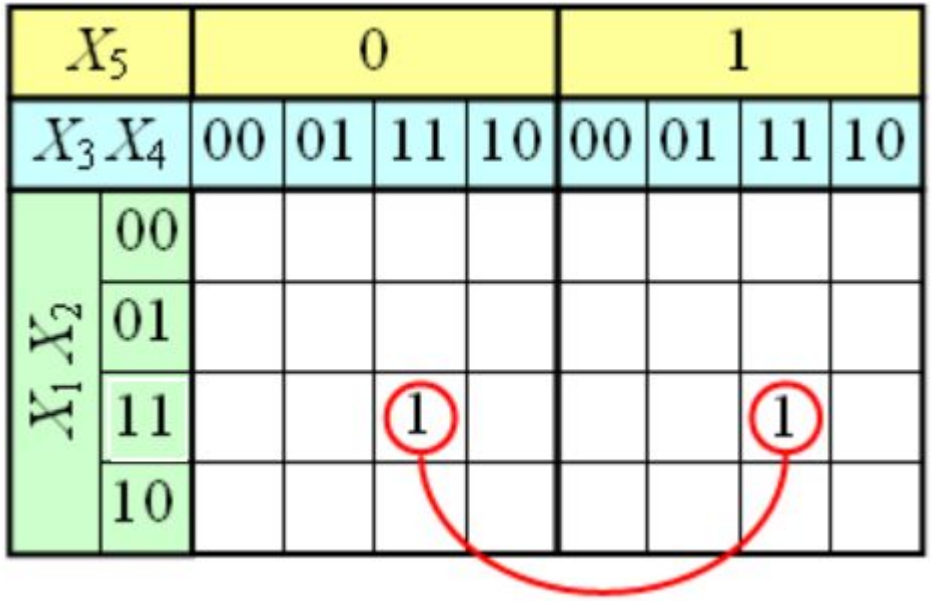
Карта является пространственным объектом и заворачивается по горизонтали и вертикали, сливаясь своими крайними горизонтальными и крайними вертикальными границами.

Можно считать, что соответствующие крайние горизонтальные и вертикальные клетки являются геометрическими соседями

Логические соседи

- Логическими соседями могут быть клетки, которые не являются геометрическими соседями. К числу таких клеток относятся клетки, которые по горизонтали или вертикали симметричны относительно линий зеркального отображения.

Логические соседи



Для таблиц 5 и более переменных нужно учитывать, что квадраты 4x4 виртуально находятся друг над другом в третьем измерении, поэтому соответственные клетки двух соседних квадратов 4x4 являются соседними, и соответствующие им клетки можно склеивать.

Правила охвата клеток

- **Охват клеток** карты контурами выполняется с соблюдением следующих правил:

1. **контурами необходимо охватить все клетки с единичными (нулевыми для СКНФ) значениями;**
2. **контур должен иметь прямоугольную форму;**
3. **в контур может входить такое количество клеток, которое равно целой степени числа «2»;**
4. **в контур могут входить клетки, являющиеся логическими соседями;**
5. **в контур необходимо включить максимальное количество клеток с учетом выше приведенных требований;**
6. **контуров должно быть минимальное количество.**

Правила записи

- Запись минимального выражения заданной функции имеет вид дизъюнкции простых конъюнкций для СДНФ (конъюнкцию дизъюнкций для СКНФ), и формируется следующим образом:
- **- соответствующая контуру конъюнкция (дизъюнкция для СКНФ), должна включать, только те переменные, которые имеют постоянное значение во всех клетках, охваченных рассматриваемым контуром;**
- **- переменные, которые имеют разные значения для клеток, охваченных рассматриваемым контуром склеиваются и не должны входить в конъюнкцию**

Функциональные узлы ЭВМ комбинационного типа

Шифратор

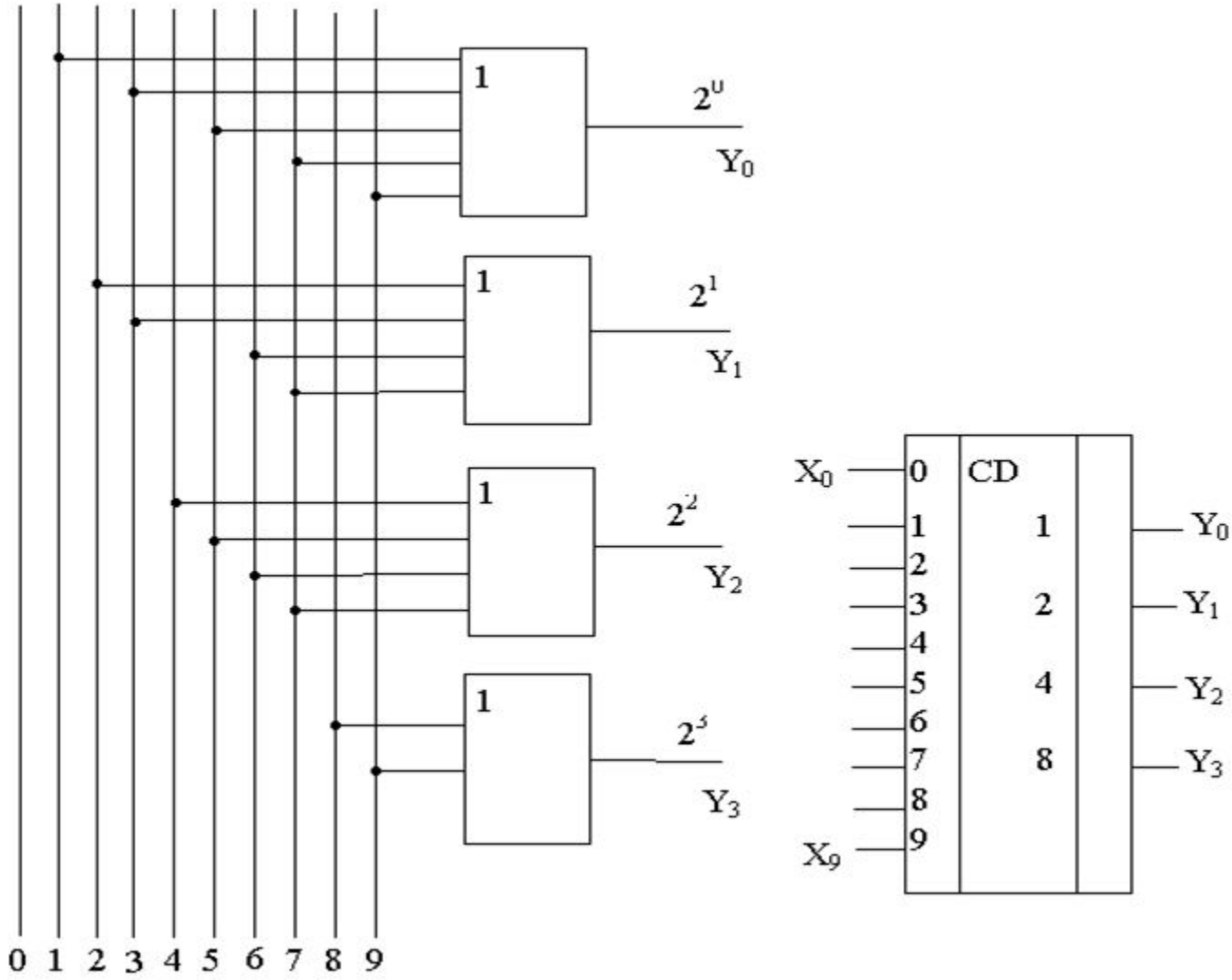
- **Шифратор (кодер) - это устройство, преобразующее m- разрядный позиционный код в n- разрядный двоичный код. В позиционном коде число определяется позицией единиц в серии нулей, или позицией нуля в серии единиц**.
- **(Или проще - единичный сигнал на одном из входов в n-разрядный двоичный код).**
- Наибольшее применение он находит в устройствах ввода информации (пультах управления) для преобразования десятичных чисел в двоичную систему счисления.

Шифратор

Входы	Выходы			
X	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

- **Предположим, на пульте десять клавиш с гравировкой от 0 до 9. При нажатии любой из них на вход шифратора подается единичный сигнал (X_0, ..., X_9). На выходе шифратора должен появиться двоичный код (Y_0, ..., Y_9) этого десятичного числа.**
- **Как видно из таблицы истинности, в этом случае нужен преобразователь с десятью входами и четырьмя выходами.**

Реализация шифратора



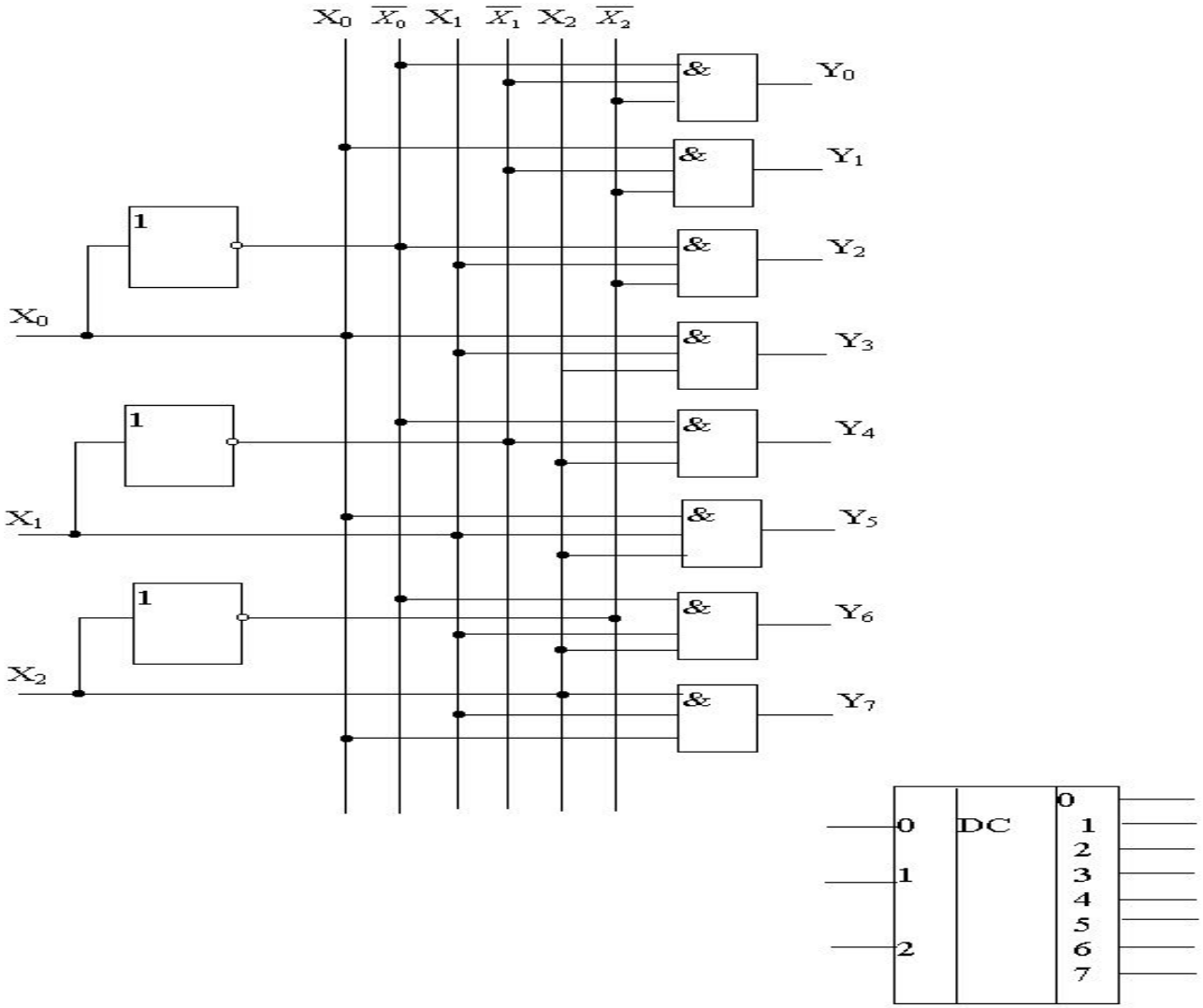
Дешифраторы

- **Дешифратор (декодер) - устройство, преобразующее n – разрядный двоичный код в m - разрядный позиционный код**
- (преобразует n - разрядный двоичный код, поступающий на его входы, в сигнал только на одном из его выходов)
- **Дешифратор двоичного n-разрядного кода имеет 2^n выходов, т.к. каждому из 2^n значений входного кода должен соответствовать единичный сигнал на одном из выходов дешифратора.**
- Дешифраторы широко применяются в устройствах управления, для построения распределителей импульсов по различным цепям и т.д
-

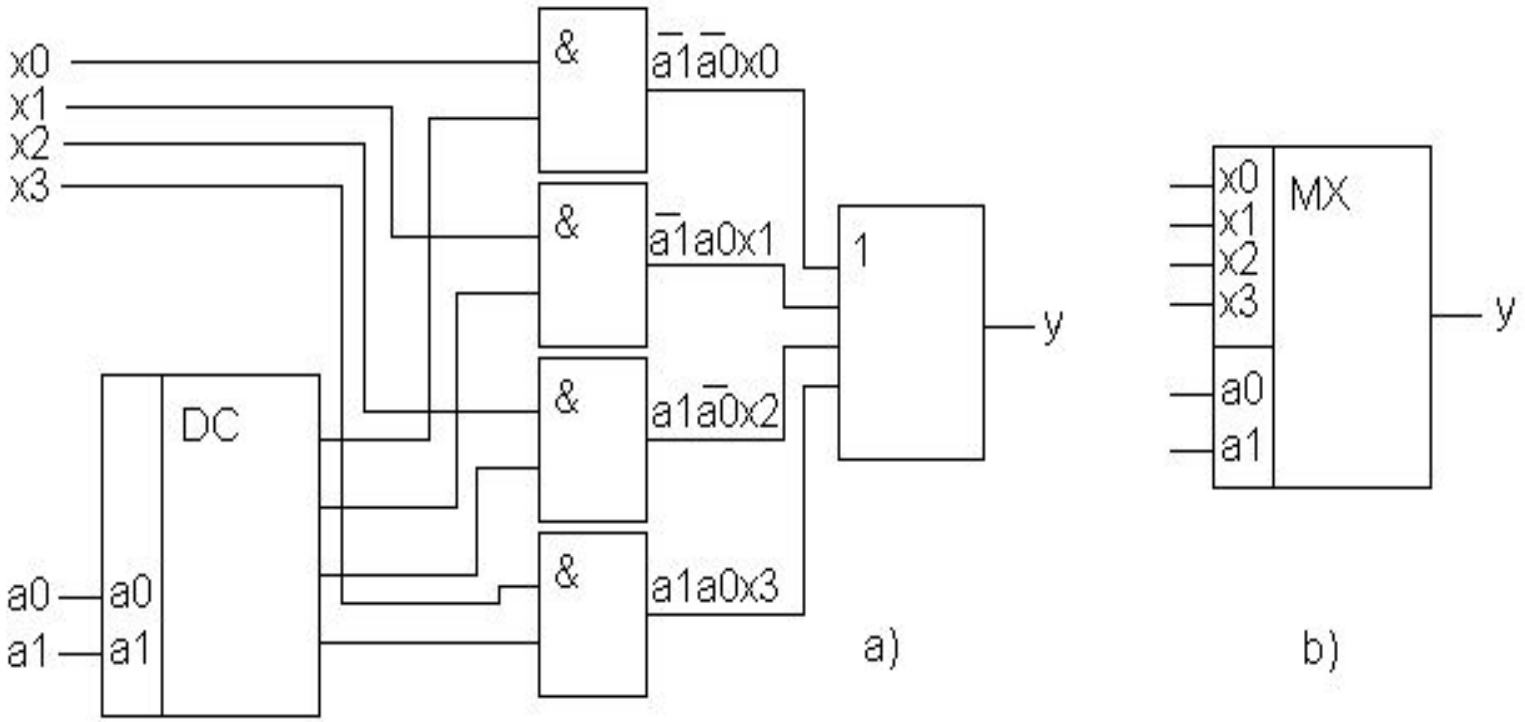
Таблица истинности для дешифратора трехразрядного двоичного кода десятичных цифр:

Входы			Выходы (Y)
X_2	X_1	X_0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Дешифратор на три входа

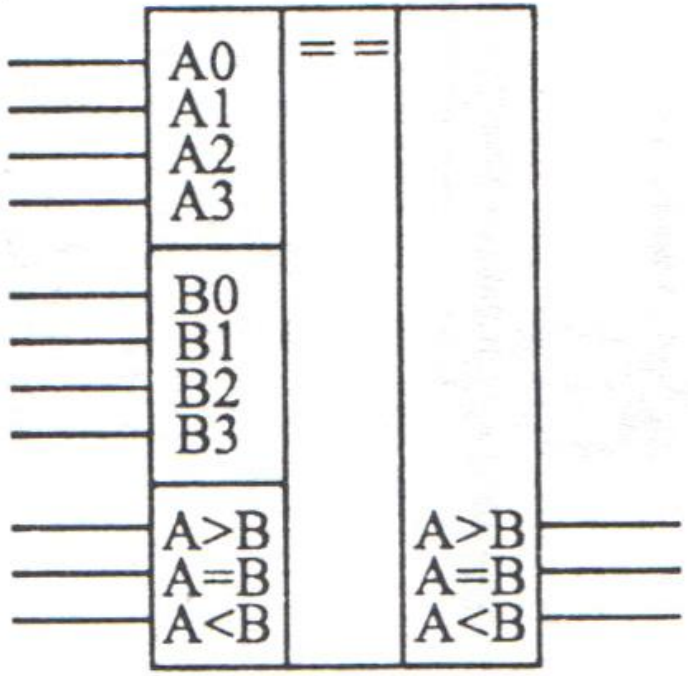


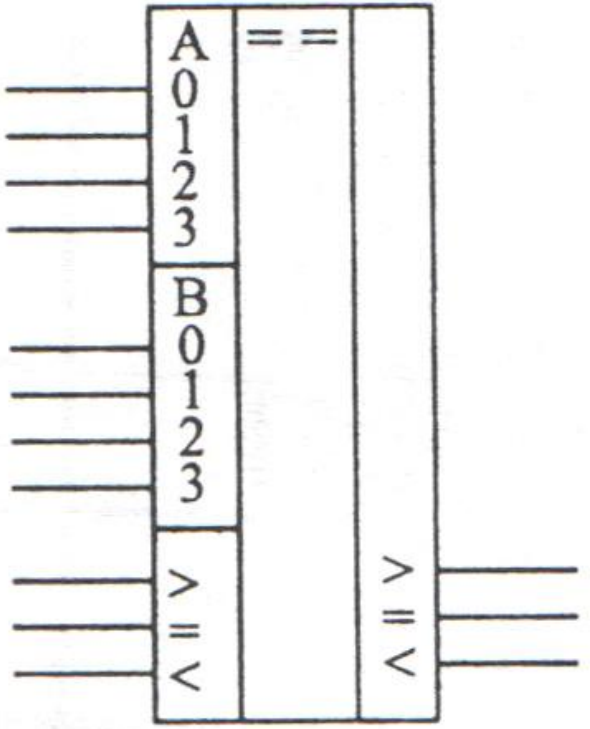
Цифровой мультиплексор



- Пропускает(коммутирует) сигнал с одного из входов на один выход в зависимости от состояния двоичного кода на адресных входах.

Цифровой компаратор





- Сравнивает два двоичных числа
- http://naf-st.ru/articles/digit/sum/

Сложе́ние по мо́дулю 2 (исключа́ющее «ИЛИ»)

Сумматор по модулю «2» вырабатывает на своем входе сигнал логической единицы, если количество его входов с сигналом логической единицы нечетное.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Одноразрядный двоичный сумматор

PI	A	B	S	PO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

PI – перенос из предыдущего разряда

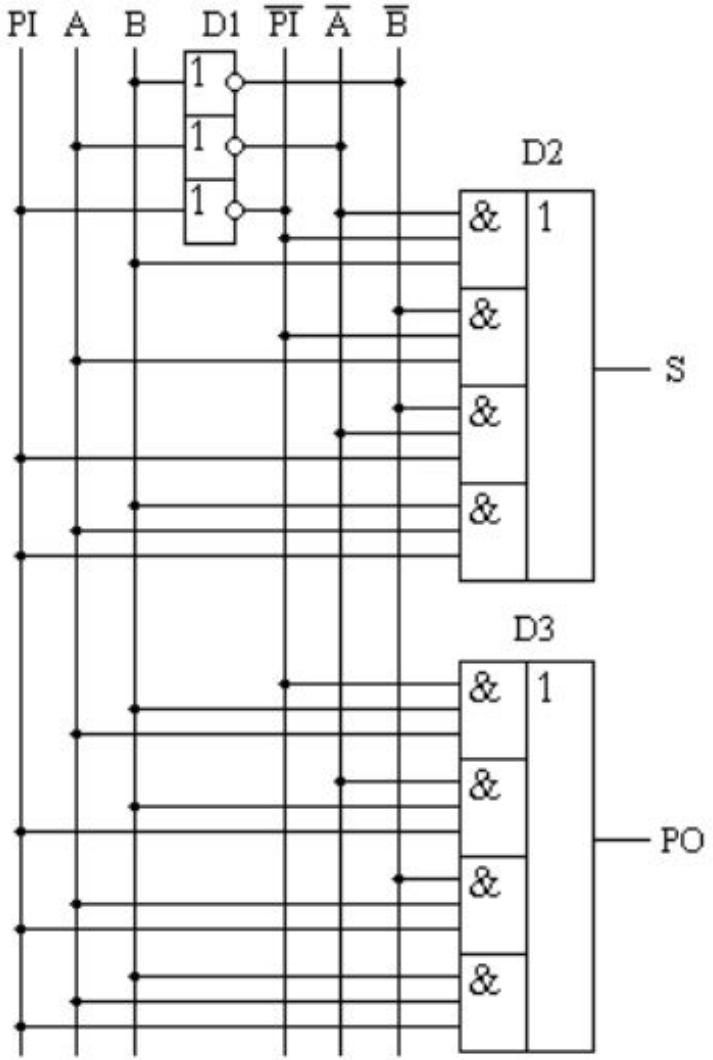
A – бит первого числа

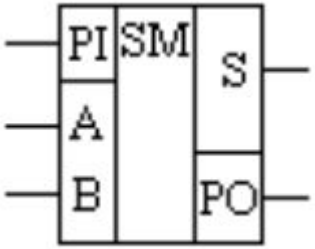
B- бит второго числа

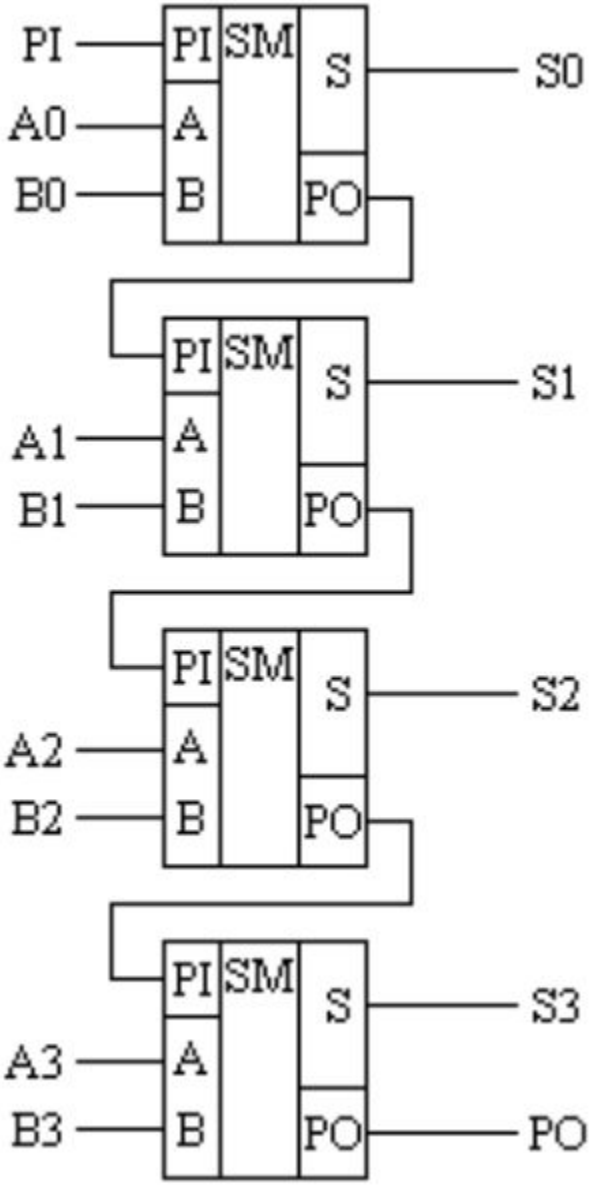
S - сумма

P0 – перенос в следующий разряд

Сумматор

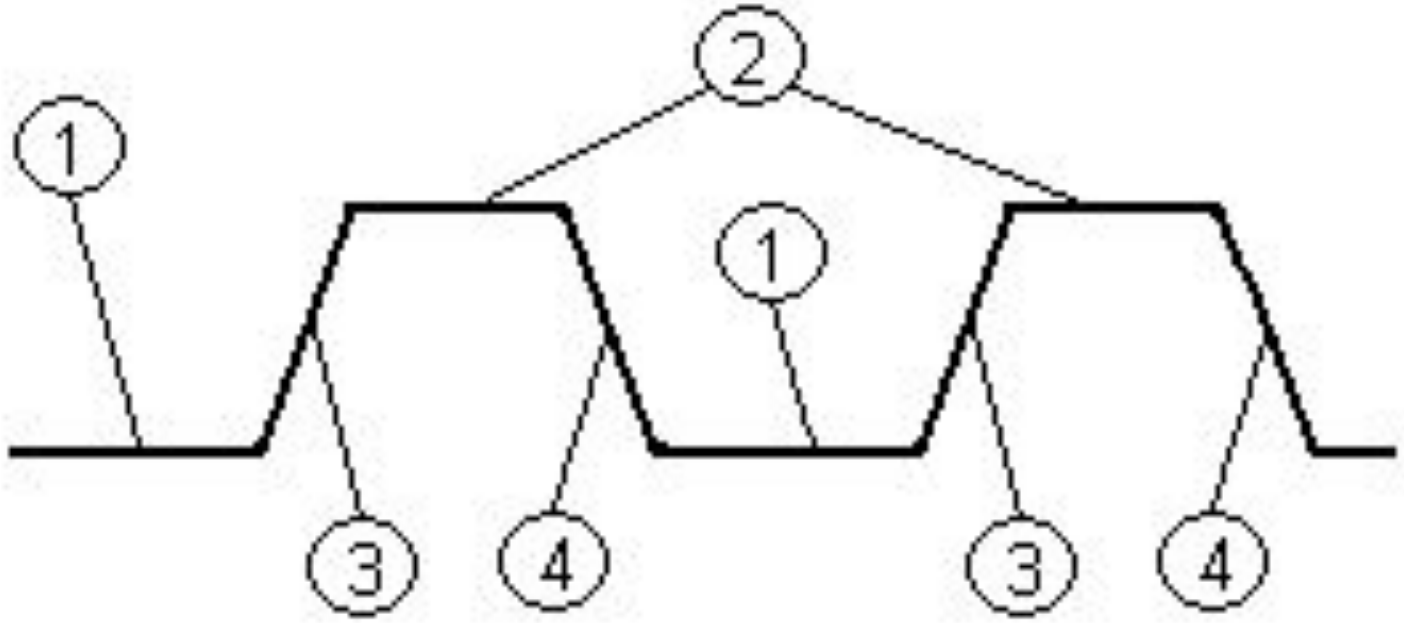

PI
A
B
D1
$\overline{PI}$
$\overline{A}$
$\overline{B}$
1
1
1
D2
&
&
&
&
1
S
D3
&
&
&
&
1
PO


PI
SM
S
A
B
PO


PI
SM
S
A
B
PO
PI
A0
B0
S0
A1
B1
S1
A2
B2
S2
A3
B3
S3
PO

Функциональные узлы ЭВМ последовательного типа (элементы с памятью)

Составляющие цифрового сигнала



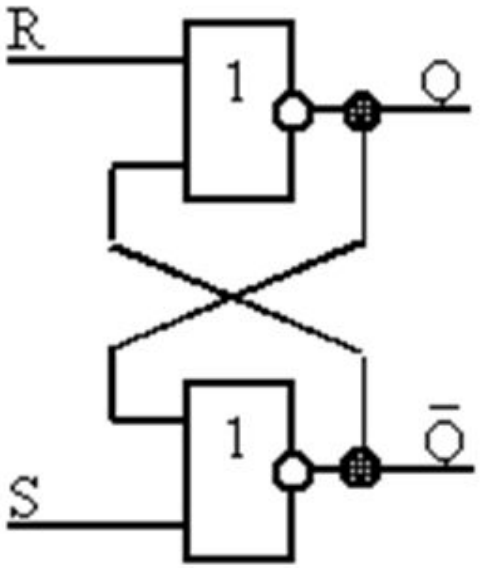
1 — низкий уровень сигнала ≃ 0,1Uпит(0,5В при Uпит = 5В);

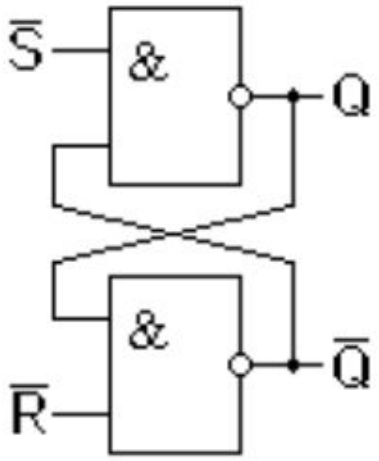
2 — высокий уровень сигнала ≃ (0,5-0,9) Uпит - (2,5-4,5)В при Uпит = 5В;

3 — нарастание сигнала (передний фронт);

4 — спад сигнала (задний фронт);

Асинхронный RS - триггер



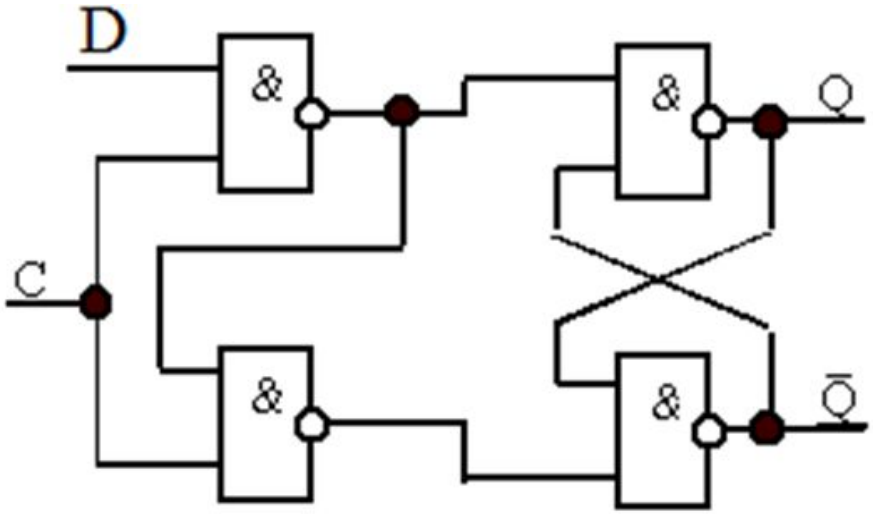


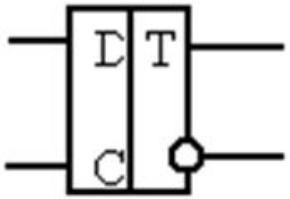
R	S	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	1	0
1	0	0	1
1	1	зап	зап

$\bar{R}$	$\bar{S}$	Q	$\bar{Q}$
0	0	зап	зап
0	1	0	1
1	0	1	0
1	1	Q	$\bar{Q}$

***RS-триггер* - устройство с двумя устойчивыми состояниями, имеющее два информационных входа R и S.**

Синхронный D-триггер



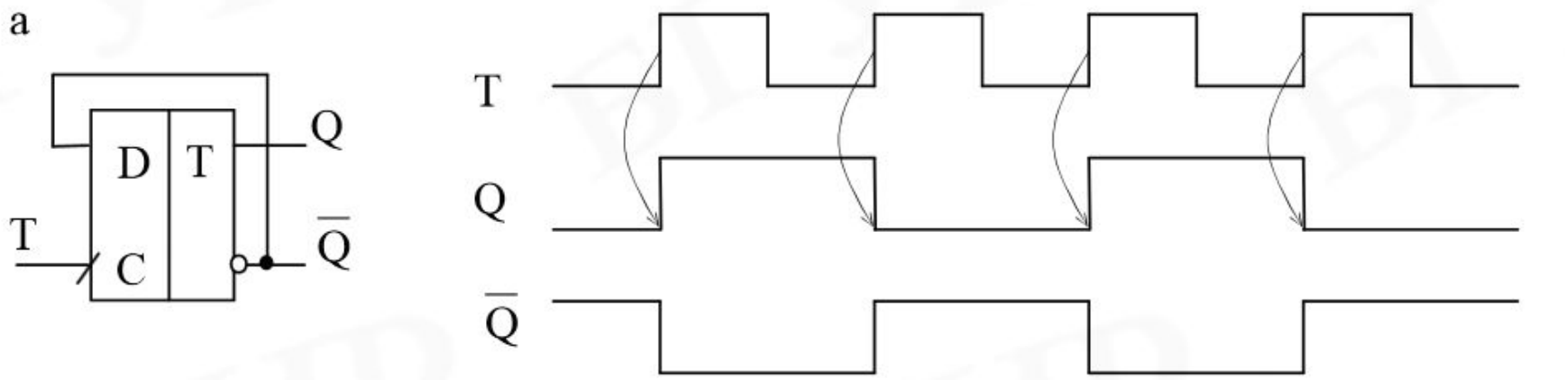


C D Q	Q^+	Режим работы
0 0 0 0 0 1 0 1 0 0 1 1	0 1 0 1	Хранение информации
1 0 0 1 0 1	0 0	Перезапись 0
1 1 0 1 1 1	1 1	Перезапись 1

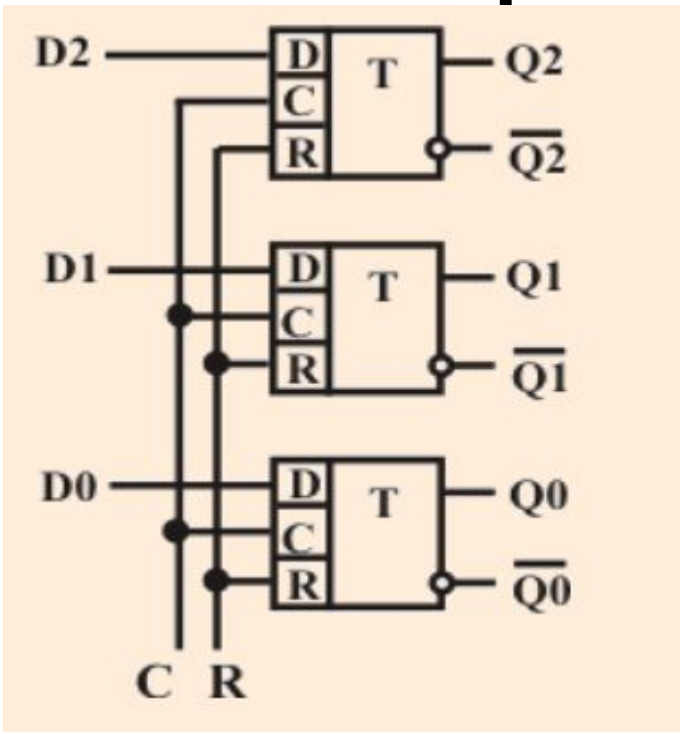
- **Когда на вход C подан логический 0, триггер хранит информацию. Если на вход C подать логическую 1, то триггер записывает значение с информационного входа D.**

Т- триггеры

Т – триггеры работают в счетном режиме и меняют свое состояние на противоположное на каждом периоде тактового сигнала



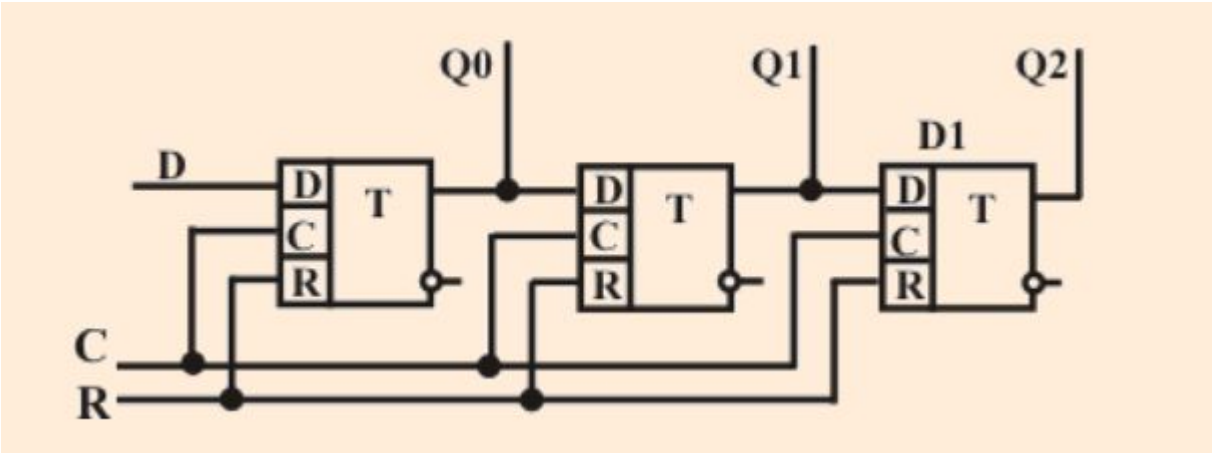
Регистр



Каждый D - триггер служит для хранения одного разряда числа. Вход R служит для установки триггеров в нулевое состояние перед записью информации. Входное двоичное число подается на входы D0-D2 и при подаче импульса на вход C записывается в триггеры

Информация может храниться сколь угодно долго, если на вход C не поступают импульсы (или если не выключается питание)

Регистр сдвига



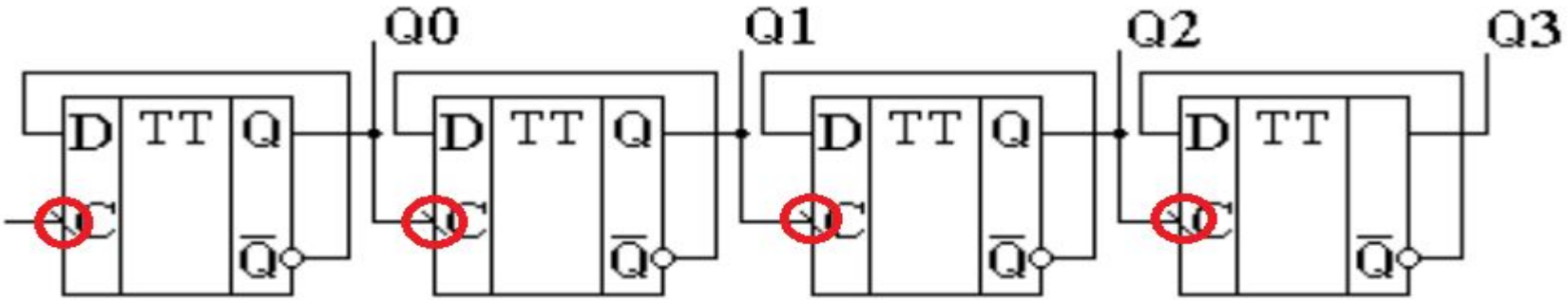
Каждый выход триггера соединен с входом следующего. Перед записью информации регистр устанавливается в нулевое состояние. Информация подается на D-вход первого триггера. При подаче импульса на вход С бит информации на D – входе записывается в первый триггер. При подаче следующего импульса этот бит перезаписывается в следующий триггер. При этом в первый триггер записывается следующий бит информации и т. д.

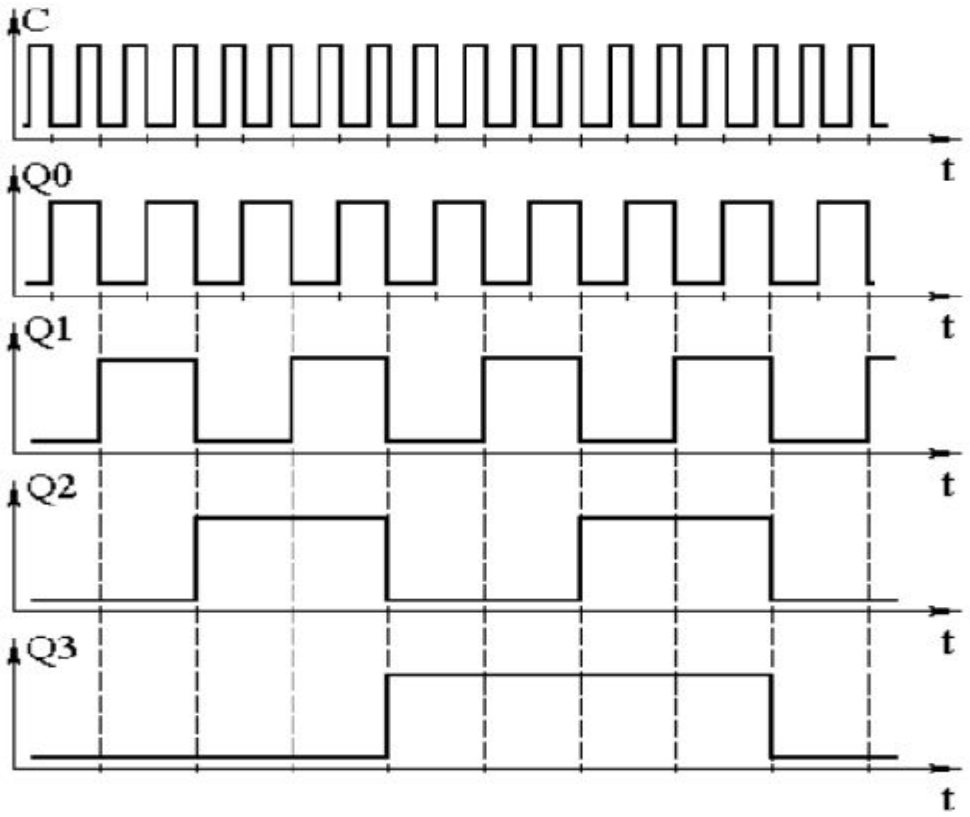
Информация продвигается по регистру от первого триггера к последнему.

Цифровой счетчик

- **Цифровой счетчик импульсов** - это цифровой узел, который осуществляет счет поступающих на его вход импульсов. Результат счета формируется счетчиком в заданном коде и может храниться требуемое время. Счетчики строятся на триггерах, при этом количество импульсов, которое может подсчитать счетчик определяется из выражения
- $N = 2^n - 1$, где **n** - число триггеров,

Четырехразрядный двоичный счетчик





Цифровой счетчик

- Используются следующие разновидности счетчика:
- - счетчики прямого счета;
- - счетчики обратного счета;
- - реверсивные счётчики.