

# ЦИФРОВЫЕ УСТРОЙСТВА

Элементная база вычислительной техники стремительно развивается благодаря успехам полупроводниковой технологии. Есть микросхемы, содержащие более миллиарда компонентов типа транзистора на одном кристалле. Общие принципы построения цифровых схем остаются неизменными и основаны на булевой алгебре и применении стандартных функциональных узлов.

Элементом цифрового устройства называют наименьшую функциональную часть, на которые можно разделить цифровое устройство при его проектировании и анализе. Набор элементов, объединенных общей конструкцией, технологией, способом представления информации, организацией связей образует **систему элементов**. Система должна быть функционально полной, чтобы реализовать любые логические функции путем суперпозиции простейших функций, предусмотренных системой.

# ЦИФРОВЫЕ УСТРОЙСТВА

Элементы малой и средней интеграции содержат в едином корпусе наборы однотипных элементов с отдельными входами и выходами. Элементы высокой степени интеграции объединяют в одном корпусе наборы различных элементов, объединенных в законченные функциональные узлы.

По функциональному назначению элементы делят на **логические, запоминающие и специальные.**

**Логические элементы** выполняют логические функции и относятся к классу комбинационных схем, в которых выходной сигнал в некоторый момент времени зависит только от входных сигналов, действующих в тот же момент времени. **Запоминающие элементы** служат для хранения цифровой информации и относятся обычно к классу последовательностных схем, в которых выходные состояния определяются не только состояниями входов в данный момент времени, но и внутренним состоянием самой схемы. **Специальные элементы** обеспечивают сопряженную работу узлов и подсистем ЭВМ, они могут быть цифровыми, аналоговыми и цифроаналоговыми.

# ЦИФРОВЫЕ УСТРОЙСТВА

Цифровые устройства разделяются на два класса: **комбинационные** устройства, которые не обладают памятью, и **последовательностные** устройства, обладающие памятью. Действия комбинационных устройств однозначно определяются только текущими значениями входных сигналов. На результат работы последовательностных устройств влияние оказывают значения входных сигналов как в настоящий, так и в предыдущий моменты времени.

В цифровых устройствах обычно используются микроэлектронные компоненты, на входах и выходах которых сигналы могут принимать одно из двух возможных значений (тока или напряжения). Таким образом, цифровые устройства реализуют логические функции, которые задают связь между входными и выходными логическими сигналами (а в последовательностных устройствах – еще и внутренними состояниями).

Теоретической основой цифровых устройств, описываемых логическими функциями, является алгебра логики (булева алгебра).

# ЦИФРОВЫЕ УСТРОЙСТВА

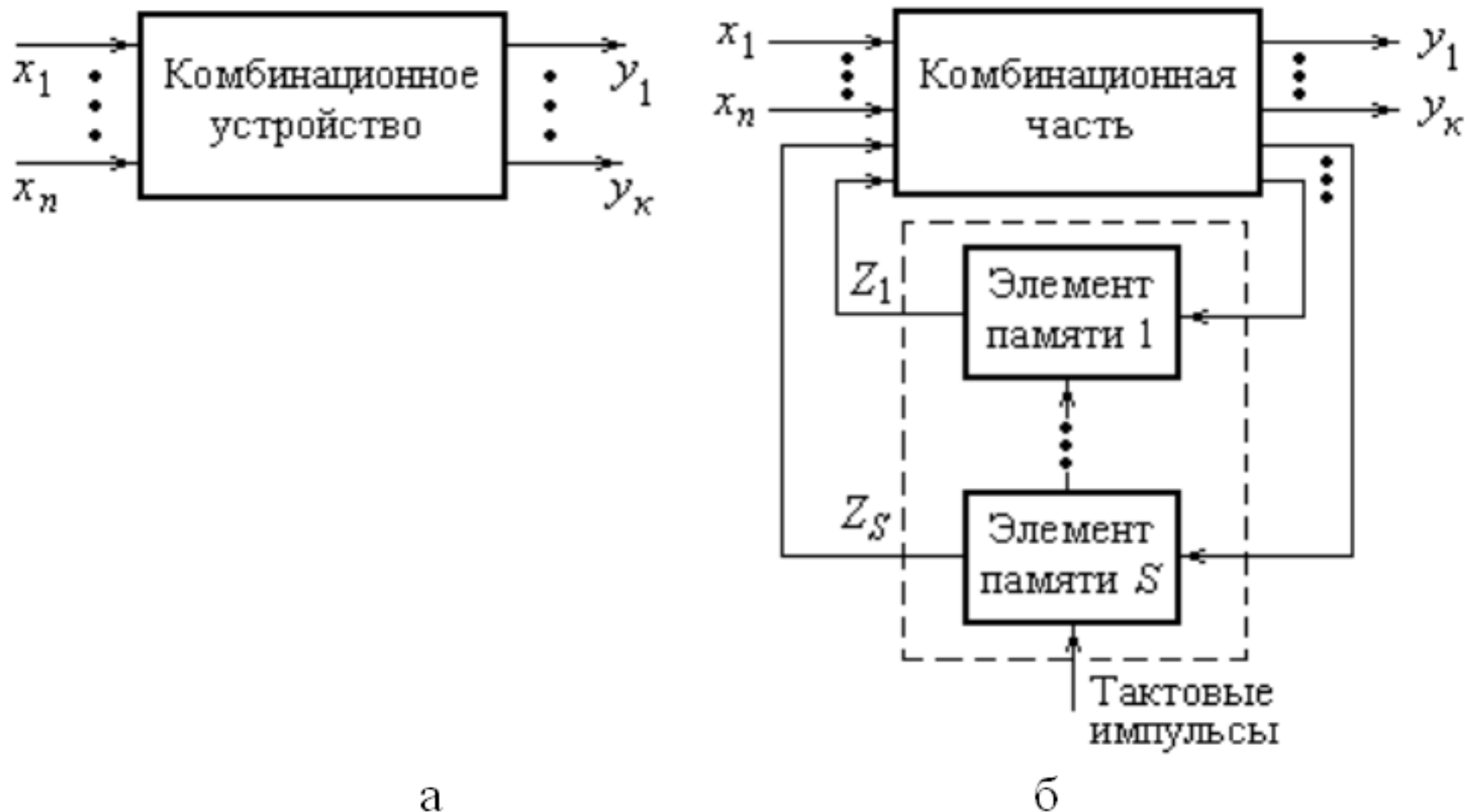


Рис. Структура а - комбинационного и б - последовательного цифровых устройств

# Логические вентили

Логическая функция может быть задана с помощью таблицы истинности или структурной формулы.








В таблице истинности для каждого возможного набора значений входных переменных указывается набор значений выходных переменных. В таблице истинности количество столбцов равно сумме числа входных  $N_{in}$  и выходных сигналов  $N_{out}$ . Количество строк (не считая заголовочной) равно  $2^{N_{in}}$ . Пример таблицы истинности для логической функции трех переменных  $y = f(x_0, x_1, x_2)$  показан в табл. 1.

*Таблица 1. Пример таблицы истинности для логической функции трех переменных*

$x_2$	$x_1$	$x_0$	$y$
0	0	0	$f(0,0,0)$
0	0	1	$f(0,0,1)$
0	1	0	$f(0,1,0)$
0	1	1	$f(0,1,1)$
1	0	0	$f(1,0,0)$
1	0	1	$f(1,0,1)$
1	1	0	$f(1,1,0)$
1	1	1	$f(1,1,1)$

При задании логической функции с помощью формулы используются различные логические операции, основными из которых являются операции логического умножения (И), сложения (ИЛИ) и отрицания (НЕ) (см. табл. 2). С помощью этих операций можно описать любую логическую функцию. Для удобства иногда используют дополнительные логические операции, такие как исключающее ИЛИ, исключающее И, а также операции И-НЕ, ИЛИ-НЕ.

Таблица 2. Виды элементарных логических операций

Название	Вид	Функция	Таблица истинности															
AND И		$A \cdot B$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$x_1$	$x_0$	$y$	0	0	0	0	1	0	1	0	0	1	1	1
$x_1$	$x_0$	$y$																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR ИЛИ		$A + B$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$x_1$	$x_0$	$y$	0	0	0	0	1	1	1	0	1	1	1	1
$x_1$	$x_0$	$y$																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT Инвертор		$\overline{A}$	<table><tr><th><math>x</math></th><th><math>y</math></th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	$x$	$y$	0	1	1	0									
$x$	$y$																	
0	1																	
1	0																	
NAND И-НЕ		$\overline{A \cdot B}$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	$x_1$	$x_0$	$y$	0	0	1	0	1	1	1	0	1	1	1	0
$x_1$	$x_0$	$y$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR ИЛИ-НЕ		$\overline{A + B}$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	$x_1$	$x_0$	$y$	0	0	1	0	1	0	1	0	0	1	1	0
$x_1$	$x_0$	$y$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR Исключающее ИЛИ (либо)		$A \oplus B$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	$x_1$	$x_0$	$y$	0	0	0	0	1	1	1	0	1	1	1	0
$x_1$	$x_0$	$y$																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR Либо-не		$\overline{A \oplus B}$ $A \odot B$	<table><tr><th><math>x_1</math></th><th><math>x_0</math></th><th><math>y</math></th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	$x_1$	$x_0$	$y$	0	0	1	0	1	0	1	0	0	1	1	1
$x_1$	$x_0$	$y$																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

# Логические вентили

При задании логической функции с помощью формулы используются различные логические операции, основными из которых являются операции логического умножения (И), сложения (ИЛИ) и отрицания (НЕ) (см. табл. 2). С помощью этих операций можно описать любую логическую функцию. Для удобства иногда используют дополнительные логические операции, такие как исключающее ИЛИ, исключающее И, а также операции И-НЕ, ИЛИ-НЕ.








Реализация этих операций выполняется электронной схемой на основе транзисторов, которая называется логическим вентилями. **Логический вентиль** – это базовый элемент цифровой схемы, выполняющий элементарную логическую операцию, преобразуя множество входных логических сигналов в выходной логический сигнал. Логика работы вентиля основана на логических (битовых) операциях над входными двоичными сигналами. Например, при реализации вентиля в ТТЛ логике используются биполярные транзисторы, у которых один, а несколько эмиттеров.

В настоящее время в современных микросхемах и цифровых устройствах доминируют логические вентили на основе полевых транзисторов, однако в прошлом для создания вентилях использовались и другие устройства, например, электромагнитные реле, гидравлические, а также механические устройства. В качестве альтернативных логических вентилях исследуются вентили на основе квантовых устройств, нейронов и молекул.

При разработке цифрового устройства логические вентили соединяют между собой, при этом выход используемого вентиля должен быть подключён к одному или к нескольким входам других вентилях.

Виды элементарных логических операций, их схематические обозначения, аналитическая запись и таблицы истинности представлены в табл. 2.

Таблица 2. Виды элементарных логических операций

Название	Вид	Функция	Таблица истинности		
AND И		$A \cdot B$	$x_1$	$x_0$	$y$
			0	0	0
			0	1	0
			1	0	0
			1	1	1
OR ИЛИ		$A + B$	$x_1$	$x_0$	$y$
			0	0	0
			0	1	1
			1	0	1
			1	1	1
NOT Инвертор		$\overline{A}$	$x$	$y$	
			0	1	
			1	0	
NAND И-НЕ		$\overline{A \cdot B}$	$x_1$	$x_0$	$y$
			0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR ИЛИ-НЕ		$\overline{A + B}$	$x_1$	$x_0$	$y$
			0	0	1
			0	1	0
			1	0	0
			1	1	0
XOR Исключающее ИЛИ (либо)		$A \oplus B$	$x_1$	$x_0$	$y$
			0	0	0
			0	1	1
			1	0	1
			1	1	0
XNOR Либо-не		$\overline{A \oplus B}$ $A \odot B$	$x_1$	$x_0$	$y$
			0	0	1
			0	1	0
			1	0	0
			1	1	1



# ЦИФРОВЫЕ УСТРОЙСТВА

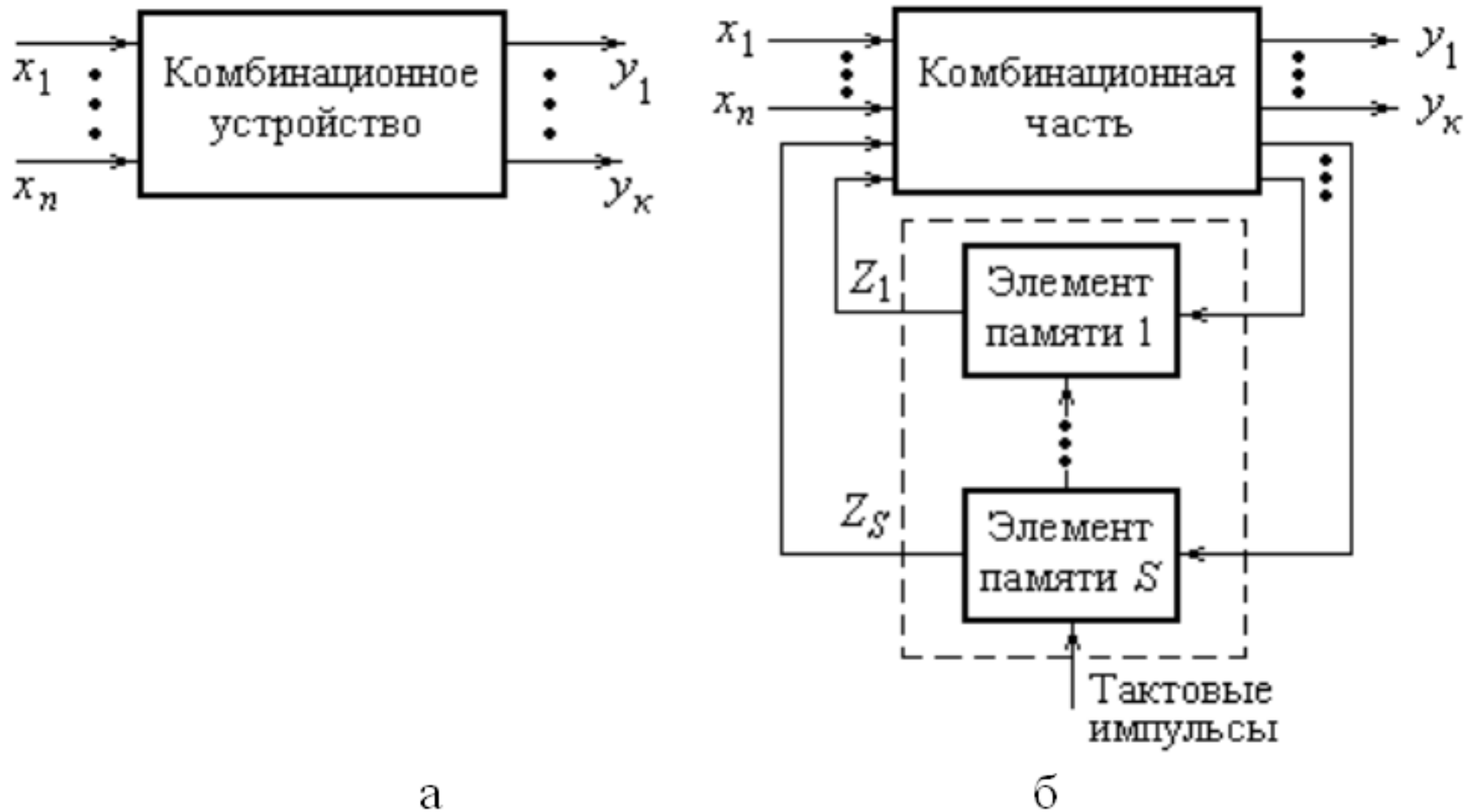


Рис. Структура а - комбинационного и б - последовательного цифровых устройств

# КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

*Комбинационные* цифровые устройства предназначены для преобразования цифровых сигналов на основе комбинационных логических схем, и, следовательно, выходные состояния таких устройств не зависят от предыстории, а однозначно определяются входными сигналами в рассматриваемые моменты времени. Другими словами комбинационные устройства не содержат элементы памяти. К основным типам комбинационных устройств относятся сумматоры, дешифраторы и шифраторы, мультиплексоры и демультимплексоры, преобразователи кодов, схемы сравнения, пороговые и мажоритарные элементы и др.

## Сумматоры

Различают полные сумматоры и полусумматоры. *Полный сумматор* предназначен для сложения трех одnorазрядных чисел  $A$ ,  $B$  и  $C$ , где в качестве числа  $C$  выступает перенос из предыдущего разряда  $P$ . В результате выполнения операции сложения в каждом разряде, кроме суммы, может возникнуть перенос в старший разряд  $P$ .

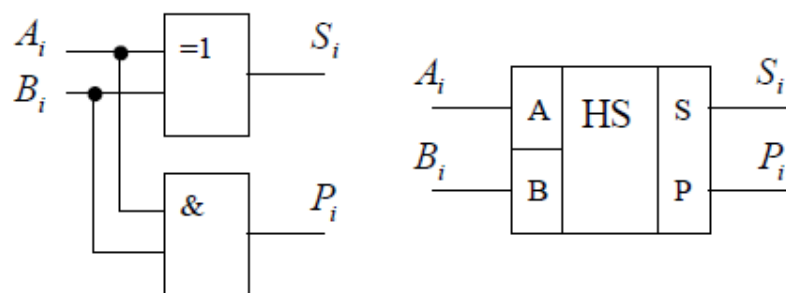
*Полусумматор* при выполнении операции сложения не учитывает переноса из предыдущего разряда.

Таблица истинности полусумматора

$A_i$	$B_i$	$S_i$	$P_i$
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

$$S_i = A_i \bar{B}_i + \bar{A}_i B_i = A_i \oplus B_i;$$

$$P_i = A_i B_i.$$



Полусумматор

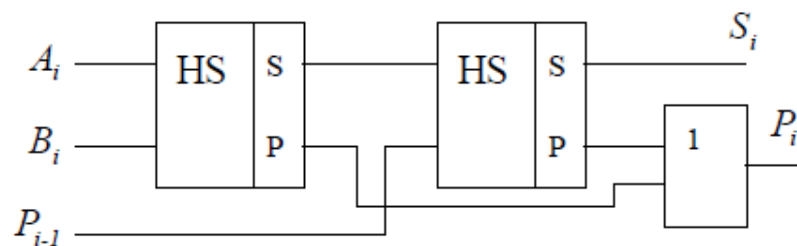


Схема полного сумматора

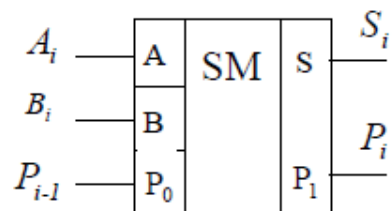


Таблица истинности полного сумматора

№	$A_i$	$B_i$	$P_{i-1}$	$S_i$	$P_i$
0	0	0	0	0	0
1	0	0	1	1	0
2	1	0	0	1	0
3	1	0	1	0	1
4	0	1	0	1	0
5	0	1	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$S_i = \bar{A}_i \bar{B}_i P_{i-1} + A_i \bar{B}_i \bar{P}_{i-1} + \bar{A}_i B_i \bar{P}_{i-1} + A_i B_i P_{i-1} = A_i \oplus B_i \oplus P_{i-1};$$

$$P_i = A_i \bar{B}_i P_{i-1} + \bar{A}_i B_i P_{i-1} + A_i B_i \bar{P}_{i-1} + A_i B_i P_{i-1} = (A_i \oplus B_i) P_{i-1} + A_i B_i.$$

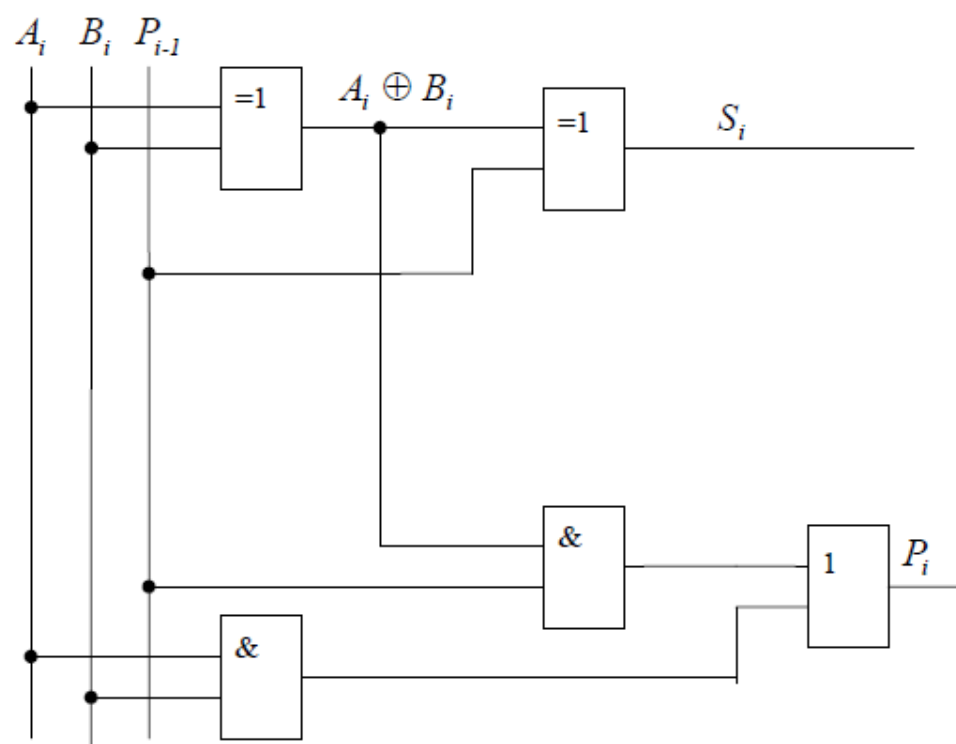
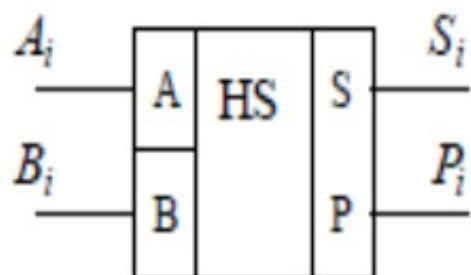
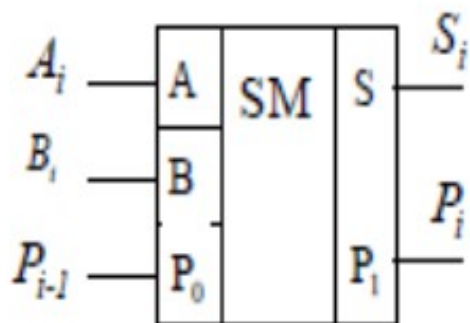


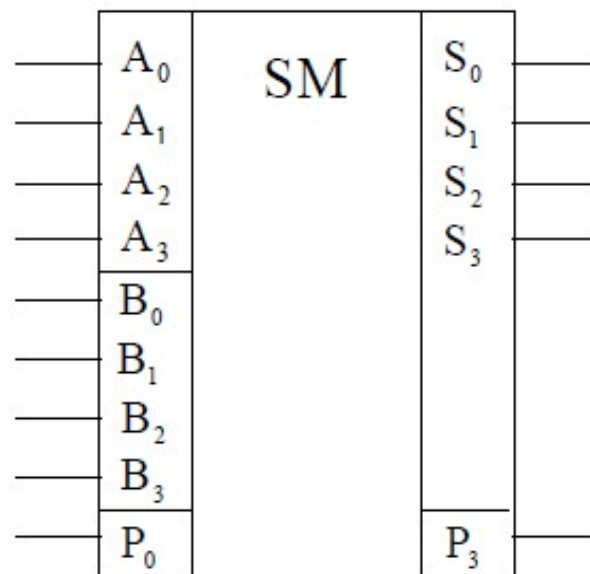
Схема полного сумматора



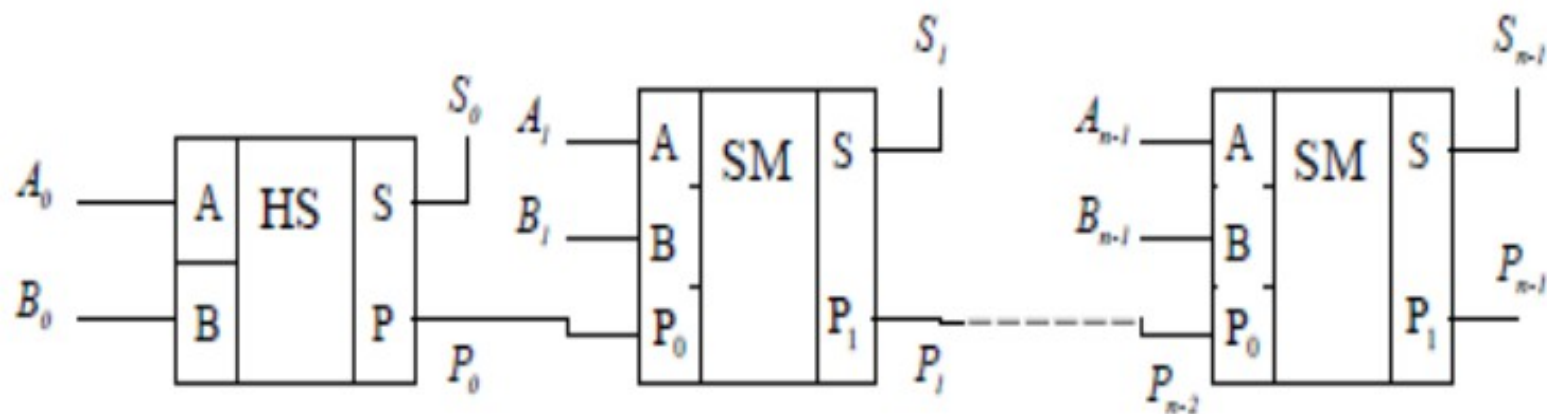
Полусумматор



полный сумматор



четырёхразрядный сумматор



Многоразрядный полный сумматор

## Дешифраторы

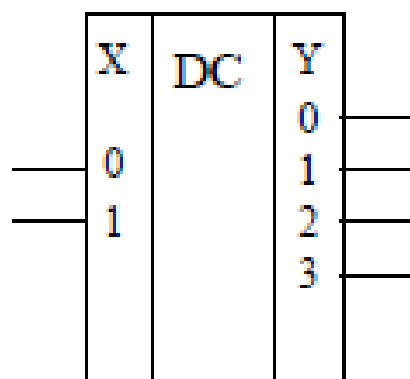
*Дешифратором* называется комбинационная схема, преобразующая двоичный  $n$ -разрядный код в унитарный  $2^n$  - разрядный код, все разряды которого, за исключением одного, равны нулю, то есть дешифратор реализует на каждом своем выходе функцию конститuenty единицы  $n$  переменных. Дешифраторы бывают полные и неполные. В полном дешифраторе выполняется условие

$$N = 2^n,$$

где  $n$  - число входов;

$N$  - число выходов.

В *неполном дешифраторе* имеется также  $n$  входов, а количество выходов  $N < 2^n$ .



условное обозначение дешифратора 2×4:

Таблица истинности дешифратора 2×4

$X_1$	$X_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

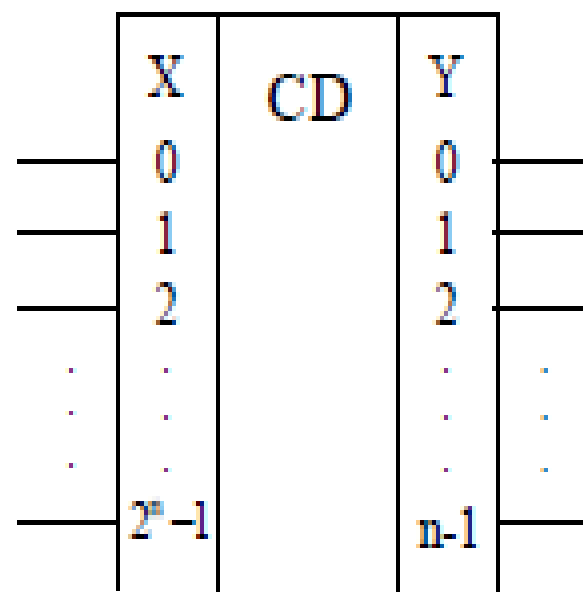
Уравнения для выходов дешифратора будут иметь вид:

$$Y_0 = \bar{X}_0 \bar{X}_1; \quad Y_1 = X_0 \bar{X}_1; \quad Y_2 = \bar{X}_0 X_1; \quad Y_3 = X_0 X_1.$$

Для расширения числа входов и выходов  
можно использовать каскадное соединение дешифраторов.

# Шифраторы

*Шифраторы* выполняют функцию обратную дешифраторам - при числе входов  $N=2^n$ , на которые подается унитарный код, устройство имеет  $n$  выходов, на которых формируется двоичный код. При этом только на одном из входов должна быть 1.



Условное обозначение  
шифратора

## Таблица истинности шифратора 4×2

Унитарный код				ДК	
$X_3$	$X_2$	$X_1$	$X_0$	$Y_1$	$Y_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Уравнения для выходов шифратора будут иметь вид:

$$Y_0 = \bar{X}_3 \bar{X}_2 X_1 \bar{X}_0 + X_3 \bar{X}_2 \bar{X}_1 \bar{X}_0 = \bar{X}_2 \bar{X}_0 (X_1 \oplus X_3);$$

$$Y_1 = \bar{X}_3 X_2 \bar{X}_1 \bar{X}_0 + X_3 \bar{X}_2 \bar{X}_1 X_0 = \bar{X}_1 \bar{X}_0 (X_2 \oplus X_3).$$



$X_1$	$X_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_0 = \bar{X}_0 \bar{X}_1; \quad Y_1 = X_0 \bar{X}_1;$$

$$Y_2 = \bar{X}_0 X_1; \quad Y_3 = X_0 X_1.$$

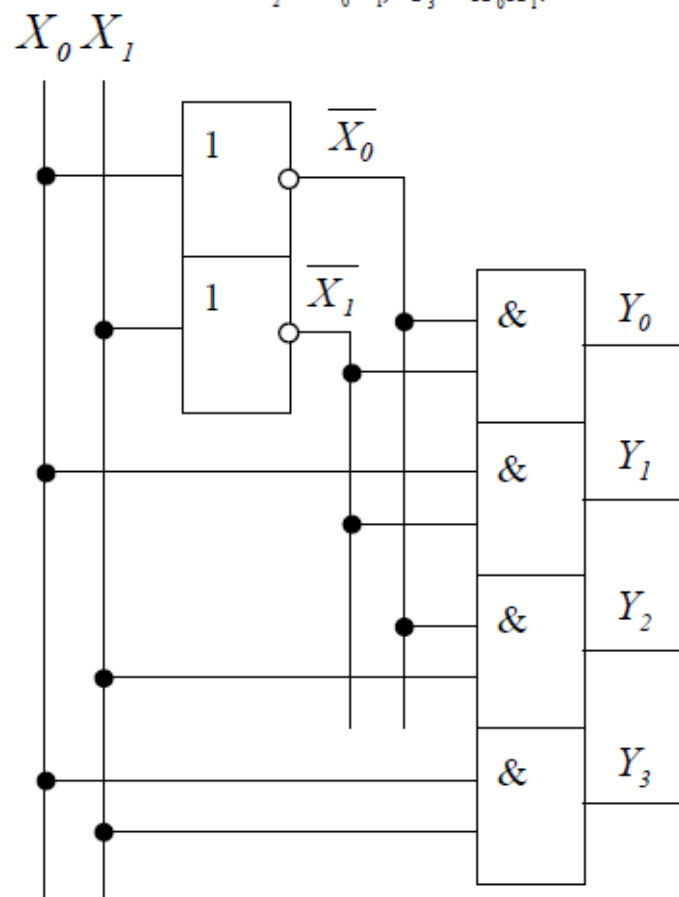


Схема дешифратора 2×4

Унитарный код				ДК	
$X_3$	$X_2$	$X_1$	$X_0$	$Y_1$	$Y_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Y_0 = \bar{X}_3 \bar{X}_2 X_1 \bar{X}_0 + X_3 \bar{X}_2 \bar{X}_1 \bar{X}_0 = \bar{X}_2 \bar{X}_0 (X_1 \oplus X_3),$$

$$Y_1 = \bar{X}_3 X_2 \bar{X}_1 \bar{X}_0 + X_3 \bar{X}_2 \bar{X}_1 X_0 = \bar{X}_1 \bar{X}_0 (X_2 \oplus X_3).$$

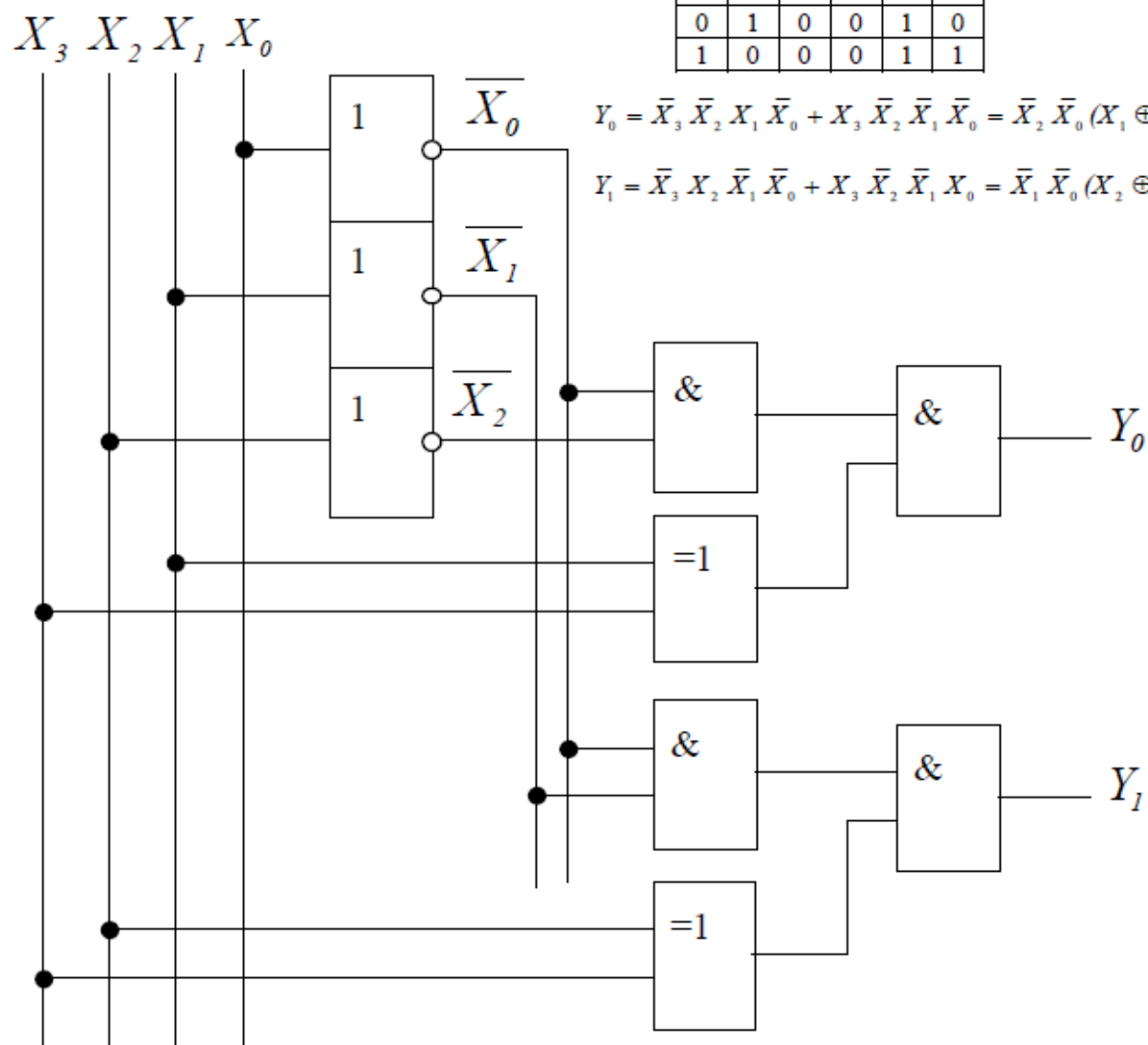


Схема шифратора 4×2

Шифраторы используются для сжатия информации при ее передаче, т.к. количество выходных линий у него меньше количества входных линий  $n < N$ . Для обратного преобразования на другом конце передачи используются дешифраторы. В цифровых системах это позволяет использовать для связи между различными устройствами ограниченное число линий связи. Другой широкой областью применения шифраторов является кодирование алфавитно-цифровой информации при ее вводе с кнопочных пультов или клавиатуры. Так, например, при нажатии одной из клавиш на одном из входов шифратора появляется «1», а на остальных входах должны быть нули. Этот входной унитарный код преобразуется в двоичный код, который обрабатывается цифровыми устройствами. Однако при этом возможны случаи одновременного нажатия нескольких клавиш, что может вызвать некорректную работу шифратора. Для исключения таких ситуаций каждому из входов назначают свой приоритет. Обычно, чем выше номер входа, тем выше его приоритет. Например, при ошибочной входной комбинации 0011 шифратор будет работать с комбинацией 0010. Такие шифраторы называются приоритетными шифраторами.

## ШИФРАТОРЫ И ДЕШИФРАТОРЫ

Итак, основным назначением шифраторов и дешифраторов является изменение кода, в котором представлено входное число. Дешифраторы позволяют обнаружить заданные комбинации сигналов на своем входе и запустить на выполнение другие компоненты системы, а шифраторы, в свою очередь, на основании номера активного входа генерируют заданные комбинации на выходе.

Некоторые дешифраторы преобразуют  $N$ -разрядный не двоичный, а троичный или  $m$ -ичный код в  $m^N$ -ичный одноединичный (унитарный) код, где  $m$  – основание системы счисления. Логический сигнал активен на том выходе, порядковый номер которого соответствует не двоичному, а троичному или  $m$ -ичному коду. Есть также шифраторы, которые выполняют преобразование позиционного  $n$ -разрядного кода в  $N$ -ичный не двоичный, а троичный или  $m$ -ичный код.

Для корректной работы шифратора только на один его вход можно подавать сигнал (единицу), при этом на всех остальных входах сигнала быть не должно (нули). **Приоритетный шифратор** применяется в случае, когда более, чем на одном входе может присутствовать активный сигнал. В таких случаях приоритетный шифратор обрабатывает (шифрует) первый вход, на котором будет присутствовать единица.

# ПРИОРИТЕТНЫЙ ШИФРАТОР

Интересно, что с помощью приоритетного шифратора можно очень просто приблизительно вычислять логарифм по основанию 2 целого числа. Логарифм по основанию 2 числа  $X$  можно представить в следующем виде (используется экспоненциальная запись с основанием 2):

$$X = 2^p M,$$

$$\log_2 X = \log_2(2^p M) = p + \log_2 M \approx p,$$

где  $p$  – порядок числа  $X$ , а  $M$  – мантисса числа  $x$ ,  $1 \leq M < 2$ . Формулу (1) можно реализовать с помощью приоритетного шифратора. В таблице ниже для примера представлена реализация логарифма по основанию 2 чисел от 0 до 15. Красным цветом выделены единицы, по которым срабатывает приоритетный шифратор. На выходе такого шифратора генерируется результат: необходимое значение порядка числа.

# ПРИОРИТЕТНЫЙ ШИФРАТОР

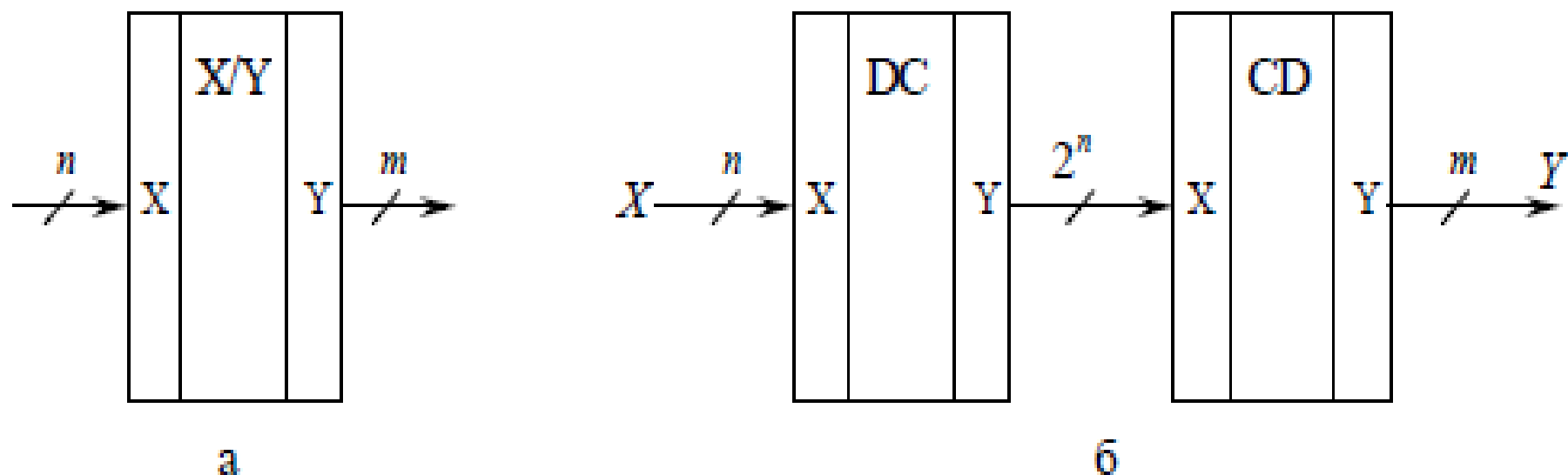
*Реализация логарифма по основанию 2*

*на основе приоритетного шифратора для чисел от 0 до 15*

Десятичный код	Прямой двоичный код				$P$
	$x_3$	$x_2$	$x_1$	$x_0$	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	2
5	0	1	0	1	2
6	0	1	1	0	2
7	0	1	1	1	2
8	1	0	0	0	3
9	1	0	0	1	3
10	1	0	1	0	3
11	1	0	1	1	3
12	1	1	0	0	3
13	1	1	0	1	3
14	1	1	1	0	3
15	1	1	1	1	3

## Преобразователи кодов

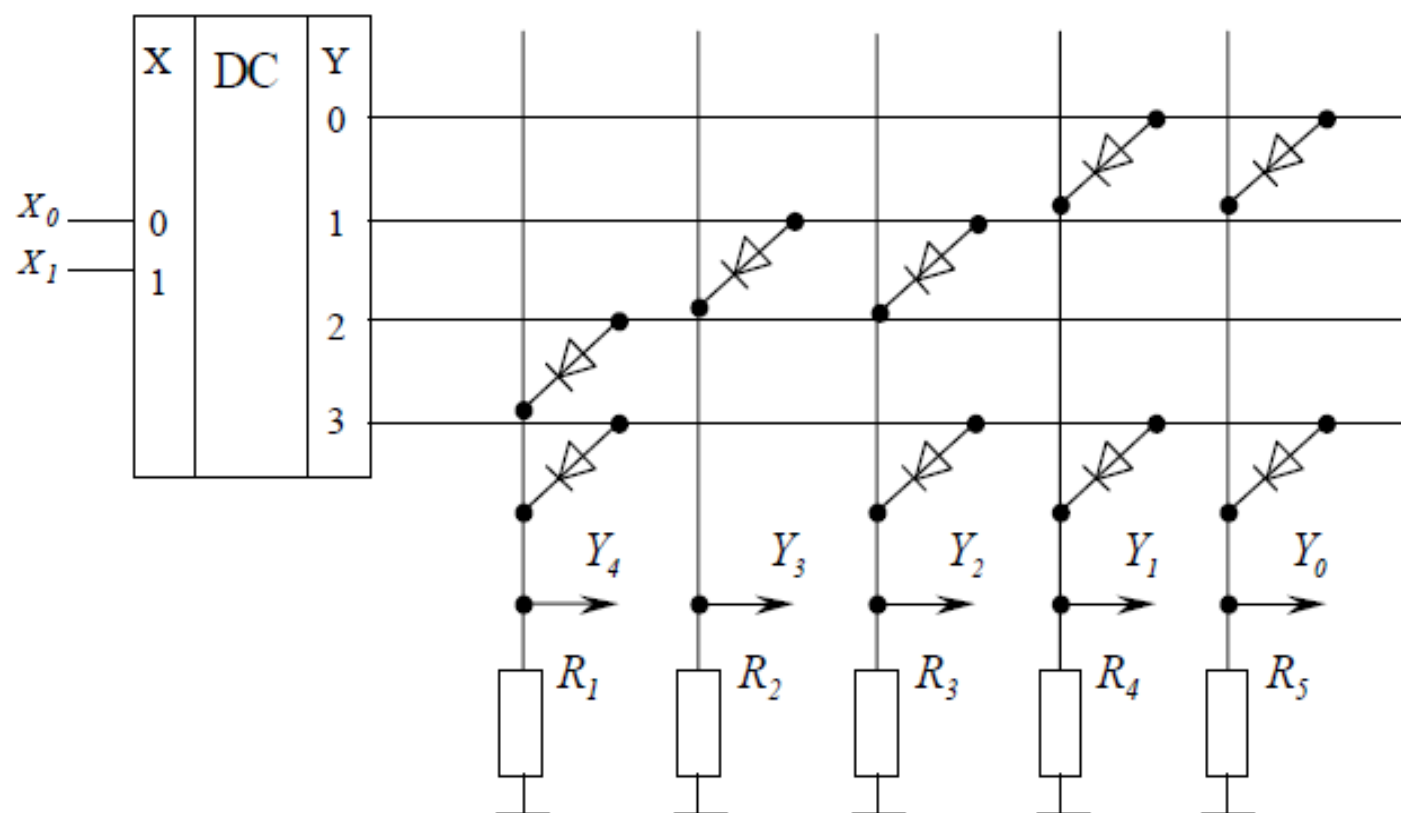
К преобразователям кодов относят широкий класс комбинационных устройств для преобразования  $n$ -разрядного кода в  $m$ -разрядный. При этом соотношения между  $n$  и  $m$  могут быть различными. С этой точки зрения дешифраторы ( $n < m$ ) и шифраторы ( $n > m$ ) тоже являются преобразователями кодов.



Условное обозначение и схема построения преобразователей кода

Таблица преобразования преобразователя кода

$X_1$	$X_0$	$Y_4$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1	1
0	1	0	1	1	0	0
1	0	1	0	0	1	0
1	1	1	0	1	1	1



Пример реализации преобразователя кода

**Преобразователь кодов (кодопреобразователь)** – это комбинационное цифровое устройство, которое изменяет код представления числа. Наиболее часто используемые кодопреобразователи: двоично-десятичный код в семисегментный, двоично-десятичный код в двоичный, двоичный код в двоично-десятичный, двоичный код в код Грея, двоичный код в код Хэмминга, двоичный код в код  $m$  из  $n$ .

При использовании двоично-десятичного кода каждый разряд десятичного числа записывается в виде его четырехбитного прямого двоичного кода. Пример таблицы истинности для кодопреобразователя из прямого двоичного кода в двоично-десятичный для чисел от 0 до 15 представлен в таблице.

Десятичный код	Прямой двоичный код				Двоично-десятичный код							
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	1	0
3	0	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	0	1	0
5	0	1	0	1	0	0	0	0	0	0	1	1
6	0	1	1	0	0	0	0	0	0	0	1	0
7	0	1	1	1	0	0	0	0	0	0	1	1
8	1	0	0	0	0	0	0	0	0	1	0	0
9	1	0	0	1	0	0	0	0	0	1	0	1
10	1	0	1	0	0	0	0	0	1	0	0	0
11	1	0	1	1	0	0	0	0	1	0	0	1
12	1	1	0	0	0	0	0	0	1	0	0	1
13	1	1	0	1	0	0	0	0	1	0	1	0
14	1	1	1	0	0	0	0	0	1	0	1	1
15	1	1	1	1	0	0	0	0	1	0	1	1

*Таблица истинности  
для кодопреобразователя  
из прямого двоичного кода  
в двоично-десятичный  
для чисел от 0 до 15*



# Двоично-десятичный код

При помощи четырех бит можно закодировать шестнадцать цифр. Из них используются 10. Остальные 6 комбинаций в двоично-десятичном коде являются запрещенными. Таблица соответствия двоично-десятичного кода и десятичных цифр:

Двоично-десятичный код				Десятичный код
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Двоично-десятичный код				Дополнительный символ
1	0	1	0	* (звёздочка)
1	0	1	1	# (решётка)
1	1	0	0	+ (плюс)
1	1	0	1	- (минус)
1	1	1	0	, (десятичная запятая)
1	1	1	1	Символ гашения

Двоично-десятичный код также применяется в телефонной связи. В этом случае кроме десятичных цифр кодируются символы '\*' или '#' или любые другие. Для записи этих символов в двоично-десятичном коде используются запрещенные комбинации.

## Преобразователи кодов

В качестве примера преобразователя кодов, выпускаемых в виде ИС, можно привести схемы, обеспечивающие преобразование информации из двоичного в двоично-десятичный код. Частным случаем преобразователей кода являются шифраторы и дешифраторы.

В качестве примера рассмотрим преобразователь двоично-десятичного кода в код для семисегментных светодиодных индикаторов (рис.). На рисунке также показан фрагмент подключения одного сегмента к выходу схемы с общим эмиттером и приведены начертания первых пяти цифр.

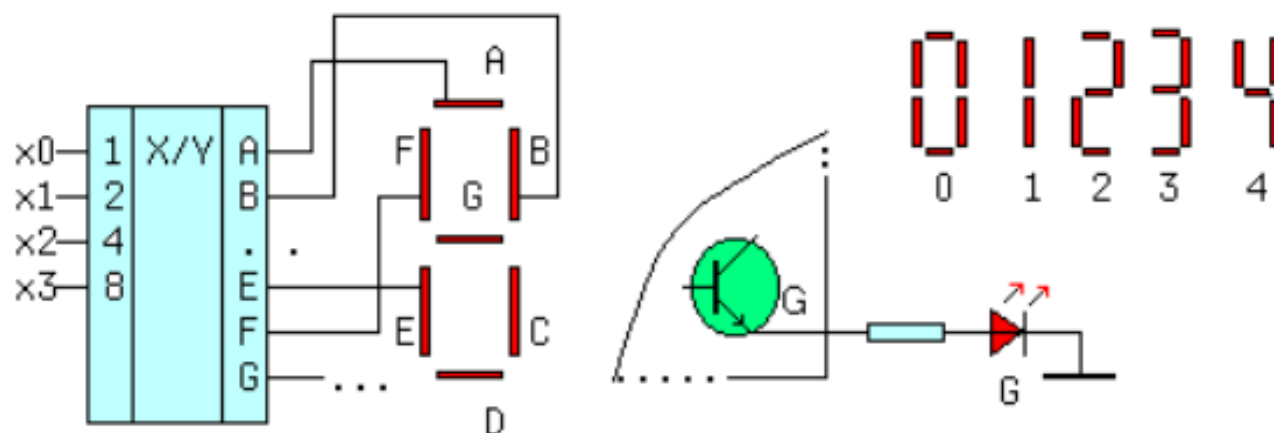


Рис.

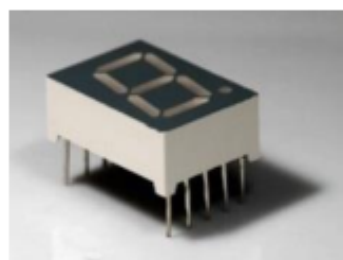
Такой преобразователь должен иметь четыре входа, т.к. для кодирования десятичных цифр от 0 до 9 достаточно четырех двоичных, и семь выходов, по одному на каждый сегмент.

# Семисегментный код

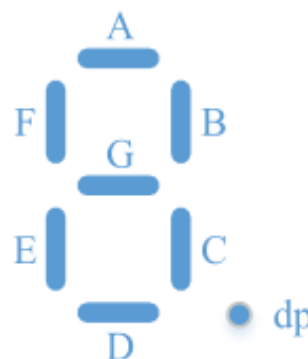
Таблица истинности для кодопреобразователя из прямого двоичного кода в семисегментный код

Двоичный код				Семисегментный код							Отображаемые цифры
$x_3$	$x_2$	$x_1$	$x_0$	$g$	$f$	$e$	$d$	$c$	$b$	$a$	
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	1	1	0	1
0	0	1	0	1	0	1	1	0	1	1	2
0	0	1	1	1	0	0	1	1	1	1	3
0	1	0	0	1	1	0	0	1	1	0	4
0	1	0	1	1	1	0	1	1	0	1	5
0	1	1	0	1	1	1	1	1	0	1	6
0	1	1	1	0	0	0	0	1	1	1	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	0	1	1	1	1	9
1	0	1	0	1	1	1	0	1	1	1	A
1	0	1	1	1	1	1	1	1	0	0	B
1	1	0	0	0	1	1	1	0	0	1	C
1	1	0	1	1	0	1	1	1	1	0	D
1	1	1	0	1	1	1	1	0	0	1	E
1	1	1	1	1	1	1	0	0	0	1	F

Одним из простейших устройств индикации (для отображения цифр и чисел) является семисегментный индикатор (обычно светодиодный) (см. рис.). Такие индикаторы, как следует из названия, состоят из семи светодиодов (иногда добавляют восьмой светодиод – для отображения точки). Зажигая различные комбинации светодиодов (см. рис.) можно отображать цифры от 0 до 9. Сегменты обозначаются буквами от A до G; восьмой сегмент (десятичная точка, decimal point, DP), предназначен для отображения десятичных дробных чисел.



*Рис. Внешний вид семисегментного индикатора с точкой*



*Рис. Схематичное обозначение сегментов*



*Рис. Пример изображения на семисегментном индикаторе цифр 2 и 9*

**Код Грея** (одношаговый) (часто слово “одношаговый” пропускают) – это такой двоичный код, в котором любые два смежных слова различаются только в одной позиции. Другими словами, расстояние Хэмминга между любыми двумя смежными словами равно единице, включая первое и последнее слова (циклически замкнутый одношаговый код Грея).

Десятичный код	Прямой двоичный код			Код Грея		
	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

*Таблица истинности для  
кодпреобразователя из прямого  
двоичного кода  
в одношаговый код Грея*

Перевод N-разрядного числа из прямого двоичного кода в двоичный код Грея выполняется по следующей формуле:  $y_i = x_i \text{ XOR } x_{i+1}$ ,  $i = 0, 1, \dots, N-1$ . Т. е. для вычисления i-го разряда в коде Грея необходимо взять исключающее ИЛИ от i-го и (i + 1)-го разрядов в прямом двоичном коде. При  $i = N-1$  считать  $x_{i+1} = 0$ . Здесь  $x_i$  – разряды исходного числа в прямом двоичном коде, а  $y_i$  – разряды числа в коде Грея.

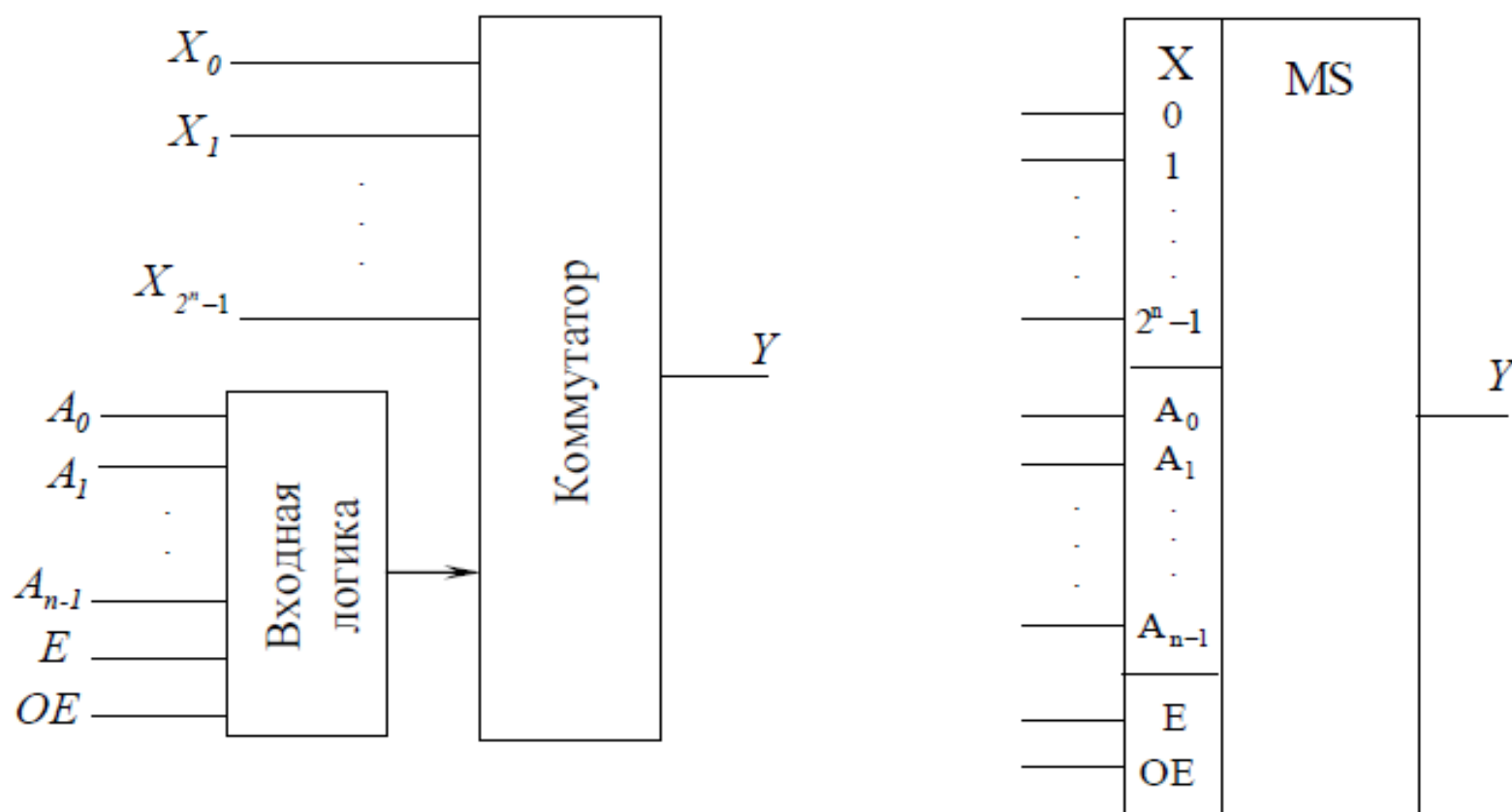
Перевод N-разрядного числа из прямого двоичного кода в двоичный код Грея выполняется по следующей формуле:  $y_i = x_i \text{ XOR } x_{i+1}$ ,  $i = 0, 1, \dots, N-1$ . Т.е. для вычисления i-го разряда в коде Грея необходимо взять исключающее ИЛИ от i-го и (i + 1)-го разрядов в прямом двоичном коде. При  $i = N-1$  считать  $x_{i+1} = 0$ . Здесь  $x_i$  – разряды исходного числа в прямом двоичном коде, а  $y_i$  – разряды числа в коде Грея.

Обратный перевод числа из кода Грея в прямой двоичный код выполняется так:  $x_i = y_i \text{ XOR } y_{i+1} \dots \text{ XOR } y_{N-1}$ . Т.е. i-й разряд в прямом двоичном коде равен исключающему ИЛИ от всех разрядов в коде Грея с номерами не менее i.

Двоичный код Грея широко применяется в качестве манипуляционного кода в различных способах цифровой модуляции. В общем случае удобство применения кода Грея в качестве манипуляционного кода, заключается в том, что при случайном (обычно, ошибочном) переходе между соседними состояниями в системе будет возникать ошибка только в одном разряде.

## Мультиплексоры

Мультиплексором называют комбинационное устройство, предназначенное для коммутации цифровых сигналов, поступающих по нескольким информационным входам, на один выход. При этом каждый информационный вход имеет свой адрес, который задается  $n$ -разрядным цифровым кодом. Тогда количество информационных входов будет равно  $N = 2^n$ . Структура мультиплексора условно обозначается записью  $(N \rightarrow 1)$ .



Обобщенная схема и условное обозначение мультиплексора

Таблица истинности демультиплексора

$A_1$	$A_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$	СКНФ
0	0	0	0	0	$X$	$\overline{A_0} \overline{A_1} XE$
0	1	0	0	$X$	0	$A_0 \overline{A_1} XE$
1	0	0	$X$	0	0	$\overline{A_0} A_1 XE$
1	1	$X$	0	0	0	$A_0 A_1 XE$

Входной сигнал  $X$  поступает на вход коммутатора и через него передается на выходы  $Y$ . Адресные сигналы  $A$  имеют то же значение, что и у мультиплексора. Сигнал стробирования  $E$  разрешает передачу входного сигнала через коммутатор.

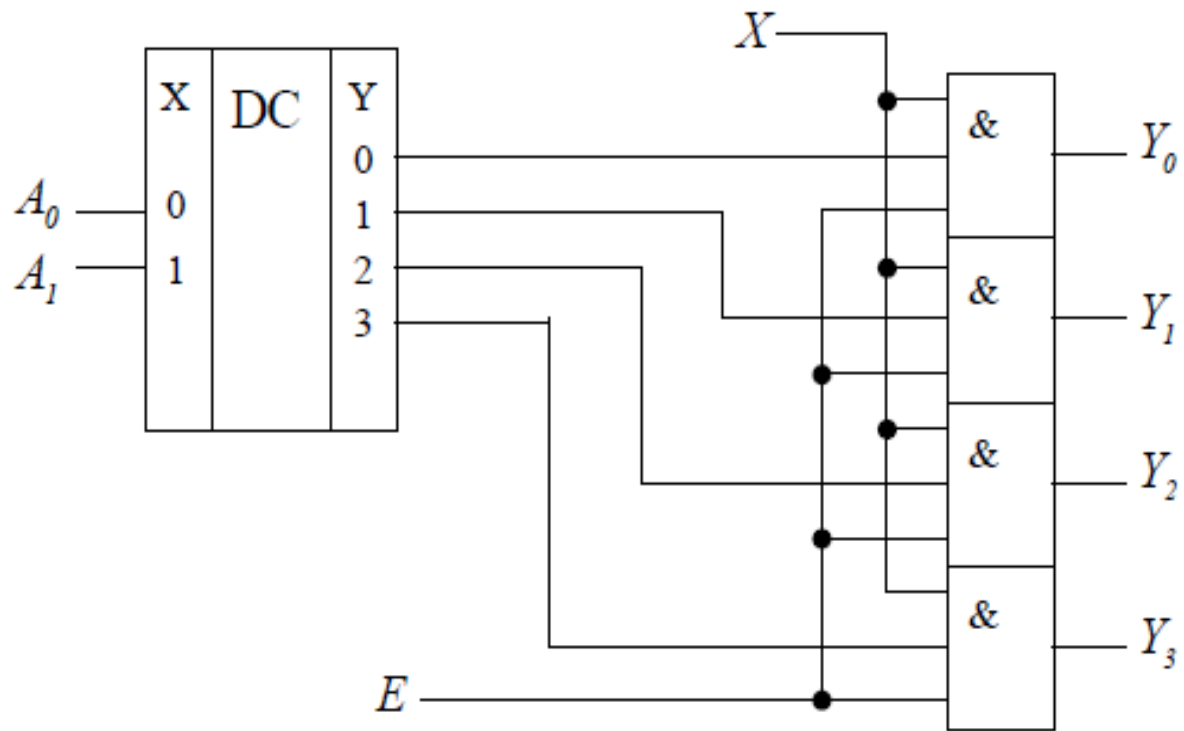


Схема демультиплексора 1 → 4

логические выражения,  
описывающие работу  
синтезируемого демультиплексора:

$$Y_0 = \overline{A_0} \overline{A_1} XE; \quad Y_1 = A_0 \overline{A_1} XE;$$

$$Y_2 = \overline{A_0} A_1 XE; \quad Y_3 = A_0 A_1 XE.$$



Демультимплексором называют функциональный узел, который обеспечивает передачу цифровой информации, поступающей по одной линии, на несколько выходных линий. Выбор выходной линии осуществляется при помощи сигналов, поступающих на адресные входы. Таким образом, демультимплексор выполняет преобразование, обратное действию мультиплексора.

Интегральные микросхемы демультимплексоров можно разделить на группы по следующим признакам:

- по числу выходов;
- по числу демультимплексоров в одном корпусе;
- по наличию стробирующего импульса Е;
- по способности передавать сигналы в двух направлениях.

Среди схем коммутации можно особо выделить схемы, которые способны пропускать сигналы в обоих направлениях. К таким элементам относятся коммутационные микросхемы, выполненные по технологии КМОП. Коммутаторы КМОП способны пропускать как аналоговые, так и цифровые сигналы, в них можно менять местами вход и выход.

# МУЛЬТИПЛЕКСОРЫ И ДЕМУЛЬТИПЛЕКСОРЫ

Мультиплексоры и демultipлексоры решают задачи направления потоков данных в программируемой логике. В зависимости от состояния управляющих сигналов мультиплексоры и демultipлексоры выбирают один из нескольких возможных путей прохождения данных.

Мультиплексоры могут использоваться для реализации логических функций многих переменных, в делителях частоты (в современных ПЛИС так делать категорически не рекомендуется), триггерных устройствах, сдвигающих устройствах и др. Кроме того, они применяются для стробирования сигналов тактирования в целях снижения энергопотребления устройства.

При проектировании цифровых устройств часто возникает задача направления потока данных от одного из множества источников к одному из множества получателей. Т.е. необходимо устройство, которое подключает (коммутирует) определенные входные порты к определенным выходным портам. Эту задачу решает **коммутатор**.

# КОММУТАТОР

При проектировании цифровых устройств часто возникает задача направления потока данных от одного из множества источников к одному из множества получателей. Т.е. необходимо устройство, которое подключает (коммутирует) определенные входные порты к определенным выходным портам. Эту задачу решает коммутатор.

Рассмотрим коммутатор, который в каждый момент времени работает только с одной парой входного и выходного портов. Такой коммутатор имеет  $N_{in}$  информационных входов,  $N_{out}$  информационных выходов,  $M_{in}$  и  $M_{out}$  адресных входов для выбора соединяемых входного и выходного информационных портов соответственно.

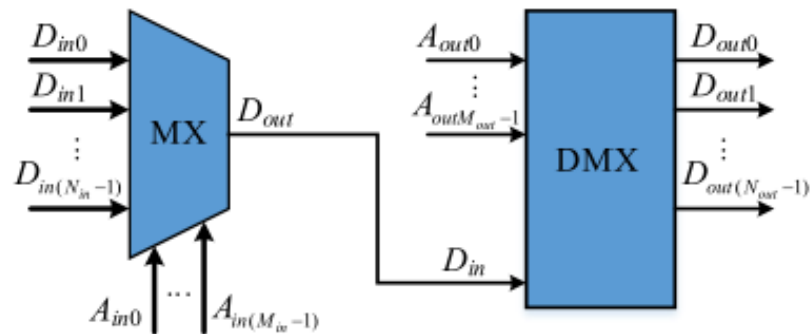


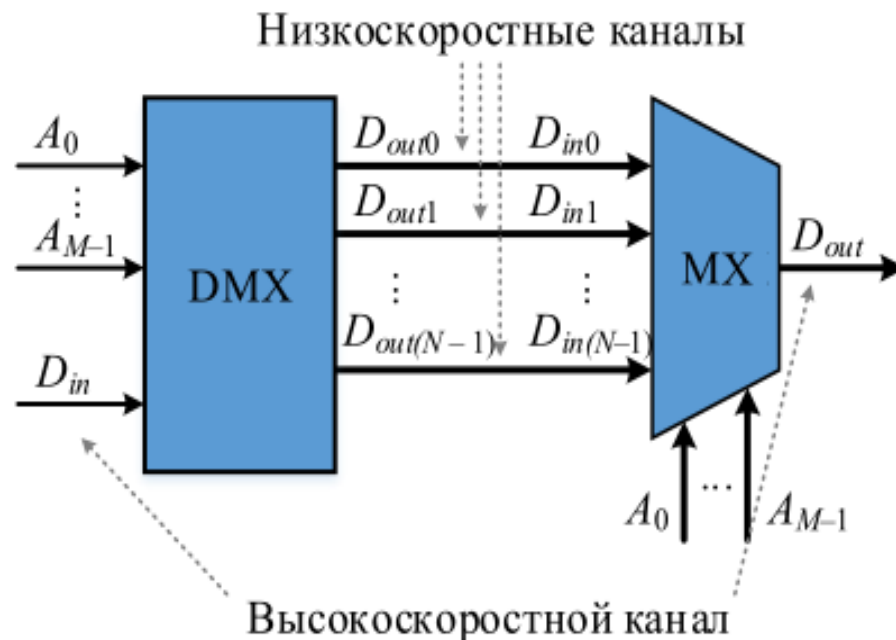
Рис. Последовательное соединение мультиплексора и демультиплексора для построения коммутатора

Таблица истинности для коммутатора 4х4.

$A_{in1}$	$A_{in0}$	$A_{out1}$	$A_{out0}$	$D_{out0}$	$D_{out1}$	$D_{out2}$	$D_{out3}$
0	0	0	0	$D_{in0}$	0	0	0
0	0	0	1	0	$D_{in0}$	0	0
0	0	1	0	0	0	$D_{in0}$	0
0	0	1	1	0	0	0	$D_{in0}$
0	1	0	0	$D_{in1}$	0	0	0
0	1	0	1	0	$D_{in1}$	0	0
0	1	1	0	0	0	$D_{in1}$	0
0	1	1	1	0	0	0	$D_{in1}$
1	0	0	0	$D_{in2}$	0	0	0
1	0	0	1	0	$D_{in2}$	0	0
1	0	1	0	0	0	$D_{in2}$	0
1	0	1	1	0	0	0	$D_{in2}$
1	1	0	0	$D_{in3}$	0	0	0
1	1	0	1	0	$D_{in3}$	0	0
1	1	1	0	0	0	$D_{in3}$	0
1	1	1	1	0	0	0	$D_{in3}$

# КОММУТАТОР

Если соединить выходы демультиплексора со входами мультиплексора (см. рис.) при  $N_{in} = N_{out}$ , то получится устройство с одним информационным входом и одним информационным выходом. Такое устройство позволяет разделять (разуплотнять) один высокоскоростной канал данных на множество низкоскоростных, что упрощает обработку данных в каждом низкоскоростном канале. После того, как данные обработаны, их можно снова объединить в один высокоскоростной канал. По такому принципу работают современные высокоскоростные последовательные трансиверы в FPGA.



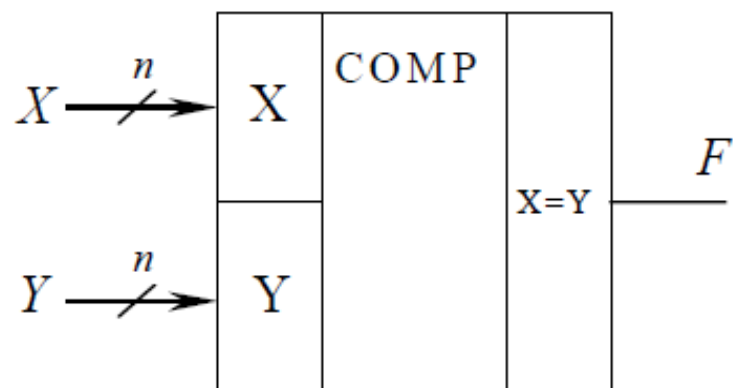
*Рис. Последовательное соединение демультиплексора и мультиплексора для разуплотнения высокоскоростного канала*

# СХЕМЫ СРАВНЕНИЯ

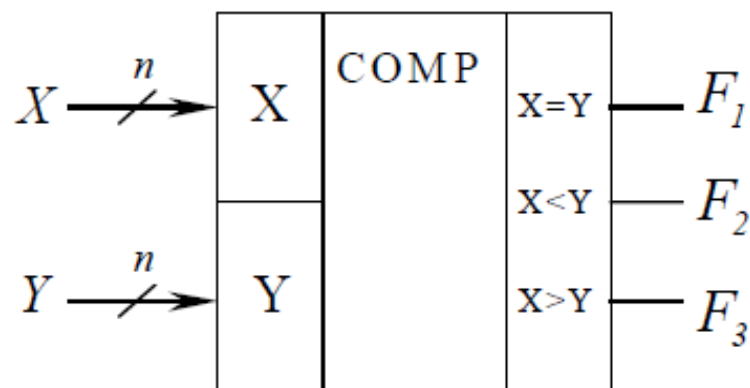
Схемы сравнения или цифровые компараторы предназначены для реализации операции сравнения двух кодов, например  $X$  и  $Y$ , и бывают двух типов:

- 1) простейшие или схемы равнозначности, выдающие сигнал  $F=1$  при  $X=Y$ ;
- 2) универсальные, выдающие три сигнала:  $F_1=1$ , если  $X=Y$ ;  
 $F_2=1$ , если  $X<Y$ ;  $F_3=1$ , если  $X>Y$ .

Условные обозначения схем сравнения приведены на рис.



а



б

Рис. Условные обозначения схем сравнения

# СХЕМЫ СРАВНЕНИЯ

Простейшие схемы сравнения строятся на базе элементов ИСКЛЮЧАЮЩЕЕ ИЛИ по схеме, приведенной на рис. (пример сравнения для 4-разрядных кодов).

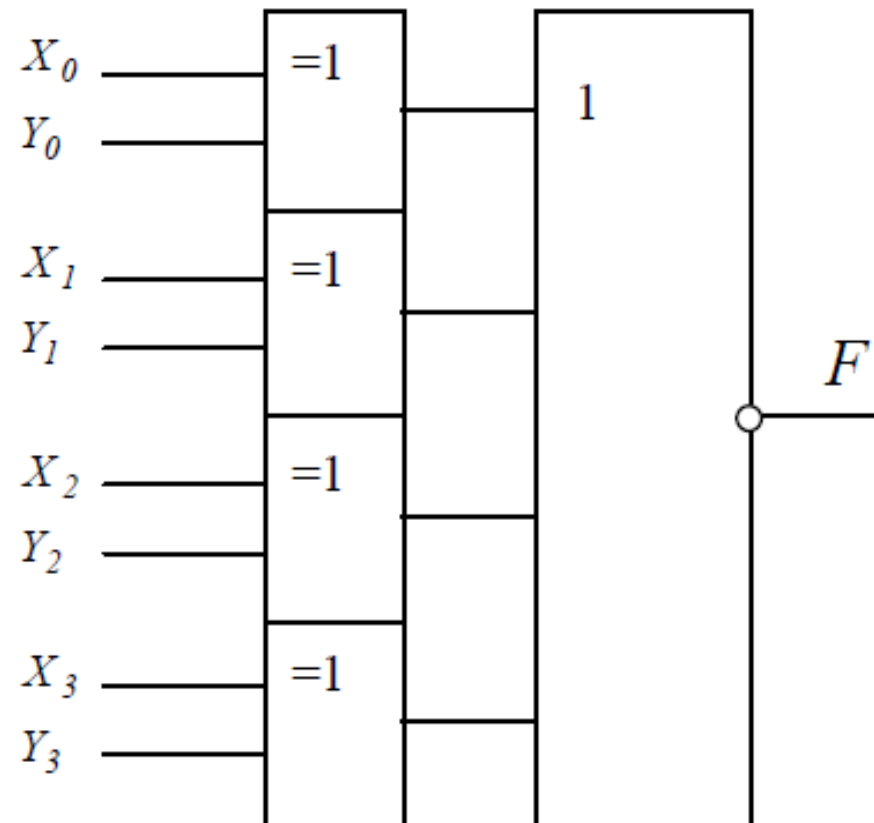
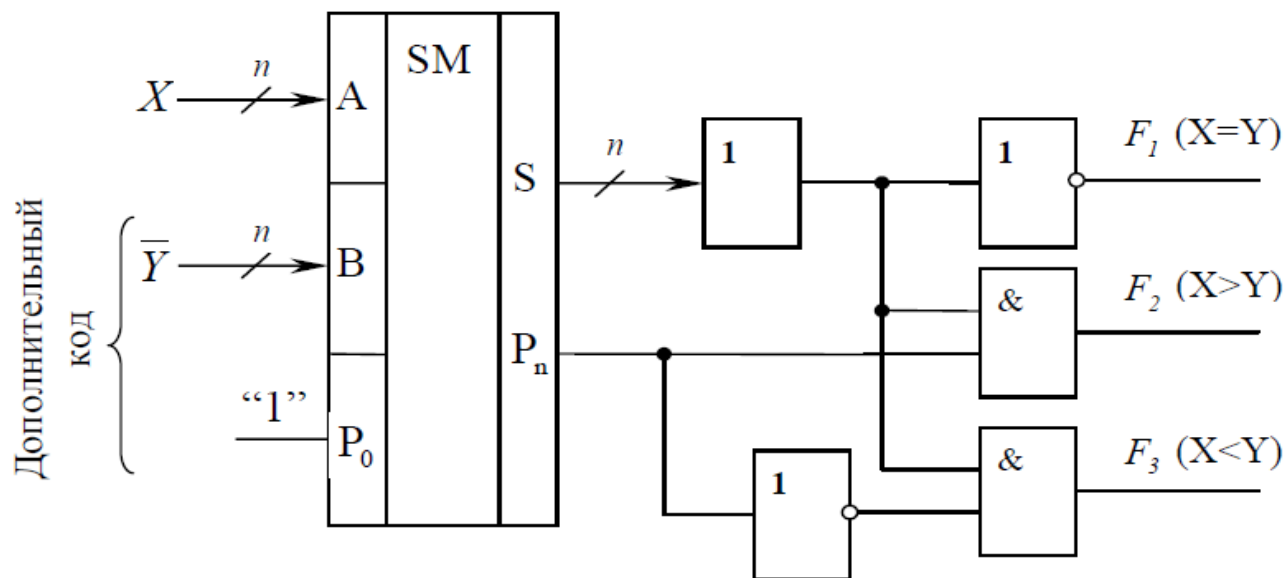


Схема равнозначности 4-разрядных кодов

# СХЕМЫ СРАВНЕНИЯ

Для построения универсальных схем сравнения используют сумматоры, на которых реализуют операцию вычитания  $X-Y$ . Для этого переменная  $Y$  подается в обратном коде и на вход переноса сумматора подается «1», что эквивалентно представлению  $Y$  в дополнительном коде. На рис. показан пример реализации схемы сравнения  $n$ -разрядных кодов.



Универсальная схема сравнения на базе сумматора

Выходная логика построена с учетом реализации следующих соотношений:

- 1) если  $X=Y$ , то  $S=0$  и  $F_1=1$ ;
- 2) если  $X<Y$ , то  $S \neq 0$ ,  $P_n=1$  и  $F_2=1$ ;
- 3) если  $X>Y$ , то  $S \neq 0$ ,  $P_n=0$  и  $F_3=1$ .

# МАЖОРИТАРНЫЕ ЭЛЕМЕНТЫ

**Мажоритарный элемент** — логический элемент, с чётным или нечётным числом входов и одним выходным сигналом, значение которого совпадает со значением, на большинстве входов. При чётном числе входов большинством считается  $n/2+1$ , соответственно  $n/2$  к большинству не относится. Таким образом, элемент работает по «принципу большинства»: если на большинстве входов будет сигнал «1», то и на выходе схемы установится сигнал «1»; и наоборот, если на большинстве входов будет сигнал «0», то и на выходе установится «0».

Мажоритарные элементы используются в цифровых высоконадёжных системах и устройствах, например, в системах резервирования, в помехоустойчивых телекоммуникационных системах.

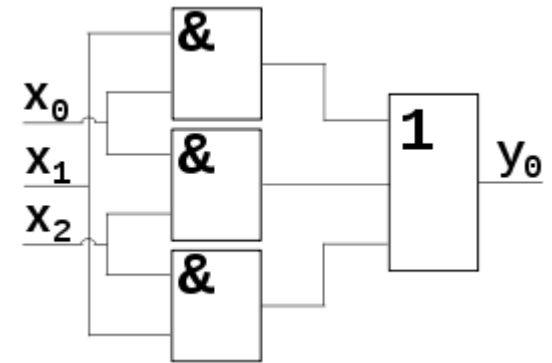


Схема мажоритарного элемента с тремя входами



# ЦИФРОВЫЕ УСТРОЙСТВА

Элементы малой и средней интеграции содержат в едином корпусе наборы однотипных элементов с отдельными входами и выходами. Элементы высокой степени интеграции объединяют в одном корпусе наборы различных элементов, объединенных в законченные функциональные узлы.

По функциональному назначению элементы делят на **логические, запоминающие и специальные.**

**Логические элементы** выполняют логические функции и относятся к классу комбинационных схем, в которых выходной сигнал в некоторый момент времени зависит только от входных сигналов, действующих в тот же момент времени. **Запоминающие элементы** служат для хранения цифровой информации и относятся обычно к классу последовательностных схем, в которых выходные состояния определяются не только состояниями входов в данный момент времени, но и внутренним состоянием самой схемы. **Специальные элементы** обеспечивают сопряженную работу узлов и подсистем ЭВМ, они могут быть цифровыми, аналоговыми и цифроаналоговыми.

# ЦИФРОВЫЕ УСТРОЙСТВА

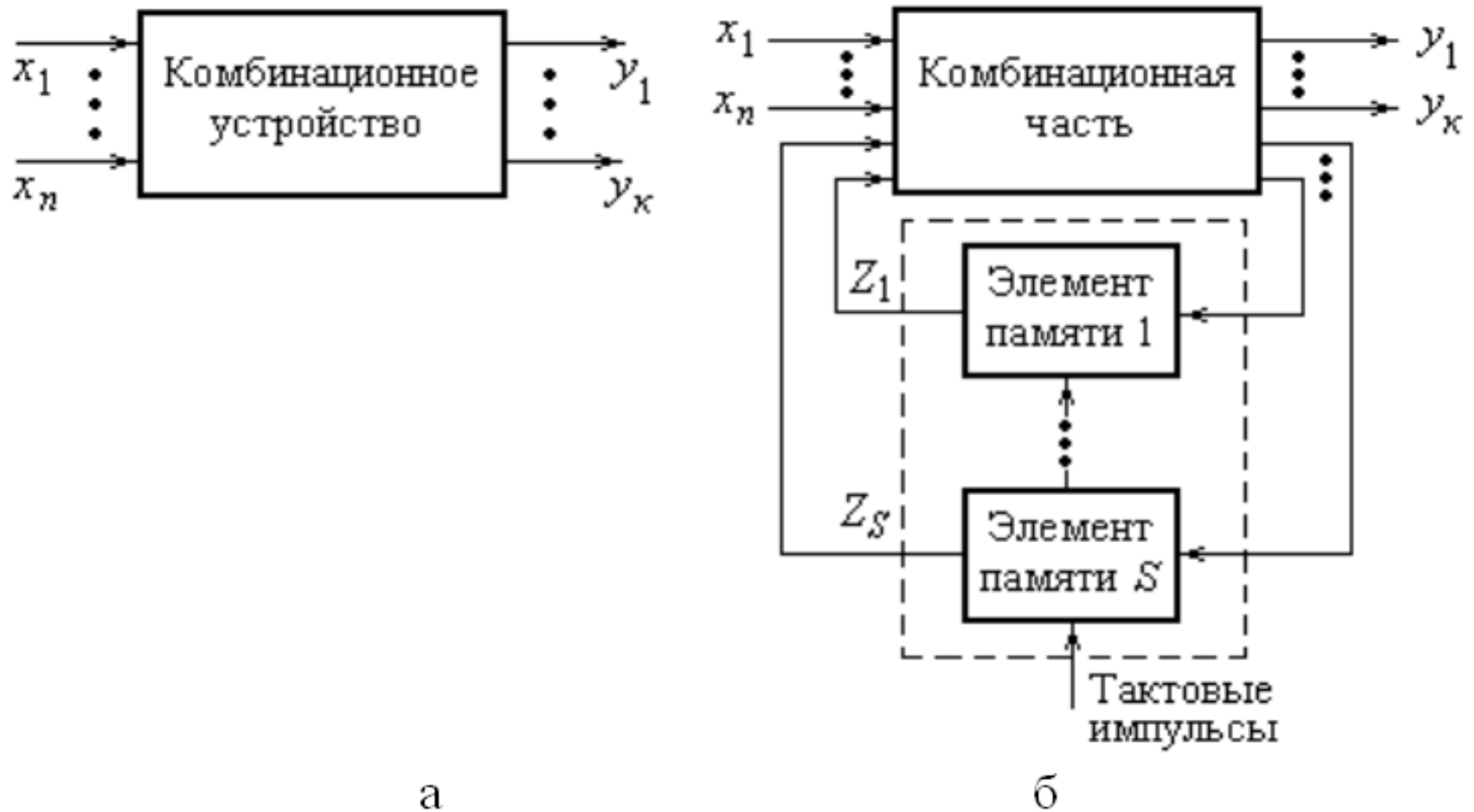


Рис. Структура а - комбинационного и б - последовательного цифровых устройств

# ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

**Последовательностная схема** – это такая логическая схема, выходные сигналы которой определяются не только текущими значениями входных сигналов, но зависят также от последовательности значений входных сигналов в прошлом. Изменение состояния последовательностных схем происходит, как правило, в моменты времени, задаваемые тактовым сигналом от внешнего источника.

Обычно в состав последовательностных устройств входят комбинационные устройства и запоминающие ячейки. Это позволяет учитывать времена задержки прохождения цифровых сигналов через комбинационные цифровые устройства. При этом обычно определяется наибольшее время распространения и изменение состояния цифрового устройства производится с периодом, большим или равным этому времени.

Последовательностными цифровыми устройствами являются триггеры, счетчики, регистры и запоминающие устройства.



# ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

**Последовательностная схема** – это такая логическая схема, выходные сигналы которой определяются не только текущими значениями входных сигналов, но зависят также от последовательности значений входных сигналов в прошлом. Изменение состояния последовательностных схем происходит, как правило, в моменты времени, задаваемые тактовым сигналом от внешнего источника.

Обычно в состав последовательностных устройств входят комбинационные устройства и запоминающие ячейки. Это позволяет учитывать времена задержки прохождения цифровых сигналов через комбинационные цифровые устройства. При этом обычно определяется наибольшее время распространения и изменение состояния цифрового устройства производится с периодом, большим или равным этому времени.

Последовательностными цифровыми устройствами являются триггеры, счетчики, регистры и запоминающие устройства.



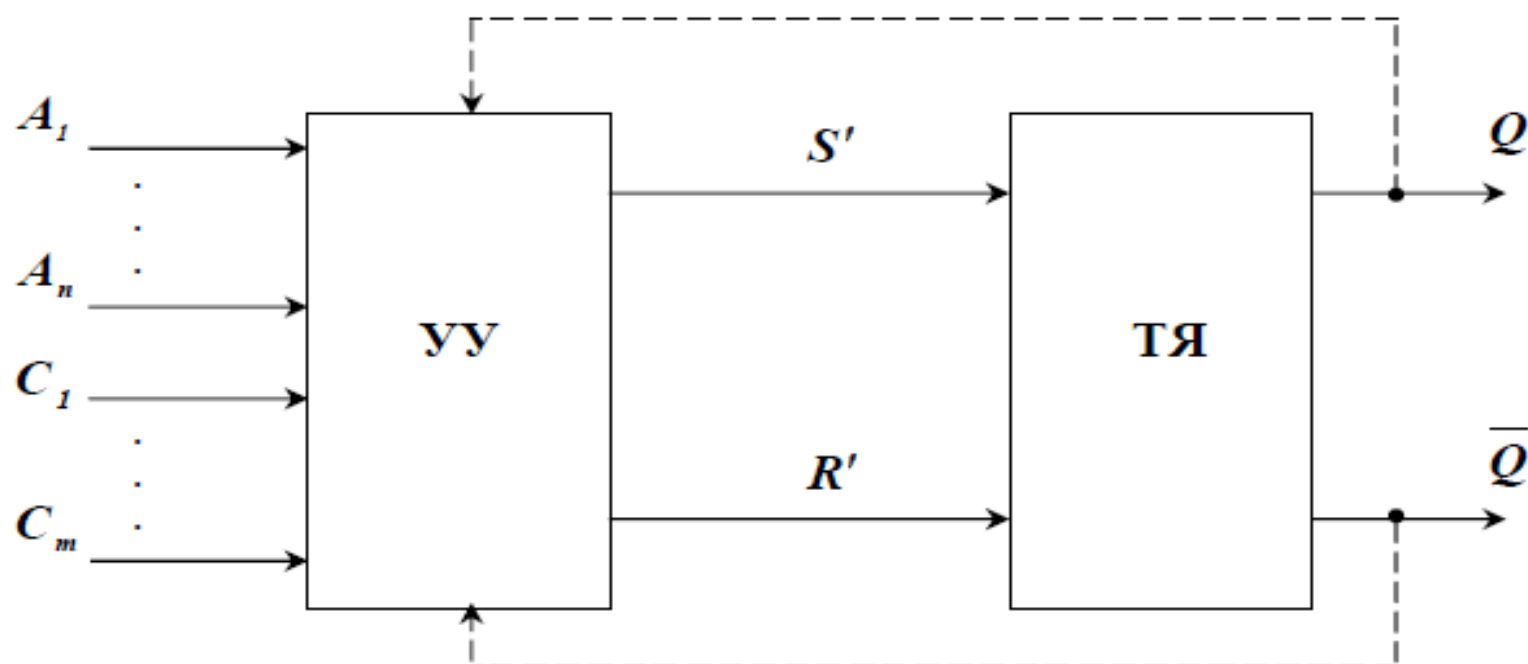
# ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

**Триггеры** представляют собой простейшие последовательные устройства и широко используются в электронных устройствах различного назначения как в виде самостоятельных узлов, так и в качестве элементов для построения более сложных цифровых устройств (счетчиков, регистров, запоминающих устройств). К триггерам относят большой класс устройств, отличительной особенностью которых является способность оставаться в одном из двух устойчивых состояний, которые могут изменяться под действием внешних сигналов. При этом состояния триггера распознаются по уровням выходных напряжений, соответствующих уровням логических "0" и "1". Основным свойством триггера является наличие памяти, под которой подразумевается его способность сохранять свое состояние ("0" или "1") и после прекращения воздействия внешних сигналов. Таким образом, триггер является элементарной ячейкой памяти для хранения одного двоичного разряда числа.

Триггер можно представить в общем случае состоящим из ячейки памяти и устройства управления (порой весьма сложного), преобразующего входную информацию в комбинацию сигналов, под воздействием которых ячейка памяти принимает одно из двух устойчивых состояний.

# ТРИГГЕРЫ

Обобщенная схема триггерного устройства состоит из устройства управления УУ и триггерной ячейки ТЯ.



Структурная схема триггера

На схеме сигналы имеют следующие наименования:  
 $A_1 \dots A_n$  - информационные сигналы;  
 $C_1 \dots C_m$  - тактовые (синхронизирующие) сигналы;  
 $S'$  (set) и  $R'$  (reset) - входные сигналы триггерной ячейки;  
 $Q$  и  $\bar{Q}$  - выходные сигналы триггера.

# ТРИГГЕРЫ

Существует множество разновидностей триггеров, отличающихся выполняемыми функциями, способами управления записью информации, схемотехническими решениями и т.д.

По выполняемым функциям классификацию триггеров производят по состоянию его выходов в момент его срабатывания и после. При этом различают следующие основные виды: RS, JK, T и D - триггеры.

По способу управления записью информации различают:

- асинхронные триггеры с записью непосредственно с поступлением информационного сигнала на его вход;
- тактируемые (синхронные) триггеры с записью информации только при подаче тактирующего импульса. При этом срабатывание триггера может происходить одновременно с поступлением тактирующего импульса (триггер, работающий по уровню), после окончания тактирующего импульса (триггер с внутренней задержкой), прохождения нескольких тактирующих импульсов (многотактные триггеры), или в моменты изменения состояния тактирующего импульса (синхронизация по фронту).



## RS – триггеры

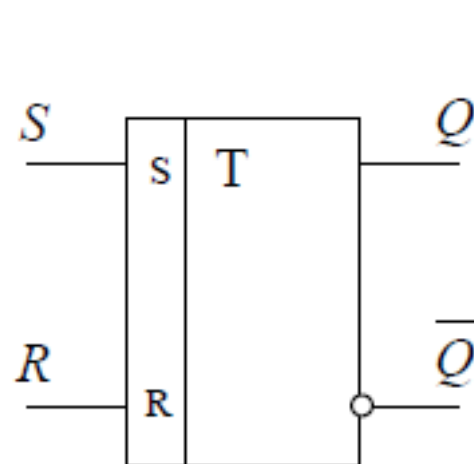
RS–триггер имеет два управляющих входа  $S$  (set) и  $R$  (reset), с помощью которых выполняются установки триггера в то или иное состояние ( $a$ ):

$Q = 1$  при  $S=1$  и  $R=0$  (установка триггера);

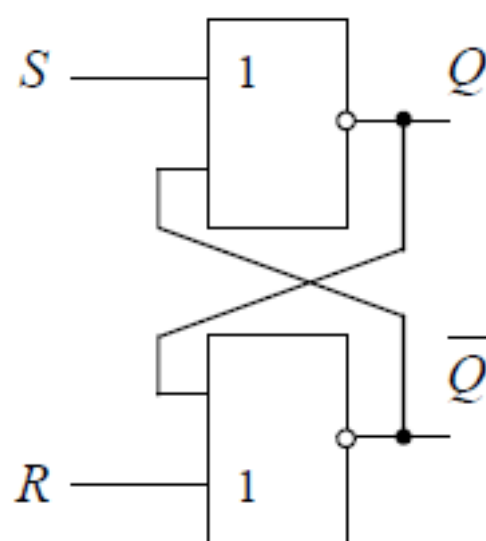
$Q = 0$  при  $S=0$  и  $R=1$  (сброс триггера);

$Q^{n+1} = Q^n$  при  $S=R=0$  (режим хранения предыдущего состояния);

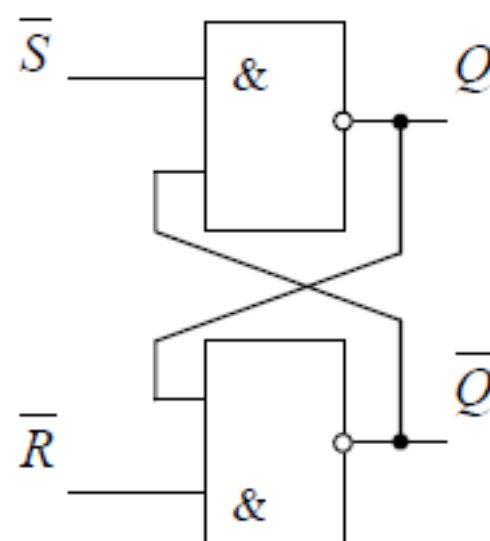
$S=R=1$  – запрещенная комбинация управляющих сигналов, которая может привести к неопределенному состоянию триггера.



а



б



в

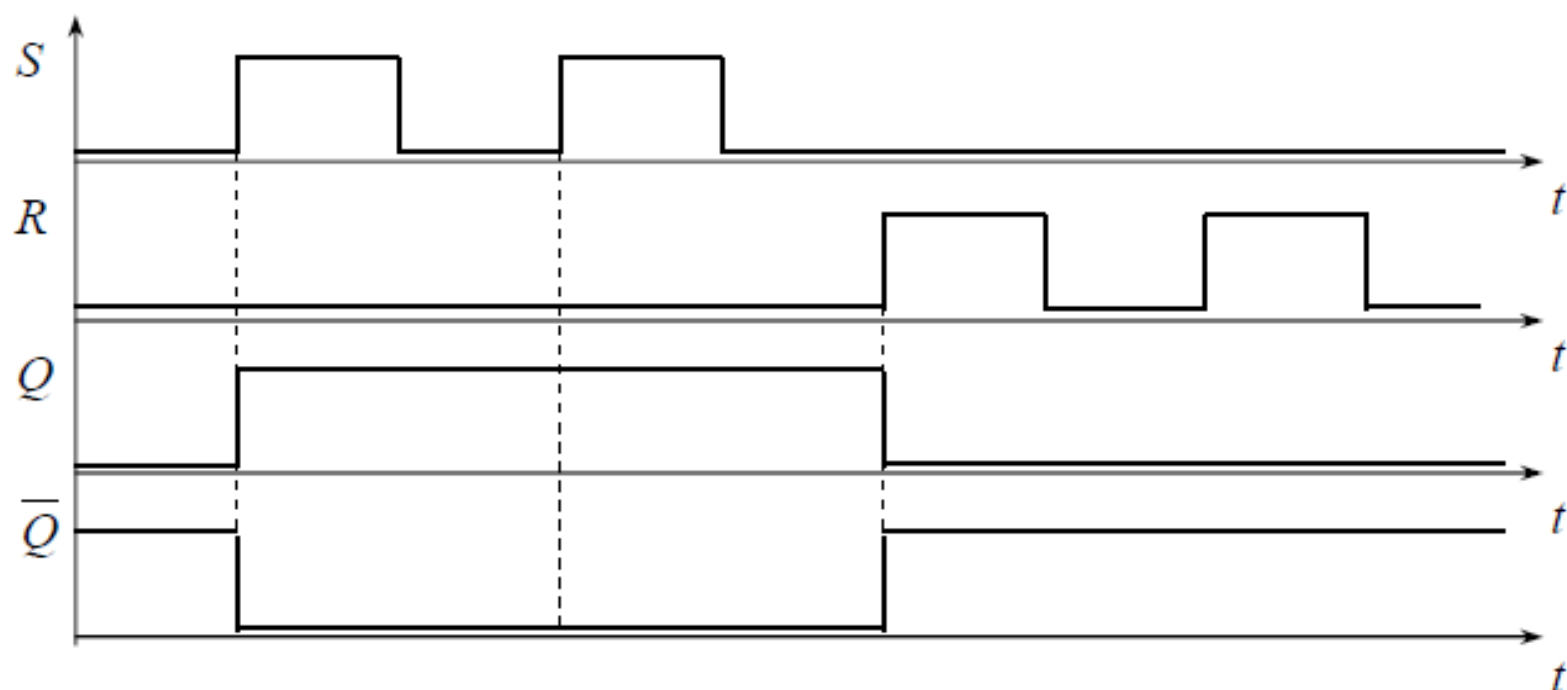
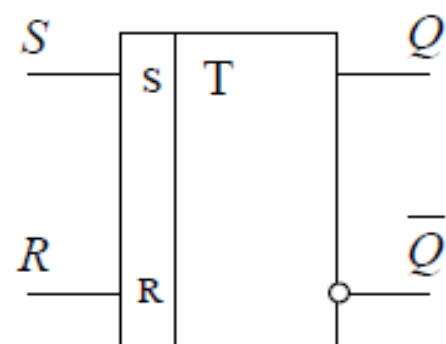
RS-триггеры



## RS – триггеры

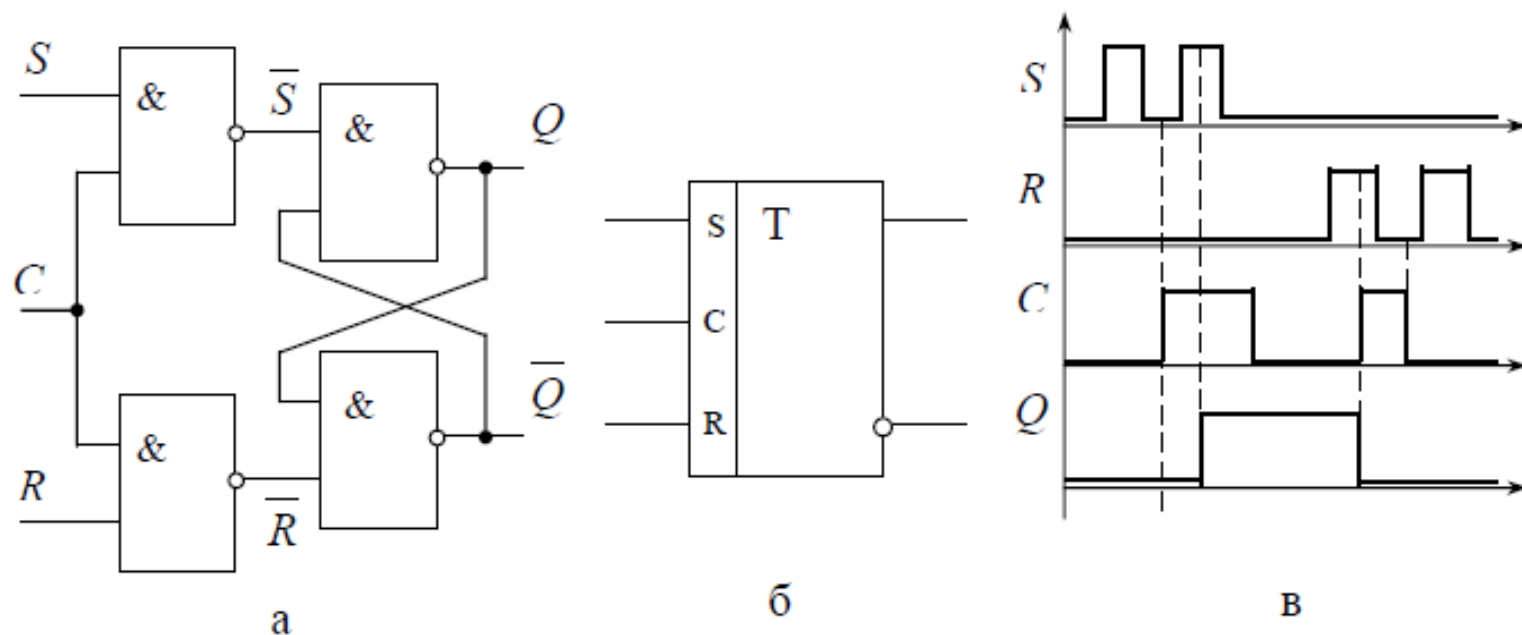
Таблица состояний  
RS-триггера

$R$	$S$	$Q^{n+1}$
0	0	$Q^n$
0	1	1
1	0	0
1	1	×

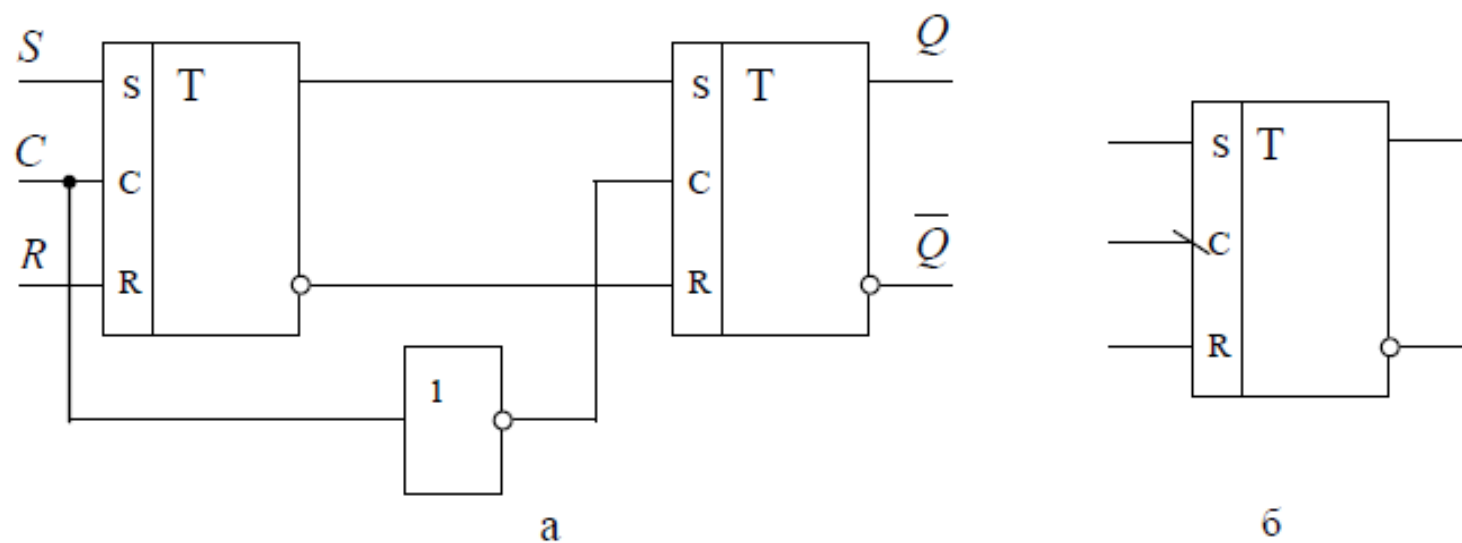


Временные диаграммы работы RS-триггера

## RS – триггеры



### RS-триггер с синхронизацией по уровню



### RS-триггер с синхронизацией по фронту

# RS- ТРИГГЕРЫ

Для RS- триггера комбинация  $S=R=1$  является запрещенной.

После такой комбинации информационных сигналов состояние триггера будет неопределенным: на его выходе Q может быть 0 и 1.

Существуют разновидности RS- триггеров, называемые E-, R- и S- триггерами, для которых сочетание  $S=R=1$  не является запрещенным.

E- триггер при  $S=R=1$  не изменяет своего состояния.

S- триггер при  $S=R=1$  устанавливается в состояние  $Q=1$ ,  
а R- триггер в этом случае устанавливается в состояние  $Q=0$ .

RS- триггеры могут быть асинхронными или синхронными  
(в этом случае у них имеется вход C).

# JK-триггеры

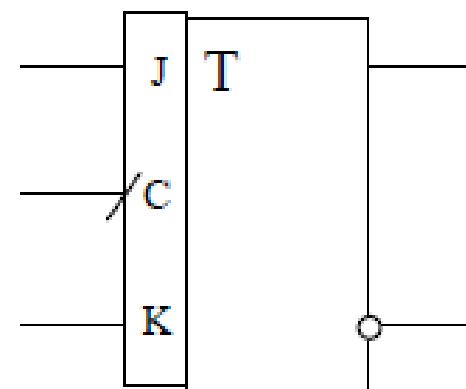
JK-триггер имеет два управляющих входа J (jump) и K (keep) и функционирует подобно RS-триггеру, но при этом не имеет запрещенных комбинаций управляющих сигналов. J - вход подобен S – входу, а K-вход подобен R-входу. При всех комбинациях сигналов на входе, кроме J=K=1, он действует подобно RS-триггеру. При J=K=1 в каждом такте происходит «опрокидывание» триггера и его состояние меняется на противоположное. На рис. показано условное обозначение JK-триггера с синхронизацией по переднему фронту.

Таблица состояний  
RS-триггера

$R$	$S$	$Q^{n+1}$
0	0	$Q^n$
0	1	1
1	0	0
1	1	×

Таблица состояний  
JK-триггера

$J$	$K$	$Q^{n+1}$
0	0	$Q^n$
0	1	0
1	0	1
1	1	$\overline{Q^n}$

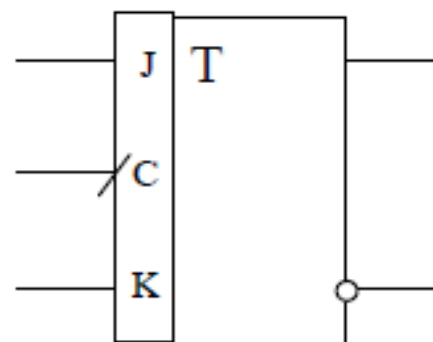


JK-триггер

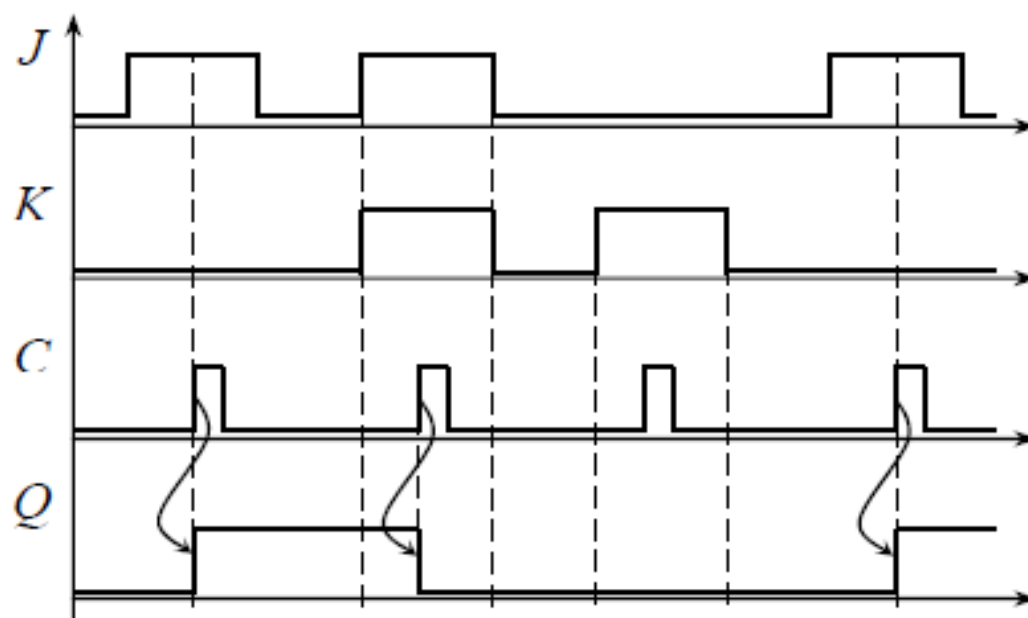
# JK-триггеры

Таблица состояний  
JK-триггера

$J$	$K$	$Q^{n+1}$
0	0	$Q^n$
0	1	0
1	0	1
1	1	$\overline{Q^n}$



JK-триггер



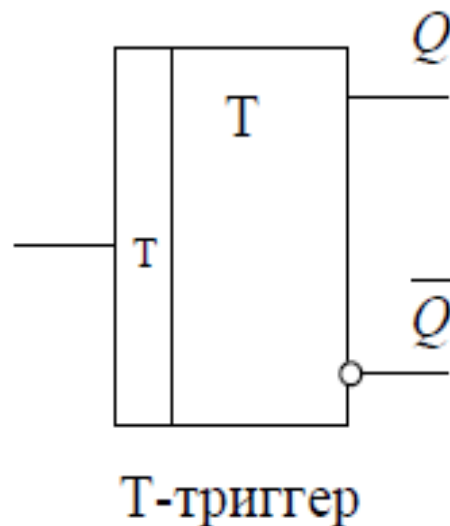
Диаграммы работы JK-триггера

## Т-триггеры

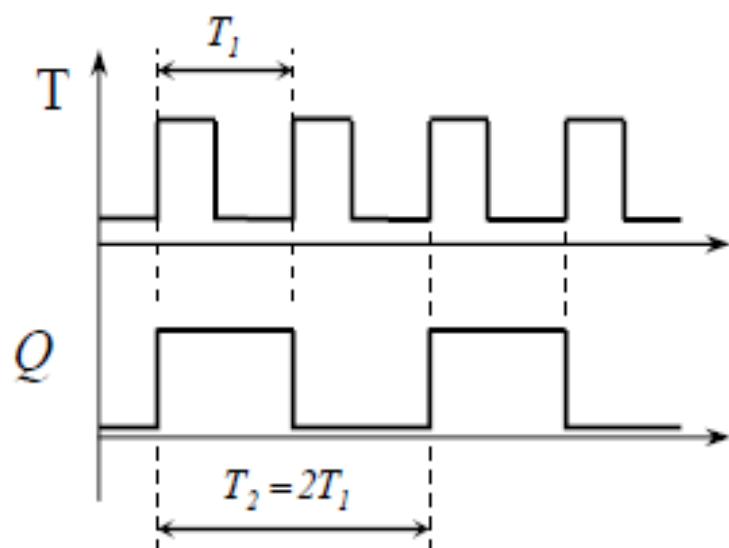
Т-триггеры иначе называются *счетными* и применяются для построения счетчиков и делителей частоты. Такой триггер имеет один тактовый вход и его состояние меняется каждый раз при подаче счетного импульса  $T=1$  и остается неизменным при  $T=0$ .

Таблица состояний  
Т-триггера

$T$	$Q^{n+1}$
0	$Q^n$
1	$\overline{Q^n}$



Т-триггер делит частоту входных импульсов в 2 раза.

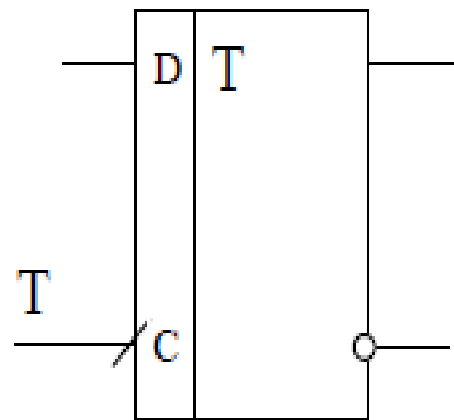


Для получения больших значений коэффициента деления частоты применяется каскадное соединение Т-триггеров.

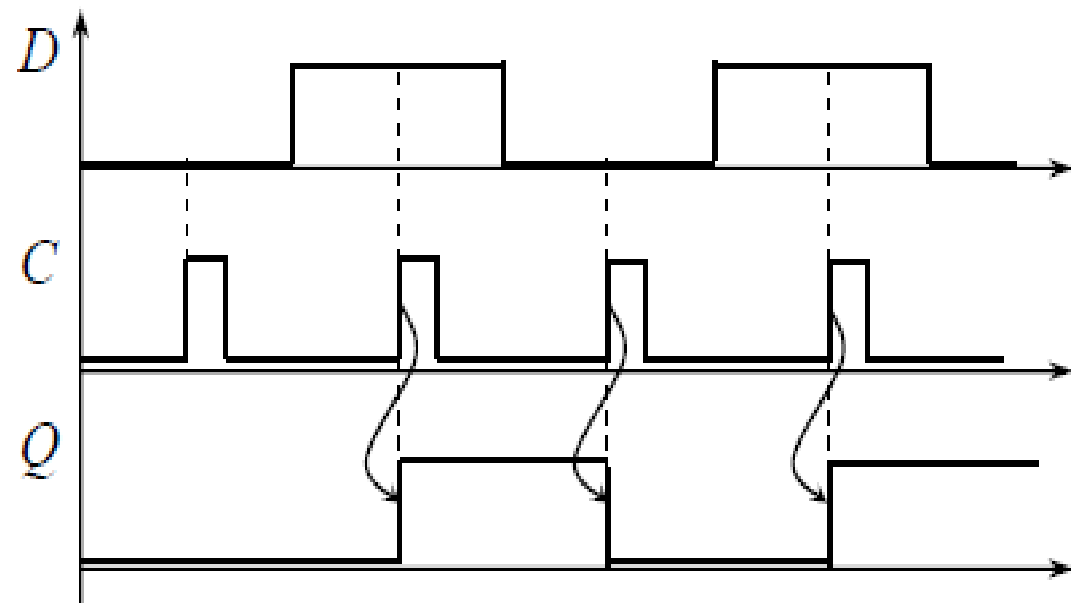
## D -триггеры

Отличительной особенностью D-триггера (триггера задержки) является то, что он сохраняет информацию, поступившую на D-вход в предыдущем такте работы до прихода синхроимпульса, т.е. его состояние может изменяться с задержкой на один такт. Синхронизация работы производится по переднему или заднему фронту.

D-триггеры являются основой для построения регистров.



## D-триггер



# СЧЕТЧИКИ

**Счетчиками** называют устройства для подсчета числа входных импульсов и фиксации этого числа в каком-либо коде. В процессе работы счетчик последовательно изменяет свое состояние. Количество возможных состояний называется *модулем счета*  $K_{СЧ}$  или *емкостью счетчика* (предельное число импульсов, которое может быть подсчитано счетчиком).

Основными элементами счетчика являются триггеры, количество которых определяет число разрядов счетчика  $n$  и его модуль счета  $K_{СЧ}$ . Чаще всего в цифровой технике используются двоичные счетчики. Нулевое состояние всех триггеров двоичного счетчика принимается за нулевое состояние счетчика. При подаче счетных импульсов счетчик последовательно изменяет свое состояние от нулевого до максимального, равного  $K_{СЧ}$ . Например, при  $n=3$  у двоичного счетчика  $K_{СЧ}=2^n=8$ , т. е. счетчик имеет 8 устойчивых состояний и каждый восьмой импульс, поступающий на его вход, будет возвращать счетчик в исходное состояние (обнуление счетчика). Это свойство позволяет использовать счетчики для деления частоты. Если входные импульсы следуют с частотой  $f_{вх}$ , то частота выходных импульсов

$$f_{вых} = f_{вх} / K_{СЧ}.$$



# СЧЕТЧИКИ

Быстродействие счетчика характеризуется максимально допустимой частотой поступления счетных импульсов  $f_{сч}$  и временем установления счетчика  $t_{уст}$  - интервалом времени между моментом окончания счетного импульса и моментом установления кода.

В зависимости от способа кодирования внутренних состояний различают:

- двоичные счетчики;
- двоично-десятичные (декадные) счетчики;
- кольцевые счетчики – состояние счетчика определяется местоположением одной единственной 1 или 0;
- счетчики Джонсона – состояние счетчика определяется количеством 1 или 0.

В зависимости от способа подсчета различают следующие виды счетчиков:

- суммирующие;
- вычитающие;
- реверсивные;
- кольцевые.

# СЧЕТЧИКИ

По способу тактирования работы различают:

- синхронные счетчики, для работы которых требуется синхросигнал;
- асинхронные счетчики, работающие без синхросигналов.

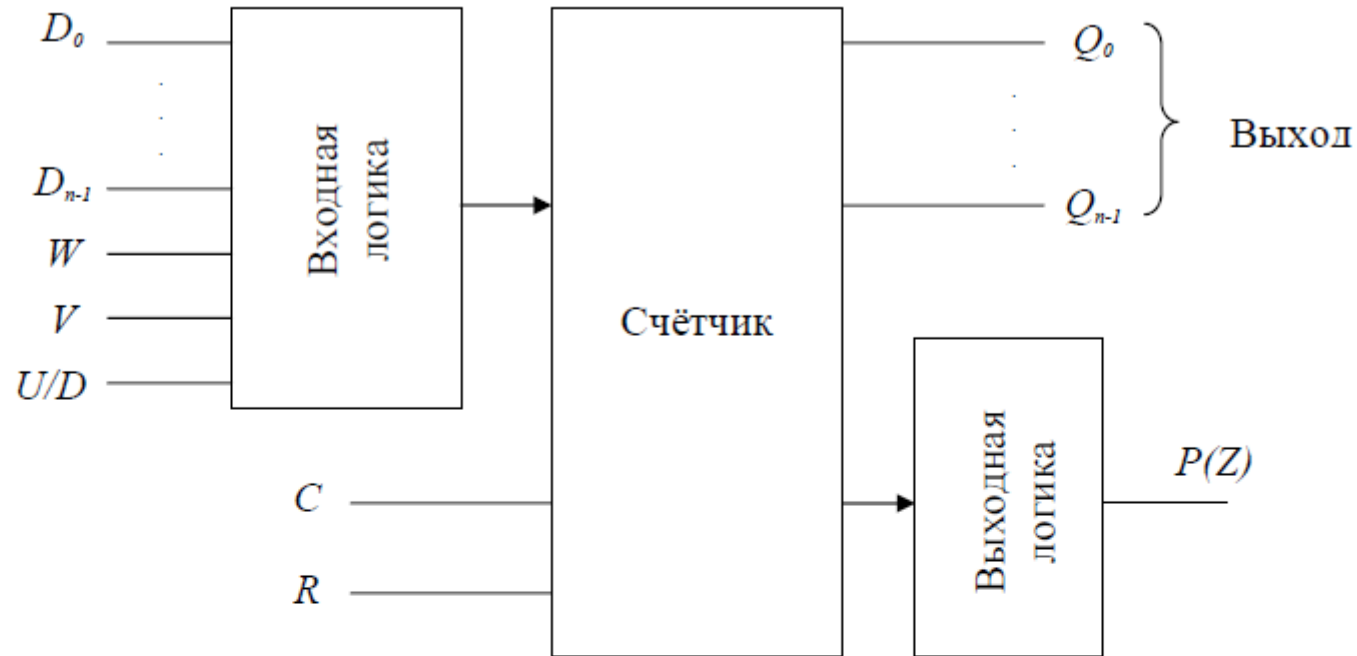
По структурной организации, зависящей от способа подачи счетных импульсов на разряды счетчиков, различают:

- счетчики с последовательным (каскадным) переносом – счетные импульсы поступают только на вход первого разряда, а с его выхода - на вход второго и т.д. (счетчики с последовательным переносом отличаются простотой, но при этом имеют низкое быстродействие);

- счетчики с параллельным переносом – счетные импульсы поступают одновременно на все разряды (такие счетчики имеют более сложную организацию, но обеспечивают высокое быстродействие);

- параллельно-последовательные счетчики, представляющие собой комбинацию первых двух способов подачи счетных импульсов (такие счетчики используются для получения больших значений модуля счета).

# СЧЕТЧИКИ



Обобщенная схема счетчика

Основными узлами являются собственно сам счетчик, входная логика и выходная логика. Входная логика предназначена для управления работой счетчика, а выходная – для индикации конца счета или формирования сигнала переноса или займа  $P(Z)$ .

Перечень основных входных сигналов:

$U/D$  – сигнал управления направлением счета (для реверсивных счетчиков);

$C$  – счетные импульсы;

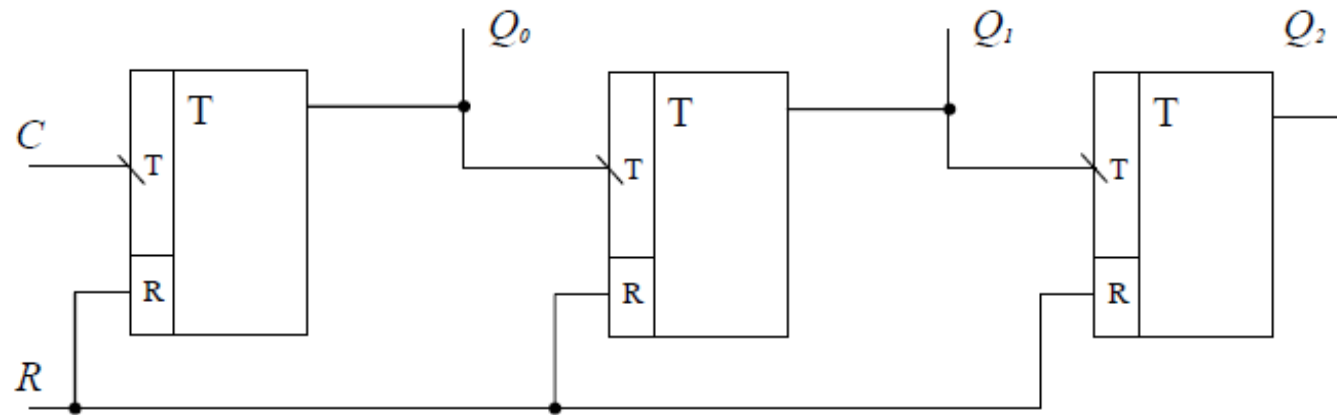
$R$  – сигнал сброса;

$V$  – сигнал разрешения счета (для синхронных счетчиков);

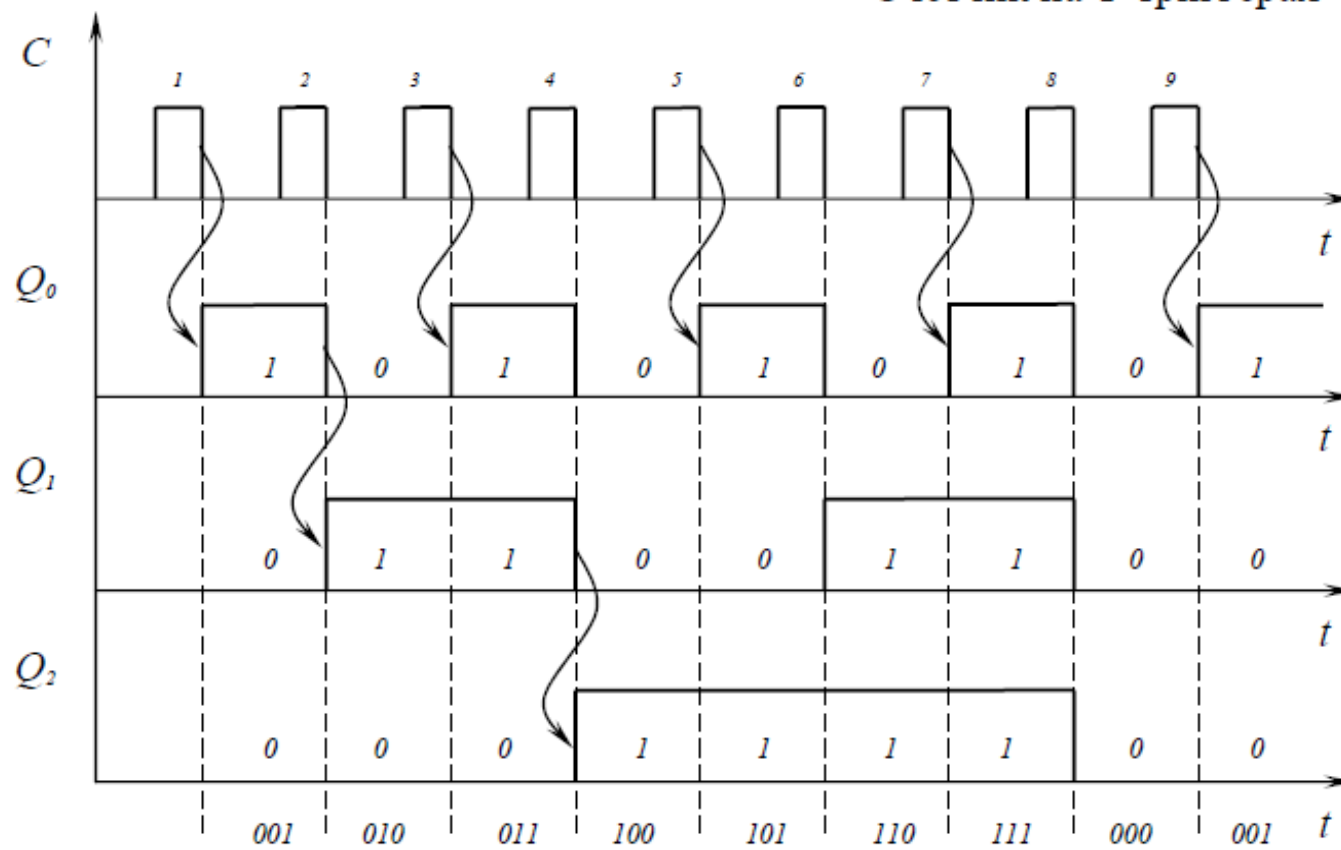
$D_0 - D_{n-1}$  – параллельный код для предварительной установки счетчика (для счетчиков с предустановкой);

$W$  – сигнал разрешения предварительной установки.

# Асинхронные двоичные счетчики



Счетчик на Т-триггерах



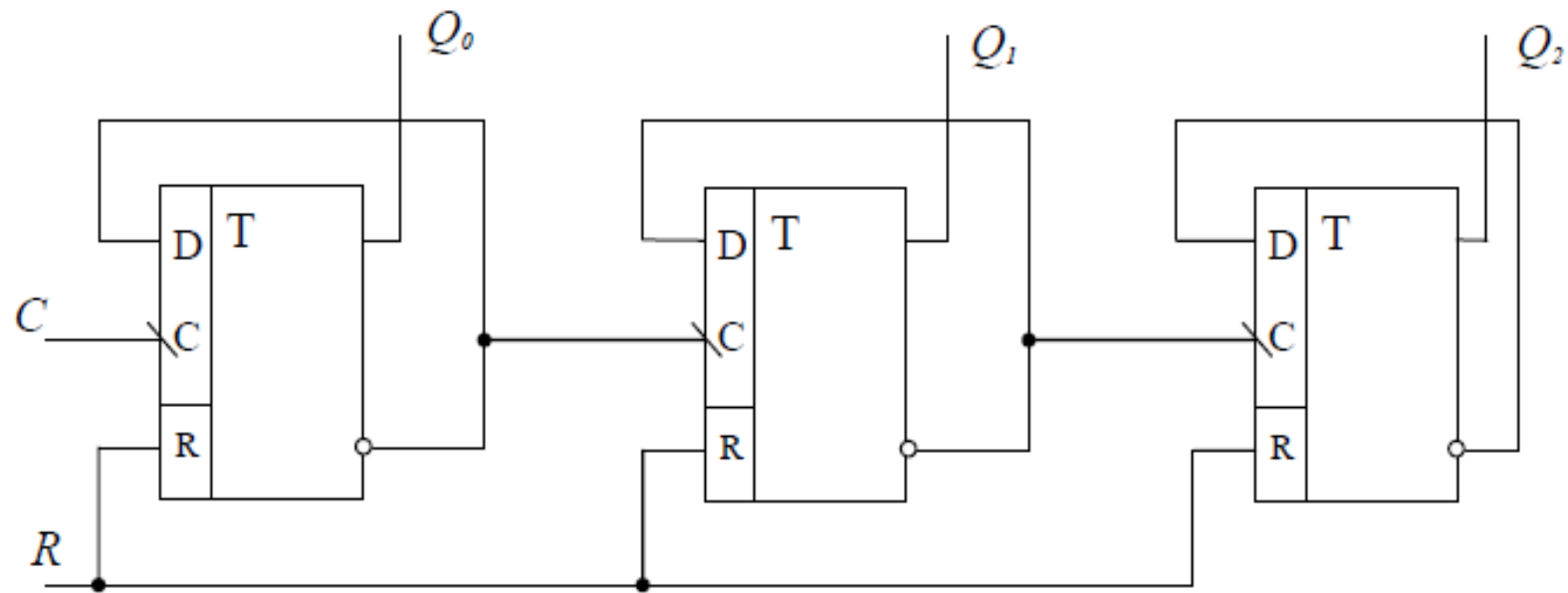
Обнуление счётчика

# Асинхронные двоичные счетчики

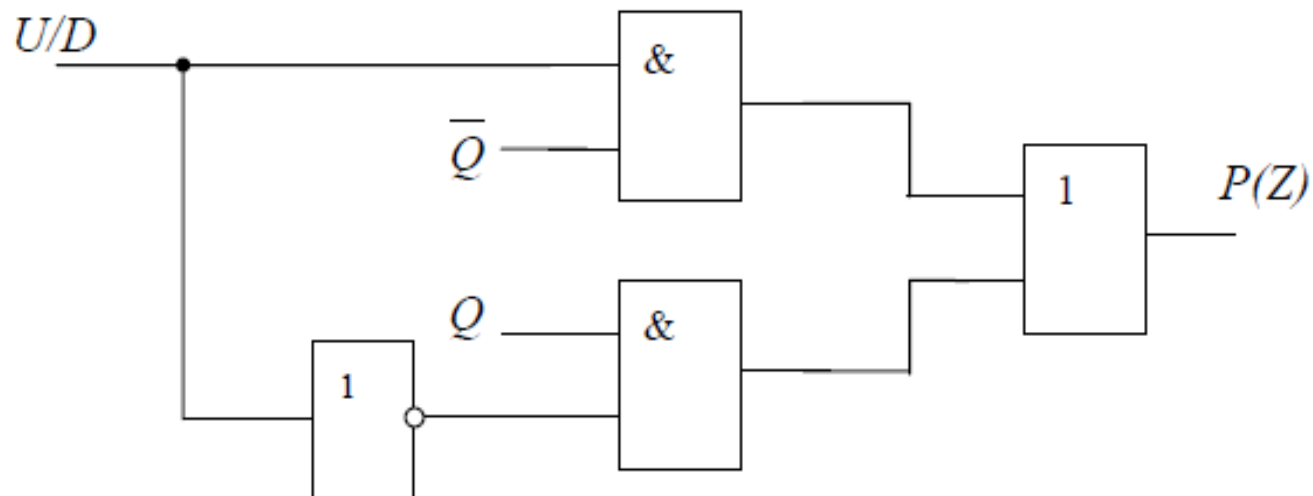
**Асинхронный** суммирующий счетчик можно построить на счетных триггерах любого типа. В большинстве случаев для этих целей используют JK- или D-триггеры в счетном режиме. Простейший трехразрядный двоичный счетчик на D-триггерах состоит из соединенных последовательно трех счетных триггеров, таким образом, что инверсный выход  $\bar{Q}_k$  каждого триггера соединен с входом  $C_{k+1}$  последующего. При поступлении счетных импульсов на вход  $C_1$  триггеры счетчика будут изменять свои состояния, описываемые последовательно возрастающими двоичными числами. Для приведения счетчика в начальное состояние используется сигнал сброса  $R$ , поступающий одновременно на все входы  $R$  триггеров.

С приходом каждого последующего импульса на вход  $C_1$  переключаются сразу несколько триггеров. Однако переключение этих триггеров происходит не одновременно, а с некоторой задержкой относительно друг друга. Это приводит к задержке в установлении выходного кода после поступления счетного импульса на вход  $C_1$ . При большом числе разрядов счетчика задержка выходного сигнала может быть значительной и сравнимой с периодом поступления счетных импульсов на вход  $C_1$ .

# Асинхронные двоичные счетчики



## Суммирующий счетчик на D-триггерах



### Схема переноса для реверсивного счетчика



## Асинхронные двоичные счетчики

При построении асинхронного **вычитающего** счетчика достаточно заменить инверсные выходы  $\bar{Q}$  триггеров на прямые выходы  $Q$ . В этом случае при поступлении импульса сброса  $R$  на всех выходах двоичного счетчика установятся единичные уровни, а при поступлении счетных импульсов на вход  $C_1$  триггеры счетчика будут изменять свои состояния, описываемые последовательно убывающими двоичными числами.

Для построения асинхронного **реверсивного** счетчика, который может работать как в режиме суммирования, так и в режиме вычитания, можно с помощью логической схемы обеспечить подачу сигналов с инверсного выхода  $\bar{Q}$  при суммировании или с прямого выхода  $Q$  – при вычитании от предыдущего триггера на счетный вход последующего, как показано на рис. Эта схема включается между выходом одного разряда счетчика и входом другого и, в зависимости от управляющих сигналов – сложение ( $U$ ) или вычитание ( $D$ ), на вход последующего разряда поступает сигнал переноса  $P$  или сигнал займа  $Z$ .



# Синхронные двоичные счетчики

У **синхронных счетчиков** все разрядные триггеры синхронизируются параллельно одними и теми же синхроимпульсами, поступающими из источника этих импульсов. **Асинхронные счетчики** имеют последовательную синхронизацию, т.е. каждый последующий разрядный триггер синхронизируется выходными импульсами триггера предыдущего разряда. Асинхронные счетчики иногда называют последовательными, а синхронные счетчики - параллельными.

**Синхронные счетчики** (параллельные), в свою очередь, подразделяются на синхронные счетчики с **последовательным переносом** (последовательно-синхронные) и синхронные счетчики с **параллельным переносом** (параллельно-синхронные). Синхронные счетчики имеют более высокую скорость счета, чем асинхронные.

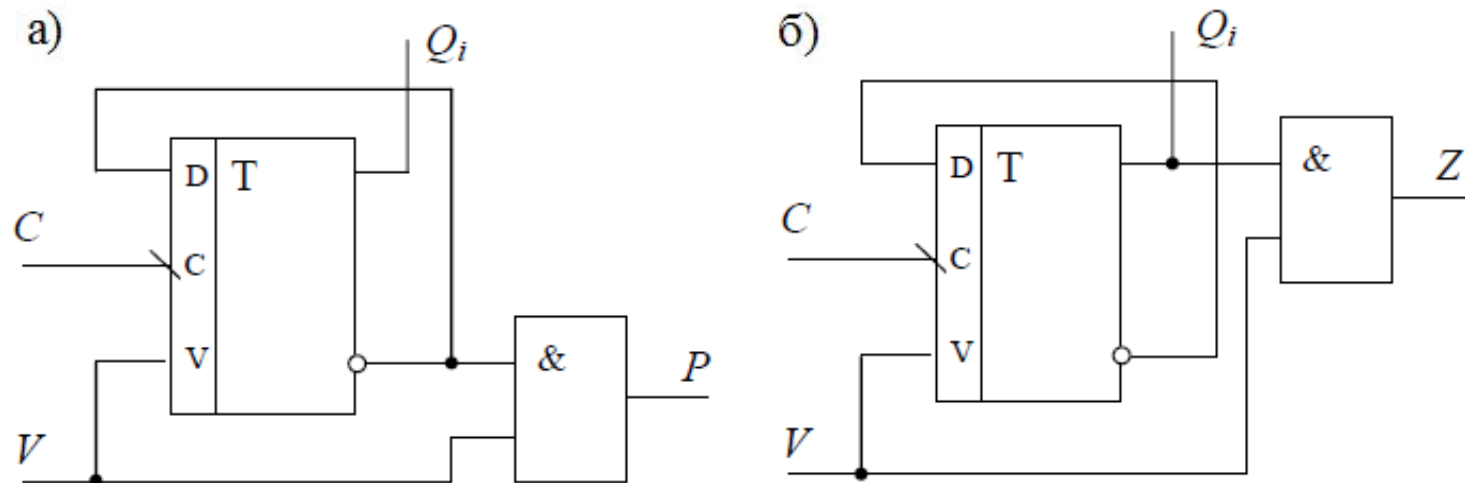
По способу подачи синхроимпульсов последовательно-синхронные счетчики параллельные, т.е. синхроимпульсы поступают на все триггеры счетчика параллельно, а по способу управления (подачи управляющих импульсов) – последовательные.

В параллельно-синхронных счетчиках формирование управляющих уровней и их подача на соответствующие входы триггеров счетчика осуществляется одновременно, т.е. параллельно.

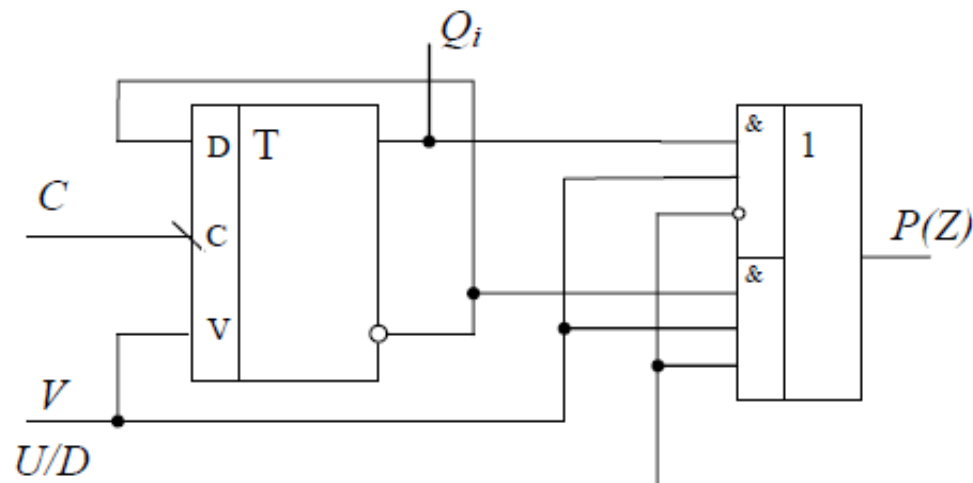


# Синхронные двоичные счетчики

Для построения **синхронных счетчиков** используют различные типы счетных синхронных триггеров. На рис. приведены схемы одноразрядных синхронных счетных ячеек (а – суммирующая, б – вычитающая).

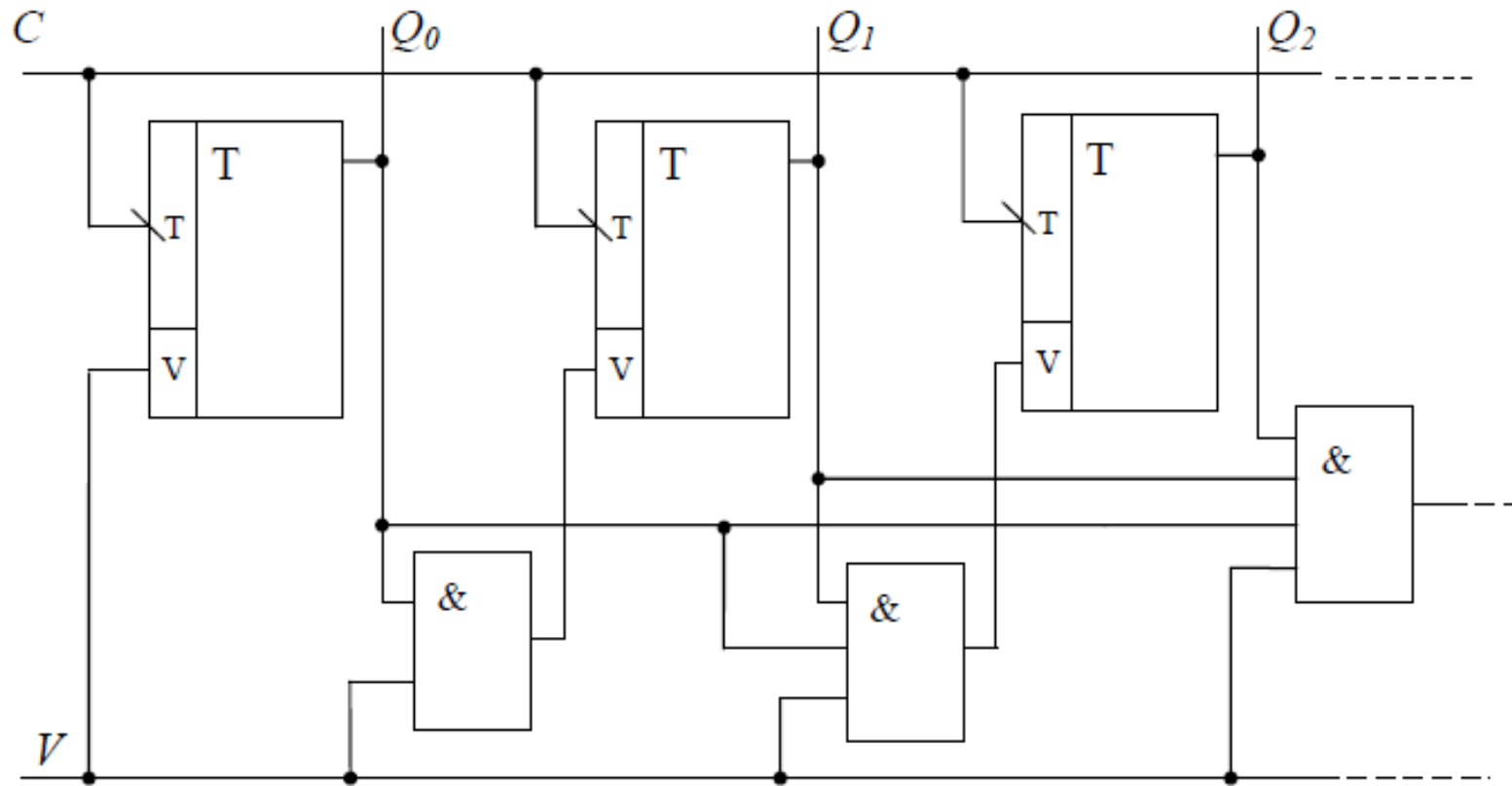


## Схемы синхронных счетных ячеек



### Счетная ячейка синхронного реверсивного счетчика

# Синхронные двоичные счетчики



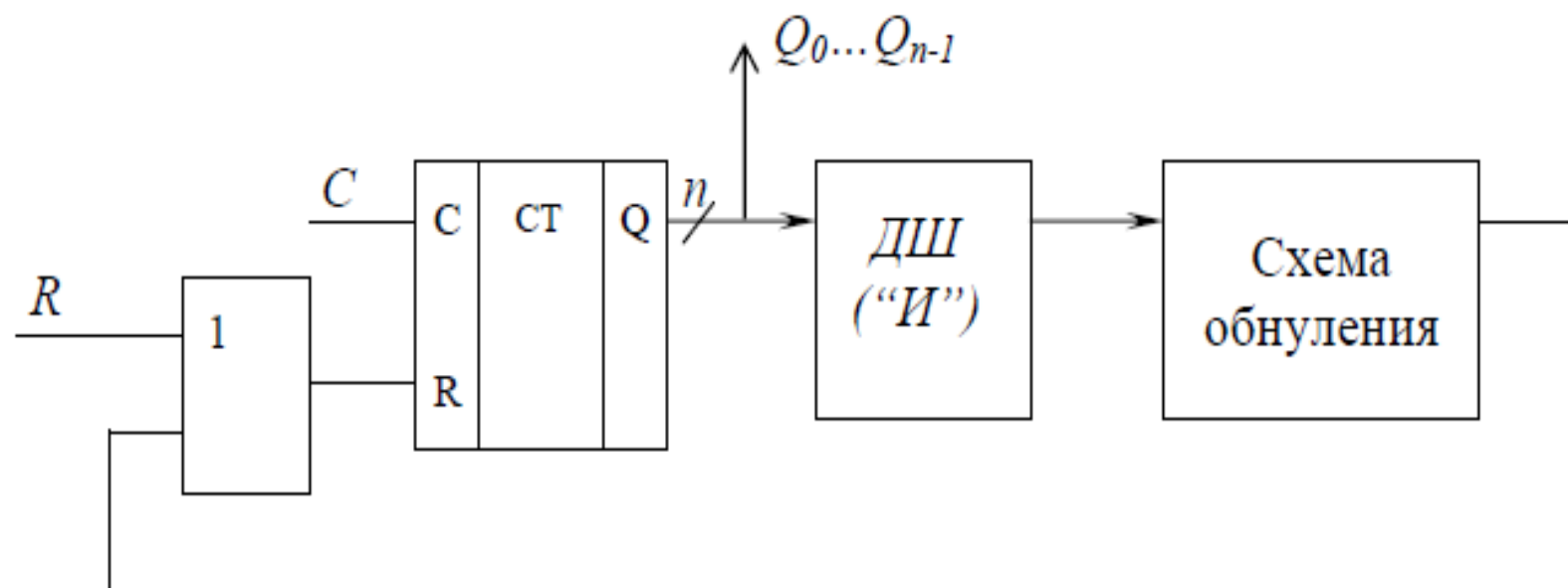
Синхронный счетчик с параллельным переносом

Быстродействие счетчиков с параллельным переносом не зависит от числа разрядов и определяется задержкой распространения сигналов через триггеры и элементы И. По сравнению со счетчиками с последовательным переносом максимальная частота счета может быть увеличена в  $(n - 1)$  раз. При повышении разрядности счетчиков возникают проблемы с реализацией многовходовых схем И, поэтому в таких случаях наиболее оптимальными по быстродействию и аппаратным затратам являются счетчики с параллельно-последовательным переносом.

## Счетчики с произвольным модулем счета

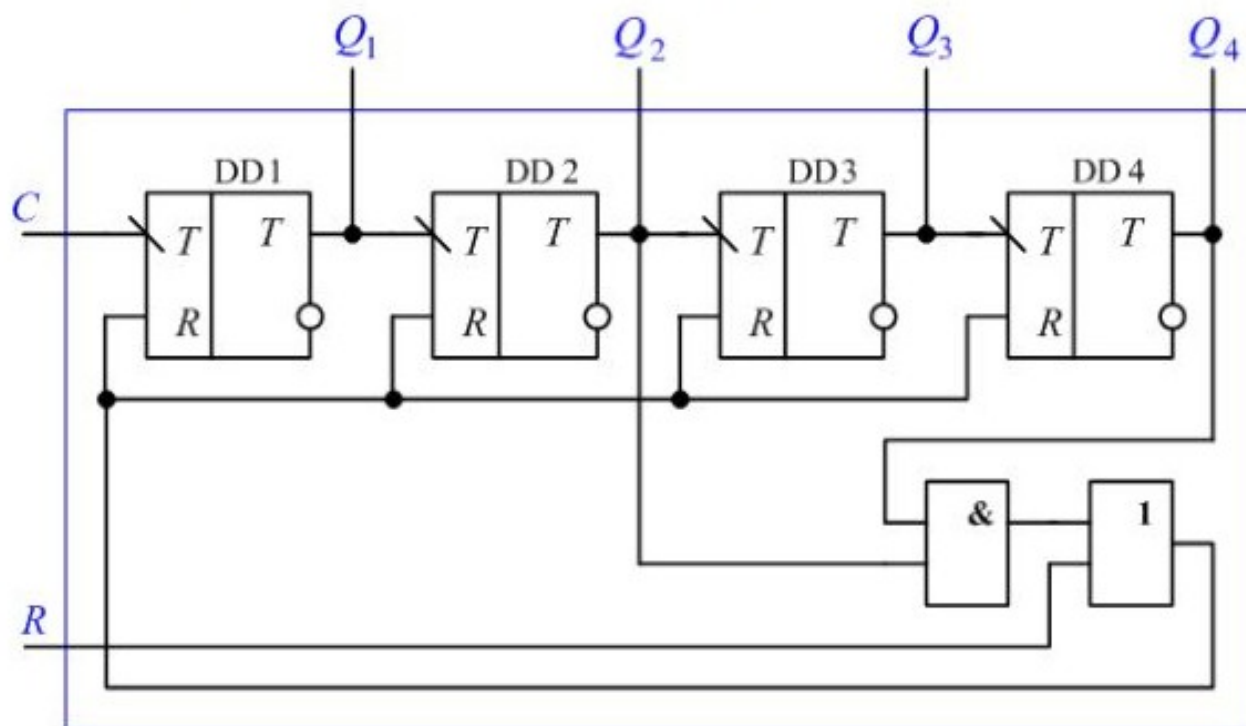
Во всех рассмотренных выше счетчиках состояние последовательно меняется от 0 до  $2^n - 1$  или от  $2^n - 1$  до 0. В ряде практических случаев требуются счетчики с произвольным модулем счета, например на 10 (двоично-десятичные счетчики) или на 9, 23, 59 (для реализации электронных часов). Существуют два основных способа реализации счетчиков с произвольным модулем счета.

Счетчики с принудительным сбросом. В таких счетчиках производится принудительный сброс по достижении заданного модуля счета.

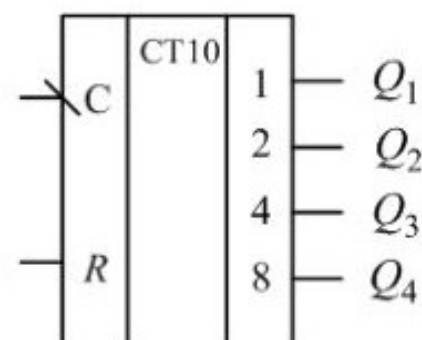


Счетчик с принудительным сбросом

Для построения счетчиков по произвольному модулю  $K_{сч}$  с принудительным сбросом в схему соответствующего двоичного счетчика вводится обратная связь для исключения лишних состояний. Двоичный суммирующий или вычитающий счетчик переключается до установки некоторого значения  $K_{сч}$ . Это состояние выявляется специальной схемой, на выходе которой формируется сигнал установки счетчика в нулевое состояние. Схемы счетчика по модулю  $K_{сч}$  наиболее просты при использовании двоичных счетчиков с последовательным переносом. Дешифратором обычно служит логический элемент И. На рис. показано как у асинхронного двоичного счетчика на четырех триггерах можно уменьшить модуль счета с 16 до 10.



Функциональная схема десятичного асинхронного суммирующего счетчика с  $k_{сч} = 10$

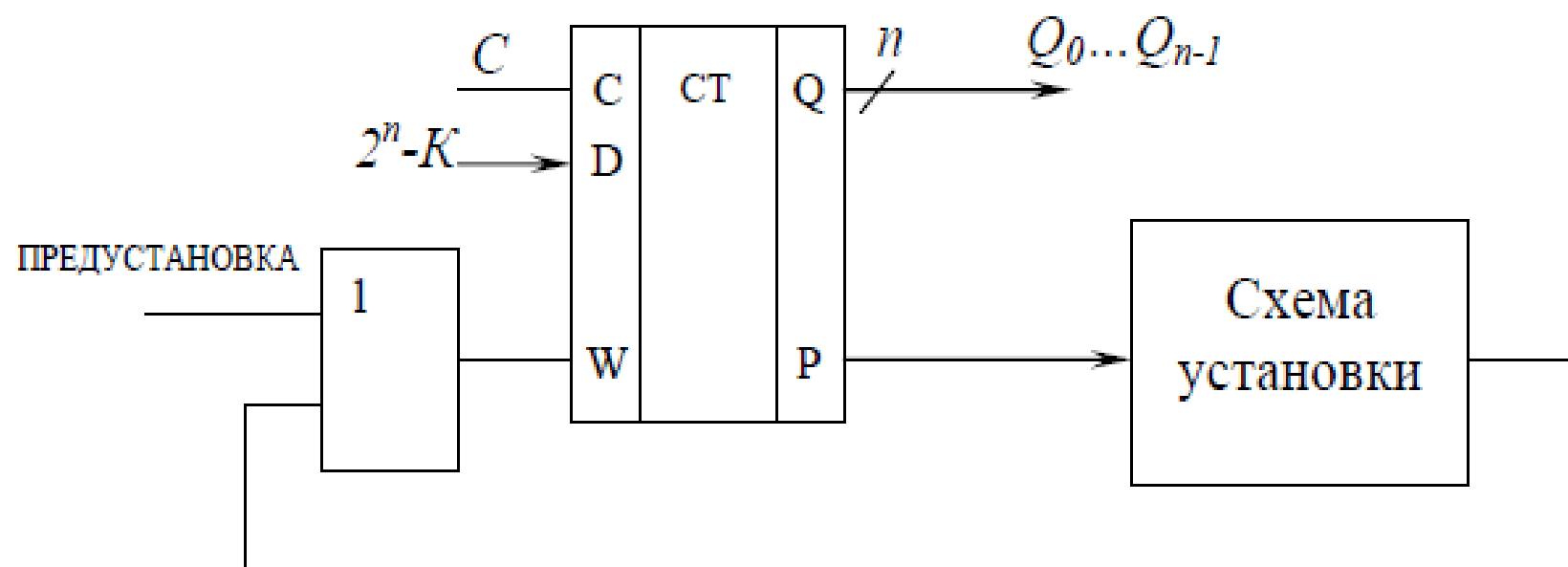


Условное графическое обозначение десятичного асинхронного суммирующего счетчика с  $k_{сч} = 10$

Здесь реализован сброс на нуль каждый раз, когда на счетчике состояние 1010.

## Счетчики с произвольным модулем счета

Счетчики с принудительным насчетом. Такие устройства реализуются на базе счетчиков с предварительной установкой кода. Схема счетчика показана на рис. В счетчик по входам параллельной загрузки загружается код дополнения  $K$  до  $2^n$ . Конец счета в этом случае обнаруживается по естественному переполнению счетчика, при котором формируется сигнал переноса, которым производится перезагрузка счетчика.



Счетчик с принудительным насчетом

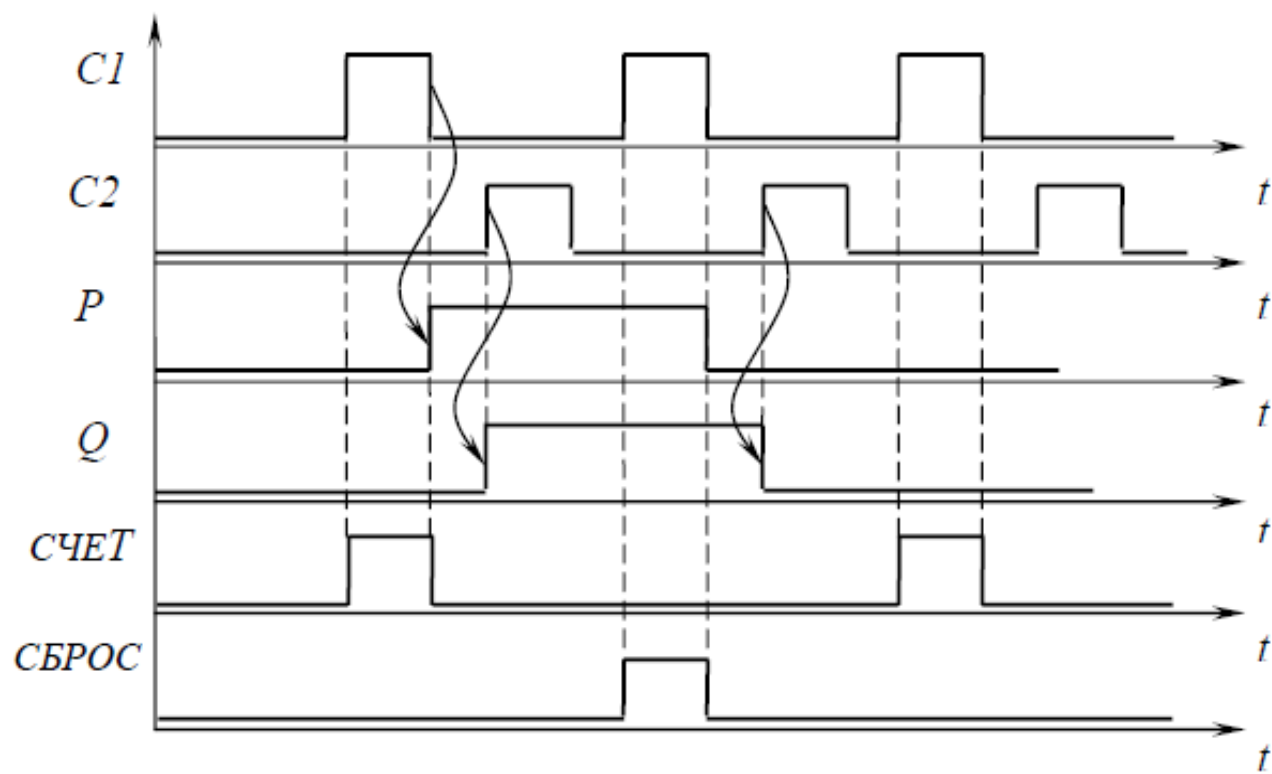
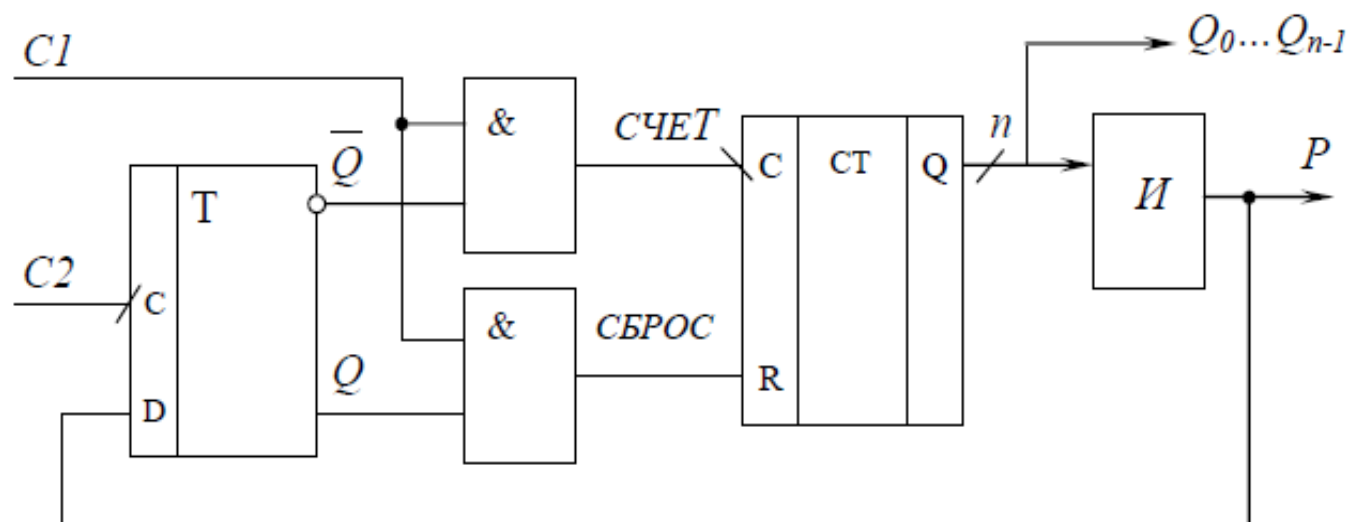


# Счетчики с произвольным модулем счета

Достоинством счетчика с насчетом является использование штатных сигналов переноса и параллельной загрузки, а также простота изменения модуля счета. Недостаток заключается в неестественности последовательности изменения кода, что может потребовать в ряде случаев дополнительного перекодирования. Чаще всего этот вариант построения счетчика применяется при делении частоты входных импульсов, когда порядок изменения кода внутри счетчика не имеет значения.

Промышленностью выпускаются ИС счетчиков с различными функциональными возможностями, например, с параллельным переносом и с предустановкой.

Счетчики с переменным коэффициентом счета имеют очень широкое применение в радиотехнике при реализации делителей частоты с переменным коэффициентом деления и при реализации различных программируемых таймеров. В основном при реализации таких устройств используются синхронные параллельные счетчики с синхронными схемами предустановки и/или сброса.

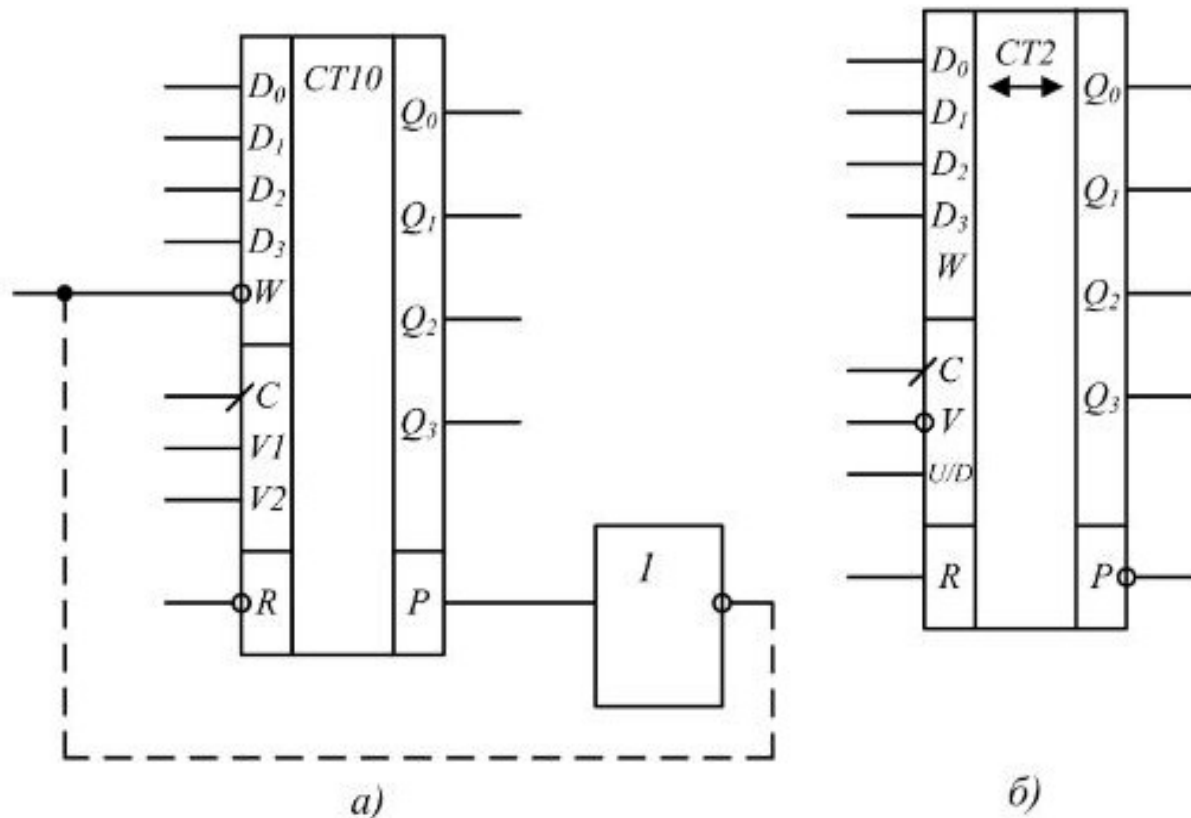


Универсальный способ построения счетчика  
с произвольным модулем счета

# ЦИС счетчиков

Промышленностью выпускаются ИС счетчиков с различными функциональными возможностями. В качестве примера приведем ИС двоично-десятичного счетчика ( $K_{сч} = 10$ ) с параллельным переносом и предустановкой (рис. а).

Счет импульсов (без ввода информации по входу предустановки), подаваемых на вход  $C$ , происходит при  $W=V1=V2=R=1$ . При этом состояния счетчика на выходах  $Q_0...Q_3$  будут изменяться в двоично-десятичном коде от 0 до 9.





## ЦИС счетчиков

Управляющие сигналы имеют следующее назначение:

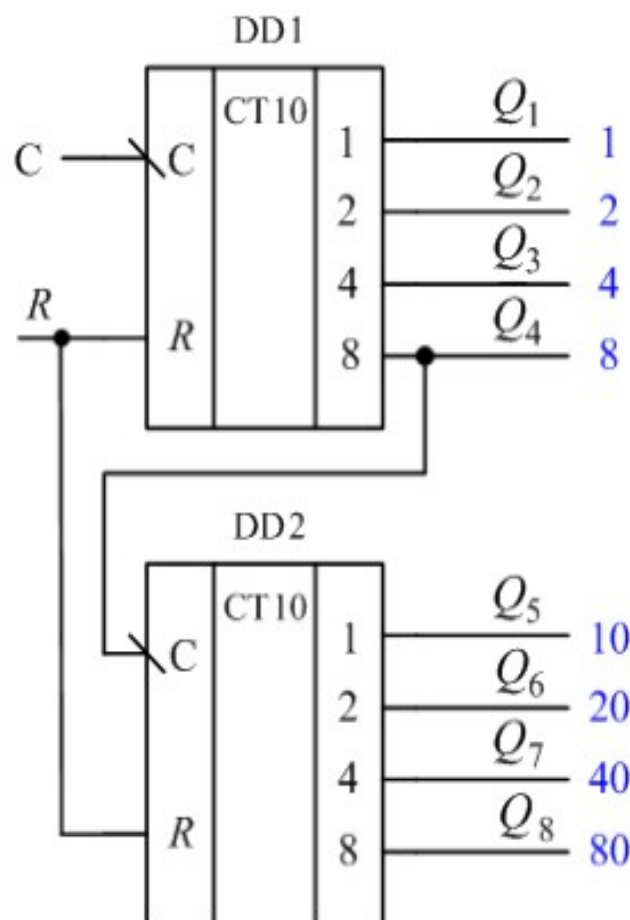
- $W = 0$  служит для ввода кода по входам  $D_0...D_3$  при предустановке;
- $V1=0$  служит для прерывания счета с сохранением кода;
- $V2=0$  разрешает выдачу импульса переноса по выходу  $P$ .

Если дополнить счетчик схемой показанной пунктиром на рис. а, то получится программируемый делитель частоты с коэффициентом деления  $K_{\text{дел}}=10-K$ , где  $K$  – число в двоично-десятичном коде, записанное через входы предустановки. При этом частота выходных импульсов

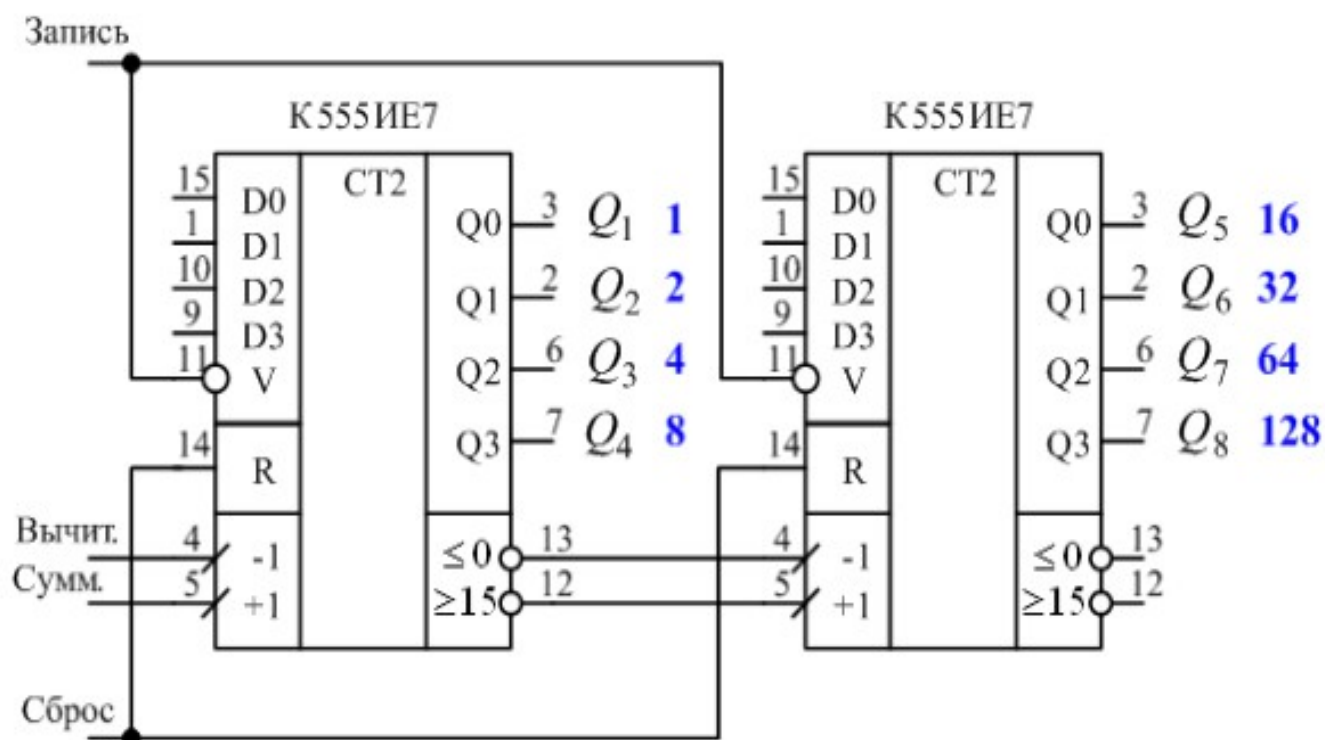
$$f_{\text{вых}} = f_{\text{вх}} / K_{\text{дел}}.$$

На рис. б показана ЦИС реверсивного счетчика. При  $U/D=1$  счетчик работает в режиме сложения, а при  $U/D=0$  – в режиме вычитания. Сигнал  $V=1$  служит для остановки счета и может быть использован как вход переноса при последовательном включении счетчиков.

# НАРАЩИВАНИЕ РАЗРЯДНОСТИ СЧЕТЧИКОВ



Функциональная схема  
асинхронного суммирующего  
счетчика с  $k_{сч} = 100$



Наращивание разрядности реверсивного  
двоичного счетчика

# РЕГИСТРЫ

*Регистрами* называют устройства для приема, хранения и выдачи данных, представленных в цифровом коде. Каждый разряд цифрового кода хранится в отдельном триггере регистра. Таким образом, регистр представляет собой разновидность устройства памяти. В общем случае регистр хранит  $n$ -разрядный код.

Классификация регистров может быть произведена по ряду признаков. Основным признаком является способ приема и выдачи данных. При этом различают:

- параллельные регистры с приемом и выдачей данных в параллельном коде;
- последовательные регистры с приемом или выдачей данных в последовательном коде;
- параллельно-последовательные.

*Состояние  $n$ -разрядного регистра* определяется состояниями  $n$  триггеров и отображается  $n$ -разрядным двоичным словом  $Y = y_n \dots y_1$ .

# РЕГИСТРЫ

Кроме основных функций приема, хранения и выдачи данных регистры могут выполнять функции преобразования данных или логические операции. По этому признаку различают:

- сдвиговые регистры;
- регистры с преобразованием параллельного кода в последовательный или наоборот;
- регистры с поразрядным логическим умножением или сложением;
- регистры последовательного приближения для построения аналого-цифровых преобразователей.

По количеству каналов, по которым поступают данные, различают:

- однофазные регистры с одним каналом;
- парафазные регистры с вводом данных по прямому и инверсному каналам.

По количеству тактов управления для записи данных различают:

- однотоковые регистры;
- двухтактные регистры;
- многотоковые регистры.

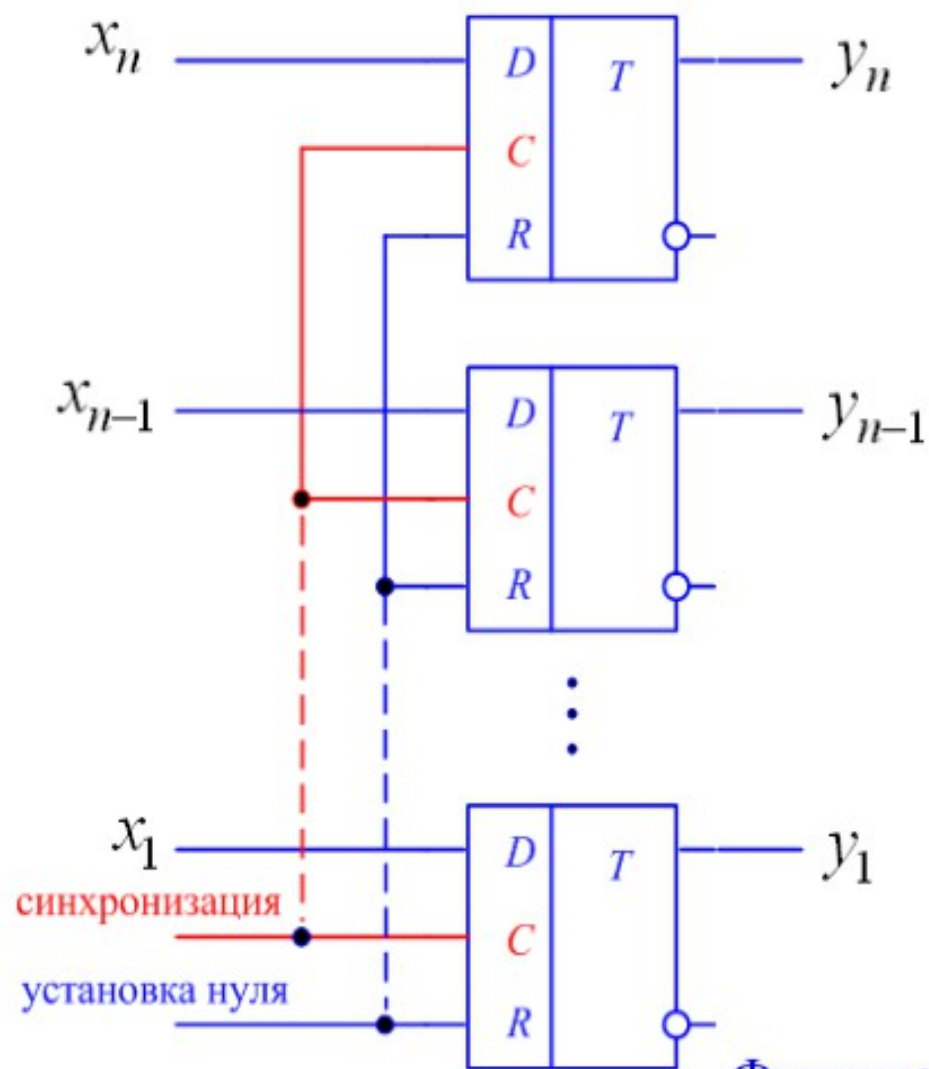
По управлению записью данных различают:

- асинхронные регистры;
- синхронные регистры.



# РЕГИСТРЫ

По назначению регистры подразделяются на *регистры памяти* и *регистры сдвига*.

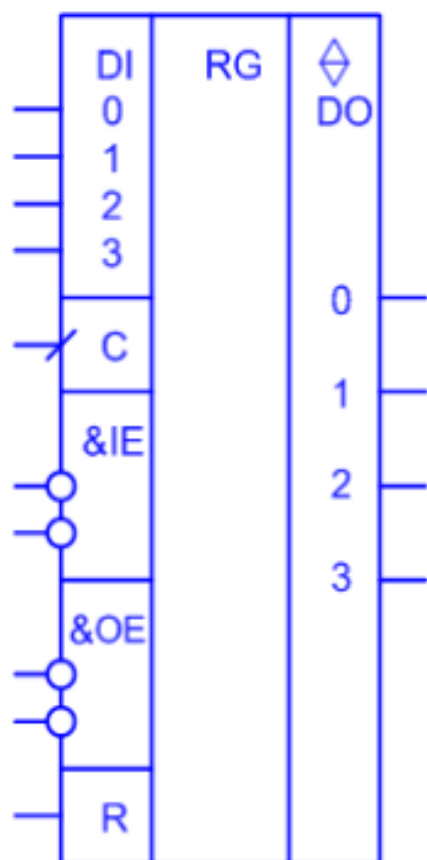


*Регистры памяти* предназначены для хранения цифровой информации небольшого объема.

Регистры памяти представляют собой наборы синхронных триггеров с независимыми информационными и объединенными **в общую шину синхронизации** тактовыми входами. Каждый из триггеров обеспечивает хранение одного разряда двоичного числа. В качестве элементов памяти регистра в основном используются синхронные  $D$ -триггеры.

Функциональная схема  $n$ -разрядного регистра памяти.

# РЕГИСТРЫ

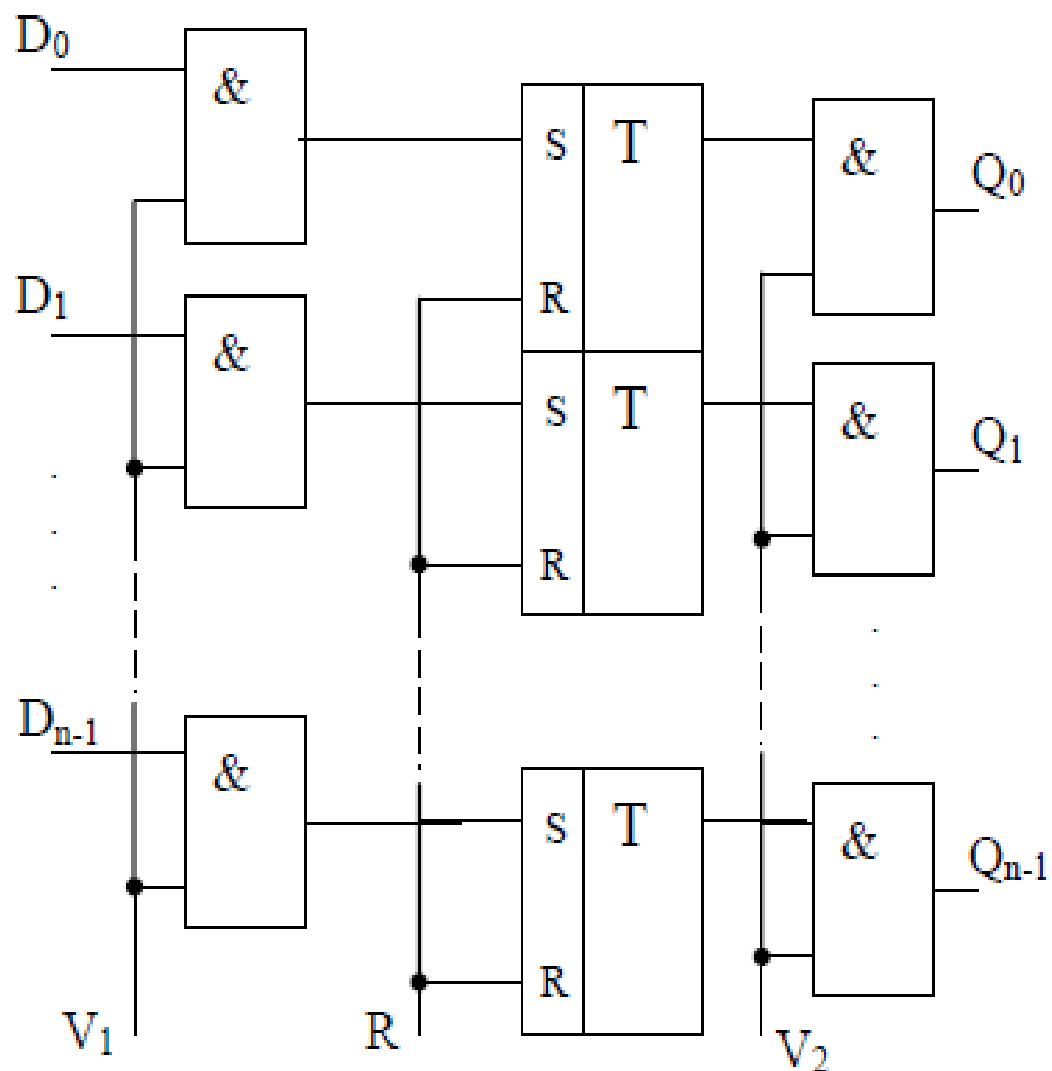


Условное обозначение  
синхронного  
четырёхразрядного  
регистра памяти

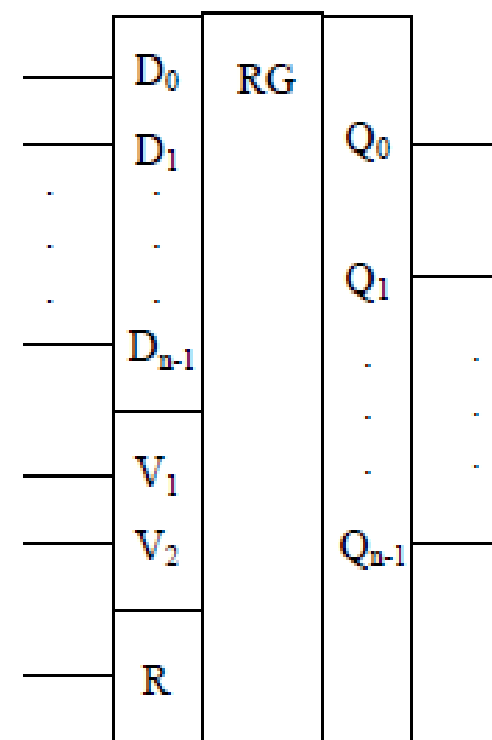
Регистр памяти содержит четырехразрядную входную шину **DI**, четырехразрядную выходную шину **DO**, динамический вход синхронизации по фронту тактовых импульсов **C**, инверсные входы **&IE** (входы “разрешение записи”), объединенные логикой И, инверсные входы **&OE** (входы “разрешение считывания”), объединенные логикой И, а также прямой вход сброса **R**. При совпадении на входах **&IE** уровней логического нуля (разрешение записи) по фронту импульса синхронизации в регистр записывается четырехразрядное слово со входной шины **DI**, которое считывается с выходной шины **DO** при совпадении уровней логического нуля на входах **&OE** (разрешение считывания). Если хотя бы на одном из входов **&OE** присутствует уровень логической единицы, микросхема находится в состоянии разомкнутого выхода. При подаче на вход **R** уровня логической единицы происходит сброс регистра (все триггеры регистра устанавливаются в нулевое состояние) независимо от сигналов на остальных входах.

Триггеры регистра могут записывать по уровню, переднему и заднему фронту.

# РЕГИСТРЫ



a)



б)

Регистр с асинхронным управлением записью



# РЕГИСТРЫ

*Регистры сдвига* помимо хранения обеспечивают выполнение операции сдвига двоичной информации.

*Сущность сдвига* состоит в том, что с приходом каждого тактового импульса происходит перезапись содержимого триггера каждого разряда в соседний разряд без изменения порядка следования двоичных цифр.

*По направлению сдвига различают однонаправленные регистры*, которые осуществляют сдвиг информации вправо (регистры прямого сдвига, регистры со сдвигом вправо) или влево (регистры обратного сдвига, регистры со сдвигом влево), и *реверсивные регистры* (допускают сдвиг в обоих направлениях).

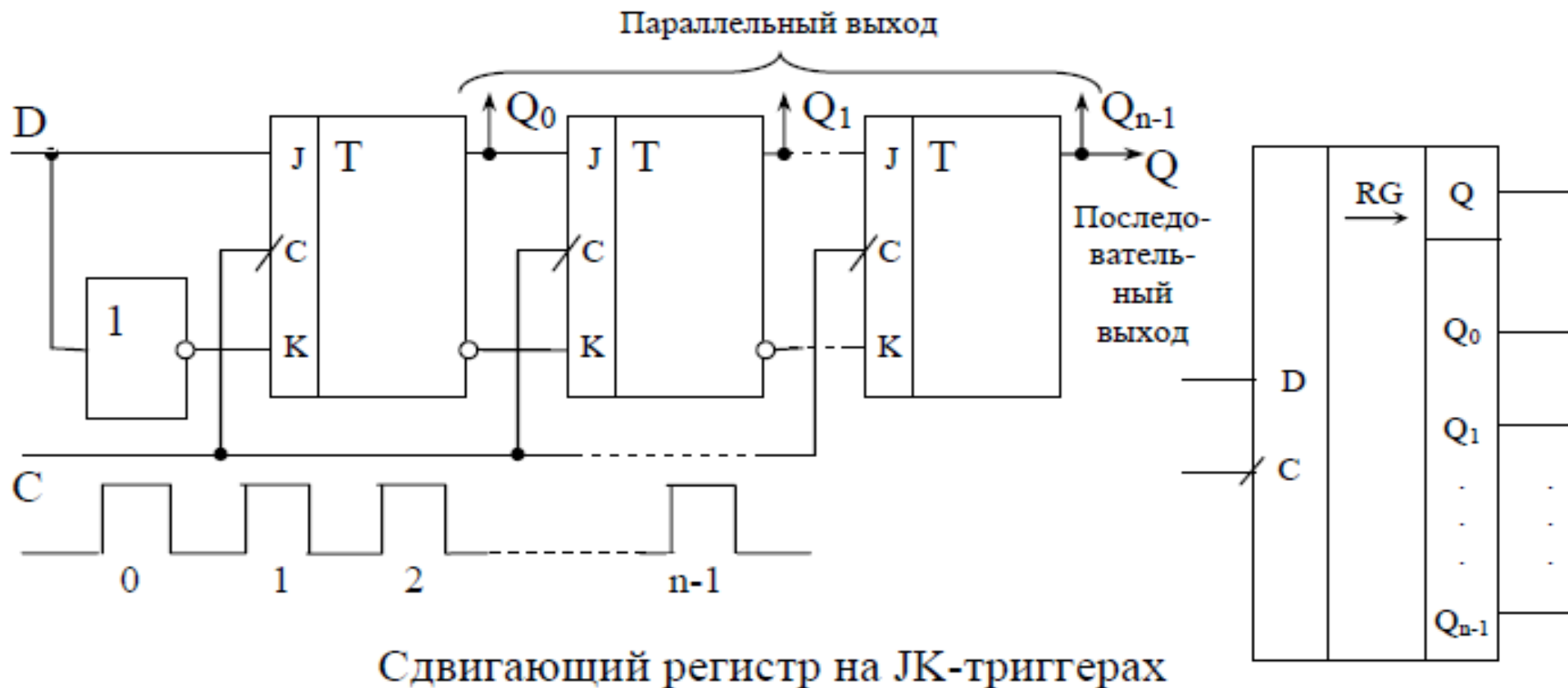
Работу регистра сдвига можно наблюдать, например, в микрокалькуляторах, где при наборе каждой цифры на клавиатуре числа на индикаторе сдвигаются влево. Например, для набора числа 41 нужно сделать следующее. Сначала нажимается и отпускается клавиша с цифрой 4. Цифра 4 появляется в крайней правой позиции на индикаторе. Затем нажимается и отпускается клавиша с цифрой 1. В результате цифра 4 на индикаторе сдвигается на одну позицию влево, а на ее месте в крайнем правом положении появляется цифра 1 – на индикаторе высвечивается число 41.



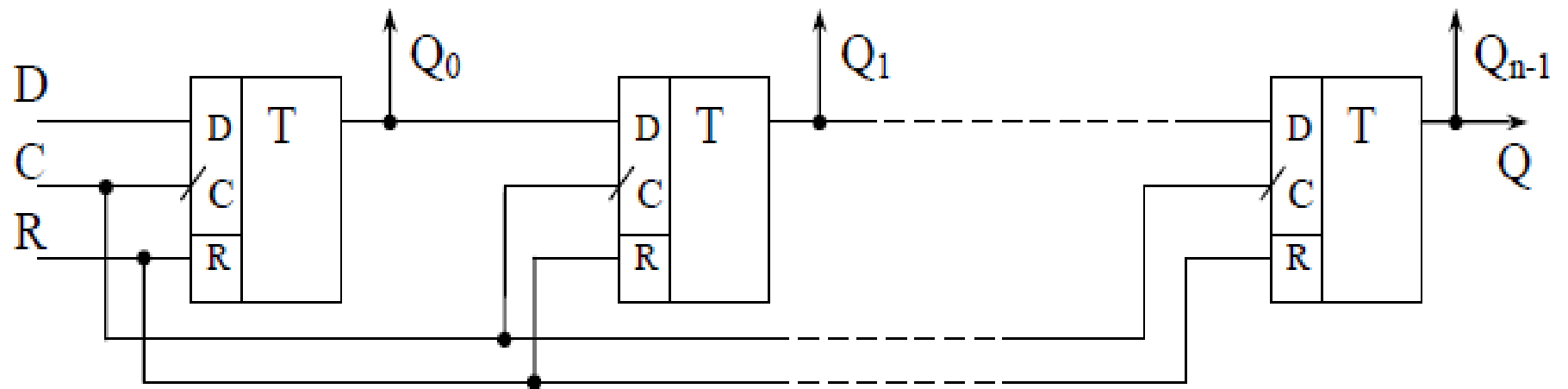
# РЕГИСТРЫ

Сдвигающие регистры применяются для выполнения операций:

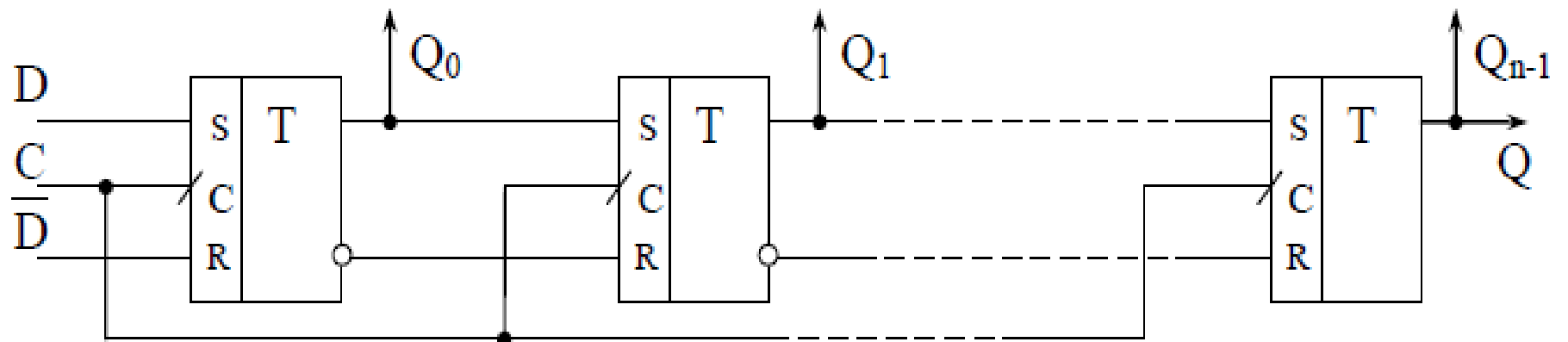
- поразрядного сдвига при реализации вычислительных алгоритмов над цифровыми кодами;
- преобразования последовательного кода в параллельный и наоборот;
- запоминания данных с последовательным доступом;
- временной задержки.



# РЕГИСТРЫ



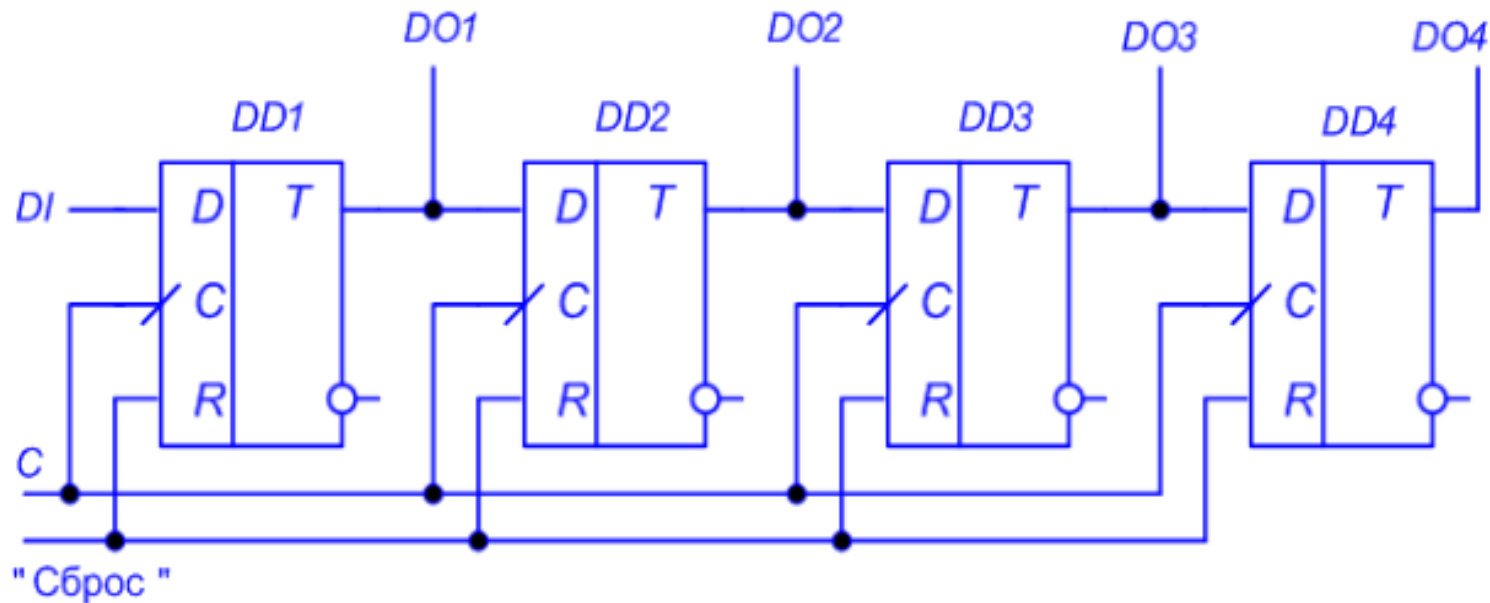
a



б

Варианты построения сдвигающих регистров

# РЕГИСТРЫ

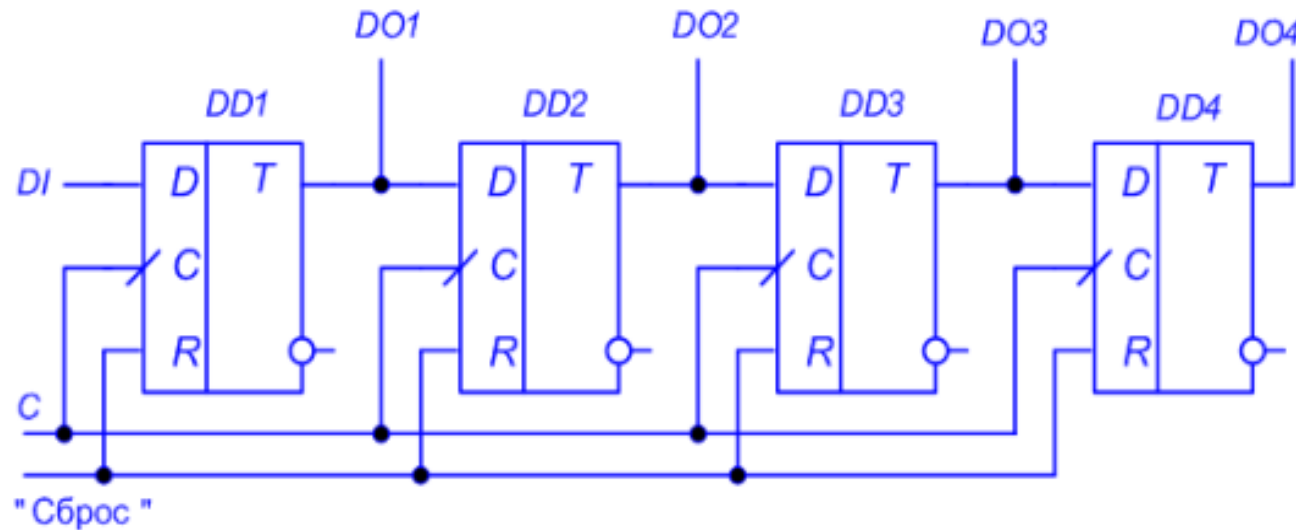


Функциональная схема четырехразрядного последовательного регистра сдвига вправо (термин *«последовательный»* отражает тот факт, что в этот регистр *данные вводятся поразрядно*).

Регистр представляет собой последовательную цепь *D*-триггеров с динамическим управлением по фронту тактовых импульсов. Тактовые импульсы (импульсы сдвига) поступают на все триггеры одновременно. Информационным входом *DI* регистра является вход триггера *DD1*. При подаче тактовых импульсов происходит последовательная запись информации со входа *DI* и ее сдвиг в сторону возрастания номеров триггеров (сдвиг вправо). Для записи в регистр *четырёхразрядного* слова необходимо *четыре* тактовых импульса.

# РЕГИСТРЫ

Допустим, что в регистр последовательно вводится, начиная с младшего разряда, двоичный код **1101**. Предварительный сброс регистра производится подачей сигнала логической единицы на вход “Сброс” (все триггеры устанавливаются в нулевое состояние).

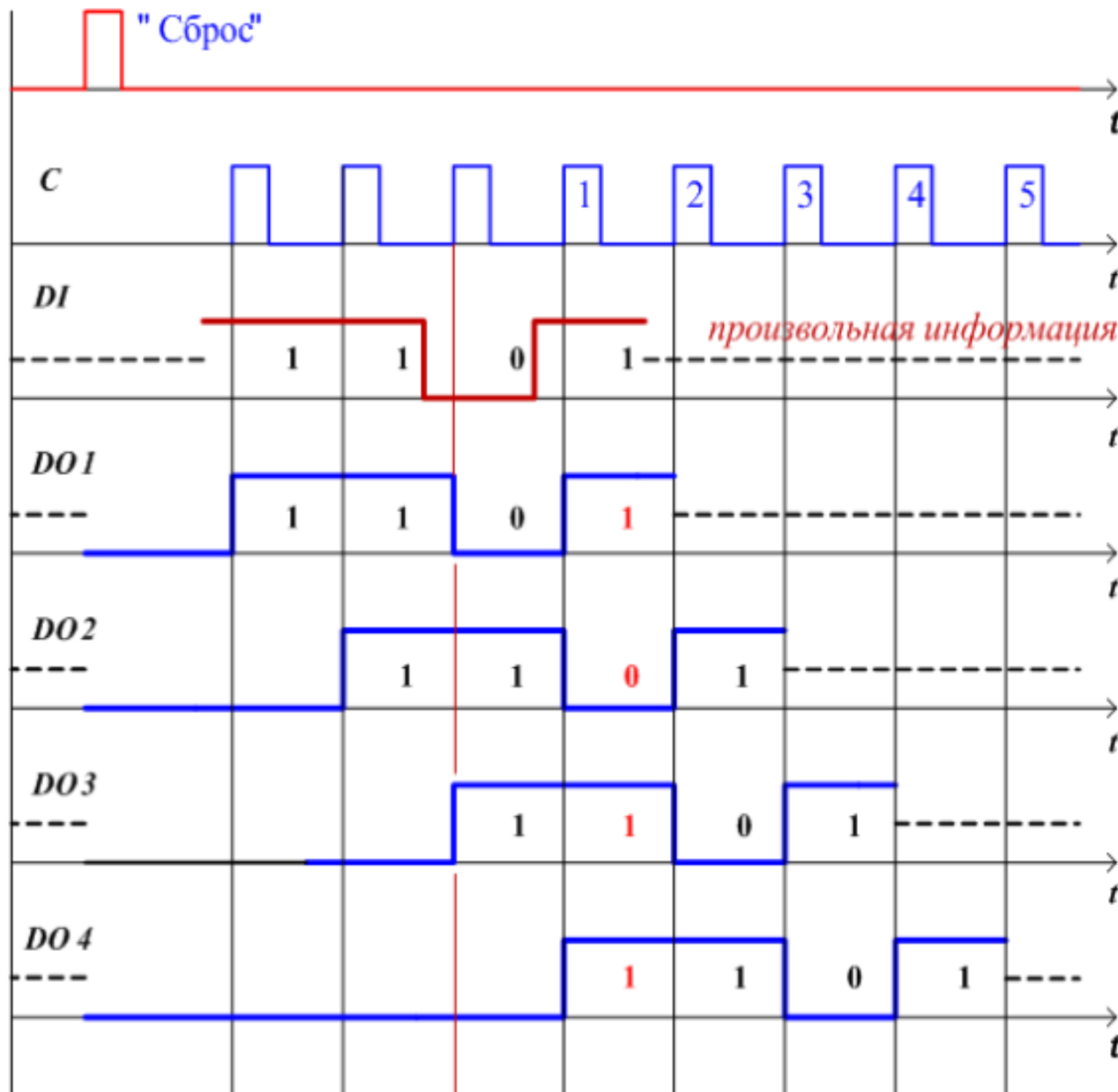


С первым тактовым импульсом в триггер *DD1* записывается единица младшего разряда входного слова. Со следующим тактовым импульсом эта единица будет сдвинута в триггер *DD2*, а в триггер *DD1* одновременно поступит информация следующего разряда входного слова. Аналогично происходит дальнейший сдвиг информации в триггеры *DD3* и *DD4*.

После четырех тактовых импульсов код на выходах *DO1-DO4* соответствует входному коду и может быть **параллельно считан** внешним устройством.

**Последовательное считывание** информации из регистра осуществляется с выхода *DO4*, начиная с пятого тактового импульса.

# РЕГИСТРЫ



В регистр последовательно вводится двоичный код **1101**, начиная с младшего разряда.

**Импульс №1:** В триггер *DD1* записывается единица младшего разряда входного слова.

**Импульс №2:** Единица младшего разряда входного слова сдвигается в триггер *DD2*.

Одновременно в триггер *DD1* поступает информация следующего разряда (ноль) входного слова.

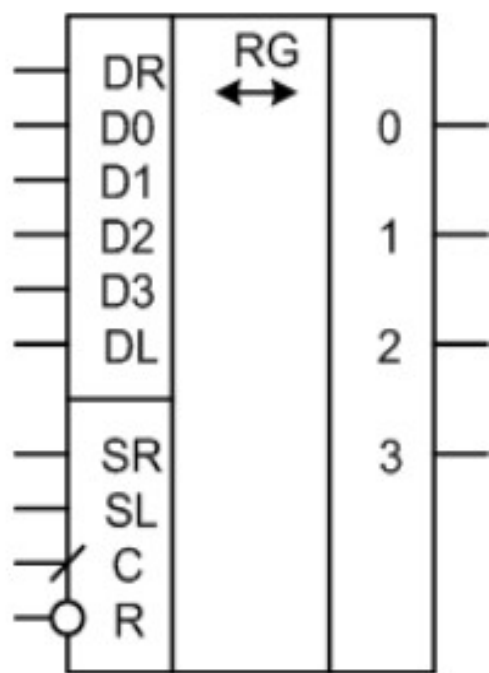
**Импульсы №3 и №4:** Происходит дальнейший сдвиг информации в триггеры *DD3* и *DD4*.

После тактовых импульсов №1 – №4 код на выходах *DO1-DO4* триггера соответствует входному коду.



# РЕГИСТРЫ

*Универсальные регистры* в зависимости от подачи управляющих сигналов могут выполнять функции регистров памяти, однонаправленных и реверсивных регистров сдвига.



Условное обозначение  
микросхемы универсального  
четырёхразрядного регистра.

Микросхема универсального четырехразрядного регистра содержит:

*входы D0-D3* для подачи информации при параллельной записи;

*вход DR* для подачи информации при последовательной записи и сдвиге вправо;

*вход DL* для подачи информации при последовательной записи и сдвиге влево;

*входы SR, SL* для управления режимом работы;

*тактовый вход C;*

*асинхронный инверсный вход сброса R.*

# РЕГИСТРЫ

При подаче сигнала логического нуля на вход  $R$  происходит установка всех триггеров регистра в нулевое состояние независимо от сигналов на остальных входах регистра.

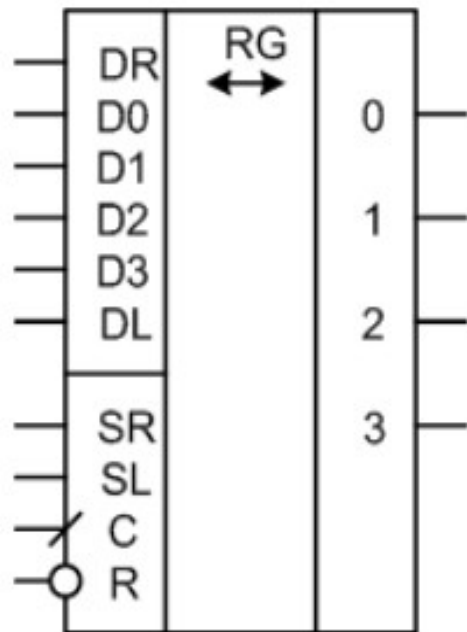
При наличии на входе  $R$  сигнала логической единицы режим работы регистра определяется комбинацией сигналов на управляющих входах  $SR$ ,  $SL$ :

1) Комбинации управляющих сигналов  $SR=0$ ,  $SL=0$  соответствует режиму *хранения*.

2) При  $SR=1$ ,  $SL=0$  по фронту тактовых импульсов происходит последовательная запись информации со входа  $DR$  и *сдвиг* ее *вправо*.

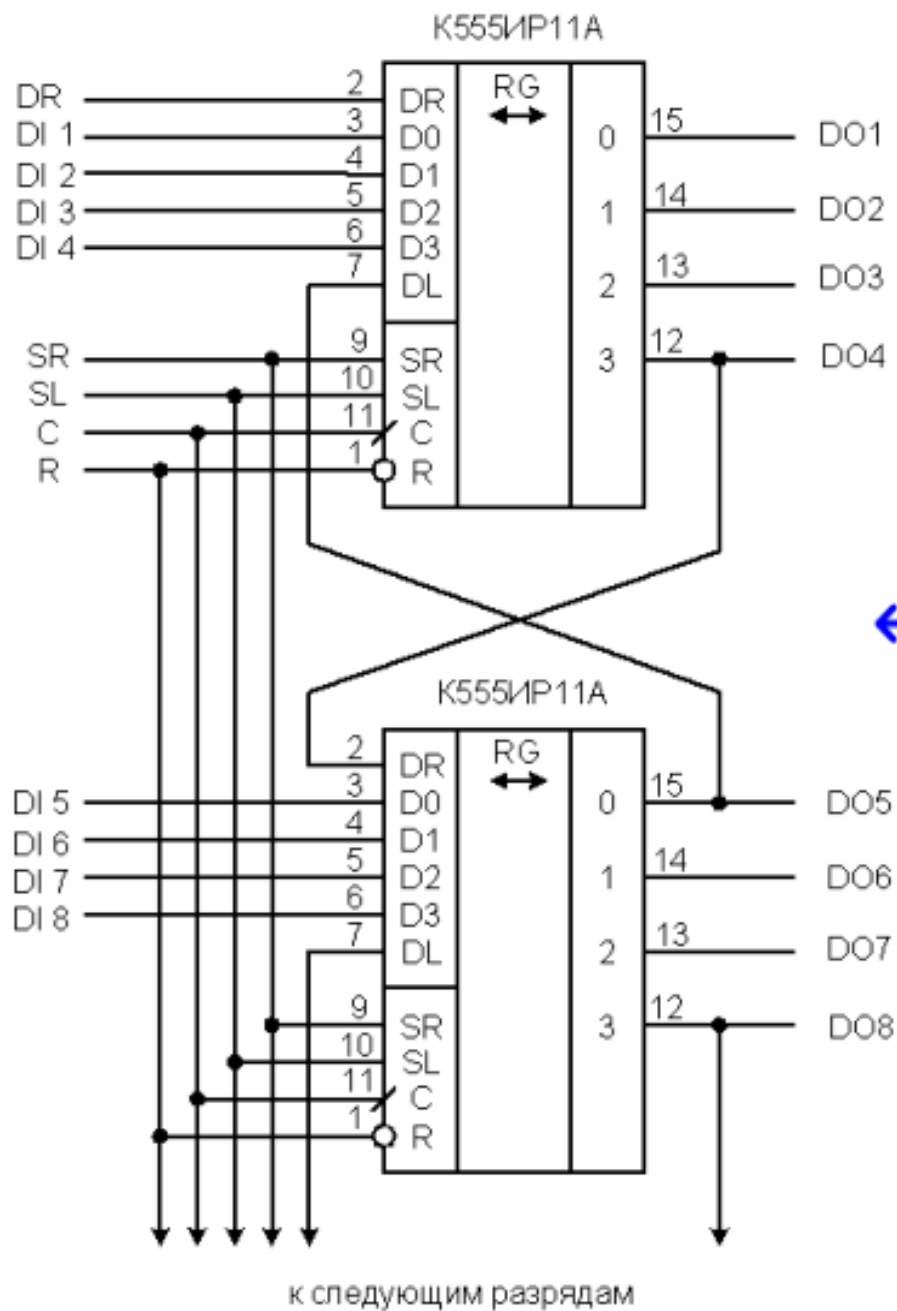
3) При  $SR=0$ ,  $SL=1$  по фронту тактовых импульсов происходит последовательная запись информации со входа  $DL$  и *сдвиг* ее *влево*.

4) При комбинации  $SR=1$ ,  $SL=1$  по фронту тактовых импульсов осуществляется параллельная *запись информации* со входов  $D0$ - $D3$ .



Условное обозначение микросхемы универсального четырехразрядного регистра.

# РЕГИСТРЫ



*Наращивание разрядности универсального регистра.*

# Арифметические операции, выполняемые с помощью операторов сдвига

С помощью регистров можно выполнять операции умножения на два и деления на два.

Умножения хранящегося числа на 2 осуществляется путем сдвига влево и записи «0» в младший разряд.

Целочисленное деление хранящегося числа на 2 осуществляется путем сдвига вправо и записи «0» в старший разряд.

## *Пример*

$00011 = 3_{10}$  — сдвигаем влево

$00110 = 6_{10}$  — сдвигаем влево

$01100 = 12_{10}$  — таким образом, умножили сначала на 2, потом еще раз на 2.

$01011 = 11_{10}$  — сдвигаем вправо

$00101 = 5_{10}$  — сдвигаем вправо

$00010 = 2_{10}$  — сдвигаем еще раз вправо

$00001 = 1_{10}$  — таким образом, число 11 разделили на 2, получилось 5, затем 5 поделили на 2, получилось 2, и 2 поделили на 2, получилось 1.

Регистры сдвига применяют не только в качестве запоминающих устройств, преобразователей последовательного кода в параллельный, устройств задержки, но и счетчиков импульсов. Применение регистров в качестве счетчиков очень неэкономично, так как модуль счета  $K_{сч} = n$ , в то время, как для двоичных счетчиков  $K_{сч} = 2^n$ .

## КОЛЬЦЕВЫЕ СЧЕТЧИКИ

Распространенной разновидностью параллельных счетчиков являются кольцевые счетчики, построенные на базе регистров сдвига. Простейшая схема кольцевого счетчика получается при замыкании прямого выхода регистра сдвига с его входом. В таком случае единица, записанная в регистр на первом такте, с выхода  $Q_n$  счетчика снова попадает на его вход и далее весь цикл повторяется. Модуль счета такого счетчика имеет то же значение, что и регистр сдвига, т. е.  $K_{сч} = n$ .

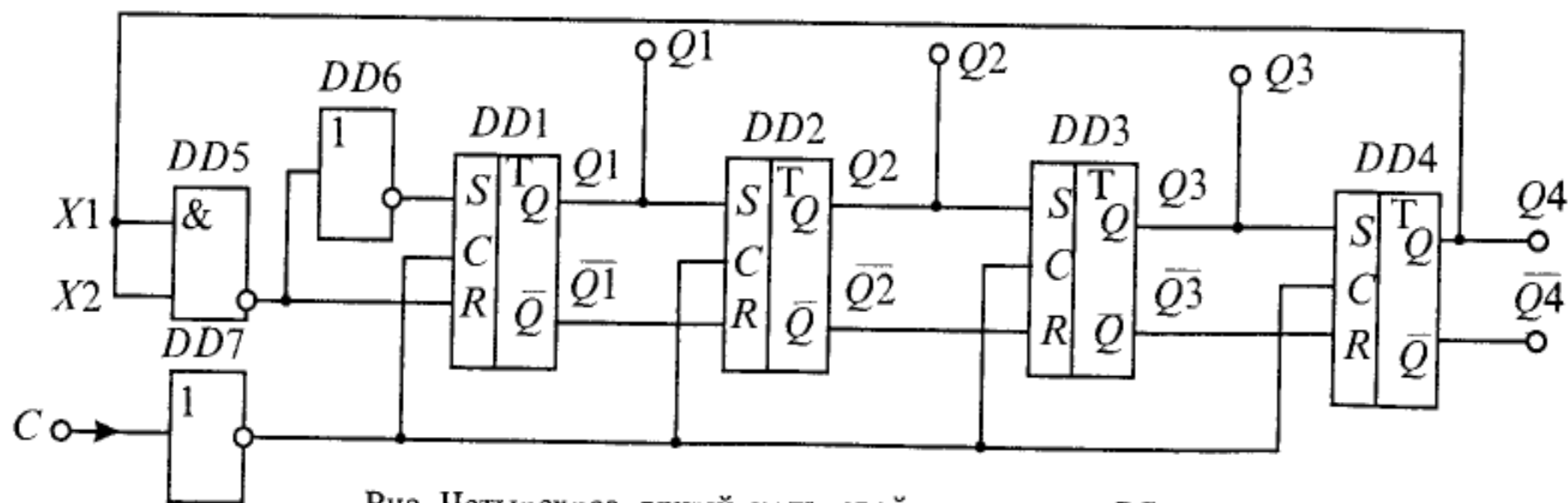


Рис. Четырехразрядный кольцевой счетчик на RS-триггерах



Для увеличения модуля счета можно или увеличивать количество триггеров в кольце, или включать счетчики последовательно. Так, например, счетчик на 10 импульсов ( $K_{сч}=10$ ) можно реализовать последовательным соединением одного счетного триггера и кольцевого счетчика из пяти триггеров.

Основным недостатком кольцевых счетчиков является их низкая помехозащищенность. Например, если под действием помехи исчезнет записанная в счетчик единица, то все триггеры окажутся в нулевом состоянии и счетчик работать не сможет. Для устранения подобных сбоев используются различные способы коррекции состояния счетчика. На рис. показана схема кольцевого счетчика с автоматической коррекцией состояния. В этой схеме независимо от того, в каком состоянии после включения окажутся триггеры, после четырех тактовых импульсов на входе  $C$  установится требуемое входное состояние (1 0 0 0).

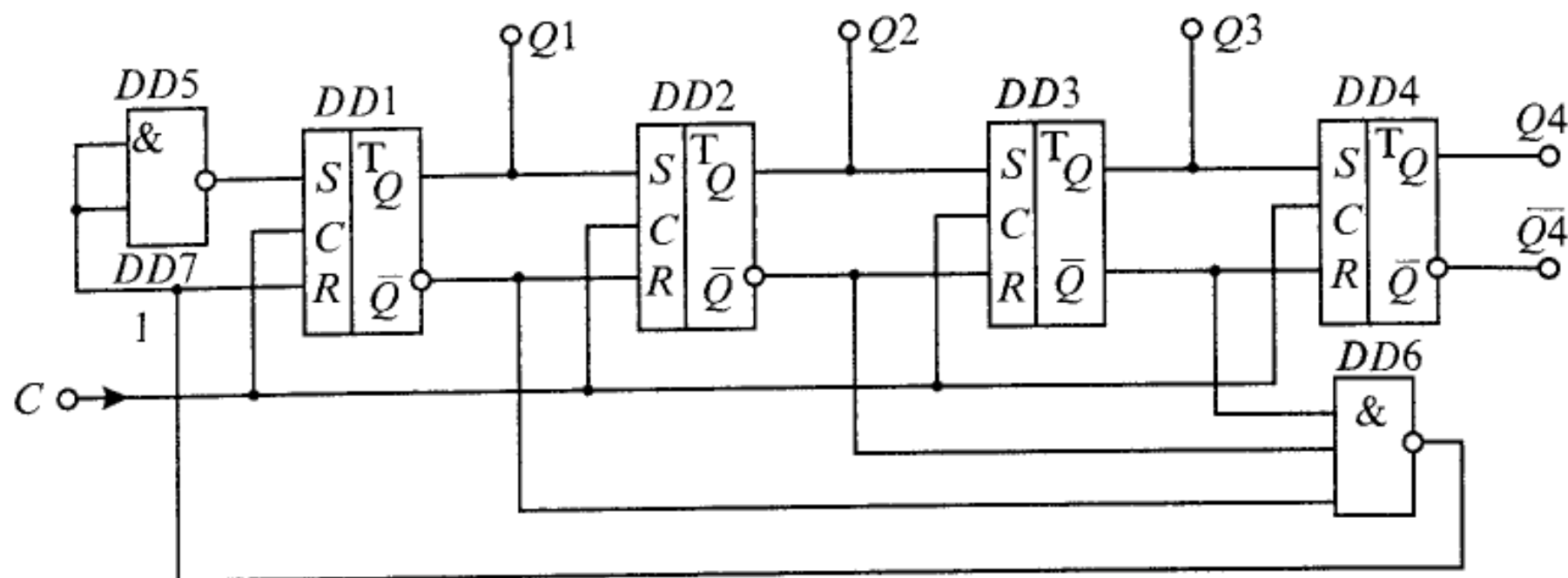
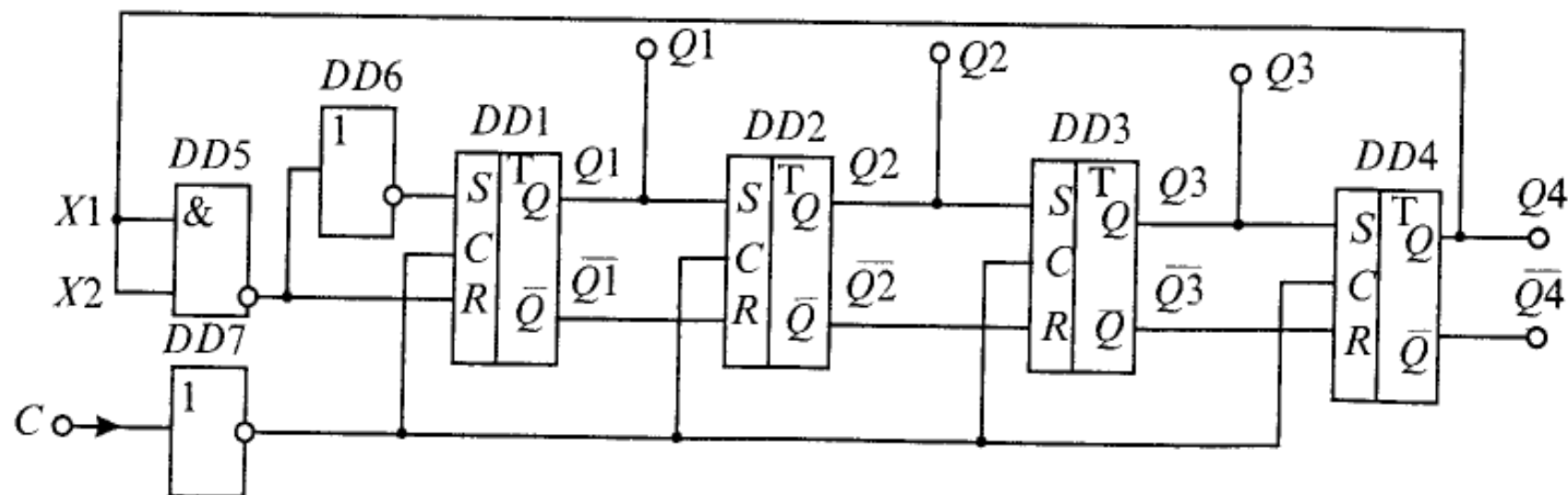


Рис. Схема кольцевого счетчика с автоматической коррекцией начального состояния.





Четырехразрядный кольцевой счетчик на RS-триггерах

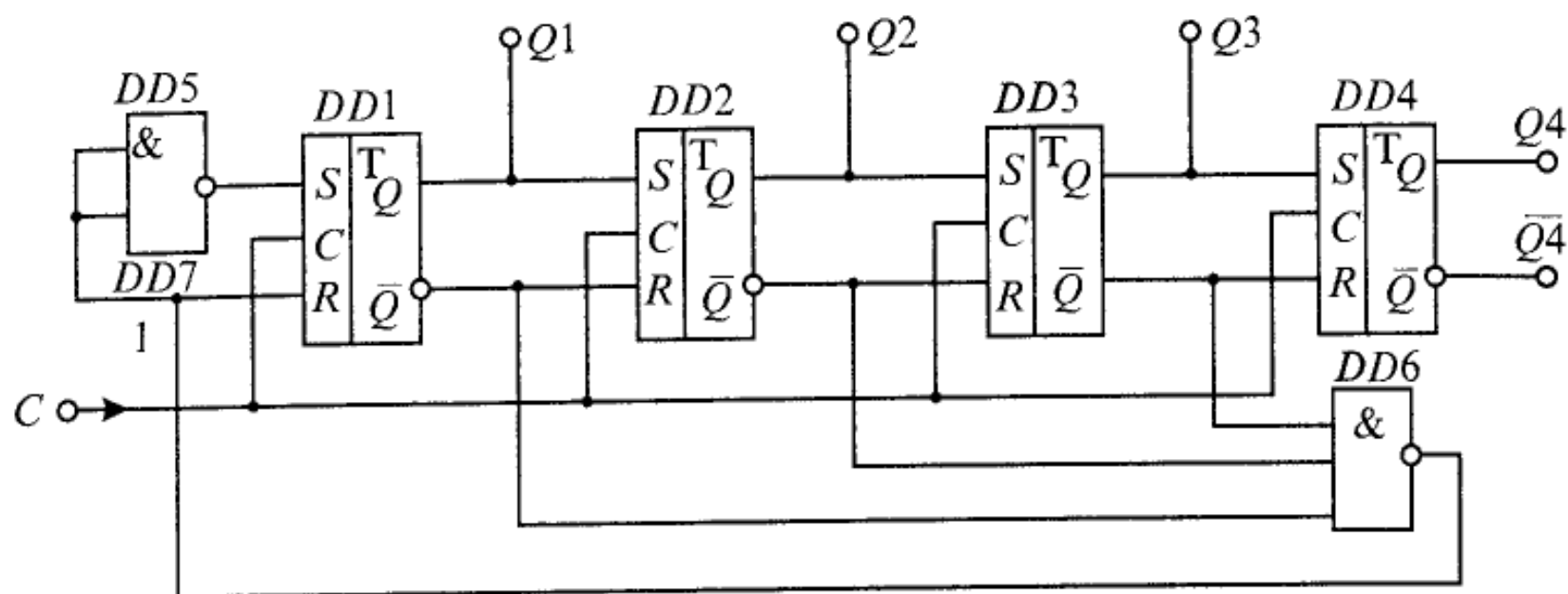
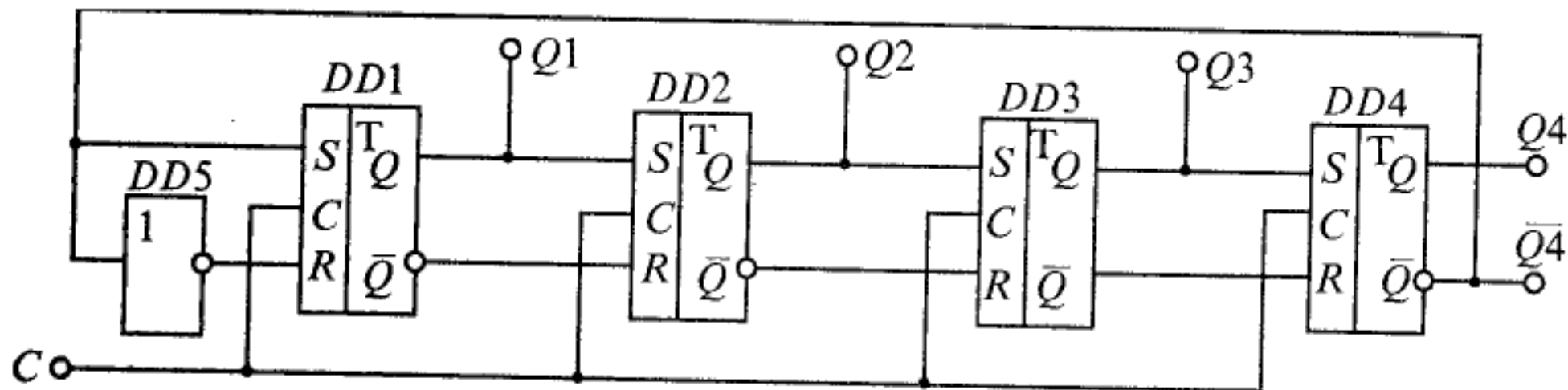


Схема кольцевого счетчика с автоматической коррекцией начального состояния.

# КОЛЬЦЕВЫЕ СЧЕТЧИКИ

Разновидностью кольцевых счетчиков являются **счетчики Джонсона**. В этих счетчиках вход регистра соединен не с выходом  $Q_n$ , а с инверсным выходом  $\bar{Q}_n$ . В результате, когда на вход счетчика поступают тактовые импульсы, то вначале все разряды счетчика заполняются единицами, а затем – нулями. Ниже приведена схема четырехразрядного счетчика Джонсона и таблица с состояниями его выходов.



Состояние выходов четырехразрядного счетчика Джонсона

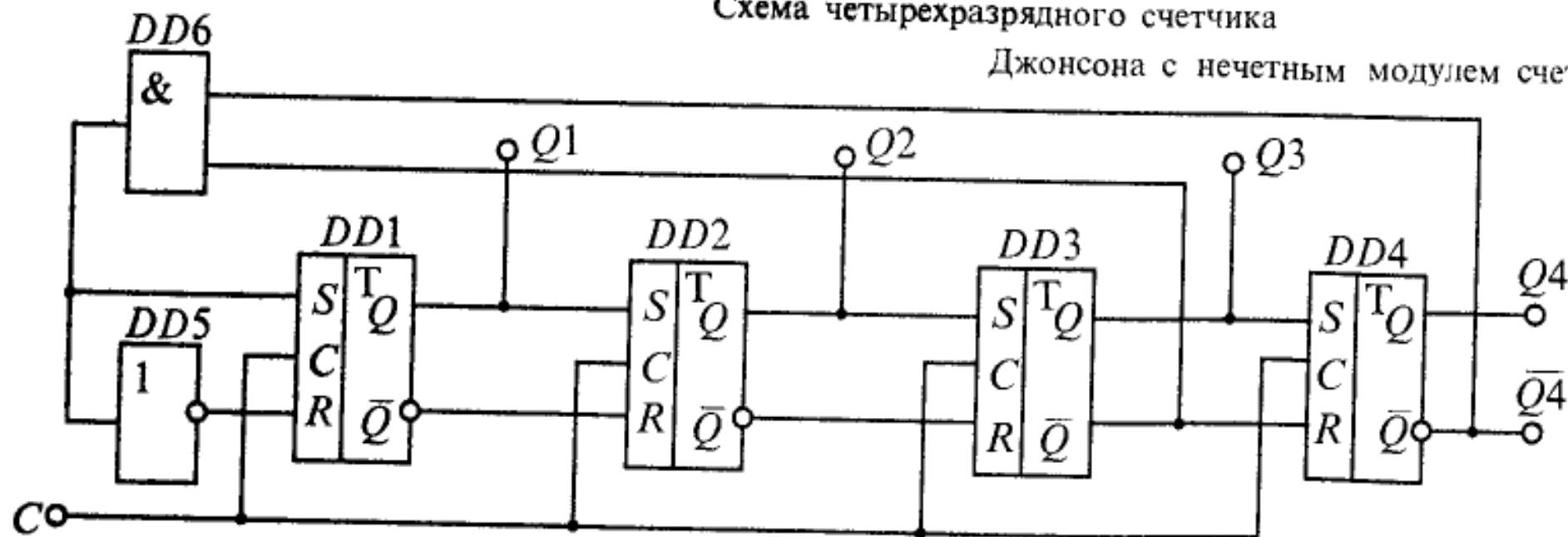
$n$	$Q_4$	$Q_3$	$Q_2$	$Q_1$	$n$	$Q_4$	$Q_3$	$Q_2$	$Q_1$
0	1	0	0	0	4	0	1	1	1
1	1	1	0	0	5	0	0	1	1
2	1	1	1	0	6	0	0	0	1
3	1	1	1	1	7	0	0	0	0

**Модуль счета** счетчика Джонсона  $K_{сч}=2n$ , т. е. в 2 раза больше модуля счета простого кольцевого счетчика. В счетчике Джонсона, как и в других кольцевых счетчиках, могут быть сбои, вызванные помехами. Для коррекции нарушений, вызванных сбоями, также используют способы, с помощью которых производится переход из любой запрещенной комбинации в одну из разрешенных.

Счетчики Джонсона широко **используются** в делителях частоты импульсов, генераторах случайных чисел, в устройствах памяти и др. На базе счетчика Джонсона можно легко реализовать счетчики с любым четным модулем счета. При необходимости иметь нечетное значение модуля счета можно на вход первого разряда подавать вместо сигнала  $\bar{Q}_n$  сигнал  $\bar{Q}_n \bar{Q}_{n-1}$ , как показано на рисунке ниже. При этом из набора выходных состояний счетчика Джонсона исключается одна кодовая комбинация, составленная из нулей.

### Схема четырехразрядного счетчика

Джонсона с нечетным модулем счета





# ДВОИЧНО-ДЕСЯТИЧНЫЕ СЧЕТЧИКИ

**Двоично-десятичные или декадные счетчики** могут быть реализованы на базе двоичных счетчиков при помощи взаимной связи между отдельными триггерами, входящими в счетчик. Ниже показана схема декадного счетчика, построенная на базе четырехразрядного двоичного счетчика. В этом счетчике взаимны соединения триггеров выполнены так, что первые девять счетных импульсов повторяют выходные сигналы триггеров для двоичного счетчика, а последний счетный импульс возвращает счетчик в исходное состояние.

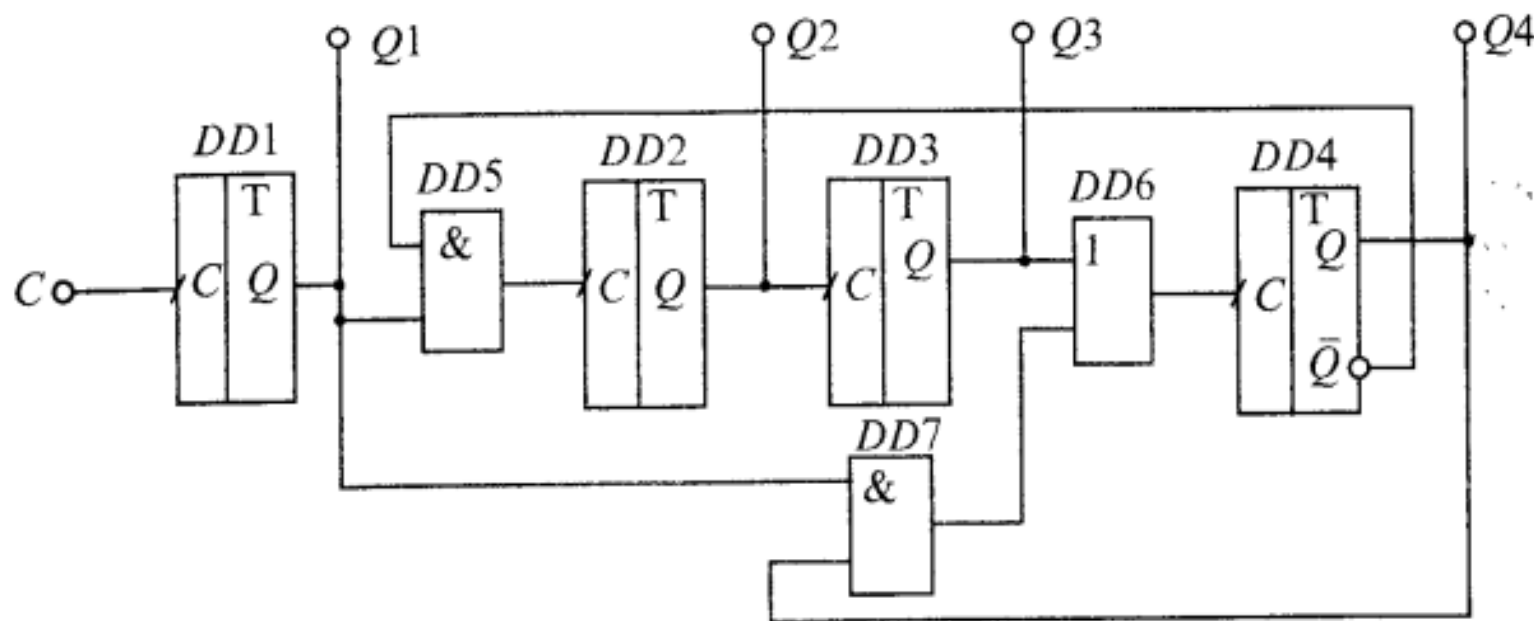


Рис. Схема декадного счетчика на счетных триггерах

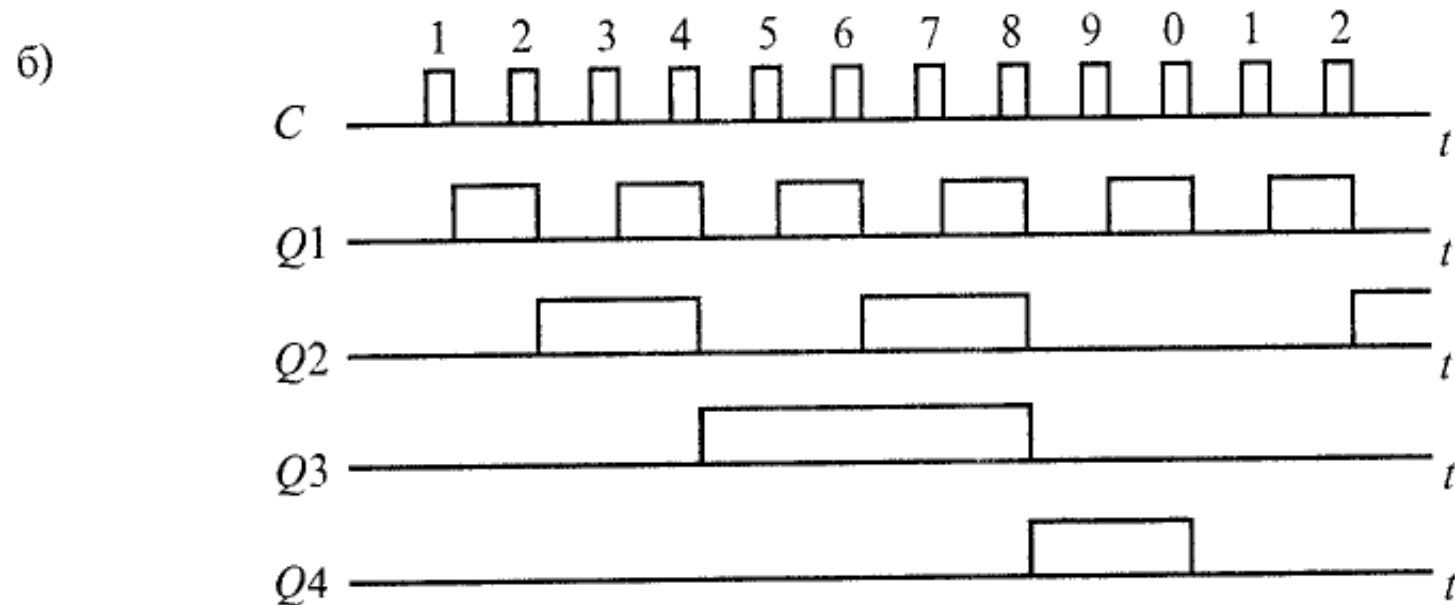
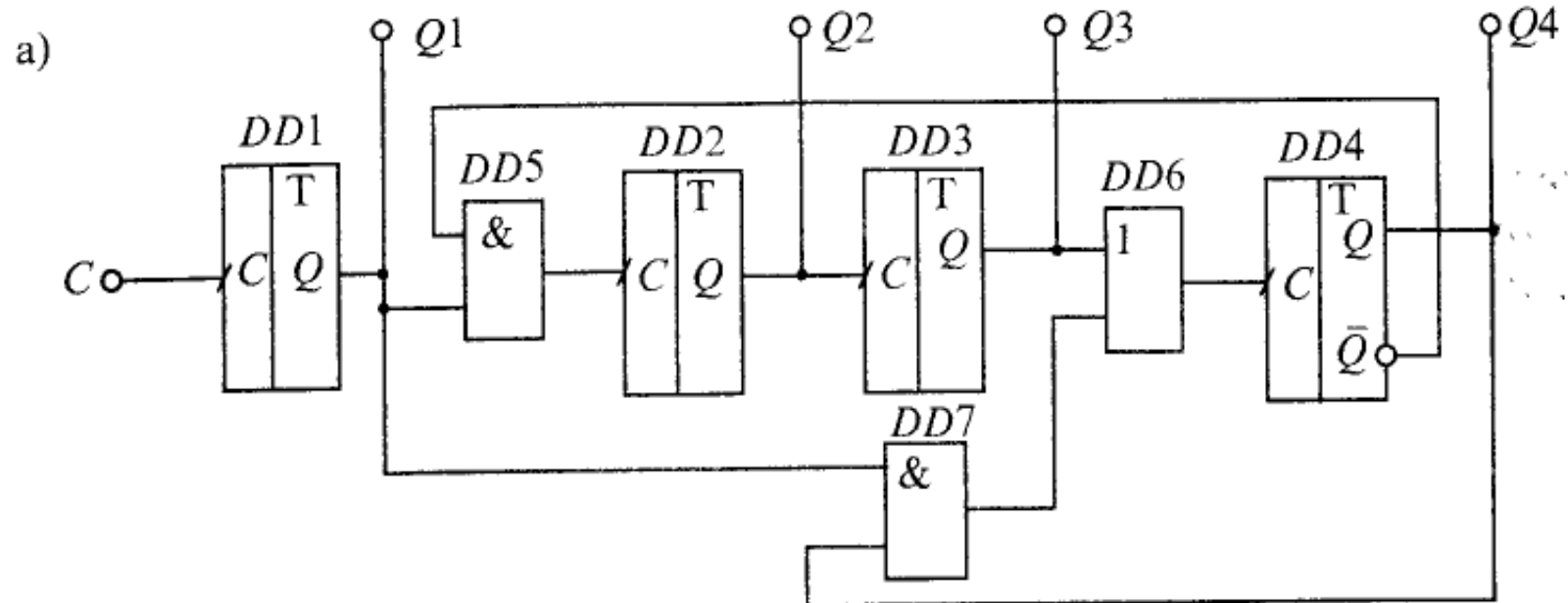


Схема декадного счетчика на счетных триггерах (а) и сигналы на его выходах (б)