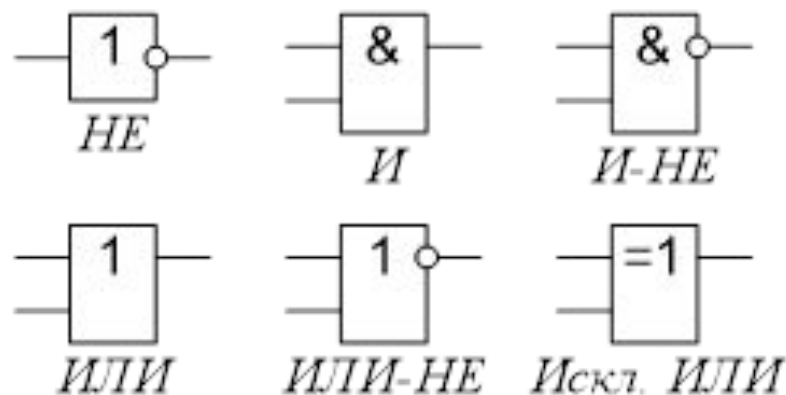


Логические устройства

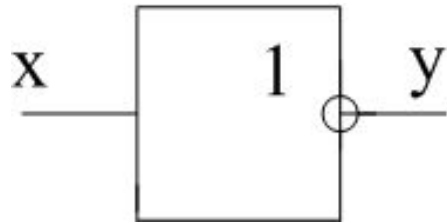
Простые логические элементы



1. НЕ: $y = \bar{x}$;
2. И: $y = x_1 x_2$;
3. И-НЕ: $y = \overline{x_1 x_2}$;
4. ИЛИ: $y = x_1 + x_2$;
5. ИЛИ-НЕ: $y = \overline{x_1 + x_2}$;
6. Исключающее ИЛИ: $y = x_1 \bar{x}_2 + \bar{x}_1 x_2$

УГО простейших логических элементов

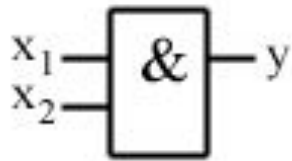
Логический элемент НЕ



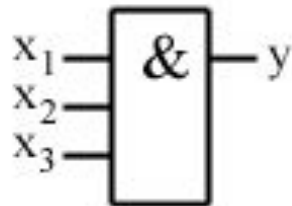
$$y = \overline{x}$$

x	y
0	1
1	0

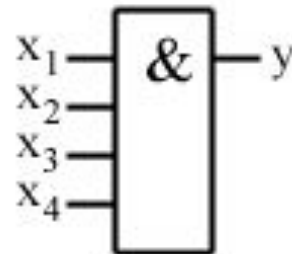
Логический элемент И



2И

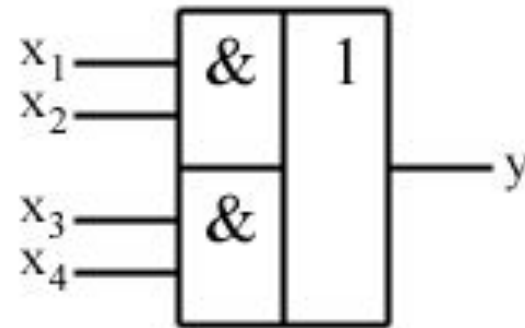
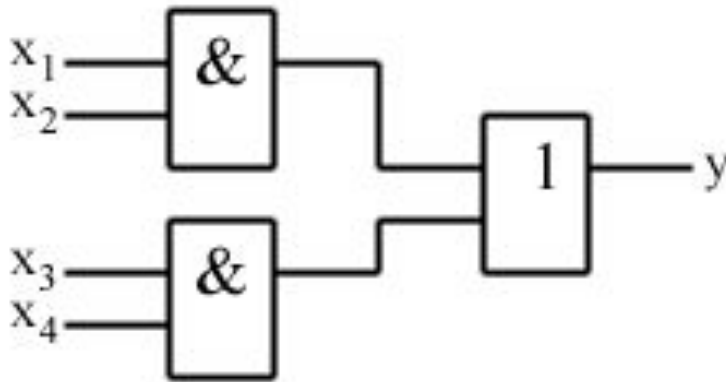


3И



4И

Сложные логические элементы



2-2И-2ИЛИ

$$y = x_1 x_2 + x_3 x_4$$

$X_{(10)}$	$X_{(2)}$	Y
0	0000	Y_1
1	0001	Y_2
2	0010	Y_3
3	0011	Y_4
4	0100	Y_5
5	0101	Y_6
6	0110	Y_7
7	0111	Y_8
8	1000	Y_9
9	1001	Y_{10}
10	1010	Y_{11}
11	1011	Y_{12}
12	1100	Y_{13}
13	1101	Y_{14}
14	1110	Y_{15}
15	1111	Y_{16}

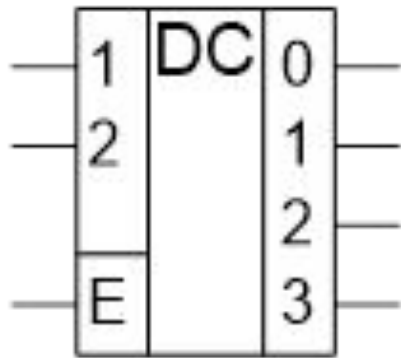
$x_3x_2x_1$	y
0 0 0	y_1
0 0 1	y_2
0 1 0	y_3
0 1 1	y_4
1 0 0	y_5
1 0 1	y_6
1 1 0	y_7
1 1 1	y_8

Комбинационные логические устройства

1. Дешифратор и шифратор.
2. Мультиплексор и демультиплексор.
3. Полусумматор и сумматор.
4. Арифметико-логическое устройство.

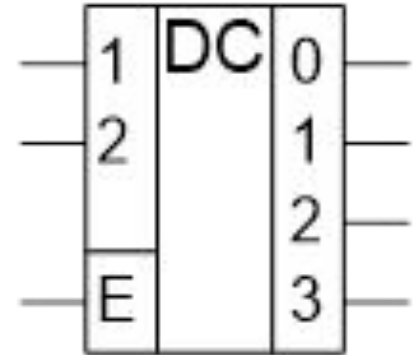
Дешифраторы

Дешифраторы осуществляют преобразование входного двоичного кода в унитарный код, т.е. код, включающий в себя одну логическую единицу, а остальные логические нули.



2×4

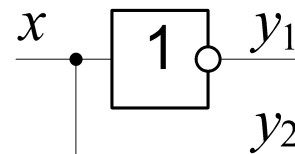
x_2	x_1	y_1	y_2	y_3	y
					4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Линейный дешифратор

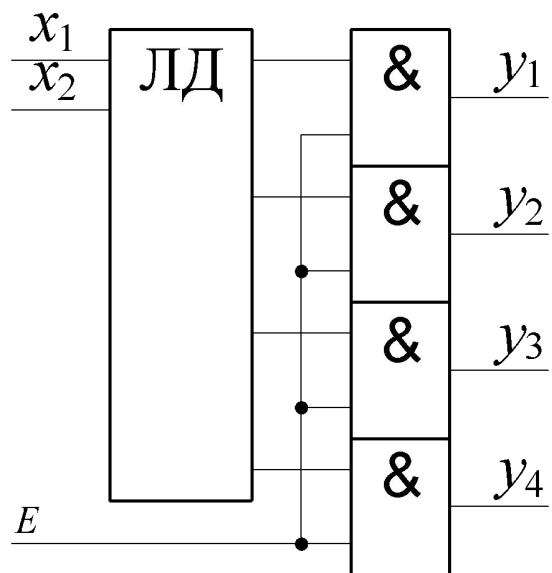
Схема такого дешифратора строится на основе его таблицы состояний.

1x2			
x	y ₁	y ₂	
0	1	0	y ₁ = $\bar{x}$
1	0	1	y ₂ = x

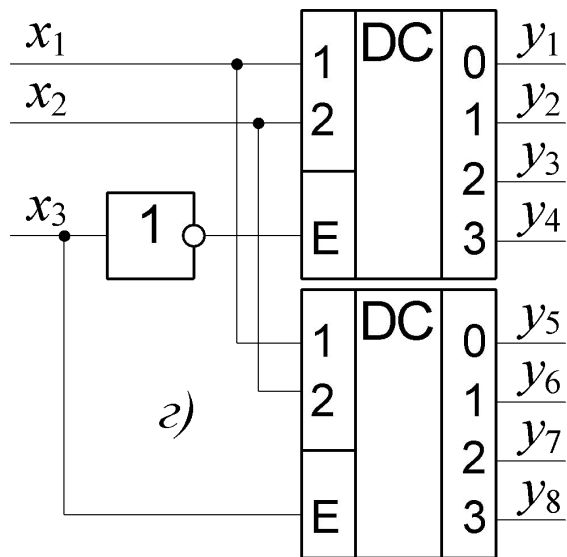


Назначение входа E :

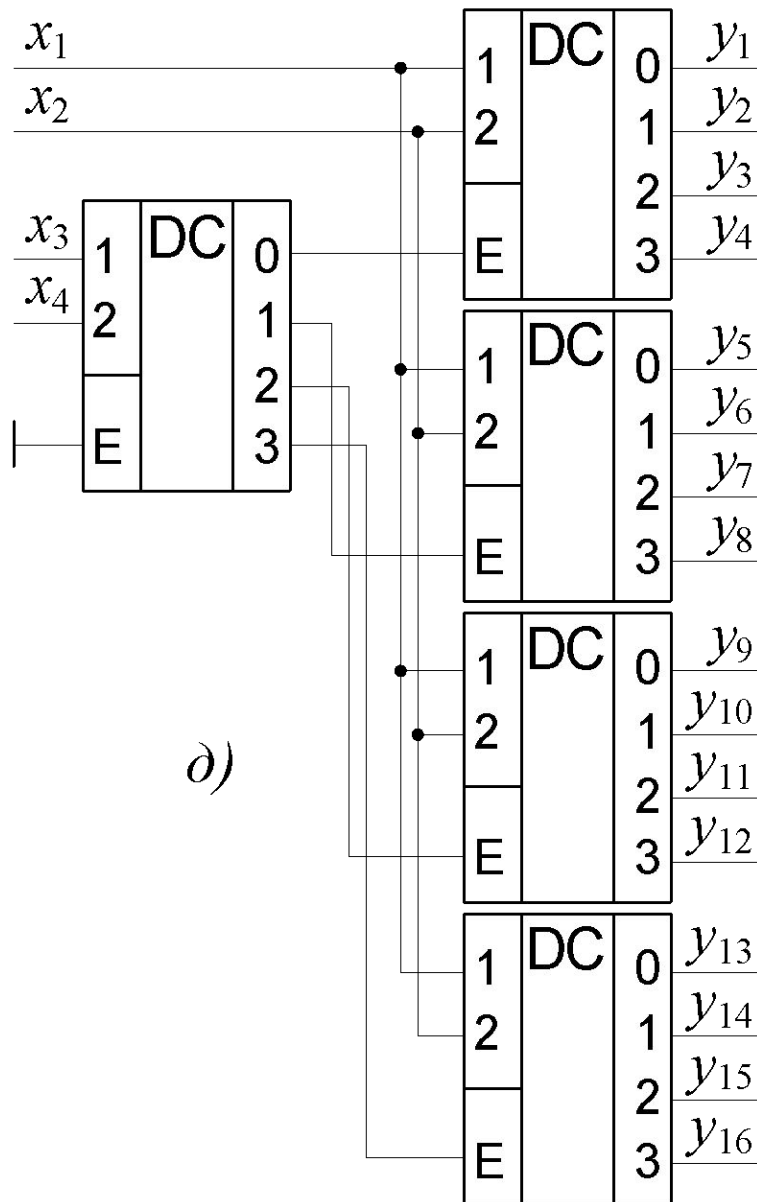
1. Разрешение работы дешифратора;
2. Нарастивание разрядности дешифратора;
3. Организация режима демультиплексора.



a)



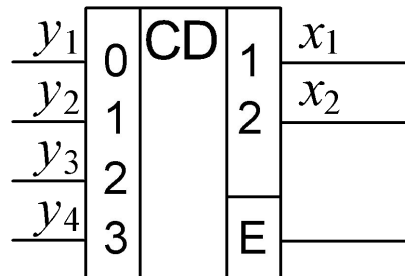
e)



d)

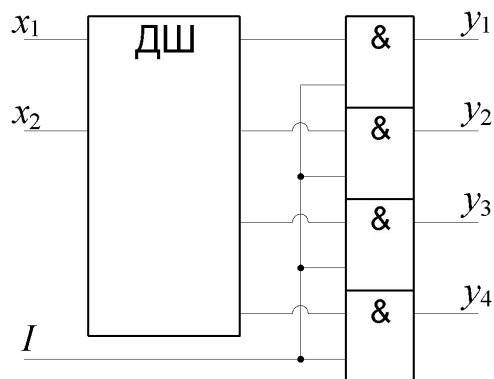
Шифраторы

Шифраторы, в отличие от дешифраторов, осуществляют обратную функцию, т.е. преобразуют унитарный код в двоичный.



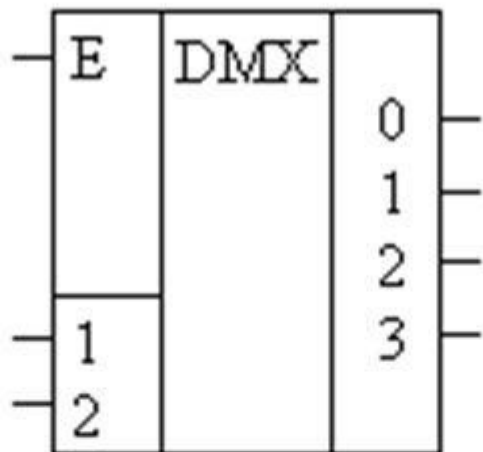
Вывод E – вход включения шифратора

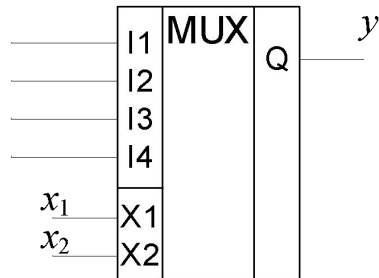
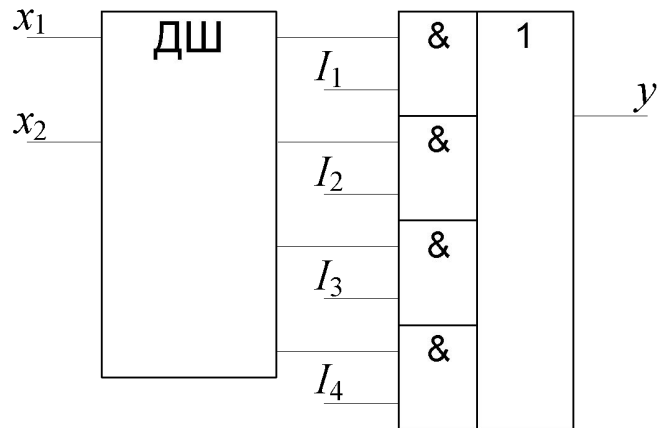
Демультимплексоры и мультиплексоры



Демультимплексор

используется в многоканальных цифровых системах для передачи цифровой информации из одного канала в любой другой, номер которого задается дешифратором.



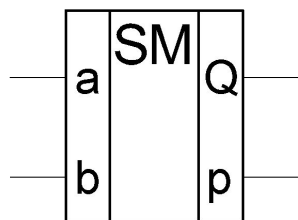


Мультиплексор

используется в многоканальных цифровых системах и, в отличие от демультиплексора, выполняет обратную функцию, т.е. собирает информацию из нескольких каналов в один общий канал.

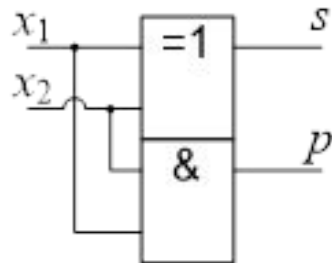
Цифровые вычислительные устройства

Полусумматоры и сумматоры цифровых кодов



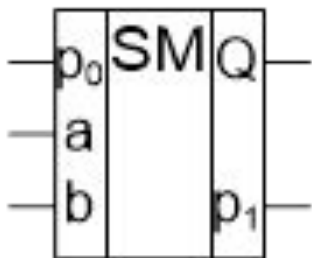
УГО полусумматора

x_2x_1	y	p
00	0	0
01	1	0
10	1	0
11	0	1



Из таблицы получаем: $y = x_1\overline{x_2} + \overline{x_1}x_2$; $p = x_1x_2$. Функция y - это функция Искл. ИЛИ, p - перенос в следующий разряд.

Полный одноразрядный сумматор



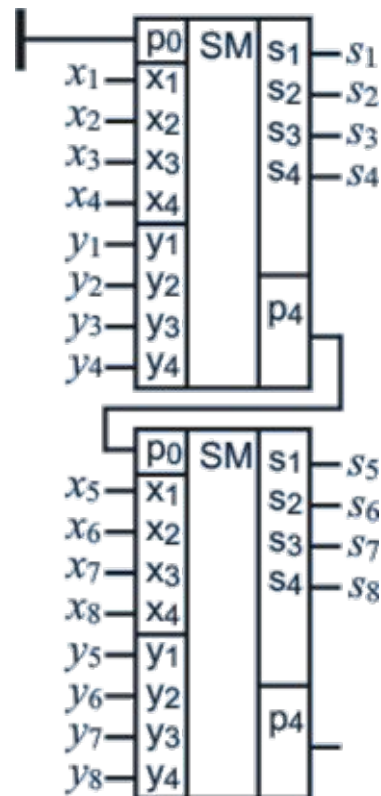
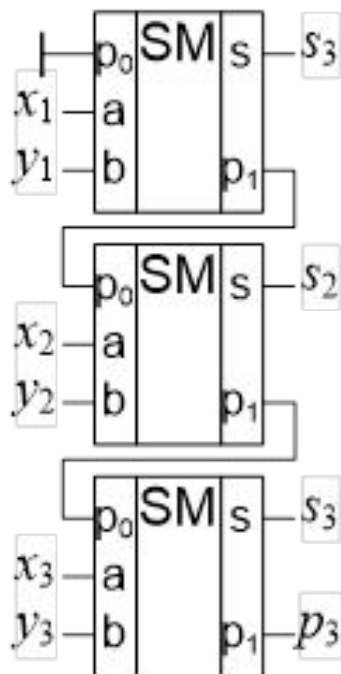
В отличие от полусумматора, полный одноразрядный сумматор учитывает перенос из предыдущего разряда.

Многоразрядные сумматоры

Многоразрядный сумматор предназначен для суммирования двух *многоразрядный двоичных* чисел и выполняется на основе *полного* *одноразрядного сумматора*.

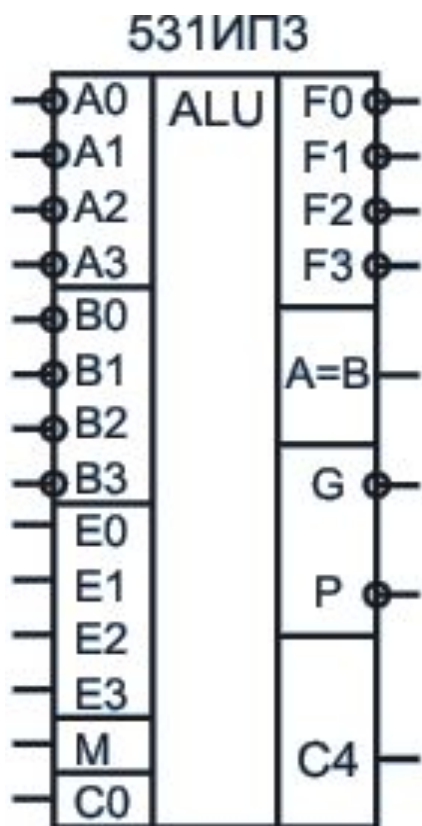
При этом могут использоваться три различных варианта построения сумматора: *параллельный многоразрядный сумматор с последовательным переносом*; *сумматор с параллельным (сквозным) переносом*; *сумматор со смешанным переносом*.

Многоразрядный сумматор с последовательным переносом и со смешанным переносом



Арифметико-логическое устройство

Таблица 3.6 - Операции АЛУ 531ИПЗ



Е-код	M=0	M=1
0000	$1+A+C_0$	
0001		
0010		
0011		1
0100		
0101		
0110		
0111		
1000		
1001		
1010		B
1011		A+B
1100		0
1101		
1110		AB
1111		A
* - в скобках логические операции		

Тактируемые цифровые устройства

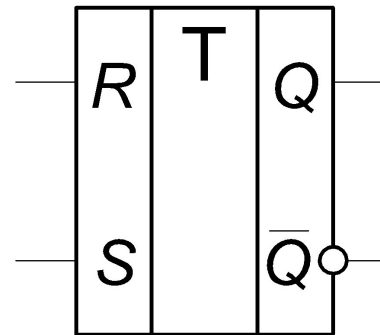
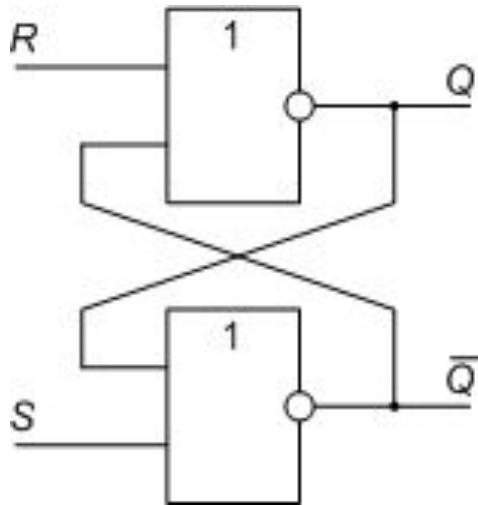
Цифровые запоминающие устройства

Триггеры

Триггер — это одноразрядное, цифровое запоминающее устройство, содержащее запоминающий элемент и схему управления его работой. Запоминающий элемент способен сохранять двоичную информацию (состояние 0 или 1) после окончания действия входных импульсов.

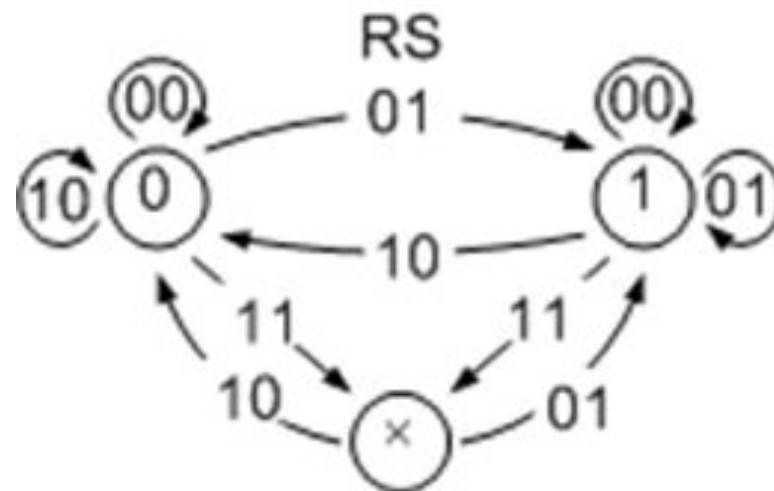
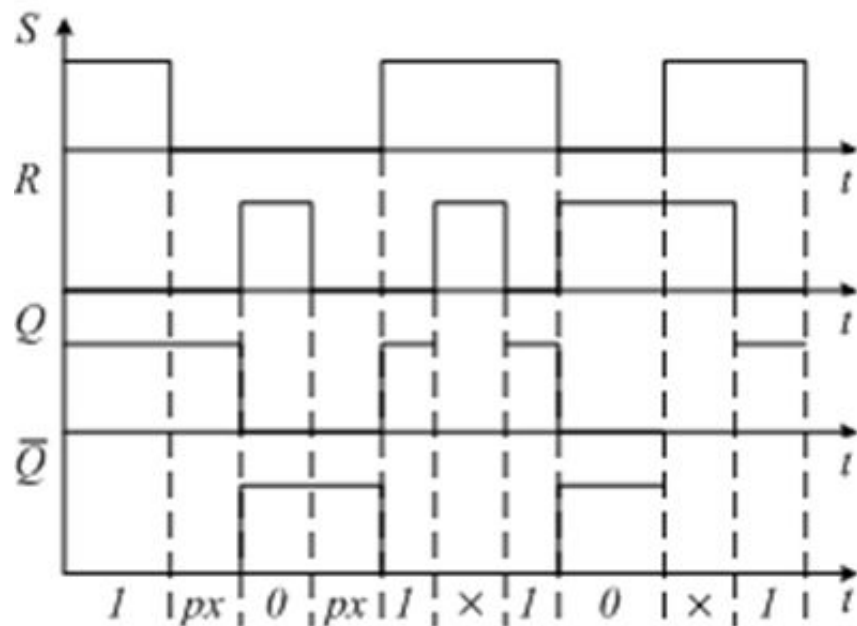
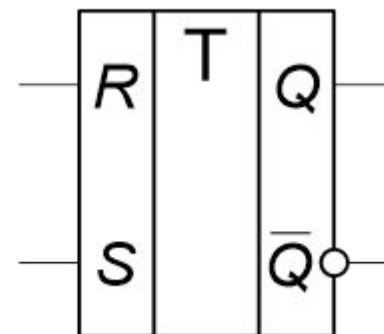
Асинхронные триггеры

- **RS-триггер.** Этот триггер является основой для построения триггеров других типов и используется в качестве запоминающего элемента.



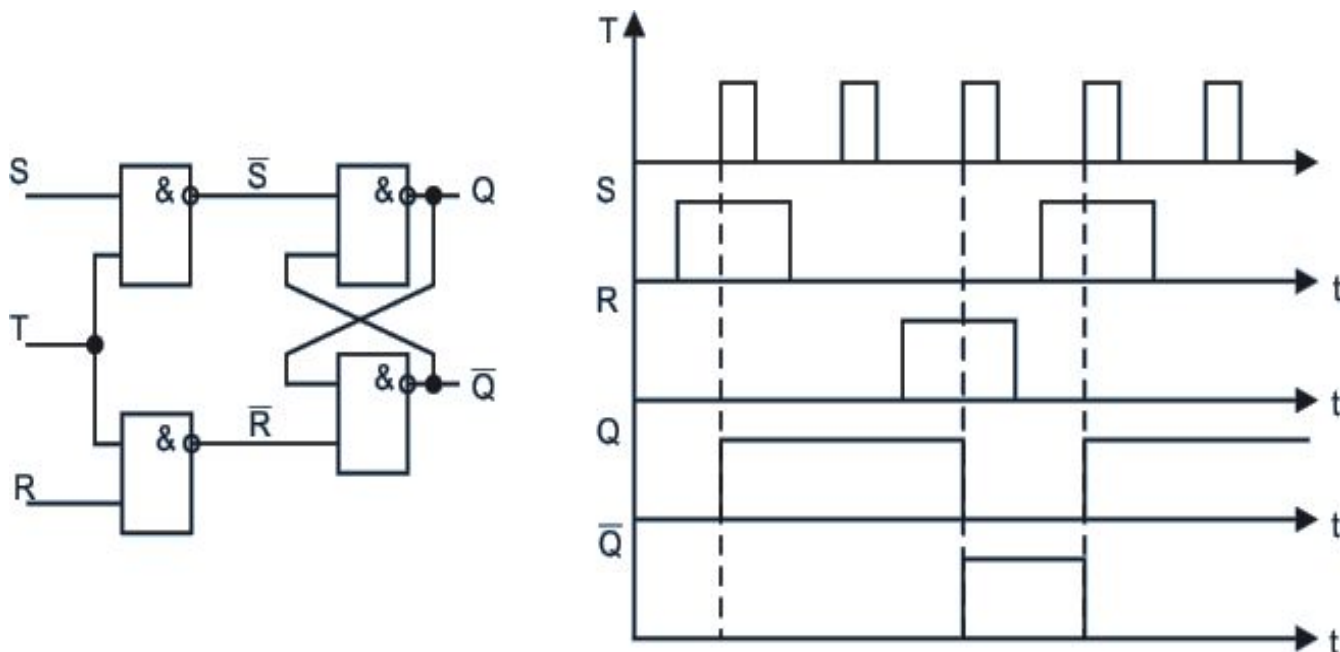
Асинхронный RS-триггер: а), б) функциональная схема триггера на ИЛИ-НЕ и его УГО

RS	Q_{n+1}	
00	Q_n	Хранение информации
01	1	Запись 1
10	0	Запись 0
11	x	Запрещенный режим



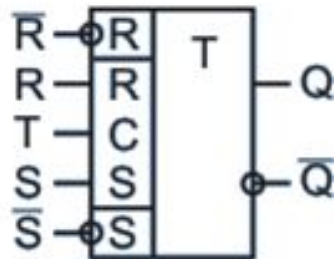
Синхронные триггеры

RS-триггер

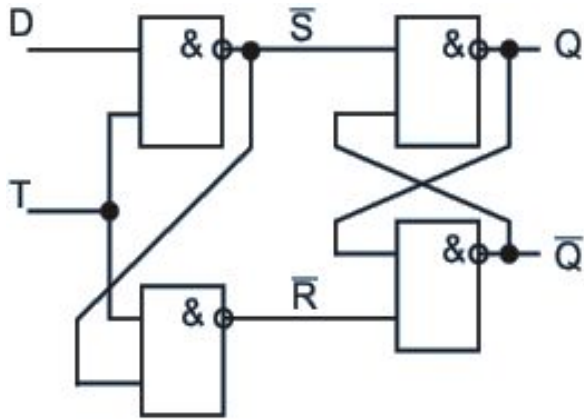


Однотактный RS триггер (слева) и его временные диаграммы (справа)

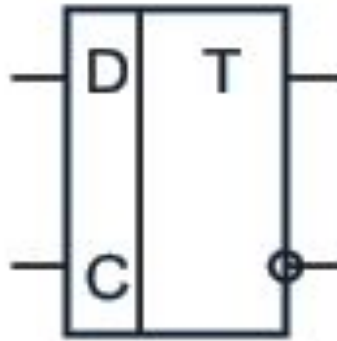
УГО однотоактного RS-триггера



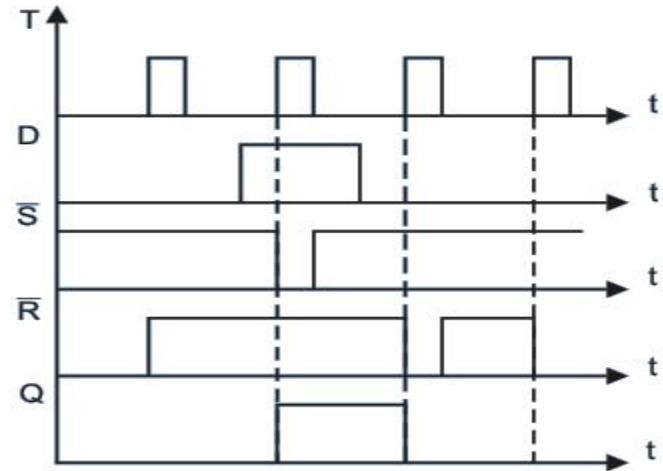
Триггер D-типа



а)



б)

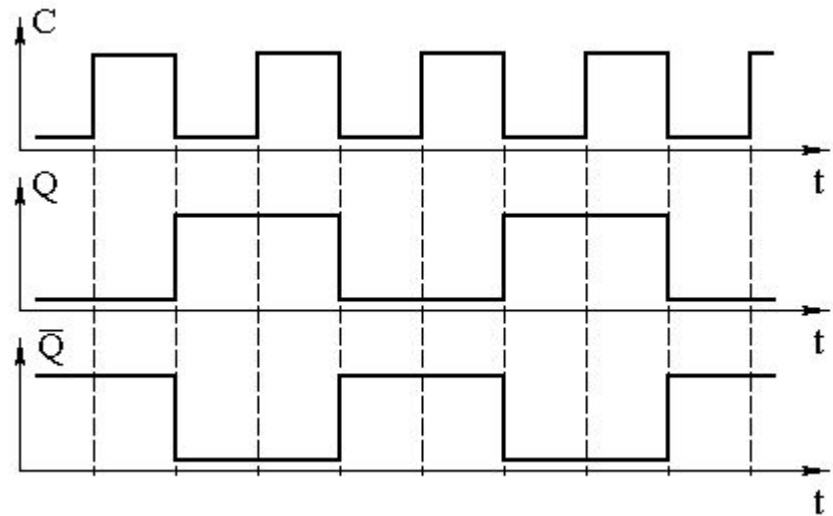
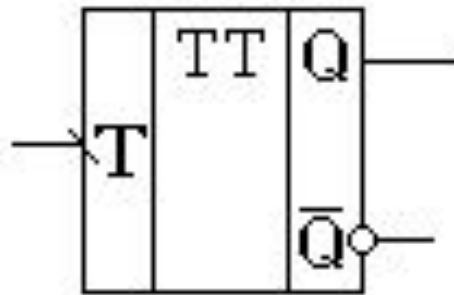


в)

Однотактный D-триггер его функциональная
схема (а),

УГО (б) и временные диаграммы (в)

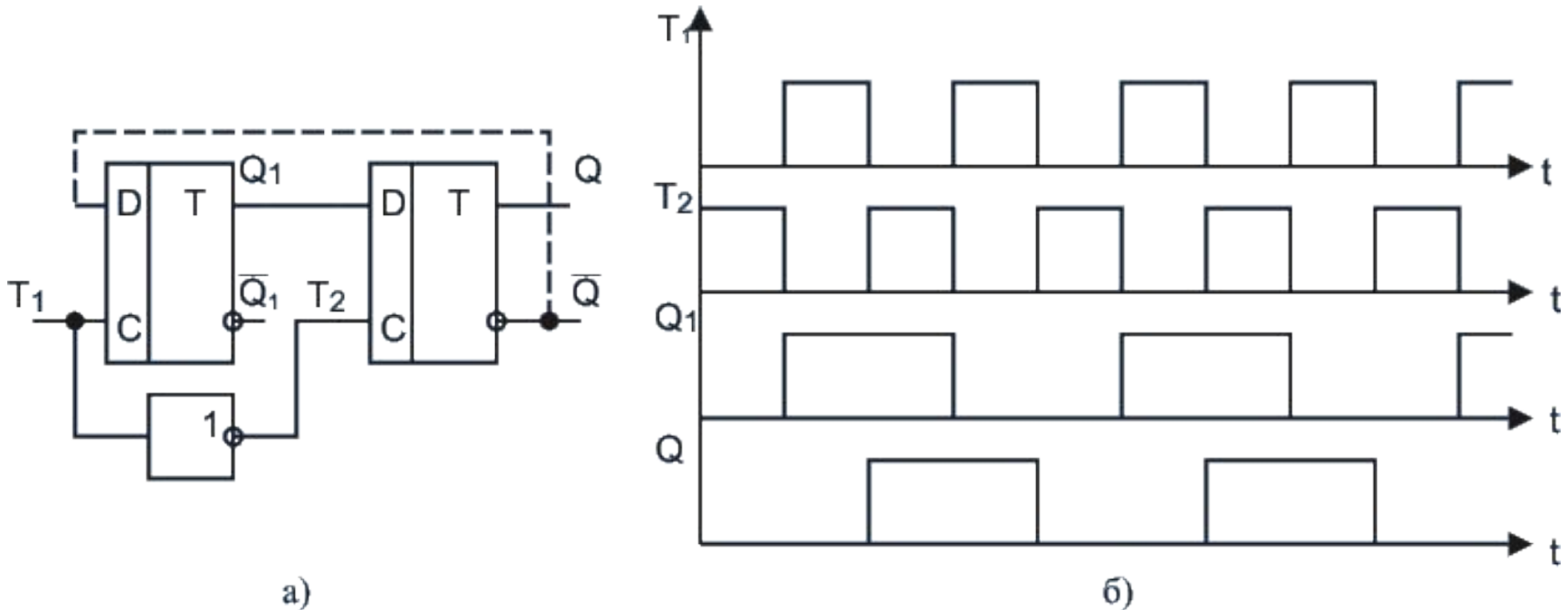
Триггер Т–типа (счётный триггер)



Условно-графическое обозначение Т-триггера и временные диаграммы

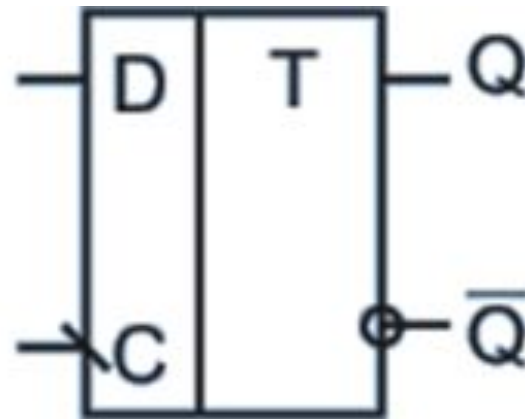
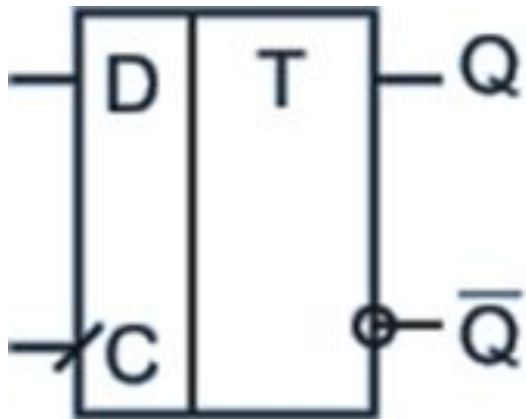
Двухтактные триггеры

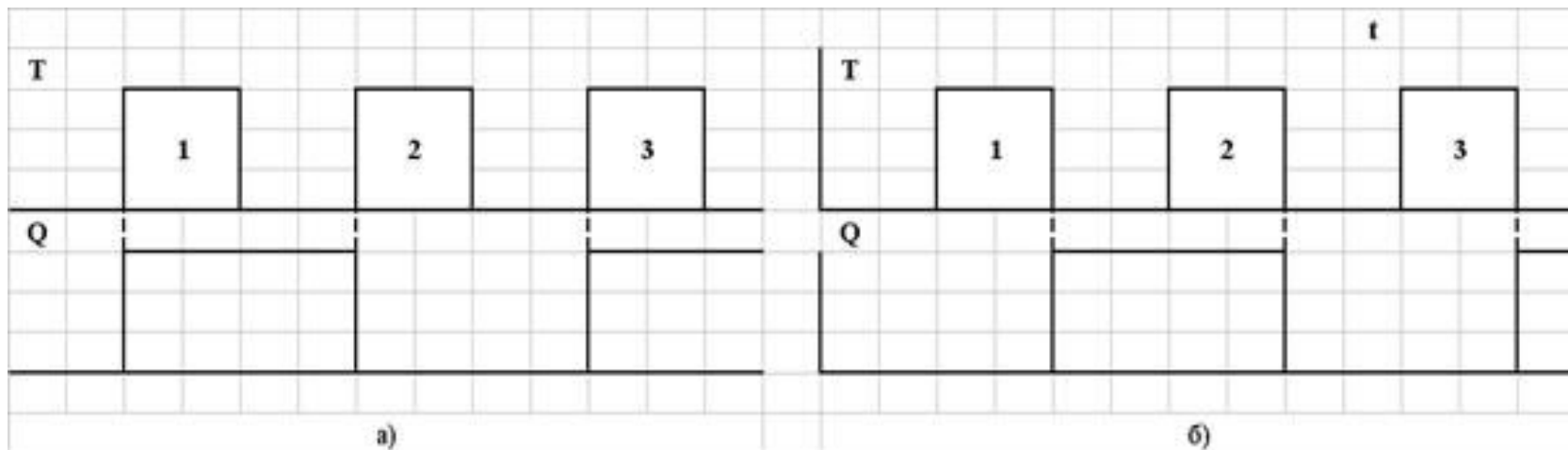
- Двухтактный триггер D-типа



Двухтактный D-триггер (а) и его временные диаграммы (б)

- Двухтактные триггеры могут работать:
- по переднему фронту;
- по заднему фронту.
- Тактовые входы таких триггеров на УГО обозначаются косой чертой





Временные диаграммы Т-триггера:

- а) срабатывание по переднему фронту импульса;
- б) срабатывание по срезу (заднему фронту) импульса

Регистры

- *Регистрами* называются многоразрядные цифровые запоминающие устройства, предназначенные для приёма, хранения, преобразования и передачи информации.
- Основу регистра составляют триггеры. Обычно используются D-триггеры, как наиболее удобные для записи и хранения информации.

Регистры делятся на:

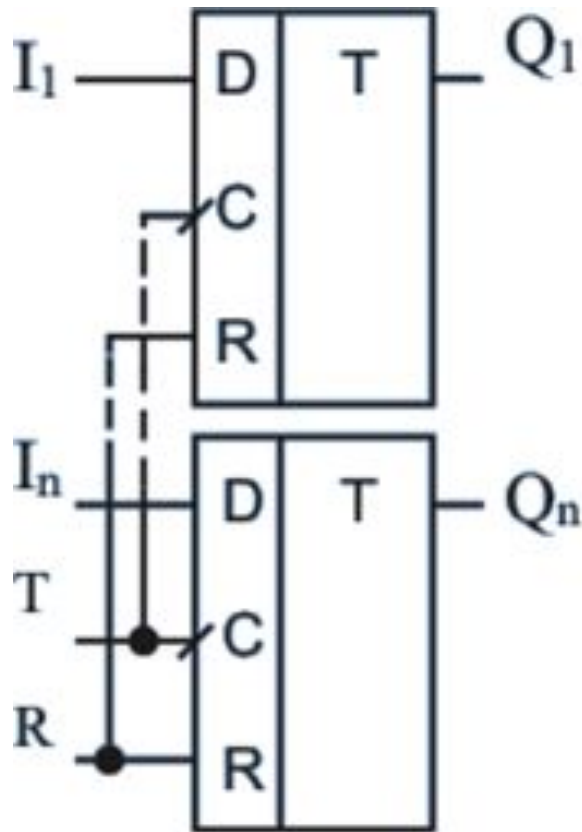
1. параллельные;
2. последовательные;
3. параллельно-последовательные.

Параллельные регистры

- Параллельный n -разрядный регистр представляет собой n триггеров, на информационные входы которых подается n -разрядный двоичный код, который необходимо запомнить, а на объединенные тактовые C -входы подаётся тактовый импульс, по переднему фронту которого осуществляется запоминание. В регистре может быть дополнительный вход R сброса регистра в нулевое состояние.

Параллельные регистры

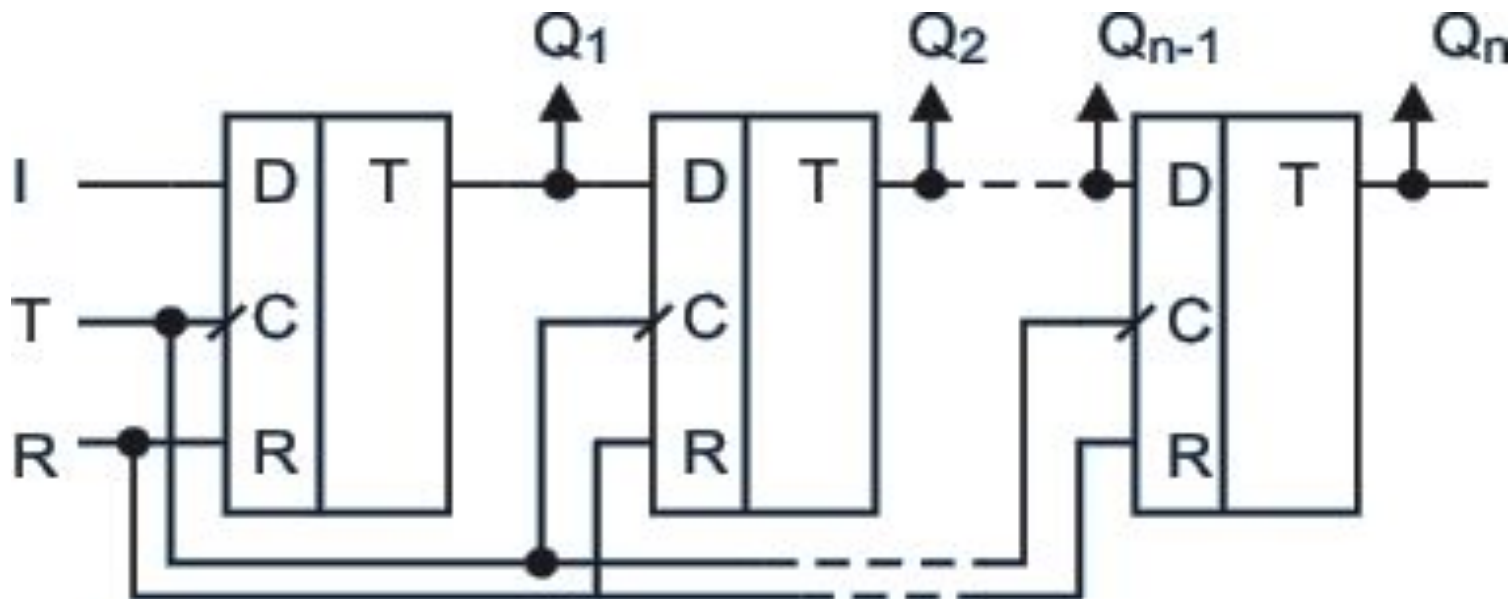
Параллельный n -разрядный регистр представляет собой n триггеров, на информационные входы которых подается n -разрядный двоичный код, который необходимо запомнить, а на объединенные тактовые C -входы подаётся тактовый импульс, по переднему фронту которого осуществляется запоминание (рисунок 3.38). В регистре может быть дополнительный вход R сброса регистра в нулевое состояние.



Последовательные регистры

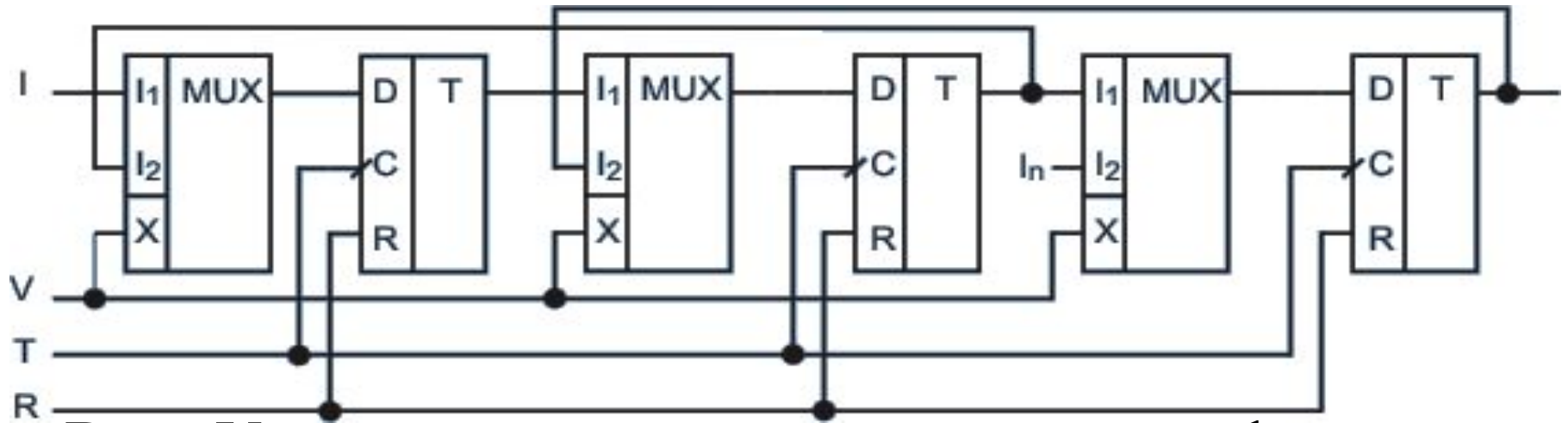
- Последовательные регистры работают с последовательным кодом, разряды которого разделены во времени на интервалы, равные периоду следования тактового импульса T .
- Различают регистры прямого сдвига и реверсивные регистры.

Последовательный регистр прямого сдвига



- Регистры прямого сдвига осуществляют приём (передачу) информации, сдвигая её в регистре на 1 разряд вправо при приходе одного тактового импульса. Для полного приёма (передачи) информации требуется n тактов. Регистр представляет собой n последовательно соединённых D-триггеров, тактовые входы которых объединены. Имеется один информационный вход I для приёма информации. В ход R позволять сбросить все триггеры регистра одновременно.

Реверсивный регистр



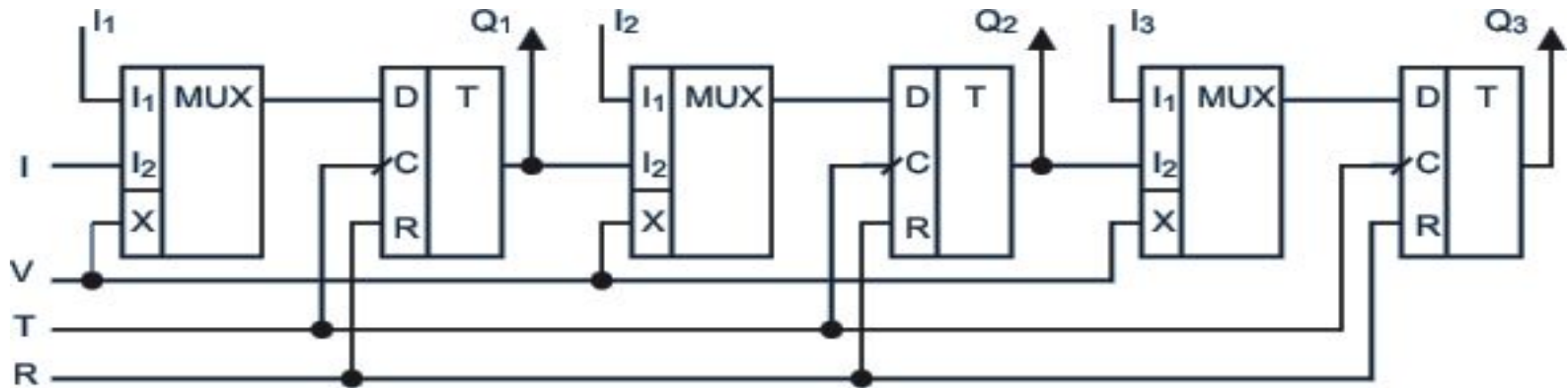
Вход V управляет направлением сдвига информации. При $V=0$ к выходу мультиплексора подключён вход I_1 и схема преобразуется в схему со сдвигом информации вправо. При $V=1$ к входам мультиплексора подключён вход I_2 . При этом выход последующего триггера подключается к информационному входу предыдущего и таким образом при приходе тактовых импульсов осуществляется сдвиг информации влево.

Реверсивный сдвиговый регистр позволяет осуществить сдвиг информации внутри регистра, как вправо, так и влево. Это может потребоваться для преобразования последующего кода. Например, если первоначально следовал старший разряд кода, то после преобразования первым будет преобразован младший разряд кода. Для организации реверсивного режима между входами и выходами триггеров включаются одноразрядные мультиплексоры с двумя информационными входами.

Параллельно-последовательные регистры

- Такие регистры служат для преобразования информации с последовательного кода в параллельный или наоборот, а также могут выполнять функции как последовательных, так и параллельных регистров.
- Для организации одновременного наличия двух режимов (параллельного и последовательного) чаще пользуются более простым схемотехническим решением - между триггерами включают одноразрядный мультиплексор.

Параллельно-последовательные регистры



Параллельно-последовательный регистр

Параллельно-последовательные регистры

- При $V=0$ организуется режим параллельного регистра. При этом в мультиплексорах к выходам подключены входы I_1 , на которые подаётся параллельный код. В момент прихода тактового импульса этот код записывается в регистр. При $V=1$ организуется последовательный режим работы со сдвигом информации вправо. В мультиплексорах к выходам подключены входы I_2 , через которые осуществляются последовательный режим работы.

Счётчики импульсов

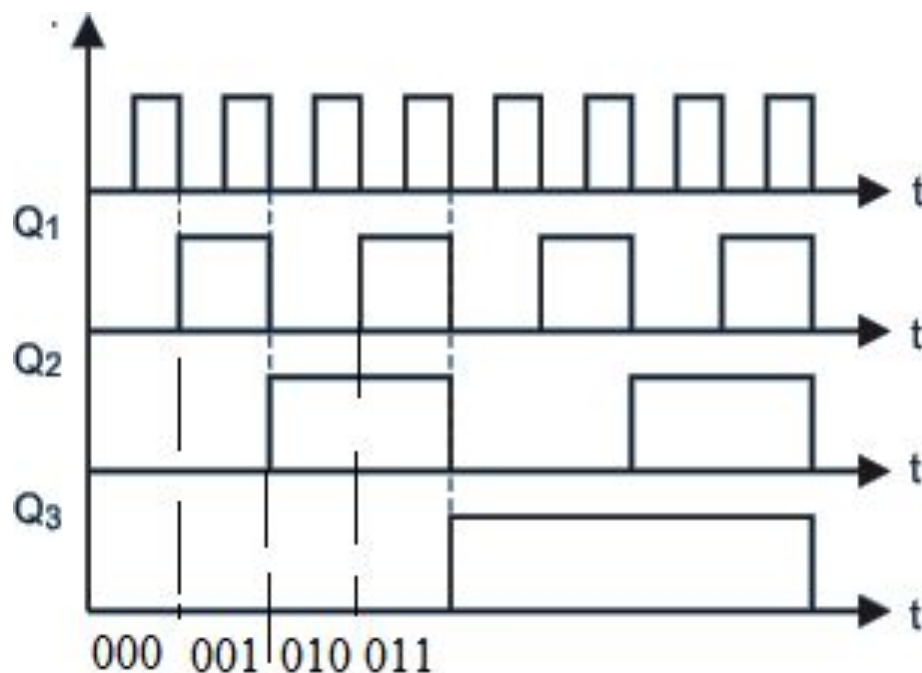
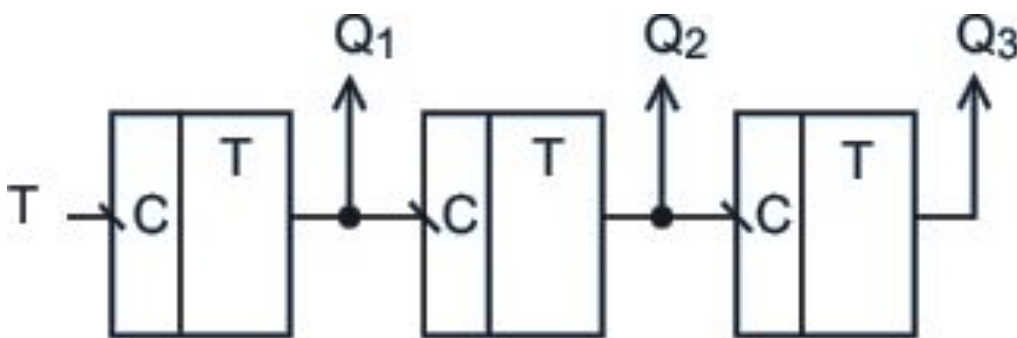
- Счетчиком называется устройство, предназначенное для счета входных импульсов и фиксации их числа в двоичном коде. По принципу действия счетчики делятся на суммирующие, вычитающие и реверсивные. По быстродействию счетчики делятся на асинхронные, счетчики с переносом и синхронные.

- К основным характеристикам счетчиков относятся *модуль счета* (коэффициент пересчета $K_{сч}$) и *быстродействие*. Модуль счета $K_{сч}$ характеризует число устойчивых состояний счетчика, т.е. предельное число импульсов, которое может быть сосчитано счетчиком.
- Основой для построения счётчиков является счётный триггер или Т-триггер. Он представляет собой простейший одноразрядный счётчик

Асинхронные счётчики импульсов

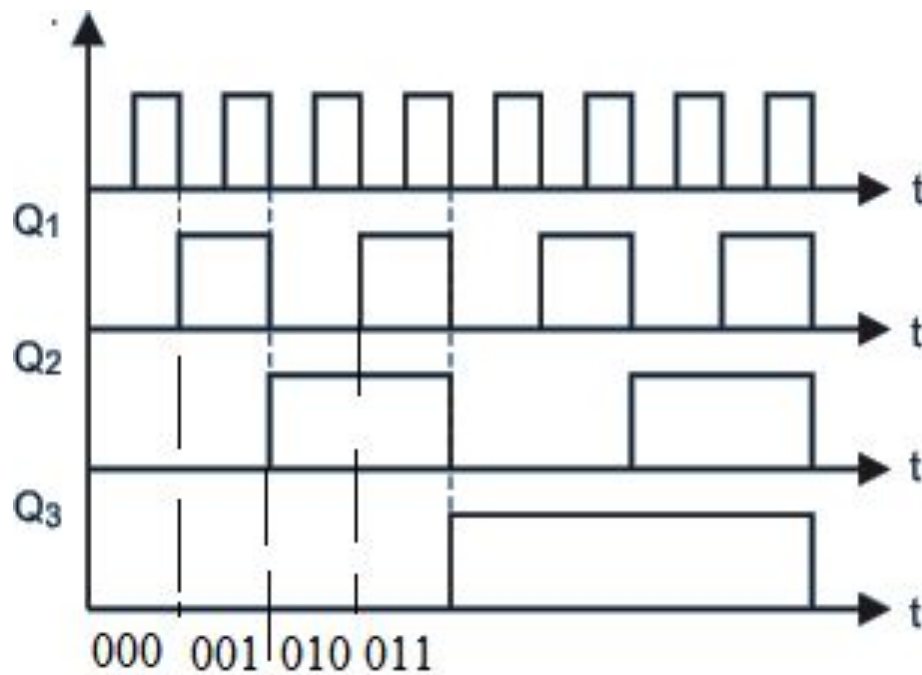
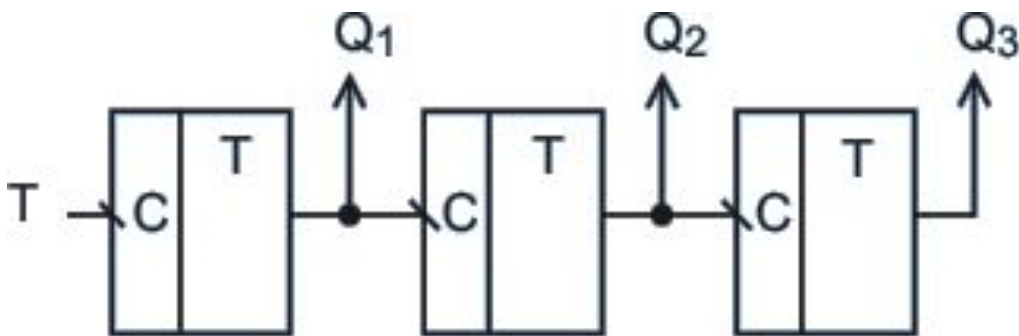
- Асинхронный счетчик импульсов представляет собой последовательно соединенные триггеры Т-типа, при этом срабатывание каждого последующего триггера осуществляется по фронту импульса, формируемого предыдущим триггером.

Суммирующие асинхронные счётчики



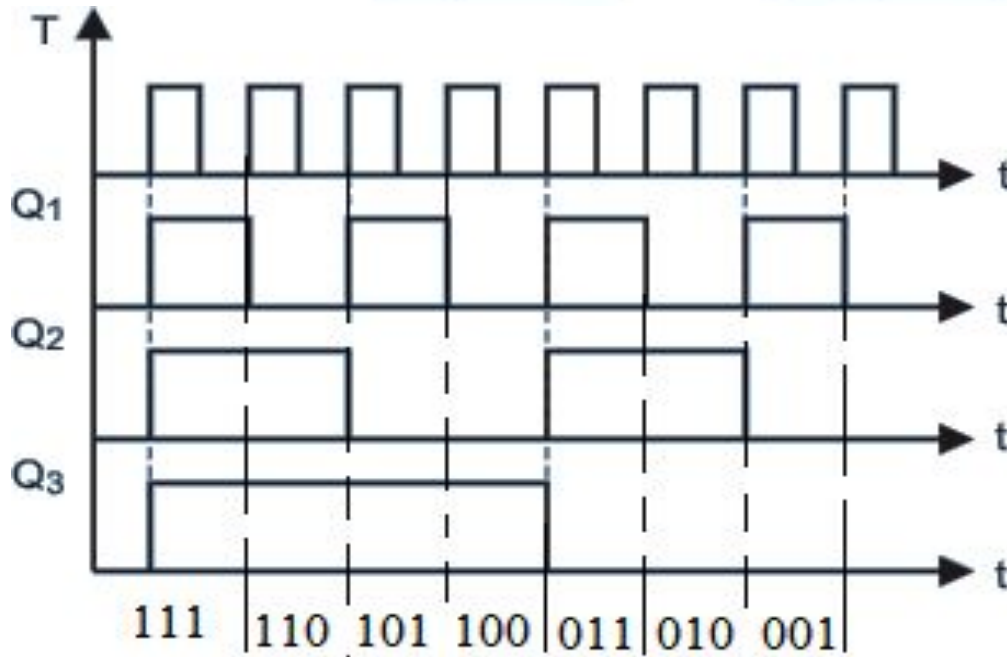
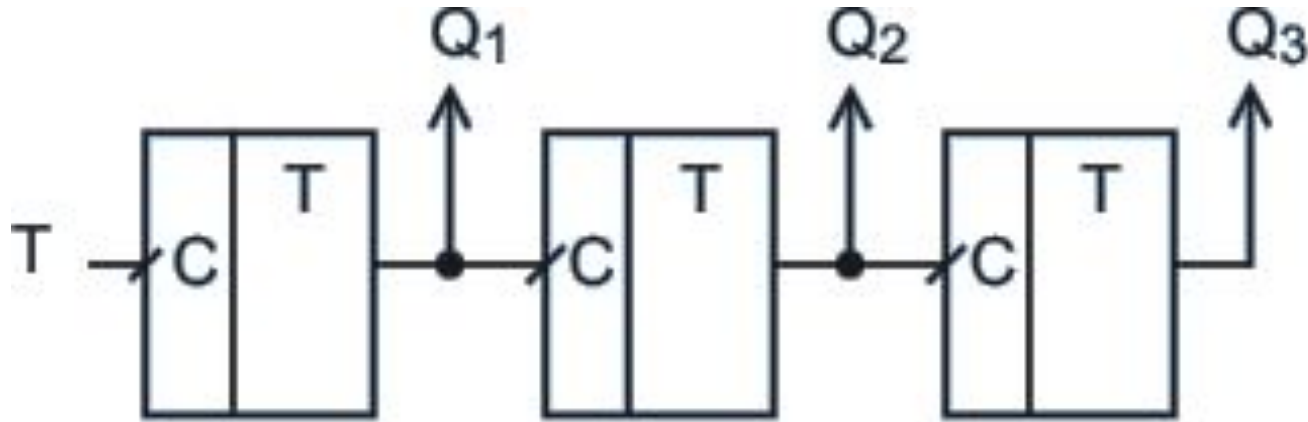
Коэффициент пересчета $K_{сч} = 2^n = 2^3 = 8$, где n — число триггеров счетчика (разрядность) и их временные диаграммы. Из временных диаграмм видно, что счётный триггер делит частоту на 2, поэтому счётные триггеры и счётчики импульсов могут использоваться как делители частоты.

Суммирующие асинхронные счётчики



$(Q_3Q_2Q_1)_{(2)}$	$X_{(10)}$
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

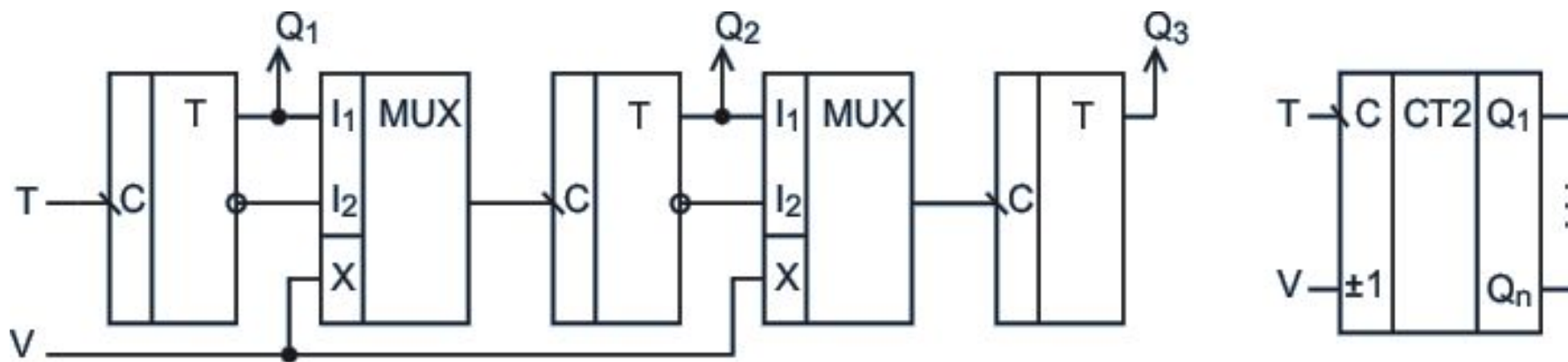
Вычитающие асинхронные счётчики



$X_{(2)}$	$X_{(10)}$
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

Реверсивные счётчики

- Такой счетчик является комбинацией двух предыдущих и позволяет осуществлять как суммирование, так и вычитание импульсов, что осуществляется посредством включения в состав каждой разрядной схемы счетчика мультиплексора.



Асинхронный реверсивный счетчик (слева) и его УГО (справа)

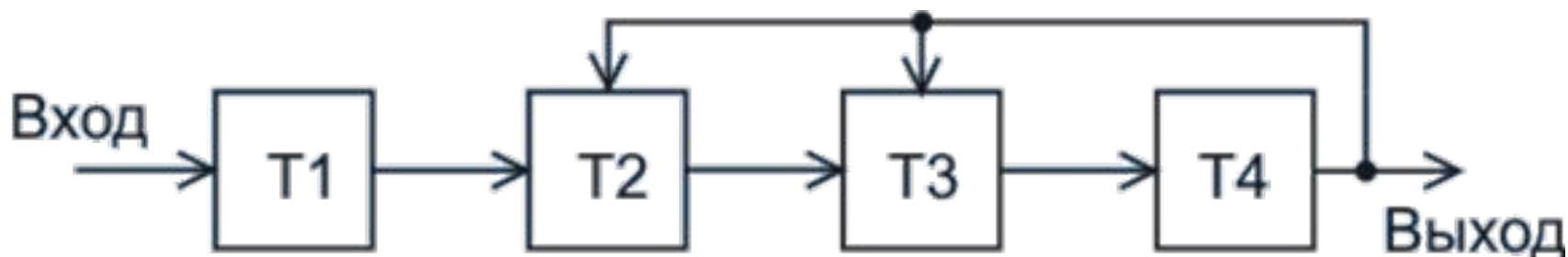
Реверсивные счётчики

- Сигнал на входе V определяет вид межразрядной связи, а, следовательно, и тип получаемого счетчика. Если на вход ± 1 подать логический 0, то подключается первый канал мультиплексора и прямые входы предыдущих триггеров подключаются к тактовому входу последующих триггеров – суммирующий режим работы. При подаче на ± 1 логической 1, инвертирующие выходы предыдущих триггеров подключаются к тактовым входам последующих - вычитающий режим работы.

Асинхронные счётчики с произвольным коэффициентом пересчёта

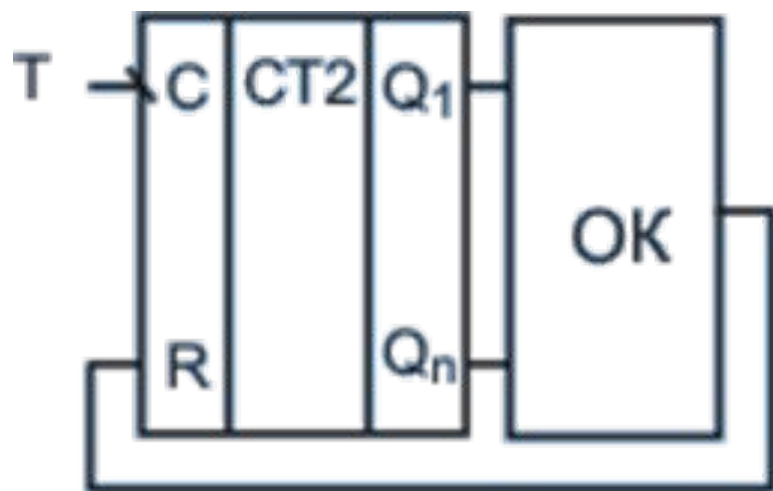
- В асинхронных счётчиках произвольный коэффициент пересчёта обеспечивается с помощью принудительной установки через асинхронные входы триггеров (R или S). Могут использоваться три способа принудительной установки: принудительный насчет, принудительная начальная установка и принудительный сброс в ноль при достижении требуемого состояния счетчика.

Счетчик с принудительным насчетом



Структурная схема асинхронного счетчика с принудительным насчетом $K_{сч}=10$

Счётчики с принудительным сбросом



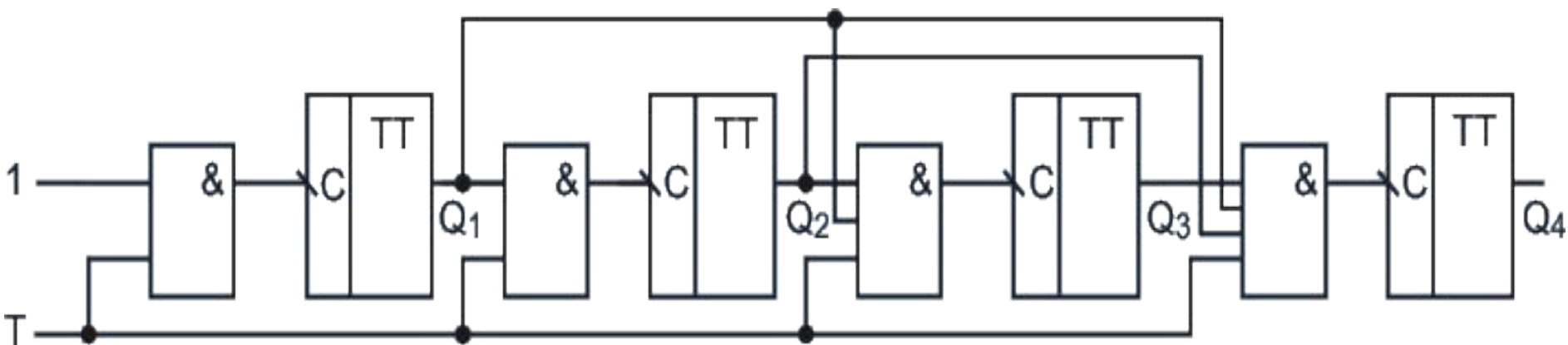
Определитель кода (ОК) должен определять соответствующий Ксч. При этом счётчик будет нормально считать в диапазоне от 0 до К.сч-1, а состояние счётчика Ксч будет сбрасывать его в ноль, так как на вход сброса R счётчика будет подан активный сигнал.

Счетчики с переносом

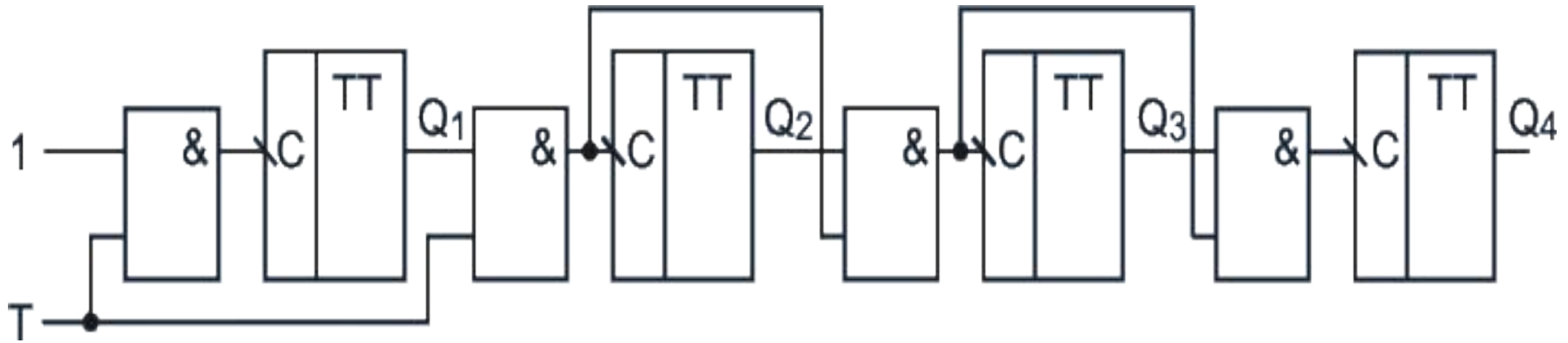
Такие счётчики делятся на 2 типа:

1. С параллельным переносом;
2. С последовательным переносом.

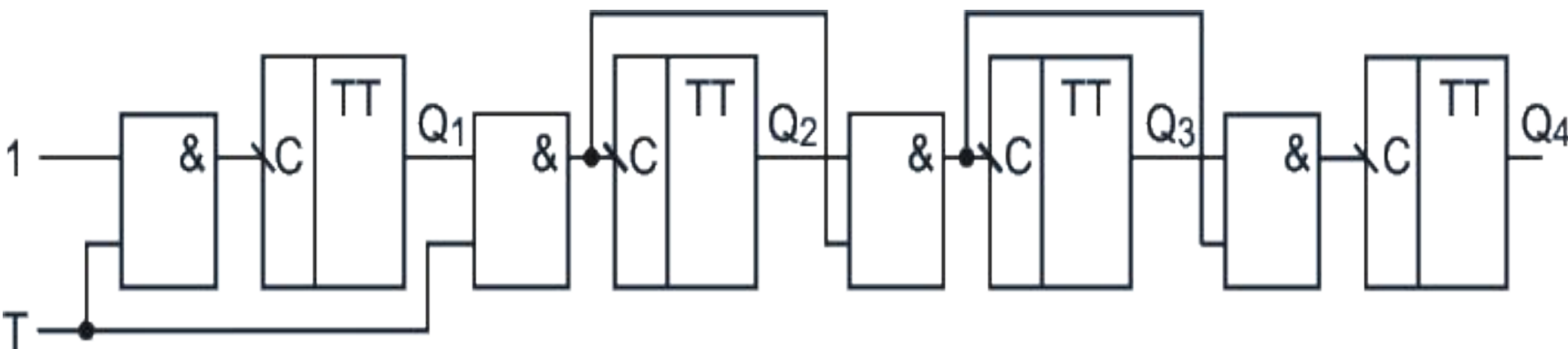
Счетчик с параллельным переносом



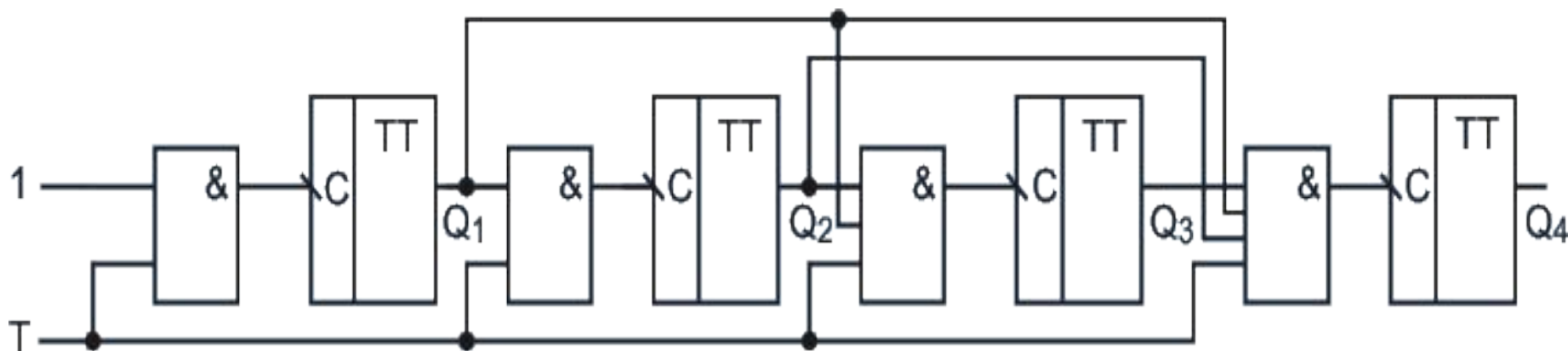
Счётчик с последовательным переносом



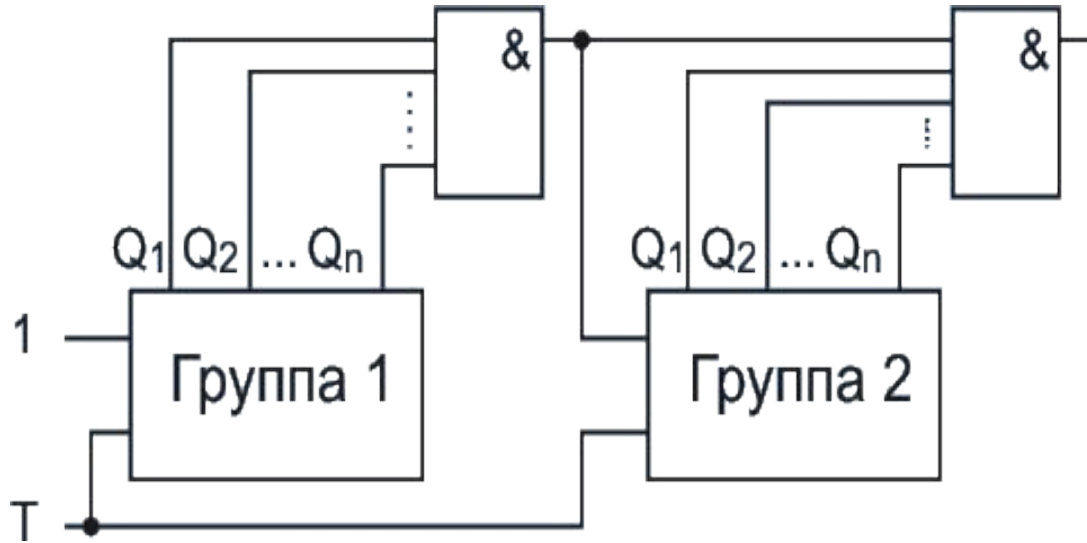
Счётчик с последовательным переносом



Счетчик с параллельным переносом

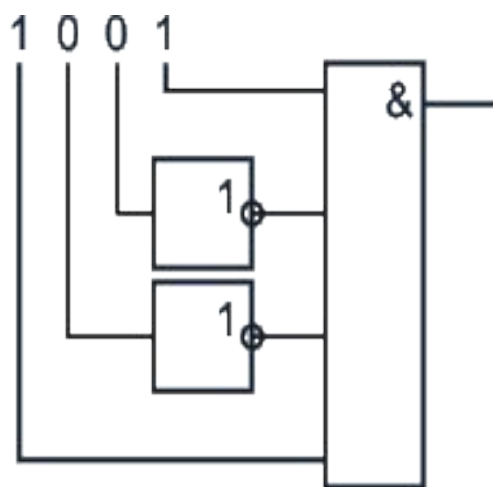


Счетчики с комбинированным переносом

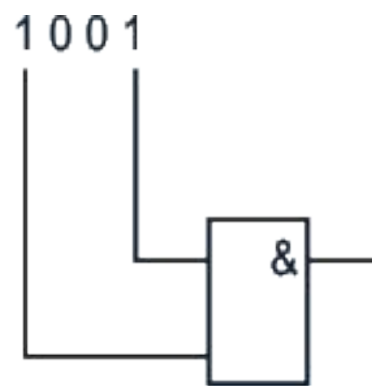


Идея построения счетчиков с комбинированным переносом (рисунок 3.58) состоит в разбиении разрядных схем счетчика на группы, внутри которых осуществляют либо последовательный, либо параллельный перенос. Формирование сигнала переноса между группами выполняется элементами И лишь в случае, когда триггеры всех входящих в данную группу разрядных схем установлены в единичное состояние, т.е. по параллельному принципу.

Счётчики с произвольным коэффициентом пересчёта



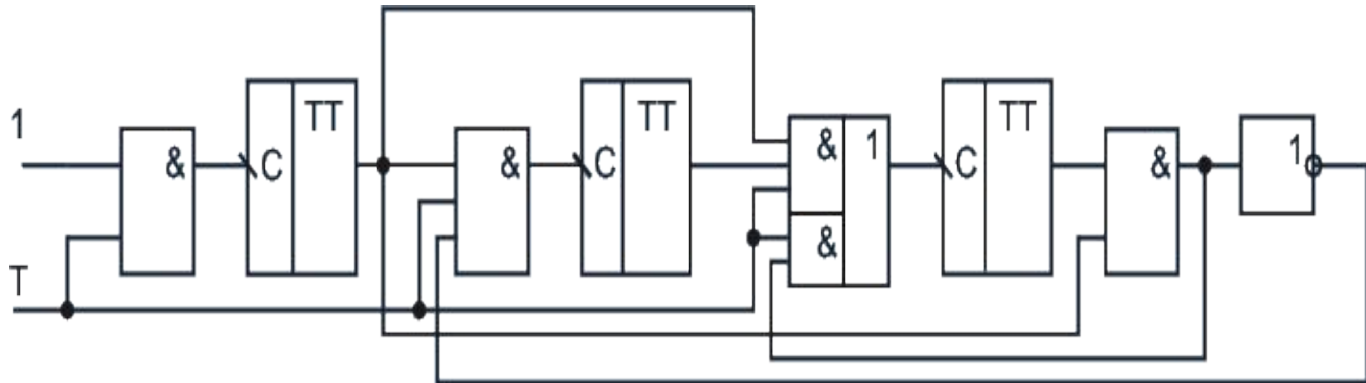
а)



б)

Функциональная схема дешифратора (определителя)
выходного кода счетчика

Счётчики с произвольным коэффициентом пересчёта



Счетчик с коэффициентом пересчета $K_{сч}=6$

Счётчики с произвольным коэффициентом пересчёта

- В качестве основы для построения счётчиков с произвольным коэффициентом пересчёта служит схема счётчика с полным параллельным переносом.
- Порядок разработки счётчика с произвольным коэффициентом пересчёта $K_{сч}$.
- 1. Выбирается n счётных триггеров (разрядов) счётчика из условия $2^{n-1} < K_{сч} < 2^n$.
- 2. В схему счётчика с параллельным переносом добавляется дешифратор (определитель) выходного кода счётчика, равного $(K_{сч} - 1)$. Он реализуется на основе n -разрядной схемы И. Можно уменьшить количество входов в схеме И, если подавать на неё только разряды, в коде которых $(K_{сч} - 1)$ присутствует единица.

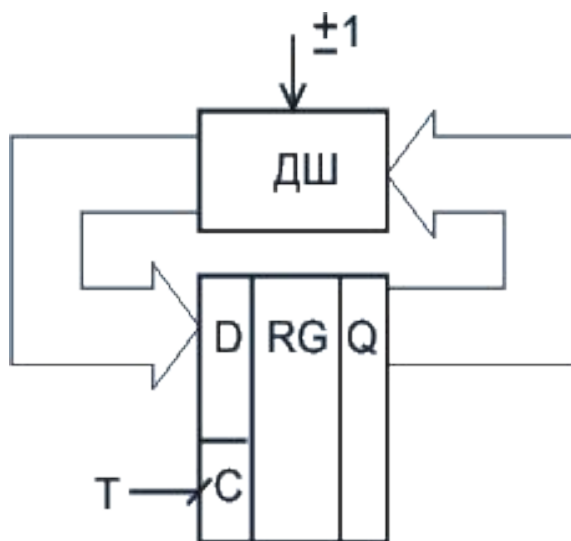
Счётчики с произвольным коэффициентом пересчёта

Записывается значение трёх кодов: $K_{сч} - 1$, $K_{сч}$ и 0. Далее анализируются действия необходимые, чтобы перевести счётчик из состояния ($K_{сч} - 1$) в состояние 0, а не в состояние $K_{сч}$. При этом возможно 3 варианта для любого разряда счётчика: *оставить формирование переноса без изменений, запретить срабатывание разряда или сброс разряда в ноль.*

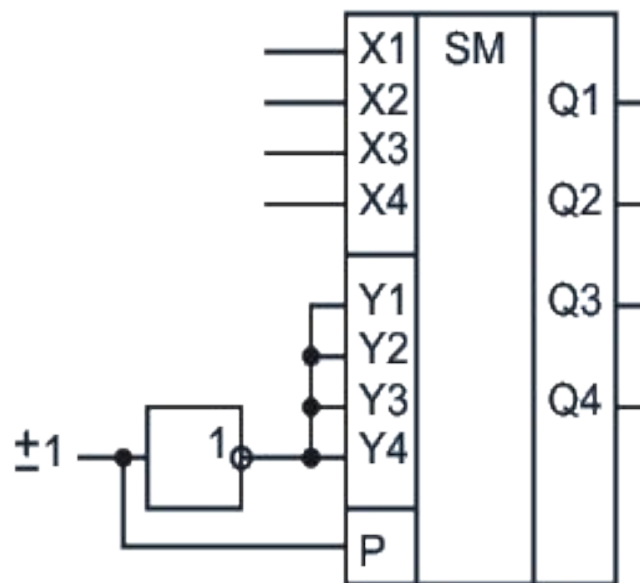
Например, для $K_{сч} = 6$ имеем:

$K_{сч} - 1$	1	0	1
$K_{сч}$	1	1	0
0	0	0	0
Действие:	сброс	запрет	Без изм.

Синхронные счётчики



а)



б)

Минимизация логических функций

- **Карта Вейча** – это прямоугольная таблица, число клеток в которой для логической функции n переменных равно 2^n , каждой из клеток поставлен в соответствие некоторый набор входных переменных, причем рядом расположенным клеткам соответствуют соседние наборы входных переменных (кодов), а в самих клетках записаны значения функции, определенные для ЭТИХ КОДОВ.

Карты Вейча

x_2x_1	y
0 0	y_1
0 1	y_2
1 0	y_3
1 1	y_4

	x_1	$\bar{x}_1$
x_2	y_4	y_3
$\bar{x}_2$	y_2	y_1

а) Карта Вейча функции
2-х переменных

$x_3x_2x_1$	y
0 0 0	y_1
0 0 1	y_2
0 1 0	y_3
0 1 1	y_4
1 0 0	y_5
1 0 1	y_6
1 1 0	y_7
1 1 1	y_8

	x_1	$\bar{x}_1$	
x_2	y_4	y_8	y_3
$\bar{x}_2$	y_2	y_6	y_1
	$\bar{x}_3$	x_3	$\bar{x}_3$

б) Карта Вейча функции
3-х переменных

$x_4x_3x_2x_1$	y
0 0 0 0	y_1
0 0 0 1	y_2
0 0 1 0	y_3
0 0 1 1	y_4
0 1 0 0	y_5
0 1 0 1	y_6
0 1 1 0	y_7
0 1 1 1	y_8
1 0 0 0	y_9
1 0 0 1	y_{10}
1 0 1 0	y_{11}
1 0 1 1	y_{12}
1 1 0 0	y_{13}
1 1 0 1	y_{14}
1 1 1 0	y_{15}
1 1 1 1	y_{16}

	x_1	$\bar{x}_1$	
x_2	y_4	y_8	y_7
$\bar{x}_2$	y_{12}	y_{16}	y_{15}
	$\bar{x}_3$	x_3	$\bar{x}_3$

в) Карта Вейча
функции
4-х переменных

Алгоритм минимизации логической функции

сводится к следующему:

1. Для логической функции составляется таблица состояний;
2. В ячейки карты *записываются* значения функции из таблицы состояний;
3. Выделяют на карте группу единиц (нулей) функции, закрываемых прямоугольниками со сторонами 2^k (где k – целое число) с учётом возможности склеивания противоположных сторон карты. Для лучшей минимизации прямоугольники нужно выбирать так, чтобы площадь была наибольшей, при этом возможно частичное наложение прямоугольников друг на друга. Задача состоит в том, чтобы минимальное количество прямоугольников закрывало, не захватывая нулей (единиц), все единицы (нули) карты.
4. Для каждого прямоугольника записывают логическую функцию в виде логического умножения аргументов, которые для данного прямоугольника не изменяют своё значение. Произведения носят название ***импликанты***.
5. Полностью минимизированная логическая функция получается путём логического сложения импликантов.

