

# Цифровая схемотехника ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

---

КУРС ЛЕКЦИЙ

ЧУ ПО «СОЦИАЛЬНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

ПРЕПОДАВАТЕЛЬ: БОРИСОВ АЛЕКСЕЙ АЛЬБЕРТОВИЧ

**RAZUMDOM**



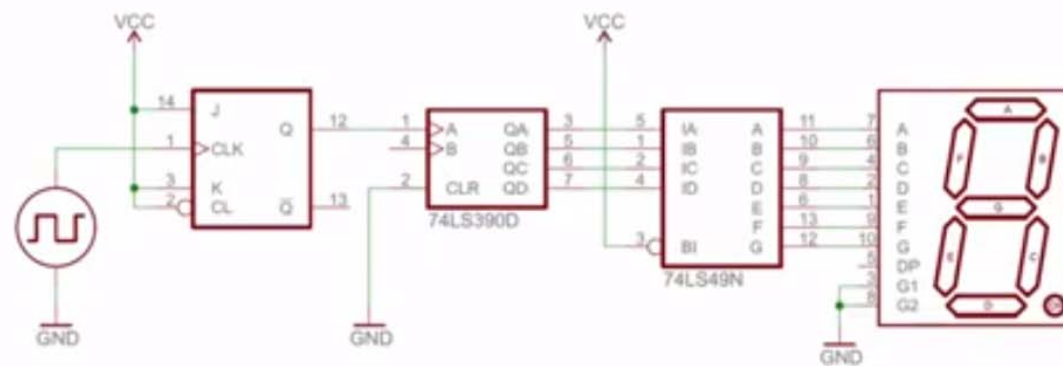
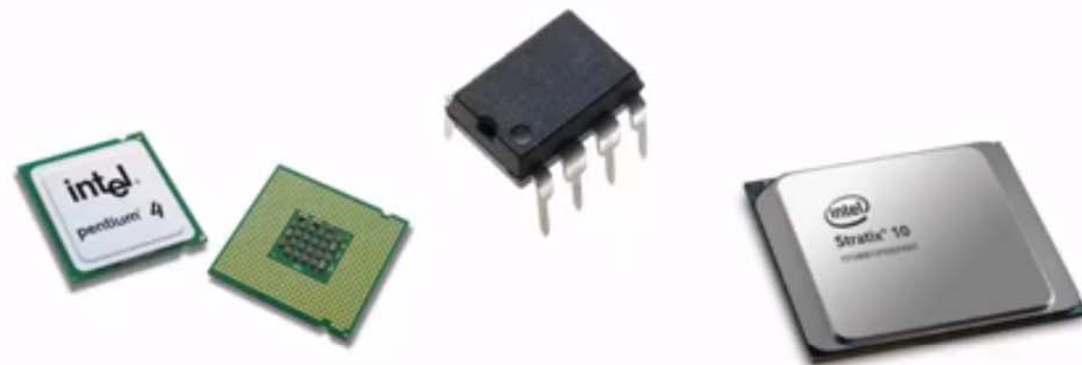
# Программа занятий

1. Вводная часть
2. Логические уровни
3. Электронные ключи
4. Операционные усилители
5. Базовые логические операции
6. Логическая арифметика Булева алгебра  
Логические функции
7. Комбинационные цифровые устройства  
Шифраторы, дешифраторы  
Мультиплексоры, демультимплексоры  
Сумматоры, умножители  
Компараторы и др
8. Последовательностные цифровые устройства  
Триггеры  
Регистры  
Счетчики
9. Элементы памяти ОЗУ, ПЗУ
11. Программируемые логические интегральные схемы

## Дополнительная часть

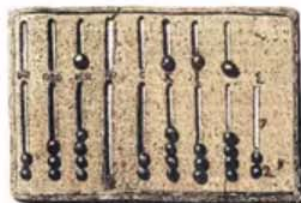
12. ЦАП и АЦП
13. Генераторы, мультивибраторы, одновибраторы
14. Микроконтроллеры
15. Параллельные интерфейсы
16. Последовательные интерфейсы
17. Беспроводные интерфейсы
18. Технологии микросхем ТТЛ, КМОП и др
19. Полупроводники: диоды, транзисторы
20. Переходные процессы при переключении

# Зачем изучать схемотехнику?



# История вычислительной техники

3000 BC



16-17 век



1920-е



1940-е



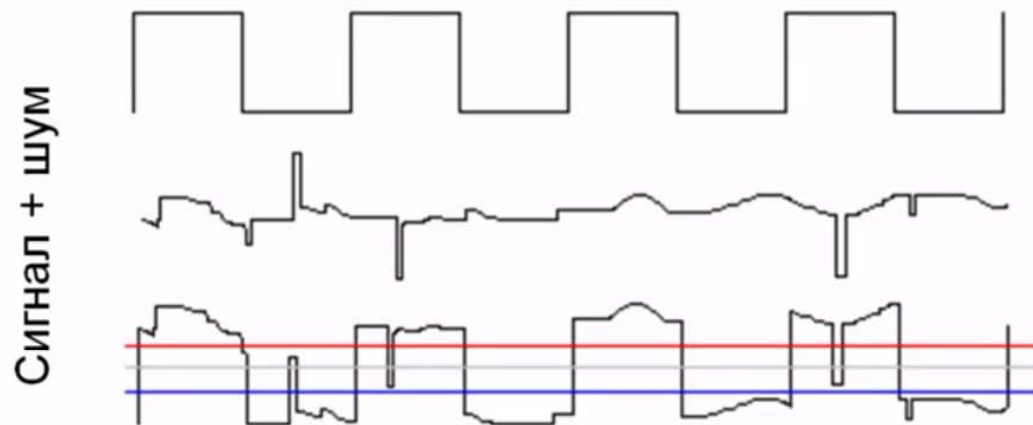
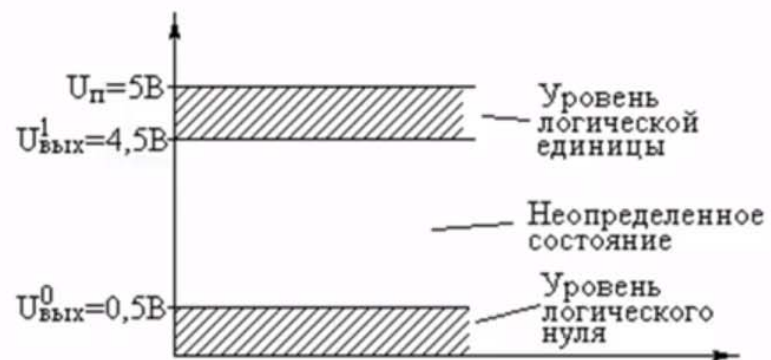
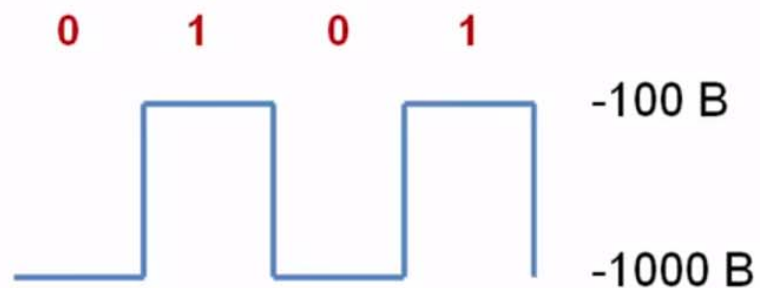
1947



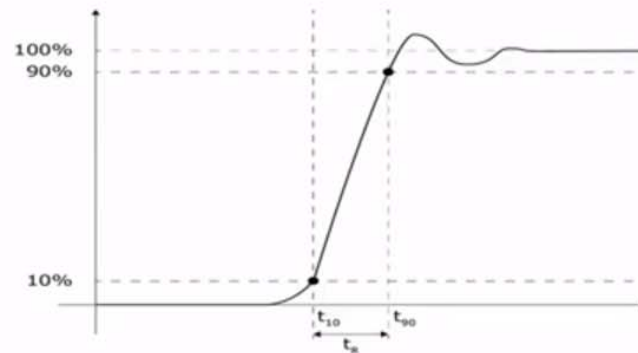
Наши дни:



# Цифровое представление информации

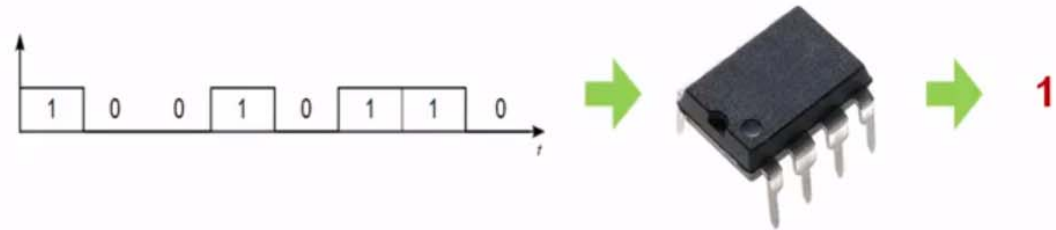
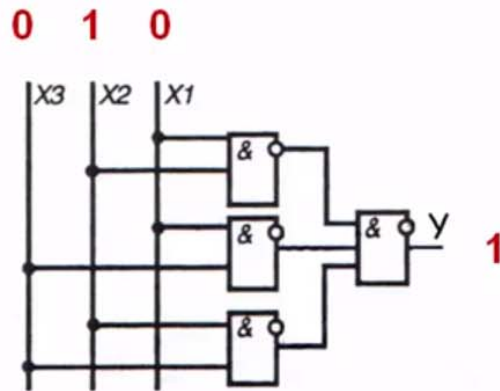
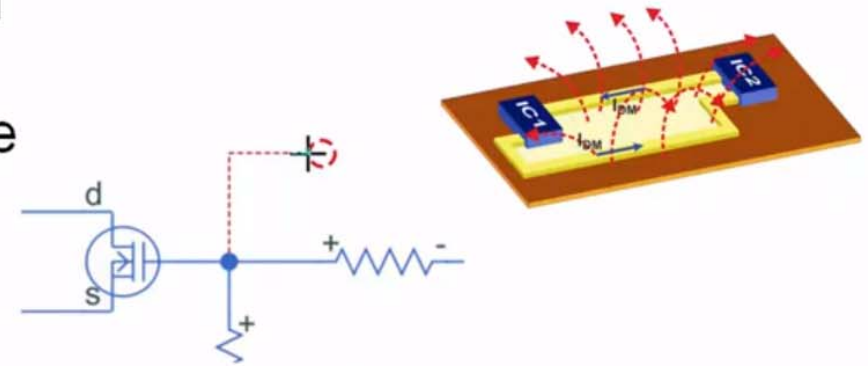
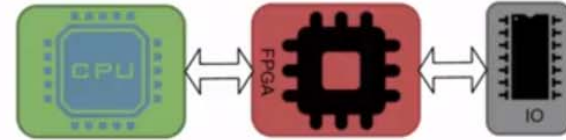


Переходный процесс  
(из 0 в 1):



# Уровни изучения цифровой схемотехники

1. **Системный уровень** – архитектура, интерфейсы, устройства
2. **Конструкторский уровень** – компоненты на плате, совместимость
3. **Электрическая схема** – сосредоточенные компоненты, токи и напряжения
4. **Логическая схема:**





# Электронные ключи

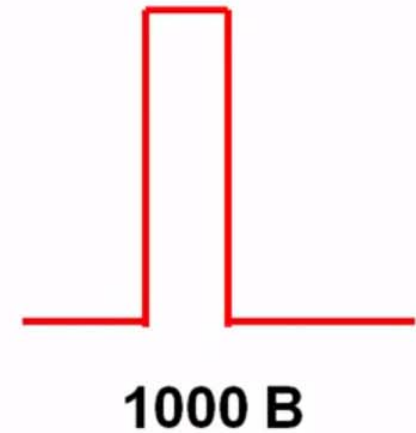
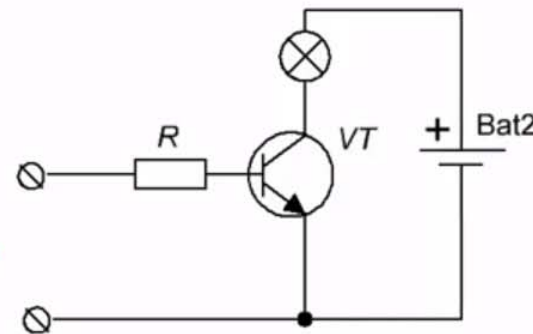
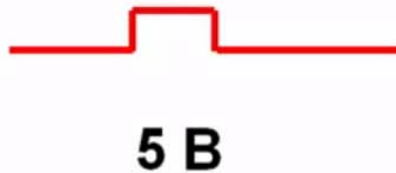
**Ключ** — электрический коммутационный аппарат, служащий для замыкания и размыкания электрической цепи.



## Виды ключей:

- Механический
- Электромагнитный
- Электронный

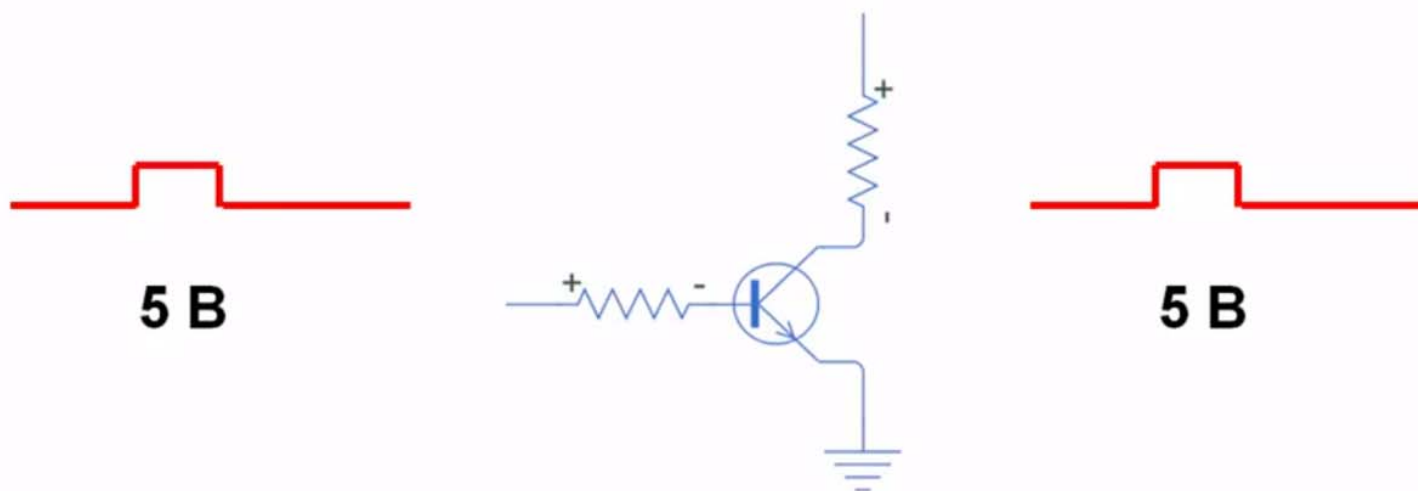
## Транзисторный ключ:



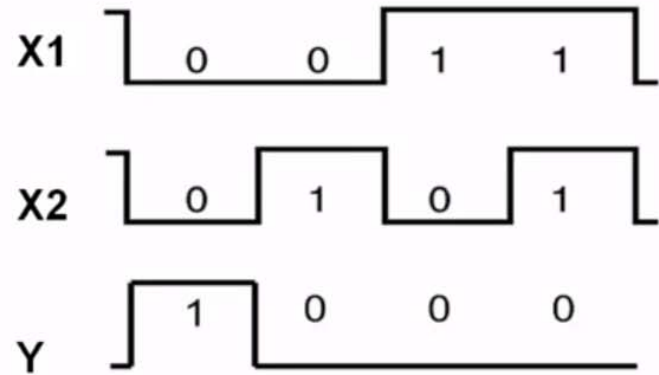


# Электронные ключи для логики?

Транзисторный ключ:



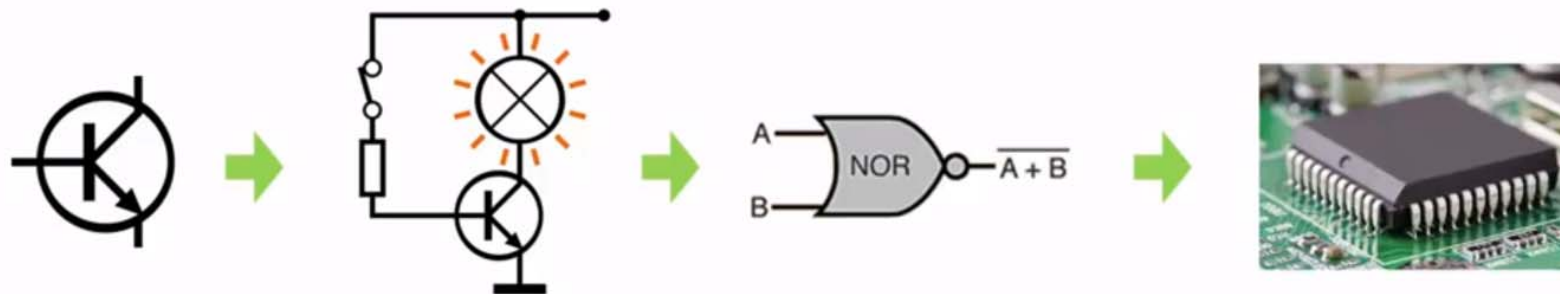
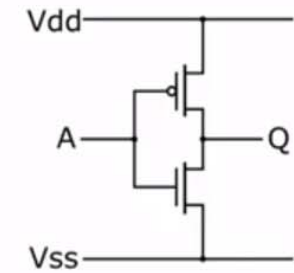
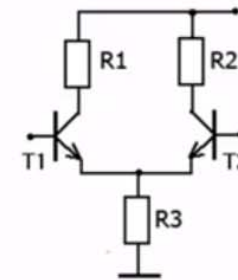
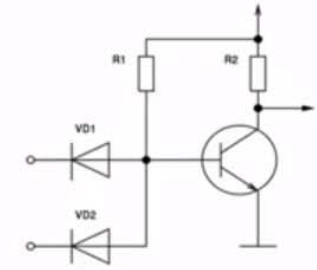
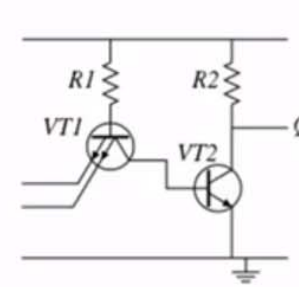
# Логические схемы



Вентиль ИЛИ-НЕ (NOR GATE)

## Виды логики:

- РТЛ
- ДТЛ
- ТТЛ (ТТЛШ)
- ЭСЛ
- КМОП
- проч



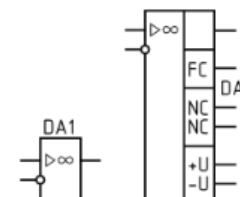
# Условные графические обозначения на принципиальных электрических схемах (УГО) ГОСТ 2.743-91

Реализуемая функция	Название	Схемное обозначение	
		ГОСТ	ANSI
НЕ	Инверсия		
ИЛИ	Сложение		
ИЛИ-НЕ	Стрелка Пирса		
И	Умножение		
И-НЕ	Функция Шеффера		
Исключающее ИЛИ	Сумма по модулю 2		

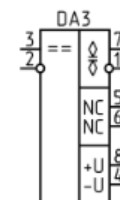
Таблица 1

ANSI		ГОСТ		Примечание
УГО	Функция	УГО	Функция	
	AND		И	$Z = X \wedge Y$
	OR		ИЛИ	$Z = X \vee Y$
	NOT		НЕ	$Z = \bar{X}$
	NAND		И-НЕ	$Z = \overline{X \wedge Y}$
	NOR		ИЛИ-НЕ	$Z = \overline{X \vee Y}$
	XOR		ИСКЛ ИЛИ	$Z = X \oplus Y$
	XNOR		ИСКЛ ИЛИ-НЕ	$Z = \overline{X \oplus Y}$
	TRISTATE BUFFER		Буфер с 3 состояниями выхода	$Z = X \cdot E$
	BUFFER		Буферный повторитель	$Z = X$

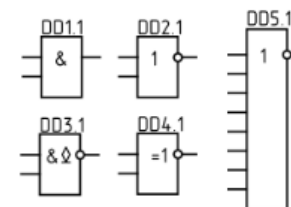
Operational amplifier  
Усилитель операционный



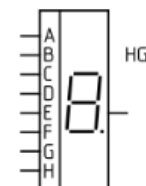
Comparator  
Компаратор



Logic gate  
Элементы логические



Seven-segment display  
Индикатор цифровой

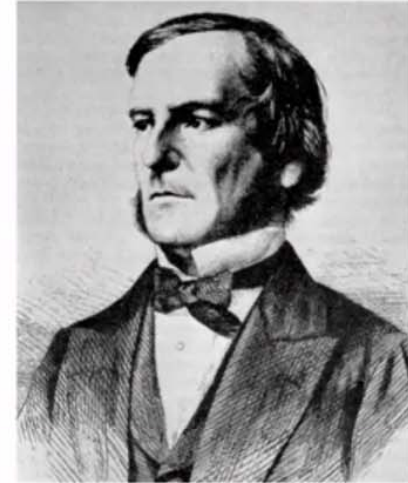


Цифровая схемотехника

# Базовые логические операции

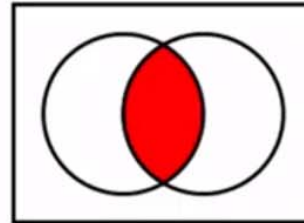
## Бинарная логика

**Алгебра логики** — раздел математической логики, в котором изучаются логические операции над *высказываниями*. Высказывания могут быть или *истинными* (логическая **единица**), или *ложными* (логический **ноль**).

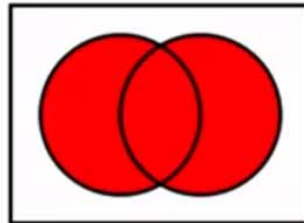


Некоторые логические операции:

$a \wedge b$ ,  $a \& \& b$ ,  $a \& b$ ,  $a \cdot b$ ,  $a$  AND  $b$ ,  $\min(a, b)$



$a \vee b$ ,  $a || b$ ,  $a | b$ ,  $a$  OR  $b$ ,  $\max(a, b)$ .



**Бинарная логика** была основным математическим инструментом при создании компьютеров!

# Логические операции

## Инверсия

**Отрицание** (инверсия, от лат. *inversio* — *переворот*, логическое «НЕ») в логике — унарная операция над суждениями, результатом которой является суждение, «противоположное» исходному. Обозначается знаком  $\neg$  перед или чертой  $\bar{\quad}$  над суждением.

Логическое НЕ (NOT GATE):

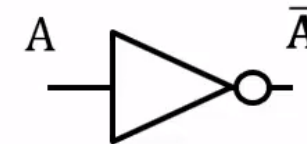
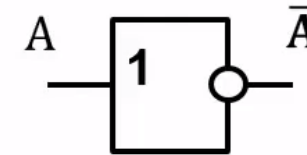


Таблица истинности:

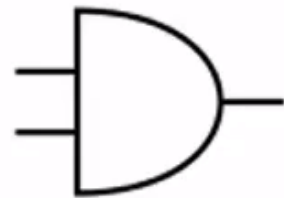
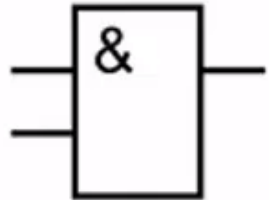
A	$\bar{A}$
0	1
1	0

# Логические операции

# Конъюнкция

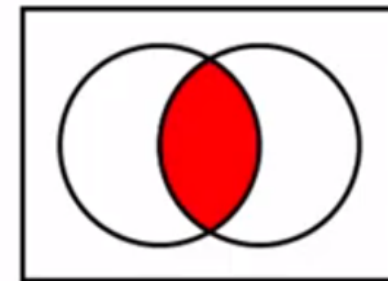
**Конъюнкция** (от лат. *conjunctio* — «союз, связь») — логическая операция, по смыслу максимально приближенная к союзу «и». Синонимы: **логическое «И»**, **логическое умножение**, иногда просто «И»

Логическое **И** (**AND GATE**):



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$a \wedge b$ ,  $a \& \& b$ ,  $a \& b$ ,  $a \cdot b$ ,  $a \text{ AND } b$ ,  $\min(a, b)$



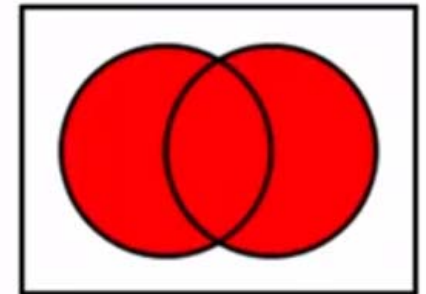


# Логические операции

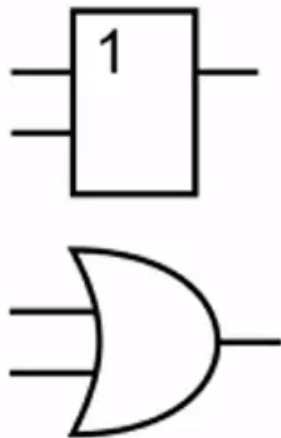
# Дизъюнкция

Дизъюнкция (от лат. *disjunctio* — «разобшение»), логическое сложение, логическое ИЛИ, включающее ИЛИ; иногда просто ИЛИ — логическая операция, по своему применению максимально приближённая к союзу «или» в смысле «или то, или это, или оба сразу»

$$a \vee b, a || b, a | b, a \text{ OR } b, \max(a, b).$$



Логическое ИЛИ (OR GATE):



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

# Логические операции

Логическое **НЕ** (**NOT GATE**):

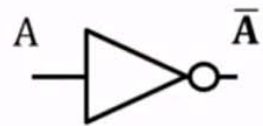
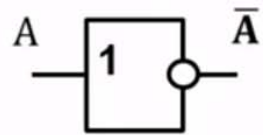
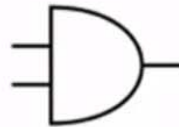
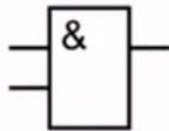


Таблица истинности:

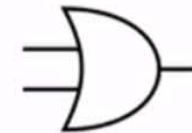
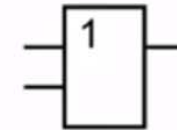
A	$\bar{A}$
0	1
1	0

Логическое **И** (**AND GATE**):



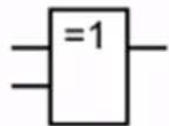
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Логическое **ИЛИ** (**OR GATE**):



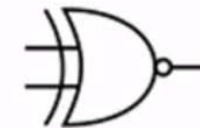
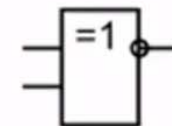
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Исключающее **ИЛИ** (**XOR GATE**):



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

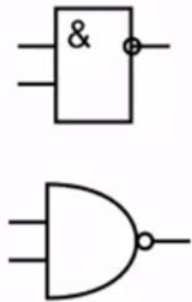
Исключающее **И** (**NXOR GATE**):



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

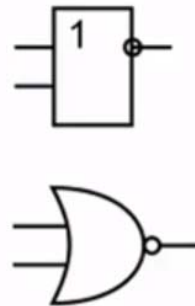
# Штрих Шеффера и стрелка Пирса

Штрих Шеффера, **И-НЕ**  
(**NAND GATE**):



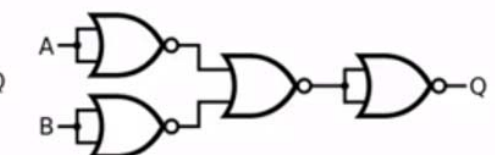
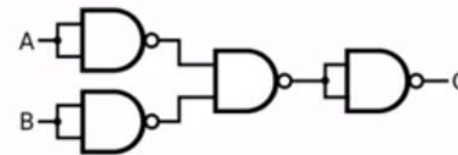
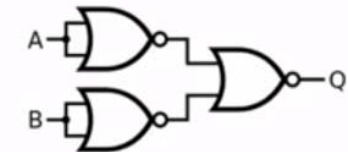
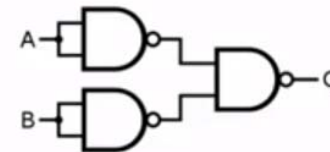
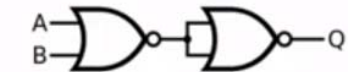
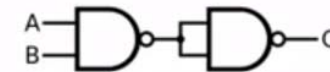
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Стрелка Пирса, **ИЛИ-НЕ**  
(**NOR GATE**):



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

**Базис для реализации любой логической схемы!**



# Классификация цифровых устройств



Цифровая схемотехника

# Комбинационные цифровые устройства КЦУ

## Комбинационные цифровые устройства

Комбинационные цифровые устройства (КЦУ) – устройства, в которых выходные сигналы определяются **только текущими** входными сигналами!

КЦУ – устройства **без памяти**

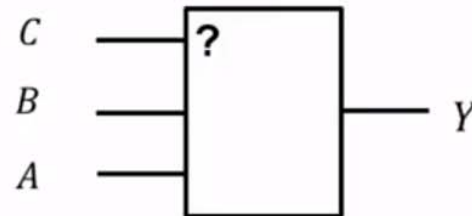


Таблица истинности устройства:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

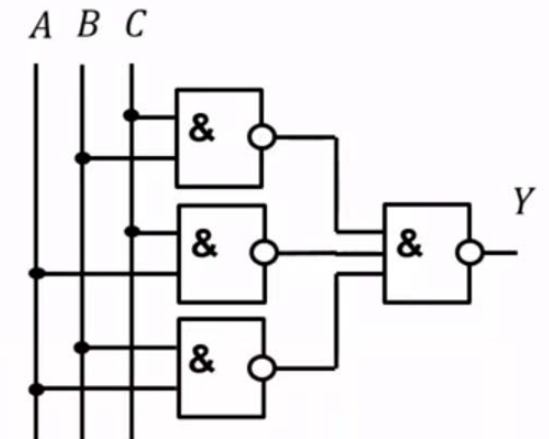
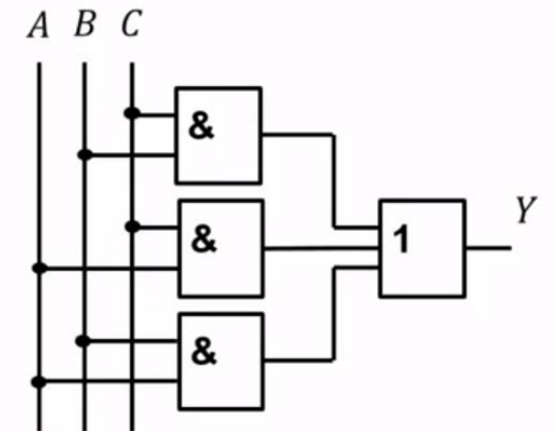
## Синтез КЦУ

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Минимизация логических выражений
- Карты Карно (диаграммы Вейча)

$$Y = BC + AC + AB$$

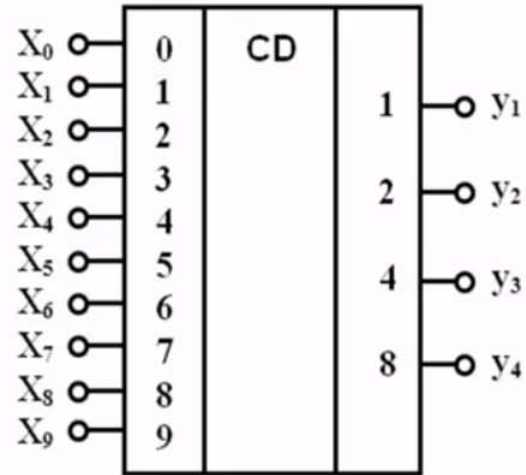
$$Y = \overline{\overline{BC} \cdot \overline{AC} \cdot \overline{AB}}$$



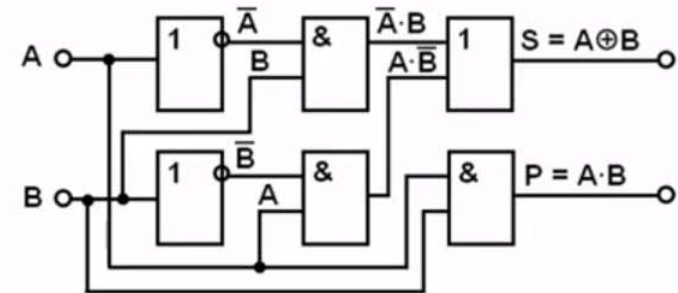
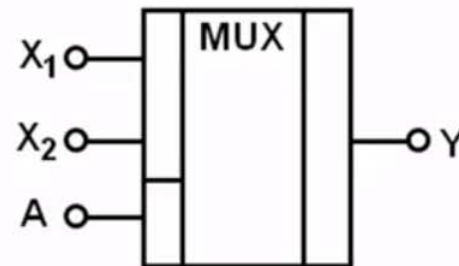


## Типичные КЦУ

- Шифраторы
- Дешифраторы
- Мультиплексоры
- Демультимплексоры
- Сумматоры
- Умножители
- Компараторы
- и проч.



Десятичное число	Двоичный код 8421			
	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

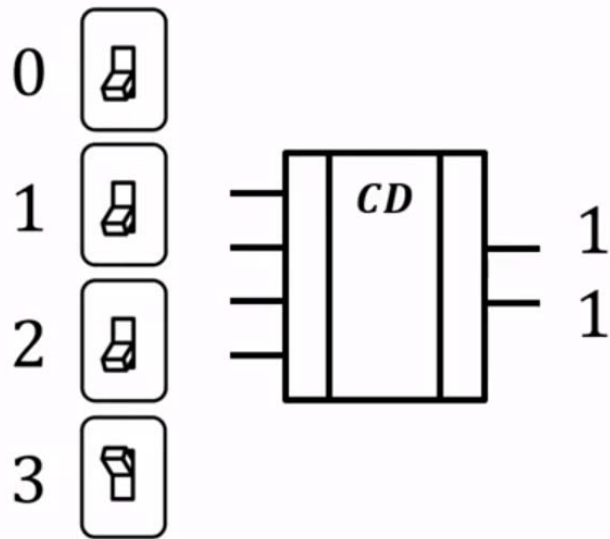


Цифровая схемотехника

# Шифраторы и дешифраторы

## Шифратор

**Шифратор (кодер)** - (англ. encoder) логическое устройство, выполняющее логическую функцию (операцию) преобразования позиционного n-разрядного кода в m-разрядный двоичный, троичный или k-ричный код.



A	B	C	D	Y1	Y0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Y1 = A + B$$

$$Y0 = A + C$$

# Приоритетный шифратор

A	B	C	D	Y1	Y0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

	A	$\bar{A}$	
B	1	1	$\bar{D}$
	1	1	D
$\bar{B}$	1	1	$\bar{D}$
	1	1	
	$\bar{C}$	C	$\bar{C}$

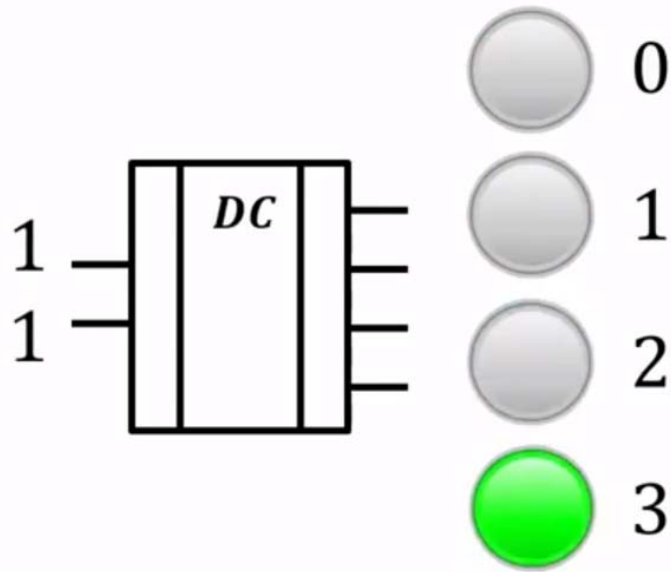
$$Y1 = A + B$$

	A	$\bar{A}$	
B	1	0	$\bar{D}$
	1	0	D
$\bar{B}$	1	1	$\bar{D}$
	1	1	
	$\bar{C}$	C	$\bar{C}$

$$Y0 = A + \bar{B}C$$

## Дешифратор

**Дешифратор (декодер)** — (англ. decoder) логическое устройство, преобразующее  $n$ -разрядный двоичный код в логический сигнал, появляющийся на том выходе, десятичный номер которого соответствует двоичному коду.



A	B	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

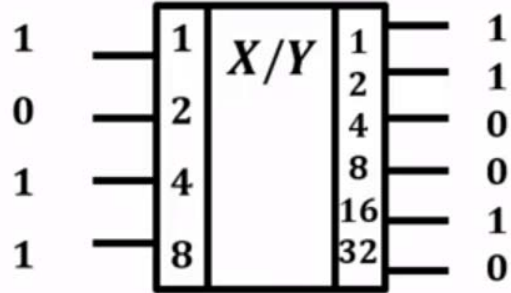
$$Y3 = AB$$

$$Y2 = \bar{A}B$$

$$Y1 = A\bar{B}$$

$$Y0 = \bar{A}\bar{B}$$

# Преобразователь кодов



A	B	C	Y2	Y1	Y0	DEC
0	0	0	0	1	0	2
0	0	1	1	1	0	6
0	1	0	0	0	1	1
0	1	1	1	0	1	5
1	0	0	0	0	1	1
1	0	1	1	0	0	4
1	1	0	0	1	1	3
1	1	1	1	1	0	6

	A	$\bar{A}$
B	0	1
$\bar{B}$	0	1
$\bar{C}$	C	

$$Y2 = C$$

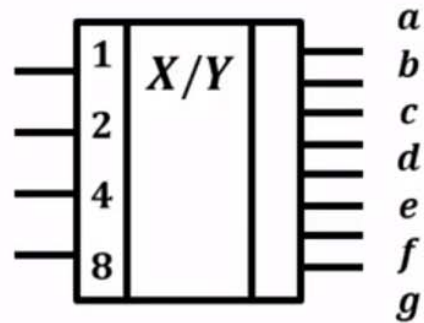
	A	$\bar{A}$
B	1	0
$\bar{B}$	0	1
$\bar{C}$	C	

$$Y1 = AB + \bar{A}\bar{B}$$

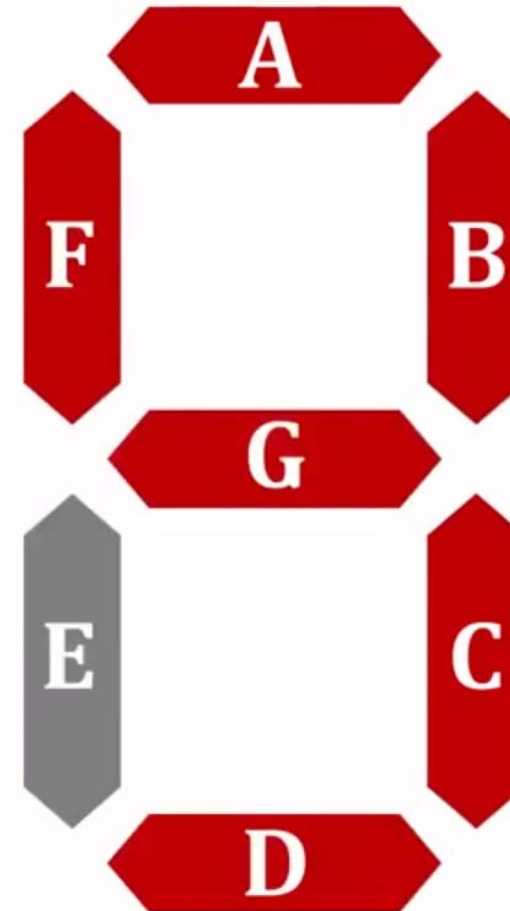
	A	$\bar{A}$
B	0	1
$\bar{B}$	0	0
$\bar{C}$	C	

$$Y0 = \overline{\bar{A}\bar{B}} \cdot AC$$

# Код для семисегментного индикатора



DEC	X3	X2	X1	X0	A	B	C	D	E	F	G
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



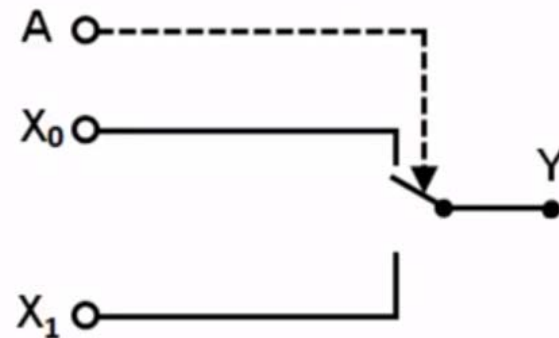
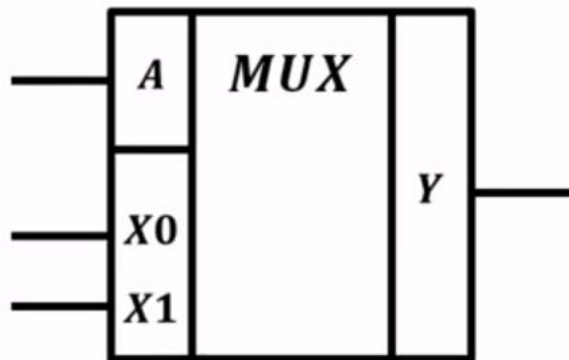


Цифровая схемотехника

# Мультиплексоры и демультиплексоры

# Мультиплексор

**Мультиплéксор** \* — устройство, имеющее несколько сигнальных входов, один или более управляющих входов и один выход. Мультиплексор позволяет передавать сигнал с *одного* из входов на выход; при этом выбор желаемого входа осуществляется подачей соответствующей комбинации управляющих сигналов.



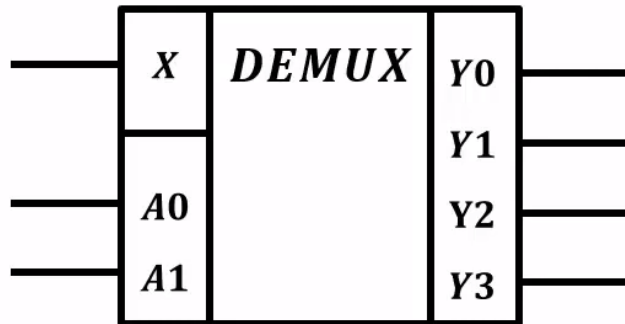
A	X1	X0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

	A	$\bar{A}$	
X1	1	1	0
$\bar{X1}$	0	0	0
	$\bar{X0}$	X0	$\bar{X0}$

$$Y = A \cdot X1 + \bar{A} \cdot X0$$

# Демультиплексор

**Демультиплексор** — это логическое устройство, предназначенное для переключения сигнала с одного информационного входа на один из нескольких информационных выходов.



A1	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

A1	A0	X	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

$$Y3 = A1 \cdot A0 \cdot X$$

$$Y2 = A1 \cdot \overline{A0} \cdot X$$

$$Y1 = \overline{A1} \cdot A0 \cdot X$$

$$Y0 = \overline{A1} \cdot \overline{A0} \cdot X$$

Цифровая схемотехника  
**Арифметические устройства**  
**Сумматоры**

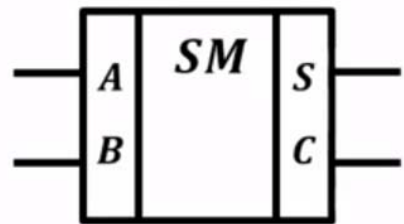
# Полу-сумматор

Сумматоры — это комбинационные устройства, предназначенные для сложения чисел.

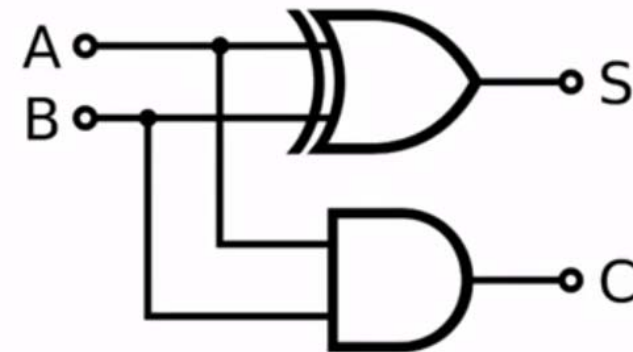
$$\begin{array}{r} 1\ 1 \\ +\ 11 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 1 \\ +\ 1\ A \\ \hline 1\ B \\ C\ S \end{array}$$

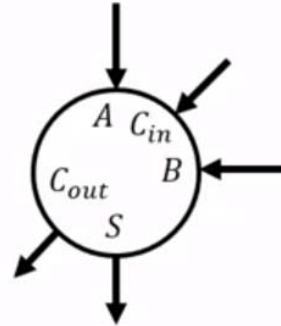
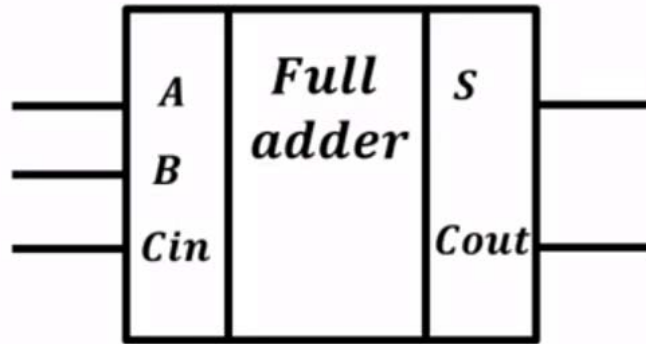
S — сумма  
C — перенос (англ. **carry**)



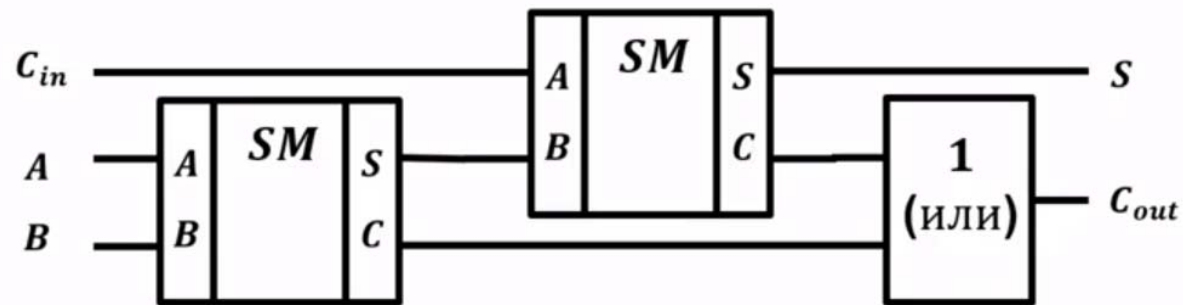
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



# Полный сумматор



Cin	A	B	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Полные сумматоры двух бит можно соединять в **каскады**, и таким образом получать сумматоры для чисел **любой битности!**

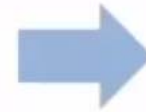
Цифровая схемотехника  
**Арифметические устройства**  
**Умножители**



## Умножение двоичных чисел

$$\begin{array}{r}
 \times 1101 \quad 13 \\
 101 \quad 5 \\
 \hline
 + 1101 \\
 + 0000 \\
 + 110100 \\
 \hline
 1000001 \quad 65
 \end{array}$$

$$\begin{array}{r}
 \times 11 \\
 11 \\
 \hline
 + 11 \\
 11 \\
 \hline
 1001
 \end{array}$$

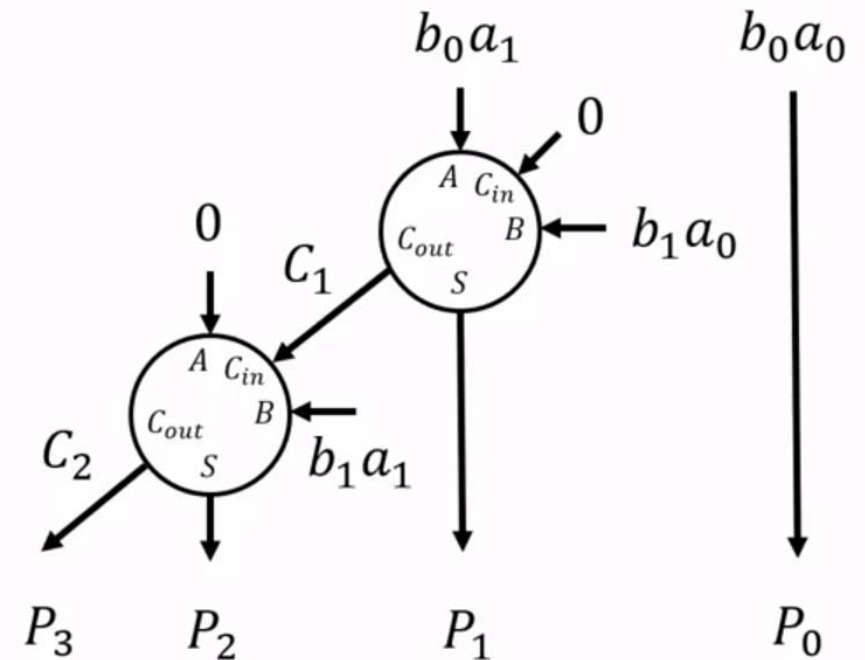
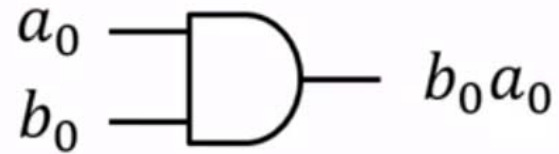


$$\begin{array}{r}
 \times \begin{array}{cc} a_1 & a_0 \\ b_1 & b_0 \end{array} \\
 \hline
 \begin{array}{cc} b_0 a_1 & b_0 a_0 \\ + & \\ b_1 a_1 & b_1 a_0 \end{array} \\
 \hline
 \begin{array}{cccc} c_2 & b_1 a_1 & b_0 a_1 & b_0 a_0 \\ & + & + & \\ & c_1 & b_1 a_0 & \end{array}
 \end{array}$$

Diagram illustrating the bit-level multiplication process. The top part shows the multiplication of two 2-bit numbers  $\begin{matrix} a_1 & a_0 \\ b_1 & b_0 \end{matrix}$ . The middle part shows the partial products:  $b_0 a_1$  and  $b_0 a_0$  in the first row, and  $b_1 a_1$  and  $b_1 a_0$  in the second row, with a plus sign between them. The bottom part shows the final result with carry bits:  $c_2$  above  $b_1 a_1$ ,  $c_1$  above  $b_1 a_0$ , and the bits  $b_1 a_1$ ,  $b_0 a_1$ , and  $b_0 a_0$  in their respective positions. Blue curved arrows indicate the carry propagation from  $b_1 a_0$  to  $c_1$  and from  $c_1$  and  $b_0 a_1$  to  $c_2$ .

# Умножение двухбитных чисел

$$\begin{array}{r}
 \begin{array}{r}
 \times \quad a_1 \quad a_0 \\
 \quad b_1 \quad b_0 \\
 \hline
 \end{array} \\
 + \quad \begin{array}{r}
 \quad b_0 a_1 \quad b_0 a_0 \\
 b_1 a_1 \quad b_1 a_0 \\
 \hline
 \end{array} \\
 \hline
 \begin{array}{r}
 C_2 \quad b_1 a_1 \quad b_0 a_1 \quad b_0 a_0 \\
 \quad + \quad + \\
 \quad C_1 \quad b_1 a_0 \\
 \hline
 \end{array} \\
 \hline
 \begin{array}{r}
 P_3 \quad P_2 \quad P_1 \quad P_0
 \end{array}
 \end{array}$$



## Умножение трёхбитных чисел

$$\begin{array}{r}
 \times 111 \\
 \hline
 111 \\
 + 111 \\
 + 111 \\
 \hline
 110001
 \end{array}$$

$$\begin{array}{r}
 \begin{array}{r}
 a_2 \quad a_1 \quad a_0 \\
 \times \\
 b_2 \quad b_1 \quad b_0 \\
 \hline
 b_0 a_2 \quad b_0 a_1 \quad b_0 a_0 \\
 + \\
 b_1 a_2 \quad b_1 a_1 \quad b_1 a_0 \\
 + \\
 b_2 a_2 \quad b_2 a_1 \quad b_2 a_0 \\
 \hline
 \end{array} \\
 P_5 \quad P_4 \quad P_3 \quad P_2 \quad P_1 \quad P_0
 \end{array}$$

## Умножение трёхбитных чисел

$$\begin{array}{r}
 \begin{array}{r}
 \times \quad a_2 \quad a_1 \quad a_0 \\
 \quad b_2 \quad b_1 \quad b_0 \\
 \hline
 \quad b_0 a_2 \quad b_0 a_1 \quad b_0 a_0 \\
 + \quad b_1 a_2 \quad b_1 a_1 \quad b_1 a_0 \\
 \hline
 \quad C_3^1 \quad S_3^1 \quad S_2^1 \quad S_1^1 \quad b_0 a_0 \\
 + \quad b_2 a_2 \quad b_2 a_1 \quad b_2 a_0 \\
 \hline
 C_4^2 \quad P_4 \quad P_3 \quad P_2 \quad S_1^1 \quad b_0 a_0
 \end{array}
 \end{array}$$

$$S_0^1 = b_0 a_0$$

$$S_1^1 = b_0 a_1 + b_1 a_0$$

$$S_2^1 = b_0 a_2 + b_1 a_1 + C_1^1$$

$$S_3^1 = b_1 a_2 + C_2^1$$

$$S_4^1 = C_3^1$$

$$P_1 = S_1^1$$

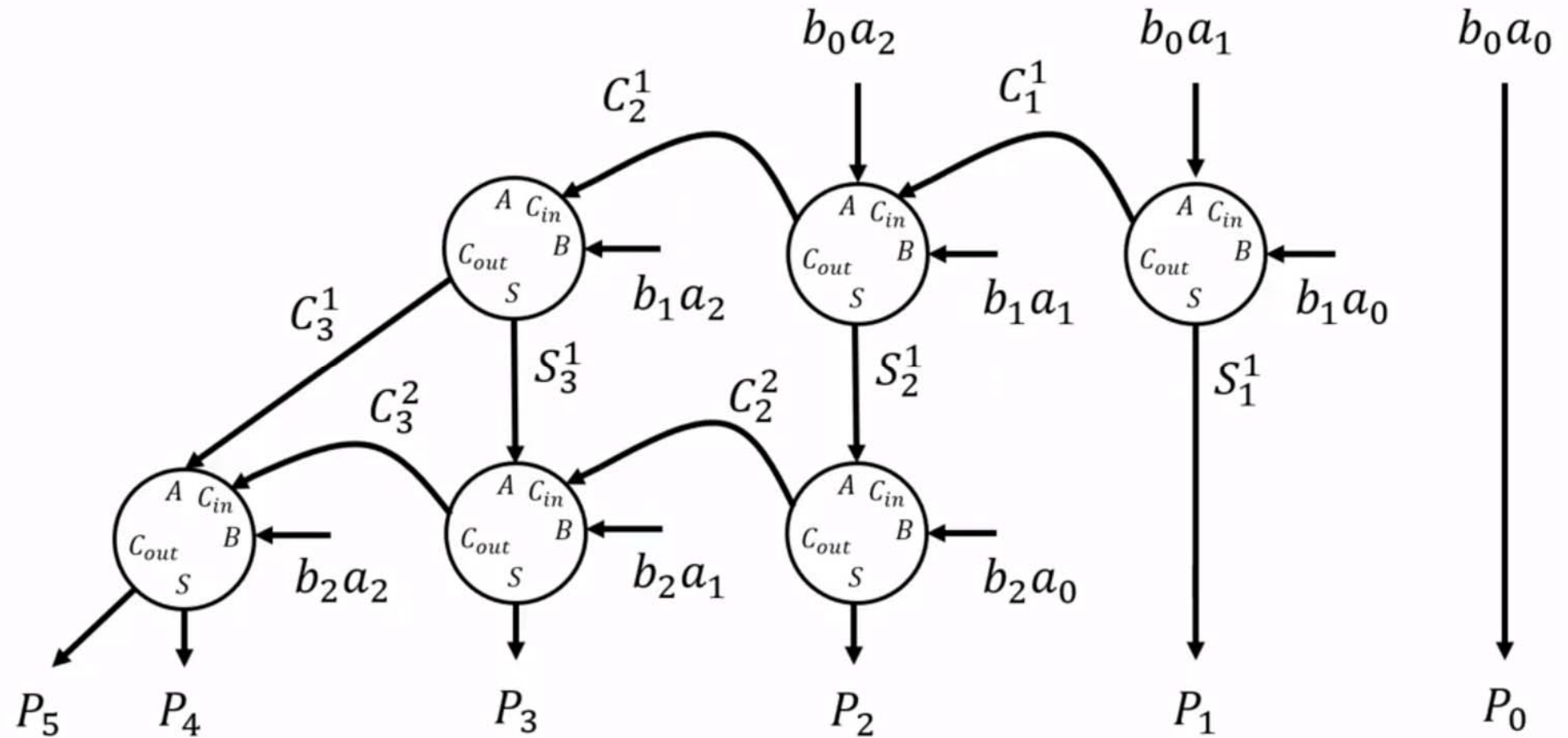
$$P_2 = S_2^1 + b_2 a_0$$

$$P_3 = S_3^1 + b_2 a_0 + C_2^2$$

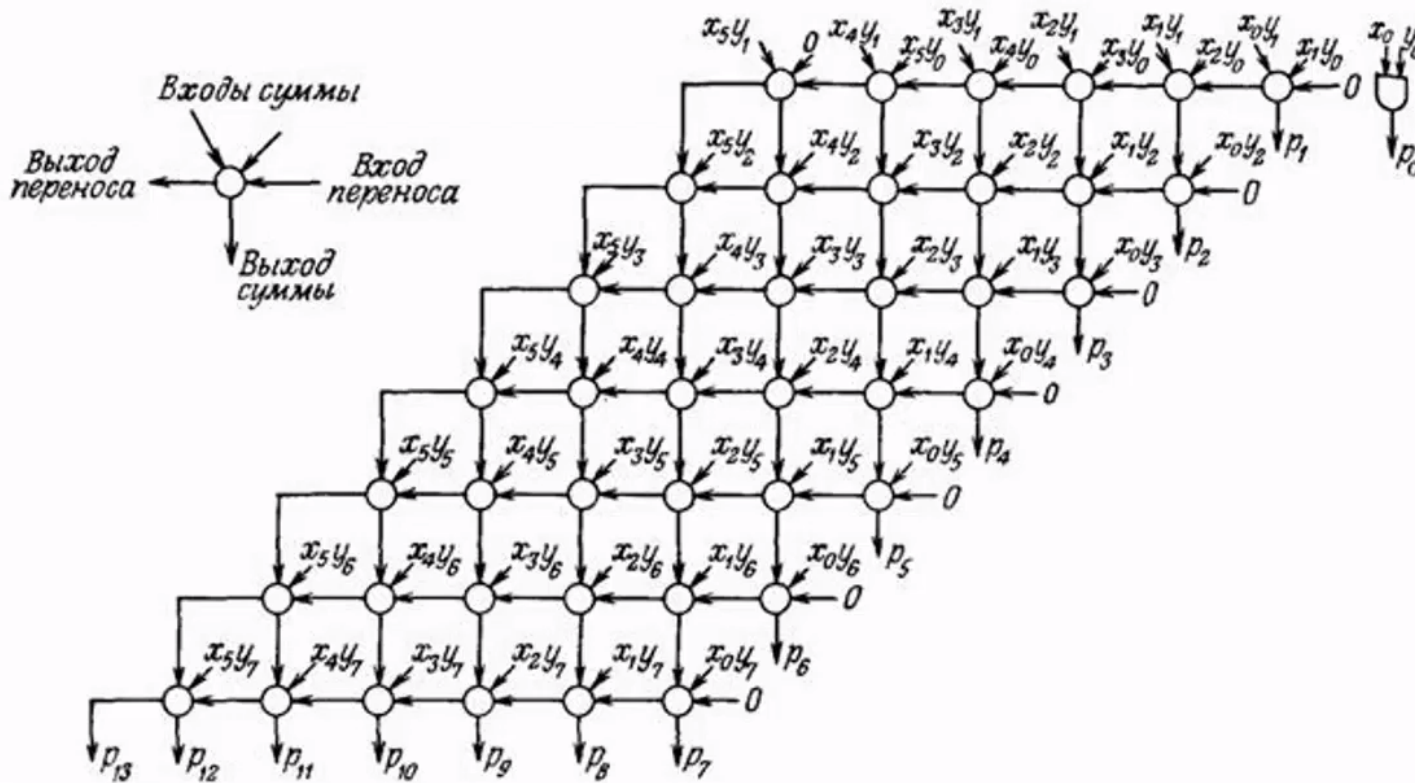
$$P_4 = C_3^1 + b_2 a_2 + C_3^2$$

$$P_5 = C_4^2$$

# Умножение трёхбитных чисел



# Матричные умножители



Чтобы перемножить два 8-Битных числа потребуется:

**56** полных сумматора

**112** полу-сумматоров

**1136** элементов 2И-НЕ

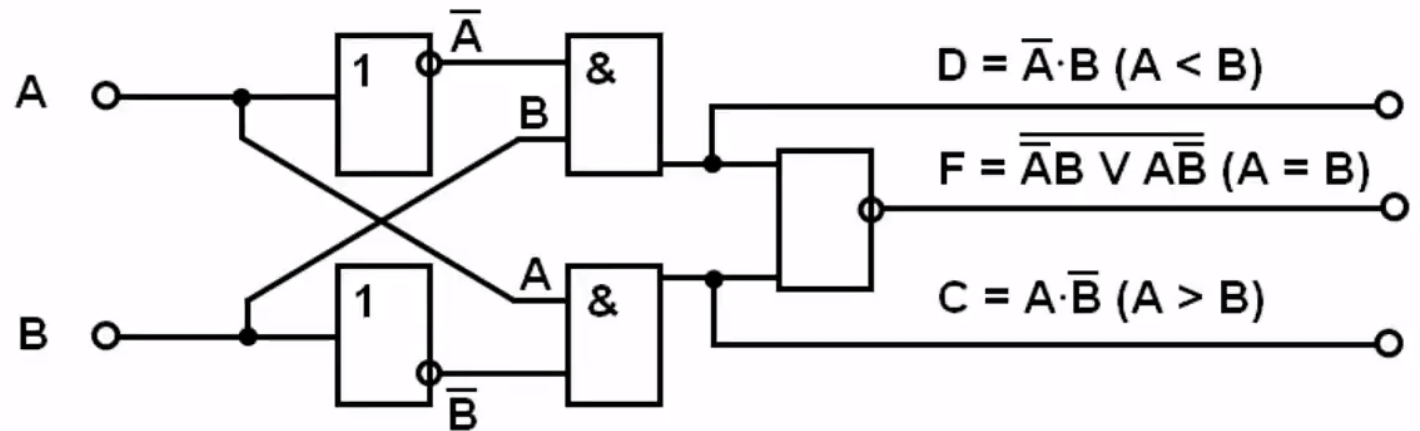
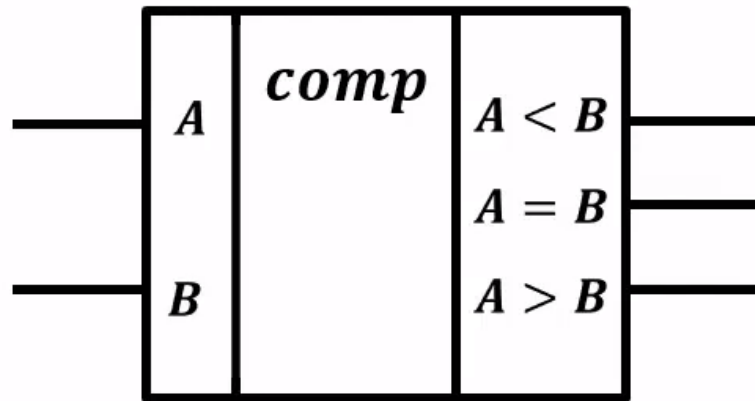
**4544** КМОП транзисторов

Цифровая схемотехника  
**Арифметические устройства**  
**Компараторы**

# Компаратор

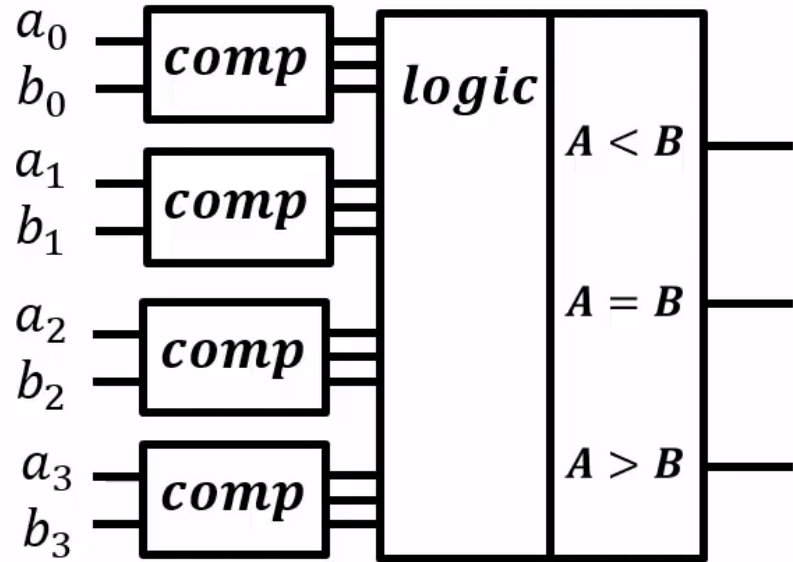
**Цифровой компаратор** или **компаратор кодов** - логическое устройство с двумя словарными входами, на которые подаются два разных двоичных слова равной в битах длины и обычно с тремя двоичными выходами, на которые выдаётся признак сравнения входных слов, — первое слово больше второго, меньше или слова равны.

A	B	A < B	A = B	A > B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0





# Наращивание разрядности компараторов



1101

|| || || ||

1101

**A = B**

**A > B ?**

A **0011**  
 || ||  
 B 0001

A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Цифровая схемотехника  
**Последовательные  
цифровые устройства  
ПЦУ**

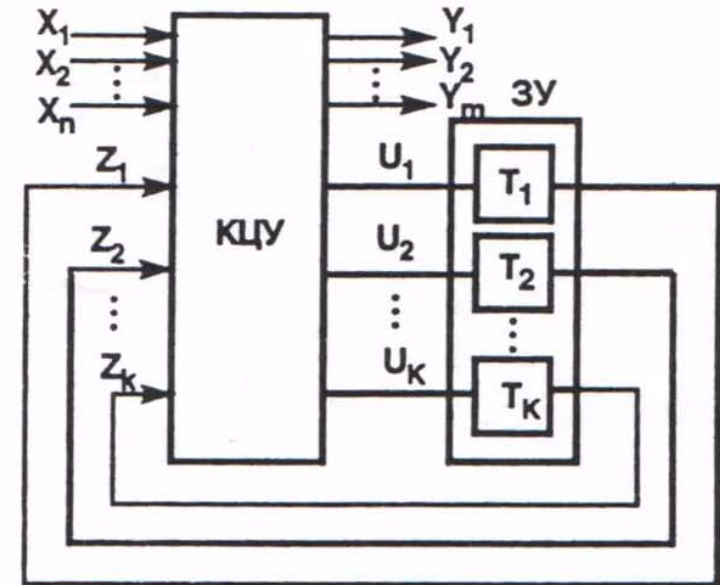
## Последовательные устройства

Последовательные (секвенциальные) цифровые устройства (**ПЦУ**) – цифровые устройства **с памятью**, в которых выходные сигналы зависят не только от текущих значений входных сигналов, но и от последовательности значений входных сигналов, поступивших на входы в предшествующие моменты времени



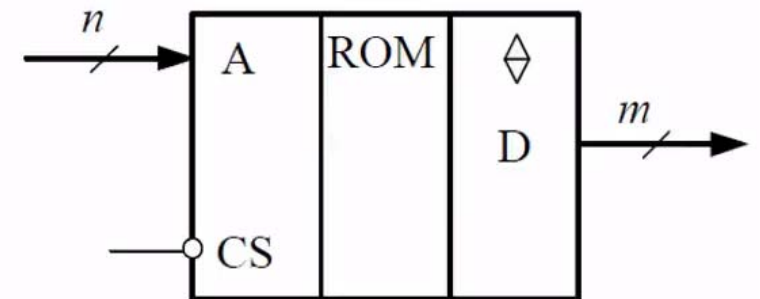
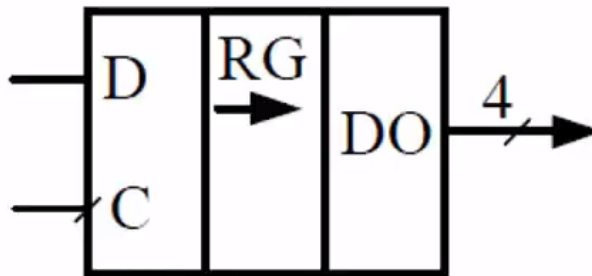
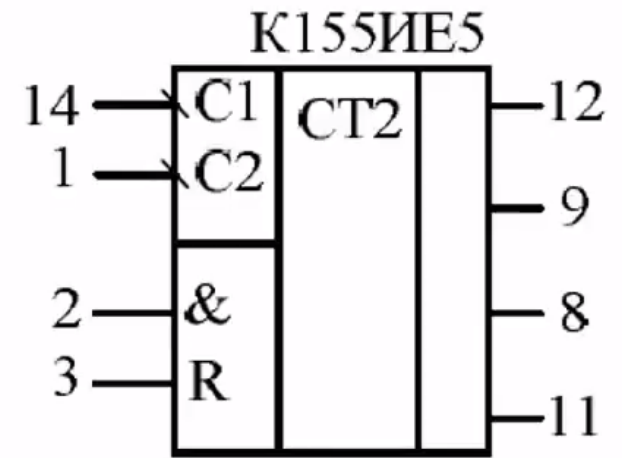
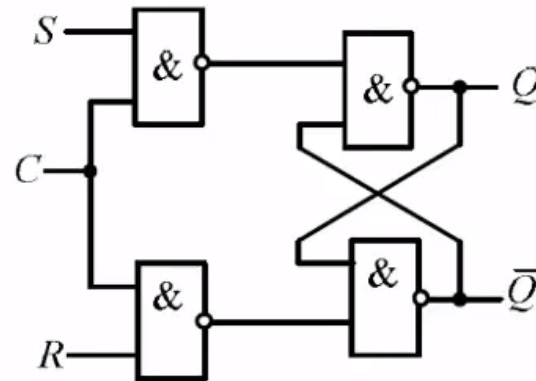
### Отличия от КЦУ:

- Устойчивые состояния
- Чувство времени



## Типичные ПЦУ

- Триггеры
- Регистры
- Счётчики
- Память
- и проч.

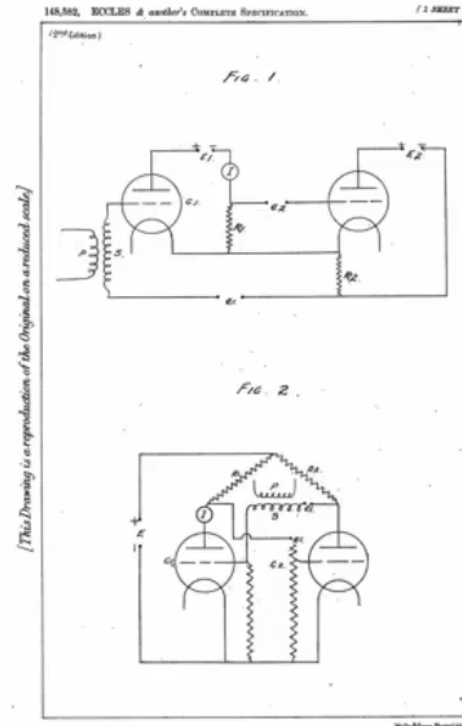
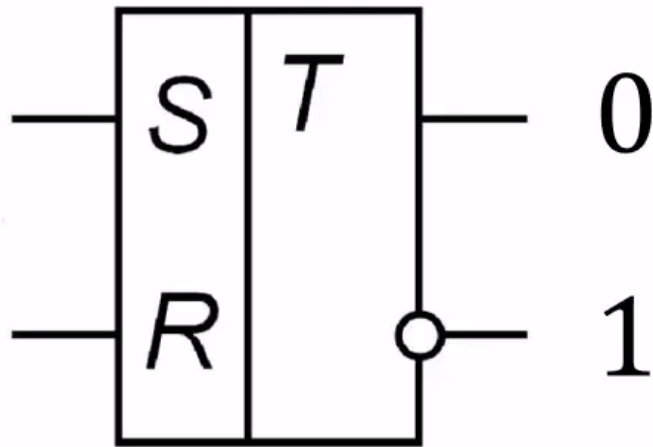


Цифровая схемотехника  
**Триггеры, Регистры**

# Цифровые триггеры

Цифровой триггер – это последовательное устройство с **двумя** устойчивыми состояниями равновесия, предназначенное для записи и хранения информации. Под действием входных сигналов триггер может переключаться из одного устойчивого состояния в другое.

Цифровой триггер:



## Классификация триггеров

### ФУНКЦИОНАЛЬНАЯ:

- RS
- JK
- D
- T
- Комбинации

### ПО СПОСОБУ ЗАПИСИ ИНФОРМАЦИИ:

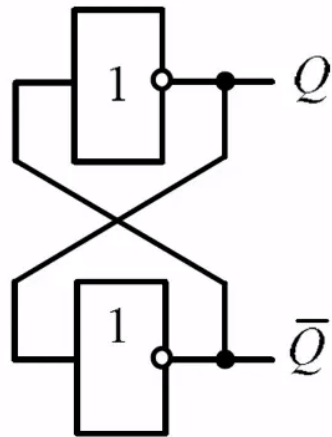
- Асинхронные
- Синхронные
- Комбинации

### ПО СТРУКТУРЕ:

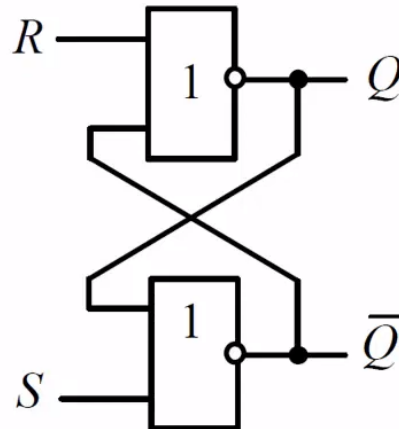
- Одноступенчатые
- Двухступенчатые
- Сложной структуры

# Асинхронный RS-триггер (защёлка)

Бистабильная ячейка:



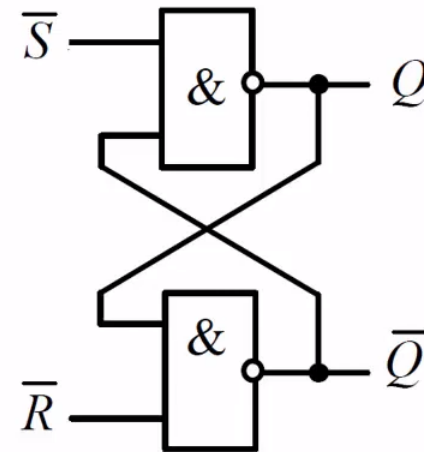
RS-защёлка на  
элементах 2ИЛИ-НЕ:



S – set, установка

R – reset, сброс

SR-защёлка на  
элементах 2И-НЕ:



«активный» уровень – 0

«неактивный» – 1



## Защёлка или триггер?

### ЗАЩЁЛКА (LATCH)

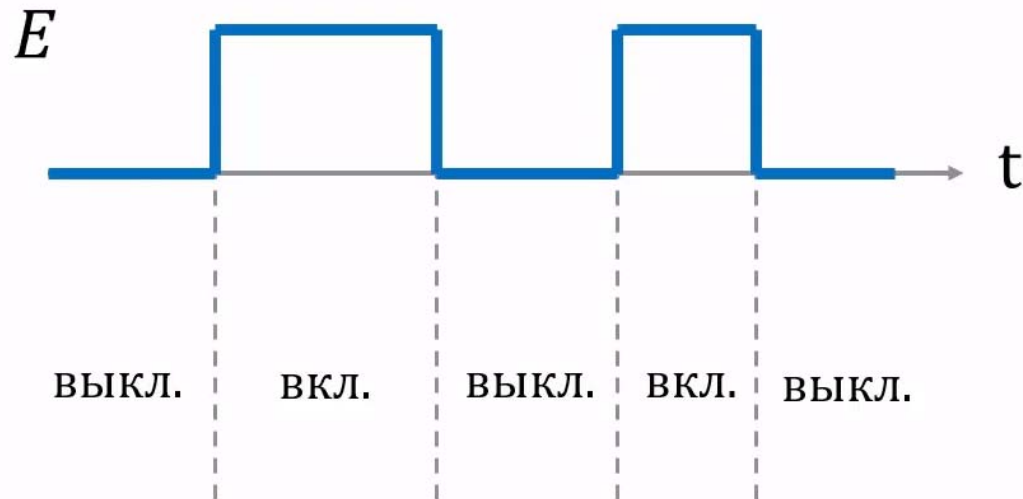
- Асинхронна
- Управляется уровнем
- «Прозрачна» - изменения сигналов на входах могут приводить к моментальным изменениям выхода
- Может хранить несколько бит информации

### ТРИГГЕР (FLIP-FLOP)

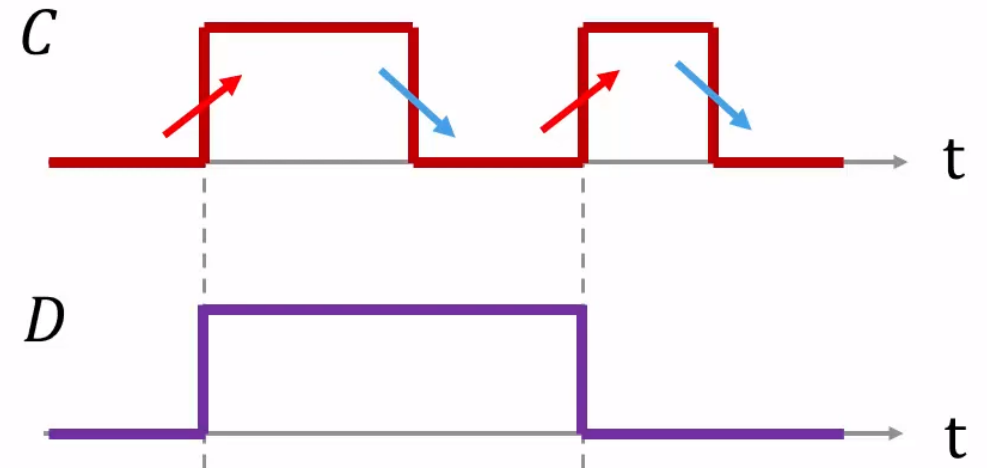
- Синхронен
- Управляется фронтом
- Изменения состояния могут происходить только в моменты перепада управляющего сигнала С
- Хранит 1 бит информации

# Статическое и динамическое управление

Статическое управление –  
реакция на **уровень** сигнала

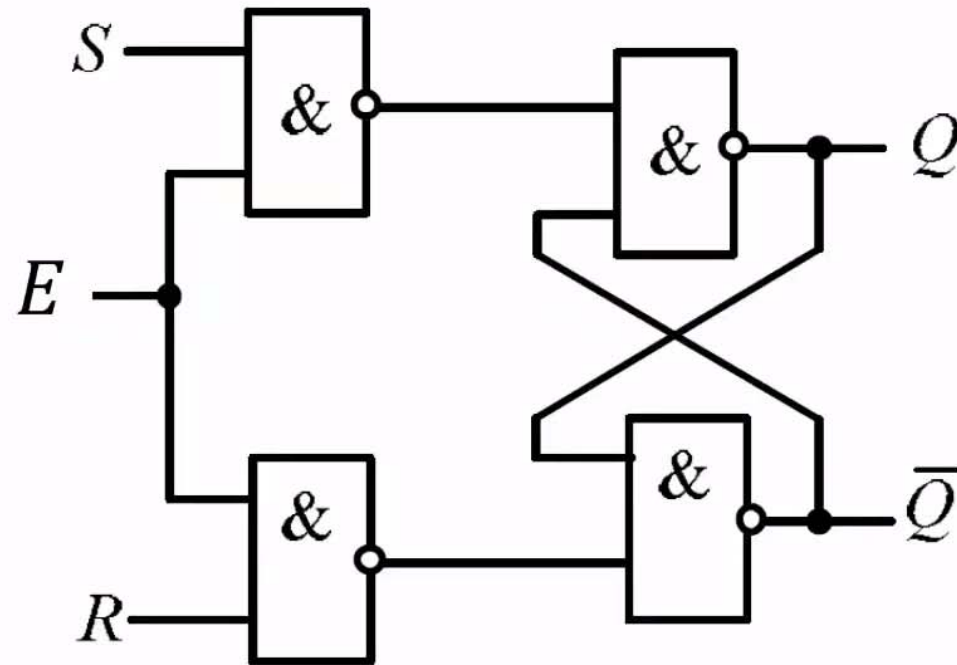


Динамическое управление –  
реакция на **перепад** сигнала



\* сигнал **D** переключается по  
положительному перепаду сигнала **C**

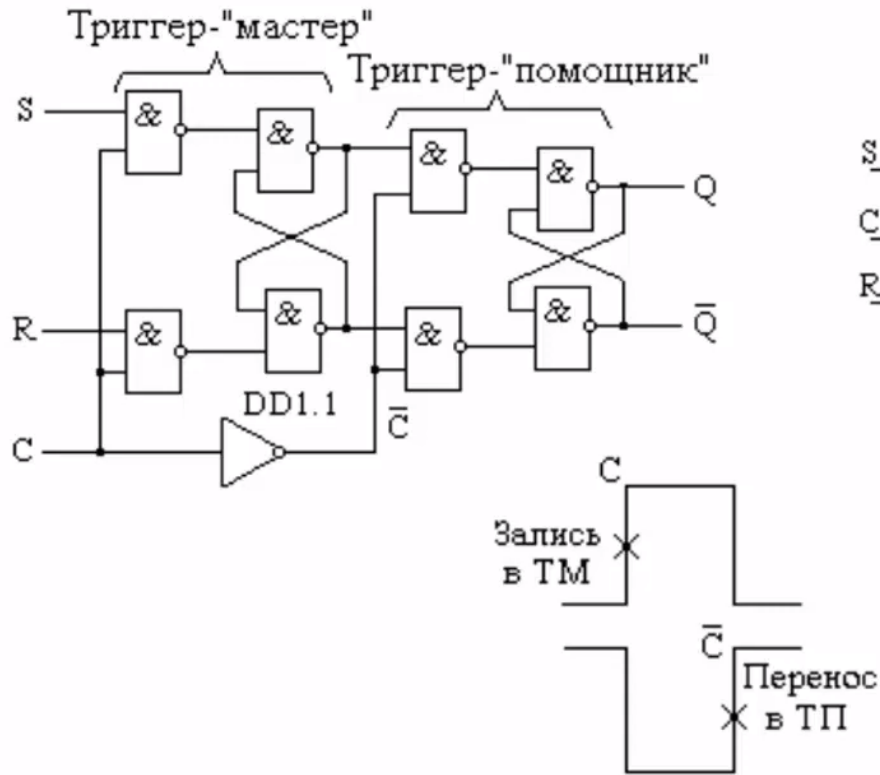
## Управляемая RS-защёлка (gated latch)



- Вход **E** (англ. **enable**) – это кнопка **ВКЛ/ВЫКЛ** для защёлки
- На вход **E** можно подать сигнал синхронизации (**ТАКТОВЫЙ** сигнал), но обычно на него поступают стробы чтения/записи
- Традиционно в литературе называется «синхронным RS-триггером», что не совсем верно по сути

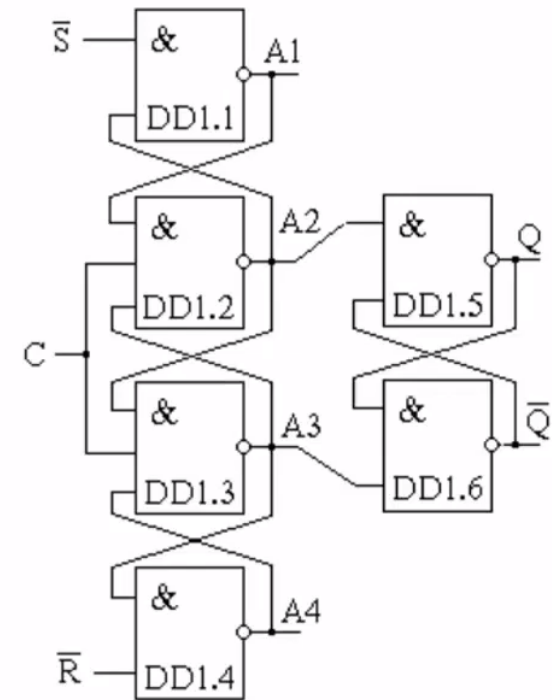
# Синхронные RS-триггеры (flip-flops)

Двухступенчатый RS-триггер:



RS-триггер с

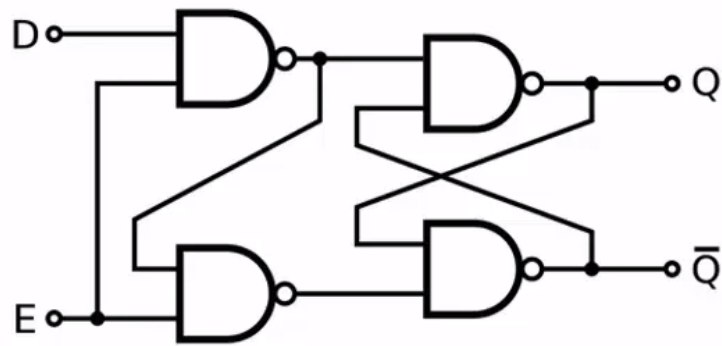
динамическим управлением:



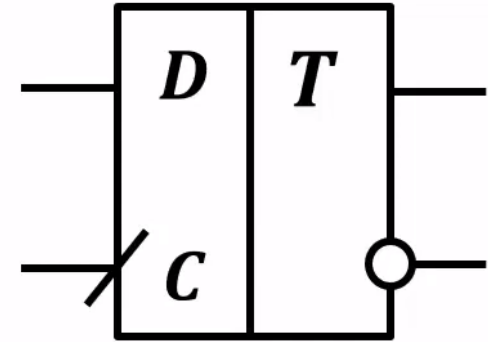
## D-триггер

**D-триггер** (**D** от англ. *delay* — задержка, либо от *data* — данные) — запоминает состояние входа и выдаёт его на выход. Имеет минимум два входа – данных и синхронизации.

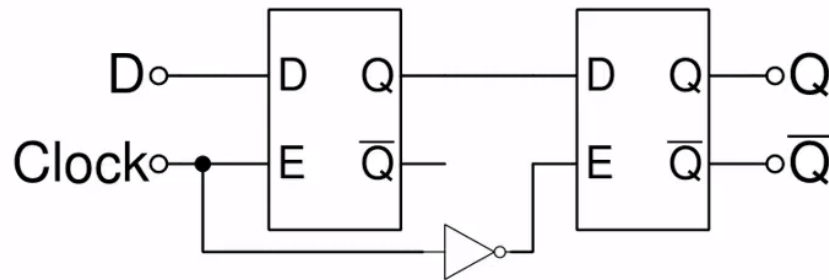
Управляемая D-защёлка:



**D-триггер** с динамическим управлением по **положительному** перепаду:

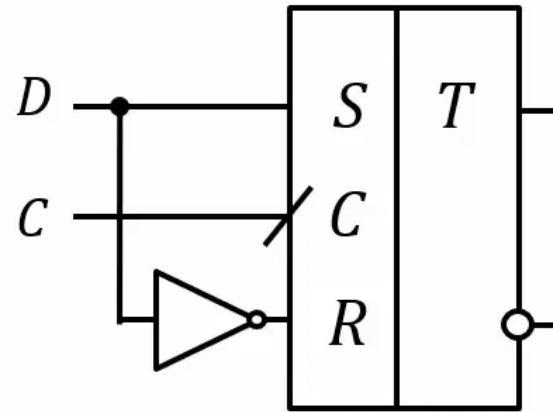
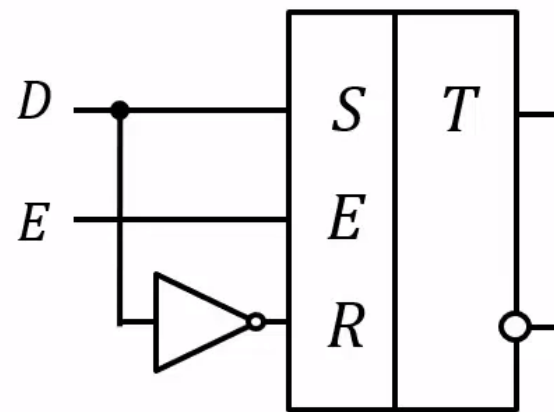
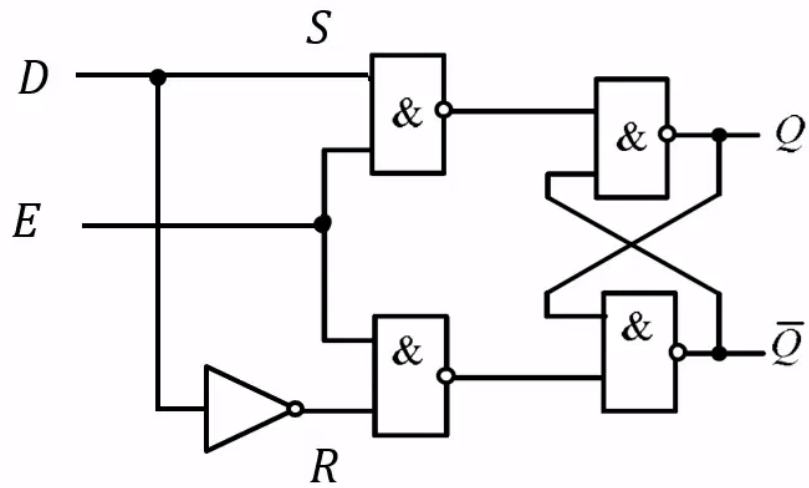


Двухступенчатый D-триггер:

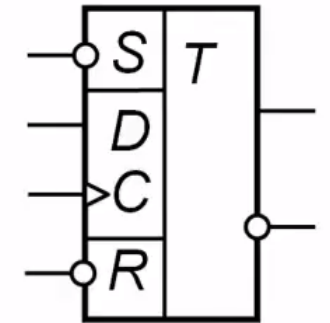
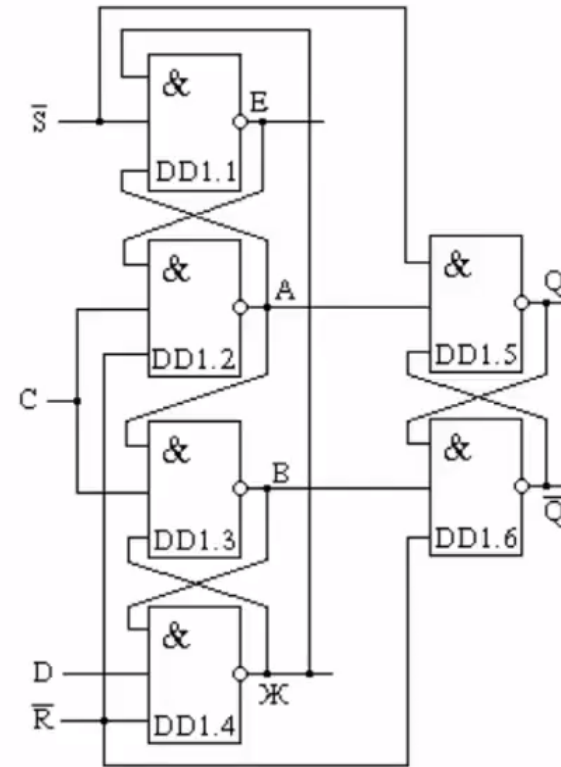


**D-триггеры** с динамическим управлением являются самыми распространёнными типами триггеров, на их основе строятся многие ПЦУ, они входят в состав большинства ПЛИС

# D-триггер на основе SR-триггера



**D-триггер с дополнительными асинхронными инверсными входами S и R:**

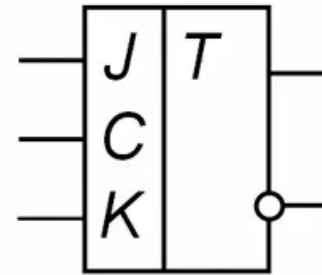


## JK-триггер

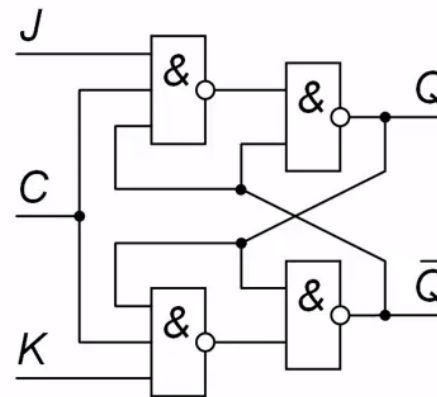
**JK-триггер** – наиболее универсальный из всех. На его основе можно построить D-триггер и T-триггер. Применяется только синхронный, с динамическим управлением.

Таблица перехода JK-триггера:

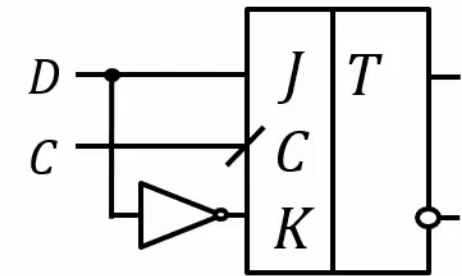
J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



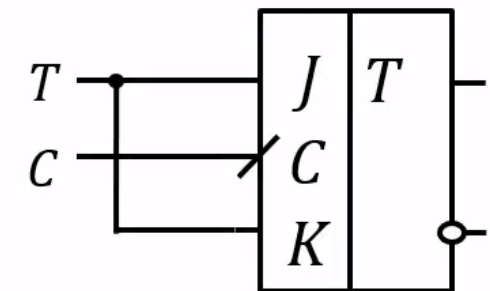
JK-триггер на основе RS-триггера:



D-триггер на основе JK-триггера:



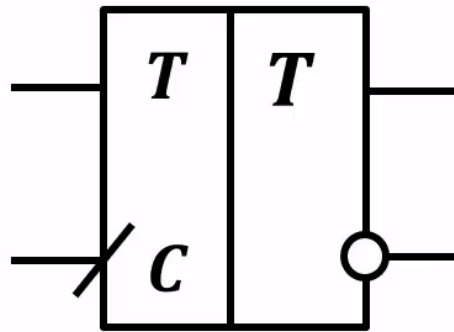
T-триггер на основе JK-триггера:



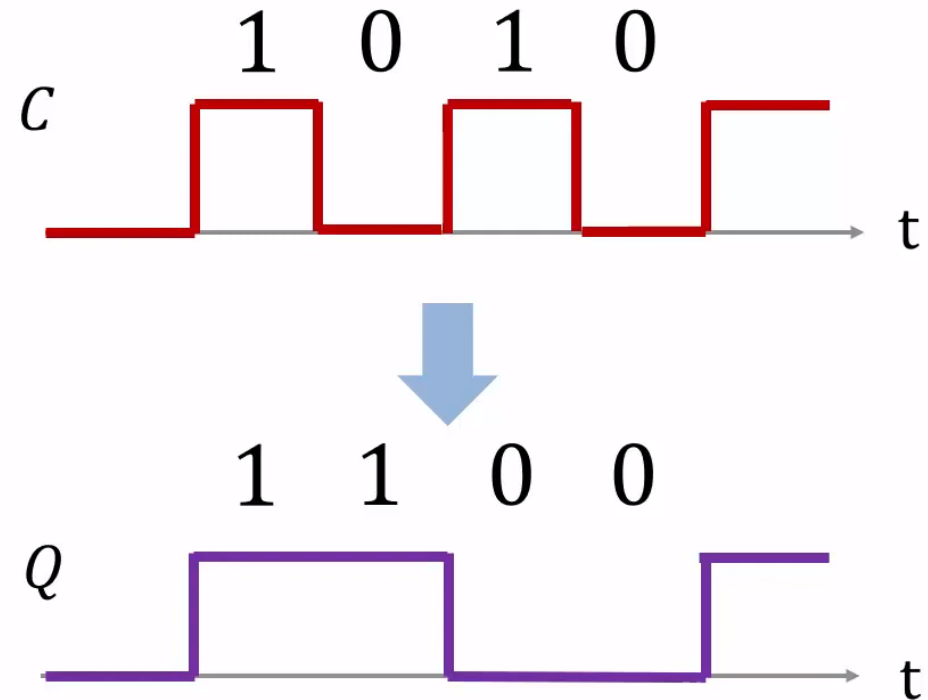


## T-триггер

**T-триггер** (от англ. *toggle* — переключатель) - меняет своё состояние на каждом перепаде синхросигнала. T-триггер часто называют счётным триггером, так как он является простейшим счётчиком по модулю 2

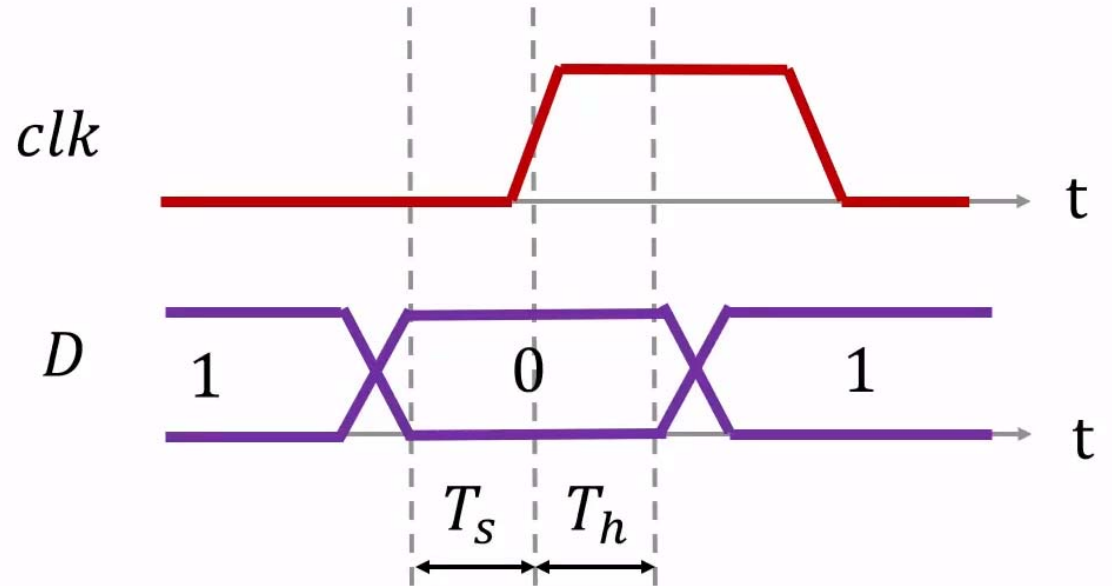
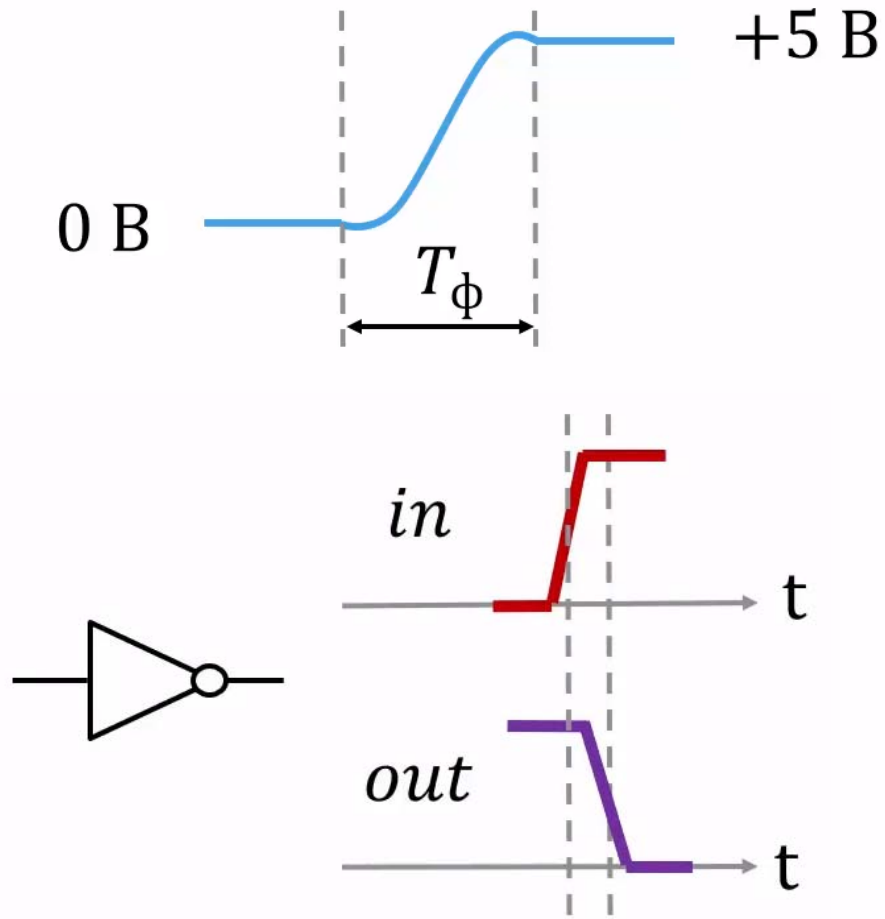


Вход T принимает разрешающий сигнал счёта/переключения (статическое управление)





# Временные ограничения триггеров



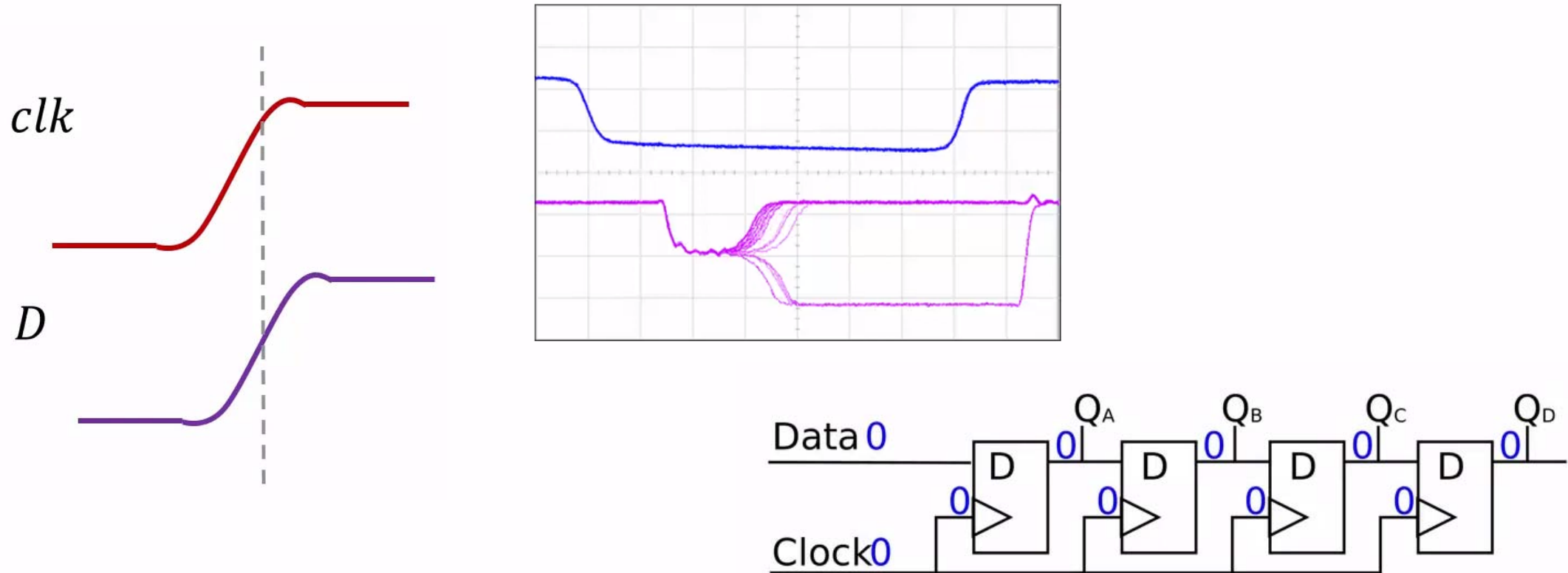
$T_s$  (setup time) — время предустановки

$T_h$  (hold time) — время удержания

$T_s + T_h$  = временное окно «стабильности» данных на входе

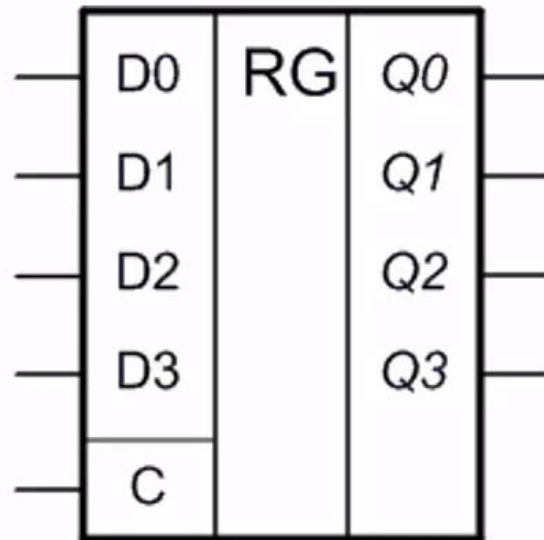
## Метастабильность

**Метастабильность** – неопределённое состояние триггера, не относящееся ни к «нулю», ни к «единице». Из этого состояния триггер может выйти в одно из определённых состояний, но процесс выхода может занять неопределённое время и закончиться неопределённым результатом.



## Регистры

**Регистр** — устройство для записи, хранения и считывания  $n$ -разрядных двоичных данных и выполнения других операций над ними. Регистр представляет собой упорядоченный набор триггеров, обычно **D-триггеров**, число которых соответствует числу разрядов в слове.



### Типичное применение в

#### вычислительной технике:

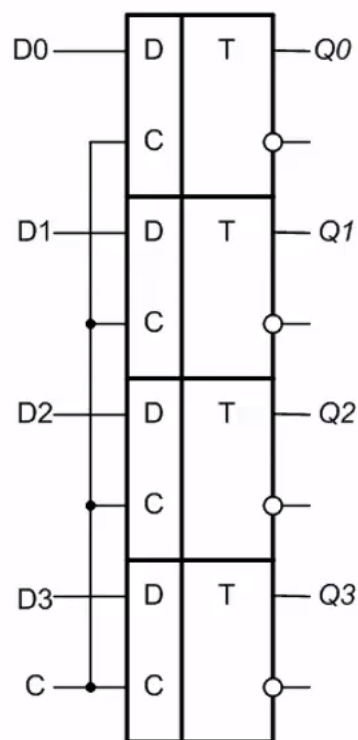
- хранение двоичных данных
- синхронизация
- инициализация
- ввод/вывод
- арифметические операции
- генерация сигналов

### Виды регистров:

- Параллельные
- Последовательные (сдвиговые)
- Комбинации

# Параллельный регистр

**Параллельный регистр**, точнее, регистр с параллельной загрузкой и чтением, строится из триггеров, не обменивающихся данными друг с другом, но имеющих общий сигнал синхронизации и другие сигналы управления \*.



\* фактически выполняет роль  
**D-триггера** для  
многоразрядных слов!

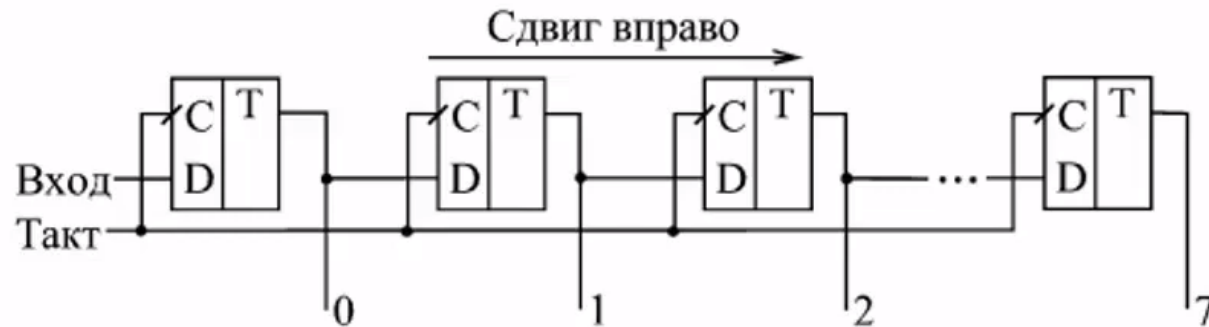
## Применение:

- хранение двоичных данных
- Инициализация
- синхронизация



# Сдвиговой регистр

**Последовательный (сдвиговой) регистр** представляет собой цепочку триггеров, связанных цепями переноса. На каждом такте синхросигнала выход одного триггера будет записываться в последующий триггер, и так далее.



## Применение:

- ввод/вывод
- арифметические операции
- генерация сигналов

## Виды:

- однонаправленные / реверсивные
- кольцевые
- последовательный ввод и вывод
- параллельный ввод и вывод
- параллельный ввод – последовательный вывод
- последовательный ввод – параллельный вывод

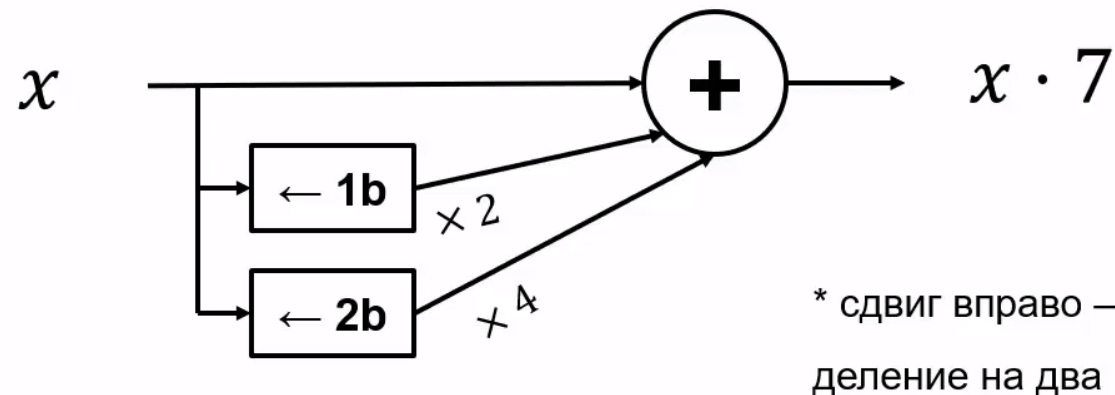
Цифровая схемотехника  
**Арифметические операции**  
**Счетчики**

# Арифметические операции и генерация чисел

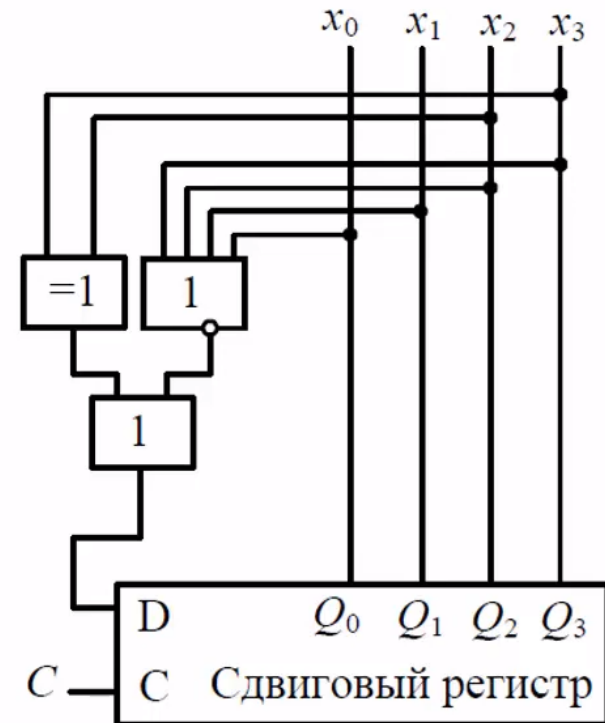
Умножение на константу:

10100 - сдвиг влево, умножение  
на степень двойки \*

Как умножить на произвольное число? Сдвиг и суммирование!



Генератор псевдослучайной  
последовательности:

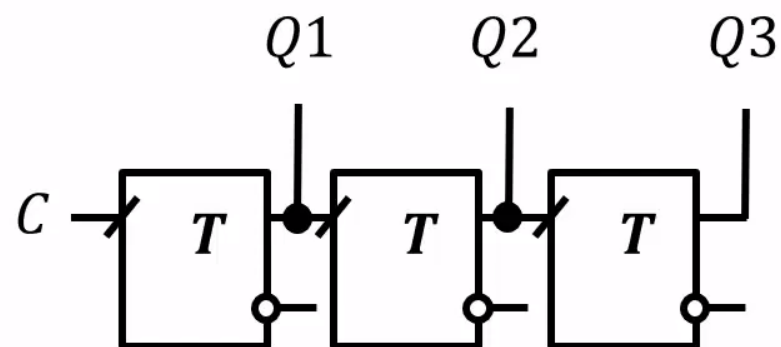
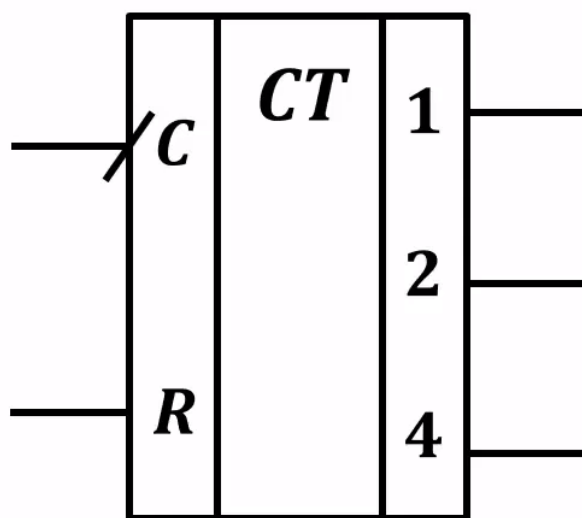




## Двоичный счетчик

**Счётчик** числа импульсов — устройство, на выходах которого получается двоичный код, определяемый числом поступивших импульсов. Счётчики могут строиться на **T-триггерах**, D-триггерах и JK-триггерах.

Основной параметр счётчика — модуль счёта — максимальное число единичных сигналов, которое может быть сосчитано счётчиком. Счётчики обозначают через **СТ** (от англ. **counter**).



**Виды:** суммирующий, вычитающий, реверсивный

**Применение:**

- деление частоты
- подсчёт импульсов



Функция счетчика состоит в подсчете количества импульсных сигналов, поступающих на его вход. Элементом структуры счетчика является  $T$ -триггер. Состояние счетчика характеризуется его содержимым

$$S(t) = s_n s_{n-1} \dots s_0.$$

Здесь  $s_i$  – состояние  $i$ -го триггера счетчика. Код  $S(t)$  так или иначе соответствует количеству счетных импульсов, поступивших на счетчик к данному моменту времени  $t$ . Одно из состояний счетчика принимают за начальное  $s(0)$ . Подсчет количества импульсов производится в счетчике по модулю  $K$ , где  $K$  – общее число различных состояний счетчика. Число  $K$  называют или модулем, или основанием, или коэффициентом пересчета, а также емкостью счетчика.

Триггер типа  $T$  выполняет подсчет по модулю двух единичных сигналов на его информационном входе, т.е. работает как одноразрядный счетчик. Если к выходу такого триггера подключить еще один  $T$ -триггер, последний будет подсчитывать количество единичных состояний первого. Соединив последовательно  $(n + 1)$  триггеров типа  $T$ , получим суммирующий бинарный счетчик с основанием  $K = 2^{n+1}$ .

Классифицируют счетчики по следующим признакам: по коэффициенту пересчета  $K$ , по направлению счета, по способу управления счетом.

**По коэффициенту пересчета**  $K$  различают:

- **бинарные счетчики**, у которых  $K = 2^{n+1}$ ;
- **счетчики с произвольным коэффициентом пересчета** ( $K \neq 2^{n+1}$ ,  $K = \text{const}$ );
- **счетчики с произвольным  $K \neq 2^{n+1}$** ,  $K = \text{var}$ .

**По направлению счета** счетчики разделяют как:

- **суммирующие**, у которых  $S(0) < S(t)$ ,  $t > 0$  (**суммирующий счетчик** – счетчик, который под действием счетного импульса переходит из состояния  $s(t)$  в  $(s(t) + 1) \bmod K$  состояние);
- **вычитающие**, у которых  $S(0) > S(t)$ ,  $t > 0$  (**вычитающий счетчик** – счетчик, который под действием счетного импульса переходит из состояния  $s(t)$  в  $(s(t) - 1) \bmod K$  состояние);
- **реверсивные счетчики** (цифровые устройства, способные работать как в режиме суммирующего, так и в режиме вычитающего счетчика).

**По способу управления счетом** счетчики бывают:

- **асинхронными**, когда смена состояния счетчика происходит только под воздействием счетных импульсов;
- **синхронными**, для работы которых необходимы еще и хронизирующие сигналы.

Признаки классификации независимы и могут встречаться в различных сочетаниях, скажем, асинхронный двоично-десятичный суммирующий счетчик или бинарный синхронный реверсивный счетчик.

## Бинарные суммирующие счетчики

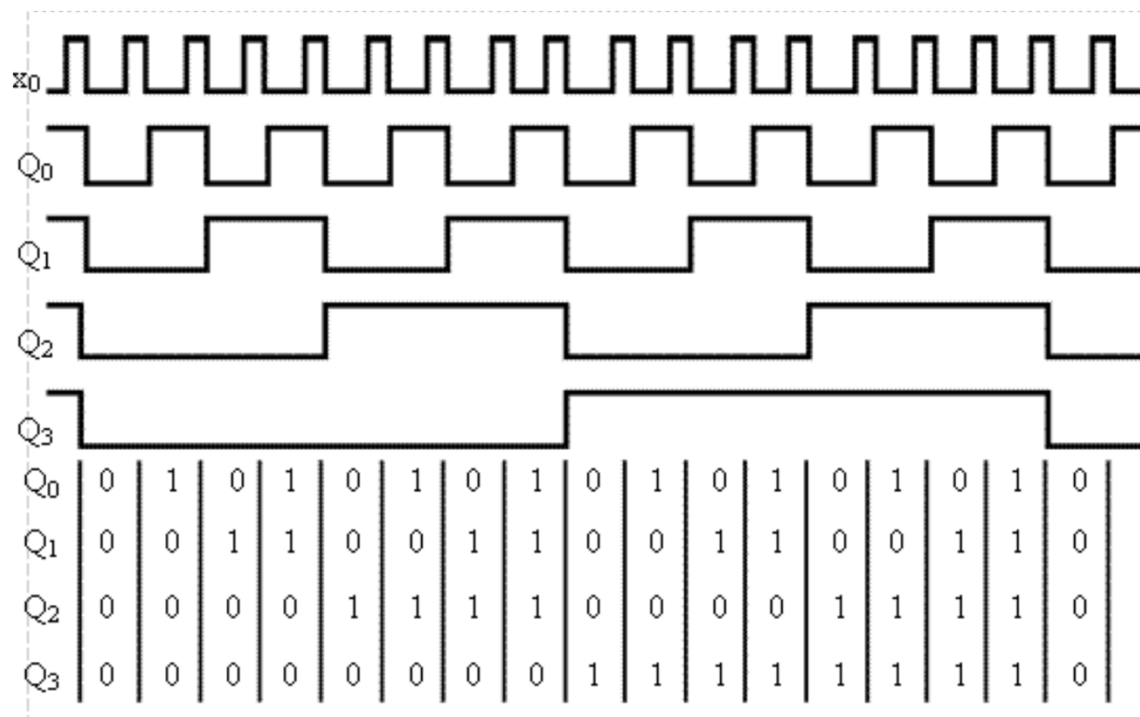
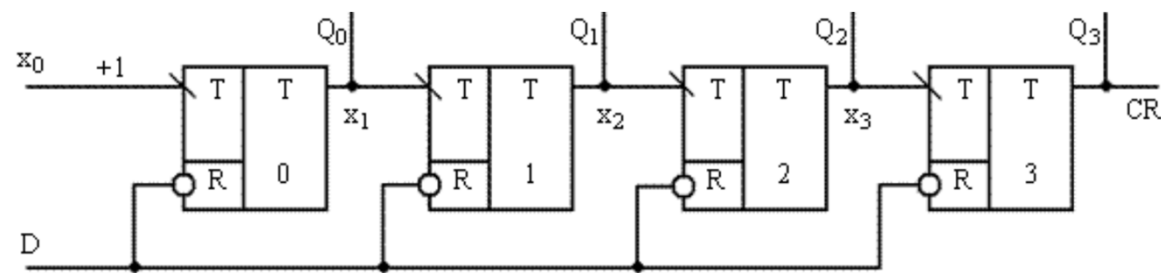
Функцию суммирующего счетчика можно задать такими соотношениями:

$$S(t+1) = \begin{cases} S(t) + 1 & \text{при } x(t) = 1 \\ S(t) & \text{при } x(t) = 0, \end{cases}$$

где  $x(t)$  – элемент потока счетных импульсов;  $S(t), S(t+1) \in [0, 2^n - 1]$  – состояния счетчика.

С каждым очередным счетным импульсом счетчик переходит из состояния  $s(t)$  в состояние  $s(t)+1$ .

Суммирующий бинарный 4-разрядный счетчик с основанием  $K = 2^n + 1$ , временные диаграммы и таблицы состояний 4-разрядного счетчика



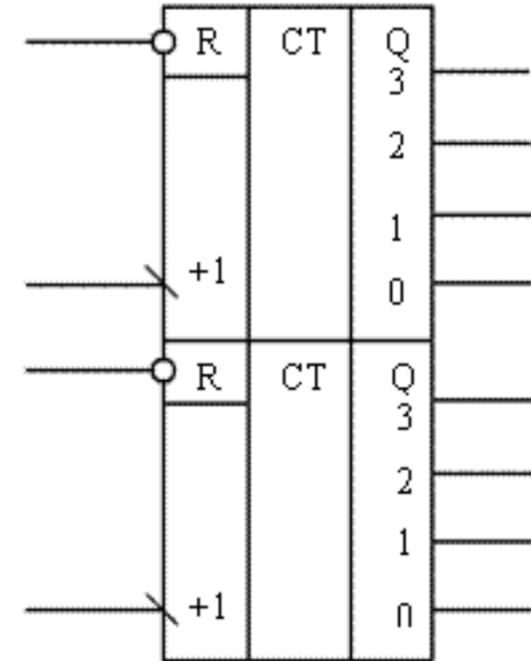
## Бинарные суммирующие счетчики

ИС 1533ИЕ19, которая содержит два автономных бинарных суммирующих счетчика с последовательным переносом.

Достоинством таких счетчиков является предельная простота их структуры и легкость наращивания разрядности.

Быстродействие счетчика оценивают максимальной величиной  $t_{S,S+1}$  времени перехода из состояния  $S$  в состояние  $S + 1$ . Ясно, что период следования счетных импульсов не может быть меньше величины  $t_{S,S+1}$ . У счетчика с последовательным переносом все его триггеры переключаются на переходе от  $S = 2^n - 1$  к  $S + 1 = 0$  один за другим. Поэтому  $t_{S,S+1} = (n + 1)t_T$ , т.е. счетчики с последовательным переносом имеют низкое быстродействие. Скажем, счетчик на рисунке имеет  $t_{S,S+1} = 56$  нс.

Способом повышения быстродействия счетчика является организация одновременного переключения всех тех его триггеров, которые должны изменить свое состояние на переходе из состояния  $S$  в состояние  $S + 1$ .

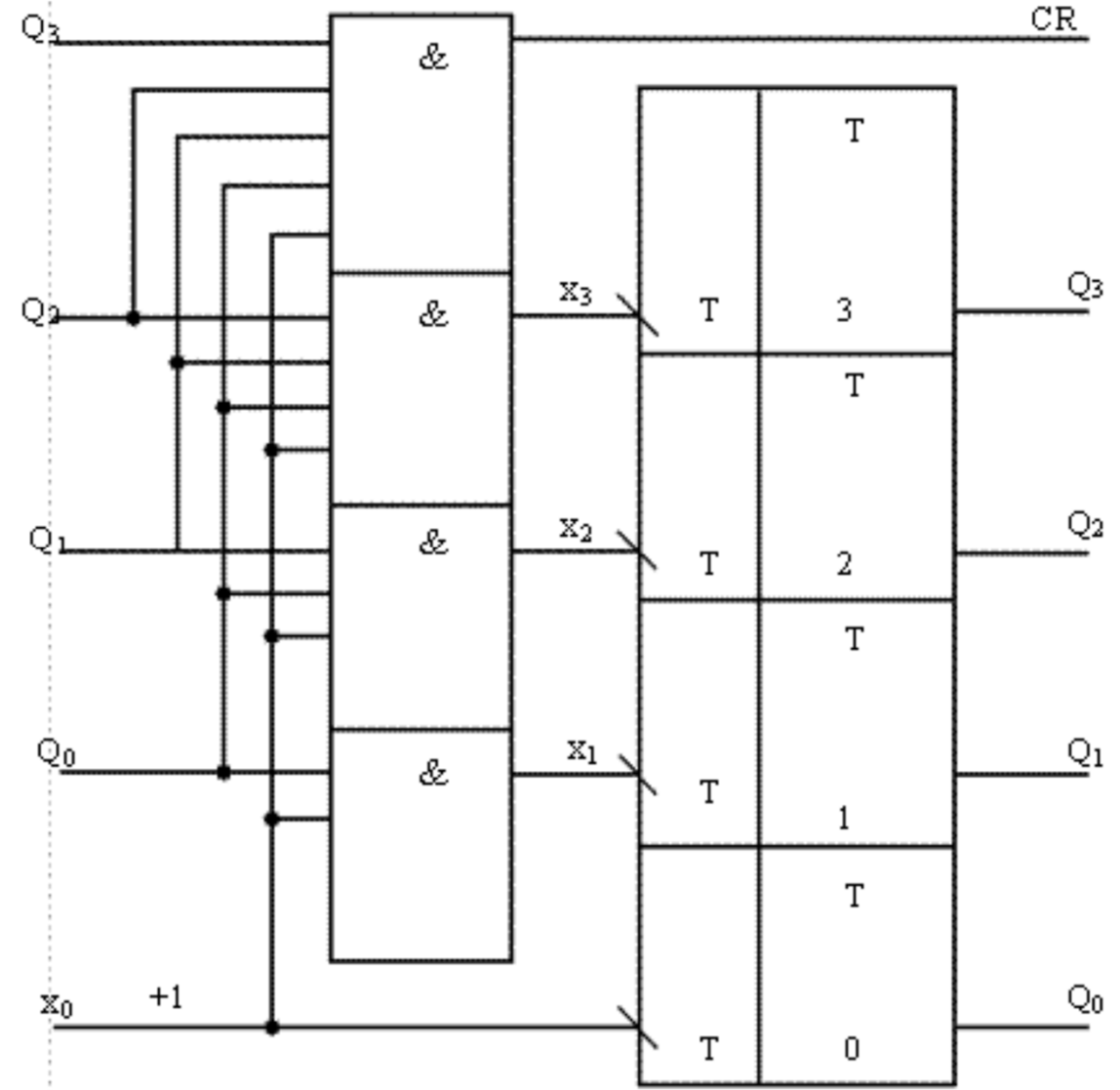


## Бинарные суммирующие счетчики

Если сигнал на  $T$ -входе каждого из триггеров счетчика формировать по формуле:

$X_i(t) = Q_{i-1}(t) \dot{\cup} \overline{Q_{i-1}(t)} \dot{\cup} \overline{Q_0(t)} \dot{\cup} x_0(t)$ ,  $i = \overline{1, n}$ , то все триггеры, которые должны переключиться на переходе от  $S$  к  $S + 1$ , сработают одновременно, в момент поступления  $x_0(t) = 1$ . Такой счетчик называют счетчиком с параллельным переносом. Время его переключения не зависит от разрядности и составляет величину  $t_{s,s+1} = t_T + t_x$ .

Рассмотренный счетчик с параллельным переносом является асинхронным. В нем триггер младшего разряда переключается прямо счетным импульсом, а остальные триггеры – сигналами с выходов конъюнкторов, т.е. с некоторой задержкой по отношению к срезу счетного импульса.

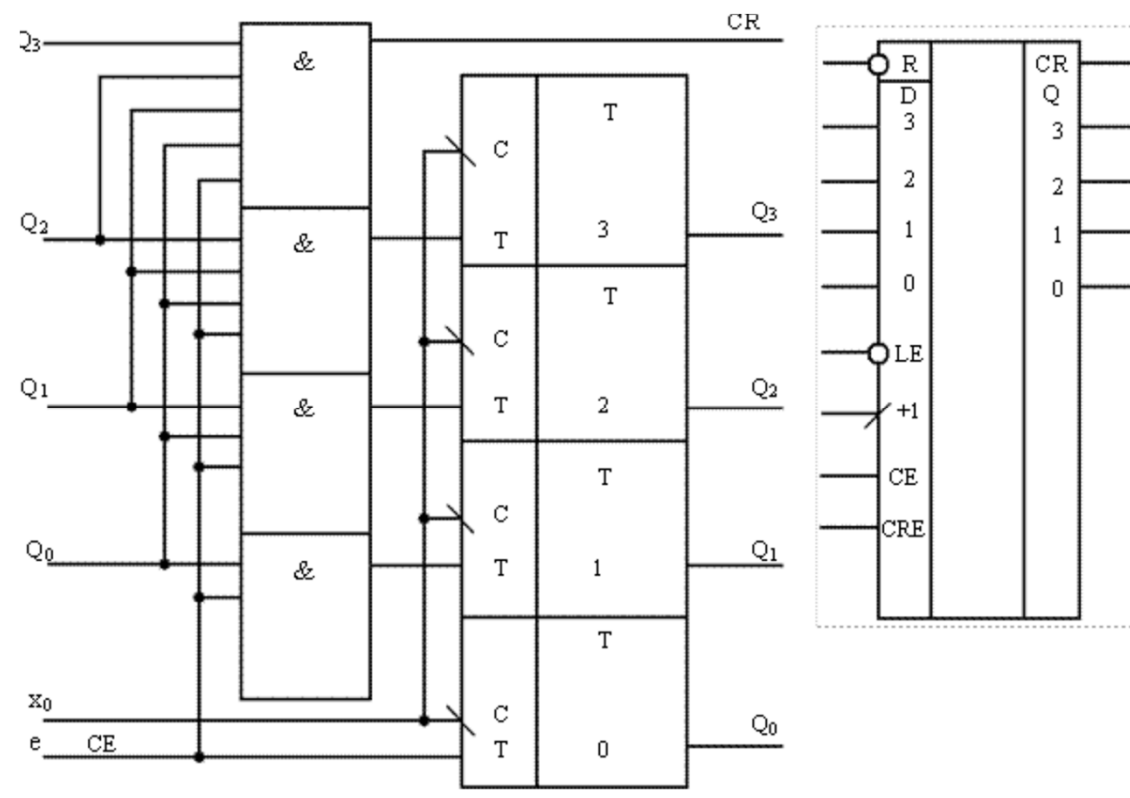


## Бинарные суммирующие счетчики

Свободен от такого недостатка синхронный счетчик с параллельным переносом, в котором применены хронизируемые  $T$ -триггеры. Здесь сигнал  $e$  на потенциальном входе  $CE$  (разрешение счета) не переключает триггеры, а обеспечивает получение единичных значений на информационных входах тех триггеров, которые должны переключиться на переходе от  $S$  к  $S + 1$ . А само переключение триггеров происходит одновременно по срезу счетного импульса  $x_0$ , который подается на вход  $+1$  счетчика, т.е. на синхро-входы всех триггеров.

Обычно счетчики допускают не только асинхронный сброс в состояние  $S(0) = 0$ , но и установку регистра в любое исходное состояние  $0 \leq S(0) \leq 2^n - 1$ . Для этого счетчики снабжаются входами для параллельной записи произвольного числа перед началом счета.

Сигналом  $R = 0$  счетчик асинхронно сбрасывается в нулевое состояние. При  $LE = 0$  фронтом импульса на входе  $+1$  код с входов  $D$  параллельно записывается в триггеры счетчика. Режим счета импульсов по входу  $+1$  задается комбинацией разрешающих сигналов  $CE \times CRE = 1$ , а при  $CE \times CRE = 0$  счетчик находится в режиме хранения.



1554IE10

## Реверсивные бинарные счетчики

Принцип работы вычитающего счетчика. Для этого сравним коды  $S$  и  $S - 1$ , например такие:

$$S = 1^3 1^2 0^1 0^0;$$

$$S - 1 = 1^3 0^2 1^1 1^0.$$

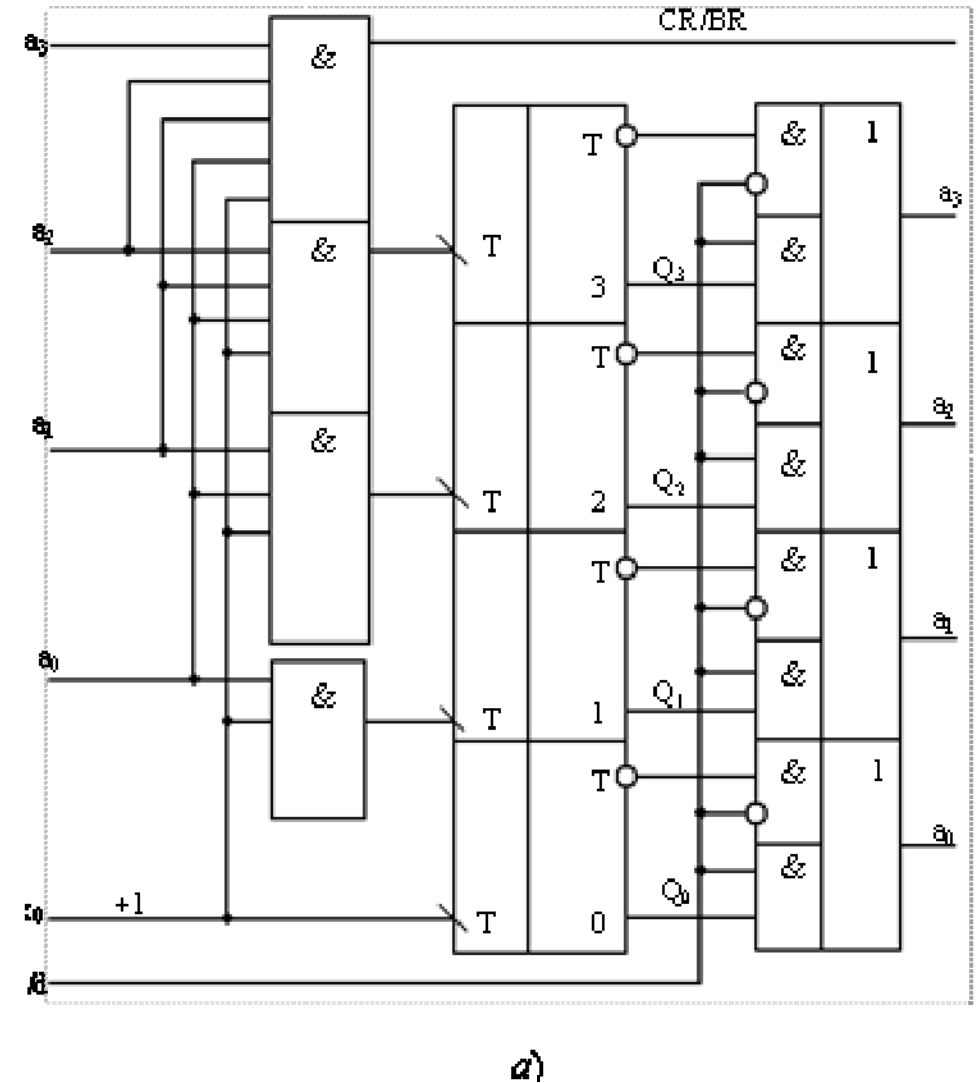
Как видим, код  $S - 1$  получают из кода  $S$  путем инверсии в записи  $S$  младшей из единиц и стоящих правее нее нулей. Следовательно, если в суммирующем счетчике прозрачной для  $x_0(t) = 1$  была цепочка из  $i$  младших триггеров, в которых записаны единицы, то в вычитающем счетчике прозрачной для  $x_0(t) = 1$  будет группа из  $i$  младших триггеров, в которых записаны нули. Поэтому для вычитающего счетчика сигнал на информационном входе триггера  $i$ -го разряда должен формироваться так:

$$x_i(t) = \overline{Q_{i-1}}(t) \wedge \overline{Q_{i-2}}(t) \wedge \dots \wedge \overline{Q_0}(t) \wedge x_0(t), \quad i = \overline{1, n}$$

А именно, сигналы на  $T$ -входах триггеров здесь формируются из сигналов не с прямых, как у суммирующего счетчика, а с инверсных выходов триггеров.

## Реверсивные бинарные счетчики

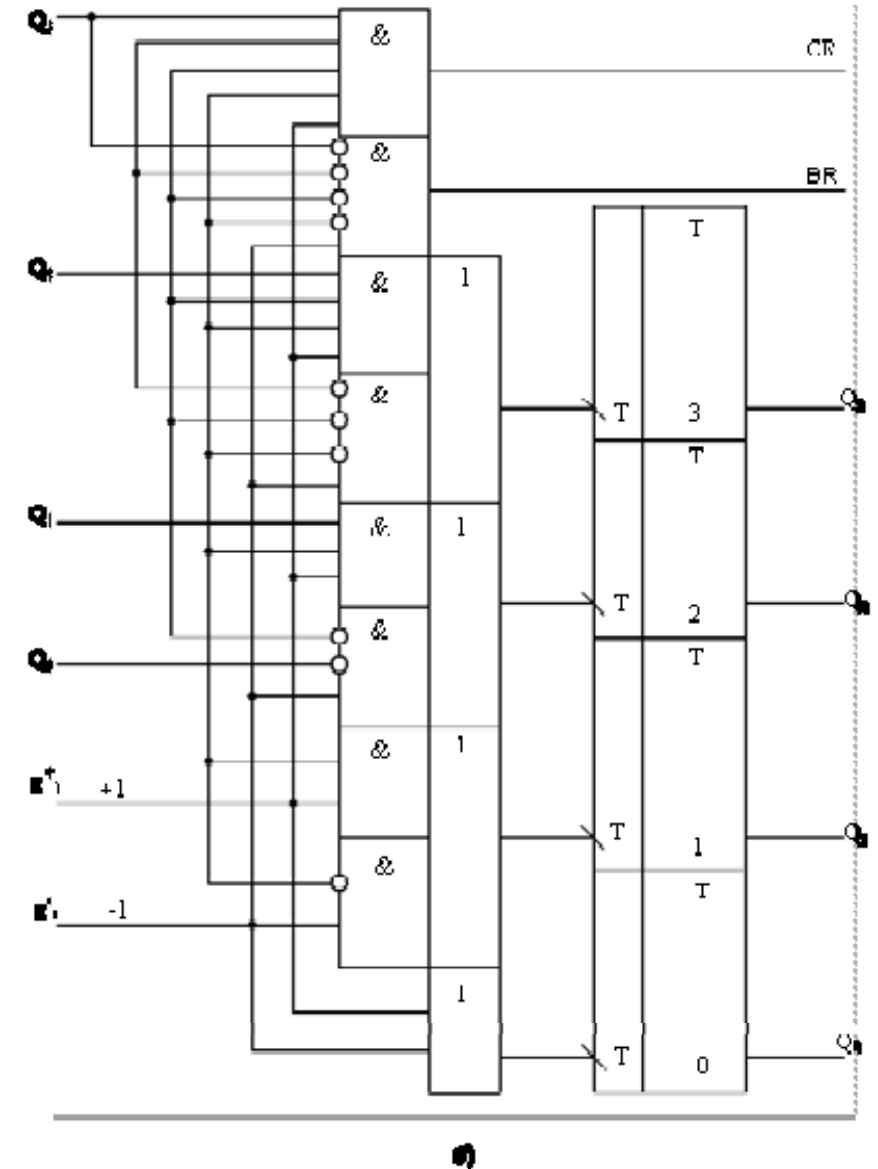
Функциональная схема 4-разрядного реверсивного счетчика с параллельным переносом. Выходы каждого триггера подключены к мультиплексору, который управляется сигналом  $u/d$  (от *up/down*). При  $u/d = 1$  счетчик работает в режиме суммирования, а при  $u/d = 0$  – в режиме вычитания счетных импульсов. Этот счетчик имеет один вход для счетных импульсов и при их суммировании, и при их вычитании, а для наращивания разрядности у него имеется один выход  $CR/BR$ . При  $S = 15$  в режиме суммирования и при  $S = 0$  в режиме вычитания  $CR/BR = 1$ .





## Реверсивные бинарные счетчики

В режиме суммирования счетные импульсы подаются на вход +1, а в режиме вычитания – на вход -1. Подавать счетные импульсы одновременно на оба этих входа нельзя. Счетчик имеет и отдельные выходы *CR* и *BR* для наращивания разрядности. Выход *CR* данной группы подключается ко входу +1, а выход *BR* – ко входу -1 следующей группы. На рис. 1.34, в показано УГО ИС 1554ИЕ7 – 4-разрядного реверсивного счетчика с отдельными счетными входами. Сигналом  $R = 1$  счетчик асинхронно сбрасывается в состояние 0. Во всех остальных режимах  $R = 0$ . Сигналом  $L = 0$  производится запись параллельного кода в счетчик по входам *D*. При суммировании счетные импульсы подаются на вход +1, а на входе -1 должна быть единица. При вычитании счетные импульсы поступают на вход -1, а единица устанавливается на входе +1. Когда единицы установлены на обоих счетных входах и  $R = 0$ , а  $L = 1$ , счетчик хранит информацию. Быстродействие счетчика составляет  $t_{s...s+1} = 17$  нс.

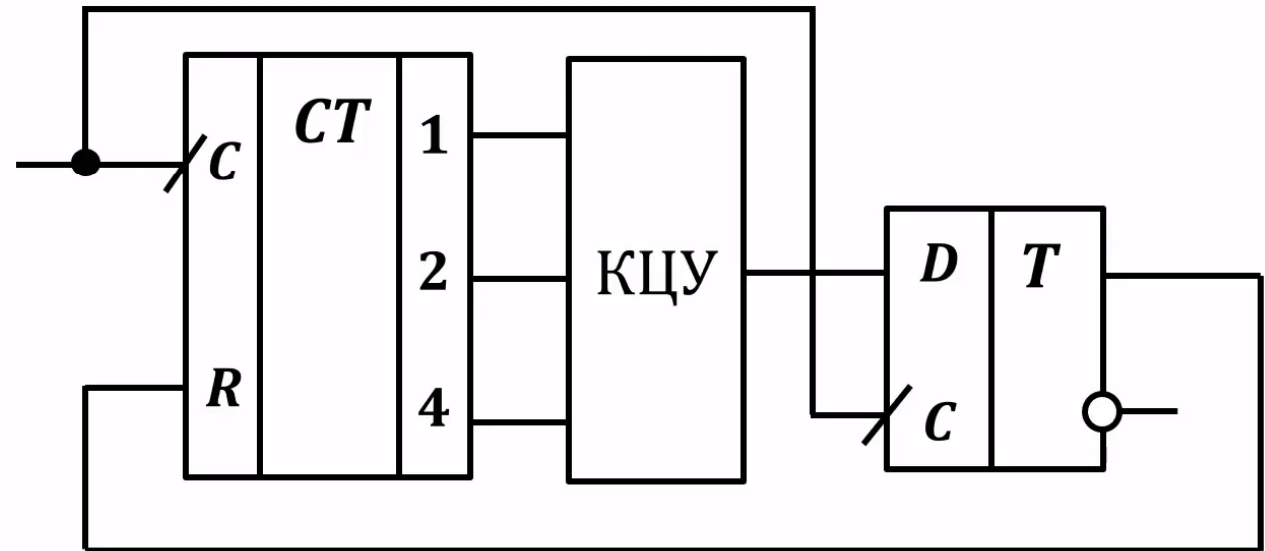


## Счетчики с произвольным коэффициентом пересчета

Во многих случаях необходимы счетчики с  $K \neq 2^n + 1$ . Скажем, в различных устройствах цифровой обработки используют двоично-десятичные счетчики с  $K = 10$ , в электронных часах – делители с  $K = (31, 30, 7, 24, 600)$  и др.

На базе бинарных счетчиков можно создавать счетчики с произвольным коэффициентом  $K$ , отвечающим условиям  $2^n < K \leq 2^{n+1}$ . Рассмотрим два способа построения счетчиков с произвольным коэффициентом пересчета.

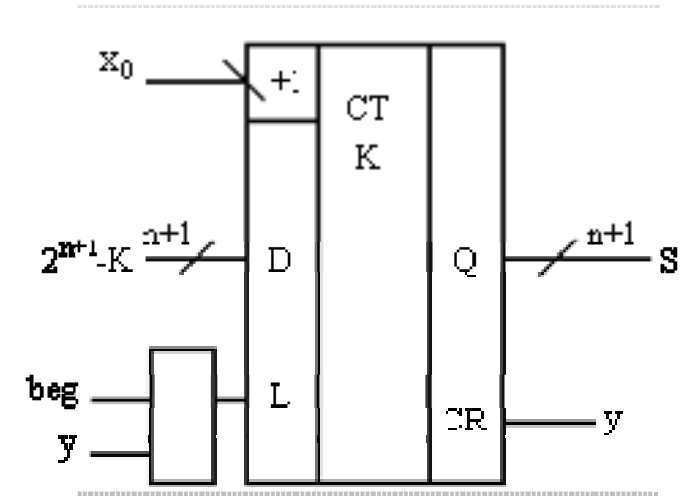
В подобных счётчиках есть обратная связь, и схема, реагирующая на определённую комбинацию выходов. В момент наступления комбинации счётчик сбрасывается на нулевое состояние.



# Счетчики с произвольным коэффициентом пересчета

## Счетчики с исключением младших состояний

Счетчики с исключением младших состояний строят на основе счетчиков с параллельной записью. На рисунке показано УГО такого счетчика. Перед началом счета сигналом  $beg = 1$  в счетчик по входам  $D$  записывается число  $S(0) = 2^{n+1} - K$ . После этого счетчик переключается в режим суммирования счетных импульсов. Таким образом, счет начинается не от нуля, а от  $S$ , чем и объясняется название метода.  $(K-1)$ -м счетным импульсом счетчик устанавливается в состояние  $2^{n+1} - 1$ , которое фиксируется значением  $CR = y = 1$ . Этим сигналом в счетчик снова заносится число  $S = 2^{n+1} - K$ , и начинается новый цикл счета. Достоинством такого способа получения  $K'2^{n+1}$  является использование штатного тракта формирования сигнала  $CR$  и системы параллельной загрузки счетчика.



# Счетчики с произвольным коэффициентом пересчета

## Счетчики с исключением старших состояний

**Счетчики с исключением старших состояний** – счетчики с произвольным коэффициентом пересчета, начальное состояние которых равно 0.

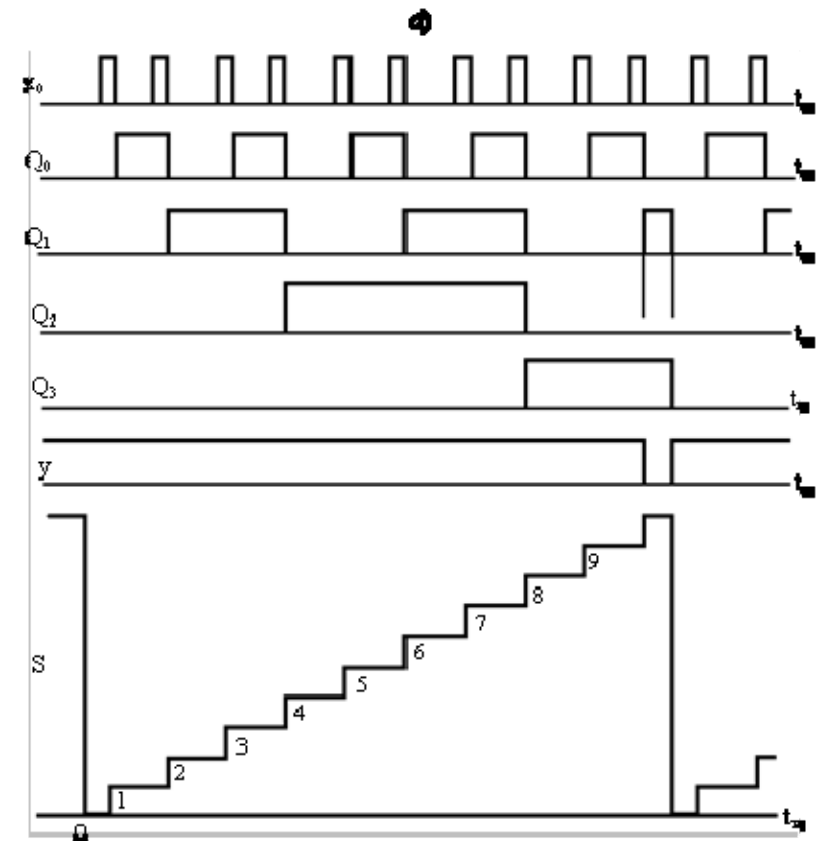
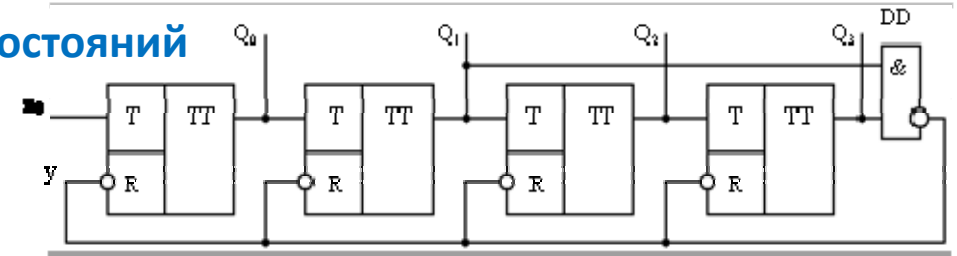
Такие счетчики имеют естественную последовательность состояний. На рисунке показана функциональная схема двоично-десятичного счетчика (т.е. у него  $K = 10$ ). И временные диаграммы, поясняющие его работу. В каждом из состояний  $S = (0, 1, \dots, 9)$  счетчик находится в течение периода следования счетных импульсов. При  $S = 10$  на выходе схемы  $DD$  формируется значение  $y = 0$ , которым счетчик сбрасывается в нулевое состояние.

Длительность пребывания счетчика в состоянии  $S = 10$  составляет

$$t_{сбр} = t_{DD} + t_R,$$

где  $t_{DD}$  – задержка в схеме  $DD$ ;  $t_R$  – время срабатывания счетчика по сигналу на асинхронном установочном входе.

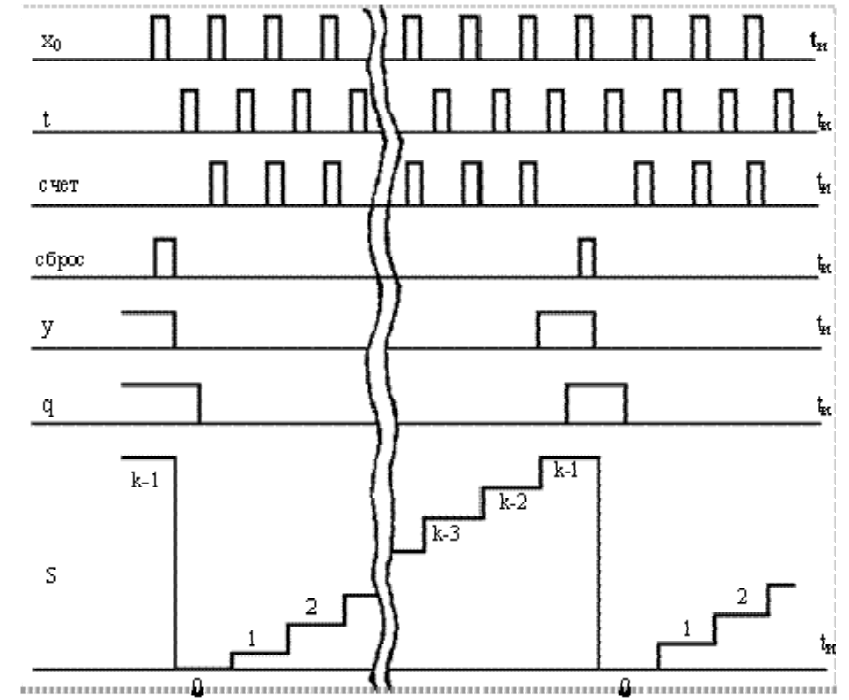
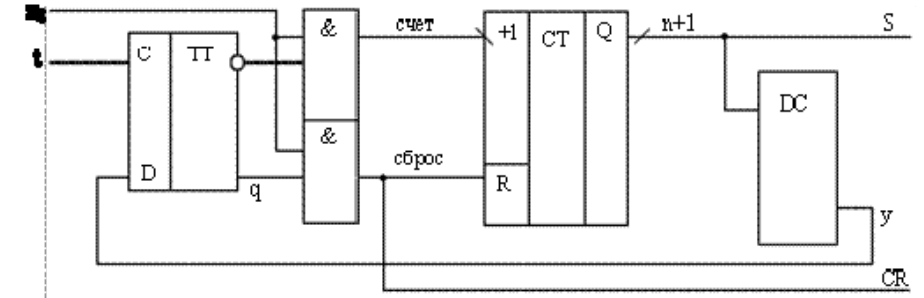
Считается, что  $t_{сбр}$  много меньше периода следования счетных импульсов, и потребитель результатов счета не успевает среагировать на показание  $S = 10$ .



## Счетчики с произвольным коэффициентом пересчета

### Счетчики с исключением всех старших состояний $S > K - 1$

Особенностью этого счетчика является использование наряду с последовательностью счетных импульсов  $x_0$  еще и последовательности  $t$ . Отсчеты  $t$  имеют тот же период следования, что и  $x_0$ , но сдвинуты относительно импульсов  $x_0$  на время, меньшее периода их следования. Последовательностью  $t$  в  $D$ -триггер записывается значение сигнала  $y$  с выхода дешифратора  $DC$ , который фиксирует последнее состояние счетчика  $S = K - 1$ . До тех пор, пока это состояние не достигнуто, сигнал с выхода  $DC$   $y = 0$  удерживает триггер в нулевом состоянии. Тем самым счетные импульсы  $x_0$  коммутируются на вход  $+1$  счетчика. Достигнув состояния  $S = K - 1$ , счетчик пребывает в нем в течение периода следования счетных импульсов. На этом интервале  $DC$  формирует значение  $y = 1$ , и оно фиксируется в  $D$ -триггере отсчетом  $t$ , который следует сразу за  $(K-1)$ -м счетным импульсом. В результате очередной  $K$ -й счетный импульс подается на  $R$ -вход счетчика и сбрасывает его в состояние  $S = 0$ . Одновременно на выходе  $DC$  получаем значение  $y = 0$ . Импульсом  $t$  это значение фиксируется в  $D$ -триггере, и счетные импульсы снова поступают на вход  $+1$  счетчика.



# Синхронный счетчик

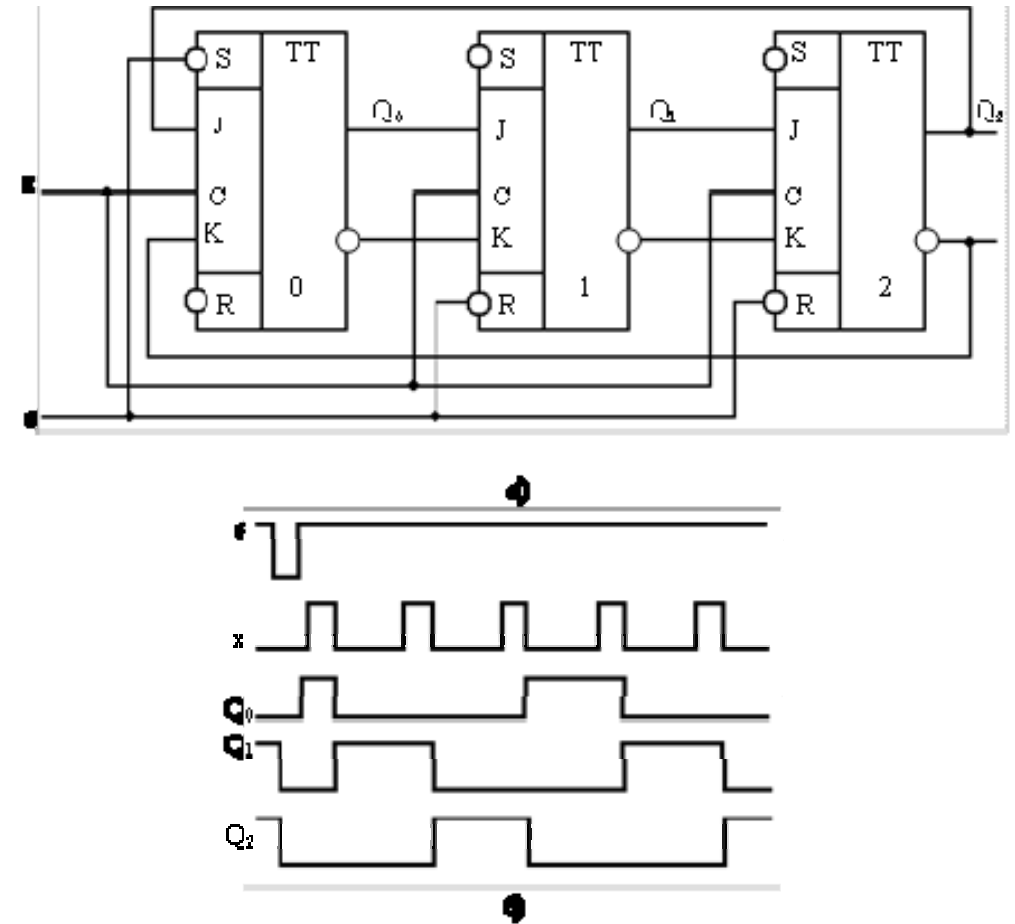
Перенос осуществляется последовательно, но на каждый триггер одновременно приходит тактовый сигнал синхронизации!



## Кольцевые распределители

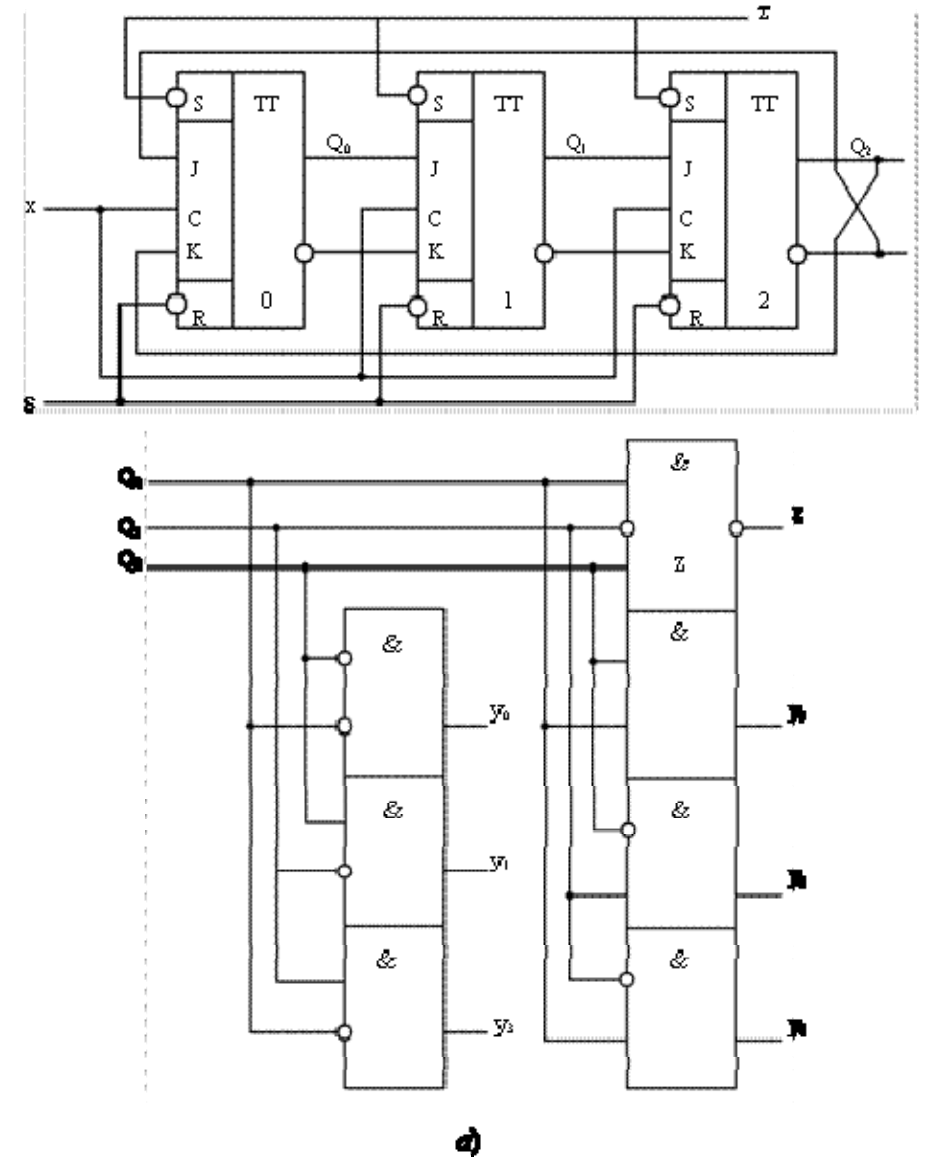
**Распределитель** – цифровой узел, который преобразует временное распределение импульсов в пространственное, когда каждый следующий импульс из входного потока отправляется на свой выход. Распределители строят на базе регистров сдвига, но называют их, тем не менее, счетчиками.

**Кольцевой счетчик** – сдвиговый регистр, у которого выход триггера самого старшего разряда соединен со входом триггера самого младшего разряда, и в любой момент времени только в одном триггере регистра записана единица. Эта единица при каждом сдвиге перемещается из одного триггера в другой. Так поток сдвиговых импульсов  $x$  распределяется по триггерам (в пространстве). Из всех  $2^n + 1$  возможных состояний разрешенными для регистра в кольцевом счетчике являются только  $(n + 1)$  состояний с единицей лишь в одном из  $(n + 1)$  разрядов.



## Кольцевые распределители

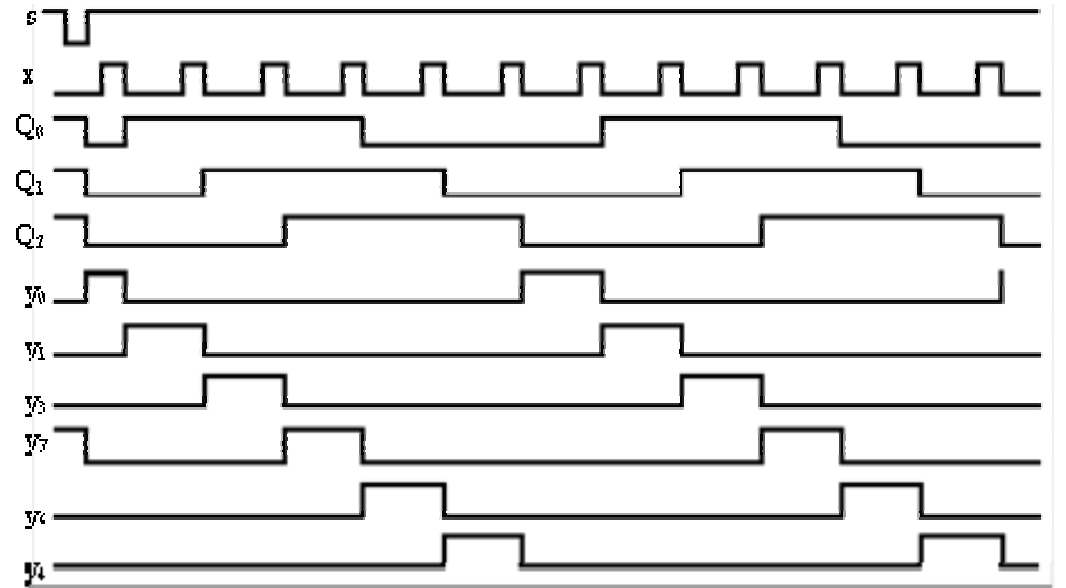
Функциональная схема 3-разрядного счетчика Джонсона. Выходы старшего разряда сдвигающего регистра соединены со входами его младшего разряда перекрестно. Поэтому такой распределитель называют еще и счетчиком Мебиуса. Сигналом  $S = 0$  все триггеры регистра сбрасываются в состояние 0. При этом на информационных входах триггера младшего разряда будут иметь место значения  $J = 1, K = 0$ . Значением  $S = 1$  задается рабочий режим распределителя. Первым же импульсом сдвига  $x$  в триггер младшего разряда регистра записывается единица. Затем по мере поступления сдвигавших импульсов  $x$  триггеры заполняются сначала единицами, а затем – после установки в состояние 1 триггера старшего разряда – нулями. В общем случае для счетчика Джонсона разрешено  $2(n + 1)$  состояний, т.е. вдвое больше, чем для кольцевого. Таким образом, из восьми возможных состояний 3-разрядного счетчика Джонсона на рис. 1.39, а разрешены шесть, а именно  $S = \{0, 1, 3, 7, 6, 4\}$ . Состояния 2 и 5 в нем запрещены. Используя этот факт, разрабатывают дешифратор состояний счетчика Джонсона, по выходам которого  $y_0, y_1, y_3, y_7, y_6, y_4$  и распределяется поток импульсов  $x$ .





## Кольцевые распределители

Общий недостаток изученных распределителей состоит в следующем. Из-за сбоя распределитель может попасть в запрещенное состояние. Ясно, что картина распределения импульсов при этом будет искажена. Так, при случайном обнулении кольцевого счетчика распределение импульсов вообще прекращается, а сам счетчик не выйдет из состояния  $S = 0$ . Если счетчик Джонсона случайно окажется в состоянии  $S = 5 = 101$ , то оно импульсом сдвига сменится на  $S = 011 = 3$ , а из него счетчик снова перейдет в  $S = 101$ . Для устранения такого недостатка распределитель дополняют схемой коррекции, с помощью которой запрещенное состояние обнаруживается, и счетчик переводится из этого запрещенного состояния в то или иное разрешенное. К примеру, ЛЭ  $z$  на рис. 1.39, а при  $S = 5$  формирует сигнал  $z = 0$ , который подается на асинхронные  $S$ -входы всех триггеров и переводит регистр в разрешенное состояние  $S = 7$ , и функция распределителя восстанавливается.



**Спасибо за внимание**

ЧУ ПО «Социально-технологический колледж»

Преподаватель: Борисов Алексей Альбертович