

# **ПРИНЦИПЫ ПОСТРОЕНИЯ ФУНКЦИОНАЛЬНЫ Х УЗЛОВ**

# 1. Логические элементы и функциональные узлы.

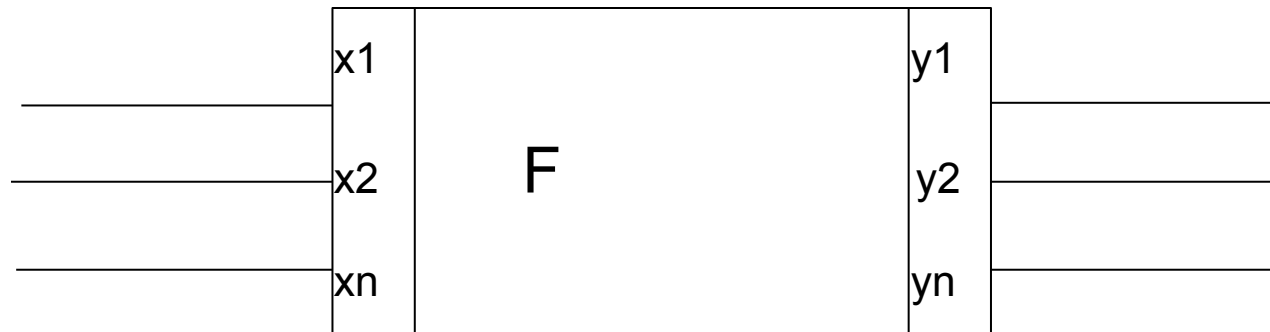
Логические элементы обеспечивают работу с одним битом информации (ноль — единица).

Числа, адреса, команды представляются в виде машинного слова, т.е. Совокупности нулей и единиц. Интерес представляет обработка и передача не одного бита, а их совокупности, т.е. машинного слова, включающей 16, 32, 64 бита.

Совокупность логических (функциональных) элементов, часть логической (функциональной) схемы, описывающая одну или несколько логических функций, объединенных в одно целое, называется функциональным узлом.

Т.е. функциональный узел — можно изобразить с помощью совокупности элементов или в виде одного графического символа.

Еще одно определение: функциональный узел — это совокупность логических элементов, обеспечивающих выполнение определенной операции.



Обозначение функциональных узлов:  $x_i$  — входные параметры,  $y_j$  - выходные параметры, F — название или условное обозначение узла.

## 2. Разновидность функциональных узлов.

Функциональные узлы могут быть **одноразрядные**, т.е. обрабатывающие один разряд слова (например, сумматоры, компараторы), и **многоразрядными**, обрабатывающие слово целиком или 2 машинных слова или слога – половину машинного слова (все остальные узлы).

Функциональные многоразрядные узлы делятся на узлы параллельного и последовательного типа, т.е. работающие в параллельном и последовательном коде передачи информации.

**Параллельный код** — каждый временной такт используется для отображения одного разряда. Двоичный код слова — в виде временной последовательности. Причем слово передается по одной шине.

**Параллельный код** — все разделы слова передаются за один такт по отдельным шинам к отдельным элементам. При этом количество шин должно равняться количеству разрядов. Двоичный код слова — в виде пространственно разнесенной последовательности.

**Последовательно-параллельный код** - слово разбивается на слоги, каждый слог представляется в параллельном виде, а сами слоги передаются последовательно.

В последовательном коде могут работать сумматоры, регистры сдвига, счетчики, большинство же узлов работает в параллельном коде.

Кроме того? передача информации может быть **однофазной** (по одной шине) и **парафазной** (по двум шинам), по одной передается сигнал, а по другой инвертированный сигнал.

Передача информации может быть асинхронной и синхронной (синхронизируемой).

**Асинхронная** (не синхронизированная) — сигналы передаются с небольшой задержкой от элемента к элементу. Элемент или узел срабатывает (меняет свое состояние) после прихода сигнала.

**Синхронизируемая** передача информации — элемент или узел срабатывает только после прихода синхроимпульса (т.е. в строго определенные моменты). Если на вход поступил сигнал, а синхроимпульс не подан, устройство не срабатывает (т.е. должен поступить сигнал плюс синхроимпульс).

**Комбинационные узлы** (узлы комбинационного типа) — функциональные узлы, логическое состояние выходов которых зависит только от комбинации логических сигналов на входе в данный момент времени, т.е. логическое состояние однозначно определяется комбинацией входных переменных в данный момент времени. Эти узлы «не помнят», не сохраняют информации от ранее пришедших сигналов, например, сумматоры, компараторы, преобразователи кодов, (де-)шифраторы, (де-)мультиплексоры, программирующие логические матрицы.

**Последовательностные узлы** (узлы последовательного типа) — функциональные узлы, логическое состояние которых определяется последовательностью поступающих входных сигналов, т.е. логическое состояние определяется комбинацией входных сигналов не только в настоящий момент, но и в предыдущие моменты времени. Говорят, что такие узлы обладают памятью, например регистры, счетчики, генераторы кодов (распределители кодов). Узлы содержат элементы памяти — триггеры.

Другое название таких узлов — цифровые автоматы  $Q(t+1)=f(Q(t),x(t))$ . Различают автомат Мили  $y(t)=\varphi(Q(t),x(t))$  и автомат Мура  $y(t)=\varphi(Q(t))$ , где  $Q(t)$  — состояние узла,  $x(t)$ ,  $y(t)$  — входные и выходные сигналы.

### 3. Функциональные узлы комбинационного типа

Комбинационные узлы (узлы комбинационного типа) — функциональные узлы, логическое состояние выходов которых зависит только от комбинации логических сигналов на входе в данный момент времени, т.е. логическое состояние однозначно определяется комбинацией входных переменных в данный момент времени.

#### 3.1. Сумматоры (цифровые или двоичные сумматоры)

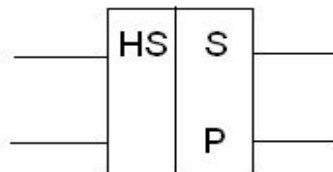
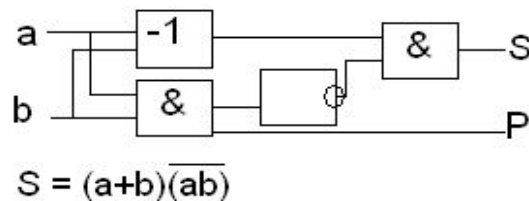
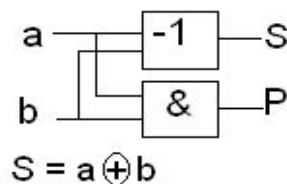
Двоичный сумматор — логическая схема, выполняющая арифметическое сложение чисел в двоичном коде, т.е. арифметическое сложение с помощью булевых операций.

Логическое сложение — дизъюнкцию будем обозначать символом «V», а арифметическое сложение «+»,  $\oplus$  сумма по модулю 2.

**Полусумматор** – схема, выполняющая сложение двух одноразрядных чисел.

| a | b | P | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

S – сумма в том же разряде,  
P – перенос в следующий разряд,  
 $S = a \vee b$ ,  
 $P = a * b$



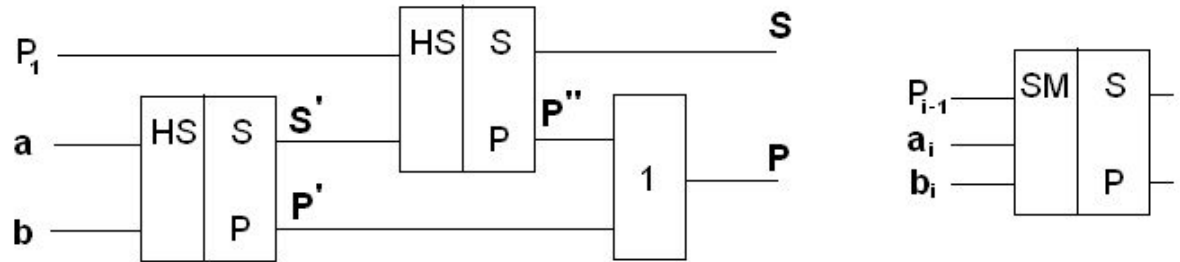
**Полный одноразрядный сумматор** – схема выполняющая сложение двух одноразрядных чисел с учетом переноса из младшего разряда.

$$P' = ab, \quad S' = a \oplus b,$$

$$S = P_{-1} \oplus a \oplus b =$$

$$= abp + ab\bar{p} + a\bar{b}p + a\bar{b}\bar{p} +$$

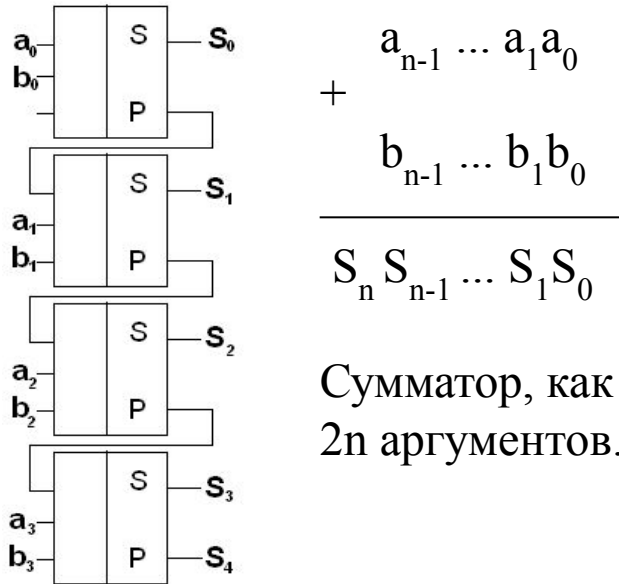
$$+ a\bar{b}p + a\bar{b}\bar{p},$$



$$P'' = (a \oplus b)P_{-1}, \quad P = ab \vee (a \oplus b)P_{-1} = ab \vee (a\bar{b} \vee a\bar{b})P_{-1} = ab \vee (a \vee b)P_{-1} = ab \vee aP_{-1} \vee bP_{-1}$$

| $P_{-1}$ | a | b | $P'$ | $S'$  | $P''$      | P       | S             |
|----------|---|---|------|-------|------------|---------|---------------|
| 0        | 0 | 0 | 0    | 0     | 0          | 0       | 0             |
| 0        | 0 | 1 | 0    | 1     | 0          | 0       | 1             |
| 0        | 1 | 0 | 0    | 1     | 0          | 0       | 1             |
| 0        | 1 | 1 | 1    | 0     | 0          | 1       | 0             |
| 1        | 0 | 0 | 0    | 0     | 0          | 0       | 1             |
| 1        | 0 | 1 | 0    | 1     | 1          | 1       | 0             |
| 1        | 1 | 0 | 0    | 1     | 1          | 1       | 0             |
| 1        | 1 | 1 | 1    | 0     | 0          | 1       | 1             |
|          |   |   | ab   | a + b | $P_{-1}S'$ | $P'P''$ | $P_{-1} + S'$ |

## Полный многоразрядный сумматор в параллельном коде (параллельный)

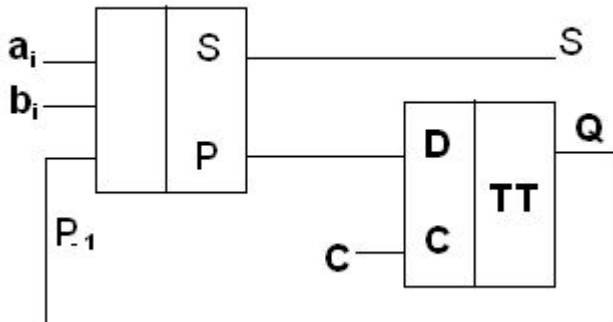


$$\begin{array}{r}
 a_{n-1} \dots a_1 a_0 \\
 + \\
 b_{n-1} \dots b_1 b_0 \\
 \hline
 S_n S_{n-1} \dots S_1 S_0
 \end{array}$$

Сумматор, как  $(n+1)$  мерная функция  
 $2n$  аргументов.

## В последовательном коде (последовательный)

На вход подается  $a_i, b_i$  – т.е. Функция 2-х аргументов, но аргументы меняются со временем.  $(a_0, b_0)(a_1, b_1) \dots$

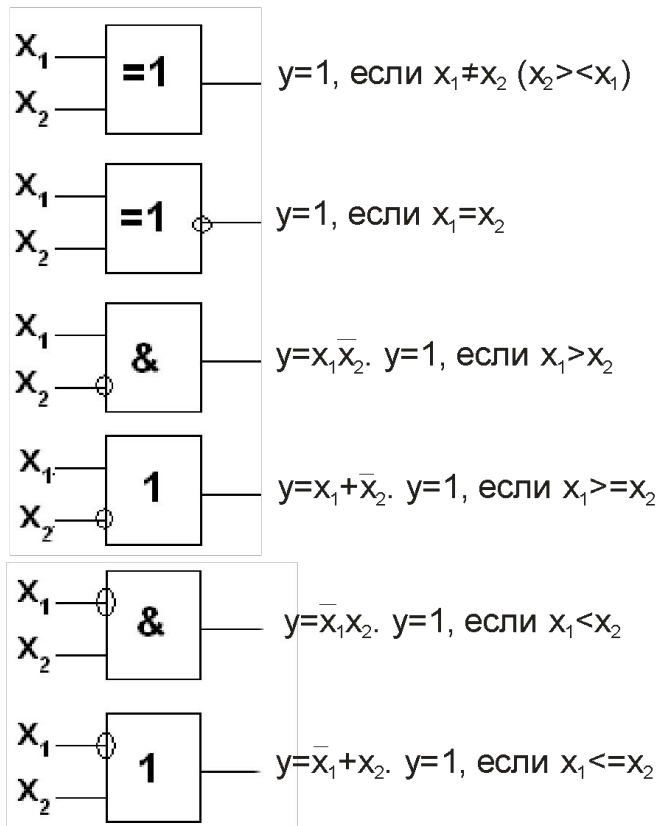


В схеме используется D – триггер, задерживающий на один такт передающуюся через него информацию.

## 3.2 Компараторы (цифровые, или двоичные)

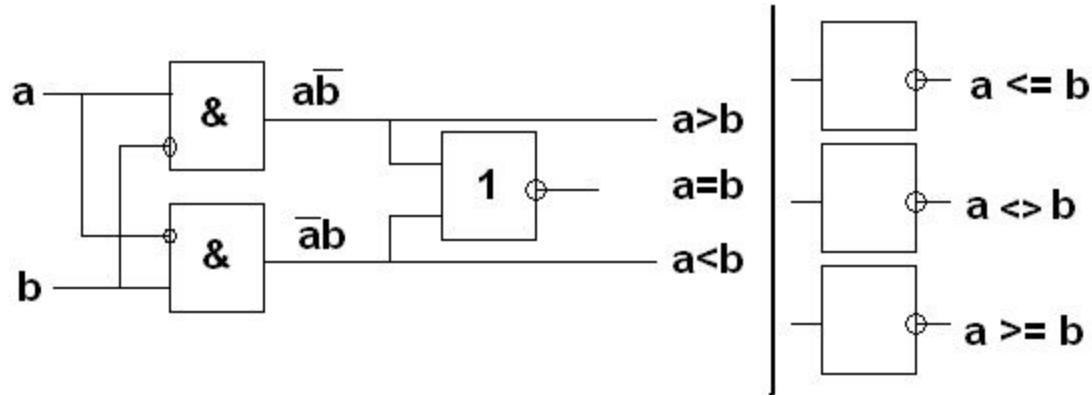
Схемы сравнения двух кодов ( $a_0a_1\dots a_m$ ) и ( $b_0b_1\dots b_m$ ) и формирования признака результата сравнения в виде логической 1 на одном из выходов, если  $A = B$ ,  $A > B$ ,  $A < B$  и т.д.

**Одноразрядные компараторы** – схемы сравнения одноразрядных чисел.



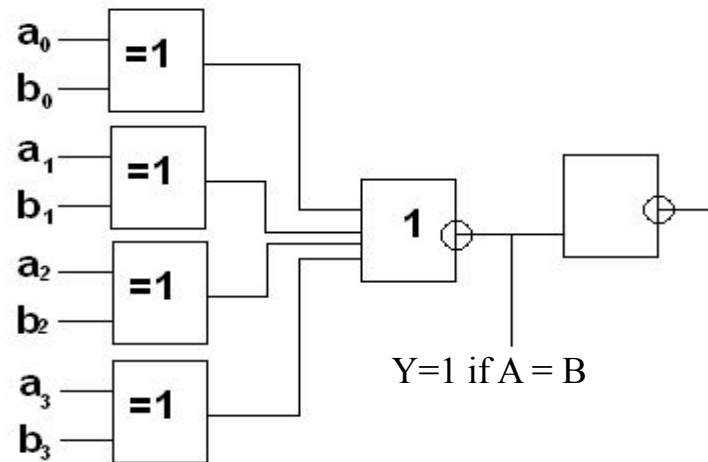


# Полные компараторы



# Многоразрядный компаратор

(пример компаратора, производящего сравнение равно\не равно)

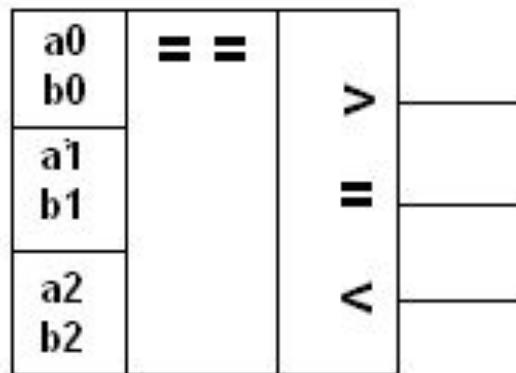
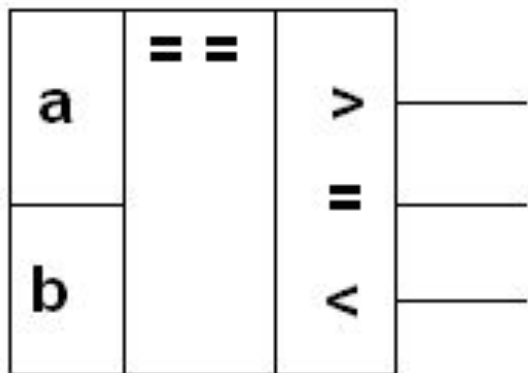


Логическая функция многоразрядного компаратора (равно/не равно)

$$y = a_0 \oplus b_0 \vee a_1 \oplus b_1 \vee a_2 \oplus b_2 \vee \dots$$

$(a_0 \oplus b_0) + (a_1 \oplus b_1) + \dots = \Sigma(a_i \oplus b_i)$ , где  $\Sigma$  – дизъюнкция (логическая сумма)

Обозначение компараторов:



В следующих схемах под двоичным кодом будем понимать любую комбинацию 0 и 1, это может быть число в двоичном виде, номер команды, устройства, адрес, выраженный двоичным числом.

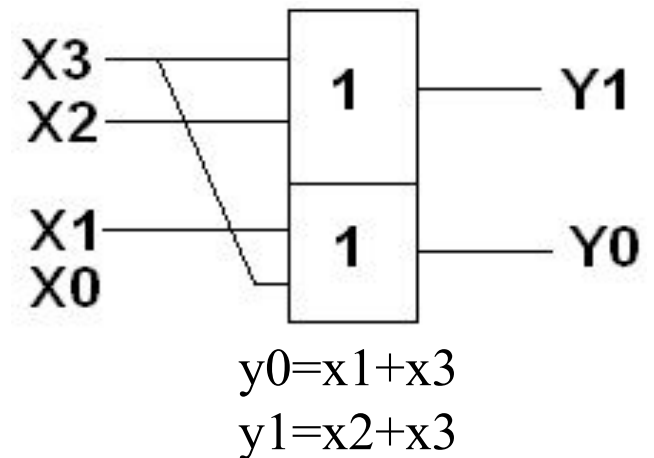
### 3.3 Шифратор

Шифратор – схема, преобразующая сигнал на одном из входов в двоичный код (номера входной линии) на выходе.

Проще: имеется несколько входов, каждому входу присваивается номер 0,1,2,... имеется несколько выходов, каждый соответствует нулевому, первому, второму,... разряду числа (номера). Логическая единица может быть подана только на одном из входов, на выходе должен быть двоичный код номера этой линии. Если выходов  $n$ , то входов может быть  $2^n$ .

Для простоты рассмотрим четырехвходовой шифратор.

| x0 | x1 | x2 | x3 | y1 | y0 | № входа |
|----|----|----|----|----|----|---------|
| 1  | 0  | 0  | 0  | 0  | 0  | 0       |
| 0  | 1  | 0  | 0  | 0  | 1  | 1       |
| 0  | 0  | 1  | 0  | 1  | 0  | 2       |
| 0  | 0  | 0  | 1  | 1  | 1  | 3       |



Шифраторы могут иметь противоположную логику. Например, на всех входах 1 кроме одного, на котором 0, на выходе двоичный код номера линий, где подан 0. Тогда логическая функция:  $y_0 = x_1 + x_3$ ,  $y_1 = x_2 + x_3$ .

Шифраторы используются во многих блоках РС. Например, клавиатура: нажимается клавиша, соответственно на линии появляется 1, которая передается как двоичный код этой клавиши (преобразуется шифратором в двоичный код клавиши.)

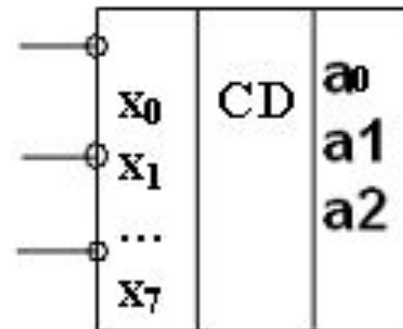
Для восьмивходового шифратора логическая функция выглядит следующим образом:

$$Y_0 = X_1 + X_3 + X_5 + X_7$$

$$Y_1 = X_2 + X_3 + X_6 + X_7$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

Обозначения шифратора

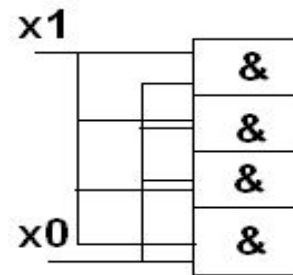


### 3.4 Дешифратор

Дешифратор – схема, преобразующая параллельный код на входах в сигнал на одном из выходов.

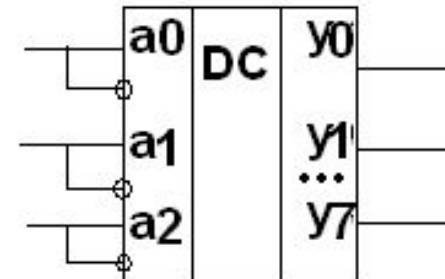
Проще: имеется несколько входов, каждый вход соответствует двоичному разряду. Имеется несколько выходов, каждому выходу приписывается определенный номер, при подаче на вход двоичного кода на одном из выходов, чей номер соответствует поданному на вход двоичному коду, появляется логическая единица, на остальных нули. Если входов  $n$ , выходов  $2^n$ .

| $x_1$ | $x_0$ | $y_0$ | $y_1$ | $y_2$ | $y_3$ |                 |
|-------|-------|-------|-------|-------|-------|-----------------|
| 0     | 0     | 1     | 0     | 0     | 0     | $y_0 = x_1 x_0$ |
| 0     | 1     | 0     | 1     | 0     | 0     | $y_1 = x_1 x_0$ |
| 1     | 0     | 0     | 0     | 1     | 0     | $y_2 = x_1 x_0$ |
| 1     | 1     | 0     | 0     | 0     | 1     | $y_3 = x_1 x_0$ |



Если 3 входа и 8 выходов

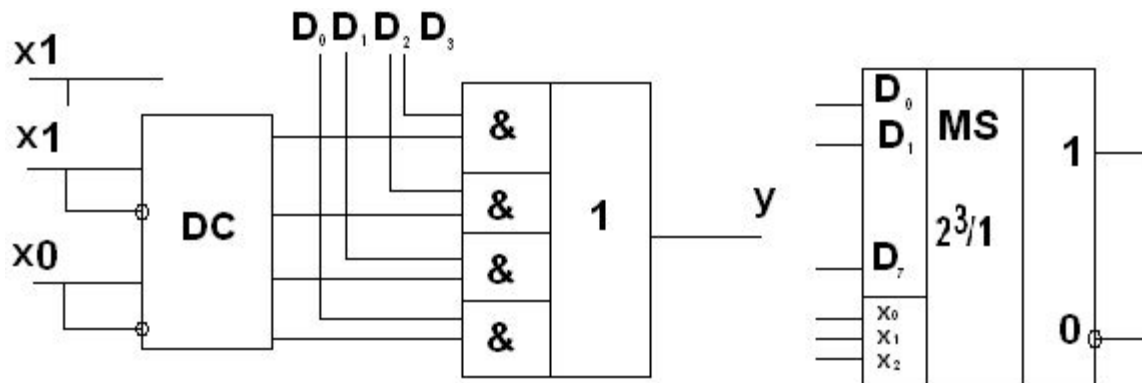
| $x_2$  | $x_1$ | $x_0$ | $y_0$ | $y_1$ | $y_2$ | ... | $y_7$ |  |
|--------|-------|-------|-------|-------|-------|-----|-------|--|
| 0      | 0     | 0     | 1     | 0     | 0     | ... | 0     | $y_0 = \overline{x_2} \overline{x_1} \overline{x_0}$ |
| 0      | 0     | 1     | 0     | 1     | 0     | ... | 0     | $y_1 = \overline{x_2} \overline{x_1} x_0$            |
| 0      | 1     | 0     | 0     | 0     | 1     | ... | 0     | $y_2 = \overline{x_2} x_1 \overline{x_0}$            |
| и т.д. |       |       |       |       |       |     |       |  |



### 3.5 Мультиплексор(селектор) – МХ, MS

Схема, подключающая один или несколько входов к единственному выходу

Проще: имеется несколько входов (информационных), имеющий номер, несколько управляющих входов и один выход. При подаче на управляющие входы двоичного кода номера конкретного информационного входа, этот вход будет подключен к выходу, остальные отключены.



На 4 информационных и 2 управляющих входа:

$$y = D_0x_1x_0 + D_1x_1x_0 + D_2x_1x_0 + D_3x_1x_0$$

На 8 информационных и 4 управляющих входа:

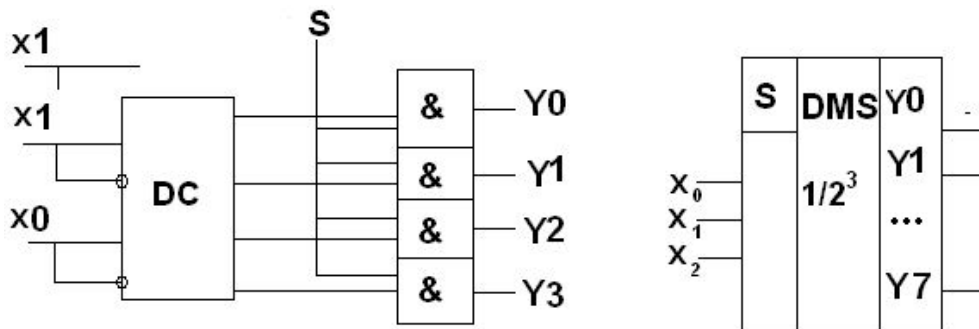
$$y = D_0x_2x_1x_0 + D_1x_2x_1x_0 + D_2x_2x_1x_0 + D_3x_2x_1x_0 + D_4x_2x_1x_0 + D_5x_2x_1x_0 + D_6x_2x_1x_0 + D_7x_2x_1x_0$$

В общем случае  $k$  управляющих и  $2^k$  информационных входов и один выход.

## 3.6. Демультимплексор

Схема, подключающая единственный вход к одному или нескольким выходам.

Проще: имеется один информационный вход, несколько управляющих входов и несколько выходов имеющих свой номер. При подаче двоичного кода на управляющий вход, информационный вход будет подключен к тому выходу, чей номер совпадает с поданным двоичным кодом.



2 управляющих входа, 4 выхода:  $y_0 = Sx_1x_0$ ,  $y_1 = Sx_1x_0$ ,  $y_2 = Sx_1x_0$ ,  $y_3 = Sx_1x_0$ .

3 управляющих входа, 9 выходов:

$y_0 = Sx_2x_1x_0$ ,  $y_1 = Sx_2x_1x_0$ ,  $y_2 = Sx_2x_1x_0$ ,  $y_3 = Sx_2x_1x_0$ ,  $y_4 = Sx_2x_1x_0$ ,  $y_5 = Sx_2x_1x_0$ ,  
 $y_6 = Sx_2x_1x_0$ ,  $y_7 = Sx_2x_1x_0$ .

Иногда используется как дешифратор, а  $S$  – это разрешающий вход, при  $S = 1$  шифратор подключается, при  $S=0$  шифратор отключается, можно на  $S$  например подавать синхронизирующие импульсы.

Важное замечание:

Логические функции всех устройств – одинаковы для всех схем, но сами схемы могут быть и бывают различные. Здесь выбраны наиболее простые для понимания, но не для технической реализации.

## 4. Функциональные узлы последовательного типа

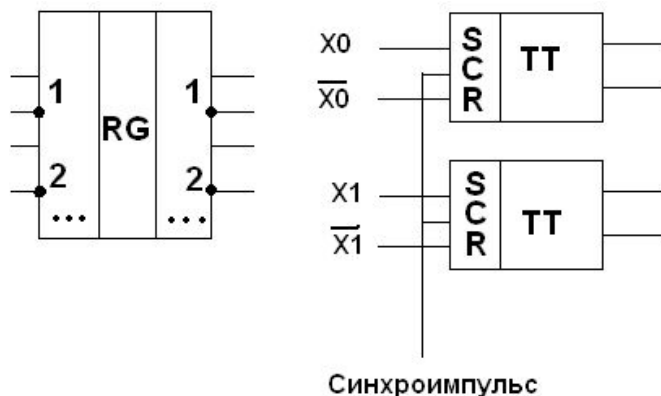
Функциональные узлы, состояние которых определяется последовательностью поступающих входных сигналов называются узлами последовательного типа. Логическое состояние таких узлов определяется комбинацией входных сигналов, как в настоящий момент, так и в предыдущие моменты времени.

### 4.1 Регистры

Триггер предназначен для запоминания 1 бита информации (0 и 1), для запоминания машинного слова используется регистры – устройства, состоящие из нескольких триггеров.

Регистровые устройства предназначены для приема, хранения и передачи в другое устройство машинного слова, а также некоторых операций над ними (сдвиг вправо, влево на требуемое число разрядов, преобразование последовательного кода в параллельный и обратно).

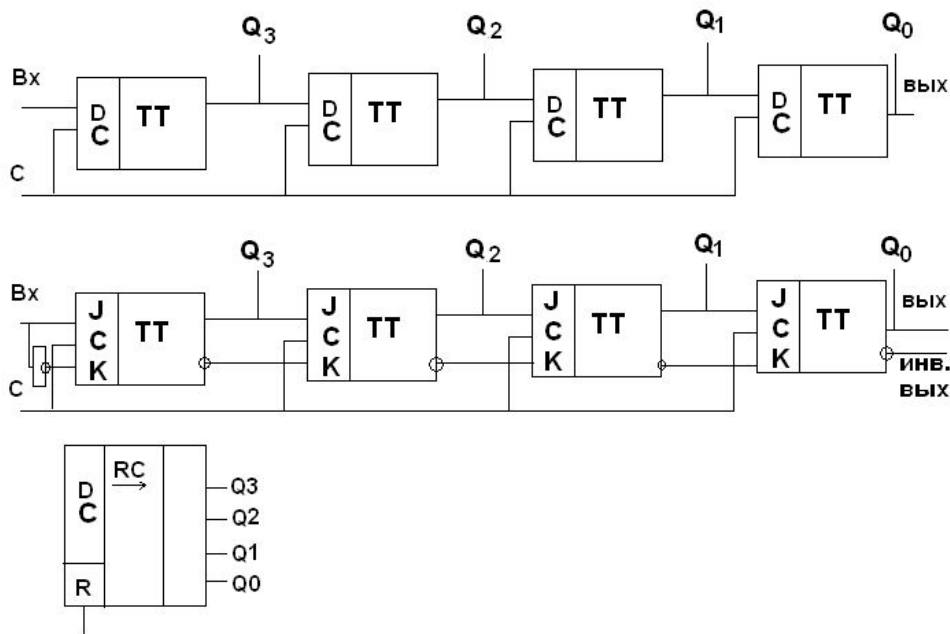
**Параллельный регистр** – регистр хранения, используется только для записи, хранения и передачи информации, может быть построен на основе любого триггера.



Можно на одноклаковых, тогда передача регулируется подачей на вход разрешающего импульса.

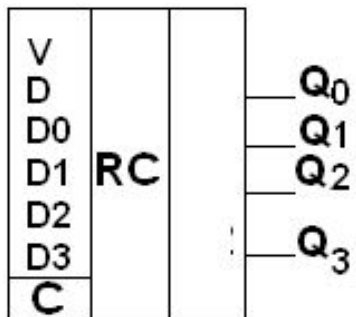


**Последовательный регистр** – регистр сдвига, кроме хранения информации позволяет осуществлять некоторые преобразования. Может быть построен на D, JK, RS двухтактных триггерах.



Ввод в последовательном коде ( $Q_0, Q_1 \dots$ ) через такт. Вывод в параллельном коде через выходы  $Q_0, Q_1$  или в последовательном, для этого нужно подавать синхроимпульсы и на выходе будет появляться  $a_0, a_1$ , и т.д. Можно подавать сигнал в параллельном коде, для этого используют несинхронные входы.

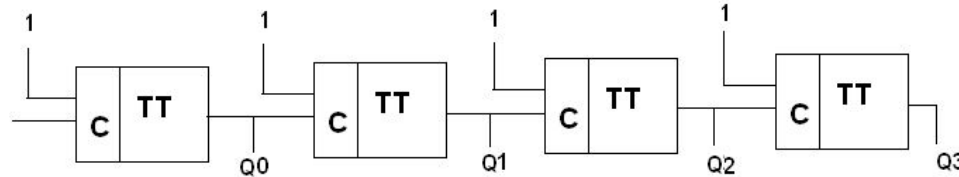
Существуют регистры, осуществляющие сдвиг вправо и влево, а также реверсивные, это совместно с сумматором используется для умножения.



**V** - управляющий вход  
**V=1** - запись  
**V=0** - сдвиг  
**D** - последовательный вход  
**D0, ... ,D3** - параллельный вход

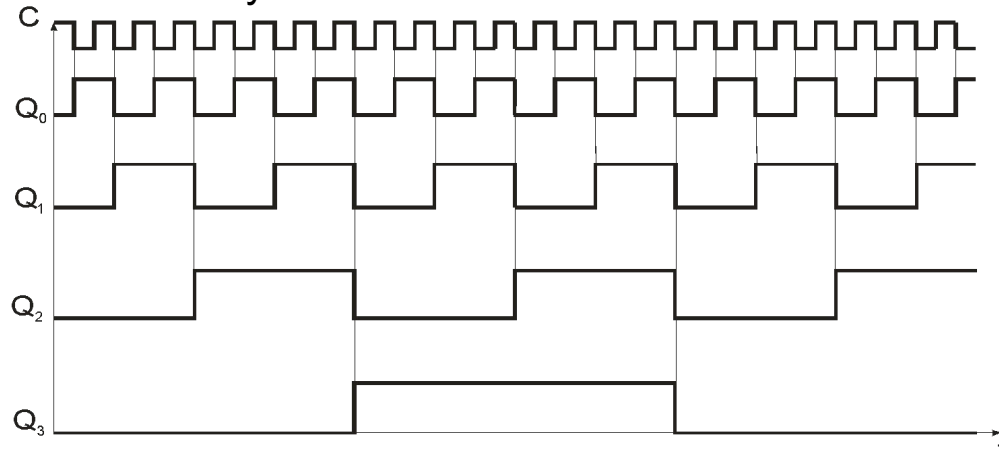
## 4.2 Счетчик

Счетчик – узел предназначенный для подсчета числа входных сигналов, напр, тактов. Есть 4,8,12 разрядные счетчики (считают до 16, 256, 4096), декадные (до 10) десятичные



Вначале все триггеры устанавливаются в нулевое положение

| C   | Q3 | Q2 | Q1 | Q0 |
|-----|----|----|----|----|
| 1   | 0  | 0  | 0  | 1  |
| 2   | 0  | 0  | 1  | 0  |
| 3   | 0  | 0  | 1  | 1  |
| ... |    |    |    |    |
| 15  | 1  | 1  | 1  | 1  |
| 16  | 0  | 0  | 0  | 0  |



Счетчики бывают суммирующие, вычитающие, реверсивные, кольцевые, асинхронные и с дополнительной синхронизацией, а так же с параллельным переносом (для уменьшения задержки).

Обозначение :

