

Дешифратор – функциональный узел комбинационного типа, преобразующий каждый набор двоичных входных сигналов в активный сигнал на выходе (только один) соответствующем этому двоичному набору. Число выходов равно разрешенному количеству входных наборов. В дешифраторе с n входами и N выходами: $N \leq 2^n$. Если $N = 2^n$ - дешифратор полный, если $N < 2^n$ - неполный.

Каждый набор входных сигналов можно поставить в соответствие n-разрядному двоичному числу.

В общем случае обозначение DC n-N. Схемы построения DC: линейные, многоступенчатые и пирамидальные.

Функционирование n -входового дешифратора определяется таблицей истинности. Информационные входы часто называются адресными и обозначаются A_1, A_2, \dots

Входы					Выходы			
EN	A _{n-1}	...	A ₁	A ₀	F ₀	F ₁	F _{N-2}	F _{N-1}
0	×	...	×	×	0	0	0	0
1	0	...	0	0	1	0	0	0
1	0	...	0	1	0	1	0	0
1	0	...	1	0	0	0	0	0
.
1	1	...	0	1	0	0	0	1

$$\begin{aligned}
 F_0 &= EN \cdot \overline{A}_{n-1} \cdot \overline{A}_{n-2} \dots \overline{A}_i \cdot \overline{A}_1 \cdot \overline{A}_0, \\
 F_1 &= EN \cdot \overline{A}_{n-1} \cdot \overline{A}_{n-2} \dots \overline{A}_i \cdot \overline{A}_1 \cdot A_0, \\
 &\dots\dots\dots \\
 F_{N-1} &= EN \cdot A_{n-1} \cdot A_{n-2} \dots A_i \cdot A_1 \cdot A_0,
 \end{aligned}
 \tag{1}$$

где $A_i (i = \overline{0, n-1})$ - двоичные сигналы на входах дешифратора, $F_j (j = \overline{1, N-1})$ - выходные сигналы (функции) дешифратора, EN- сигнал разрешения работы дешифратора (EN=0 – все выходные сигналы неактивные, EN=1 – DC выполняет свою функцию)

$$F_0 = \overline{EN} \vee A_{n-1} \vee A_{n-2} \vee \dots \vee A_1 \vee A_0, \quad \dots$$

$$F_{N-1} = \overline{\overline{EN} \vee \overline{A_{n-1}} \vee \overline{A_{n-2}} \vee \dots \vee \overline{A_1} \vee \overline{A_0}} \quad (2)$$

Компактный вид: $F_j = EN \cdot m_j$, $j = \overline{(0, N-1)}$, где $m_j = m_j(A_{n-1}; A_{n-2}; \dots; A_0)$ – конститuenta единицы, j – номер набора, на котором m_i равна единице.

Параметры дешифратора.

- количество адресных входов n ;
- количество выходов N ;
- входы разрешения работы EN ;

Статические параметры: (в большинстве определяются элементной базой).

- потребляемая мощность $P_{\text{пот}}$ (зависит от элементной мощности и структуры)

Динамические параметры: времена задержки распространения сигнала по каждому тракту от адресного входа до выхода, то разрешающего входа до выхода, $t_{зд.p}^{1,0}$ и $t_{зд.p}^{0,1}$, такие как входные и выходные токи и напряжения логических “0” и “1”, напряжения допустимых статических помех $U_{пом}^0, U_{пом}^1$.

- динамические параметры: времена задержки распространения сигнала при включении и выключении $t_{зд.p}^{1,0}$ и $t_{зд.p}^{0,1}$, характеризующие быстродействие дешифратора.

Быстродействие и потребляемая мощность зависят как от используемой элементной базы, так и от функциональной схемы дешифратора, определяемой способом его построения и количеством выходов.

Функции дешифратора:

- собственно дешифратор,
- демультиплексор (при наличии входа разрешения),
- формирователь произвольной функции ФАЛ при построении комбинационных узлов.

Наращивание дешифраторов.

Наращивание происходит по каскадной схеме.

- 1) Определить количество ступеней DC (количество каскадов).
- 2) Определить количество элементарных дешифраторов в каждом из каскадов.
- 3) Определить порядок соединения и подачи сигналов на каскады.

$$1) P = \left\lceil \frac{n}{n1} \right\rceil - \text{количество ступеней.}$$

- 2) Определение начинается с последнего (выходного) каскада.

$$\frac{N}{N_1}, \frac{N}{N_1^2}, \frac{N}{N_1^3} \text{ пока не станет } < 1.$$

- 4) Каждый каскад управляется $n1$ адресным сигналом, причем всякая ступень управляется младшим разрядом адресного кода, эти младшие разряды подаются на все адресные входы всех дешифраторов выходной ступени. Следующий каскад описывается следующей группой сигналов $n1$. Выходные сигналы предыдущей ступени управляют по входам EN дешифраторами выходной ступени.

$$\text{DC } n1-N1 (3-8), \text{ DC } n-N (7-128), S = \left\lceil \frac{n}{n1} \right\rceil = \left\lceil \frac{7}{3} \right\rceil = 2$$

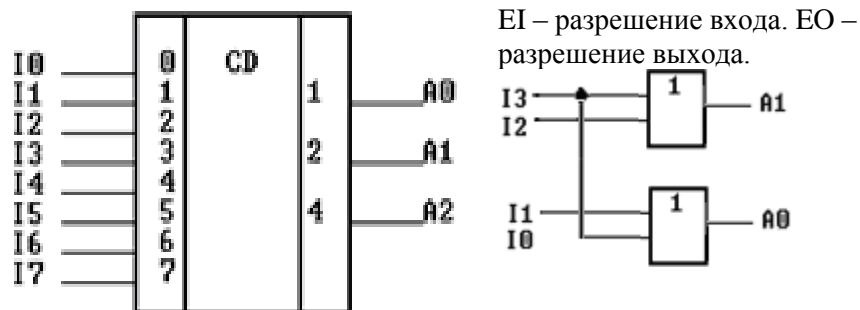
$$P_K = \frac{N}{N1} = \frac{128}{8} = 16,$$

$$P_{K-1} = \frac{P_K}{N1} = \frac{16}{8} = 2,$$

$$P_{K-2} = \frac{P_{K-1}}{N1} = \frac{2}{8} - \text{значит } 1$$

Шифраторы. Приоритетные шифраторы. Синтез. Наращивание.

Шифратор – функциональный узел комбинационного типа, формирующий на выходе 2/2-10 код активизированного входа. Таким образом, шифратор - это комбинационное устройство, реализующее обратную дешифратору функцию. Шифратор (coder) выполняет функцию преобразования унарного кода в двоичный. При подаче сигнала на один из входов (обязательно на один, не более) на выходе появляется двоичный код номера активного входа.



I3	I2	I1	I0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$A1 = I3 + I2 \quad A0 = I3 + I1$$

Недостаток: есть неиспользуемые состояния.

Методика наращивания

1. Входы разбиваются по группам. Приоритеты назначаются и входам и группам.
2. Каждая группа входов образует своим шифратором.

3. Выход ЕО шифратора старшей группы объединяется со входом ЕІ шифратора следующей младшей группы.
4. На вход ЕІ шифратора старшей группы подается сигнал разрешения работы.

Одноименные адресные входы объединяются по ИЛИ.

На выходах схем ИЛИ формируется двоичный код активного кода в данной группе. Если не активен ни один из входов, то на выходе схем ИЛИ формируется сигнал «0».

5. Старшие разряды DC определяются сигналами GS.

Активный сигнал GS одного из шифраторов определяет активизированную группу, т.е. старшие разряды кодов.

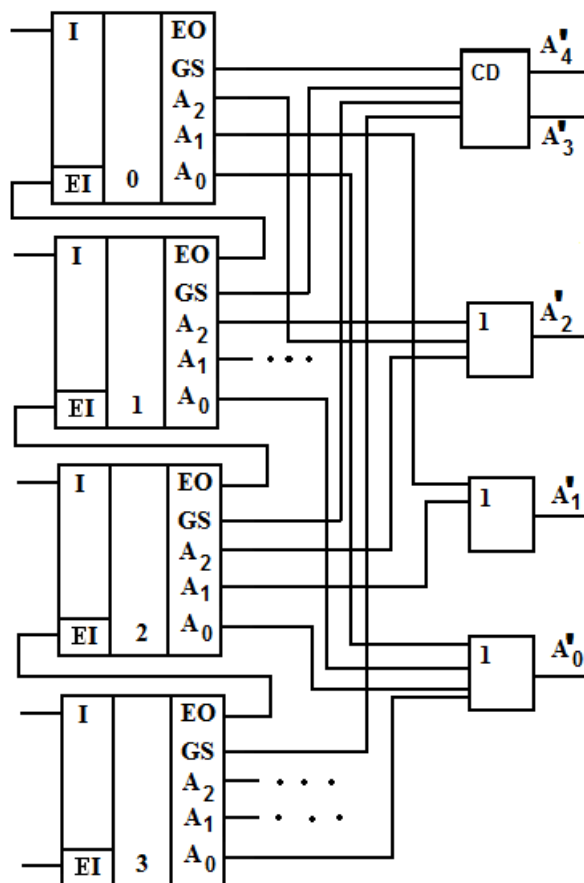
Примечание: шифратор с тремя состояниями позволяет исключить использование ИЛИ при объединении выходов.

Приоритетный шифратор - входам присваивается приоритет- вход с наибольшим номером имеет более высокий приоритет, в нем на выходе формируется 2/2-10 код, активизируемый входом с большим приоритетом. Это реализуется дополнительной логической схемой. Она выделяет активный уровень старшего входа для обеспечения условия работоспособности шифратора (только один уровень на входе активный). Уровни сигналов на остальных входах схемой игнорируются. Схема выделения строится на конъюнкторах и инверторах таким образом, чтобы любое число старших нулей, образуемая после инверторов логические единицы, не влияло на работу конъюнкторов в цепи младших входов. Любая самая старшая единица (после инвертора - логический ноль) запирает конъюнкторы младших входов.

ЕІ –вход разрешения работы.

GS – групповой выход, GS=1 если активизирован хотя бы 1 вход.

ЕО – enable output, ЕО =1 если не активизирован ни один вход дешифратора.



EI	I7	I6	I5	I4	I3	I2	I1	I0	A2	A1	A0	GS	EO
0	X	x	x	x	x	x	x	x	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	1	x	0	0	1	1	0
1	0	0	0	0	1	x	x	x	0	1	0	1	0
1	0	0	0	1	x	x	x	x	1	0	0	1	0
1	0	0	1	x	x	x	x	x	1	0	1	1	0
1	0	1	x	x	x	x	x	x	1	1	0	1	0
1	1	x	x	x	x	x	x	x	1	1	1	1	0

G3	G2	G1	G0	A4	A3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$A2 = EI(I7 + I6 + I5 + I4)$$

$$A1 = EI(I7 + I6 + \text{not } I5 * \text{not } I4 * I3 + \text{not } I5 * \text{not } I4 * I2)$$

$$A0 = EI(I7 + \text{not } I6 * I5 + \text{not } I6 * \text{not } I4 * I3 + \text{not } I6 * \text{not } I4 * \text{not } I2 * I1)$$

$$EO = EI * \bigwedge_{i=0..7} \text{not } I_i$$

$$GS = EI * \text{not } EO$$

Преобразователи кодов ДДК-ДК.

$$A_2 = a_n \cdot 2^n + \dots + a_0 \cdot 2^0 \stackrel{\text{по схеме Горнера}}{=} (((\dots(a_n \cdot 2 + a_{n-1}) \cdot 2 + \dots) \cdot 2 + a_1) \cdot 2 + a_0$$

1. Деление на 2 даёт остаток от деления. Оно реализовано сдвигом.
2. Если единица пересекает границы разрядной сетки, нужна коррекция, она даст 8, а далее, после вычитания 3, получится 5. Признак коррекции- перенос из одной тетрады в другую.

Пример число 576. Представим в ДДК 8421.

Исходное число	0101	0111	0110		
	0010	1011	1011		
коррекция (-3)	0010	1000	1000	0	288
	0001	0100	0100	0	144
	0000	1010	0010		
коррекция (-3)	0000	0111	0010	0	72
	0000	0011	1001		
коррекция (-3)	0000	0011	0110	0	36
	0000	0001	1011		
	0000	0001	1000	0	18
коррекция (-3)	0000	0000	1100		
	0000	0000	1001	0	9
	0000	0000	0100	1	4
	0000	0000	0010	0	2
	0000	0000	0001	0	1
	0000	0000	0000	1	0

Коррекция требуется, когда единица пересекает границу разрядов – отнимаем от этой тетрады тройку.

Для выполнения перевода требуются операции деления сводимые к сдвигу, определения остатка, которым является младший разряд сдвигаемого числа и коррекция. Корректируется тетрада, в которую переносится единица из старшей тетрады. Эта 1 приобретает вес 8, а должна иметь вес 5, поэтому вычитаем из этой тетрады 3.

Таблица истинности элементарного преобразователя

X4	X3	X2	X1	Y4	Y3	Y2	Y1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

$$Y = \begin{cases} X, \text{если } X_4 = 0 \text{ или } 0 \leq X \leq 4 \\ X - 3, \text{если } X_4 = 1 \text{ или } 8 \leq X \leq 12 \end{cases}$$

Неиспользуемые наборы:

0101 =5
0110 =6
0111 =7
1101 =13
1110 =14
1111 =15

Т.к. схема каскадная, то задержка распространения суммируется.

В интегральном исполнении нет преобразователей с 4 входами и 4 выходами, есть К155ПР6.

$$Y = \begin{cases} X, \text{если } 0 \leq X \leq 4 \\ X - 3, \text{если } 8 \leq X \leq 12 \\ X - 6, \text{если } 16 \leq X \leq 20 \\ X - 9, \text{если } 24 \leq X \leq 28 \end{cases}$$

Реализация такого преобразователя на основе ПЗУ.

ПЗУ: 32x8 – 32 восьмиразрядных слова. Т.е. записывается 32 байта.

Сначала строим схему на 4-х входовых преобразователях, а затем заменяем пару их на пятиходовый ПР6.

К155ПР6 м.б. использован для преобразования правильной дроби из двоичной системы в двоично десятичную.

Преобразователи кодов ДК-ДДК.

$$A_2 = a_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0 \stackrel{\text{по схеме Горнера}}{=} ((\dots(a_{n-1} \cdot 2 + a_{n-2}) \cdot 2 + \dots) \cdot 2 + a_1) \cdot 2 + a_0$$

Две операции: умножение на 2 и прибавление двоичного числа, т.е. 0 или 1. Умножение на 2 – это сдвиг влево на один разряд. Само преобразование выполняется в двоично-десятичной разрядной сетке покаскадно (один каскад выполняет только одну операцию). При сдвиге числа, если происходит передача «1» из младшей тетрады в старшую, требуется коррекция. Преобразование ДК в ДДК сводится к введению двоичного числа в двоично-десятич. сетку, начиная со своих старших разрядов. А введение двоичного числа происходит со стороны младших разрядов двоично-десятичного числа.

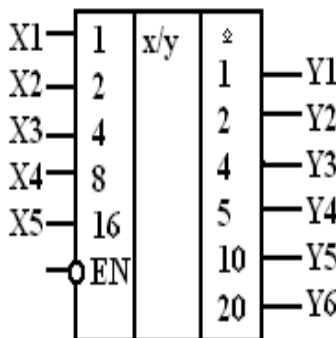
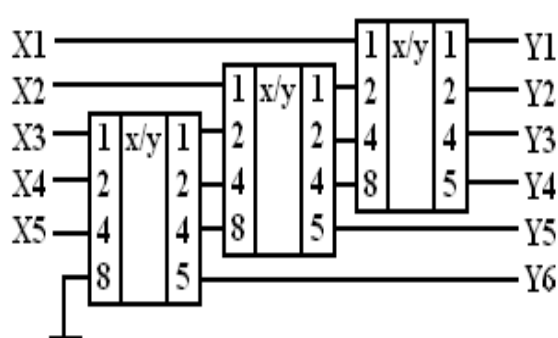
Пример: 11011010(2) → 218(2/10)

10^2	10	1	
		0001	1011010, сдвиг
		0011	011010, сдвиг
0000	0000	0110	11010, коррекция
0000	0000	1001	11010, сдвиг
0000	1	0011	1010, сдвиг
	0010	0111	010, коррекция
	0010	1010	010, сдвиг
0000	0101	0100	10, коррекция
	1000	0100	10, сдвиг
0001	0000	1001	0, коррекция
0001	0000	1100	0, сдвиг
0010	0001	1000	Всё!

Аналитическое выражения для преобразования:

$$Y = \begin{cases} X, & \text{если } 0 \leq X \leq 4 \\ X + 3, & \text{если } 5 \leq X \leq 9 \end{cases}$$

Преобразователь К155ПР7:



$$Y = \begin{cases} X, & \text{если } 0 \leq X \leq 4 \\ X + 3, & \text{если } 5 \leq X \leq 9 \\ X + 6, & \text{если } 10 \leq X \leq 14 \\ X + 9, & \text{если } 15 \leq X \leq 19 \\ X + 12, & \text{если } 20 \leq X \leq 24 \\ X + 15, & \text{если } 25 \leq X \leq 28 \\ X + 18, & \text{если } 29 \leq X \leq 31 \end{cases}$$

ПР6 может быть использован для преобразования целых чисел из ДДК в ДК и правильных дробей из ДК в ДДК. А ПР7 предназначен для преобразования целых чисел из ДК в ДДК и может быть использован для преобразования правильных дробей из ДДК в ДК.

Примечание: когда говорим о ДК, то имеем в виду только один – с весами 8421, т.к. он наиболее удобен для преобразования.

Контроль по четности/нечетности и коду Хемминга. Схемы контроля.

Контроль – проверка правильности функций. Различают аппаратную (с помощью дублирования аппаратуры или параллельным решением задачи - осуществляется во время выполнения задачи) и программную (просчет тестовой задачи, но нужно время + память). Результатом проверки м.б.:

1. обнаружение одиночных ошибок
2. обнаружение и исправление одиночных ошибок
3. обнаружение и исправление одиночных ошибок и ошибок большей кратности

Паритет- соответствие чётный/нечётный.

Контроль по нечетности

Для контроля передачи информации передаваемое слово на передающей стороне дополняется контрольным разрядом. В него записывается 0 или 1 так, чтобы число единиц в кодовом слове было нечетным (или четным для контроля по четности).

Информационное слово $D_7D_6D_5D_4D_3D_2D_1D_0$, контрольный разряд α .

Кодовое слово $D_7D_6D_5D_4D_3D_2D_1D_0\alpha$.

Контроль по нечетности используется чаще, т.к. при контроле по четности невозможно отличить верную передачу от отсутствия передачи (например, при разрыве линии): 0000 – четное число единиц.

На приемной стороне кодовое слово проверяется на четность/нечетность. Для определения четности/нечетности в разрядах слова служат узлы, которые называются узлами свертки.

Значение α при контроле по нечетности $\alpha = F_0$, где F_0 – функция четности ($F_0 = 1$, если число единиц в разрядах информационного слова четно). При контроле по четности $\alpha = F_1$, где F_1 – функция нечетности. Для

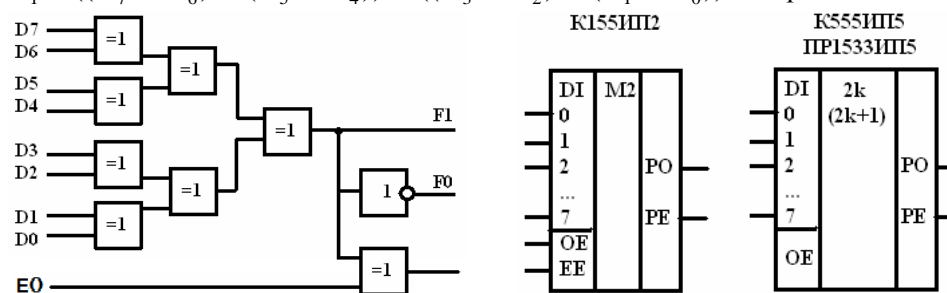
определения F_1 используют сложение по модулю 2: $F_1 = D_7 \oplus D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0$, $F_0 = \overline{F_1}$.

Сложение по модулю 2 не минимизируется.

$F_1 = ((((((D_7 \oplus D_6) \oplus D_5) \oplus D_4) \oplus D_3) \oplus D_2) \oplus D_1) \oplus D_0$ – последовательная реализация.



$F_1 = ((D_7 \oplus D_6) \oplus (D_5 \oplus D_4)) \oplus ((D_3 \oplus D_2) \oplus (D_1 \oplus D_0))$ – пирамидальная схема.



O-odd, E-even, E-enable. OE – разрешение нечетности, EE – четности.

PO – paritet odd (выход нечетности), PE – paritet even (выход четности).

$$t_{зд.п.св} = (\log_2 n)t_{зд.п.сл} \approx 3t_{зд.п.сл}$$

Обозначение схем свертки

Выходы взаимно ортогональны. $PO = OE \oplus DI_0 \oplus DI_1 \oplus \dots \oplus DI_7 = OE \oplus \sum_{i=0}^7 *DI_i$.

Вход OE используется для управления микросхемой и перевода ее в режим четности/нечетности и наращивания схемы.

OE=0: $PO = \sum_{i=0}^7 *DI_i$ – функция нечетности. OE=1: $PO = \overline{\sum_{i=0}^7 *DI_i}$ – функция четности.

При OE=0 значения выходов совпадают со смысловыми названиями.

Функциональная схема контроля по нечетности.

Каждый байт снабжается контрольным разрядом, для которого надо сформировать контрольное значение бита.



DI0..DI7 – информационное слово.

а – дополнительный разряд.

DI0..DI7а – кодовое слово.

Если на принимающей стороне

PO=1 – передача верная при допущении одиночной ошибки (двойные не ловит)

PO=0 – передача неверная, тогда неверное слово может исключаться из обработки, либо запрашивается повторная передача.

Схема контроля по коду Хэмминга: основана на многократном контроле по нечетности.

Позволяет обнаружить и исправить одиночные ошибки, а с помощью модифицированного кода Хэмминга обнаружить ошибки больше кратности и осуществить контроль всего слова.

Допустим, что информационное слово содержит m разрядов. k - число контрольных разрядов, которое надо определить. $n=m+k$ – число разрядов кодового слова.

k должно быть таким, чтобы двоичное число такой разрядности могло указать разряд, в котором ошибка. Если допущена единичная ошибка, то число событий $m+k+1=n+1$, где $m+k$ -суммарное число разрядов, где может быть допущена ошибка. $\Rightarrow k \geq \log_2(m+k+1)$.

Кодирование по Хэммингу: $m=11, k=4$

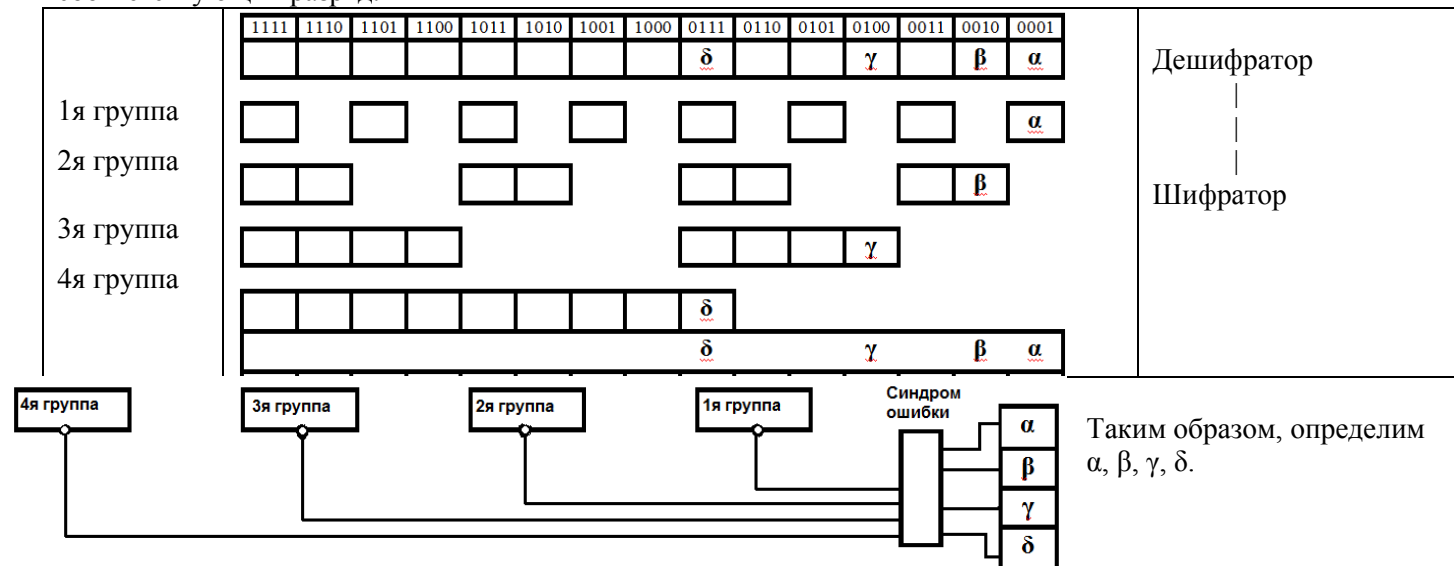
1. Информационное слово дополняется контрольным разрядом
2. Разряды кодового слова нумеруются, начиная с 1 (нумерация в двоичном коде)
3. Контрольные разряды занимают позиции 1, 2, 4, 8 и т.д. (степени двойки)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<input type="checkbox"/>	<input type="checkbox"/>	d1	<input type="checkbox"/>	d2	d3	d4	<input type="checkbox"/>	d5	d6	d7	d8	d9	d10	d11
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

4. Кодовое слово разбивается на группы. Количество групп равно количеству контрольных разрядов. В i группу входят разряды кодового слова, у которого в i позиции двоичный номер разряда содержит «1». Получаем таким образом каждую группу обработанную по нечетному с целью получить значение контрольного разряда (он входит только в одну группу, в отличие от информационных разрядов). Значение контрольного разряда образует слово-синдром ошибки (указывает номер разряда, в котором произошла ошибка). Кодовое слово передается линией связи, на принимающую сторону – проверка на нечетность.

Декодирующее устройство

Каждая группа разрядов слова проверяется на нечетность. В результате формируется синдром ошибки (код – опознаватель разряда, в котором произошла ошибка, если он нулевой, значит, ошибок не было). Синдром ошибки расшифровывается дешифратором, который генерирует управляющий сигнал, который инвертирует соответствующий разряд.



Модифицированный код Хэмминга

Кодовое слово формируется также. Потом полученное кодовое слово дополняется еще одним контрольным разрядом P для контроля передачи всего кодового слова по четности.

$m=n+k$

$n+k+1$

P – дополнительный контрольный разряд, K – синдром для m -разрядного слова

Обнаруживает также групповые ошибки

1. $P=0, K=0$ – ошибка отсутствует
2. $P \neq 0, K \neq 0$ – одиночная ошибка
3. $P=0, K \neq 0$ – четная ошибка, четная групповая ошибка
4. $P \neq 0, K=0$ – нечетная групповая ошибка

K555BЖ1 – данная МС служит для кодирования и проверки 16-разрядного информационного слова. МС выдает сигналы одиночной и многократной ошибки.

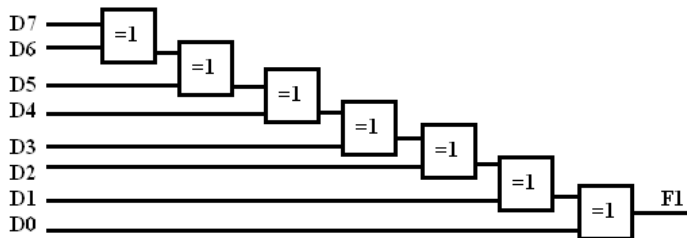
Схемы свертки.

Для определения четности/нечетности в разрядах слова служат узлы, которые называются узлами свертки.

Значение контрольного разряда α при контроле по нечетности $\alpha = F_0$, где F_0 – функция четности ($F_0 = 1$, если число единиц в разрядах информационного слова четно). При контроле по четности $\alpha = F_1$, где F_1 – функция нечетности. Для определения F_1 используют сложение по модулю 2:

$F_1 = D_7 \oplus D_6 \oplus D_5 \oplus D_4 \oplus D_3 \oplus D_2 \oplus D_1 \oplus D_0$, $F_0 = \overline{F_1}$. Сложение по модулю 2 не минимизируется.

$F_1 = ((((((D_7 \oplus D_6) \oplus D_5) \oplus D_4) \oplus D_3) \oplus D_2) \oplus D_1) \oplus D_0$ – последовательная реализация.



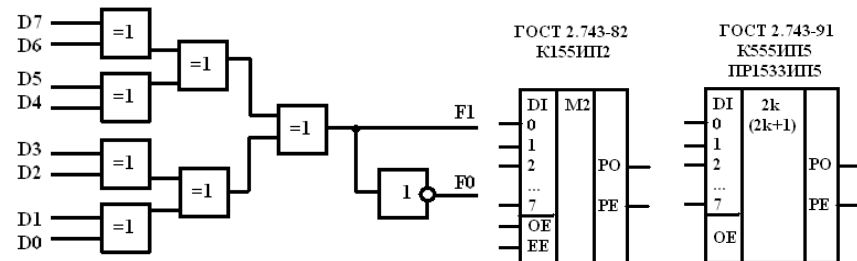
Время задержки распространения свертки зависит от времени задержки распространения в ЛЭ сложения по модулю 2: $t_{зд.п.св} = (n-1)t_{зд.п.сл} \approx 7t_{зд.п.сл}$, n – количество

разрядов информационного слова.

Для повышения быстродействия используют не параллельное, а последовательное включение элементов.

$F_1 = ((D_7 \oplus D_6) \oplus (D_5 \oplus D_4)) \oplus ((D_3 \oplus D_2) \oplus (D_1 \oplus D_0))$

- пирамидальная схема.



$t_{зд.п.св} = (\log_2 n)t_{зд.п.сл} \approx 3t_{зд.п.сл}$ Обозначени

е схем свертки: O-odd, E-even, E-enable. OE – разр. нечетности, EE – четности. PO – paritet odd (выход нечетности), PE – paritet even (выход четности).

Выходы взаимно ортогональны.

$$PO = OE \oplus DI_0 \oplus DI_1 \oplus \dots \oplus DI_7 = OE \oplus \sum_{i=0}^7 DI_i$$

Вход OE используется для управления микросхемой и перевода ее в режим четности/нечетности и наращивания схемы.

OE=0: $PO = \sum_{i=0}^7 DI_i$ – функция нечетности. OE=1: $PO = \overline{\sum_{i=0}^7 DI_i}$ – функция четности.

При OE=0 значения выходов совпадают со смысловыми названиями.

Функциональная схема контроля по нечетности.

Каждый байт снабжается контрольным разрядом, для которого надо сформировать контрольное значение бита.



DI0..DI7 – информационное слово.

a – дополнительный разряд.

DI0..DI7a – кодовое слово.

Если на принимающей стороне

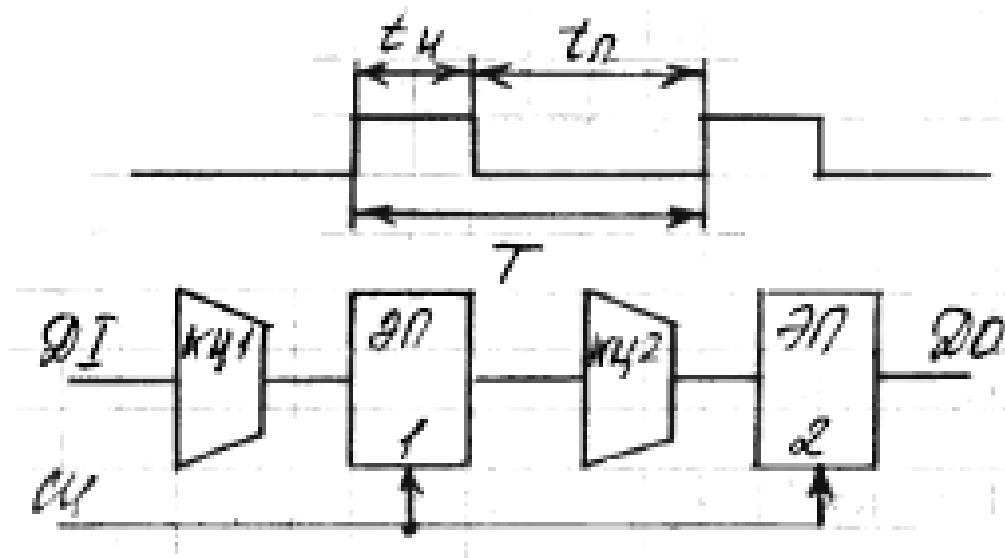
PO=1 – передача верная при допущении одиночной ошибки (двойные не ловит)

PO=0 – передача неверная, тогда неверное слово может исключаться из обработки, либо запрашивается повторная передача.

Система синхронизации. Основные параметры. Структура. Однофазная и двухфазная синхронизация.

Однофазная система синхронизации. Основные параметры и соотношения.

$K_\phi = 1$. Главным преимуществом однофазной синхронизации является простота конструкции и высокое быстродействие.



Триггеры:

1) синхронные одноступенчатые статические

$$t_u \geq t_{\text{зап.тр.макс}}, t_{\text{зап.тр.}} \approx 3t_{\text{зд.р.ЛЭ}}, t_u \leq t_{\text{зап.тр.мин}} + t_{\text{зд.р.КЦ.мин}}$$

$$t_n \geq t_{\text{зд.р.КЦ}} + t_{\text{линий_связи}} + t_{\text{пу}}$$

Для большого количества элементов такая организация практически нереализуема, следовательно, их не применяют в однофазной синхронизации.

2) синхронные одноступенчатые с динамическим управлением записью

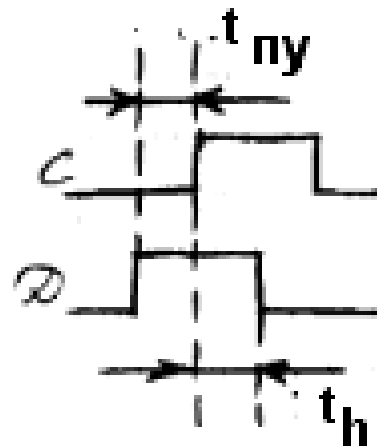
$$t_u \geq t_{\text{зап.тр.макс}}, t_{\text{зап.тр.}} \approx 3t_{\text{зд.р.ЛЭ}}$$

$$t_n \geq t_{\text{зд.р.КЦ}} + t_{\text{линий_связи}} + t_{\text{пу}},$$

$$t_{\text{пу}} = 5 \div 6 \text{ нс}$$

$$t_h = t_{\text{зд.р.ЛЭ}} - \text{время выдержки (hold)}$$

$$t_{\text{зап.тр.мин}} + t_{\text{зд.р.КЦ.мин}} \geq t_{h\text{макс}}$$



3) синхронные с двухступенчатым запоминанием информации

Характерной чертой таких схем (при правильном подборе параметров) является отсутствие временных состязаний. Однако такая схема приводит к снижению быстродействия (необходимо дополнительное время для переключения выходных ступеней).

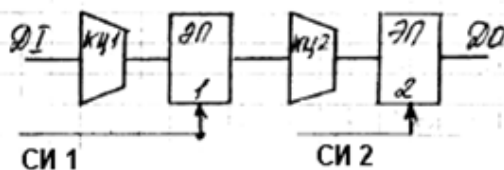
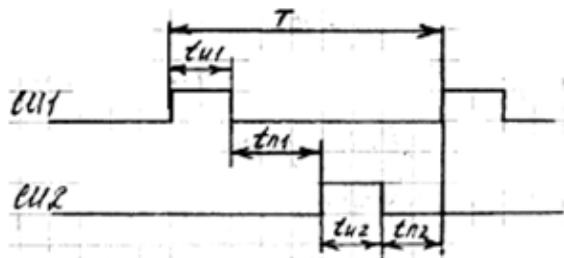
$$t_u \geq t_{\text{зап.тр.I_ст}}$$

$$t_n \geq t_{\text{зап.тр.II_ст}} + t_{\text{зд.р.КЦ}} + t_{\text{линий_связи}} + t_{\text{пу}}$$

Двухфазная система синхронизации. Основные параметры и соотношения.

В данном случае используются две последовательности СИ, сдвинутые друг относительно друга. Соседние каскады получают разноимённые серии СИ.

Могут применяться синхронные одноступенчатые триггеры со статическим управлением записи (можно и другие, но это будет избыточность).



$$t_{u1} \geq t_{\text{зап.тр.}}$$

$$t_{n1} \geq t_{\text{зд.р.КЦ1}} + t_{\text{линий_связи_1}} + t_{\text{пу}}$$

$$t_{u2} \geq t_{\text{зап.тр.}}$$

$$t_{n2} \geq t_{\text{зд.р.КЦ2}} + t_{\text{линий_связи_2}} + t_{\text{пу}}$$

Система синхронизации. Основные параметры. Структура.

Стробирование – выделение из полезного сигнала той его части, которая свободна от помех, вызванных гонками сигналов. Стробирование всегда ортогонально сигналу помех, т.е. обеспечивается независимость сигнала от помех.

Процесс стробирования периодическими сигналами на входах элементов памяти (триггеров) называется синхронизацией. А сами сигналы стробирования называются синхронизирующими.

Существуют 2 вида схем обработки информации: синхронные и асинхронные.

В асинхронных схемах все процессы преобразования информации выполняются под действием информационных сигналов, а задержка в таких схемах определяется естественным образом (как сигнал проходит через схему).

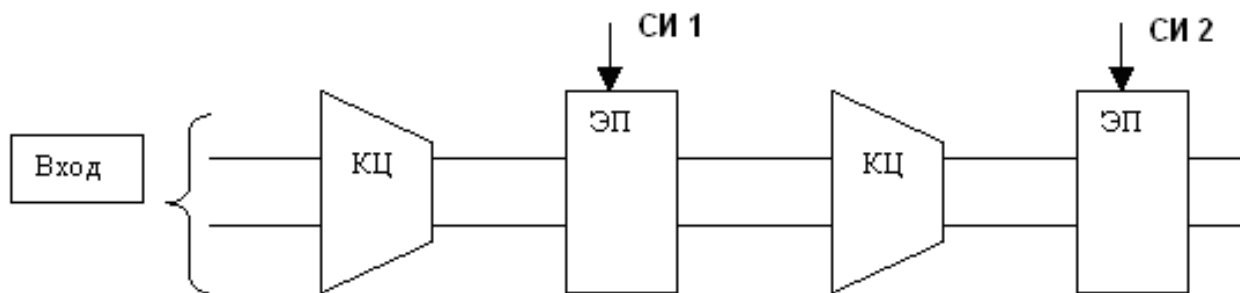
В синхронных схемах процессы обработки информации синхронизируются, т.е. выполняются под действием синхросигнала. Период действия синхросигнала выбирается так, чтобы во время его действия переходные процессы были закончены.

Синхронные схемы обладают меньшим быстродействием, чем асинхронные. Асинхронные применяются шире.

Все цифровые устройства делятся на комбинационные цепи (КЦ) и автоматы с памятью (АП). Выходная переменная КЦ зависит лишь от действующих на входах текущих значений входных переменных. Выходная переменная АП определяется не только действующими на входах значениями переменных, но и состоянием узла в момент прихода входных переменных. АП состоят из двух частей: элементы памяти (ЭП) и КЦ для управления памятью и выработки выходных переменных. К КЦ относят преобразователи кодов, шифраторы, дешифраторы, компараторы и сумматоры. АП – это триггеры, регистры, счётчики и т.д.

В АП из-за задержек сигналов могут наблюдаться гонки (состязания) сигналов, при которых последующее состояние узла определяется не всем набором входных переменных, а лишь самым быстрым сигналом. Если в результате нарушается нормальная работа АП, то гонки являются критическими.

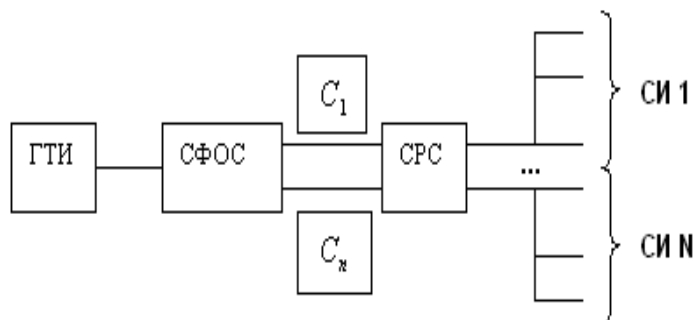
Самый распространённый механизм гонок таков. Состояние АП изменяется сигналами, зависящими от состояния ЭП. Из-за задержки сигналов установка ЭП не происходит одновременно, и одни из ЭП изменяют состояние раньше других. При этом они влияют на установку других ЭП или даже переводят их в непредусмотренное состояние.



Основные параметры схемы синхронизации:

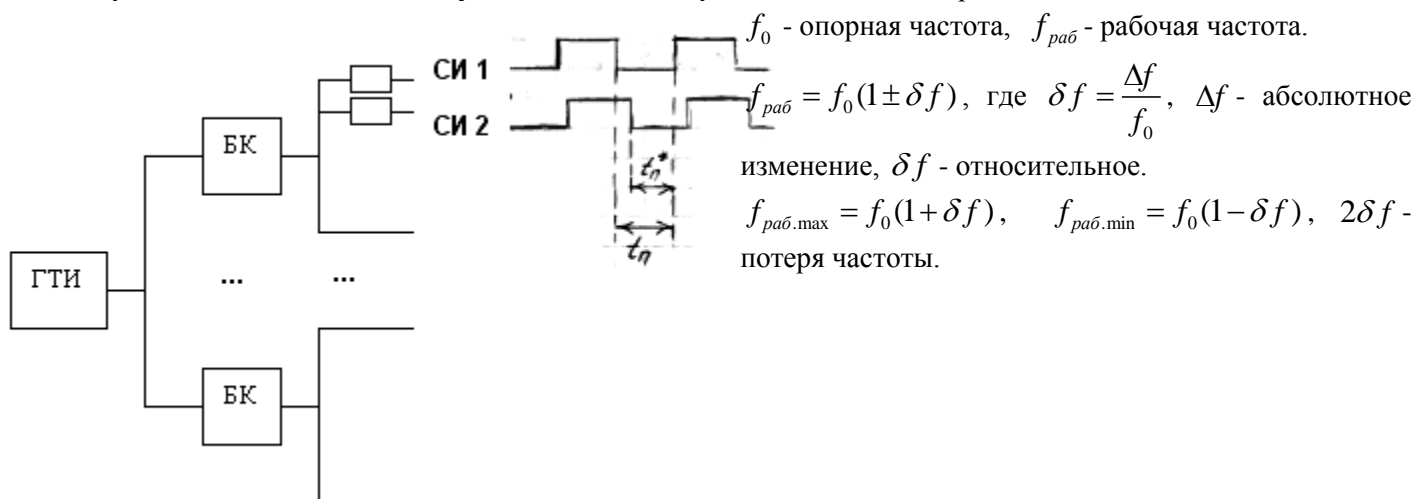
- 1) t_u - длительность импульса. За длительность импульса информация записывается в триггере, т.е. $t_u \geq t_{\text{зан.тр.макс}}$. Но $t_{\text{зан.тр.}} \approx t_{\text{зд.р.тр.}}$, значит $t_u \geq t_{\text{зд.р.тр.}}$. Если элементы КЦ и ЭП безинерционные (задержка = 0), то синхронизация невозможна.
- 2) t_n - длительность паузы. $t_n \geq t_{\text{зд.р.КЦ макс}} + t_{\text{линий связи}} + t_{\text{пу макс}}$
- 3) $T = t_u + t_n$ - период тактового генератора.
- 4) фазность – определяется количеством серий импульсов, формируемых в течение одного периода системой синхронизации.

Структура схем синхронизации.



ГТИ – генератор тактовых импульсов.
СФОС – схема формирования опорных синхросигналов.
СРС – схема размножения синхросигнала. Для обеспечения требуемой нагрузочной способности.
БК – буферные каскады.
Но может идти временное рассогласование этих серий.

Зачастую необходимо обеспечить тактовыми импульсами большое число ЭП. Импульсы должны задаваться одним ГТИ, а использоваться тысячами ЭП. Использовать мощный генератор нельзя из-за помех. Наилучшее решение – размножение тактовых импульсов с помощью разветвляющейся пирамидальной схемы.



Синхронные счетчики. Основные параметры. Методика синтеза.

Счетчик – ФУ, выполняющий функции счёта числа событий(импульсов) и хранения этого числа в определенной системе счисления. Событие- изменение сигнала 0->1 и обратно.

Различают счетчики и их разновидность – пересчетные устройства. Они преобразуют последовательность входных сигналов в предписанную последовательность состояний. Счетчик же- совокупность элементов памяти и функциональной схемы. Для пересчетной схемы не важен способ кодирования. Состояние счетчика – конкретное значение цифрового кода, которое формируется на выходах триггеров (разрядов) счетчика. Одно из состояний принимается за начальное, часто нулевое.

Основные параметры счетчиков:

I) Статические: такие же как у его элементной базы: $U_{ВХ}^0, U_{ВХ}^1, U_{ВЫХ}^0, U_{ВЫХ}^1, U_{ПИТ}, P_{ПОТР}$, логический перепад и прочие..

II) Динамические:

1) частота счета максимальная и рабочая ($1/10 \max$)

2) задержки распространения сигналов от вх. до вых. по различным трактам $t_{зд.р.}, t_{зд.р.С-Q}, t_{зд.р. R-Q}$

3) время предустановки

4) время удержания информационного сигнала на определенных входах по отношению к управляющему сигналу

5) частота переключений

III) Специфические: Модуль счета М и емкость счетчика Е. М определяется числом используемых состояний(числом импульсов, которое возвращает счетчик в начальное состояние).

$M = q^n$, (q- основание системы счета, n- количество разрядов).

$E = M - 1$ - количество событий, которое может быть зафиксировано в счетчике(кроме 0).

Методика синтеза синхронных счетчиков с произвольным модулем счёта.

При синтезе счетчика задаются: 1) модуль счета; 2) может быть задан порядок изменения состояния; 3) может быть задан тип триггера. Если 2) не задается то по усмотрению разработчика (естественный). Требуется построить схему.

Основные положения методики:

1. Определяется количество триггеров n. $n = \lceil \log_2 M \rceil$, где М – наибольший номер состояния. Если модуль

счета М и порядок счета естественный, то $n = \lceil \log_2 M \rceil$, если порядок счета неестественный, то

$n = \lceil \log_2 M_1 \rceil$, где M_1 – наибольший номер состояния.

2. Составляется таблица переходов и функция возбуждения.

Пример: $M=5$ суммирующий с естественным порядком изм. сост., $n = \lceil \log_2 5 \rceil = 3$

N	Q2	Q1	Q0	Q2*	Q1*	Q0*	J2	K2	J1	K1	J0	K0	D2	D1	D0
0	0	0	0	0	0	1	0	a	0	a	1	a	0	0	1
1	0	0	1	0	1	0	0	a	1	a	a	1	0	1	0
2	0	1	0	0	1	1	0	a	a	0	1	a	0	1	1
3	0	1	1	1	0	0	1	a	a	1	a	1	1	0	0
4	1	0	0	0	0	0	a	1	0	a	0	a	0	0	0

$$K_0=1, J_0 = \overline{Q_2},$$

$$K_1 = Q_0,$$

$$J_1 = Q_0,$$

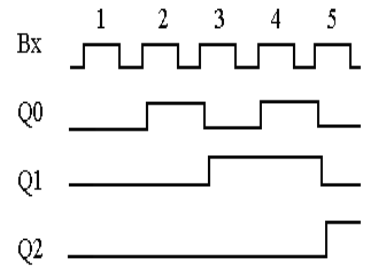
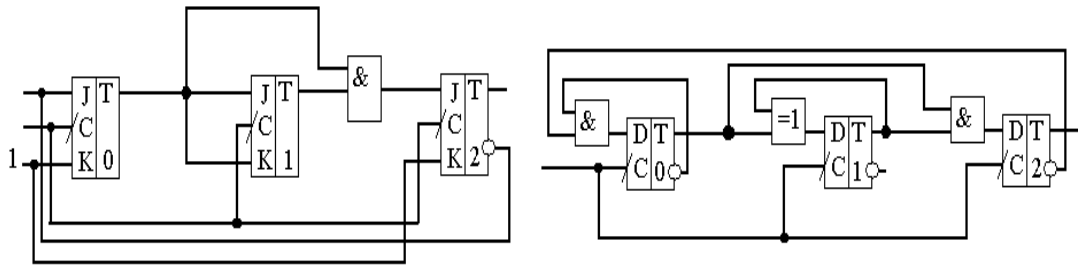
$$K_2=1, J_2 = Q_0 Q_1$$

Q2, Q1 \ Q0	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10	
0	0	1	1	b	0	a	0	b	a	0	a	b	0	0	0	b	a
1	1	a	a	b	b	a	1	b	b	1	a	b	b	0	1	b	b

$$D_2 = Q_0 Q_1, D_1 = Q_0 \overline{Q_1} + \overline{Q_0} Q_1 = Q_0 \oplus Q_1, D_0 = \overline{Q_0} \overline{Q_2}$$

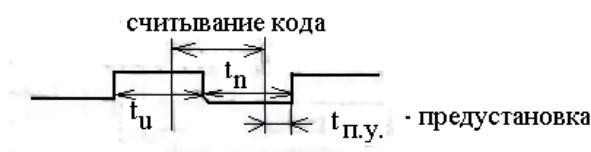
Q2, Q1	00	01	11	10	00	01	11	10	00	01	11	10
--------	----	----	----	----	----	----	----	----	----	----	----	----

Q0 \													
0	0	0	0	b	0	0	1	b	0	1	1	b	0
1	0	1	b	b	1	0	b	b	0	0	b	b	



Параметры быстродействия:

$$t_{зд.р.С-Q} = t_{зд.р.мр.}$$



$$T = t_{зд.р.мр.} + t_{зд.р.к.с.} + t_{n.у.}$$

$$t_u = t_{зд.р.мр.}$$

$$t_n = t_{зд.р.к.с.} + t_{n.у.}$$

$$t_u > t_{зд.р.мр.}$$

Обычно $t_{n.у.}$ в справочных данных.

Синхронные триггеры. Принцип работы триггеров со статическим и динамическим управлением записью. Применение в регистрах и счетчиках.

Синхронный триггер со статическим управлением записью (одноступенчатый)

При $C=0$ триггер находится в режиме хранения информации, при $C=1$ переключается в состояние, определенное информационным сигналом. В качестве 3Я – асинхронный RS-триггер.

Q \ SR	00	01	11	10
0			X	1
1	1		X	1

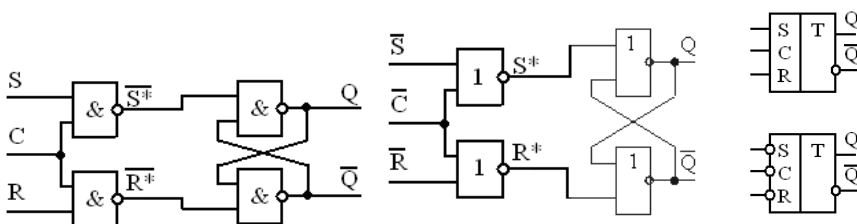
$$Q^* = S + \bar{R}Q$$

$$\bar{Q}^* = R + \bar{S}\bar{Q}$$

$$\text{В базисе И-НЕ: } Q^* = \overline{\overline{S} + \overline{\bar{R}Q}} = \overline{\bar{S} \cdot \bar{R}Q}$$

$$\text{В базисе ИЛИ-НЕ: } \bar{Q}^* = \overline{R + \bar{S}\bar{Q}} = \overline{R + S + Q}$$

Реализация синхронного RS-триггера со статическим управлением в базисах И-НЕ, ИЛИ-НЕ: (СУ+ЗЯ)



C	S	R	Q_n	Q_{n+1}	S^*	R^*
0	0	0	0	0	0	α
0	0	0	1	1	α	0
0	0	1	0	0	0	α
0	0	1	1	1	α	0
0	1	0	0	0	0	α
0	1	0	1	1	α	0
0	1	1	0	0	0	α
0	1	1	1	1	α	0
1	0	0	0	0	0	α
1	0	0	1	1	α	0
1	0	1	0	0	0	α
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	0	1	1	α	0
1	1	1	0	X	β	β
1	1	1	1	X	β	β

CS					CS				
					RQ				
	00	01	11	10		00	01	11	10
00			1		00	α	α		α
01	α	α	α	α	01				
11	α	α	β		11			β	1
10			β		10	α	α	β	α

$$S^* = CS, \quad \overline{S^*} = \overline{CS} \quad R^* = CR, \quad \overline{R^*} = \overline{CR}$$

$$\text{В базисе И-НЕ: } \overline{S^*} = \overline{CS}, \quad \overline{R^*} = \overline{CR}$$

$$\text{В базисе ИЛИ-НЕ: } S^* = \overline{\overline{CS}} = \overline{\overline{C} + \overline{S}}, \quad R^* = \overline{\overline{C} + \overline{R}}$$

Синхронный Rs-триггер со статическим управлением (двухступенчатый).

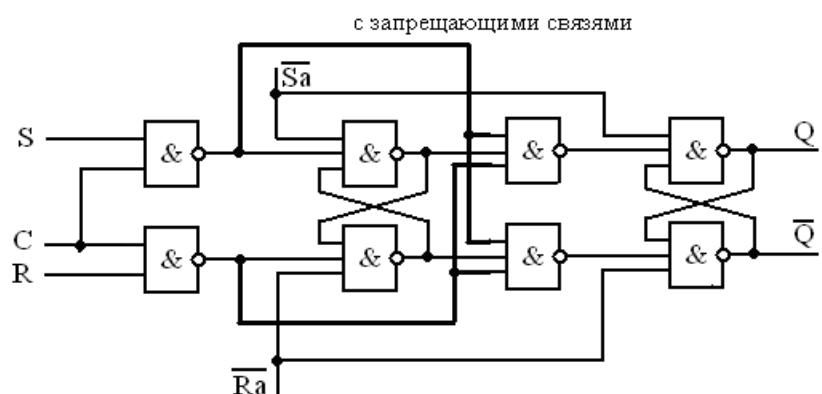
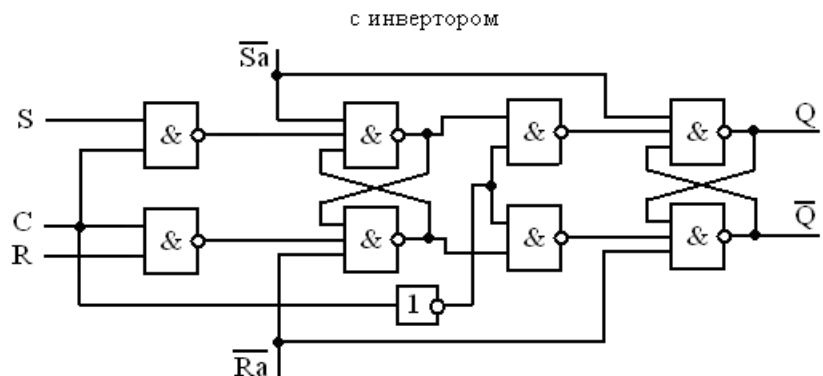
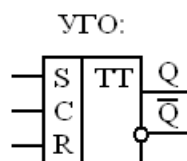
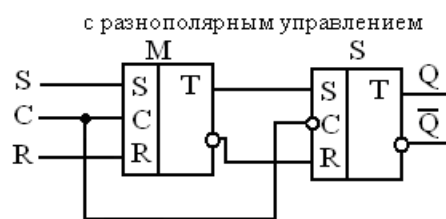
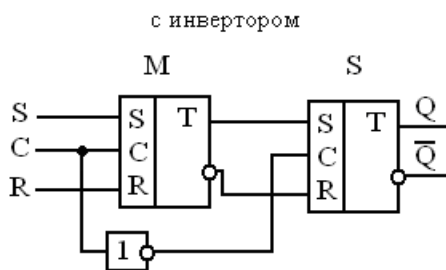
В двухступенчатых триггерах во входной и выходной ступенях прием информации разрешается поочередно. Их схемы более сложные, чем с динамическим входом, а быстродействие несколько ниже.

Двухступенчатые триггеры строятся несколькими способами: с инвертором, с разнополярным управлением ступенями, с запрещающими связями. В данном случае обе ступени – синхронные RS-триггеры.

MS-триггера (триггер типа Flip-Flop)

M – Master, S- Slave.

При C=0 перепись информации из первой ступени во вторую и перевод первой ступени в режим хранения. При C=1 разрешена запись в первую ступень и запрещена перепись из первой ступени во вторую.



Синхронный JK-триггер с динамическим управлением записью

$$Q^* = J\bar{Q} + \bar{K}Q, D=Q^* = J\bar{Q} + \bar{K}Q$$

Из D-триггера получаем JK.

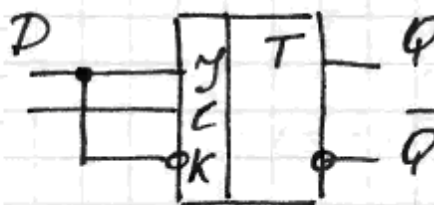
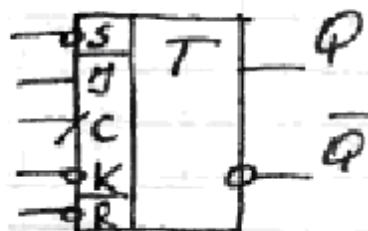
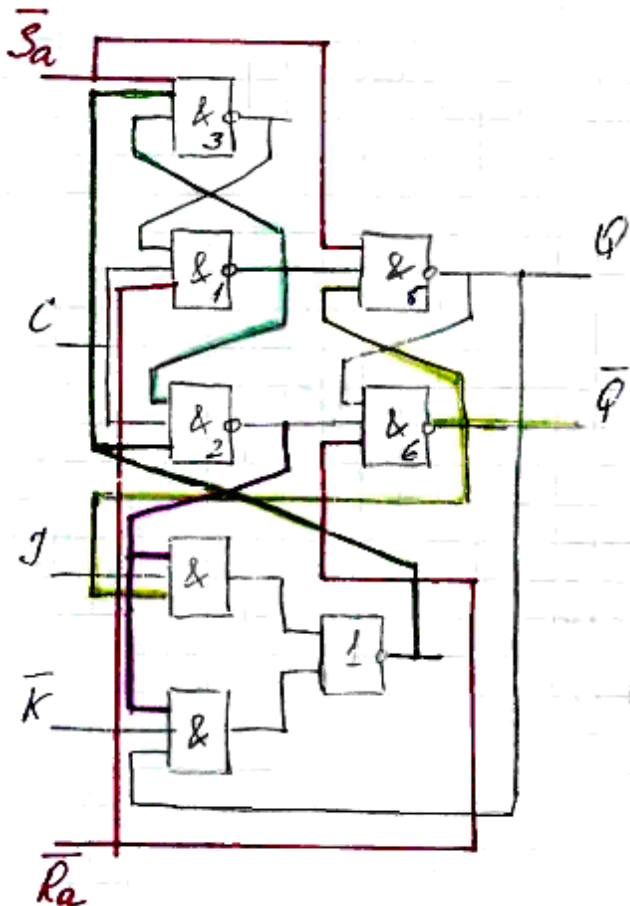
J=0, K=1 – режим хранения

J=0, K=0 – режим записи 0

J=1, K=1 – режим записи 1

J=1, K=0 – счетный режим

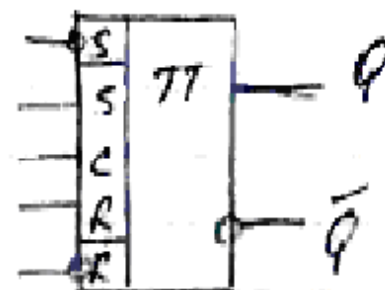
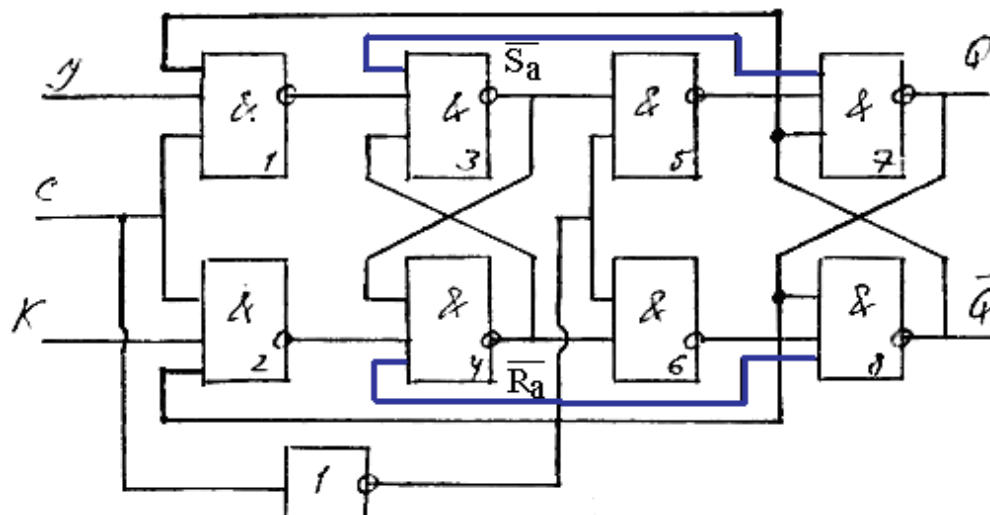
Если объединить входы J и K, то получится D-триггер

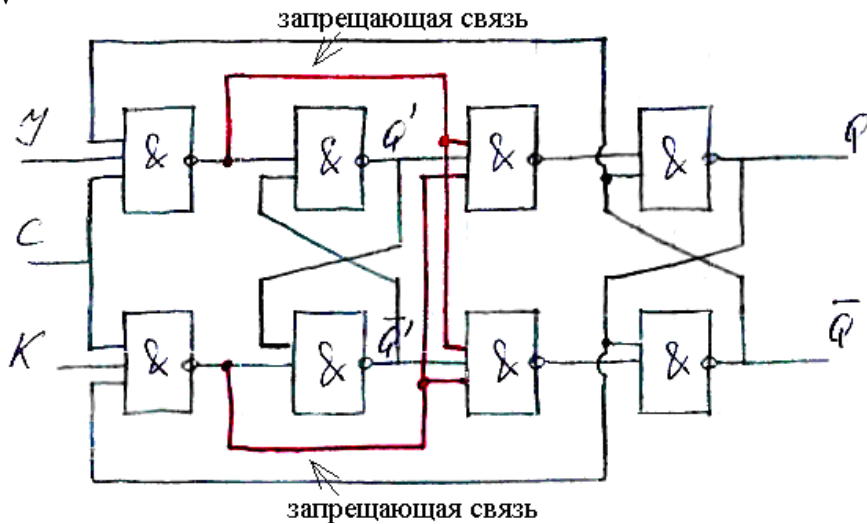


Синхронный JK-триггер со статическим управлением (двухступенчатый)

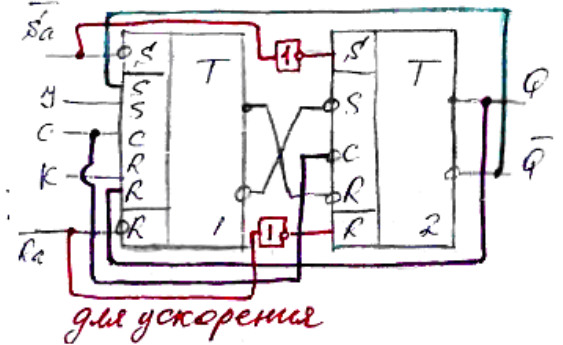
Двухступенчатые триггеры. При C=1 разрешается запись в 1ю ступень и запрещается перепись из 1й во 2ю. При C=0 происходит перепись информации из 1й ступени во 2ю, и перевод 1й ступени в режим хранения.

$$S' = CJ\bar{Q}, R' = CKQ, \bar{S}' = C\bar{J}Q, \bar{R}' = C\bar{K}\bar{Q}$$



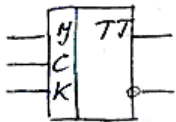


Разнополярное управление: I ступень имеет управляющие сигналы высокого уровня, а II ступень - низкого уровня



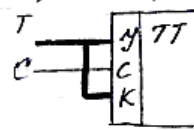
Ф-из, кот-е может входить JK-триггер: 3) синхронный T-триггер

1) RS-триггер на кадрах $J=K=0$

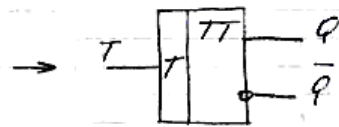
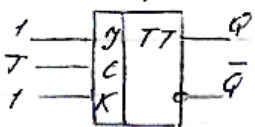


$$J=1, K=0$$

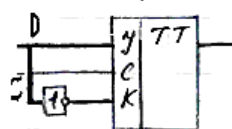
$$J=0, K=1$$



2) асинхронный T-триггер



4) D-триггер синхронный



По сигналу с записыва-
ем по, что посылка
на D.

Серия ТВ15 имеет J прямой и
K инверсный \Rightarrow D-триггер

C	J	K	Q_n	Q_{n+1}	S^*	R^*
0	0	0	0	0	0	α
0	0	0	1	1	α	0
0	0	1	0	0	0	α
0	0	1	1	1	α	0
0	1	0	0	0	0	α
0	1	0	1	1	α	0
0	1	1	0	0	0	α
0	1	1	1	1	α	0
1	0	0	0	0	0	α
1	0	0	1	1	α	0
1	0	1	0	0	0	α
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	0	1	1	α	0
1	1	1	0	1	1	0
1	1	1	1	0	0	1

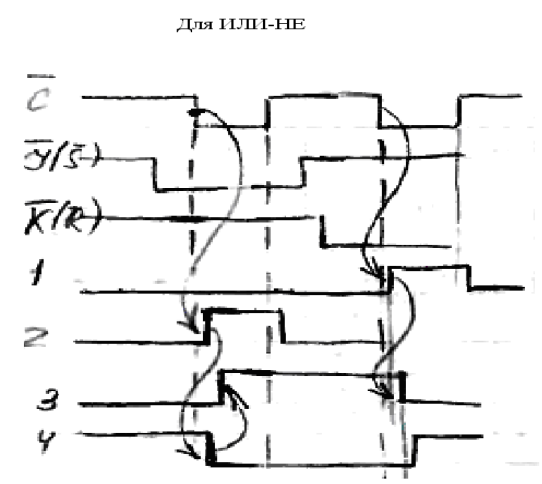
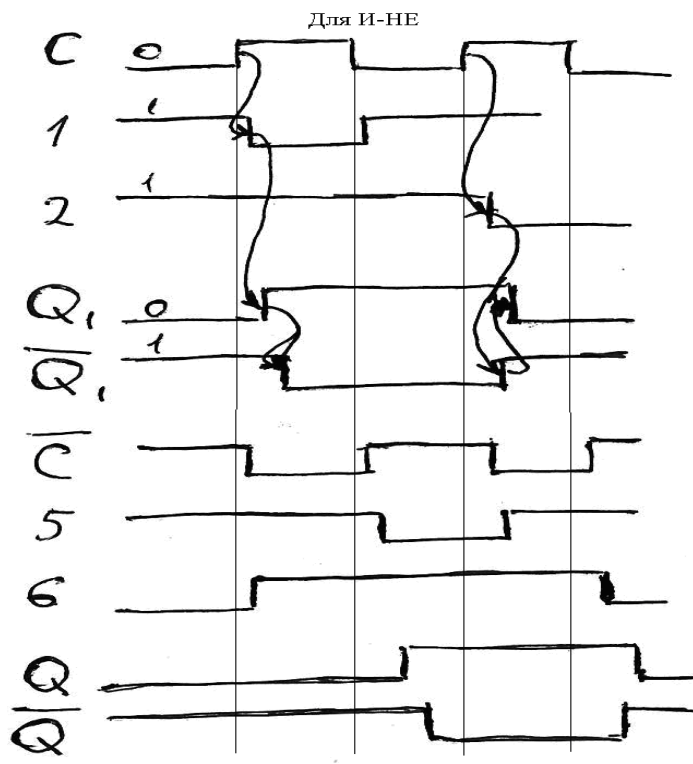
CJ					CJ				
KQ	00	01	11	10	KQ	00	01	11	10
00			1		00	α	α		α
01	α	α	α	α	01				
11	α	α			11			1	1
10			1		10	α	α		α

$$S^* = CJ\bar{Q}, \bar{S}^* = \overline{CJ\bar{Q}}$$

$$R^* = CKQ, \bar{R}^* = \overline{CKQ}$$

В базисе И-НЕ: $\bar{S}^* = \overline{CJ\bar{Q}}, \bar{R}^* = \overline{CKQ}$

В базисе ИЛИ-НЕ: $S^* = \overline{\overline{CJ\bar{Q}}} = \overline{\bar{C} + \bar{J} + Q}, R^* = \overline{\bar{C} + \bar{K} + \bar{Q}}$



$$t_{\text{зг.р. I}} = 3 t_{\text{зг.р. лэ}}$$

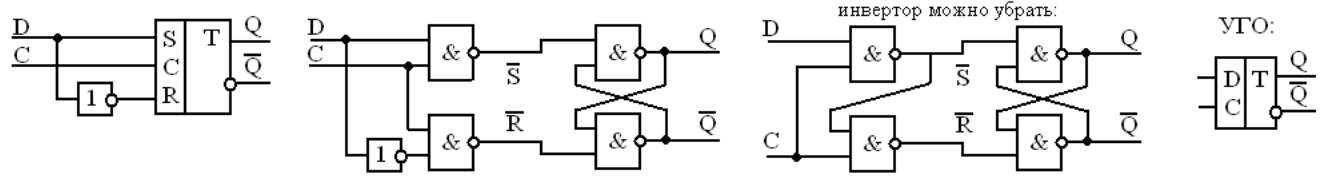
$$t_{\text{зг.р. II}} = 5 t_{\text{зг.р. лэ}}$$

$$T_{\text{мин}} = 7 t_{\text{зг.р. лэ}}$$

Просвещать

Синхронный D-триггер со статическим управлением (одноступенчатый)

При C=0 триггер находится в режиме хранения информации, при C=1 переключается как асинхронный триггер соответствующего типа.

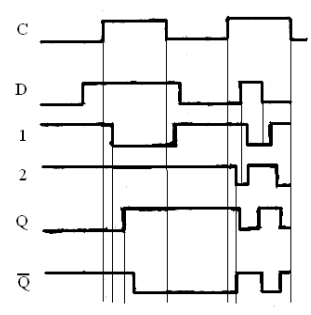


C	D	Q _n	Q _{n+1}	S*	R*
0	0	0	0	0	α
0	0	1	1	α	0
0	1	0	0	0	α
0	1	1	1	α	0
1	0	0	0	0	α
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	1	α	0

$$\bar{S} = \overline{CD}$$

$$\bar{R} = \overline{CD}$$

$$\bar{R} = \overline{CD + CC} = \overline{C(\bar{C} + D)} = \overline{C \cdot D\bar{C}} = \overline{C \cdot S}$$



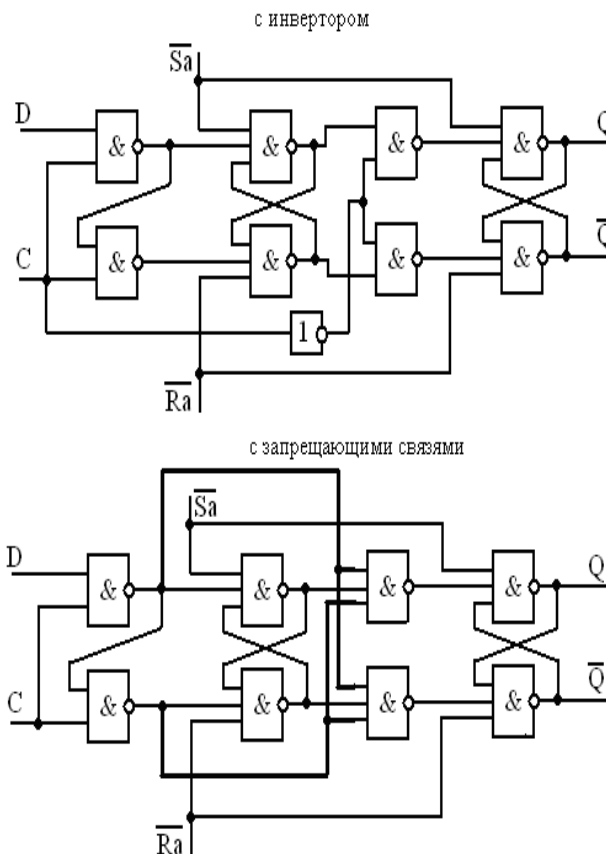
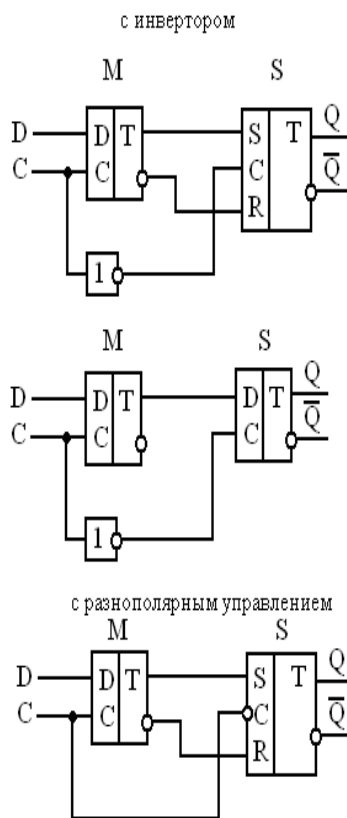
Синхронный D-триггер с двухступенчатым запоминанием информации

В двухступенчатых триггерах входная и выходная ступени тактируются «антисинхронно», прием информации разрешается в них поочередно. Вследствие этого отсутствует режим прозрачности триггера при любом уровне синхросигнала, что позволяет реализовать любые типы триггеров, свободные от режимов генерации, и дает возможность построения синхронных автоматов без опасных временных состязаний. Однако, их схемы более сложные, чем с динамическим входом, а быстродействие несколько ниже.

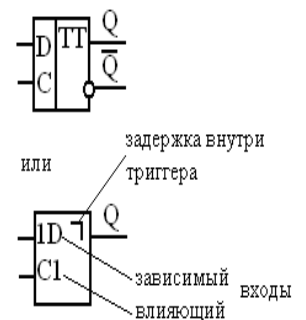
Двухступенчатые триггеры строятся несколькими способами: с инвертором, с разнополярным управлением ступенями, с запрещающими связями. Первая ступень – всегда D-триггер.

M – Master, S- Slave.

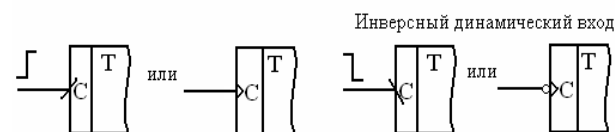
При C=0 перепись информации из первой ступени во вторую и перевод первой ступени в режим хранения. При C=1 разрешена запись в первую ступень и запрещена перепись из первой ступени во вторую.



УГО:



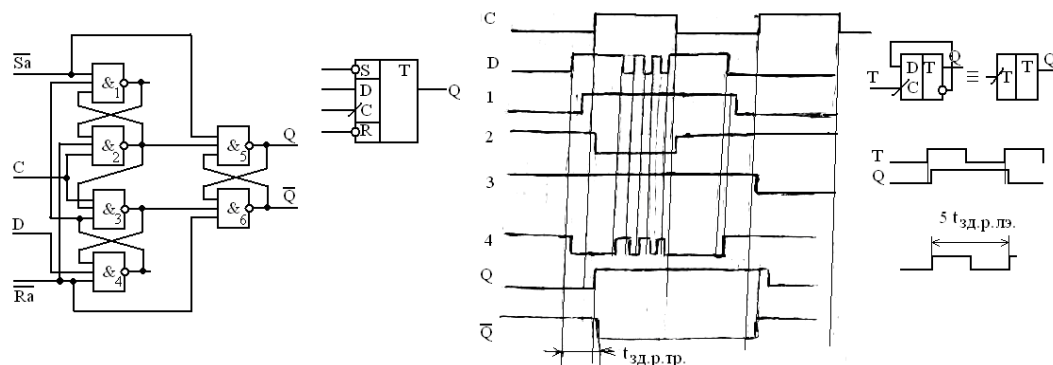
Синхронный D-триггер с динамическим управлением записью (одноступенчатый).



Строятся как по одноступенчатым, так и по двухступенчатым схемам.

Принцип работы – переключаются перепадом сигнала C. Чувствительность к изменению состояния сохраняется в течении $t_{зд.р.ЛЭ}$.

Время предустановки – время, в течение которого перед переключением, триггер должен быть предустановлен. Может выполнять функции D триггера и T триггера.



$$t_{зд.р.тр} = 3 \cdot t_{зд.р.ЛЭ}$$

$$f = \frac{1}{T_c} = \frac{1}{5 \cdot t_{зд.р.ЛЭ}}$$

работа в сч. режиме. Триггер на 3-х RS-триггерах. 5,6 – осн. триггер (ЗЯ).

D-триггер - TM2: K155TM2, K555TM2, KP1533TM2. В корпусе

МС расположено 2 триггера (14 выводов).

Применение триггеров:

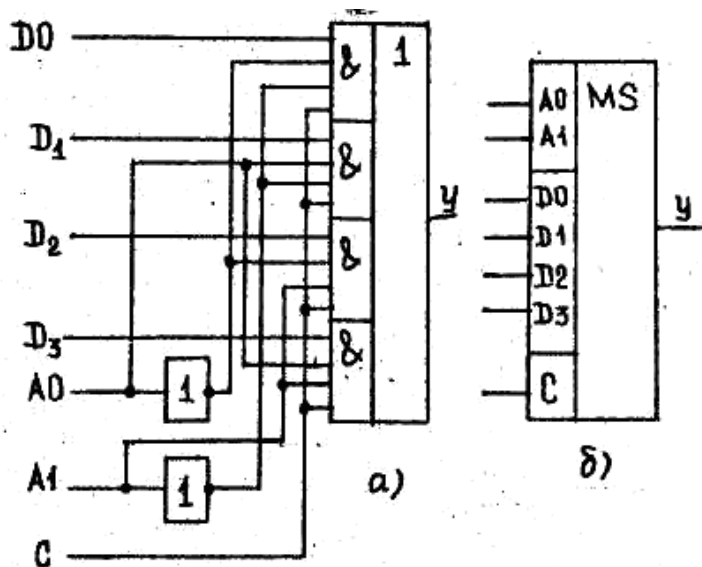
- 1) Одноступенчатые статические - в регистрах памяти, в регистрах сдвига, только если используется 2-х фазная система синхронизации (это значит что не 1 синхросигнал, а 2)
- 2) Одноступенчатые динамические - в регистрах сдвига с однофазной системой синхронизации, в регистрах памяти, в счётчиках
- 3) Двухступенчатые статические - во всех регистрах и счётчиках.

Мультиплексоры. Основные параметры. Выполняемые функции. Нарращивание.

Мультиплексор - это функциональный узел, комбинационного типа, имеющий n , адресных и $k = 2^n$ информационных входов и выполняющий коммутацию сигнала на выход с того информационного входа, адрес которого установлен на адресных входах.

Мультиплексор- адресный коммутатор.

Мультиплексор реализует логическую функцию $y = EN \vee_{j=0}^{2^n-1} D_j K_j(A_{n-1}, A_{n-2}, \dots, A_1, A_0)$, где A – адресные входы и сигналы, $j = 0, 1, \dots, 2^n-1$, K – конstituэнта A , т.е. конъюнкция всех аргументов, номер которой равен числу, образованному двоичным кодом сигналов на адресных входах.



Мультиплексор состоит из дешифратора, каждый конъюнктор которого имеет дополнительный вход для соответствующего информационного сигнала D . Выходы конъюнкторов объединяются - по операции ИЛИ дизъюнктом. Время задержки распространения сигнала в мультиплексоре определяется задержкой цепи последовательно соединенных элементов НЕ, И, ИЛИ.

Мультиплексоры широко применяются для построения: коммутаторов-селекторов цифровых сигналов; постоянных запоминающих устройств; комбинационных схем, реализующих логические функции; преобразователей кодов (например, параллельного кода в последовательный или кода с одними весами разрядов в код с другими весами разрядов) и других узлов.

Нарращивание мультиплексоров.

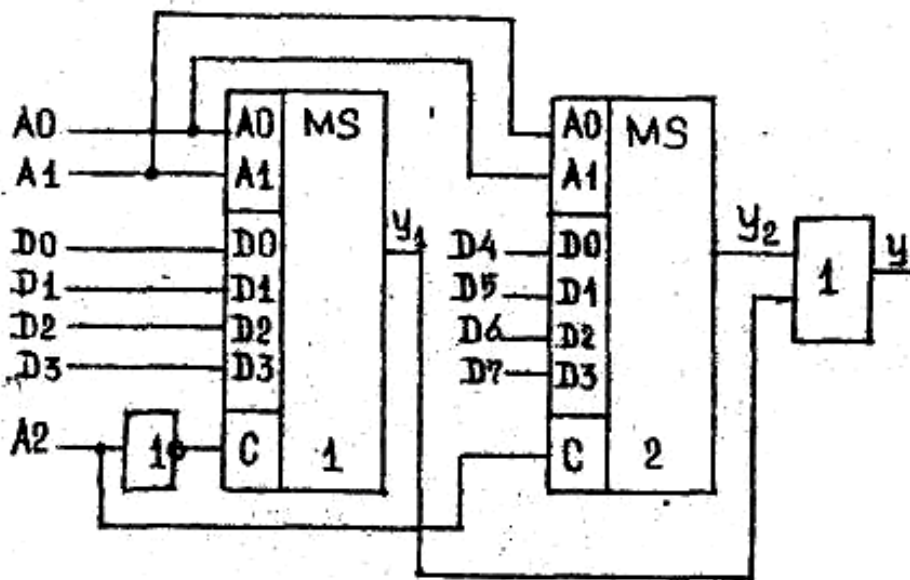
Существуют 2 схемы наращивания:
построенная только на **мультиплексорах** и
построенная на мультиплексорах и дешифраторах.

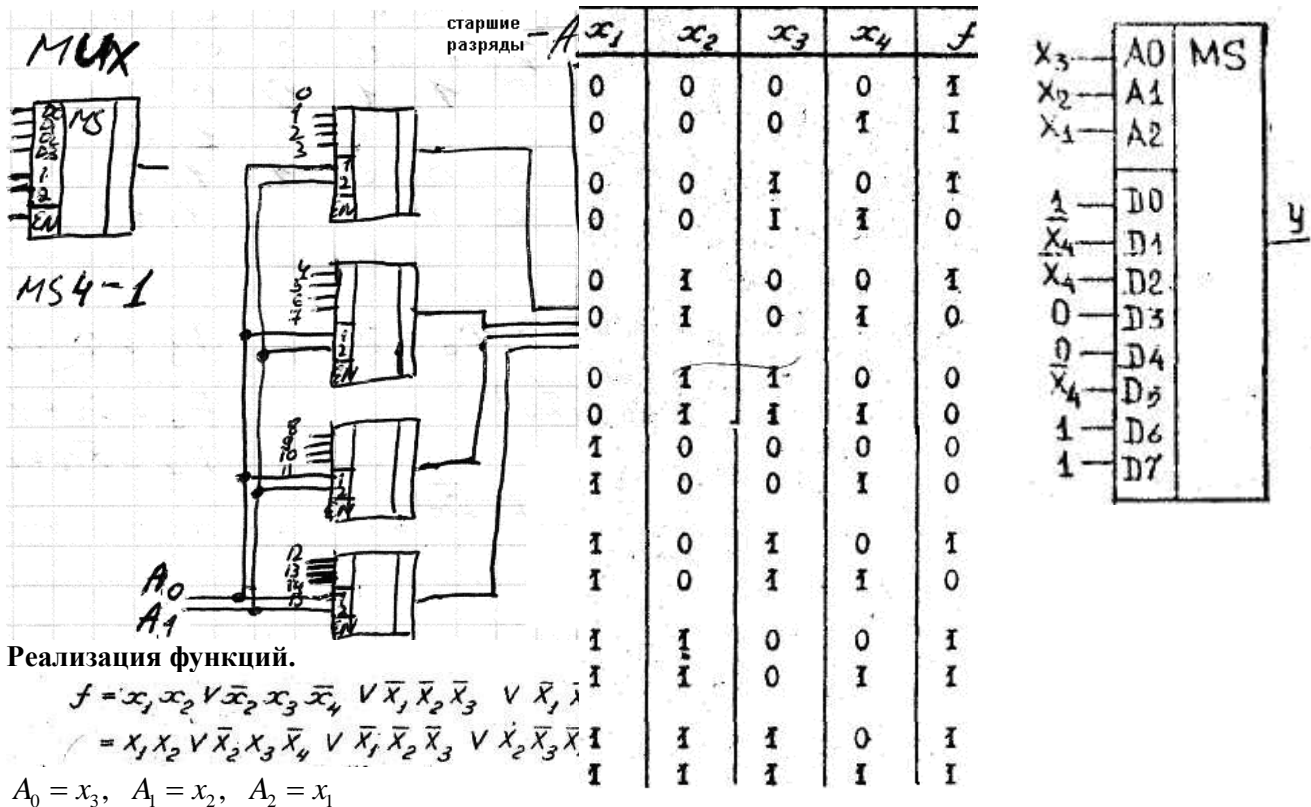
1. Определим количество мультиплексоров. $K = n/n'$, n и n' -количество разрядов.

Если мультиплексор сложный, то 1я ступень: младшие n' разрядов адресного кода, количество мультиплексоров: N/N' . Во 2й ступени следующая группа из n' разрядов адресного кода и тд., пока не шифруется в последней ступени последняя группа разрядов.

2. Информационные разряды разбиваются на группы. В каждой группе N разрядов.

Выходы сигнала 1го каскада подаются информационными сигналами на 2ю ступень.





на мультиплексорах и дешифраторах

Младшие из разрядов адресного кода подаются на адресные входы всех мультиплексоров. $N_{вх}/N_{-}$ количество мультиплексоров. Входы этих мультиплексоров объединены по «ИЛИ». Старшие разряды адреса декодируются дешифратором. Выходные сигналы дешифраторов управляют входами разрешения соответствующего мультиплексора.

$$f(x_4, x_3, x_2, x_1) = M_0 + M_1 + M_3 + M_8 + M_9 + M_{15}$$

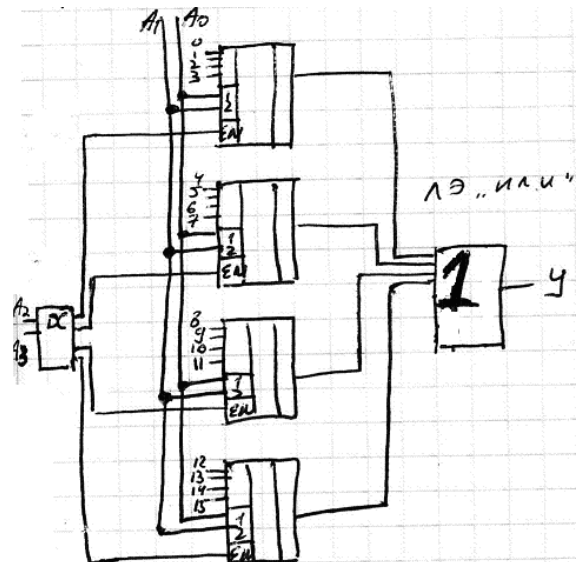
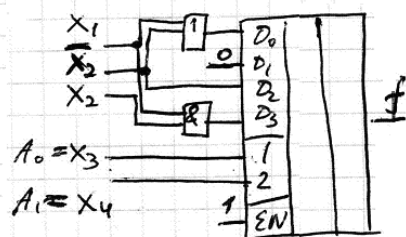
$x_4 x_3 x_2 x_1$	f
0 0 0 0	1
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1

$$D_0 = \bar{x}_2 \vee x_4$$

$$D_1 = 0$$

$$D_2 = \bar{x}_2$$

$$D_3 = x_2 x_1$$



Эта схема не универсальна. Есть другая реализация. Подаем на $D_0, D_1, D_3, D_8, D_9, D_{15}$ единицы, а на остальные — нули, тогда x_4, x_3, x_2, x_1 будут A_3, A_2, A_1, A_0 соответственно.

1я схема предпочтительнее, тк все элементы однотипны. Хотя 2я схема тоже используется.

Старшие разряды выбирают строку (по DC), а мультиплексоры — слово.

Регистры сдвига.

Регистры сдвига строятся, как правило, или на триггерах с динамическим управлением записью (одноступенчатые), или на триггерах с двухступенчатым запоминанием информации. Регистры сдвига состоят из элементов памяти и комбинационных схем (для передачи информации из разряда в разряд). Как правило, схема однородная, кроме первого и последнего разрядов.

Регистры сдвига обычно строятся на D-триггерах. Сдвиг информации осуществляется тактовыми сигналами, поступающими на входы С всех триггеров регистра.

Таблица переходов i-го разряда.

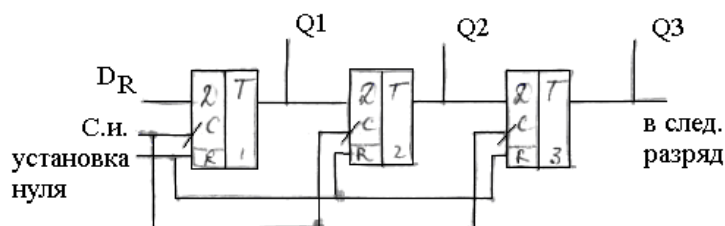
M	Q_{i-1}	Q_i	Q_{i+1}	Q_i^*
0	0	X	X	0
0	1	X	X	1
1	X	X	0	0
1	X	X	1	1

M (Mode) – управляющий сигнал, задает режим сдвига

M=0 – сдвиг вправо

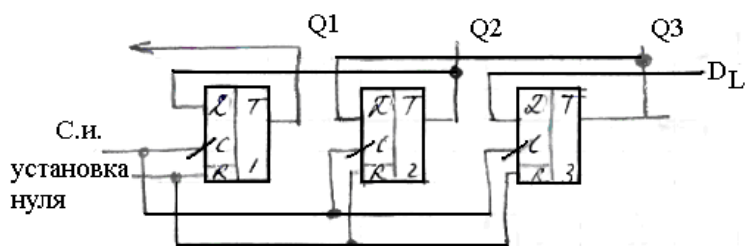
M=1 – сдвиг влево

$$D_i = Q_i^* = \bar{M} \cdot Q_{i-1} + M \cdot Q_{i+1} \Rightarrow \text{при } M = 0 \rightarrow D_i = Q_{i-1} - \text{вправо}$$



Синхроимпульсы (С.и.) поступают на входы С. Они обеспечивают сдвиг (в данном случае вправо).

$$D_1 = D_R \text{ (right)}$$



$$D_n = D_L \text{ (left)}$$

Синтез универсальных регистров сдвига

Основные микрооперации: 1) сдвиг влево, 2) сдвиг вправо, 3) загрузка начального слова, 4) хранение, 5) установка начального состояния.

Чтобы задать микрооперацию надо ввести управляющее слово. Это программируемый регистр M(mode). Два входа M_1 , M_0 – управляющие слова.

M1	M0	Dвх	Qi-1	Qi	Qi+1	Qi*
0	0	x	x	0	x	0
0	0	x	x	1	x	1
0	1	x		0	x	0
0	1	x		1	x	1
1	0	x	x	x		0
1	0	x	x	x		1
1	1	0	x	x	x	0
1	1	1	x	x	x	1

Все
сигналом

M1	M0	Микрооперация
0	0	Хранение
0	1	Сдвиг вправо
1	0	Сдвиг влево
1	1	Загрузка нач. слова

микрооперации
выполняются тактов.
С

Таблица переходов универсального регистра сдвига (i-й разряд)

M1, M0, Dвх	Qi-1, Qi, Qi+1	000	001	011	010	110	111	101	100
000							1		
001							1	1	1
011							1	1	1
010							1		
110							1		
111							1	1	1
101							1	1	1
100							1		

$$Q_i^* = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0 Q_{i-1} + M_1 \overline{M_0} Q_{i+1} + M_1 M_0 D_{iBX}$$

$$D_i = Q_i^*$$

$$(1) D_i = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0 Q_{i-1} + M_1 \overline{M_0} Q_{i+1} + M_1 M_0 D_{iBX}$$

$$(2) D_i = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0 D_R + M_1 \overline{M_0} Q_{i+1} + M_1 M_0 D_{iBX}$$

$$(3) D_i = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0 Q_{i-1} + M_1 \overline{M_0} D_L + M_1 M_0 D_{iBX}$$

(1), (2), (3) – функции демультиплексирования

Установка в 0 может быть выполнена асинхронно (через R).

Входы D_R и D_L используются для наращивания разрядности.

