

## Оглавление

1. Дешифраторы. Определение, назначение, основные параметры. Линейный дешифратор. Многоступенчатые дешифраторы. Пирамидальные дешифраторы. Временные диаграммы. Стробирование в дешифраторах. Применение стробирования. Дешифратор-демультиплексор. Нарращивание дешифраторов. Реализация ФАЛ на дешифраторах. ....	3
<i>Дешифратор</i> .....	3
<i>Применение</i> .....	4
<i>Параметры дешифратора</i> .....	4
<i>Линейный дешифратор</i> .....	4
<i>Пирамидальный дешифратор</i> .....	5
Многоступенчатые дешифраторы. ....	6
Стробирование в дешифраторах. ....	7
Нарращивание дешифраторов. ....	8
<i>Дешифратор-демультиплексор</i> . ....	9
Реализация ФАЛ на дешифраторах. ....	9
2. Шифраторы. Определение, назначение, основные параметры. Приоритетные шифраторы. Нарращивание шифраторов. ....	10
<i>Шифратор</i> .....	10
<i>Приоритетный шифратор</i> .....	10
Методика наращивания <по Жиркову>: .....	11
3. Преобразователи кодов. ....	12
3.1 прямого кода в обратный и дополнительный и наоборот .....	12
3.2 одного ДДК в другой ДДК .....	13
3.3 ДДК в ДК (математическая основа, элементарный преобразователь, многоразрядный преобразователь целых чисел) .....	14
3.4 ДК в ДДК (математическая основа, элементарный преобразователь, многоразрядный преобразователь целых чисел) .....	15
3.5 простого двоичного кода в код Грея и наоборот .....	16
4. Схемы равнозначности. Компараторы (схемы сравнения). Синтез. Нарращивание компараторов (последовательная и пирамидальная схемы) .....	17
5. Контроль по четности/нечетности, контроль по коду Хемминга. ....	22
Контроль по нечетности. ....	22
Схема контроля по коду Хэмминга .....	23
Модифицированный код Хэмминга .....	24
6. Схемы свертки. ....	25
7. Однофазная система синхронизации. Основные параметры и соотношения. ....	26
8. Двухфазная система синхронизации. Основные параметры и соотношения. ....	27
9. Система синхронизации. Основные параметры. Структура. ....	28
10. Регистры. Классификация. Параметры. Параллельные регистры. Регистры сдвига. Универсальные регистры. Синтез. ....	30
<i>Регистр</i> .....	30
<i>Регистры памяти (параллельные регистры)</i> . ....	30
<i>Регистры сдвига</i> . ....	31
Универсальные регистры. ....	32
11. Синхронные счетчики. Основные параметры. Методика синтеза. Синтез суммирующих, вычитающих и реверсивных счетчиков. Нарращивание разрядности счетчиков по последовательной и «быстрой» схемам. ....	34
Основные параметры счетчиков: .....	34
Методика синтеза счетчиков. ....	35
Нарращивание по последовательной схеме: .....	36

<i>Наращивание синхронных счетчиков по схеме высокоскоростного счета</i>	36
12. Серии ИМС ТТЛ(Ш), основные параметры, особенности применения.	37
13. Серии ИМС КМОП, основные параметры, особенности применения.	39
14. Типы триггеров (одноступенчатые, двухступенчатые, со статическим и динамическим управлением записью) и их применение в регистрах и счетчиках.	41
Синхронный триггер со статическим управлением записью (одноступенчатый) ...	41
Синхронный триггер с двухступенчатым запоминанием информации.	41
Синхронный триггер с динамическим управлением записью	41
<i>Применение триггеров</i>	42
15. Мультиплексоры. Определение, назначение, основные параметры. Реализация математических функций на мультиплексорах. Наращивание мультиплексоров.	43
<i>Мультиплексор</i>	43
Наращивание мультиплексоров.	43
Реализация функций.	44
16. Функциональные узлы и элементы управления знакоиндикаторами.	45
17. Преобразователи уровней.	46
18. Сумматоры. Классификация. Одноразрядный сумматор. Многоразрядные сумматоры с последовательным и параллельным переносами. Оценка быстродействия.	48
Полный одноразрядный двоичный сумматор	49
19. Инкременторы и декременторы.	52
20. Матричные множительные устройства. Множительные и множительно-суммирующие блоки. Наращивание умножителей	53



$$F_j = EN \cdot m_j = \overline{EN \vee M_j}, \quad j = (\overline{0, N-1}), \quad (3)$$

где  $m_j$  и  $M_j$  – конститутенты единицы и нуля соответственно,  $j$  – номер набора, на котором  $m_j$  равна единице, а  $M_j$  – нулю.

### **Применение:**

- 1) Широко применяются в запоминающих устройствах при преобразовании № ячейки памяти в управляющий сигнал для обращения к этой ячейки памяти
- 2) В арифметических устройствах для преобразования № команды в управляющие сигналы, необходимые для выполнения этих команд
- 3) Счет времени (в счетчиках и индикаторах)
- 4) Для расшифровки № устройства к которому обращаются и т.д.

Дешифратор может применяться:

- в качестве собственно дешифратора,
- как демультиплексор при наличии входа разрешения,
- для формирования функций алгебры логики при построении комбинационных узлов.

Входы разрешения служат:

- для временного выделения (стробирования) той части выходного сигнала, которая не имеет искажений, вызываемых гонками входных сигналов;
- для выполнения функции демультиплексирования;
- для наращивания разрядности дешифратора, т.е.увеличения числа адресных входов и соответственно выходов.

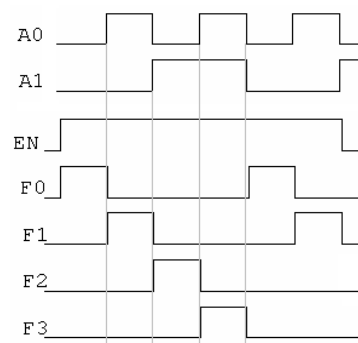
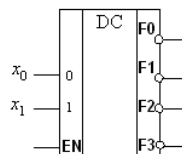
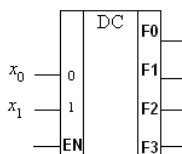
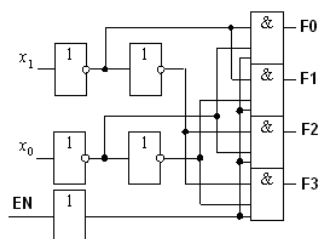
**Параметры дешифратора.** К основным параметрам дешифратора относятся:

- количество входов, определяемое разрядностью преобразуемого двоичного слова,
- количество выходов,
- статические параметры, такие как входные и выходные токи и напряжения логических “0” и “1”, напряжения допустимых статических помех  $U_{\text{пом}}^0, U_{\text{пом}}^1$ , коэффициент разветвления по выходу  $K_{\text{раз}}$  и др., которые определяются аналогичными параметрами логических элементов, на которых он строится,
- потребляемая мощность  $P_{\text{пот}}$  (или ток  $I_{\text{пот}}$ ),
- динамические параметры: времена задержки распространения сигнала при включении и выключении  $t_{\text{зд.р}}^{1,0}$  и  $t_{\text{зд.р}}^{0,1}$ , 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.

**Линейный дешифратор** строится в соответствии с системой функцией (1) или (2) и представляет собой  $2^n$  конъюкторов или ЛЭ ИЛИ-НЕ с  $n$ -входами при отсутствии стробирования и с  $(n+1)$  входами - при его наличии. Линейный дешифратор DC 2-4 работает в соответствии с таблицей истинности и реализует логические функции:

$$\begin{aligned} F_0 &= EN \cdot \bar{A}_1 \cdot \bar{A}_0 = \overline{EN \vee A_1 \vee A_0}, \quad F_1 = EN \cdot \bar{A}_1 \cdot A_0 = \overline{EN \vee A_1 \vee \bar{A}_0}, \\ F_2 &= EN \cdot A_1 \cdot \bar{A}_0 = \overline{EN \vee \bar{A}_1 \vee A_0}, \quad F_3 = EN \cdot A_1 \cdot A_0 = \overline{EN \vee \bar{A}_1 \vee \bar{A}_0}, \end{aligned} \quad (4)$$



$$t_{зд.p.DC} = 2t_{зд.p.HE} + t_{зд.p.ЛЭ}$$

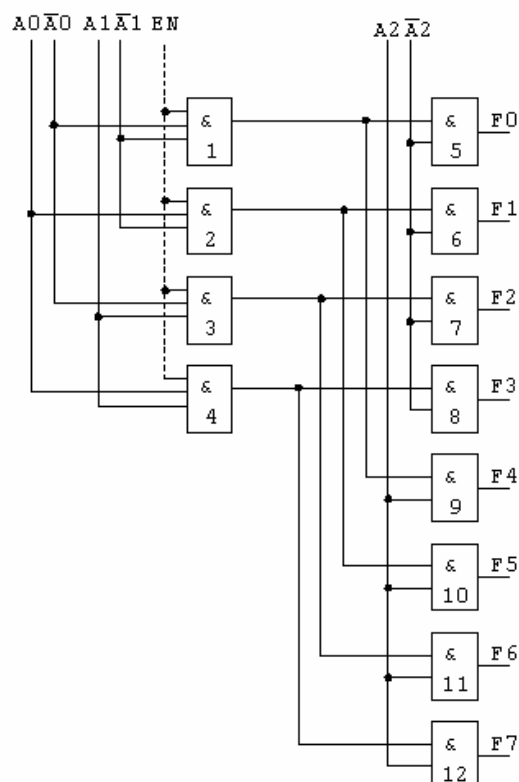
Недостаток: при увеличении числа входов увеличивается число входов конъюнктора.

**Пирамидальный дешифратор.** Недостаток – снижается быстродействие каскада, достоинство – требуются ЛЭ И, И-НЕ, НЕ с 2 входами. Строится на основе последовательной (каскадной) реализации выходных функций. Сначала реализуются функции двух переменных  $A_1$  и  $A_0$ :  $\bar{A}_1 \bar{A}_0; \bar{A}_1 A_0; A_1 \bar{A}_0; A_1 A_0$ . ЛЭ «И», формирующие эти конъюнкции, образуют DC 2-4.

На втором этапе реализуют все конъюнкции трех переменных путем логического умножения каждой ранее полученной конъюнкции двух переменных на переменную  $A_2$  и  $\bar{A}_2$ :  $\bar{A}_2 \bar{A}_1 \bar{A}_0, \bar{A}_2 \bar{A}_1 A_0, \bar{A}_2 A_1 \bar{A}_0, \bar{A}_2 A_1 A_0, A_2 \bar{A}_1 \bar{A}_0, A_2 \bar{A}_1 A_0, A_2 A_1 \bar{A}_0, A_2 A_1 A_0$ .

ЛЭ «И», формирующие конъюнкции, полученные на первом и втором этапах образуют DC3-8.

На следующем этапе каждую из полученных конъюнкций трех переменных умножают на  $A_3$  и  $\bar{A}_3$  и т.д. Таким образом на каждом следующем этапе получают вдвое больше конъюнкций, чем на предыдущем этапе. Как видно построение дешифратора начинается с двух входов, т.к. одного адресного сигнала является вырожденным и представляет собой повторитель и инвертор этого адресного сигнала.



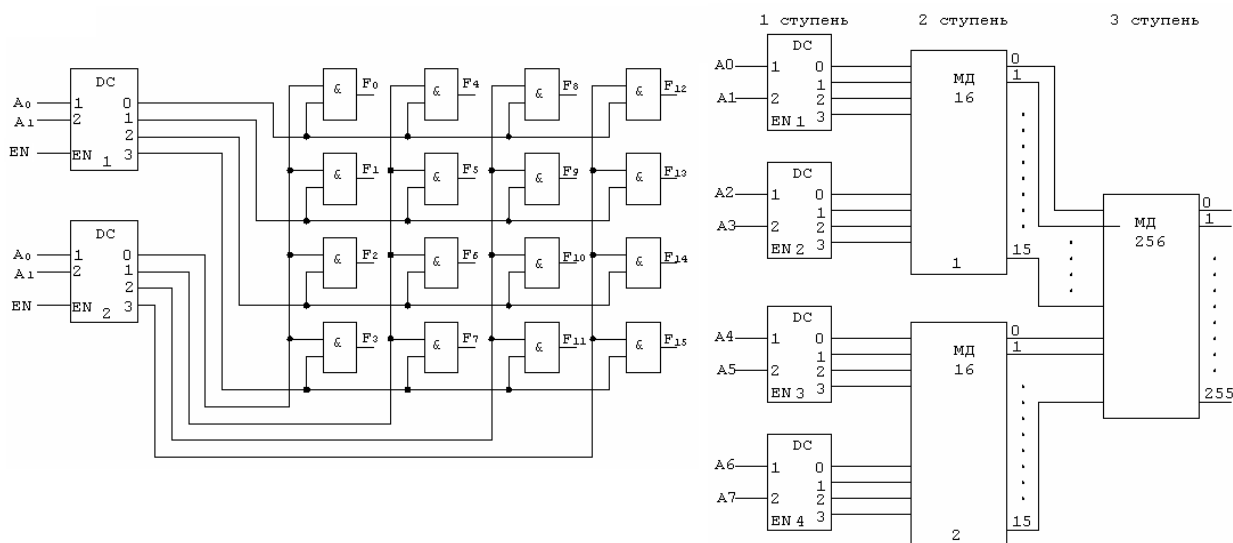
$$t_{зд.p.DC} = t_{зд.p.HE} + (n-1)t_{зд.p.ЛЭ} \text{ для DC n-N}$$

### Многоступенчатые дешифраторы.

Дешифратор, который реализуется по выражению  $f_i = ((\tilde{x}_3 \tilde{x}_2)(\tilde{x}_1 \tilde{x}_0))$ , называется

двухступенчатым. Разбиваем все конъюнкции попарно: по 2 переменные в группы.

Отдельно строим DC 2-4 для переменных двух групп и объединяем их выходы попарно во всевозможных комбинациях.  $t_{зд.р. DC} = t_{зд.р. DC2-4} + t_{зд.р. ЛЭ}$



Первая ступень дешифратора содержит четыре дешифратора 2-4, вторая ступень — два матричных дешифратора (МД 16), представляющих 16 ЛЭ «И» каждый, третья ступень —

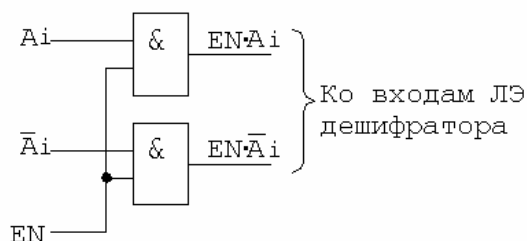
матричный дешифратор (МД 256), содержащий матрицу из 256 ЛЭ «И».  $t_{зд.р. DC} = \sum_{i=1}^S t_{зд.р. i}$ ,

где  $t_i$  — время задержки в  $i$ -ой ступени,  $s$  — число ступеней.

### Стробирование в дешифраторах.

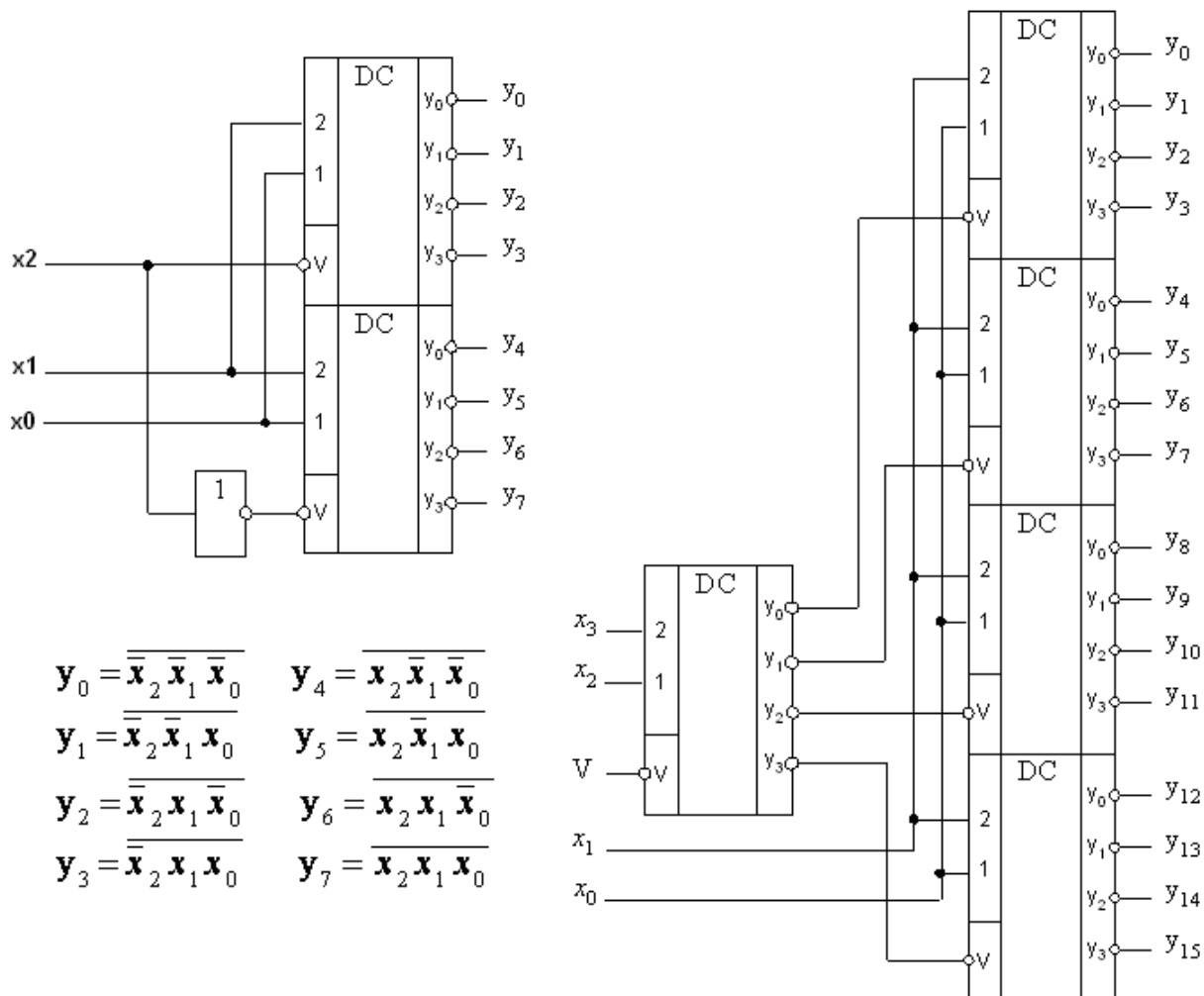
Сигнал EN воздействует непосредственно на все дешифрирующие логические элементы и осуществляет стробирование по выходу. Этот вариант реализации стробирования требует увеличения на единицу числа входов дешифрирующих элементов.

Стробирование можно реализовать другим способом, а именно по входу путем разрешения всех дешифрирующих элементов прямым и инверсным значениями одного из адресных сигналов. Стробирование по входу экономично по оборудованию, но несколько увеличивает задержку дешифратора.

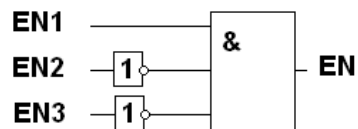
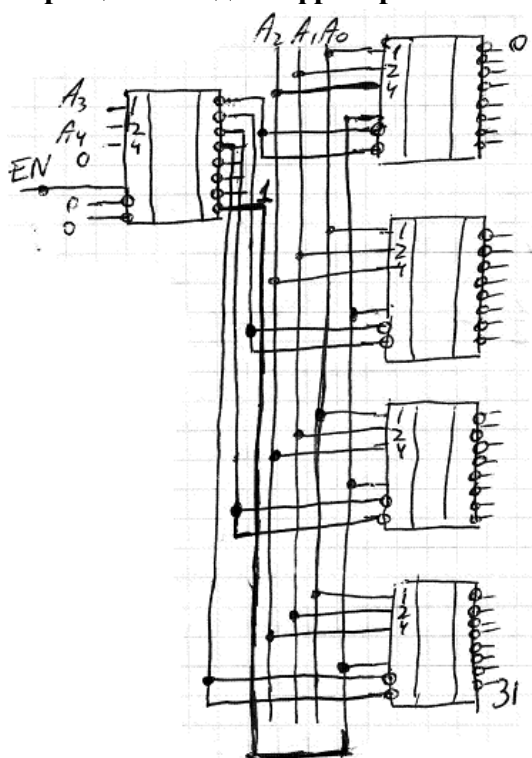


### Применение стробирования.

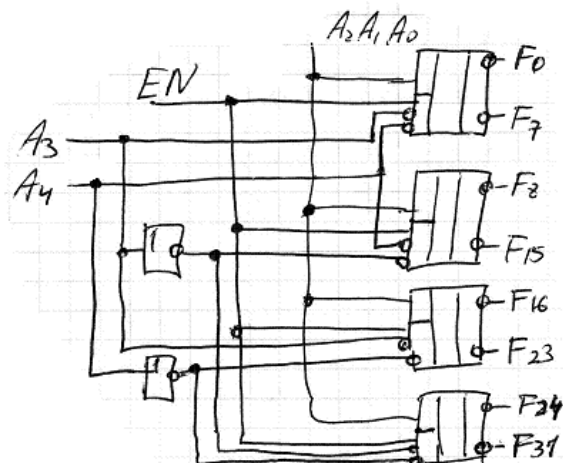
В DC 3-8 переменная  $x_2$  использована в качестве стробирующего входа, в результате дешифратор построен из двух DC 2-4. Нетрудно видеть, что так же легко реализуется дешифратор DC 4-16 на пяти стробируемых дешифраторах DC 2-4, причем один из них будет дешифрировать наборы, представленные двумя старшими переменными, и стробировать остальные четыре. Вход  $v$  этого дешифратора будет являться стробирующим входом для всего дешифратора DC 4-16.



# Наращивание дешифраторов.

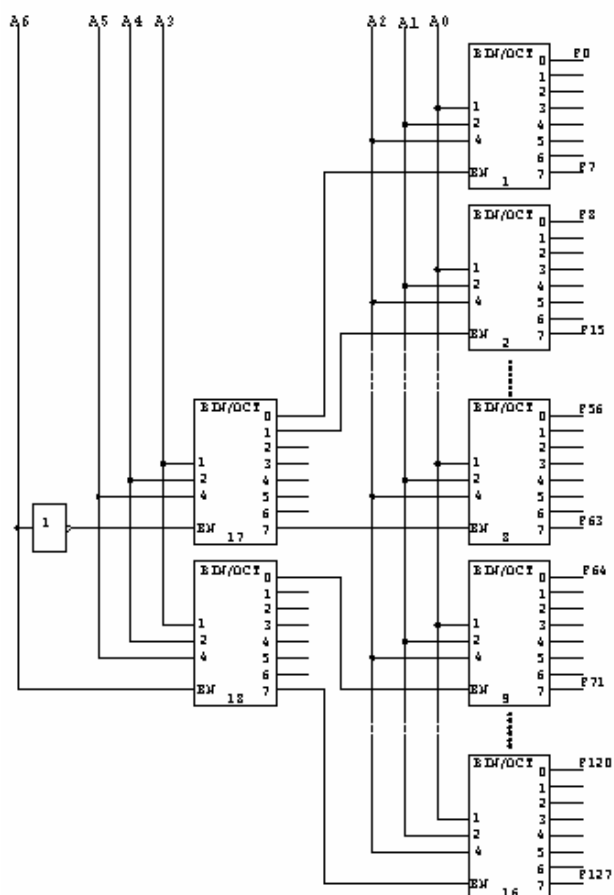


Убирается 1й дешифратор и добавляется 2 инвертора



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

$$P_K = \frac{N}{N1} = \frac{128}{8} = 16, \quad P_{K-1} = \frac{P_K}{N1} = \frac{16}{8} = 2, \quad P_{K-2} = \frac{P_{K-1}}{N1} = \frac{2}{8} - \text{значит } 1$$





**Дешифратор-демультиплексор.** Демультиплексор- функциональный узел, имеющий информационный вход, адресные входы и выходы, и переключающий сигнал, поданный на информационный вход, на тот выход, номер которого задается набором входных адресных сигналов. Дешифратор, имеющий вход разрешения, может выполнять функцию демультиплексора, если на этот вход подавать информационный сигнал  $D$ , а на адресные входы - коды адресов выходов. Если демультиплексор имеет два входа разрешения  $EN_1$  и  $EN_2$ , объединенные конъюнктивно, то на один из них подается информационный сигнал  $D$ , а на другой сигнал разрешения, т.е.:  

$$F_j = EN_1 \cdot EN_2 \cdot m_j = EN_1 \cdot D \cdot m_j, j = \overline{0, 2^n - 1}$$

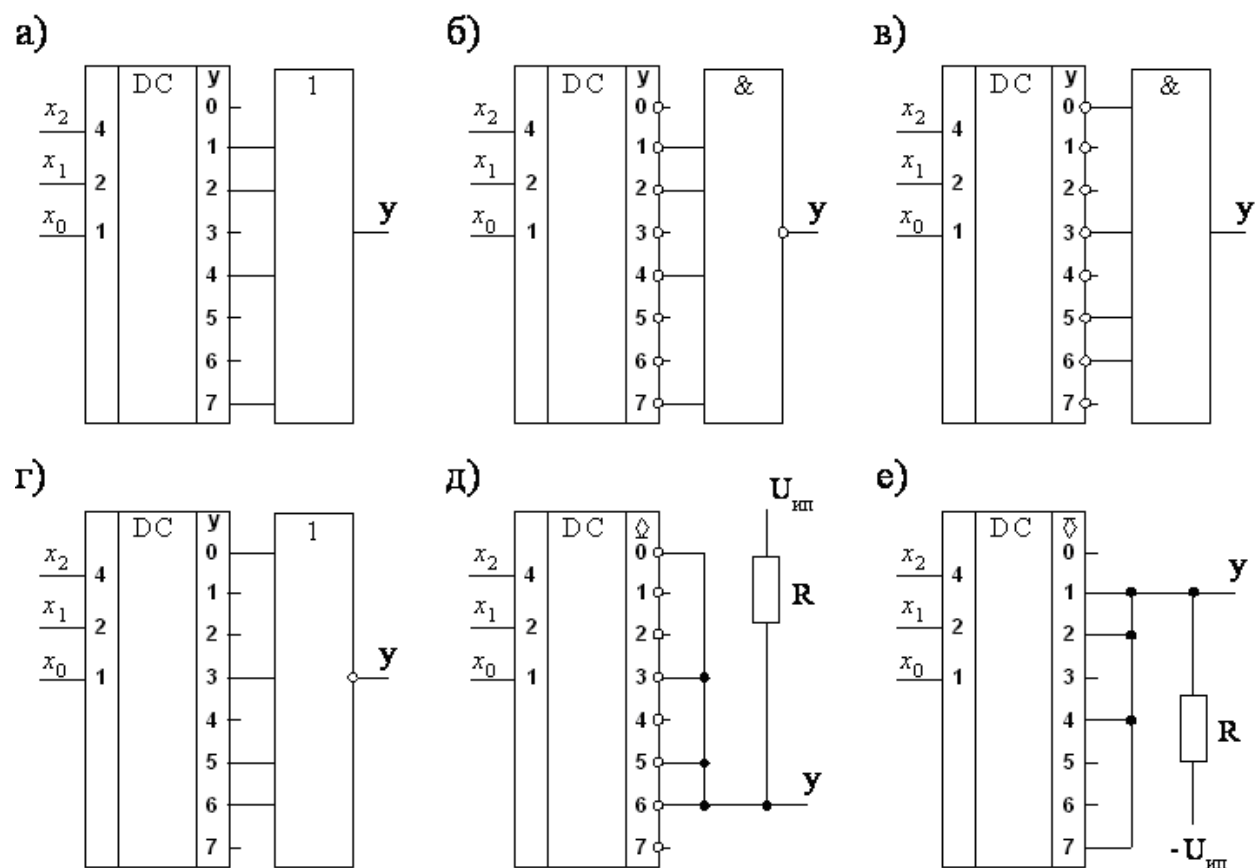
Демультиплексоры применяются для распределения данных, передаваемых по мультиплексированной магистрали, в последовательные моменты времени, по своим приемникам. Демультиплексор может применяться для преобразования последовательного кода числа, подаваемого на вход  $EN$ , в параллельный. При этом номер разряда преобразуемого числа поступает в демультиплексор в качестве адреса, например, со счетчика, а выход должен фиксироваться в регистре.

### Реализация ФАЛ на дешифраторах.

СДНФ:  $y = 1 + 2 + 4 + 7$  (а),  $y = \overline{1 \cdot 2 \cdot 4 \cdot 7}$  (б)

СКНФ:  $y = 0 \cdot 3 \cdot 5 \cdot 6$  (в),  $y = \overline{0 + 3 + 5 + 6}$  (г)

(д) – ДС с ОК (монтажная операции «И»), (е) – ДС с ОЭ (монтажная операции «ИЛИ»)



## 2. Шифраторы. Определение, назначение, основные параметры. Приоритетные шифраторы. Нарращивание шифраторов.

**Шифратор** - логическое устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код. 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.



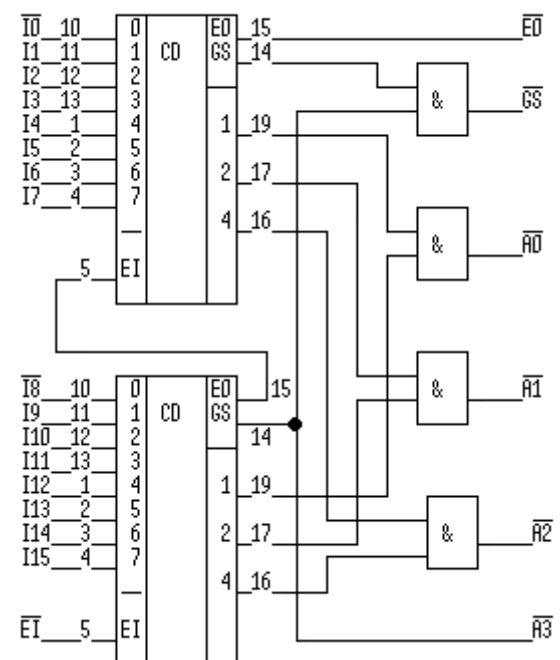
EI – разрешение входа. EO – разрешение выхода. GS – групповой сигнал.

**Приоритетный шифратор** отличается от шифратора наличием дополнительной логической схемы выделения активного уровня 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.

Особенностью шифраторов K555IB1 (IB3) являются инверсные входы и выходы. K555IB1 (155IB1, 533IB1) – преобразовывает унарный код в трехразрядный двоичный с формированием групповых сигналов для наращивания разрядности. EI – вход разрешения. K555IB3 – преобразовывает унарный код в двоично-десятичный.

Нарращивание разрядности шифраторов основано на том обстоятельстве, что все цифры, начиная с восьми, имеют "1" в старшем разряде. Следовательно, выход GS шифратора, образующего входы I8 - I15, может быть использован для образования старшего разряда A3.

Схема 4-х разрядного шифратора на микросхемах K555IB1.





### 3. Преобразователи кодов.

#### 3.1 прямого кода в обратный и дополнительный и наоборот

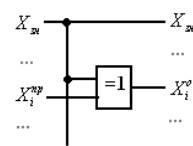
Обратный и дополнительный коды применяются для замены операции вычитания сложением в этих кодах. Следует различать инвертирование и преобразование прямого кода в обратный. Инвертирование – операция дополнения, выполняется над всеми разрядами, в т.ч. и знаковыми. В двоичной системе счисления инвертирование совпадает с преобразованием пр. кода в обр. Код Айкена, код с избытком 3 – самодополняющиеся коды.

**Преобразование пр. кода в обр.** – поразрядная операция. Знак не меняется.

$$[X]_{Pr} = X_{zn} X_n X_{n-1} \dots X_1$$

$X_{zn}$	$X_i^{np}$	$X_i^o$
0	0	0
0	1	1
1	0	1
1	1	0

$$X_i^o = \overline{X_{zn}} X_i^{np} + X_{zn} \overline{X_i^{np}} = X_{zn} \oplus X_i^{np}$$



Два изображения нуля:

$$[+0]^{np} = 0.00\dots0 \quad [+0]^o = 0.00\dots0$$

$$[-0]^{np} = 1.00\dots0 \quad [-0]^o = 1.11\dots1$$

Обратное преобразование из обратного кода в прямой совершается точно по такой же формуле и схеме.

**Преобразование пр. кода в доп.** – поразрядная операция (это недостаток).

В доп. коде ноль имеет одно изображение.

$X_{zn}$	$X_2^{np}$	$X_1^{np}$	$X_0^{np}$	$Y_{zn}$	$X_2^o$	$X_1^o$	$X_0^o$
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
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	0
1	0	1	1	1	1	0	1
1	1	0	0	1	1	0	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	1	0
1	1	1	1	1	0	0	1

В таблице выделены изображения нуля, т.к. одно изображение нуля: 0000 инвертируется, 1111 +1, получается 0000.

$X_1, X_0$	$X_{zn}, X_2$	00	01	11	10	$Y_{zn}$
00				1		
01				1	1	
11				1	1	
10				1	1	

$X_1, X_0$	$X_{zn}, X_2$	00	01	11	10	$X_{1d}$
00						
01				1	1	
11				1	1	
10				1	1	

$X_1, X_0$	$X_{zn}, X_2$	00	01	11	10	$X_{2d}$
00			1	1		
01			1		1	
11			1		1	
10			1		1	

$X_1, X_0$	$X_{zn}, X_2$	00	01	11	10	$X_{0d}$
00						
01			1	1	1	1
11			1	1	1	1
10						

$$X_0^o = X_0^{np}$$

$$X_1^o = \overline{X_{zn}} X_1 + X_{zn} (X_1 \oplus X_0)$$

$$X_2^o = \overline{X_{zn}} X_2 + X_{zn} (X_2 \oplus (X_1 + X_0))$$

$$Y_{zn} = X_{zn} (X_2 + X_1 + X_0)$$

Для n-разрядного

$$X_i^o = \overline{X_{zn}} X_i + X_{zn} (X_i \oplus (X_{i-1} + \dots + X_0))$$

Само преобразование пр. – доп. код сложное, удобнее в машинах использовать обр. код. Обратный код более приспособлен для чисел в последовательном виде.

**Преобр. пр. кода в доп. в устройствах последовательного действия**



$[X]^{np} = 1.01101100$ ,  $[X]^o = 1.10010100$ . Пропускаем все нули до первой единицы и эту единицу. Используя ее инвертируем все остальные разряды. (см. рис. Послед. преобр.)

### 3.2 одного ДДК в другой ДДК

	ДДК 8421				ДДК 8421 +3				ДДК Айкена			
	X4	X3	X2	X1	Y4	Y3	Y2	Y1	Z4	Z3	Z2	Z1
0	0	0	0	0	0	0	1	1	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	0	0	1	0	1	0	0	1	0
3	0	0	1	1	0	1	1	0	0	0	1	1
4	0	1	0	0	0	1	1	1	0	1	0	0
5	0	1	0	1	1	0	0	0	1	0	1	1
6	0	1	1	0	1	0	0	1	1	1	0	0
7	0	1	1	1	1	0	1	0	1	1	0	1
8	1	0	0	0	1	0	1	1	1	1	1	0
9	1	0	0	1	1	1	0	0	1	1	1	1

X4, X3					
X2, X1	00	01	11	10	Z1
00			x		
01	1	1	x	1	
11	1	1	x	x	
10			x	x	

X4, X3					
X2, X1	00	01	11	10	Z3
00			1	1	
01			x	1	
11			1	x	
10			1	x	

X4, X3					
X2, X1	00	01	11	10	Z2
00			x	1	
01		1	x	1	
11	1		x	x	
10	1		x	x	

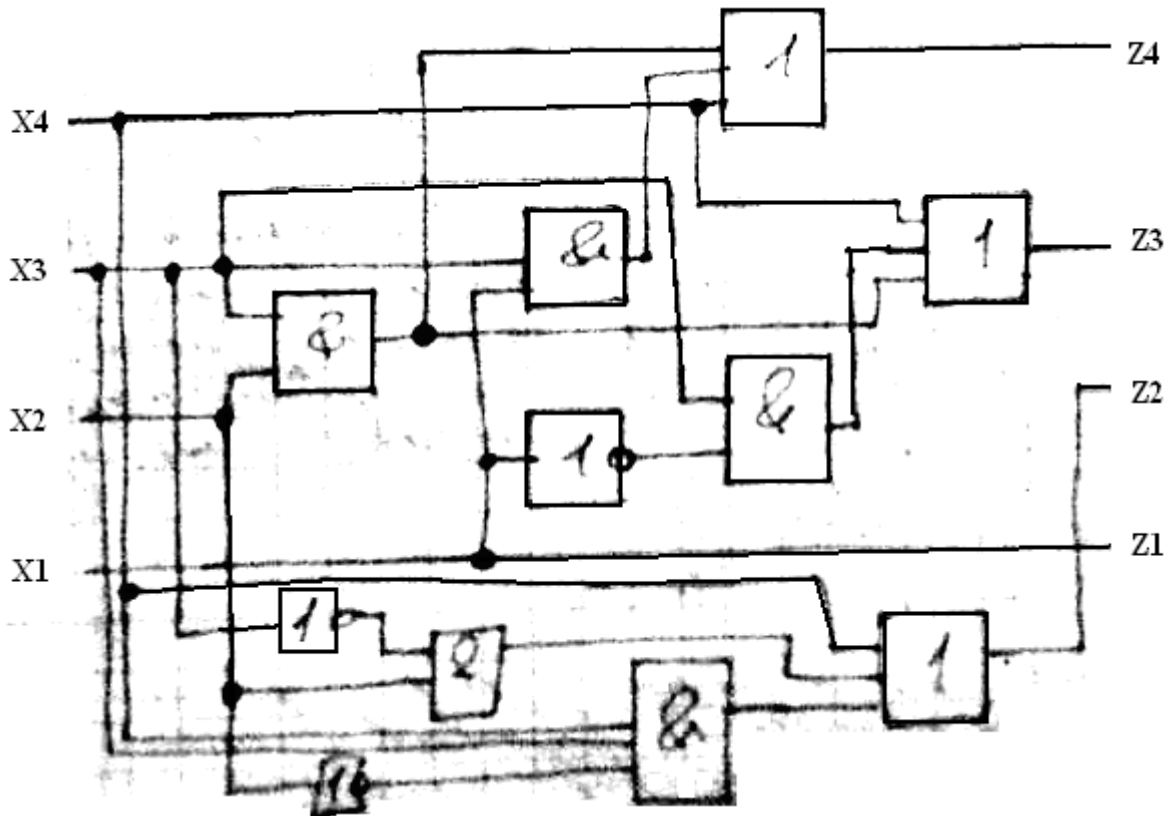
X4, X3					
X2, X1	00	01	11	10	Z4
00			x	1	
01		1	x	1	
11		1	x	x	
10		1	x	x	

$$Z_1 = X_1$$

$$Z_2 = X_4 + X_2 \overline{X_3} + X_3 \overline{X_2} X_1$$

$$Z_3 = X_4 + X_2 X_3 + X_3 \overline{X_1}$$

$$Z_4 = X_4 + X_1 X_3 + X_2 X_3$$



Аналогично для кода с избытком 3.

$$Y_i = f_i(X_4, X_3, X_2, X_1), i = \overline{1, 4}.$$

Строятся карты Карно, определяются функции возбуждения, составляется схема.

### 3.3 ДДК в ДК (математическая основа, элементарный преобразователь, многоразрядный преобразователь целых чисел)

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

Деление на основание системы счисления можно осуществлять сдвигом числа вправо на разряд. Но т.к. ДДК, а не ДК, то есть особенности.

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  
0110  
0111  
1101  
1110  
1111

В интегральном исполнении нет преобраз-лей с 4 входами и 4 выходами, есть K155IP6.

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

Реализация такого преобразователя на основе ПЗУ.

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

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

K155IP6 м.б. использован для преобразования правильной дроби из двоичной системы в двоично десятичную.

### 3.4 ДК в ДДК (математическая основа, элементарный преобразователь, многоразрядный преобразователь целых чисел)

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

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-десятичную сетку, начиная со своих старших разрядов. А введение двоичного числа происходит со стороны младших разрядов двоично-десятичного числа.

Пример:  $54_{(10)} = 110110_{(2)}$

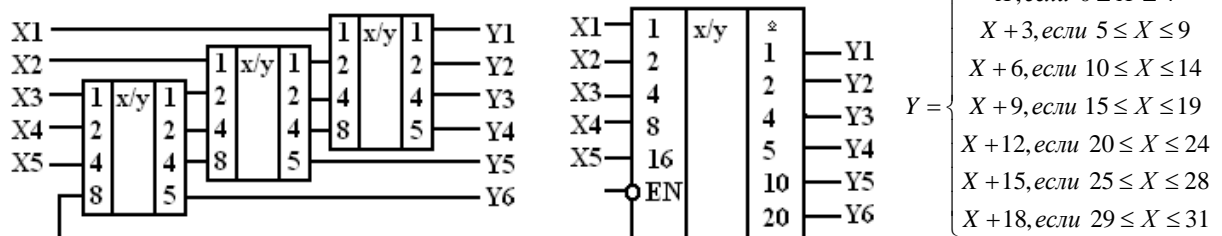
	x10^1					x10^0				
	8	4	2	1		8	4	2	1	
	0	0	0	0		0	0	0	1	100110-
<- u+1	0	0	0	0		0	0	1	1	00110--
<- u+0	0	0	0	0		0	1	1	0	110---
<- u+1	0	0	0	0		1	1	0	1	10----
						13>10 - псевдотетрада				
коррекция +6	0	0	0	1		0	0	1	1	0-----
<- u+1	0	0	1	0		1	1	1	0	
						псевдотетрада				
коррекция +6	0	1	0	1		0	1	0	0	

«1» должна приобретать вес 16, а приобретает вес 10. Коррекция +6 к той тетраде, в которой появляется единица в старшем разряде, и тетраде, величина которых лежит от 5 до 7: 0101, 0110, 0111 – при их сдвиге тоже появляются псевдотетрады.

Аналитическое выражения для преобразования:

$$Y = \begin{cases} X, & \text{если } 0 \leq X \leq 4 \\ X + 3, & \text{если } 5 \leq X \leq 9 \end{cases} + 3, \text{ а не } +6, \text{ т.к. коррекция на предыдущем шаге.}$$

Преобразователь К155ПР7:



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.

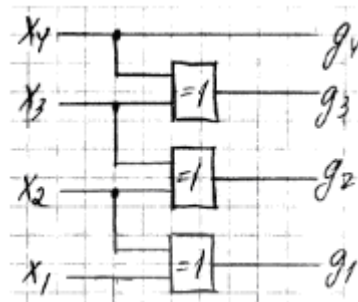
### 3.5 простого двоичного кода в код Грея и наоборот

Дес. число	Простой двоичный код				Код Грея			
	x4	x3	x2	x1	g4	g3	g2	g1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Прямое преобразование:

$$g_4 = x_4, \quad g_3 = x_4 \oplus x_3, \quad g_2 = x_3 \oplus x_2, \quad g_1 = x_1 \oplus x_2$$

x4,x3 x2,x1	00				01				11				10			
	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00		1	1	1	1											
01		1	1	1	1		1	1	1	1						
11		1	1	1	1		1		1							
10		1	1	1	1		1	1	1	1						

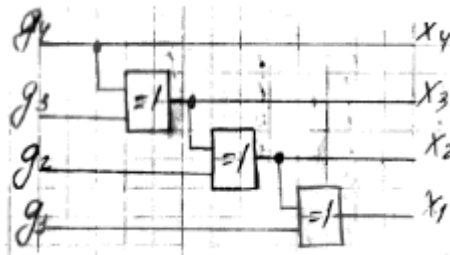


$$g_i = x_{i+1} \oplus x_i, \quad i \neq n, \quad g_n = x_n, \quad t_{\text{зд.п.пр.}} = t_{\text{зд.п.ЛЭ}}$$

Обратное преобразование:

$$x_4 = g_4, \quad x_3 = g_4 \oplus g_3, \quad x_2 = g_4 \oplus g_3 \oplus g_2, \quad x_1 = g_4 \oplus g_3 \oplus g_2 \oplus g_1$$

x4,x3 x2,x1	00				01				11				10			
	00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
00		1	1	1	1											
01		1	1	1	1		1	1								
11		1	1	1	1		1		1							
10		1	1	1	1		1	1	1							



$$x_i = \sum_{j=i}^{n-1} g_j, \quad i \neq n, \quad x_n = g_n, \quad t_{\text{зд.п.пр.}} = (n-1)t_{\text{зд.п.ЛЭ}}$$

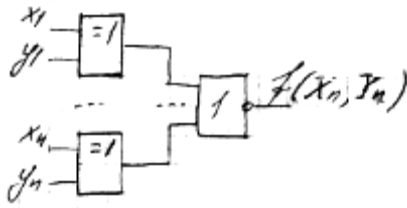


#### 4. Схемы равнозначности. Компараторы (схемы сравнения). Синтез. Нарращивание компараторов (последовательная и пирамидальная схемы)

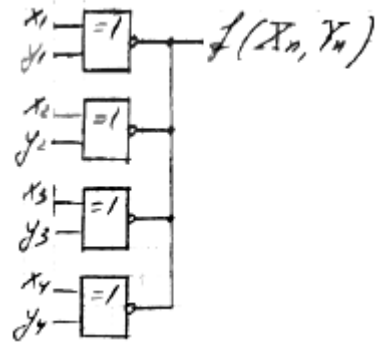
$$X_n = x_n \dots x_p \dots x_1, \quad Y_n = y_n \dots y_p \dots y_1$$

$$\text{Функция равнозначности слов: } f(X_n, Y_n) = \begin{cases} 1, & \text{если } x_p = y_p, p = \overline{1, n} \\ 0 & \text{иначе} \end{cases}$$

$$1) f(X_n, Y_n) = \overline{\bigvee_{p=1}^n (x_p \oplus y_p)}$$



$$2) f(X_n, Y_n) = \overline{\bigwedge_{p=1}^n x_p \oplus y_p}$$



Если использовать элементы с открытым коллектором:

**Схемы сравнения чисел.** 5 ф-ий:  $>, \geq, =, \leq, <$ .  $f(X_n \text{ условие } Y_n) = \begin{cases} 1, \text{ если условие вып-ся} \\ 0 \text{ иначе} \end{cases}$

Но достаточно 2 функций, т.к. остальные можно выразить через них.

Возьмем следующие функции:

$$f_n = F(X_n, Y_n) = F(X_n \leq Y_n) = \begin{cases} 0, X_n > Y_n \\ 1, X_n \leq Y_n \end{cases}, \quad \varphi_n = F(X_n, Y_n) = F(X_n = Y_n) = \begin{cases} 0, X_n \neq Y_n \\ 1, X_n = Y_n \end{cases}$$

Мы будем анализировать числа с младших разрядов, т.к. это допускает простое наращивание.

$$1) X_1 = x_1, Y_1 = y_1$$

$X_1$	$Y_1$	$f_1$	$\varphi_1$
0	0	1	1
0	1	1	0
1	0	0	0
1	1	1	1

$$f_1 = \overline{x_1 y_1} + \overline{x_1 \oplus y_1}$$

$$\varphi_1 = x_1 \oplus y_1 \quad (1)$$

$$2) X_2 = x_2 x_1, Y_2 = y_2 y_1$$

$f_1$	$\varphi_1$	$x_2$	$y_2$	$f_2$	$\varphi_2$		$f_1$	$\varphi_1$	$x_2$	$y_2$	$f_2$	$\varphi_2$	
0	0	0	0	0	0	$x_1 \neq y_1$	1	0	0	0	1	0	$x_1 \neq y_1$
0	0	0	1	1	0		1	0	0	1	1	0	
0	0	1	0	0	0		1	0	1	0	0	0	
0	0	1	1	0	0		1	0	1	1	1	0	
0	1	0	0	x	x	Несовм	1	1	0	0	1	1	$x_1 = y_1$
0	1	0	1	x	x		1	1	0	1	1	0	
0	1	1	0	x	x		1	1	1	0	0	0	
0	1	1	1	x	x		1	1	1	1	1	1	

		$f_2$				$\varphi_2$			
$f_1, \varphi_1$	$x_2, y_2$	00	01	11	10	00	01	11	10
		00		*	1	1		*	1
	01		1	*	1	1		*	
	11			*	1	1		*	1
	10			*				*	

$$\begin{aligned} f_2 &= \overline{x_2 y_2} + \overline{f_1 x_2 y_2} + \overline{f_1 x_2 y_2} = \varphi_2 = \overline{\varphi_1 (x_2 y_2 + x_2 y_2)} = \\ &= \overline{x_2 y_2} + \overline{f_1 \cdot x_2 \oplus y_2} = \overline{\varphi_1 \cdot x_2 \oplus y_2} \end{aligned} \quad (2)$$

$$3) f_3 = \overline{x_3 y_3} + \overline{f_2 \cdot x_3 \oplus y_3}, \quad \varphi_3 = \overline{\varphi_2 \cdot x_3 \oplus y_3} \quad (3)$$

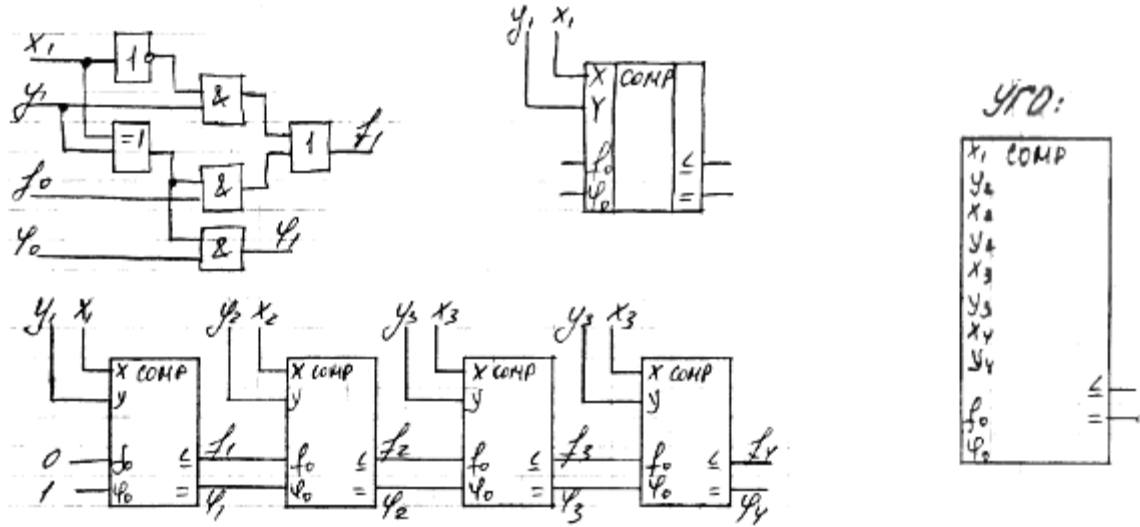
$$4) f_4 = \overline{x_4 y_4} + \overline{f_3 \cdot x_4 \oplus y_4}, \quad \varphi_4 = \overline{\varphi_3 \cdot x_4 \oplus y_4} \quad (4)$$

$$n) f_n = \overline{x_n y_n} + \overline{f_{n-1} \cdot x_n \oplus y_n}, \quad \varphi_n = \overline{\varphi_{n-1} \cdot x_n \oplus y_n} \quad (5)$$

$$f_1 = \overline{x_1 y_1} + \overline{f_0 \cdot x_1 \oplus y_1}, \quad \varphi_1 = \overline{\varphi_0 \cdot x_1 \oplus y_1} \quad (6)$$

$f_0 = 0$  и  $\varphi_0 = 1$  - начальные условия для формирования функций сравнения.

Используя (5) получаем последовательное соединение. В интегральной схемотехнике используются схемы 4-разрядных чисел.



Недостаток последовательного включения: общая задержка распространения равна сумме задержек всех схем сравнения.

Если подставить в (4) формулы (3), (2), (1):

$$\begin{aligned} f_4 &= \overline{x_4 y_4} + \overline{x_3 y_3 \cdot x_4 \oplus y_4 + x_2 y_2 \cdot x_3 \oplus y_3 \cdot x_4 \oplus y_4 +} \\ &+ \overline{x_1 y_1 \cdot x_2 \oplus y_2 \cdot x_3 \oplus y_3 \cdot x_4 \oplus y_4 + f_0 \cdot \bigwedge_{p=1}^4 x_p \oplus y_p} \\ \varphi_4 &= \overline{\varphi_0 \cdot \bigwedge_{p=1}^4 x_p \oplus y_p} \end{aligned} \quad (7)$$

При реализации в соответствии с (7) время задержки распространения уменьшается.

### К561ИП2:

$$g_4 = E(f_4 + \varphi_4) = E\left[F(X_4 < Y_4) + \overline{f_0} \overline{\varphi_0} \cdot F(X_4 = Y_4)\right], \quad E - \text{разрешение работы.}$$

Доказательство:

$$\begin{aligned}
\overline{f_4} \vee \overline{\varphi_4} &= \overline{F(X_4 < Y_4) \vee f_0 F(X_4 = Y_4) \vee \varphi_0 F(X_4 = Y_4)} = \\
&= \overline{F(X_4 < Y_4) \vee (f_0 \vee \varphi_0) F(X_4 = Y_4)} = \\
&= \overline{F(X_4 < Y_4) \vee \overline{f_0} \cdot \overline{\varphi_0} \cdot F(X_4 = Y_4)} = \\
&= \overline{F(X_4 < Y_4)} \cdot \overline{[\overline{f_0} \cdot \overline{\varphi_0} \vee F(X_4 = Y_4)]} = \\
&= \overline{F(X_4 < Y_4)} \cdot \overline{F(X_4 = Y_4) \vee \overline{f_0} \cdot \overline{\varphi_0} F(X_4 < Y_4)} = \\
&= \overline{F(X_4 > Y_4) \vee f_0 \overline{\varphi_0} [F(X_4 > Y_4) \vee E(X_4 = Y_4)]} = \\
&= \overline{F(X_4 > Y_4) \vee f_0 \overline{\varphi_0} F(X_4 > Y_4) \vee f_0 \overline{\varphi_0} F(X_4 = Y_4)} = \\
&= \overline{F(X_4 > Y_4) \vee f_0 \overline{\varphi_0} F(X_4 = Y_4)}
\end{aligned}$$

Функции  $f_4, \varphi_4, g_4$  реализованы в К561ИП2.

$$f_4 = F(X_4 \leq Y_4) = F(X_4 < Y_4) + f_0 F(X_4 = Y_4) = \begin{cases} F(X_4 < Y_4) & \text{при } f_0 = 0 \\ F(X_4 \leq Y_4) & \text{при } f_0 = 1 \end{cases}$$

$$\varphi_4 = F(X_4 = Y_4) = \varphi_0 \cdot \bigwedge_{p=1}^4 \overline{x_p \oplus y_p} = \begin{cases} 0 & \text{при } \varphi_0 = 0 \\ F(X_4 = Y_4) & \text{при } \varphi_0 = 1 \end{cases}$$

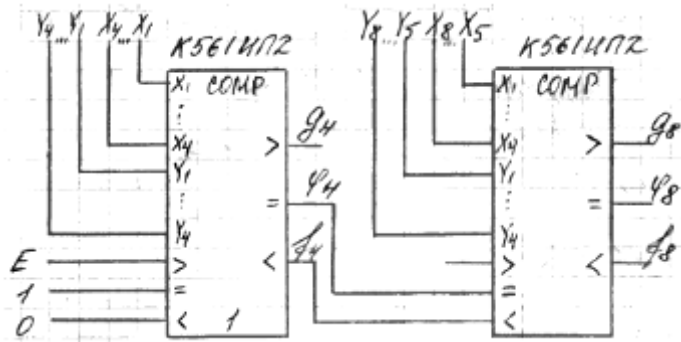
$$g_4 = E(\overline{f_4 + \varphi_4}) = E[F(X_4 < Y_4) + \overline{f_0} \cdot \overline{\varphi_0} \cdot F(X_4 = Y_4)] = \begin{cases} F(X_4 \geq Y_4), f_0 + \varphi_0 = 0, E = 1 \\ F(X_4 > Y_4), f_0 + \varphi_0 = 1, E = 1 \\ 0, E = 0 \end{cases}$$

Полученные формулы можно обобщить для  $n$  (заменив 4 на  $n$ ).

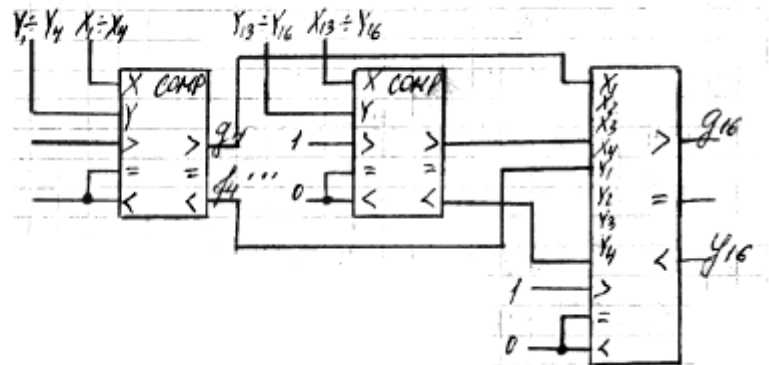
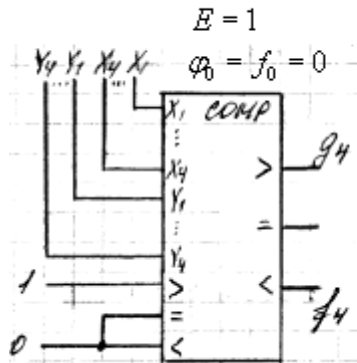
E	$f_0$	$\varphi_0$	$g_4$	$\varphi_4$	$f_4$
0	0	0	0	0	$X_4 < Y_4$
0	0	1	0	0	$X_4 \leq Y_4$
0	1	0	0	$X_4 = Y_4$	$X_4 < Y_4$
0	1	1	0	$X_4 = Y_4$	$X_4 \leq Y_4$
1	0	0	$X_4 \geq Y_4$	0	$X_4 < Y_4$
1	0	1	$X_4 > Y_4$	0	$X_4 \leq Y_4$
1	1	0	$X_4 > Y_4$	$X_4 = Y_4$	$X_4 < Y_4$
1	1	1	$X_4 > Y_4$	$X_4 = Y_4$	$X_4 \leq Y_4$

Нарращивание разрядности схем сравнения.

$$\begin{aligned}
\varphi_4 &= \varphi_0 \bigwedge_{p=1}^4 \overline{x_p \oplus y_p} \\
f_4 &= F(X_4 < Y_4) \vee f_0 F(X_4 = Y_4) \\
g_4 &= E[F(X_4 > Y_4) \vee \overline{f_0} \cdot \overline{\varphi_0} F(X_4 = Y_4)] \\
\varphi_8 &= \varphi_0 \bigwedge_{p=1}^8 \overline{x_p \oplus y_p} \\
f_8 &= F(X_8 < Y_8) \vee f_0 F(X_8 = Y_8) \\
g_8 &= E[F(X_8 > Y_8) \vee \overline{f_0} \cdot \overline{\varphi_0} F(X_8 = Y_8)]
\end{aligned}$$

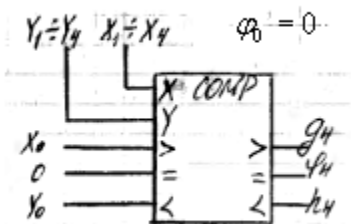


К561ИП2 в качестве преобразователя разрядности



**К555СП1** – схема сравнения для четырехразрядных чисел, симметрична относительно функций  $=$ ,  $>$ ,  $<$ . Может использоваться в качестве 5-разрядной схемы и как преобразователь разрядности.

$$\begin{aligned}
 \bar{F}_4 &= F(X_4 < Y_4) \vee \bar{F}_0 \cdot F(X_4 = Y_4) \quad ( < ) \\
 \bar{L}_4 &= F(X_4 > Y_4) \vee \bar{F}_0 \cdot F(X_4 = Y_4) \quad ( > ) \\
 G_4 &= E \cdot \bar{F}_4 \vee \bar{F}_4 = E [F(X_4 > Y_4) \vee \bar{F}_0 \cdot F(X_4 = Y_4)] \\
 L_4 &= E \cdot \bar{L}_4 \vee \bar{L}_4 = E [F(X_4 < Y_4) \vee \bar{F}_0 \cdot F(X_4 = Y_4)] \\
 F_4 &= F_0 \cdot F(X_4 = Y_4)
 \end{aligned}$$



## 5. Контроль по четности/нечетности, контроль по коду Хемминга.

Контроль – выявление неисправности или правильности работы ЭВМ. Существуют 2 направления контроля: программный (нужно время + память), аппаратный. Передача информации – наиболее популярная операция ЭВМ.

Одиночные ошибки (в одном разряде) – наиболее вероятные. Двоичные реже. Основная задача – обнаружение одиночных ошибок.

Результатом проверки м.б.:

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

### 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  $\alpha$ .

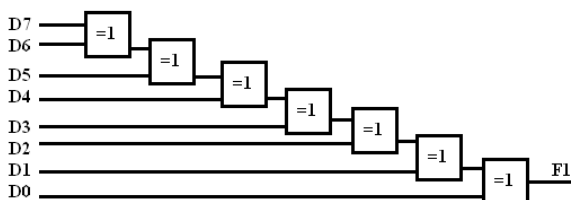
Кодовое слово  $D_7D_6D_5D_4D_3D_2D_1D_0\alpha$ .

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  $\alpha$  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.



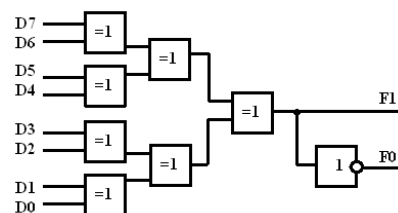
Время задержки распространения свертки зависит от времени задержки распространения в ЛЭ сложения по модулю 2:

$$t_{зд.п.св} = (n-1)t_{зд.п.сл} \underset{n=8}{=} 7t_{зд.п.сл}, \quad n - \text{количество}$$

разрядов информационного слова.

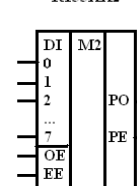
Для повышения быстродействия используют не параллельное, а последовательное включение элементов.

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

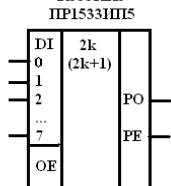


$$t_{зд.п.св} = (\log_2 n)t_{зд.п.сл} \underset{n=8}{=} 3t_{зд.п.сл}$$

ГОСТ 2.743-82  
К155ИП2



ГОСТ 2.743-91  
К555ИП5  
ПП1533ИП5



Обозначение схем свертки

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

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

Выходы взаимно ортогональны.  $PO = OE \oplus DI0 \oplus DI1 \oplus \dots \oplus DI7 = OE \oplus \sum_{i=0}^7 *DI_i$ .

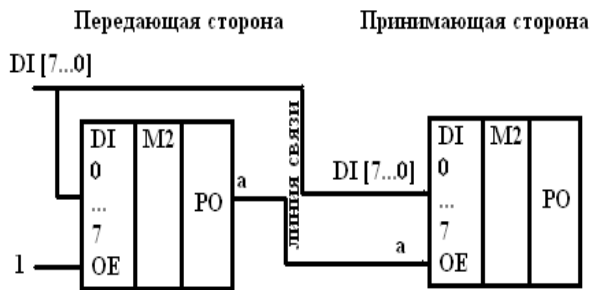
Вход OE используется для управления микросхемой и перевода ее в режим четности/нечетности и наращивания схемы.

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.

### Схема контроля по коду Хэмминга

Позволяет обнаружить и исправить одиночные ошибки, а также обнаружить ошибки больше кратности (с помощью модифицированного кода Хэмминга).

Допустим, что информационное слово содержит n разрядов. Требуется число контрольных разрядов.  $m=n+k$  – кодовое слово. Цифровой код на приемной стороне из контрольных разрядов образует слово, называемое синдромом (опознавателем). При этом синдром указывает на номер разряда, в котором возникла ошибка. При передаче m-разрядного кодового слова возможно появление (m+1) события  $\Rightarrow k \geq \log_2(m+1) = \log_2(n+k+1)$ .

Кодирование по Хэммингу:

1. Определяется количество контрольных разрядов k. Например,  $n=11$ ,  $k=4$ ,  $m=n+k=15$ .
2. Формируется кодовое слово и его разрядная сетка. Нумерация разрядов начинается с 1.
3. Контрольные разряды занимают позиции 1,2,4,8 и т.д., т.е. степени двойки.

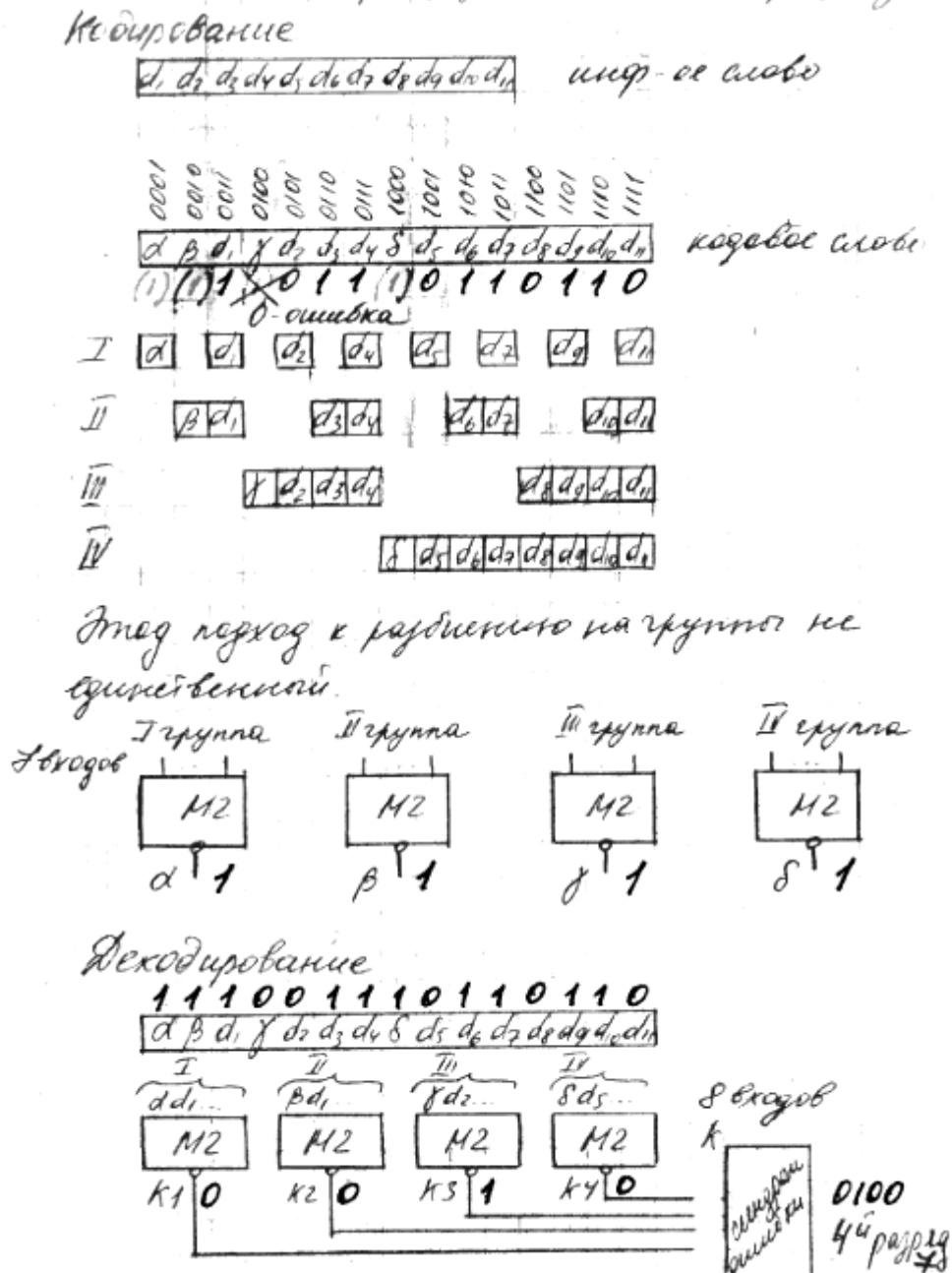
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\alpha$	$\beta$	d1	$\gamma$	d2	d3	d4	$\delta$	d5	d6	d7	d8	d9	d10	d11
1001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

4. Разряды кодового слова нумеруются двоичными числами.
5. Кодовое слово разбивается на группы. Количество групп равно количеству контрольных разрядов. В 1 группу включаются разряды кодового слова, в младшей позиции номеров которых содержится 1. Во 2 группу – разряды, во 2-ой позиции двоичных номеров которых содержится 1 и т.д. Контрольный разряд всегда входит в одну группу, а информационные могут входить сразу в несколько групп (по крайней мере в две).
6. Для каждой группы определяется функция четности, поэтому при передаче каждой группы осуществляется контроль по нечетности.

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

K555ВЖ1 – данная МС служит для кодирования и проверки 16-разрядного информационного слова. МС выдает сигналы одиночной и многократной ошибки.

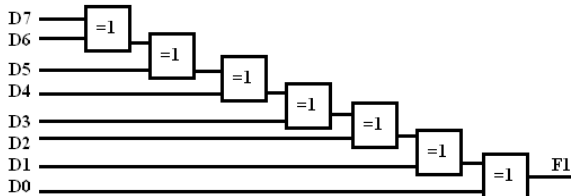


## 6. Схемы свертки.

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

Значение контрольного разряда  $\alpha$  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.

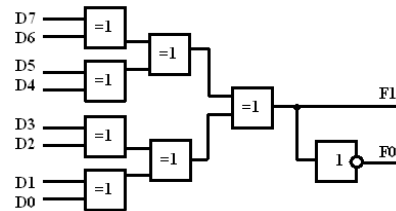
$F_1 = ((((((D_7 \oplus D_6) \oplus D_5) \oplus D_4) \oplus D_3) \oplus D_2) \oplus D_1) \oplus D_0$  - последовательная реализация.



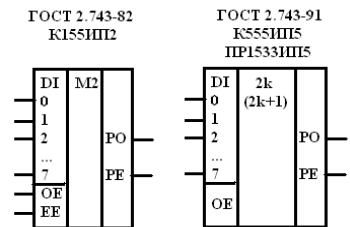
Время задержки распространения свертки зависит от времени задержки распространения в ЛЭ сложения по модулю 2:  
 $t_{зд.п.св} = (n-1)t_{зд.п.сл} \underset{n=8}{=} 7t_{зд.п.сл}$ ,  $n$  - количество разрядов информационного слова.

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.



$$t_{зд.п.св} = (\log_2 n)t_{зд.п.сл} \underset{n=8}{=} 3t_{зд.п.сл}$$



Обозначение схем свертки

O-odd, E-even, E-enable. OE – разрешение 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.

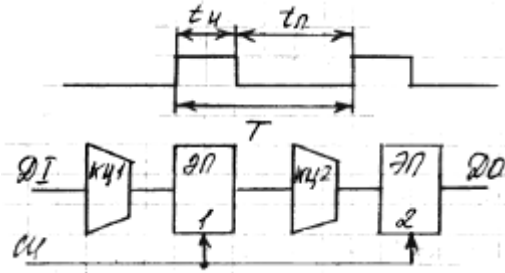
Если на принимающей стороне

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.

## 7. 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

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

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

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

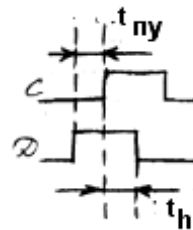
2) синхронные одноступенчатые с динамическим управлением записью

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

$$t_n \geq t_{\text{зд.р.КЦ}} + t_{\text{линий_связи}} + t_{\text{ny}}, t_{\text{ny}} = 5 \div 6 \text{ нс}$$

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

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



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).

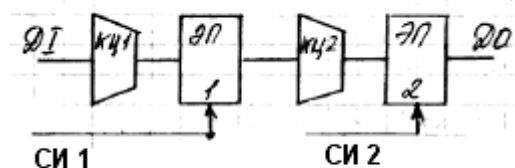
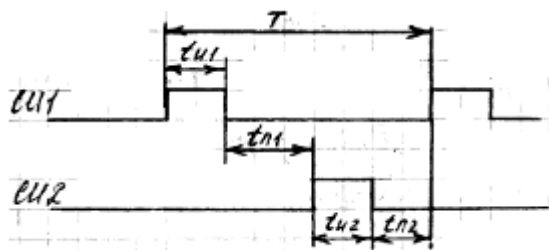
$$t_u \geq t_{\text{зап.тр.И_ст}}$$

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

## 8. 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).



$$t_{u1} \geq t_{\text{зан.тр.}}$$

$$t_{n1} \geq t_{\text{зд.р.КЦ1}} + t_{\text{линий\_связи\_1}} + t_{\text{пу}}$$

$$t_{u2} \geq t_{\text{зан.тр.}}$$

$$t_{n2} \geq t_{\text{зд.р.КЦ2}} + t_{\text{линий\_связи\_2}} + t_{\text{пу}}$$

## 9. 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).

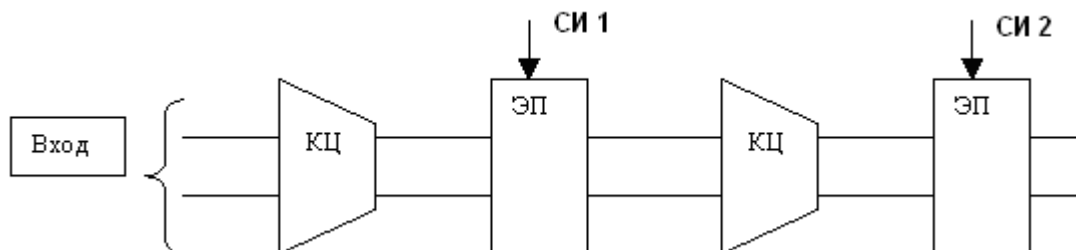
В синхронных схемах процессы обработки информации синхронизируются, т.е. выполняются под действием синхросигнала. Период действия синхросигнала выбирается так, чтобы во время его действия переходные процессы были закончены.

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.

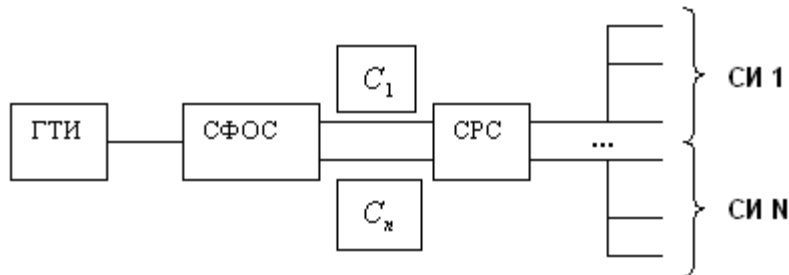


Основные параметры схемы синхронизации:

- 1)  $t_u$  - длительность импульса. За длительность импульса информация записывается в триггере, т.е.  $t_u \geq t_{\text{зап.тр. max}}$ . Но  $t_{\text{зап.тр.}} \approx t_{\text{зд.п.тр.}}$ , значит  $t_u \geq t_{\text{зд.п.тр.}}$ . Если элементы КЦ и ЭП безинерционные (задержка = 0), то синхронизация невозможна.
- 2)  $t_n$  - длительность паузы.  $t_n \geq t_{\text{зд.п.КЦ max}} + t_{\text{линий связи}} + t_{\text{пу max}}$
- 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.

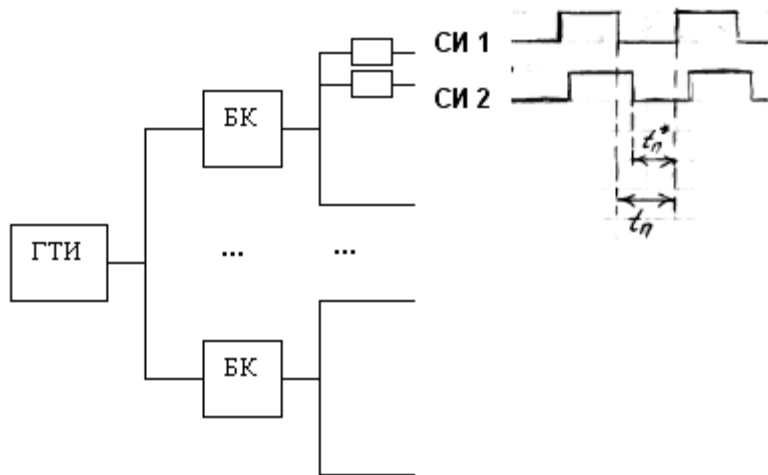
СФОС – схема формирования опорных синхросигналов.

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.



$f_0$  - опорная частота,  $f_{\text{раб}}$  - рабочая частота.

$f_{\text{раб}} = f_0(1 \pm \delta f)$ , где  $\delta f = \frac{\Delta f}{f_0}$ ,  $\Delta f$  - абсолютное изменение,  $\delta f$  - относительное.

$f_{\text{раб. max}} = f_0(1 + \delta f)$ ,  $f_{\text{раб. min}} = f_0(1 - \delta f)$ ,  $2\delta f$  - потеря частоты.

## 10. Регистры. Классификация. Параметры. Параллельные регистры. Регистры сдвига. Универсальные регистры. Синтез.

**Регистр** – это операционный узел, выполняющий функции записи, хранения, считывания информации, а также преобразования последовательного кода слова в параллельный и обратно, и выполняющий поразрядные логические операции (логическое сложение, умножение, исключающее или, инверсию).

Регистр представляет собой набор элементов памяти (триггеров), число которых равно или кратно числу разрядов преобразуемого слова, и комбинационных схем.

Регистры имеют независимые информационные входы для каждого разряда и общие входы управления.

Основной признак классификации: способ ввода/вывода:

- 1) Параллельный. Все разряды слова передаются одновременно, за 1 такт  $T$  (простым или парафазным кодом:  $x_i$  -простой,  $\overline{x_i} x_i$  -парафазный).
- 2) Последовательный. Все разряды слова передаются последовательно во времени, каждый за 1 такт  $T$ . Время передачи  $nT$ .
- 3) Параллельно-последовательный способ. Слово разбивается на группы разрядов. Группы передаются параллельно, разряды в группах последовательно.

В зависимости от выполняемых функций и способа ввода/выводы:

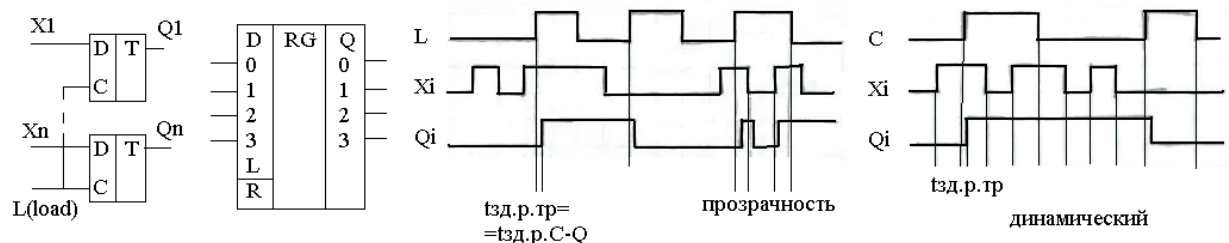
- 1) регистры памяти (параллельные)
- 2) регистры сдвига (последовательные)
- 3) в интегральной схемотехнике имеются универсальные регистры (ввод, хранение, сдвиг). Они могут быть программно управляемы.

В регистрах используются D, DV, RS триггеры (JK не применяется, т.к. его функция избыточна для регистра). Чаще всего D.

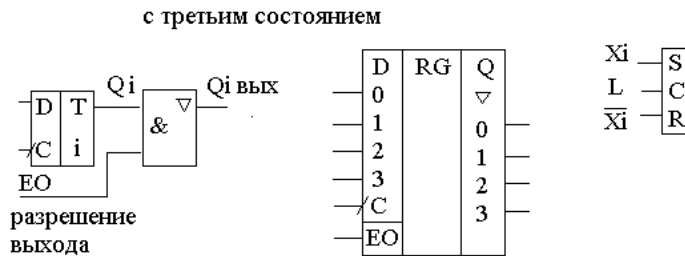
Регистры могут иметь сложности цепей ввода и вывода: на входе может осуществляться мультиплексирование. Вывод может выполняться простым/обратным/парафазным кодом. Выход может иметь или три состояния или открытый коллектор.

**Регистры памяти (параллельные регистры).** Запись, хранение, выдача информации. Для временного хранения информации в течении 1 или нескольких циклов работы устройства. Чаще всего с однофазным вводом на основе D-триггеров (как со статическим, так и с динамическим управлением записью).

Со статическим входом – прозрачный триггер. Обычно 4х или 8ми разрядные.



С динамическим управлением записью. Данные триггера непрозрачны. Вход R установки в начальное нулевое состояние – асинхронный. Это расширяет возможности регистра – возможна очистка регистра.



Нарращивание: секции параллельных регистров запарааллеливают по С, по линии сброса в 0, и по другим управляющим сигналам.

**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

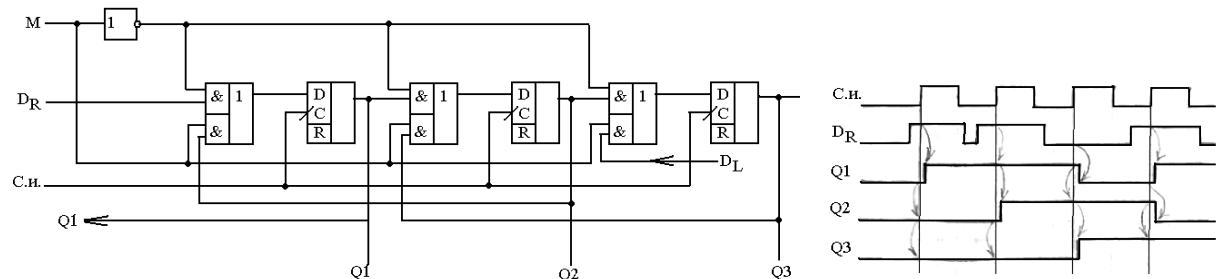
M=0 – сдвиг вправо

M=1 – сдвиг влево

$$D_i = Q_i^* = \bar{M} \cdot Q_{i-1} + M \cdot Q_{i+1}$$

Со входом M  $\Rightarrow$  в обе стороны.

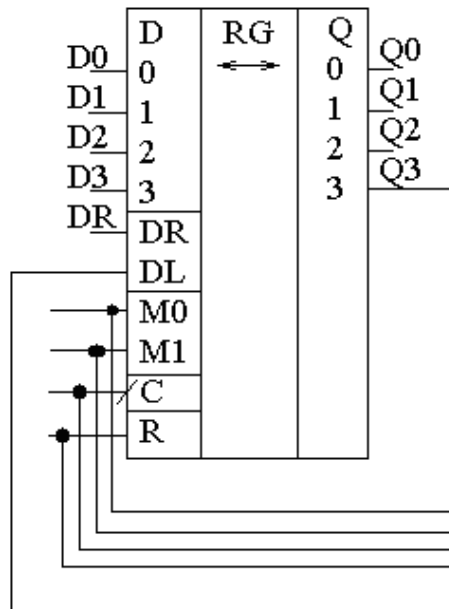
Временная диаграмма для сдвига вправо.



Быстродействие падает из-за использования элементов И-ИЛИ:

$$t_{зд.p.RG} = t_{зд.p.mp} + t_{зд.p."И-ИЛИ"}$$

Нарращивание:



M <sub>1</sub>	M <sub>0</sub>	Микрооперация
0	0	Хранение
0	1	Сдвиг вправо
1	0	Сдвиг влево
1	1	Загрузка нач. слова

### Универсальные регистры.

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<sub>1</sub>, M<sub>0</sub> – управляющие слова.

M <sub>1</sub>	M <sub>0</sub>	D <sub>вх</sub>	Q <sub>i-1</sub>	Q <sub>i</sub>	Q <sub>i+1</sub>	Q <sub>i</sub> *
0	0	x	x	0	x	0
0	0	x	x	1	x	1
0	1	x	0	x	x	0
0	1	x	1	x	x	1
1	0	x	x	x	0	0
1	0	x	x	x	1	1
1	1	0	x	x	x	0
1	1	1	x	x	x	1

Все микрооперации выполняются тактов. сигналом C

Таблица переходов универсального регистра сдвига (i-й разряд)

M <sub>1</sub> , M <sub>0</sub> , D <sub>вх</sub>	Q <sub>i-1</sub> , Q <sub>i</sub> , Q <sub>i+1</sub>	000	001	011	010	110	111	101	100
000							1		
001							1	1	1
011	1	1					1	1	1
010	1	1					1		
110	1	1	1	1			1		
111	1	1	1	1			1	1	1
101			1	1			1	1	1
100			1	1			1		

$$Q_i^* = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0$$

$$D_i = Q_i^*$$

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

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

$$(3) D_i = \overline{M_1} \overline{M_0} Q_i + \overline{M_1} M_0 Q_{i-1} + M_1 \overline{M_0} D_L + M_1 M_0 D_{iBX} \quad (1), (2), (3)$$

—

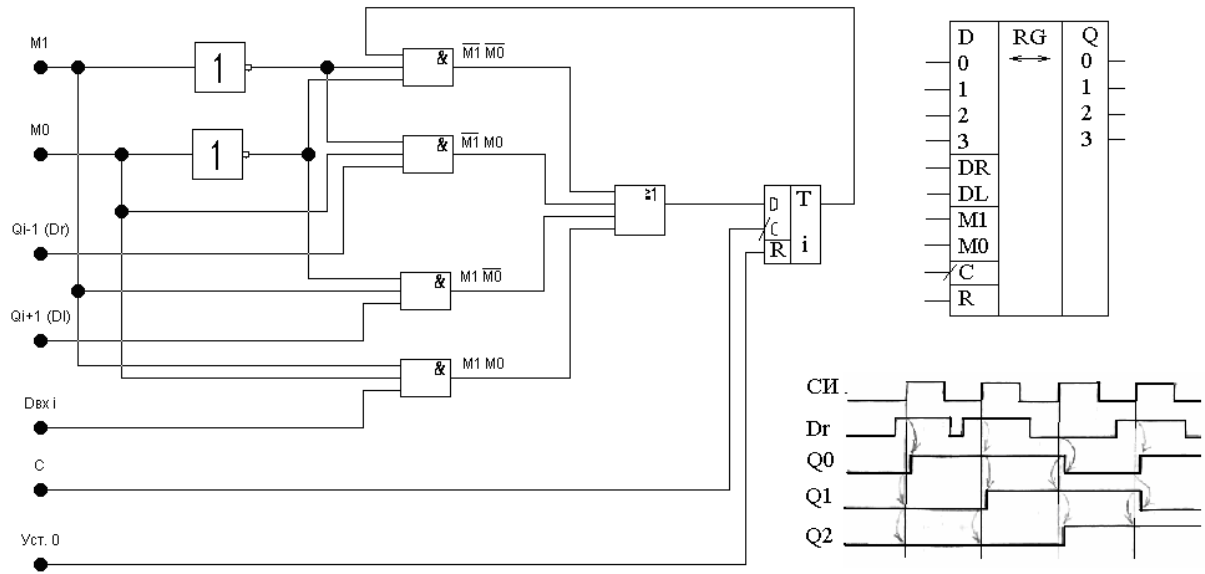
функции

демультиплексирования

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

Входы D<sub>R</sub> и D<sub>L</sub> используются для наращивания разрядности.





## 11. Синхронные счетчики. Основные параметры. Методика синтеза. Синтез суммирующих, вычитающих и реверсивных счетчиков. Нарращивание разрядности счетчиков по последовательной и «быстрой» схемам.

Счетчик – ФУ, выполняющий микрооперации счета и хранения в определенной системе счисления числа событий (импульсов) в какой-либо точке цифрового устройства. Различают счетчики и их разновидности – пересчетные устройства. В счетчике важен не только процесс счета, но и представление числа в определенном коде. В пересчетном устройстве способ представления не так важен, оно выдает сигнал на выход при наступлении определенного числа событий. 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:  $U_{BX}^0, U_{BX}^1, U_{ВЫХ}^0, U_{ВЫХ}^1, U_{ПИТ}, P_{ПОТР}$ , логический перепад и прочие..

II ) Динамические:

- 1) частота счета максимальная и рабочая ( $1/10 \max$ )
- 2) задержки распространения сигналов от вх. до вх. по различным трактам  $t_{зд.р}, t_{зд.р.С-Q}, t_{зд.р.В-Q}$
- 3) время предустановки
- 4) время удержания информационного сигнала на определенных входах по отношению к управляющему сигналу
- 5) минимальная длительность входных сигналов по различным трактам

III ) Специфические: Модуль счета  $M$  и емкость счетчика  $E$ .  $M$  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.

Классификация:

I ) по модулю счета:

- 1) двоичные
- 2) двоично-кодированные
- 3) с произвольным модулем
- 4) программируемые

II ) по выполняемой функции:

- 1) суммирующие  $СТ := СТ + 1$  (count – счетчик)
- 2) вычитающие  $СТ := СТ - 1$
- 3) реверсивные (и то, и другое, по управляющему сигналу): с одним счетным входом, с двумя счетными входами.

III ) по способу организации переносов:

- 1) с последовательным
- 2) со сквозным
- 3) с параллельным
- 4) с групповым

IV) по способу переключения счетчика:

- 1) асинхронные. Переключаются или входными сигналами, или выходными сигналами других разрядов, или их комбинацией (самые простые).

2) синхронные. Переключаются входными сигналами, т.е. триггеры (разряды счетчика) переключаются в один момент времени входными сигналами счета (самые быстродействующие)  $t_{зд.р} = t_{зд.р.1mp-pa}$

V) по порядку изменения состояний:

1) с естественным порядком изменения состояний; 2) с произвольным порядком

### Методика синтеза счетчиков

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  $M$  – наибольший номер состояния.

2. Составляется таблица переходов и функция возбуждения.

**Синтез реверсивного счетчика по модулю 5 на JK и на D триггерах: (new, июнь 2007, сделал его, ибо он сложнее сумм. и вычит. – для них берете часть таблиц, где соответственно  $M=0$  и  $M=1$ )**

$n = \lceil \log_2 5 \rceil = 3$ .  $M=0$  – суммирующий счетчик.  $M=1$  – вычитающий счетчик.

M	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
0	0	0	1	0	1	0	0	a	1	a	a	1	0	1	0
0	0	1	0	0	1	1	0	a	a	0	1	a	0	1	1
0	0	1	1	1	0	0	1	a	a	1	a	1	1	0	0
0	1	0	0	0	0	0	a	1	0	a	0	a	0	0	0
1	1	0	0	0	1	1	a	1	1	a	1	a	0	1	1
1	0	1	1	0	1	0	0	a	a	0	a	1	0	1	0
1	0	1	0	0	0	1	0	a	a	1	1	a	0	0	1
1	0	0	1	0	0	0	0	a	0	a	a	1	0	0	0
1	0	0	0	1	0	0	1	a	0	a	0	a	1	0	0

$$K_0=1, J_0 = M \oplus \overline{Q_2} + MQ_1, K_1 = M \oplus Q_0, J_1 = \overline{M}Q_0 + MQ_2, K_2=1, J_2 = \overline{M}Q_0Q_1 + M\overline{Q_0}Q_1$$

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

$$D_2 = \overline{M}Q_0Q_1 + M\overline{Q_0}Q_1Q_2, D_1 = \overline{M}(Q_0 \oplus Q_1) + MQ_0Q_1 + MQ_2Q_1, D_0 = \overline{M}Q_0Q_2 + Q_0Q_1 + MQ_2$$

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

Далее тупо строишь схему на JK-триггерах и на D-триггерах.  $\oplus$  - сложение по модулю 2.

Параметры быстродействия:



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

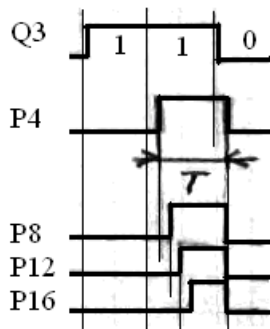
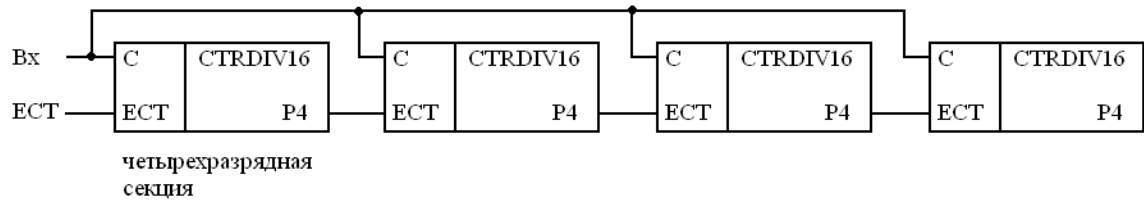
$$t_u = t_{зд.р.мр.}$$

$$t_u > t_{зд.р.мр.}$$

$$t_n = t_{зд.р.к.с.} + t_{н.у.}$$

Обычно  $t_{н.у.}$  в справочных данных.  $t_{зд.р.с-Q} = t_{зд.р.мр.}$

**Наращивание по последовательной схеме:**

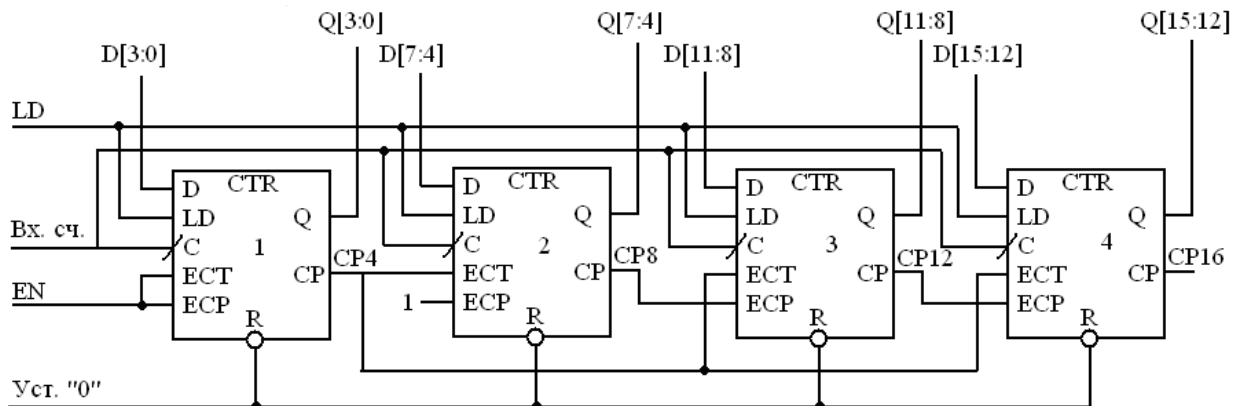


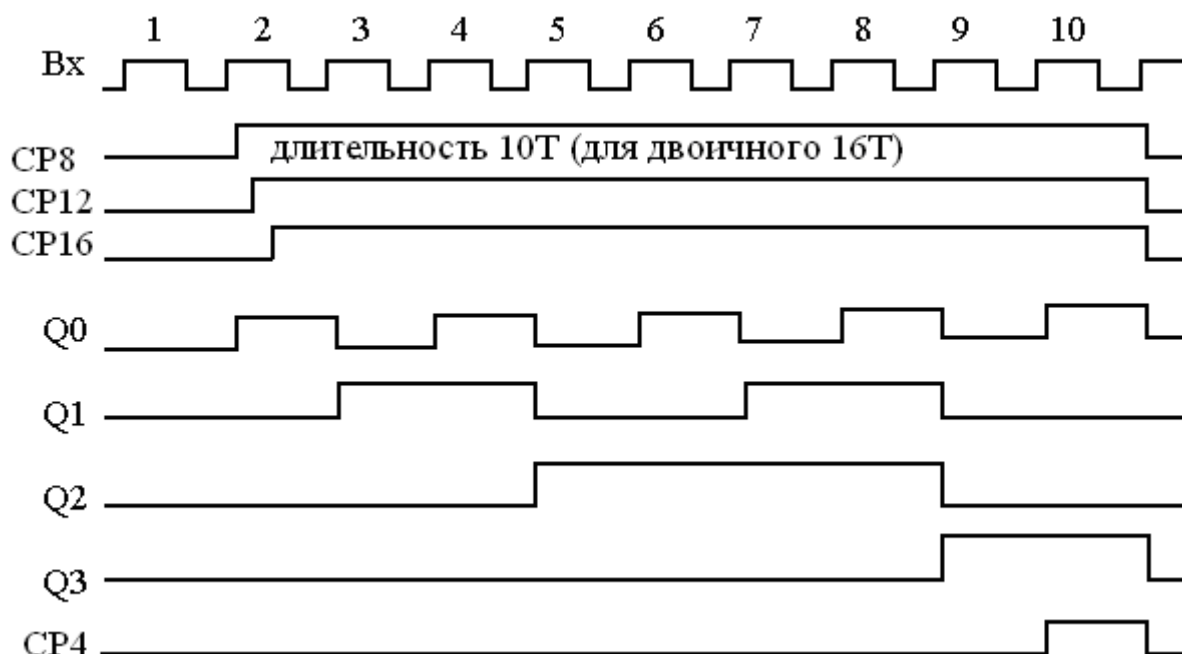
Уменьшение длительности сигнала Р от секции к секции приводит к ограничению по разрядности

**Наращивание синхронных счетчиков по схеме высокоскоростного счета.**

Построить систему из нескольких разрядов, обладающую такой же рабочей частотой, как и отдельный счетчик. EN=0 – счет запрещен. EN=1 – счет и перенос разрешен.

Временная диаграмма двоично-десятичного счетчика.  $CP_4 = ECP \cdot Q_0 \cdot Q_3$





## 12. Серии ИМС ТТЛ(Ш), основные параметры, особенности применения.

Зарубежные серии ИМС	Отечественные Серии ИМС	
SN54	133	Общего применения
SN74	K155	Специального применения
SN54L SN74L	136,158 K134	С пониженным потреблением мощности
SN54H SN74H	130 K131	С повышенным быстродействием, но повышенным потр. мощн.
SN54S SN74S	530 K531	Шоттки Schottky
SN54LS SN74LS	533 K555	
SN54ALS SN74ALS	1533 KP1533	Усовершенствованные Advance
SN54F SN74F	1531 KP1531	

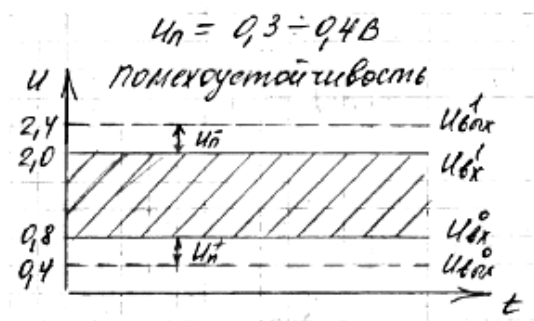
Особенность применения: Если в схеме не используется какой-либо ЛЭ, то он должен быть переведен в режим наименьшей мощности. В ТТЛ(Ш) есть выходной каскад, в котором в течение короткого времени протекает сквозной ток, т.е. повышается  $P_{пот}$ . Она определяется  $P_{пот.ст}$  - недостаток. В ТТЛ(Ш) с ОК нет сквозных токов.

Динамические параметры ИМС (усредненные).

	$t_{зд.р.}$ , нс	$P_{пот}$ , мВт	$A_n$ , пДж
K155	10	10	100
K134	33	1	33
K131	6	22	132
K531	3	19	57
K555	9.5	2	19
KP1533	4	3	12
KP1531	2	4	8

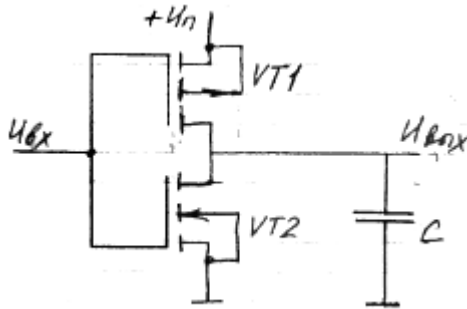
Параметры базового ЛЭ «И-НЕ» серии ТТЛ(Ш)

	K155	K134	K131	K531	K555	KP1533	KP1531
$I_{\text{ex}}^0$ , мА, не более	-1.6	-0.18	-2.3	-2.0	-0.4	-0.2	-0.6
$I_{\text{ex}}^1$ , мкА, не более	40	12	70	50	20	20	20
$U_{\text{вых}}^0$ , В, не более	0.4	0.3	0.35	0.5	0.4	0.5	0.5
$U_{\text{вых}}^1$ , В, не менее	2.4	2.5	2.7	2.5	2.4	2.5	2.5
$K_{\text{PA3}}$	10	10	10	10	20	20	20



### 13. Серии ИМС КМОП, основные параметры, особенности применения.

Зарубежные серии ИМС	Отечественные Серии ИМС	КМОП (комплементарная логика на транзисторах металл-оксид-полупроводник)
CD4000	164, K176	
CD4000A	564, K561	A-Advance
CD4000B	KP1561	В-буферные
CD4000UB	-	
54/74 HC	1564	C-CMOS, КМОП
54/74 AC	KP1554	Advanced CMOS
54/74 ACT	1544	Т-совместимые с ТТЛ сериями



$$I_C = K \left[ (U_{3И} - U_{ПОР}) U_{СИ} - \frac{U_{СИ}^2}{2} \right]$$

$$U_{СИ} \ll 1 B \Rightarrow I_C = K (U_{3И} - U_{ПОР}) U_{СИ}$$

$$R_K = \frac{U_{СИ}}{I_C} = \frac{1}{K (U_{3И} - U_{ПОР})}$$

$$U_{3И} = U_{П} \Rightarrow R_K = \frac{1}{K (U_{П} - U_{ПОР})}$$

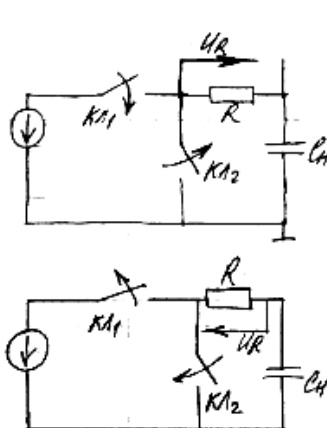
Мощность, потребляемая на КМОП – маленькая, поэтому КМОП широко применяются. Обычно применяются транзисторы с изолированным истоком, следовательно входные токи малы и определяется фактически током утечки:

$$I_{ВХ} = I_{УТ} \leq 1 \text{ мкА}$$

$$P_{ПОТР} = P_{СТАТИЧ} + P_{ДИНАМ}, P_{СТ} \approx 0 \text{ (неск. мкВт)}$$

$$P_{ДИН} = P_{ДИН1} + P_{ДИН2}, \text{ где } P_{ДИН2} - \text{мощность сквозного тока при переключении, } P_{ДИН1} -$$

мощность на нагрузке (ее надо перезаряжать),  $C_H = \sum_{i=1}^n C_{ВХi} + C_{СВ}$ ,  $C_{СВ}$  - связи.



$$U_R(t) = U_{П} \cdot e^{-\frac{t}{T}}, \quad I_R(t) = \frac{U_{П}}{R} \cdot e^{-\frac{t}{T}}$$

$$P(t) = U_R(t) \cdot I_R(t) = \frac{U_{П}^2}{R} \cdot e^{-\frac{2t}{T}}$$

$$P^* = \int_0^{\infty} \frac{U_{П}^2}{R} \cdot e^{-\frac{2t}{T}} dt = \frac{U_{П}^2}{R} \cdot \frac{T}{2} = \frac{U_{П}^2}{R} \cdot \frac{RC_H}{2} = \frac{U_{П}^2 C_H}{2}$$

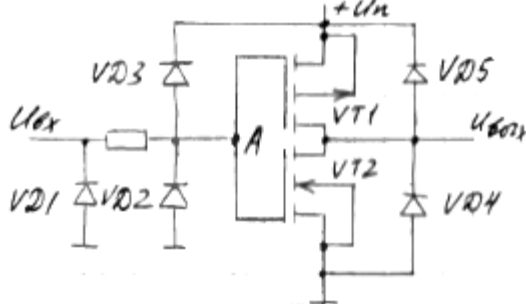
$$P_1 = U_{П}^2 C_H - \text{ для одного периода}$$

$P_{\text{дин1}} = U_{\text{п}}^2 \cdot C_{\text{н}} \cdot f$  ,  $P_{\text{дин2}} = U_{\text{п}}^2 \cdot C_{\text{PD}} \cdot f$  ,  $C_{\text{PD}}$  - справочный параметр, указывается для каждой МС

$$P_{\text{п}} = P_{\text{СТ}} + P_{\text{дин1}} + P_{\text{дин2}} = P_{\text{СТ}} + U_{\text{п}}^2 \cdot (C_{\text{н}} + C_{\text{PD}}) \cdot f$$

Особенности:

1) Схемы КМОП имеют цепи защиты от статического напряжения (иначе может произойти пробой транзисторов)



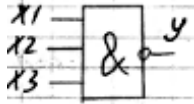
VD1, VD2, VD3 работают как ограничители.

Если  $U_{\text{BX}} > U_{\text{п}}$ , то  $U_{\text{A}} = U_{\text{п}} + U_{\text{д}}$ , а за счет VD1 и VD2  $U_{\text{A}} \geq -U_{\text{д}}$  (если  $U_{\text{BX}}$  отрицательно)  
 $-U_{\text{д}} \leq U_{\text{A}} \leq U_{\text{п}} + U_{\text{д}}$

Аналогично VD4 и VD5 на выходе:  $-U_{\text{д}} \leq U_{\text{ВЫХ}} \leq U_{\text{п}} + U_{\text{д}}$

При пайке схем КМОП паяльник должен иметь заземленное жало. Должны быть предусмотрены меры защиты человека.

2)



$y = x_1 x_2 x_3$ , если  $x_3$  не нужен, то  $y = x_1 x_2$ .

В логических схемах И, И-НЕ на незадействованный вход необходимо подавать «1», а на ИЛИ, ИЛИ-НЕ – «0» (или объединять с задействованными входами).

3) Для схем КМОП характерно такое явление как «зашелкивание». Если транзисторы р-п-р и п-р-п одновременно открыты, то в МС течет очень большой ток, и МС может выйти из строя

4) Особенности КМОП – появление паразитных транзисторов

5) На КМОП схемы нужно сначала подавать питание, и лишь затем входные сигналы

6) Схемы КМОП очень «боятся» обратного напряжения, сразу выходят из строя

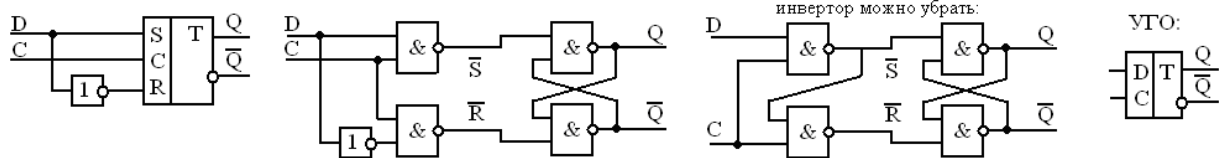
В КМОП есть выходной каскад, в котором в течение короткого времени протекает сквозной ток, т.е. повышается потребляемая мощность.



## 14. Типы триггеров (одноступенчатые, двухступенчатые, со статическим и динамическим управлением записью) и их применение в регистрах и счетчиках.

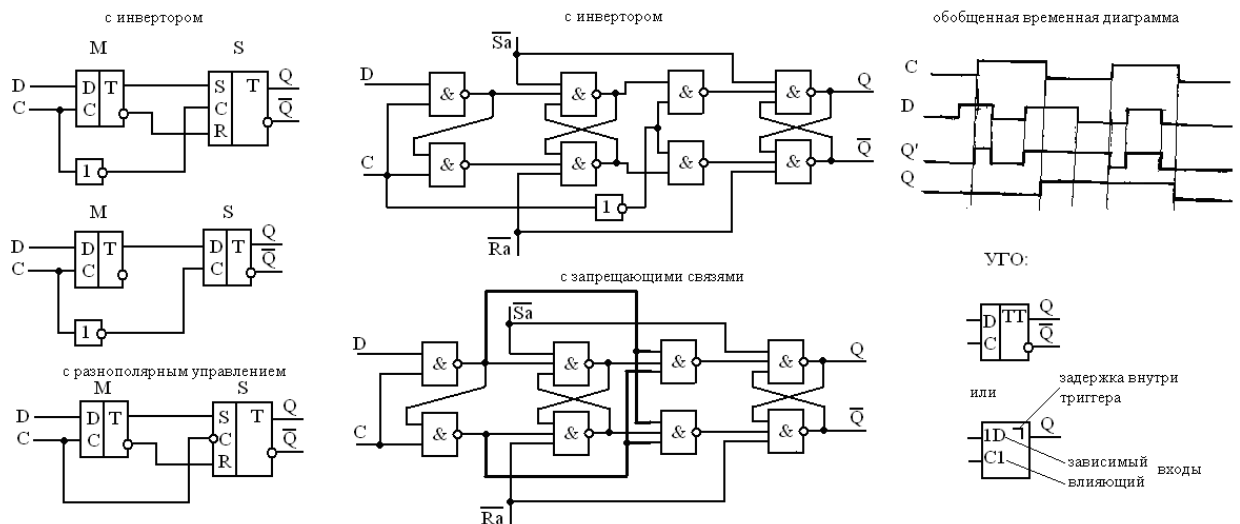
### Синхронный триггер со статическим управлением записью (одноступенчатый)

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



### Синхронный триггер с двухступенчатым запоминанием информации

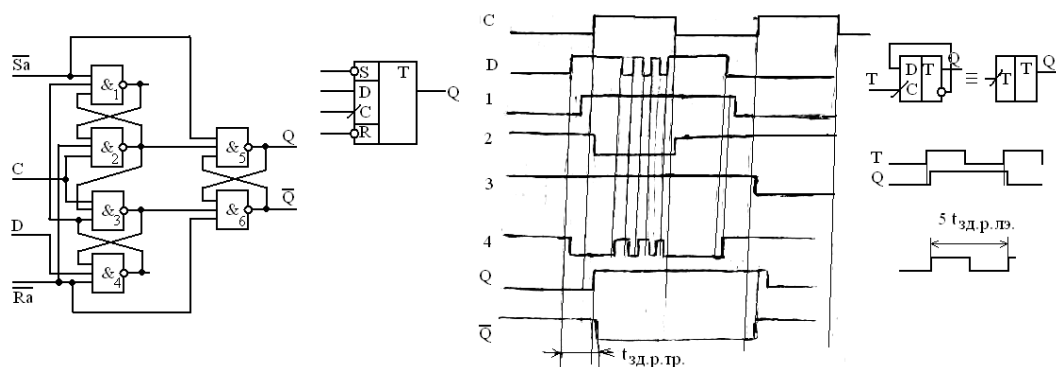
В двухступенчатых триггерах входная и выходная ступени тактируются «антисинхронно», прием информации разрешается в них поочередно. Вследствие этого отсутствует режим прозрачности триггера при любом уровне синхросигнала, что позволяет реализовать любые типы триггеров, свободные от режимов генерации, и 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.



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 триггер с динамическим управлением записью

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.



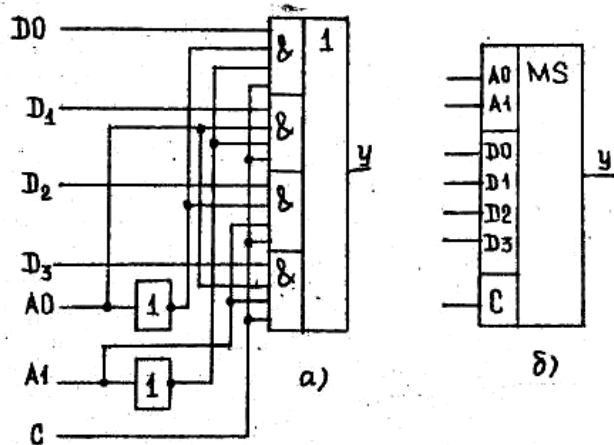
### Применение триггеров:

- 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.

## 15. Мультиплексоры. Определение, назначение, основные параметры. Реализация математических функций на мультиплексорах. Нарращивание мультиплексоров.

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

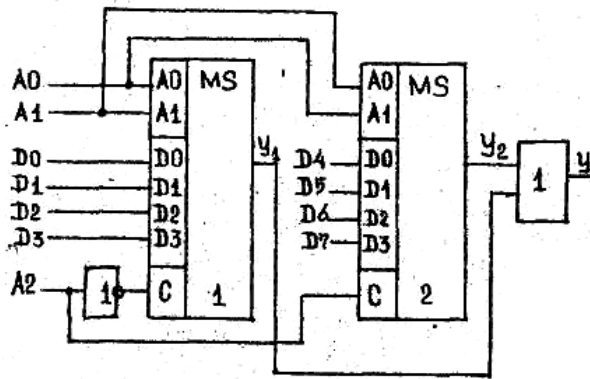
Мультиплексор реализует логическую функцию  $y = \bigvee_{j=0}^{2^n-1} D_j K_j(A_{n-1}, A_{n-2}, \dots, A_1, A_0)$ , где  $A$  - адресные входы и сигналы,  $j = 0, 1, \dots, 2^n-1$ ,  $K$  - конъюнкта  $A$ , т.е. конъюнкция всех аргументов, номер которой равен числу, образованному двоичным кодом сигналов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

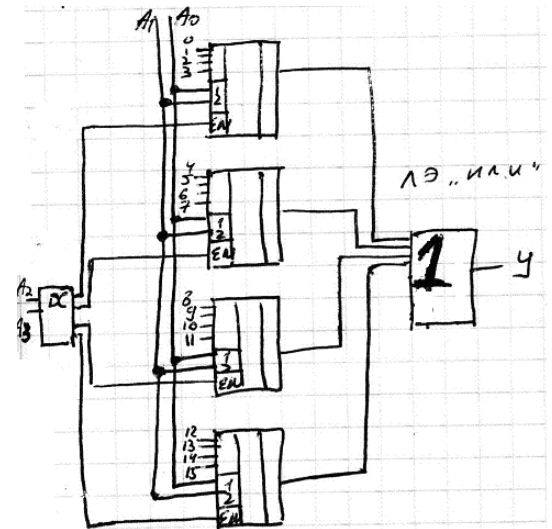
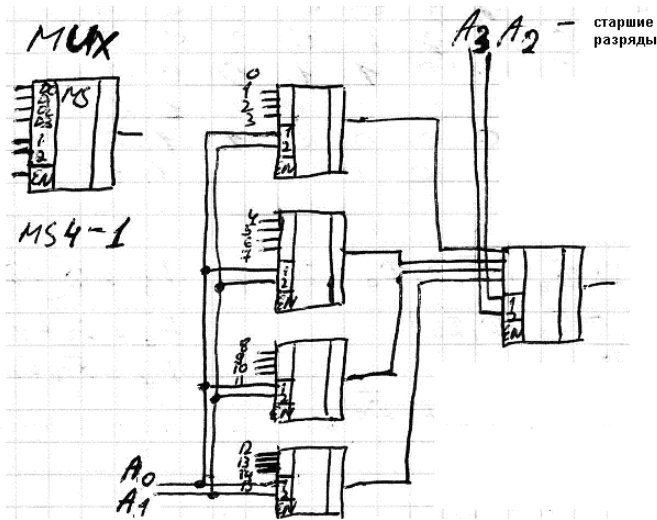


Мультиплексор состоит из дешифратора, каждый конъюнктор которого имеет дополнительный вход для соответствующего информационного сигнала  $D$ . Выходы конъюнкторов объединяются - по операции ИЛИ дизъюнктором. Время задержки распространения сигнала в мультиплексоре определяется задержкой цепи последовательно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





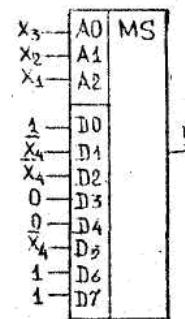
### Реализация функций.

$$f = x_1 x_2 \vee \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_3 \bar{x}_4 =$$

$$= x_1 x_2 \vee \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4.$$

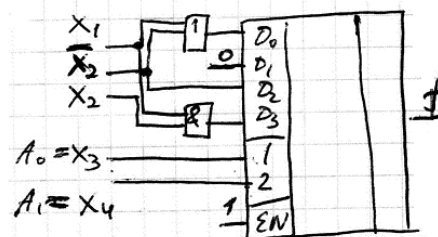
$$A_0 = x_3, \quad A_1 = x_2, \quad A_2 = x_1$$

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



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

$x_4$	$x_3$	$x_2$	$x_1$	$f$
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

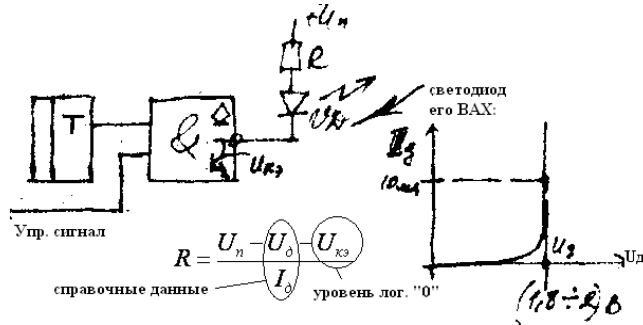


Эта схема не универсальна. Есть другая реализация. Подаем на D0,D1,D3,D8,D9,D15 единицы, а на остальные – нули, тогда  $x_4, x_3, x_2, x_1$  будут  $A_3, A_2, A_1, A_0$  соответственно.

## 16. Функциональные узлы и элементы управления знакоиндикаторами.

Шкала светоиндикации:

F7	7		111
F6	6		110
F5	5		101
F4	4		100
F3	3		011
F2	2		010
F1	1		001
F0	0		000

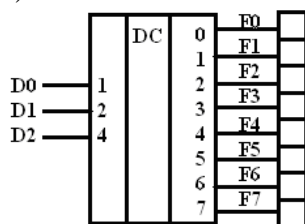


Можно использовать и ЛЭ со стандартными выходами, но чтобы они могли выдавать выходной ток порядка 10мА (КМОП такие токи выдавать не могут, только некоторые с открытым стоком, ТТЛШ определяется для конкретного случая).

Три варианта реализации индикации и управления шкалой:

D2	D1	D0	F0	F1	F2	F3	F4	F5	F6	F7	F0	F1	F2	F3	F4	F5	F6	F7	F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	1								1								1							
0	0	1		1							1	1							1	1						
0	1	0			1							1	1						1	1	1					
0	1	1				1							1	1					1	1	1	1				
1	0	0					1							1	1				1	1	1	1	1			
1	0	1						1							1	1			1	1	1	1	1	1		
1	1	0							1							1	1		1	1	1	1	1	1	1	
1	1	1								1							1	1	1	1	1	1	1	1	1	1

1) DC3-8



2)

$$F_0 = \overline{D_2} \overline{D_1}$$

$$F_1 = \overline{D_2} (D_1 \oplus D_0)$$

$$F_2 = \overline{D_2} D_1$$

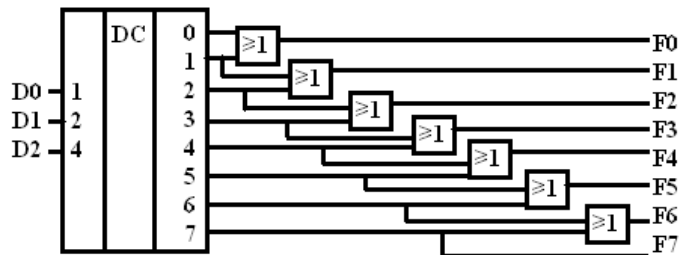
$$F_3 = \overline{D_2} D_1 D_0 + D_2 \overline{D_1} \overline{D_0}$$

$$F_4 = D_2 \overline{D_1}$$

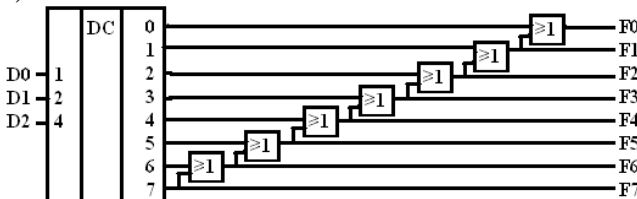
$$F_5 = D_2 (D_1 \oplus D_0)$$

$$F_6 = D_2 D_1$$

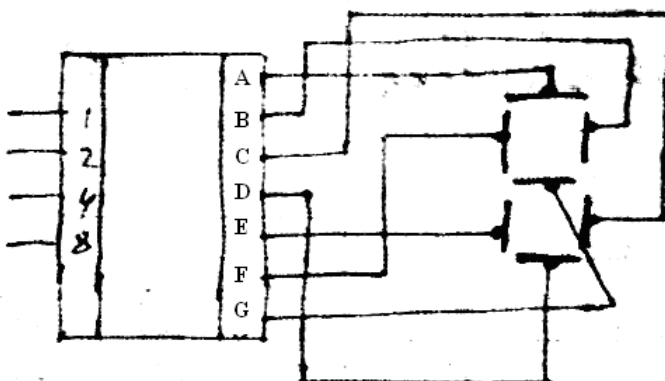
$$F_7 = D_2 D_1 D_0$$



3)



Семисегментный индикатор:



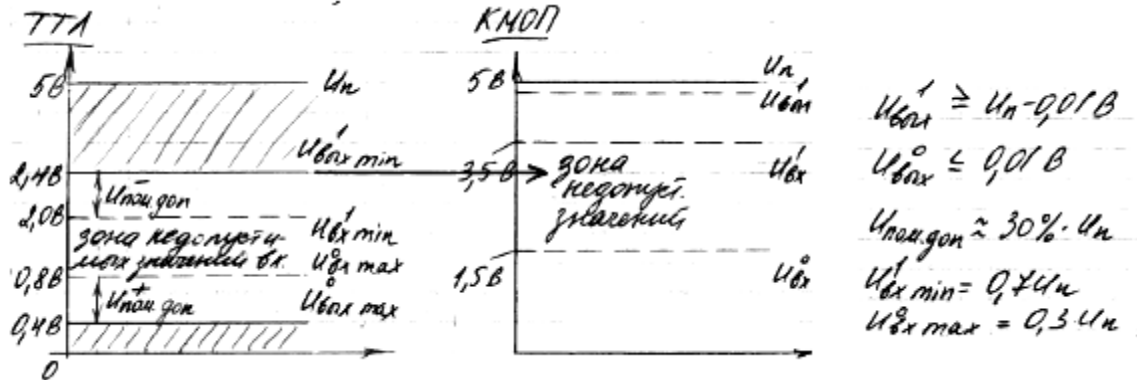
Строится таблица истинности, карты Карно, определяются логические функции, строится логическая схема.

## 17. Преобразователи уровней.

Назначение: для обеспечения совместной работы устройств на различных элементах.

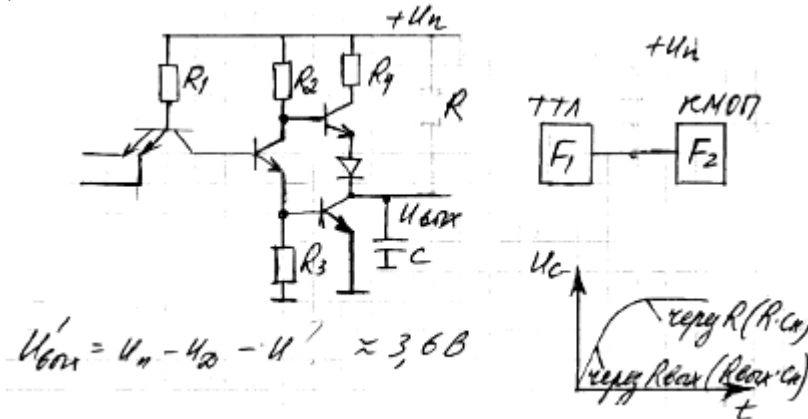
Пример: преобразователи ТТЛ-ЭСЛ (К500ПУ124), ЭСЛ-ТТЛ (К500ПУ125).

При этом преобразователи не должны снижать помехоустойчивость и быстродействие схем. ТТЛ(Ш)-КМОП, КМОП-ТТЛ(Ш)



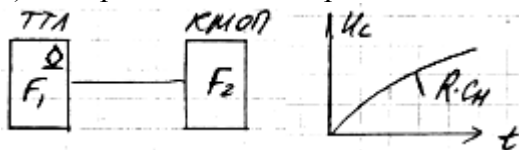
$$1. U_{П.ТТЛ} = U_{П.КМОП} = 5V$$

а)



Желательно  $R$  брать достаточно малым, чтобы в процессе заряда  $C$  не слишком замедлялся. Но чем меньше  $R$ , тем больше потребляемая мощность.

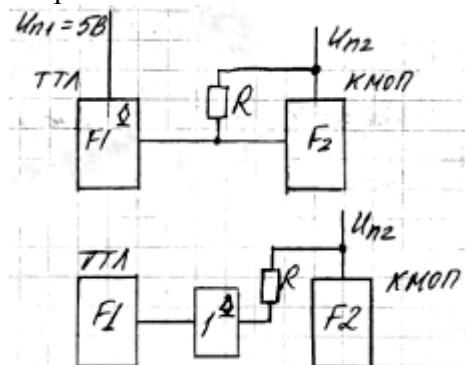
б) с открытым коллектором



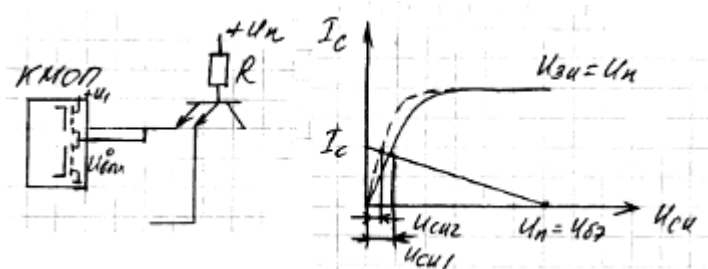
Все то же самое, только быстродействие еще больше снижается.

$$2) U_{П.ТТЛ} = 5V < U_{П.КМОП}$$

Между элементами ТТЛ и КМОП ставится буферный каскад (схема ТТЛ с ОК, напряжение питания в ней как в КМОП)



КМОП – ТТЛ : ТТЛ не могут принять такие большие входные токи.



$$I_C = \frac{U_{II} - U_{БЭ}}{R}, \text{ т.к. } I_C = b \left[ (U_{ЗИ} - U_{ПОР}) U_{СИ} - \frac{U_{СИ}^2}{2} \right]$$

Нужно делать характеристику более крутой. Должно быть  $U_{СИ} < 0.4V$ .

Используются:	K176ПУ1	
	K176ПУ2	
$U_{П1} = +9V, U_{П2} = 5V$	K176ПУ3	Два источника питания
$U_{II} = 5V$ , доп. $U_{ВХ} > U_{II}$	K561ПУ4	Имеют на выходах транзисторы с повышенной удельной крутизной $b$ и могут работать на ТТЛ
В буферных инвертирующих каскадах с входом разрешения и входом перевода в 3 состояние	K561АН1	
	K561АН2	
	K561ПУ6	

## 18. Сумматоры. Классификация. Одноразрядный сумматор. Многоразрядные сумматоры с последовательным и параллельным переносами. Оценка быстродействия.

**Сумматор** — логический операционный узел, выполняющий арифметическое сложение кодов двух чисел. При арифметическом сложении выполняются и другие дополнительные операции: учёт знаков чисел, выравнивание порядков слагаемых и тому подобное. Указанные операции выполняются в арифметическо-логических устройствах (АЛУ) или процессорных элементах, ядром которых являются сумматоры.

Сумматоры классифицируют по различным признакам.

**В зависимости от системы счисления** различают:

- двоичные;
- двоично-десятичные (в общем случае двоично-кодированные);
- десятичные;
- прочие (например, амплитудные).

**По количеству одновременно обрабатываемых разрядов складываемых чисел:**

- одноразрядные,
- многоразрядные.

**По числу входов и выходов одноразрядных двоичных сумматоров:**

- четвертьсумматоры (элементы “сумма по модулю 2”; элементы “исключающее ИЛИ”), характеризующиеся наличием двух входов, на которые подаются два одноразрядных числа, и одним выходом, на котором реализуется их арифметическая сумма;
- полусумматоры, характеризующиеся наличием двух входов, на которые подаются одноимённые разряды двух чисел, и двух выходов: на одном реализуется арифметическая сумма в данном разряде, а на другом — перенос в следующий (более старший разряд);
- полные одноразрядные двоичные сумматоры, характеризующиеся наличием трёх входов, на которые подаются одноимённые разряды двух складываемых чисел и перенос из предыдущего (более младшего) разряда, и двумя выходами: на одном реализуется арифметическая сумма в данном разряде, а на другом — перенос в следующий (более старший разряд).

**По способу представления и обработки складываемых чисел** многоразрядные сумматоры подразделяются на:

- последовательные, в которых обработка чисел ведётся поочерёдно, разряд за разрядом на одном и том же оборудовании;
- параллельные, в которых слагаемые складываются одновременно по всем разрядам, и для каждого разряда имеется своё оборудование.

Параллельный сумматор в простейшем случае представляет собой  $n$  одноразрядных сумматоров, последовательно (от младших разрядов к старшим) соединённых цепями переноса. Однако такая схема сумматора характеризуется сравнительно невысоким быстродействием, так как формирование сигналов суммы и переноса в каждом  $i$ -ом разряде производится лишь после того, как поступит сигнал переноса с  $(i-1)$ -го разряда. **Таким образом, быстродействие сумматора определяется временем распространения сигнала по цепи переноса.**

**Уменьшение этого времени — основная задача при построении параллельных сумматоров.**

Для уменьшения времени распространения сигнала переноса применяют: **конструктивные решения**, когда используют в цепи переноса наиболее быстродействующие элементы; тщательно выполняют монтаж без длинных проводников и паразитных ёмкостных составляющих нагрузки и (наиболее часто) структурные методы ускорения прохождения сигнала переноса.

**По способу организации межразрядных переносов параллельные сумматоры, реализующие структурные методы**, делят на сумматоры:

- с последовательным переносом;
- с параллельным переносом;
- с групповой структурой;
- со специальной организацией цепей переноса.

Три первых структуры будут подробно рассмотрены в последующих статьях. **Среди сумматоров со специальной организацией цепей переноса можно указать:**



- сумматоры со сквозным переносом, в которых между входом и выходом переноса одноразрядного сумматора оказывается наименьшее число логических уровней [1];
- сумматоры с двухпроводной передачей сигналов переноса [1, 2];
- сумматоры с условным переносом (вариант сумматора с групповой структурой, позволяющий уменьшить время суммирования в 2 раза при увеличении оборудования в 1,5 раза) [3];
- асинхронные сумматоры, вырабатывающие признак завершения операции суммирования, при этом среднее время суммирования уменьшается, поскольку оно существенно меньше максимального.

Сумматоры, которые имеют постоянное время, отводимое для суммирования, независимое от значений слагаемых, называют **синхронными**.

По способу выполнения операции сложения и возможности сохранения результата сложения можно выделить три основных вида сумматоров:

- комбинационный, выполняющий микрооперацию “ $S = A$  плюс  $B$ ”, в котором результат выдаётся по мере его образования (это комбинационная схема в общепринятом смысле слова);
- сумматор с сохранением результата “ $S = A$  плюс  $B$ ”;
- накапливающий, выполняющий микрооперацию “ $S = S$  плюс  $B$ ”.

Последние две структуры строятся либо на счётных триггерах (сейчас практически не используются), либо по структуре “комбинационный сумматор – регистр хранения” (сейчас наиболее употребляемая схема).

**Важнейшими параметрами сумматоров** являются:

- разрядность;
- статические параметры:  $U_{вх}$ ,  $U_{вых}$ ,  $I_{вх}$  и так далее, то есть обычные параметры интегральных схем;
- динамические параметры. Сумматоры характеризуются четырьмя задержками распространения:
- от подачи входного переноса до установления всех выходов суммы при постоянном уровне на всех входах слагаемых;
- от одновременной подачи всех слагаемых до установления всех выходов суммы при постоянном уровне на входе переноса;
- от подачи входного переноса до установления выходного переноса при постоянном уровне на входах слагаемых;
- от подачи всех слагаемых до установления выходного переноса при постоянном уровне на входах слагаемых.

### Полный одноразрядный двоичный сумматор

Он (рис. 4) имеет три входа:  $a$ ,  $b$  — для двух слагаемых и  $p$  — для переноса из предыдущего (более младшего) разряда и два выхода:  $S$  — сумма,  $P$  — перенос в следующий (более старший) разряд. Обозначением полного двоичного сумматора служат буквы  $SM$ . Работу его отражает таблица истинности 3 (табл. 3).

Таблица 3

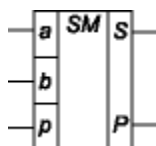


Рис. 4

№ наб.	a	b	p	P	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0

7	1	1	1	1	1
---	---	---	---	---	---

Отметим два момента. **Первый:** в табл. 2 и 3 выходные сигналы P и S не случайно расположены именно в такой последовательности. Это подчеркивает, что PS рассматривается как двухразрядное двоичное число, например,  $1 + 1 = 2_{10} = 10_2$ , то есть P = 1, а S = 0 или  $1 + 1 + 1 = 3_{10} = 11_2$ , то есть P = 1, а S = 1.

**Второй:** выходные сигналы P и S полного двоичного сумматора относятся к классу самодвойственных функций алгебры логики. **Самодвойственными** называют функции, инвертирующие своё значение при инвертировании всех переменных, от которых они зависят. Обратите внимание, что P и S для четвертьсумматора и полусумматора не являются самодвойственными функциями! Преимущества, вытекающие из этого свойства полного двоичного сумматора, будут рассмотрены при анализе возможностей ИС типа 155ИМ1.

Уравнения, описывающие работу полного двоичного сумматора, представленные в совершенной дизъюнктивной нормальной форме (СДНФ), имеют вид:

$$\left. \begin{aligned} S &= \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp \\ P &= \bar{a}bp + a\bar{b}p + ab\bar{p} + abp \end{aligned} \right\} \quad (6)$$

Уравнение для переноса может быть минимизировано:

$$P = ab + ap + bp. \quad (7)$$

При практическом проектировании сумматора уравнения (6) и (7) могут быть преобразованы к виду, удобному для реализации на заданных логических элементах с некоторыми ограничениями (по числу логических входов и др.) и удовлетворяющему предъявляемым к сумматору требованиям по быстродействию.

Например, преобразуем уравнения (6) следующим образом:

$$\left. \begin{aligned} S &= (\bar{a}b + a\bar{b})p + (\bar{a}\bar{b} + ab)p = S'p + \bar{S}'p \\ P &= ab + \bar{a}bp + a\bar{b}p = ab + S'p = P' + P'' \end{aligned} \right\} \quad (8)$$

Из выражений (8) следует, что полный двоичный сумматор может быть реализован на двух полусумматорах и одном двухвходовом элементе ИЛИ. Соответствующая схема приведена на рис. 5.

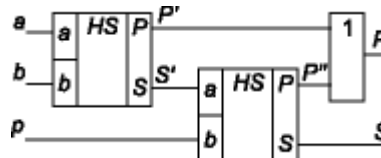


Рис. 5

Из выражения (8) для S также следует:

$$S = a \oplus b \oplus p. \quad (9)$$

К настоящему времени разработано большое число схем сумматоров. Доказано (нашим отечественным ученым Вайнштейном), что при использовании только одного инвертора нельзя реализовать полный двоичный сумматор со сложностью  $P_{KB} < 16$ , а при двух инверторах —  $P_{KB} < 14$ , где  $P_{KB}$  — вес по Квайну, используемый как оценка сложности любых комбинационных схем.  $P_{KB}$  — это общее число всех входов всех логических элементов схемы **без учёта инверторов**.

	$p$			
$a \backslash b$	x	0	x	1
	x	1	0	x
	x	0	x	1
	1	x	x	0
	$P$			

Рис. 6

Покажем, используя два метода, как была получена рациональная (с использованием только одного инвертора) схема полного двоичного сумматора, явившаяся основой схем ИС сумматоров типа 7480, 155ИМ1 и др.

**Первый метод** основан на использовании значения выходного переноса  $P$  как вспомогательной переменной при определении выходной суммы  $S$  (табл. 4). В табл. 4 при наборах переменных, являющихся нереальными (например, единичное значение переноса при нулевых значениях всех входных переменных), поставлены безразличные значения (крестик) для функции  $S$ , которые можно доопределять произвольным образом.

№ наб.	a	b	p	P	S
0	0	0	0	0	0
1	0	0	0	1	x
2	0	0	1	0	1
3	0	0	1	1	x
4	0	1	0	0	1
5	0	1	0	1	x
6	0	1	1	0	x
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	x
10	1	0	1	0	x
11	1	0	1	1	0
12	1	1	0	0	x
13	1	1	0	1	0
14	1	1	1	0	x
15	1	1	1	1	1

Из карты Карно для функции  $S$  (рис. 6) следует:  
 $S = abp + Pa + Pb + Pp = abp + P(a + b + p)$ . (10)

Схема сумматора, реализованного по уравнениям (7) и (10), приведена

на рис. 8а. В данной схеме используются многовходовые логические элементы И и ИЛИ. Если использовать только двухвходовые элементы, то получаются схемы, приведённые на рис. 8б,в.

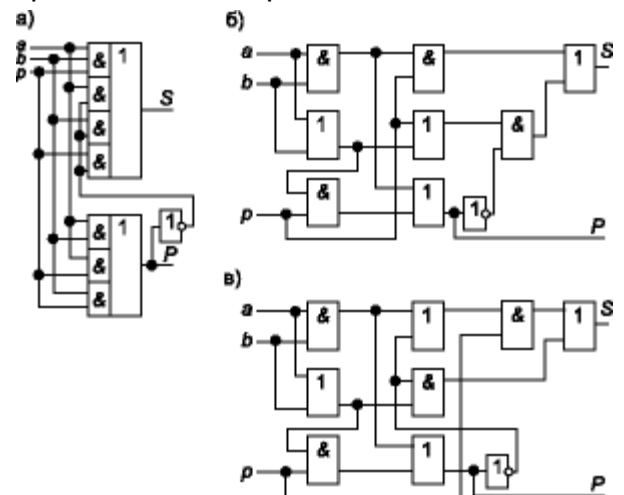


Рис. 8

19. Инкременторы и декременторы.

Пусть на вход поступает два 1-разрядных числа А и В. Сумма этих чисел будет представлена 2-разрядным кодом (см. табл. истинности).

A	B	CR	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

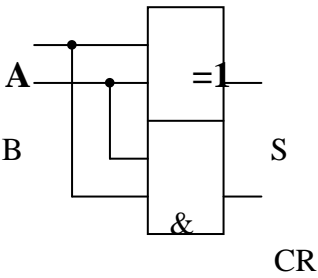
Младший разряд суммы – S (от англ. Sum – сумма).

Старший разряд – CR (от англ. Carry – перенос).

По табл. истинности легко построить схему (рис. 56).

$S=A\oplus B$

$CR=AB$



Полученная схема наз. **полусумматором**.

Введем следующие обозначения.  $A_i$  и  $B_i$  – значение i-го разряда многоразрядных чисел А и В.  $S_i$  – значение i-го разряда суммы,  $cr_i$  – вход переноса из (i-1)-го разряда в i-й.  $CR_i$  – выход переноса из i-го разряда. При этом:

$S_i = A_i \oplus B_i \oplus cr_i$

Из этой формулы ясно, что полусумматор нельзя использовать для построения сумматора любой разрядности, т. к. в нем отсутствует вход переноса  $cr_i$ .

Схема полусумматора имеет и самостоятельное значение (рис. 57).

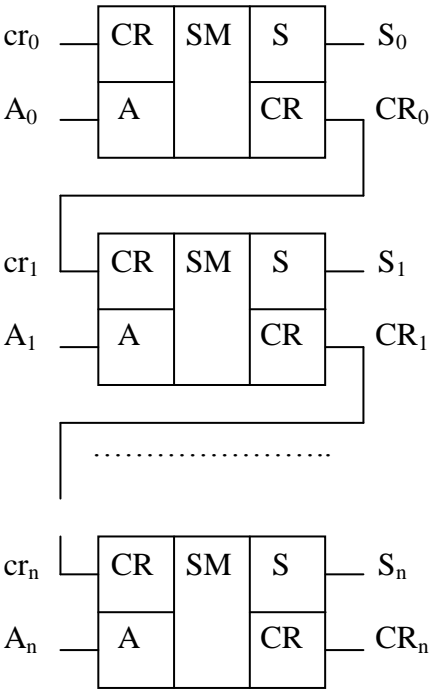


Рис. 57

Если заменить В на вход переноса  $CR_i$ , то на схему можно подать n-разрядный код А.

Если  $cr_0 = 0$ , то  $S_i = A_i$  и схема является повторителем входного числа А.

Если  $cr_0 = 1$ , то выходной код  $S = A + 1$  и схема называется **инкрементором**.

## 20. Матричные множительные устройства.

### Множительные и множительно-суммирующие блоки.

### Наращивание умножителей

#### Умножители

Умножение в двоичном виде производится подобно умножению в десятичной системе счисления. Как мы это помним из школьного курса, легче всего осуществлять умножение в столбик. При этом потребуется перемножить каждый разряд множимого на соответствующий разряд множителя.

Рассмотрим в качестве примера умножение двух четырёхразрядных двоичных чисел. Пусть требуется умножить число  $1011_2$  ( $11_{10}$ ) на число  $1101_2$  ( $13_{10}$ ). Мы в результате умножения ожидаем получить число  $10001111_2$  ( $143_{10}$ ). Выполним операцию умножения в столбик, как это показано на рисунке 1.

$$\begin{array}{r}
 1101 \\
 \times 1101 \\
 \hline
 1101 \\
 + 1101 \\
 + 0000 \\
 + 1101 \\
 \hline
 10001111
 \end{array}$$

Рисунок 1. Выполнение операции умножения в столбик.

Для формирования произведения требуется вычислить четыре частичных произведения. Обратите внимание, что в двоичной арифметике требуется выполнять умножение только на числа 0 и 1. Это означает, что нужно либо суммировать множимое к сумме остальных частичных произведений, либо нет. Таким образом, для формирования частичного произведения можно воспользоваться логическими элементами “2И”.

Для формирования частичного произведения, кроме операции умножения на один разряд, требуется осуществлять его сдвиг влево на число разрядов, соответствующее весу разряда множителя. Сдвиг можно осуществить простым соединением соответствующих разрядов частичных произведений к необходимым разрядам двоичного сумматора.

Для того чтобы принципиальная схема умножителя была похожа на алгоритм двоичного умножения, приведённый на рисунке 1, используем условно-графические изображения микросхем, где входы расположены сверху, а выходы снизу. В полном соответствии с алгоритмом умножения в столбик нам потребуются три четырёхразрядных сумматора.

Принципиальная схема умножителя, реализующая алгоритм двоичного умножения в столбик, приведена на рисунке 2. Формирование частичных произведений в этой схеме осуществляют микросхемы D1, D3, D5, D7. В этих микросхемах содержится сразу четыре логических элемента “2И”.

Сумматор, выполненный на микросхеме D6, суммирует первое и второе частные произведения. При этом младший разряд первого частного произведения не нуждается в суммировании (см. рисунок 1). Поэтому он подаётся на выход умножителя непосредственно (разряд M0). Второе частное произведение должно быть сдвинуто на один разряд. Это осуществляется тем, что младший разряд выходного числа сумматора D6 соединяется со вторым разрядом произведения (M1). Но тогда первое частное произведение необходимо сдвинуть на один разряд по отношению ко второму частному произведению! Это выполняется тем, что младший разряд группы входов A соединяется с первым разрядом частного произведения, первый разряд группы входов A соединяется со

вторым разрядом частного произведения, и т.д. Однако старший разряд группы входов А не с чем соединять! Вспомним, что если добавить к числу слева ноль, то значение числа не изменится, поэтому мы можем этот разряд соединить с общим проводом схемы.

Точно таким же образом осуществляется суммирование третьего и четвёртого частного произведения. Это суммирование выполняют микросхемы D4 и D2 соответственно. Отличие заключается только в том, что здесь не нужно задумываться о старшем разряде предыдущей суммы, ведь предыдущая микросхема сумматора формирует сигнал переноса.

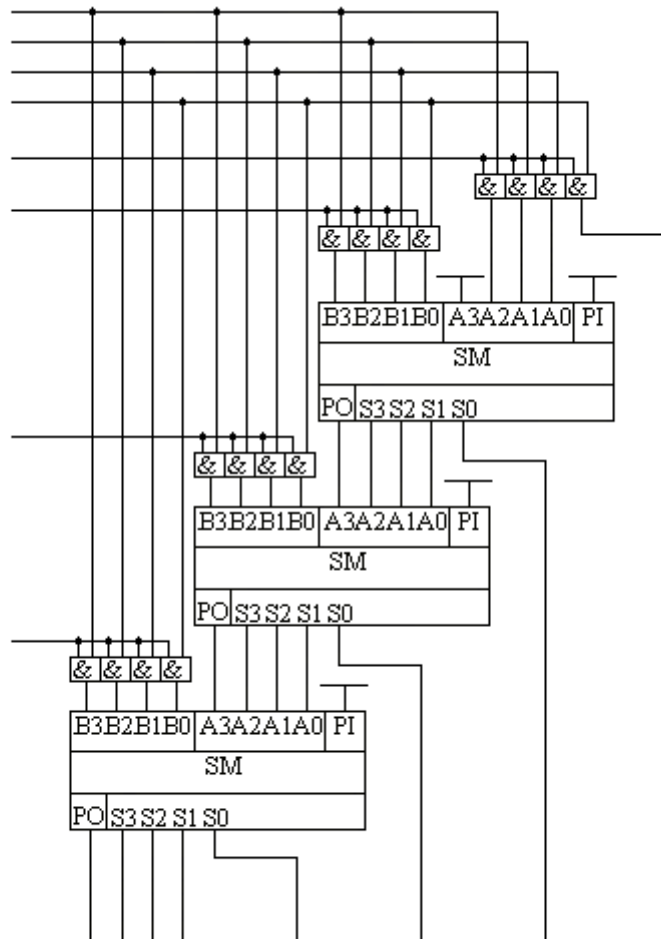


Рисунок 2. Схема матричного умножителя 4\*4.

Если внимательно посмотреть на схему умножителя, то можно увидеть, что она образует матрицу, сформированную проводниками, по которым передаются разряды числа А и числа В. В точках пересечения этих проводников находятся логические элементы “2И”. Именно по этой причине умножители, реализованные по данной схеме, получили название матричных умножителей.

Скорость работы схемы, приведенной на рисунке 2 определяется максимальным временем распространения сигнала. Это цепь D7, D6, D4, D2. Время работы схемы можно сократить, если сумматоры располагать не последовательно друг за другом, как это предполагается алгоритмом, приведенным на рис. 1, а суммировать частичные произведения попарно, затем суммировать пары частичных произведений и т.д. В этом случае время выполнения операции умножения значительно сократится.

Особенно заметен выигрыш в быстродействии при построении многоразрядных умножителей, однако ничего не бывает бесплатно. В обмен на быстродействие придётся заплатить увеличением разрядности сумматоров, а значит сложностью схемы. Если

сумматоры частных произведений останутся той же разрядности, что и ранее, то разрядность сумматоров пар частичных произведений должна быть увеличена на единицу. Разрядность сумматоров четвёрок частичных произведений будет на два разряда больше разрядности сумматоров частичных произведений и т.д