

СОДЕРЖАНИЕ

Введение.	4
Глава 1. Основы построения цифровых устройств.	4
1.1. Общие сведения о цифровых устройствах.	4
1.2. Логические элементы И, ИЛИ, НЕ и построение простейших функциональных схем на их основе.	6
1.3. Способы задания переключательных функций.	10
1.3.1. Составление формулы с помощью совершенной дизъюнктивной нормальной формы.	10
1.3.2. Составление формулы с помощью совершенной конъюнктивной нормальной формы.	11
1.4. Методы минимизации переключательных функций.	12
1.4.1. Алгебраический метод минимизации.	13
1.4.2. Метод диаграмм Вейча.	14
1.5. Реализация переключательных функций в базисе «И-НЕ» и «ИЛИ-НЕ».	18
1.5.1. Операция Шеффера (базис «И-НЕ»).	18
1.5.2. Операция Пирса (базис «ИЛИ-НЕ»).	20
1.5.3. Взаимосвязь совершенной конъюнктивной и дизъюнктивной нормальной формы.	21
1.5.4. Преобразователи произвольных кодов.	23
1.6. Технологическая реализация логических элементов.	25
Глава 2. Узлы цифровых устройств.	30
2.1. Цифровые узлы комбинационного типа.	30
2.1.1. Преобразователи кодов.	30
2.1.2. Коммутаторы.	32
2.1.3. Комбинационные сумматоры.	35
2.2. Цифровые узлы накапливающего типа.	38
2.2.1. Принцип работы триггеров и их виды.	38
2.2.2. Счетчики.	46
Задания для самостоятельного решения.	51
Список использованных сокращений.	56
Список литературы.	56

ВВЕДЕНИЕ

Известно, что современная электроника, особенно в настоящее время, базируется в основном на цифровых электронных элементах. Кроме этого, немаловажную роль также продолжает играть и аналоговая электроника, которая является основой усилительной техники. Открытие транзистора в середине прошлого века привело к существенной миниатюризации как аналоговых, так и цифровых электронных приборов. Это позволило значительно повысить эффективность и уменьшить габариты электронных компонентов. Совокупность этих двух направлений электронной техники позволяет создавать сложные электронные приборы, которые широко применяются в промышленности, науке и технике. На основе простейших цифровых элементов строятся такие цифровые элементы как триггеры, шифраторы, счетчики и т.д., являющиеся базовыми элементами современных электронных приборов. Поэтому изучение принципов работы электронных схем, а также способов построения цифровых и аналоговых устройств с заданными характеристиками является одним из важнейших и востребованных направлений современного физического образования.

Глава 1. ОСНОВЫ ПОСТРОЕНИЯ ЦИФРОВЫХ УСТРОЙСТВ

1.1. ОБЩИЕ СВЕДЕНИЯ О ЦИФРОВЫХ УСТРОЙСТВАХ

В цифровых устройствах (ЦУ) действуют цифровые сигналы. Обычные непрерывные синусоидальные электрические сигналы называют аналоговыми. Цифровые же сигналы представляют собой импульсы прямоугольной формы с определенной амплитудой максимума и определенным минимальным значением. Цифровой сигнал характеризуется изменением напряжения от одного уровня к другому. Эти уровни (максимум и минимум) могут иметь различное значение напряжения в вольтах в зависимости от серии элементов ЦУ. Причем принято максимальное значение напряжения $U_{\text{в}}$ называть «уровень 1», а минимальное $U_{\text{н}}$ - «уровень 0». Конкретные величины напряжений нуля и единицы зависят от серии цифровых элементов (микросхем и др.). Так, например, в серии микросхем К155 за уровень 1 принимается напряжение от 2,4 до 5,0 В, а за уровень 0 - от 0 до 0,4 В. Кроме этого различные серии цифровых микросхем отличаются между собой не только значениями напряжений уровней 0 и 1, но и соответствующими им значениям токов, временем задержки и др.

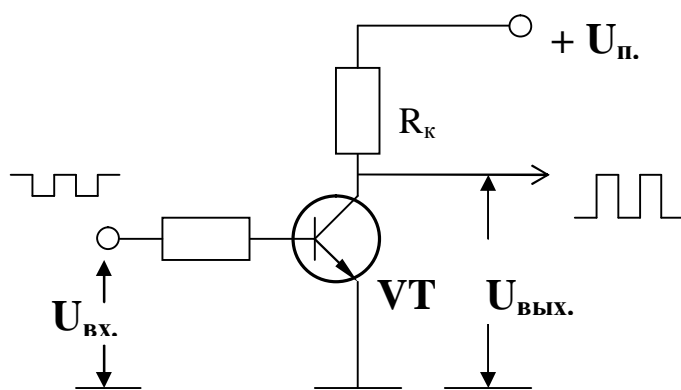


Рис. 1.1

вен 1», а минимальное $U_{\text{н}}$ - «уровень 0». Конкретные величины напряжений нуля и единицы зависят от серии цифровых элементов (микросхем и др.). Так, например, в серии микросхем К155 за уровень 1 принимается напряжение от 2,4 до 5,0 В, а за уровень 0 - от 0 до 0,4 В. Кроме этого различные серии цифровых микросхем отличаются между собой не только значениями напряжений

уровней 0 и 1, но и соответствующими им значениям токов, временем задержки и др.

При работе цифровых схем на транзисторах, микросхемах, как правило, не предъявляются жесткие требования к форме токов и напряжений. При использовании транзисторов в цифровых электронных схемах они работают чаще только в ключевом режиме (режим насыщения), т. е. или «открыты», или «закрыты». На рис. 1.1 приведена простейшая схема включения транзистора (схема с общим эмиттером) в ключевую цепь. При поступлении на базу транзистора V_T прямоугольных электрических сигналов транзистор, в соответствии с входными импульсами, поочередно открывается и закрывается, формируя на выходе практически прямоугольные импульсы, которые могут использоваться для дальнейшего управления схемой, исполнительными элементами и др. Если вместо R_k будет обмотка реле или обмотка соленоида, то при открывании транзистора сработает данный элемент. В ключевых схемах также не предъявляются жесткие требования и к амплитуде импульса. Важно, чтобы напряжение надежно перекрывало уровни 0 и 1 (но было не больше допустимого). В цифровых элементах необходимо, чтобы переход от одного состояния (открытый транзистор) к другому (закрытый) происходил как можно быстрее. Для этого промышленность выпускает переключающие транзисторы, у которых время переключения меньше, чем у обыкновенных усилительных. Частота переключения современных переключающихся транзисторов достигает десятки, а иногда и сотни мегагерц. В соответствии с кодировкой сигнала сравнительно просто записывать и считывать такой сигнал, который состоит только из нулей и единиц. При этом магнитная лента или магнитный диск или намагничены (1), или размагничены (0) (запись программ для компьютеров). При этом не предъявляются жесткие условия и к степени их намагниченности, т. е. к качеству, как при записи аналоговых звуковых сигналов.

Цифровой сигнал в устройствах управления, программных устройствах может быть представлен в закодированном (зашифрованном) виде, т. е. набором нулей и единиц. Такой код называют двоичным. Например, «близкие» нам десятичные числа (до 15) могут быть представлены в двоичном коде при четырех разрядах:

A	0	1	2	3	4	5	6	7	8	9	10	...	15
B	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	...	1111

Здесь A – десятичное число, B – соответствующий двоичный код.

При представлении чисел кодом идет заполнение разрядов справа налево.

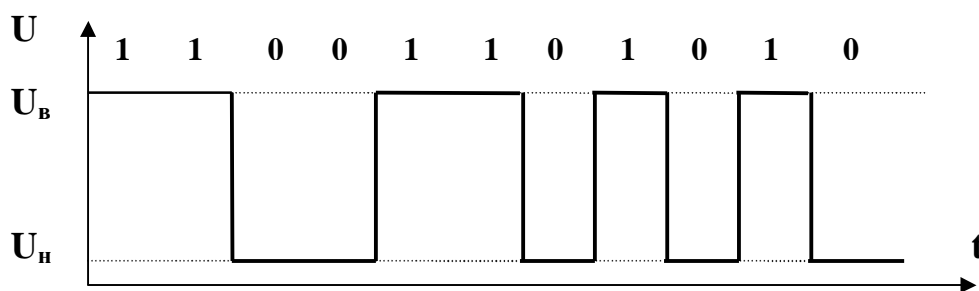


Рис. 1.2

Когда все четыре разряда заполнятся, что соответствует цифре 15, цифру 16

уже представить четырьмя разрядами нельзя, так как в этом случае необходимо больше разрядов.

Временная диаграмма цифрового сигнала представляет собой последовательность импульсов в координатах $U(t)$. Считается, что фронтальная и тыльная стороны импульса (перепад уровней $U_H \rightarrow U_B$ или $U_B \rightarrow U_H$) имеют бесконечно малое время. Например, цифровой электрический сигнал, соответствующий последовательности 11001101010, можно представить в виде временной диаграммы электрических импульсов $U(t)$, изображенной на рис. 1.2. Здесь уровню логической 1 соответствует напряжение U_B , а логическому 0 – напряжение U_H .

1.2. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ «И», «ИЛИ», «НЕ» И ПОСТРОЕНИЕ ПРОСТЕЙШИХ ФУНКЦИОНАЛЬНЫХ СХЕМ НА ИХ ОСНОВЕ

Любое ЦУ состоит из логических элементов (ЛЭ), среди которых можно выделить три основных, называемых «И», «ИЛИ», «НЕ». Каждый из этих элементов выполняет вполне определенную логическую функцию. Для описания работы ЛЭ применяют булеву алгебру (БА).

БА представляет собой математический аппарат формального описания процессов в элементах и узлах ЭВМ. Оперирует БА с логическими переменными (операндами). Логическая переменная может принимать одно из двух возможных значений «Да», «Нет» (истина, ложь). В применении к описанию ЦУ значение булевой переменной – 1 или 0.

На множестве $\{0, 1\}$ в БА определены 3 логические операции: «НЕ», «И», «ИЛИ». Каждая из этих операций задается таблицей, в которой установлено взаимно однозначное соответствие между значениями логической переменной и результатом операции.

Логическая операция «НЕ» (отрицание, инверсия) – одноместная операция,

x	Y
0	1
1	0

которая задается таблицей 1.1. Здесь x – операнд, Y – результат операции. Для обозначения операции «НЕ» используют символ « $\bar{}$ », расположенный над операндом. Также используют символ: « \neg », который располагают перед определенным операндом. Например, по табл. 1.1 можно записать: $Y = \bar{x}$ или $Y = \neg x$.

Таблица 1.1

Таким образом, логическая операция «НЕ» принимает значения, противоположные значению операнда.

Логическая операция «И» (логическое умножение, конъюнкция) – многоместная операция, т.е. для представления этой функции необходимо не менее 2-х операндов. Операция «И» задается табл. 1.2. Как

x_1	x_0	Y
0	0	0
0	1	0
1	0	0
1	1	1

видно из представленной таблицы, $Y = 1$ только тогда, когда и $x_0 = 1$ и $x_1 = 1$. Аналогично, данная операция определяется и для $(n + 1)$ операндов: $Y = 1$ если все операнды $x_0 \dots x_n = 1$. Здесь $(n + 1)$ – разрядность машинного слова (двоичного кода). Если, например, $n = 3$, то разрядность кода равна четырем. Тогда $Y = 1$, если $x_0 = 1, x_1 = 1, x_2 = 1, x_3 = 1$.

Таблица 1.2

Как видно, нумерация операндов начинается с нуля, причем отсчет производится справа налево. Например, для 6-разрядного кода запись выглядит следующим образом: $x_5 x_4 x_3 x_2 x_1 x_0$.

Для обозначения операции «И» применяют либо символ « \cdot », либо « \wedge ». Согласно табл. 1.2 операцию «И» в терминах БА для двух операндов можно записать в виде: $Y = x_1 \cdot x_0$ или $Y = x_1 \wedge x_0$. Операция «И» над $(n + 1)$ операндами: $Y = x_n \cdot x_{n-1} \cdot x_{n-2} \cdot \dots \cdot x_0$.

Логическая операция «ИЛИ» (логическое сложение, дизъюнкция) – многоместная операция (как и операция «И»). Операция «ИЛИ» задается табл. 1.3. Как видно из таблицы, $Y = 1$, если хотя бы один из операндов равен единице, т.е. или $x_0 = 1$, или $x_1 = 1$. Аналогично, данная операция определяется и для $(n + 1)$ операндов: $Y=1$ если хотя бы один из них равен единице. Если $n=2$, то разрядность кода равна трем. Тогда $Y=1$, если, например, $x_0 = 1, x_1 = 0, x_2 = 1$.

x_1	x_0	Y
0	0	0
0	1	1
1	0	1
1	1	1

Для обозначения операции «ИЛИ» применяют символ « \vee ». Согласно табл. 1.3 операцию «ИЛИ» для двух операндов можно записать в виде: $Y = x_1 \vee x_0$. Операция «ИЛИ» над $(n + 1)$ операндами: $Y = x_n \vee x_{n-1} \vee x_{n-2} \vee \dots \vee x_0$.

Поскольку результат у любой из логических операций принимает значения из того же множества $\{0, 1\}$, что и операнды $x_n, x_{n-1}, x_{n-2}, \dots, x_0$ можно составлять суперпозиции логических операций, когда результат одной операции используется в качестве операнда в другой.

$\bar{0} = 1, \bar{1} = 0$	(1.1)
$A \vee 0 = A, A \cdot 1 = A$	(1.2)
$A \vee 1 = 1, A \cdot 0 = 0$	(1.3)
$A \vee A = A, A \cdot A = A$	(1.4)
$\overline{(\bar{A})} = A$	(1.5)
$A \cdot \bar{A} = 0$	(1.6)
$A \vee B = B \vee A, A \cdot B = B \cdot A$	(1.7)
$A \vee B \vee C = A \vee (B \vee C)$ $A \cdot B \cdot C = A \cdot (B \cdot C)$	(1.8)
$A \cdot (B \vee C) = A \cdot B \vee A \cdot C$ $A \vee (B \cdot C) = (A \vee B) \cdot (A \vee C)$	(1.9)
$A \vee (A \cdot B) = A$ $A \cdot (A \vee B) = A$	(1.10)
$(A \cdot B) \vee (A \cdot \bar{B}) = A$ $(A \vee B) \cdot (A \vee \bar{B}) = A$	(1.11)
$(A \vee B) = \bar{\bar{A}} \cdot \bar{\bar{B}}$ $(A \cdot B) = \bar{A} \vee \bar{B}$	(1.12)

Например, можно записать для некоторой функции следующую формулу $Y = ((x_2 \cdot \bar{x}_0) \vee x_1 \vee (x_1 \vee x_2)) \cdot \bar{x}_2$.

Подобные суперпозиции логических операций являются переключательными функциями (ПФ). ПФ – это такая функция $Y = f(x_n, x_{n-1}, x_{n-2}, \dots, x_0)$, в которой $Y, x_n, x_{n-1}, x_{n-2}, \dots, x_0$ могут принимать только значения 0 или 1 (суперпозиция логических операций). БА основана на системе тождеств, которые даны в табл. 1.4.

Здесь A, B, C – либо логические переменные, либо логические функции. С помощью приведенных тождеств можно производить различные эквивалентные преобразования булевых выражений.

Таблица 1.4

Например, по правилу (1.11) можно записать:

$$(x_1 \cdot x_0) \vee (x_1 \cdot \bar{x}_0) = x_1 \quad \text{или}$$

$$(x_2 \cdot x_1 \cdot x_0) \vee (x_2 \cdot \bar{x}_1 \cdot x_0) = x_2 \cdot x_0$$

В соответствии с выполняемыми операциями указанные соотношения имеют следующие названия: (1.5) - закон двойного отрицания; (1.7) – переместительный закон; (1.8) – сочетательный закон; (1.9) – распределительный закон; (1.10) – закон поглощения; (1.11) – закон склеивания; (1.12) – правило Де Моргана (закон инверсии).

Одним из основных конструктивных документов, разрабатываемых в процессе технического проектирования ЦУ, является его функциональная схема (ФС). ФС – графический документ, на котором своими графическими обозначениями представлены элементы и узлы устройства, указаны все связи между входами и выходами этих узлов и устройств.

Условное графическое обозначение (УГО) для элемента или узла цифрового устройства – это прямоугольник, внутри которого указывается функция этого элемента (кроме зарубежного обозначения). ЦУ, работа которого описывается средствами БА называется комбинационной схемой (КС). Простейшими из КС являются ЛЭ – устройства, реализующие логические операции по правилам БА.

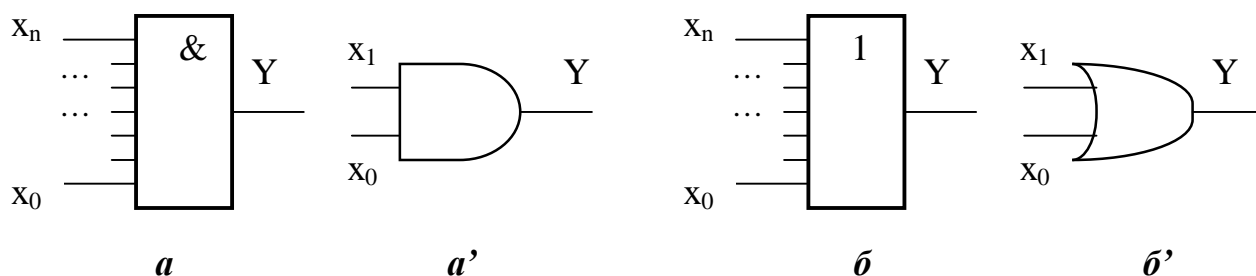


Рис. 1.3. Условное графическое обозначение конъюнктора (логический элемент «И») – *a*, *a'* (зарубежное обозначение) и дизъюнктора (элемент «ИЛИ») – *б*, *б'* (зарубежное обозначение).

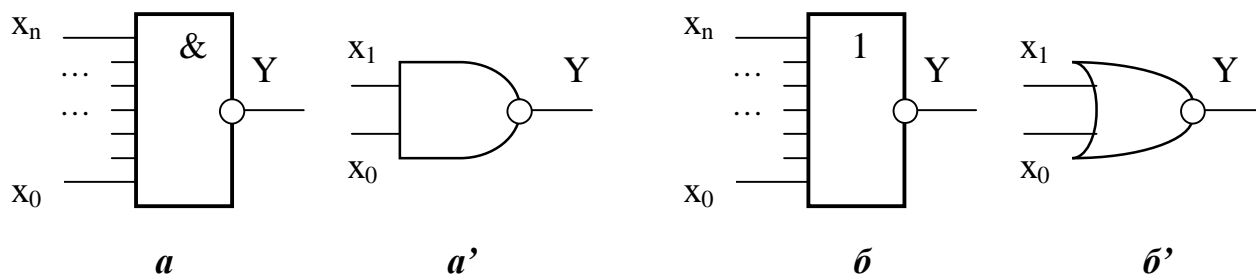


Рис. 1.4. Условное графическое обозначение логического элемента «И-НЕ» – *a*, *a'* (зарубежное обозначение) и элемента «ИЛИ-НЕ» – *б*, *б'* (зарубежное обозначение).

УГО конъюнктора и дизъюнктора показаны на рис. 1.3 (отечественное и зарубежное обозначения). Кроме этого, входы и выходы этих ЛЭ могут иметь

инверсию, которая обозначается в виде кружка, установленного на определенном входе и/или выходе. Элементы, у которых выход является инвертированным, называются элементами «И-НЕ» и «ИЛИ-НЕ» (рис. 1.4).

Таким образом, можно записать формулы, описывающие рассмотренные выше ЛЭ:

$$\begin{aligned} \text{«И»} : Y &= (x_1 \cdot x_0) & \text{«И-НЕ»} : Y &= (x_1 \cdot x_0) \\ \text{«ИЛИ»} : Y &= (x_1 \vee x_0) & \text{«ИЛИ-НЕ»} : Y &= (x_1 \vee x_0) \end{aligned}$$

Пользуясь описанными выше средствами, нетрудно построить ФС для следующей формулы: $Y = (\bar{x}_1 \cdot x_0)$



А для формулы $Y = (x_1 \cdot x_0) \vee (x_2 \cdot x_0) \vee (\bar{x}_2 \cdot \bar{x}_1)$ ФС выглядит так, как это представлено на рис. 1.5.

И наоборот, имея ФС можно записать формулу, которую эта схема реализует (рис. 1.6):

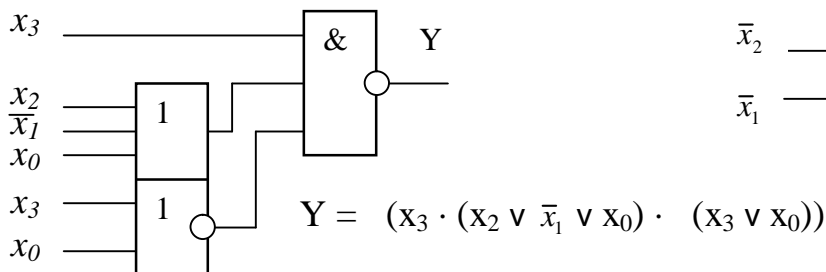
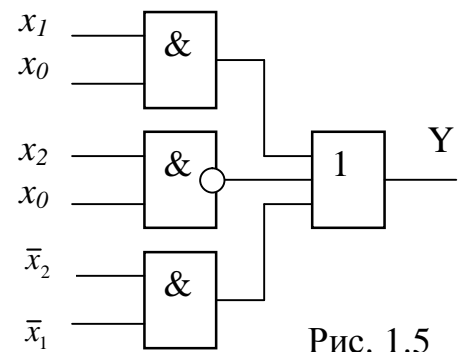
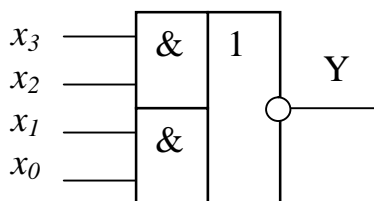


Рис. 1.6

Иногда, отдельный элемент может включать в себя сразу несколько ЛЭ, объединенных в одну общую ФС. Таковым, например, является элемент «И-ИЛИ-НЕ». В данном элементе последовательно, согласно записи, производятся определенные логические операции. УГО этого ЛЭ и соответствующая формула, приведены на рис. 1.7.



$$\text{«И-ИЛИ-НЕ»} : Y = ((x_1 \cdot x_0) \vee (x_2 \cdot x_3))$$

Рис. 1.7

Часто, при записи ЛЭ, такого как, например представленного на рис. 1.7, указывают число входных элементов. Например, 2И-ИЛИ-НЕ (представлен на рис. 1.7), 3И-ИЛИ-НЕ, 4И-ИЛИ-НЕ и т.д.

Иногда, при наличии 2-х входных элементов (2 логических входа), необходимым является построение схем для 3-х, 4-х входов. В этом случае элемент расширяют, т.е. определенным способом соединяют однотипные цифровые элементы. Например, в случае 2-х входного ЛЭ «ИЛИ» его можно расширить, т.е. увеличить количество входов, следующим образом: $Y = (x_2 \vee x_1) \vee x_0$ - для 3-х операндов; $Y = (x_3 \vee x_2) \vee (x_1 \vee x_0)$ - для 4-х операндов и т.д.

1.3. СПОСОБЫ ЗАДАНИЯ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ

Самым простым и наглядным способом задания ПФ является табличный способ, когда составляется таблица с перечислением всех наборов аргументов $x_0 \dots x_n$, с указанием для каждого из них значения функции Y . Такая таблица называется таблицей истинности (ТИ) ПФ. Для удобства в ТИ переменные нумеруют справа налево. Составим, к примеру, ТИ для ПФ $Y = f(x_2, x_1, x_0)$, которая равна 1 в случае, если хотя бы два из трех аргументов равны логическому 0 (табл. 1.5). Для удобства в крайнем левом столбце обычно записывают десятичный эквивалент двоичной записи $x_2 x_1 x_0 - \alpha$. Приведенную ТИ можно представить также и в следующем виде:

α	x_2	x_1	x_0	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

Таблица 1.5

ментов $x_0 \dots x_n$, с указанием для каждого из них значения функции Y . Такая таблица называется таблицей истинности (ТИ) ПФ. Для удобства в ТИ переменные нумеруют справа налево. Составим, к примеру, ТИ для ПФ $Y = f(x_2, x_1, x_0)$, которая равна 1 в случае, если хотя бы два из трех аргументов равны логическому 0 (табл. 1.5). Для удобства в крайнем левом столбце обычно записывают десятичный эквивалент двоичной записи $x_2 x_1 x_0 - \alpha$. Приведенную ТИ можно представить также и в следующем виде:

α	0	1	2	3	4	5	6	7
Y	1	1	1	0	1	0	0	0

В БА функции задаются аналитически – формулами, где операнды соединены знаками алгебраических операций.

Составим формулу, описывающую рассмотренную функцию.

Для составления формулы ПФ существуют 2 основных метода:

- 1 – Совершенная дизъюнктивная нормальная форма (СДНФ);
- 2 – Совершенная конъюнктивная нормальная форма (СКНФ).

1.3.1. Составление формулы с помощью СДНФ

Для рассмотрения этого метода, сначала необходимо ввести понятие так называемой *элементарной конъюнкции* (минтерма) ES_i :

$$ES_i = x_n \cdot x_{n-1} \cdot \dots \cdot x_0 \quad (1.13)$$

При этом $x_k = x$, если на k -ом наборе $Y = 1$. Если же на k -ом наборе $Y = 0$, то $x_k = \bar{x}$. Здесь $k = \{n, n-1, \dots, 0\}$.

Таким образом, минтерм EC_i берется, если результат Y , на какой либо комбинации равен 1. Поэтому методику получения аналитического выражения по элементарной конъюнкции также называют составлением формулы по единицам.

Из табл. 1.5 видно, что $Y = 1$ при $\alpha = 0, 1, 2, 4$. Соответственно, можно составить следующие минтермы:

$$\begin{aligned} EC_0 &= \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0 & EC_1 &= \bar{x}_2 \cdot \bar{x}_1 \cdot x_0 \\ EC_2 &= \bar{x}_2 \cdot x_1 \cdot \bar{x}_0 & EC_4 &= x_2 \cdot \bar{x}_1 \cdot \bar{x}_0 \end{aligned}$$

СДНФ состоит из дизъюнкции всех минтермов, т.е. окончательное выражение:

$$\begin{aligned} Y &= EC_0 \vee EC_1 \vee EC_2 \vee EC_4 = \\ &= (\bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot x_0) \vee (\bar{x}_2 \cdot x_1 \cdot \bar{x}_0) \vee (x_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \end{aligned} \quad (1.14)$$

Таким образом, полученная формула полностью описывает приведенную табл. 1.5.

1.3.2. Составление формулы с помощью СКНФ

Для рассмотрения этого метода, также сначала необходимо ввести понятие так называемой *элементарной дизъюнкции* (макстерма) ED_i :

$$ED_i = x_n \vee x_{n-1} \vee \dots \vee x_0 \quad (1.15)$$

При этом $x_k = x$, если на k -ом наборе $Y = 0$. Если же на k -ом наборе $Y = 1$, то $x_k = \bar{x}$; $k = \{ n, n - 1, \dots, 0 \}$.

Таким образом, макстерм ED_i берется, если результат Y , на какой либо комбинации равен 0. Поэтому методика получения выражения по элементарной дизъюнкции называется составлением формулы по нулям.

Из табл. 1.5 видно, что $Y = 0$ при $\alpha = 3, 5, 6, 7$. Соответственно, можно составить следующие макстермы:

$$ED_3 = x_2 \vee \bar{x}_1 \vee \bar{x}_0$$

$$ED_5 = \bar{x}_2 \vee x_1 \vee \bar{x}_0$$

$$ED_6 = \bar{x}_2 \vee \bar{x}_1 \vee x_0$$

$$ED_7 = \bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_0$$

СКНФ состоит из конъюнкции всех макстермов, и окончательное выражение:

$$\begin{aligned} Y &= ED_0 \cdot ED_1 \cdot ED_2 \cdot ED_4 = (x_2 \vee \bar{x}_1 \vee \bar{x}_0) \cdot (\bar{x}_2 \vee x_1 \vee \bar{x}_0) \cdot \\ &\cdot (\bar{x}_2 \vee \bar{x}_1 \vee x_0) \cdot (\bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_0) \end{aligned} \quad (1.16)$$

Таким образом, ранее полученная формула (1.14), также как и (1.16), точно описывают табл. 1.5. Поэтому как с помощью СДНФ, так и с помощью СКНФ можно составлять аналитические выражения. Однако в целях минимизации получаемых выражений, существует критерий выбора метода, поскольку минимальную формулу легче анализировать и представлять в технической реализации.

В случае, если количество $Y = 1$ в ТИ меньше, чем $Y = 0$, то целесообразно выбрать метод СДНФ. И наоборот, если количество $Y = 0$ в ТИ меньше, чем $Y = 1$, то целесообразно выбрать метод СКНФ. Из табл. 1.5 видно, что количество $Y = 1$ и $Y = 0$ одинаково, т.е. равно четырем. Поэтому в данном случае можно выбрать любой из методов, либо СДНФ, либо СКНФ.

α	x_2	x_1	x_0	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Таблица 1.6

Проиллюстрируем вышесказанное следующим примером. Пусть дана ТИ для ПФ (табл. 1.6) в которой количество $Y = 1$ и $Y = 0$ не одинаково. Тогда, применив метод СДНФ, получим:

$$Y = (\bar{x}_2 \cdot x_1 \cdot \bar{x}_0) \vee (x_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (x_2 \cdot x_1 \cdot \bar{x}_0) \quad (1.17)$$

Теперь применим метод СКНФ. Получим:

$$Y = (x_2 \vee x_1 \vee x_0) \cdot (x_2 \vee x_1 \vee \bar{x}_0) \cdot (x_2 \vee \bar{x}_1 \vee \bar{x}_0) \cdot (\bar{x}_2 \vee x_1 \vee \bar{x}_0) \cdot (\bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_0) \quad (1.18)$$

Формулы (1.17) и (1.18) одинаково описывают табл. 1.6. Однако видно, что выражение (1.17) является более компактным, чем формула (1.18), т.е. (1.17) – содержит всего 3 минтерма, в то время как (1.18) – 5 макстермов.

Таким образом, для построения минимальной формулы, необходимо выбрать соответствующий метод составления.

Для определения компактности формулы часто используют оценивание стоимости формулы. Стоимость оценивается по шкале Квайна в относительных единицах сложности ПФ. Операнд под прямым знаком (без инверсии) оценивается в 1 единицу, а с инверсией – в 2 единицы. При определении стоимости формулы сначала подсчитывается количество операндов во всей формуле (с учетом инверсии), а затем общее количество минтермов или макстермов. Например, стоимость формулы (1.17) – 17 единиц, а формулы (1.18) – 28 единиц.

1.4. МЕТОДЫ МИНИМИЗАЦИИ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ

Любая ПФ может быть выражена как суперпозиция двоичных переменных, взятых в конечном числе. Естественно, чем меньшее число логических переменных входит в рассматриваемую формулу, описывающую заданную функцию, тем проще произвести ее анализ и реализовать ее на практике, составив ее ФС. Для этого применяют различные методы минимизации ПФ.

1.4.1. Алгебраический метод минимизации

Алгебраический метод минимизации ПФ основан на применении к рассматриваемой формуле определенных тождеств БА, т.е. полученную формулу в некоторых случаях можно привести к еще более компактному виду – минимизировать (уменьшить стоимость формулы). Рассмотрим данный метод на следующем примере. Из полученных выражений (1.14) и (1.16) видно, что они имеют одинаковую стоимость. Для наглядности перепишем формулу (1.14):

$$Y = (\bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot x_0) \vee (\bar{x}_2 \cdot x_1 \cdot \bar{x}_0) \vee (x_2 \cdot \bar{x}_1 \cdot \bar{x}_0)$$

Для минимизации этого выражения необходимо применить тождества БА (1.4) и (1.11) (см. табл. 1.4):

$$A \vee A \vee A = A \quad \text{и} \quad (A \cdot B \cdot C) \vee (A \cdot \bar{B} \cdot C) = (A \cdot C)$$

По тождеству (1.4) преобразуем формулу (1.13) в следующую эквивалентную формулу:

$$Y = (\bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot x_0) \vee (\bar{x}_2 \cdot x_1 \cdot \bar{x}_0) \vee (x_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0)$$

Такое преобразование применяется из-за того, что 2-ой, 3-ий и 4-ый минтермы формулы (1.14) последовательно склеиваются с 1-ым по различным переменным, и для получения минимальной формулы необходимо увеличить количество минтермов, аналогичных 1-ому в формуле (1.14). После этого применим тождество (11). Получим:

$$Y = (\bar{x}_2 \cdot \bar{x}_1) \vee (\bar{x}_2 \cdot \bar{x}_0) \vee (\bar{x}_1 \cdot \bar{x}_0) \quad (1.19)$$

Отсюда видно, что формула (1.19) получилась значительно меньше, чем (1.14). При этом они одинаково описывают одну и ту же функцию (табл. 1.5).

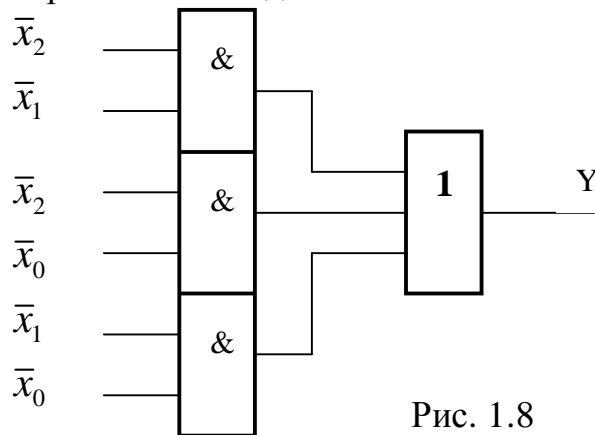


Рис. 1.8

Зная формулу, описывающую некоторую функцию, легко можно составить ее ФС. Например, для формулы (1.19) она представлена на рис. 1.8.

Достоинством данного метода минимизации является то, что с помощью этого метода, можно производить минимизацию аналитических выражений с любой разрядностью двоичных слов, входящих в рассматриваемую формулу.

1.4.2. Метод диаграмм Вейча

Наиболее удобным и эффективным способом минимизации ПФ для двоичных слов длиной $(n+1) = 2, 3, 4$ является применение метода диаграмм Вейча (ДВ).

ДВ представляет собой прямоугольник на N клеток, где $N = 2^{(n+1)}$, $(n+1)$ – длина слова. Каждая клетка ставится в соответствие одному из N наборов аргументов ПФ. Данный метод основан на том, что соседние минтермы (макстермы), расположенные в определенных клетках диаграммы, всегда склеиваются, уменьшая при этом количество аргументов ПФ.

Вид ДВ для слов различной длины представлены на рис. 1.9.

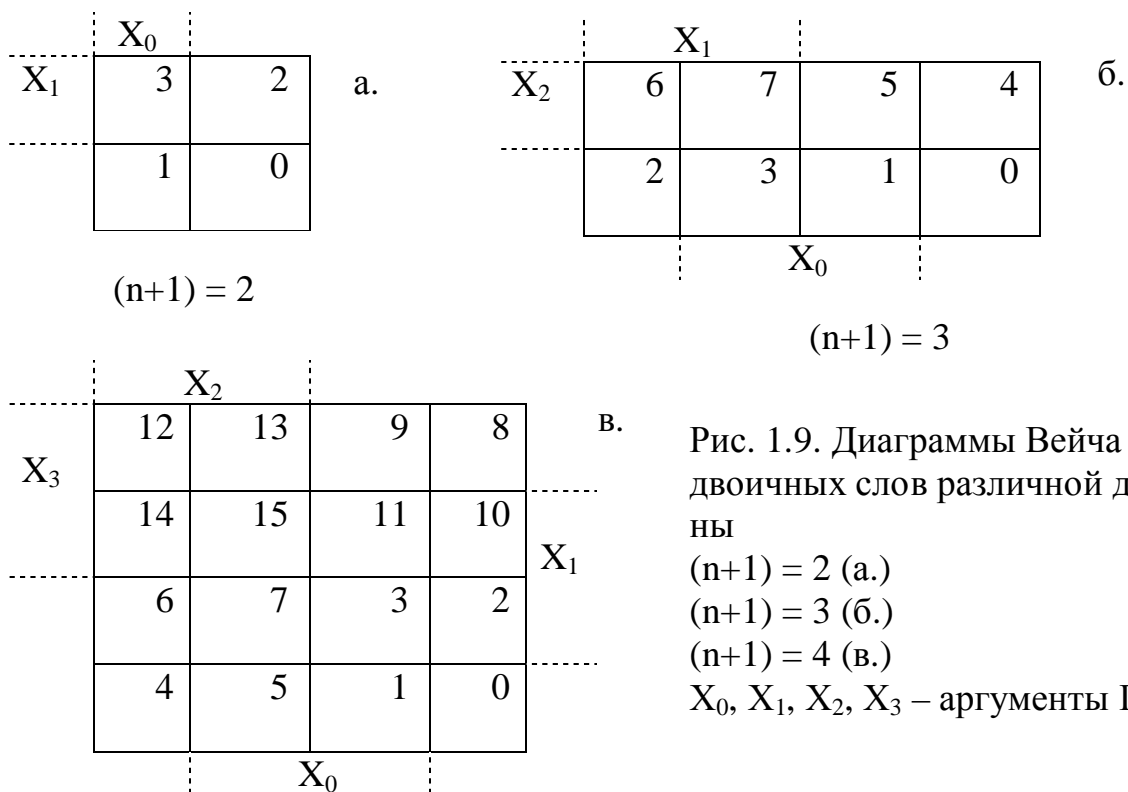


Рис. 1.9. Диаграммы Вейча для двоичных слов различной длины
 $(n+1) = 2$ (а.)
 $(n+1) = 3$ (б.)
 $(n+1) = 4$ (в.)
 X_0, X_1, X_2, X_3 – аргументы ПФ.

На границах ДВ указываются операнды, которые отсекают на ДВ либо столбец, либо строку. Это указывает на то, что отсекаемый набор каким либо операндом, в своем наборе содержит единицы. Например, для ДВ $(n+1) = 3$ (рис. 1.9, б) операнд X_2 отсекает строку, содержащую наборы с номерами 6, 7, 5, 4. Сравнив, табл. 1.6 для $(n+1) = 3$ и ДВ для $(n+1) = 3$ обнаружим, что операнд $X_2 = 1$ лишь на указанных наборах. Аналогично определяется содержание единиц и для других случаев. Необходимо отметить, что такая запись производится лишь для удобства и при решении задач с использованием ДВ запись операндов по строкам и столбцам не обязательна.

Заполняется ДВ следующим образом: если на наборе α_i ПФ $Y = 1$, то в клетку с номером i записывают 1. Если же на наборе α_i ПФ $Y = 0$, то клетку с номером i оставляют пустой. Указанное правило справедливо в случае применения метода СДНФ (составление формулы по единицам).

В случае применения метода СКНФ (составление формулы по нулям), ДВ заполняется следующим образом: если на наборе α_i ПФ $Y = 0$, то в клетку с номером i записывают 0. Если же на наборе α_i ПФ $Y = 1$, то клетку с номером i оставляют пустой.

После заполнения ДВ необходимо выделить склеивающиеся минтермы (макстермы). Указанный процесс производят заключением в овал группы соседних единиц (нулей), причем объединяются значения, расположенные в ряд и/или в столбик. При этом единицы (нули) должны идти друг за другом. Если существует разрыв, то объединять их в общий овал нельзя, и их следует объединять по отдельности, в зависимости от склеивающихся элементов. Это можно проверять по ГИ, рассматриваемой ПФ. Заполненные клетки, расположенные по диагонали - не склеиваются. Пример заполнения ДВ для $(n+1) = 2$, и объединение соседних минтермов, представлено на рис. 1.10.

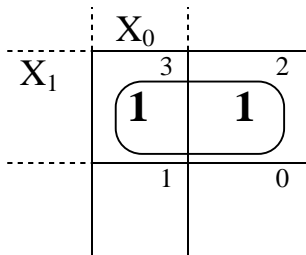


Рис. 1.10

Иногда, при заполнении ДВ, заполненные клетки ни с чем не склеиваются. В этом случае, данный минтерм (макстерм) записывается в формуле полностью.

В общем случае, конечная формула включает все те минтермы (макстермы), которые заключены в овалы. Все овалы, присутствующие в ДВ входят в конечную формулу.

Нередко, расположение единиц (нулей) на ДВ допускает их объединение в группы различными способами. Пусть даны ДВ (рис. 1.11), которые заполнены указанным образом.

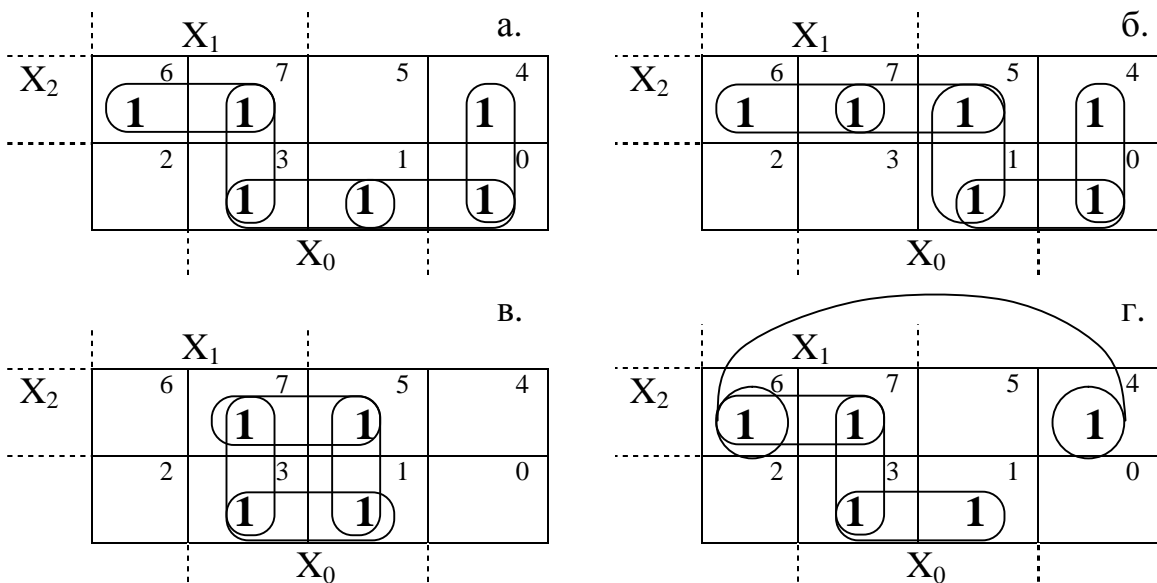


Рис. 1.11

Особенность объединения минтермов (макстермов) в ДВ с получением минимальной формулы заключается в том, что чередующиеся таким образом овалы можно брать не все, а лишь их часть. Начальный овал, рекомендуется выбирать таким образом, чтобы он находился в начале или в конце цепочки овалов.

Из этой ДВ (рис. 1.11, а.) можно взять овалы 6-7, 3-1, 0-4. Овалы между ними окажутся лишними.

Из этой ДВ (рис. 1.11, б.) можно взять овалы 6-7, 5-1, 0-4. Овалы между ними также окажутся лишними.

Из этой ДВ (рис. 1.11, в.) можно взять овалы 7-3 и 5-1. Также можно взять 7-5 и 3-1. В данном случае овалы формируют замкнутую цепочку, и поэтому существует 2 варианта составления формулы.

ДВ, представленная на рис. 1.11, (г), соответствует более сложному типу заполнения. В данном случае, как следует из рассмотренных выше примеров, можно взять овалы 6-7 и 3-1 (цепочка овалов). Однако существует заполнение и в клетке 4. Поскольку клетка 4 находится в одном ряду с клеткой 6, то клетка 4 склеивается с данной клеткой. Поэтому на ДВ клетки 4 и 6 объединены. Клетки 4 и 7 склеиваться не будут. Особенность ДВ заключается в том, что отдельно могут склеиваться элементы, находящиеся на периферии ДВ, т.е. самые крайние клетки, находящиеся в одном ряду или в одном столбце ДВ. Таким образом, конечная формула для данной ПФ будет включать склеенные минтермы 6-7, 3-1 и 6-4. При составлении формулы по нулям (СКНФ) действуют те же правила работы с ДВ.

Рассмотрим пример составления формулы и ФС с помощью ДВ.

Пример 1.1. Составить формулу и построить ФС для цифрового автомата, ТИ которого представлена в следующем виде:

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y	1	1	0	1	0	1	1	1	0	0	0	1	1	1	0	1

Решение:

α	x_3	x_2	x_1	x_0	Y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Таблица 1.7

1. Составление ТИ и выбор метода составления формулы – СКНФ или СДНФ.

ТИ с записью операндов $x_3 - x_0$ выглядит так, как это представлено в табл. 1.7. Подсчитаем количество нулей и единиц в колонке Y. Так как количество единиц превышает количество нулей, то метод составления формулы в данном случае будет СКНФ – составление формулы по нулям.

2. Заполнение ДВ.

Так как $(n+1) = 4$, то ДВ, необходимая для составления формулы будет выглядеть так, как это представлено на рис. 1.9 (случай $(n+1) = 4$).

В клетки ДВ (рис. 1.12) под номерами 2, 4, 8, 9, 10, 14 записываются нули, после чего производится их объединение.

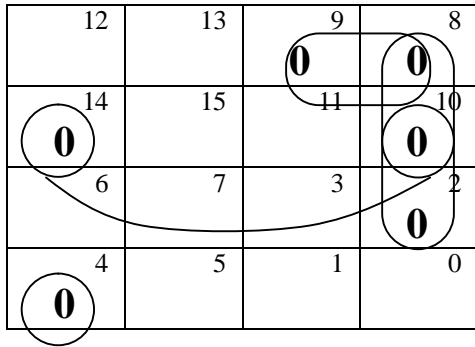


Рис.1.12

Овалы, которые будут входить в конечную формулу следующие: 9-8, 10-2, 14-10 и 4. Овал 8-10 оказывается лишним и поэтому в формулу не входит. Клетка 4 не склеивается с клеткой 14 и поэтому записывается отдельно и полностью.

3. Запись конечной формулы.

Для записи конечной формулы необходимо рассмотреть ТИ данной функции. В ТИ, согласно склеивающимся макстермам, определяются те компоненты формулы, по которым производится склеивание. Например, из ДВ видно, что клетки 9 и 8 склеиваются. Рассмотрев комбинации 9 и 8 в ТИ можно обнаружить, что единственное отличие их записи в двоичной форме заключается в операнде x_0 . Остальные операнды x_3, x_2, x_1 для этих двух слов по своим значениям не отличаются друг от друга. Таким образом, склеивание производится по x_0 и в конечной формуле эта компонента не присутствует. Поэтому для овала 9-8 можно записать:

$$ED_{9-8} = \bar{x}_3 \vee x_2 \vee x_1$$

Соответственно, для овалов 10-2, 14-10 и 4 будем иметь:

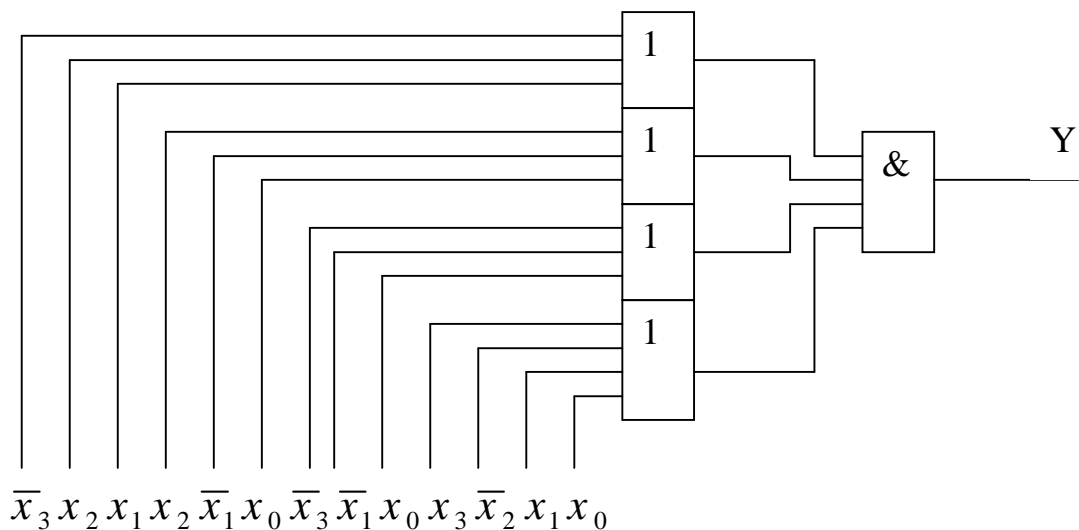
$$ED_{10-2} = x_2 \vee \bar{x}_1 \vee x_0 \quad ED_{14-10} = \bar{x}_3 \vee \bar{x}_1 \vee x_0 \quad ED_4 = x_3 \vee \bar{x}_2 \vee x_1 \vee x_0$$

Общий вид конечной формулы следующий:

$$Y = ED_{9-8} \cdot ED_{10-2} \cdot ED_{14-10} \cdot ED_4$$

$$Y = (\bar{x}_3 \vee x_2 \vee x_1) \cdot (x_2 \vee \bar{x}_1 \vee x_0) \cdot (\bar{x}_3 \vee \bar{x}_1 \vee x_0) \cdot (x_3 \vee \bar{x}_2 \vee x_1 \vee x_0) \quad (1.20)$$

4. Построение ФС ПФ.



1.5. РЕАЛИЗАЦИЯ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ В БАЗИСЕ «И-НЕ» И «ИЛИ-НЕ»

Любая ПФ может быть выражена как суперпозиция двоичных переменных, взятых в конечном числе. Естественно, чем меньшее число ЛЭ входит в данную схему, описывающую заданную функцию, тем проще реализовать ее на практике. Для этого применяют различные методы минимизации ПФ, которые были рассмотрены в предыдущем параграфе.

Набор ЛЭ, необходимый и достаточный для построения любой функции, называют функционально полным набором или базисом. Набор логических функций «И», «ИЛИ», «НЕ», суперпозиция которых позволяет записать любую ПФ, достаточен для того, чтобы синтезировать любую КС.

Однако, набор «И», «ИЛИ», «НЕ» на самом деле избыточен. Из него можно исключить один из ЛЭ – либо «И», либо «ИЛИ». Набор из оставшихся двух ЛЭ и будет функционально полным. Например, по правилу Де Моргана операцию «И» можно заменить эквивалентной суперпозицией операций «ИЛИ» и «НЕ»:

$$A \cdot B = \overline{(\overline{A} \vee \overline{B})}$$

а операцию «ИЛИ» заменить следующей суперпозицией операций «И» и «НЕ»:

$$A \vee B = \overline{(\overline{A} \cdot \overline{B})}$$

Таким образом, из 3 - х ЛЭ можно образовать 2 базиса ЛЭ: «И-НЕ» и «ИЛИ-НЕ».

1.5.1. Операция Шеффера

x_1	x_0	Y
0	0	0
0	1	0
1	0	0
1	1	1

Таблица 1.8

ЛЭ «И-НЕ» выполняет над логическими переменными операцию, которая называется функцией Шеффера. Данная операция – многоместная. Для двух аргументов она задается табл. 1.8.

Обозначают операцию Шеффера символом « $|$ », который также называют штрихом Шеффера. Операцию Шеффера над двумя аргументами задают записью: $Y = x_1 | x_0$, а над $(n+1)$ переменными: $Y = x_n | x_{n-1} | \dots | x_0$

Операция Шеффера средствами БА представляется в виде:

$$A | B = \overline{A \cdot B}$$

где A и B – логические переменные.

На рис. 1.13 представлено УГО ЛЭ Шеффера для $(n+1)$ аргументов.

Рассмотрим представление логических операций для двух операндов A и B в базисе «И-НЕ».

А). Логическая операция «НЕ»: $Y = \overline{A} = (A \cdot A) = (A | A)$

ФС операции инверсии представлена на рис. 1.14. Как видно, входы ЛЭ соединяются между собой. В случае, если количество входов ЛЭ более двух, то для формирования операции «НЕ» поступают аналогичным образом.

Б). Логическая операция «И»: $Y = A \cdot B = \overline{(\overline{A} | \overline{B})}$

ФС формулы представлена на рис. 1.15.

В). Логическая операция «ИЛИ»: $Y = A \vee B = (A \vee B) = (\bar{A} \cdot \bar{B}) = (\bar{A} | \bar{B})$

ФС формулы представлена на рис. 1.16.

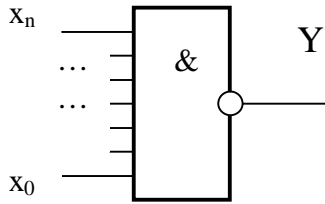


Рис.1.13

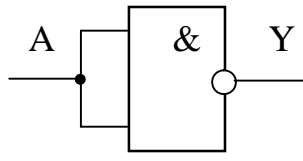


Рис. 1.14

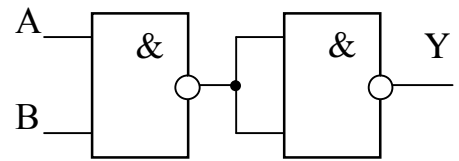


Рис. 1.15

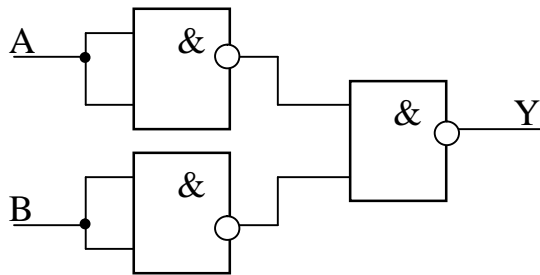


Рис.1.16

Как видно ФС (рис. 1.15, 1.16), составленные на основе ЛЭ Шеффера, получаются несколько громоздкими по сравнению с исходными функциями (например, см. рис. 1.3). Однако эти ФС требуют для своего составления только единственный вид ЛЭ – ЛЭ «И-НЕ».

В базисе «И-НЕ» справедливы следующие тождества:

$$A | B = B | A ; \quad A | B | C = (A | B) | C \quad (1.21)$$

Последнее тождество позволяет свести трехместную операцию Шеффера к последовательности двухместных.

Сформулируем правило перехода от записи в базисе «И», «ИЛИ», «НЕ» к ее записи в базисе «И-НЕ»:

1. Получить СДНФ для заданной функции и заключить в скобки их конъюнктивные члены.
2. Каждый символ конъюнкции (в скобках) и каждый символ дизъюнкции (между скобок) заменить на штрих Шеффера.
3. Если ПФ содержит только одну скобку с конъюнкциями, то оно переходит в базис «И-НЕ» со знаком инверсии.

Пример 1.2: Дана функция $Y = (x_2 \cdot x_1) \vee (x_3 \cdot x_2 \cdot x_0)$. Построить ФС в базисе «И-НЕ».

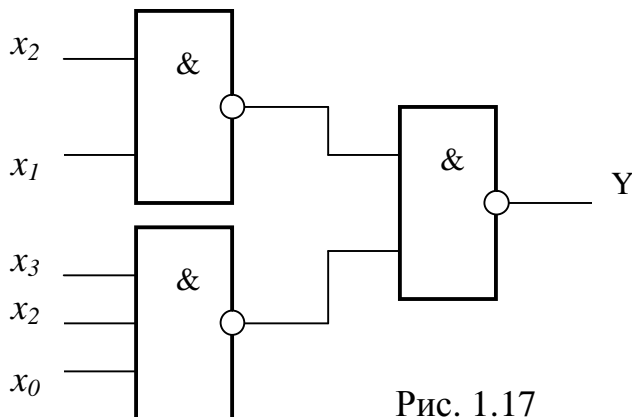


Рис. 1.17

Решение: Заменяем в формуле каждый знак конъюнкции и дизъюнкции на штрих Шеффера. Получим:

$$Y = (x_2 | x_1) | (x_3 | x_2 | x_0)$$

На рис. 1.17 представлена ФС указанной ПФ.

1.5.2. Операция Пирса

ЛЭ «ИЛИ-НЕ» выполняет над логическими переменными операцию, которая называется функцией Пирса. Данная операция, как и вышерассмотренная – многоместная. Для двух аргументов она задается табл. 1.9.

x_1	x_0	Y
0	0	1
0	1	0
1	0	0
1	1	0

Таблица 1.9

Обозначают операцию Шеффера символом « \uparrow », который также называют стрелкой Пирса. Операцию Пирса над двумя аргументами задают записью: $Y = x_1 \uparrow x_0$, а над $(n+1)$ переменными: $Y = x_n \uparrow x_{n-1} \uparrow \dots \uparrow x_0$

Операция Пирса средствами БА представляется в виде:

$$A \uparrow B = (A \vee B).$$

где A и B – логические переменные.

На рис. 1.18 представлено УГО ЛЭ Пирса для $(n+1)$ аргументов.

Рассмотрим представление логических операций для двух операндов A и B в базисе «ИЛИ-НЕ».

А). Логическая операция «НЕ»: $Y = \bar{A} = (A \vee A) = (A \uparrow A)$

ФС операции инверсии представлена на рис. 1.19. Как и в предыдущем случае входы ЛЭ соединяются между собой. Если количество входов ЛЭ более двух, то для создания операции «НЕ» их также соединяют между собой.

Б). Логическая операция «И»: $Y = A \cdot B = \overline{(A \vee B)} = (\bar{A} \vee \bar{B}) = (\bar{A} \uparrow \bar{B})$

ФС формулы представлена на рис. 1.20.

В). Логическая операция «ИЛИ»: $Y = A \vee B = (A \uparrow B)$

ФС формулы представлена на рис. 1.21.

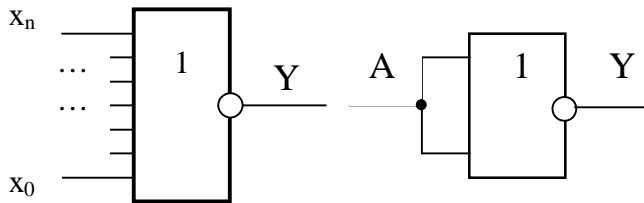


Рис.1.18

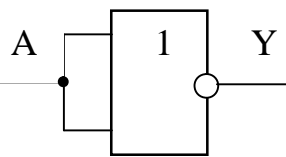


Рис. 1.19

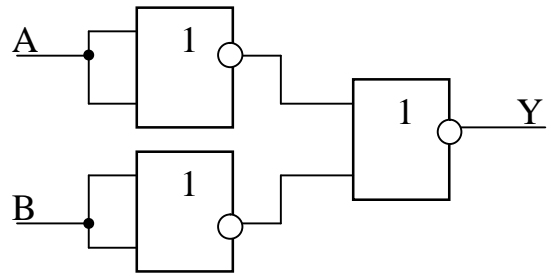


Рис. 1.20

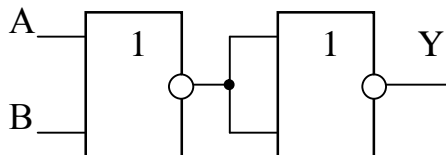


Рис.1.21

Как и в предыдущем случае ФС, составленные на ЛЭ Пирса также получают достаточно громоздкими по сравнению с исходными функциями (например, см. рис. 1.3). Но при этом ФС требуют для своего составления только единственный вид ЛЭ – ЛЭ «ИЛИ-НЕ».

В базисе «ИЛИ-НЕ» справедливы следующие тождества:

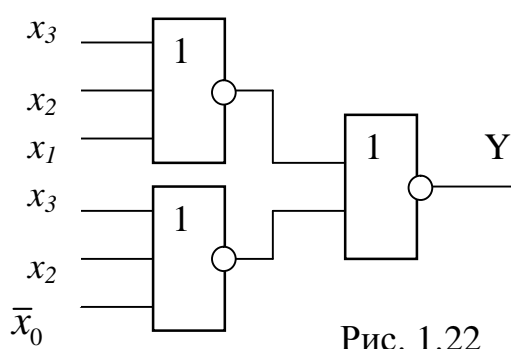
$$A \uparrow B = B \uparrow A; \quad A \uparrow B \uparrow C = (A \uparrow B) \uparrow C \quad (1.22)$$

Эти тождества также позволяют свести трехместную операцию Пирса к последовательности двухместных.

Сформулируем правило перехода от записи в базисе «И», «ИЛИ», «НЕ» к ее записи в базисе «ИЛИ-НЕ»:

1. Получить СКНФ для заданной функции и заключить в скобки их дизъюнктивные члены.
2. Каждый символ дизъюнкции (в скобках) и каждый символ конъюнкции (между скобок) заменить на стрелку Пирса.
3. Если ПФ содержит только одну скобку с дизъюнкциями, то оно переходит в базис «ИЛИ-НЕ» со знаком инверсии.

Пример 1.3: Дана функция $Y = (x_3 \vee x_2 \vee x_1) \cdot (x_3 \vee x_2 \vee \bar{x}_0)$. Построить ФС в базисе «ИЛИ-НЕ».



Решение: Заменяем в формуле каждый знак конъюнкции и дизъюнкции на стрелку Пирса. Получим следующую КС:

$$Y = (x_3 \uparrow x_2 \uparrow x_1) \uparrow (x_3 \uparrow x_2 \uparrow \bar{x}_0)$$

На рис. 1.22 представлена ФС полученной КС.

1.5.3. Взаимосвязь совершенной конъюнктивной и дизъюнктивной нормальной формы

В ряде случаев, необходимым является использование в схеме только определенных ЛЭ, например, только ЛЭ Шеффера. Однако КС для заданной ПФ в минимальной форме (минимальной стоимости) составляется только с помощью СКНФ, т.е. для построения схемы необходимы ЛЭ Пирса. В этом случае необходимо применить правило перехода от СКНФ к СДНФ:

$$Y_{\text{СДНФ}} = (\bar{Y}_{\text{СКНФ}}) \text{ и наоборот } Y_{\text{СКНФ}} = (\bar{Y}_{\text{СДНФ}}) \quad (1.23)$$

После преобразования от одной формы записи к другой схемы строятся по указанным выше правилам.

Пример 1.4. Дана следующая функция:

$$Y = (\bar{x}_3 \vee \bar{x}_2 \vee x_1) \cdot (x_3 \vee \bar{x}_2 \vee \bar{x}_0) \cdot (\bar{x}_3 \vee x_0)$$

Построить ФС данной функции с использованием ЛЭ «И-НЕ».

Решение. Для построения ФС данной ПФ необходим базис «ИЛИ-НЕ», т.к. эта функция построена с использованием метода СКНФ. Однако, по условию задачи, требуется построение этой функции в базисе «И-НЕ». Тогда необходимо произвести перевод формулы по правилу (1.23). Получим

$$Y = ((x_3 \cdot x_2 \cdot \bar{x}_1) \vee (\bar{x}_3 \cdot x_2 \cdot x_0) \vee (x_3 \cdot \bar{x}_0)) \quad (1.24)$$

Заменим каждый знак конъюнкции, и каждый знак дизъюнкции в (1.24) на штрих Шеффера. Получим

$$Y = ((x_3 | x_2 | \bar{x}_1) | (\bar{x}_3 | x_2 | x_0) | (x_3 | \bar{x}_0)) \quad (1.25)$$

Используя полученную формулу можно построить ФС в базисе «И-НЕ». Общй знак инверсии, стоящий перед скобками формулы (1.25), означает включение в ФС инвертора (ЛЭ «НЕ»), который может быть составлен на основе ЛЭ Шеффера или ЛЭ Пирса. При этом все входы ЛЭ объединены в общую точку. ФС ПФ (1.25) представлена на рис. 1.23.

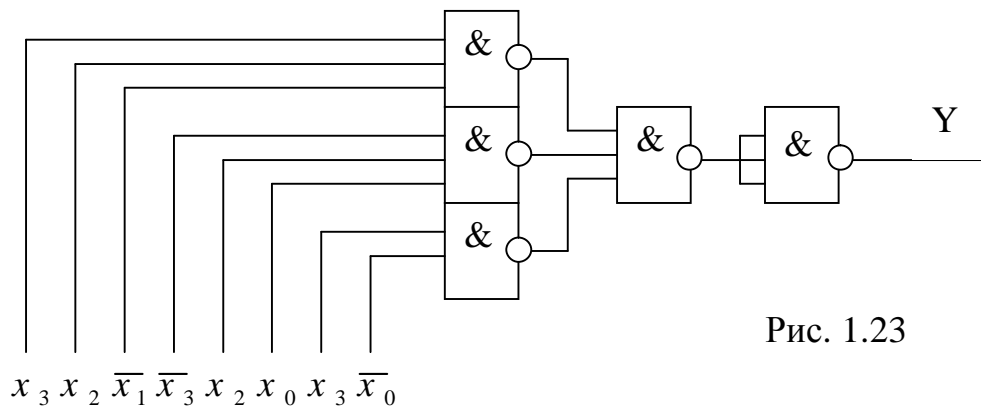


Рис. 1.23

Пример 1.5. Построить ФС, ПФ которого дается формулой (1.20).

$$Y = (\bar{x}_3 \vee x_2 \vee x_1) \cdot (x_2 \vee \bar{x}_1 \vee x_0) \cdot (\bar{x}_3 \vee \bar{x}_1 \vee x_0) \cdot (x_3 \vee \bar{x}_2 \vee x_1 \vee x_0)$$

При построении использовать базис «3И-НЕ» (3-х входовые ЛЭ Шеффера).

Решение:

1. Выбор базиса построения ФС.

Рассматриваемую ПФ необходимо перевести в базис «И-НЕ». Для этого используем правило (1.23). Получим:

$$Y = ((x_3 \cdot \bar{x}_2 \cdot \bar{x}_1) \vee (\bar{x}_2 \cdot x_1 \cdot \bar{x}_0) \vee (x_3 \cdot x_1 \cdot \bar{x}_0) \vee (\bar{x}_3 \cdot x_2 \cdot \bar{x}_1 \cdot \bar{x}_0))$$

После замены знаков конъюнкции и дизъюнкции на штрих Шеффера будем иметь:

$$Y = ((x_3 | \bar{x}_2 | \bar{x}_1) | (\bar{x}_2 | x_1 | \bar{x}_0) | (x_3 | x_1 | \bar{x}_0) | (\bar{x}_3 | x_2 | \bar{x}_1 | \bar{x}_0))$$

2. Разбиение формулы на необходимое количество входов ЛЭ.

Без преобразования полученной формулы, для построения ее ФС, необходимы как минимум 4-х входовые ЛЭ «И-НЕ». Однако требуется использование в ФС 3-х входовых ЛЭ «И-НЕ». Поэтому необходимо разбить формулу на последовательность 3-х местных операций. Применим правило (1.21). Тогда

$$Y = (((x_3 | \bar{x}_2 | \bar{x}_1) | (\bar{x}_2 | x_1 | \bar{x}_0)) | (x_3 | x_1 | \bar{x}_0) | ((\bar{x}_3 | x_2) | \bar{x}_1 | \bar{x}_0))$$

3. Построение ФС ПФ. (рис. 1.24)

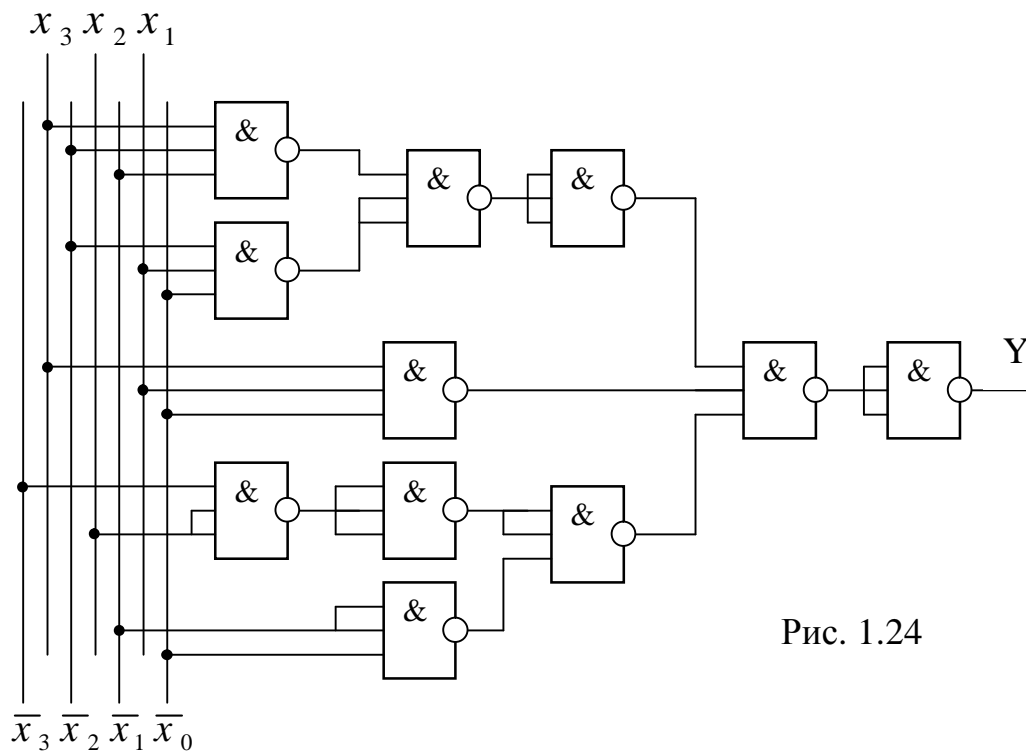


Рис. 1.24

1.5.4. Преобразователи произвольных кодов

В общем случае преобразователь кода - КС, которая выполняет трансформацию входного $(n + 1)$ -разрядного кода, построенного по одному закону, в выходной $(m + 1)$ -разрядный код, построенный по другому закону. Трансформацию кода $X = x_n x_{n-1} \dots x_0$ в код $Y = y_m y_{m-1} \dots y_0$ всегда можно описать таблицей, в которой каждому значению $x_n x_{n-1} \dots x_0$ ставится в соответствие значение $y_m y_{m-1} \dots y_0$. Входы этой таблицы можно рассматривать как перечень наборов логических переменных $x_n x_{n-1} \dots x_0$, а каждую цифру y_i выходного кода $y_m y_{m-1} \dots y_0$ как значение ПФ $y_i(x_n, x_{n-1}, \dots, x_0)$. С помощью этой таблицы можно синтезировать КС, свою для каждого y_i . ФС преобразователя кодов строится по рассмотренным выше правилам. Однако в этом случае ФС оказывается достаточно громоздкой, объем которой в общем случае зависит от разрядности входного X и выходного Y кодов.

Пример 1.6. Построить ФС преобразователя произвольных кодов (код X в код Y) в соответствии с заданной таблицей преобразования:

$\alpha 1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\alpha 2$	2	5	1	0	0	14	3	4	6	7	8	9	11	10	15	2

В таблице $\alpha 1$ – набор входного кода X , $\alpha 2$ – набор выходного кода Y . При построении ФС использовать а). Базис «И-НЕ» (ЛЭ Шеффера); б). Базис «ИЛИ-НЕ» (ЛЭ Пирса).

Решение:

1. Построение ТИ для системы ПФ, задающих закон преобразования.

$\alpha 1$	x_3	x_2	x_1	x_0	$\alpha 2$	y_3	y_2	y_1	y_0
0	0	0	0	0	2	0	0	1	0
1	0	0	0	1	5	0	1	0	1
2	0	0	1	0	1	0	0	0	1
3	0	0	1	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	1	14	1	1	1	0
6	0	1	1	0	3	0	0	1	1
7	0	1	1	1	4	0	1	0	0
8	1	0	0	0	6	0	1	1	0
9	1	0	0	1	7	0	1	1	1
10	1	0	1	0	8	1	0	0	0
11	1	0	1	1	9	1	0	0	1
12	1	1	0	0	11	1	0	1	1
13	1	1	0	1	10	1	0	1	0
14	1	1	1	0	15	1	1	1	1
15	1	1	1	1	2	0	0	1	0

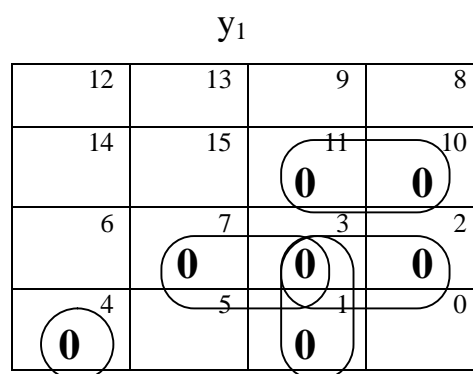
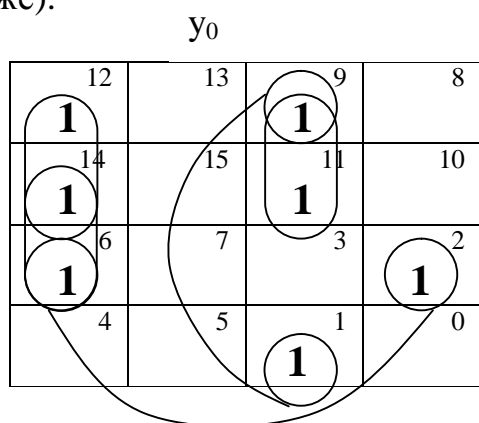
2. Выбор метода построения ПФ.

Подсчитаем количество «0» и «1» в каждой колонке y_i , $i = 0,1,2,3$ и выберем соответствующий метод построения ПФ.

Колонка y_0 – метод СДНФ; колонка y_1 – метод СКНФ; колонка y_2 – метод СДНФ; колонка y_3 – метод СДНФ;

3. Выполнение минимизации для каждой из ПФ y_i .

Для минимизации ПФ воспользуемся методом ДВ (ДВ для y_0 и y_1 приведены ниже).



Имеем следующий результат:

$$y_0 = (x_3 \cdot x_2 \cdot \bar{x}_0) \vee (x_2 \cdot x_1 \cdot \bar{x}_0) \vee (\bar{x}_3 \cdot x_1 \cdot \bar{x}_0) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot x_0) \vee (x_3 \cdot \bar{x}_2 \cdot x_0)$$

$$y_1 = (x_3 \vee x_2 \vee \bar{x}_0) \cdot (x_3 \vee \bar{x}_1 \vee \bar{x}_0) \cdot (x_2 \vee \bar{x}_1) \cdot (x_3 \vee \bar{x}_2 \vee x_1 \vee x_0)$$

Аналогично, составив ДВ для y_2 и y_3 , получим:

$$y_2 = (\bar{x}_3 \cdot \bar{x}_1 \cdot x_0) \vee (\bar{x}_3 \cdot x_2 \cdot x_0) \vee (x_3 \cdot \bar{x}_2 \cdot \bar{x}_1) \vee (\bar{x}_2 \cdot \bar{x}_1 \cdot x_0) \vee (x_3 \cdot x_2 \cdot x_1 \cdot \bar{x}_0)$$

$$y_3 = (x_3 \cdot \bar{x}_2 \cdot x_1) \vee (x_3 \cdot x_1 \cdot \bar{x}_0) \vee (x_3 \cdot x_2 \cdot \bar{x}_0) \vee (x_3 \cdot x_2 \cdot \bar{x}_1) \vee (x_2 \cdot \bar{x}_1 \cdot x_0)$$

4. Перевод полученных ПФ к требуемому базису.

I). Преобразуем полученные ПФ к базису И-НЕ. Имеем результат:

$$y_0 = (x_3 | x_2 | \bar{x}_0) | (x_2 | x_1 | \bar{x}_0) | (\bar{x}_3 | x_1 | \bar{x}_0) | (\bar{x}_2 | \bar{x}_1 | x_0) | (x_3 | \bar{x}_2 | x_0)$$

$$y_1 = ((\bar{x}_3 | \bar{x}_2 | x_0) | (\bar{x}_3 | x_1 | x_0) | (\bar{x}_2 | x_1) | (\bar{x}_3 | x_2 | \bar{x}_1 | \bar{x}_0))$$

$$y_2 = (\bar{x}_3 | \bar{x}_1 | x_0) | (\bar{x}_3 | x_2 | x_0) | (x_3 | \bar{x}_2 | \bar{x}_1) | (\bar{x}_2 | \bar{x}_1 | x_0) | (x_3 | x_2 | x_1 | \bar{x}_0)$$

$$y_3 = (x_3 | \bar{x}_2 | x_1) | (x_3 | x_1 | \bar{x}_0) | (x_3 | x_2 | \bar{x}_0) | (x_3 | x_2 | \bar{x}_1) | (x_2 | \bar{x}_1 | x_0)$$

II). Преобразуем полученные ПФ к базису «ИЛИ-НЕ». Имеем результат:

$$y_0 = ((\bar{x}_3 \uparrow \bar{x}_2 \uparrow x_0) \uparrow (\bar{x}_2 \uparrow \bar{x}_1 \uparrow x_0) \uparrow (\bar{x}_3 \uparrow x_1 \uparrow \bar{x}_0) \uparrow (x_2 \uparrow x_1 \uparrow \bar{x}_0) \uparrow (\bar{x}_3 \uparrow x_2 \uparrow \bar{x}_0))$$

$$y_1 = (x_3 \uparrow x_2 \uparrow \bar{x}_0) \uparrow (x_3 \uparrow \bar{x}_1 \uparrow \bar{x}_0) \uparrow (x_2 \uparrow \bar{x}_1) \uparrow (x_3 \uparrow \bar{x}_2 \uparrow x_1 \uparrow x_0)$$

$$y_2 = ((x_3 \uparrow x_1 \uparrow \bar{x}_0) \uparrow (x_3 \uparrow \bar{x}_2 \uparrow \bar{x}_0) \uparrow (\bar{x}_3 \uparrow x_2 \uparrow x_1) \uparrow (x_2 \uparrow x_1 \uparrow \bar{x}_0) \uparrow (\bar{x}_3 \uparrow \bar{x}_2 \uparrow \bar{x}_1 \uparrow x_0))$$

$$y_3 = ((\bar{x}_3 \uparrow x_2 \uparrow \bar{x}_1) \uparrow (\bar{x}_3 \uparrow \bar{x}_1 \uparrow x_0) \uparrow (\bar{x}_3 \uparrow \bar{x}_2 \uparrow x_0) \uparrow (\bar{x}_3 \uparrow \bar{x}_2 \uparrow x_1) \uparrow (\bar{x}_2 \uparrow x_1 \uparrow \bar{x}_0))$$

5. Построение ФС системы ПФ. (рис. 1.25 (а) – (I), рис. 1.25 (б) – (II)).

1.6. ТЕХНОЛОГИЧЕСКАЯ РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

В зависимости от применяемых элементов и схемотехники выделяются следующие классы ЛЭ (так называемые логики):

- резисторно-транзисторная логика (РТЛ);
- диодно-транзисторная логика (ДТЛ);
- транзисторно-транзисторная логика (ТТЛ);
- эмиттерно-связанная логика (ЭСЛ);
- транзисторно-транзисторная логика с диодами Шоттки (ТТЛШ);
- логика на основе МОП-транзисторов с каналами типа p (p-МДП);
- логика на основе МОП-транзисторов с каналами типа n (n-МДП);
- логика на основе комплементарных ключей на МДП-транзисторах (КМДП, КМОП);
- интегральная инжекционная логика И²Л;
- логика на основе полупроводника из арсенида галлия (GaAs).

В настоящее время наиболее широко используются следующие логики: ТТЛ, ТТЛШ, КМОП, ЭСЛ. Устарела и практически не используется РТЛ. Для разрабатываемых в настоящее время устройств можно рекомендовать использовать КМОП-логику, а также логику на основе GaAs.

$x_3 x_2 x_1 x_0 \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$

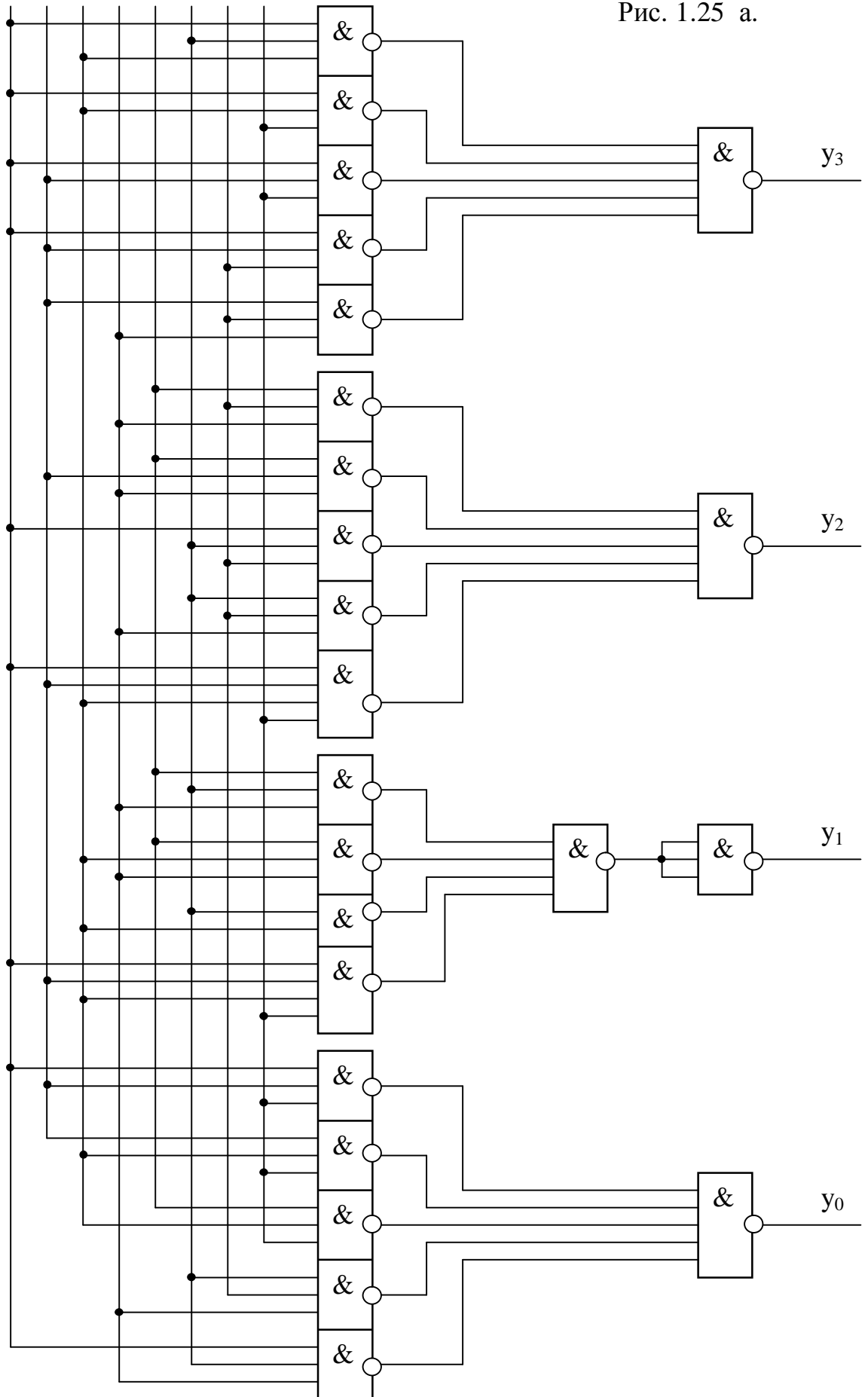


Рис. 1.25 а.

$\overline{x_3} \overline{x_2} \overline{x_1} \overline{x_0} x_3 x_2 x_1 x_0$

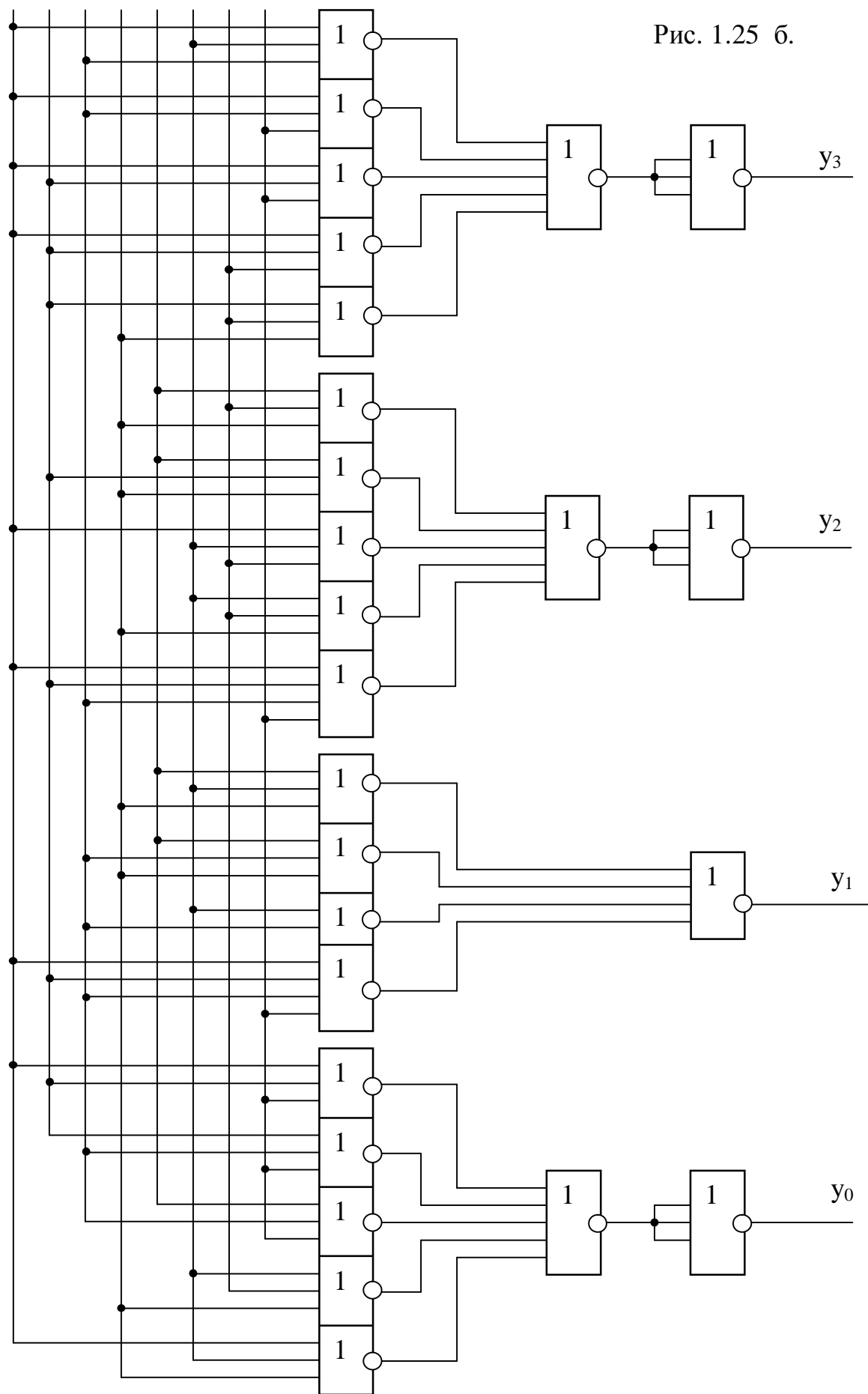


Рис. 1.25 б.

ЛЭ и другие ЦУ выпускаются в составе серий микросхем (интегральных схем (ИС)). Серия микросхем — это совокупность ИС, характеризующихся общими технологическими и схемотехническими решениями, а также уровнями электрических сигналов и напряжения питания.

Приведем примеры серии микросхем: ТТЛ - К155, КМ155, К133, КМ133; ТТЛШ - К530, КР531, КМ531, КР1531, 533, К555, КМ555, 1533, КР1533; ЭСЛ - К500, К1500; КМОП - 564, К561, 1564, КР1554; GaAs -К6500. Каждая серия ИС, несмотря на то, что она обычно содержит самые разнообразные ЦУ, характеризуется некоторым набором параметров, дающих достаточно подробное представление об этой серии. При определении этих параметров ориентируются именно на ЛЭ - простейшие устройства серии ИС. В соответствии с этим говорят о параметрах не серии микросхем, а о параметрах ЛЭ данной серии.

Транзисторно-транзисторные базовые элементы выполняются с использованием биполярных транзисторов.

КМДП-логика выполняется с использованием комплементарных полевых транзисторов с изолированными затворами и с индуцированными каналами.

Эмиттерно-связанные логические элементы выполняются на основе дифференциального усилителя с использованием высокочастотных биполярных транзисторов. ЭСЛ-элементы используются при создании сверхбыстродействующих специализированных ИС, работающих в СВЧ-диапазоне. Эти элементы не отличаются малым потреблением энергии и в настоящее время используются, как правило, в больших ЭВМ. Особенность ЭСЛ-элементов - возникающие в них логические уровни, как правило, существенно отличаются от напряжения питания и от нулевого напряжения

Транзисторно-транзисторные базовые элементы на основе диодов Шоттки. Микросхемы ТТЛШ обычно совместимы по логическим уровням, помехоустойчивости и напряжению питания с микросхемами ТТЛ. Время задержки распространения сигнала элементов ТТЛШ в среднем в два раза меньше по сравнению с аналогичными элементами ТТЛ. К тому же они значительно меньших размеров, что уменьшает емкости их р-п-переходов. Мощность, потребляемая ими, в 4-10 раз меньше.

Интегральная инжекционная логика (ИИЛ или И²Л) построена на использовании биполярных транзисторов и применении оригинальных схемотехнических и технологических решений. Для нее характерно очень экономичное использование площади кристалла полупроводника. Элементы И²Л могут быть реализованы только в интегральном исполнении.

Логика на основе полупроводника GaAs характеризуется наиболее высоким быстродействием, что является следствием высокой подвижности электронов (в 3...6 раз больше по сравнению с кремнием). ИС на основе GaAs могут работать на частотах порядка 10 ГГц и более.

УГО ЛЭ, объединенных в один корпус микросхемы, представлены на рис. 1.26, в котором приведены несколько ИС различных серий.

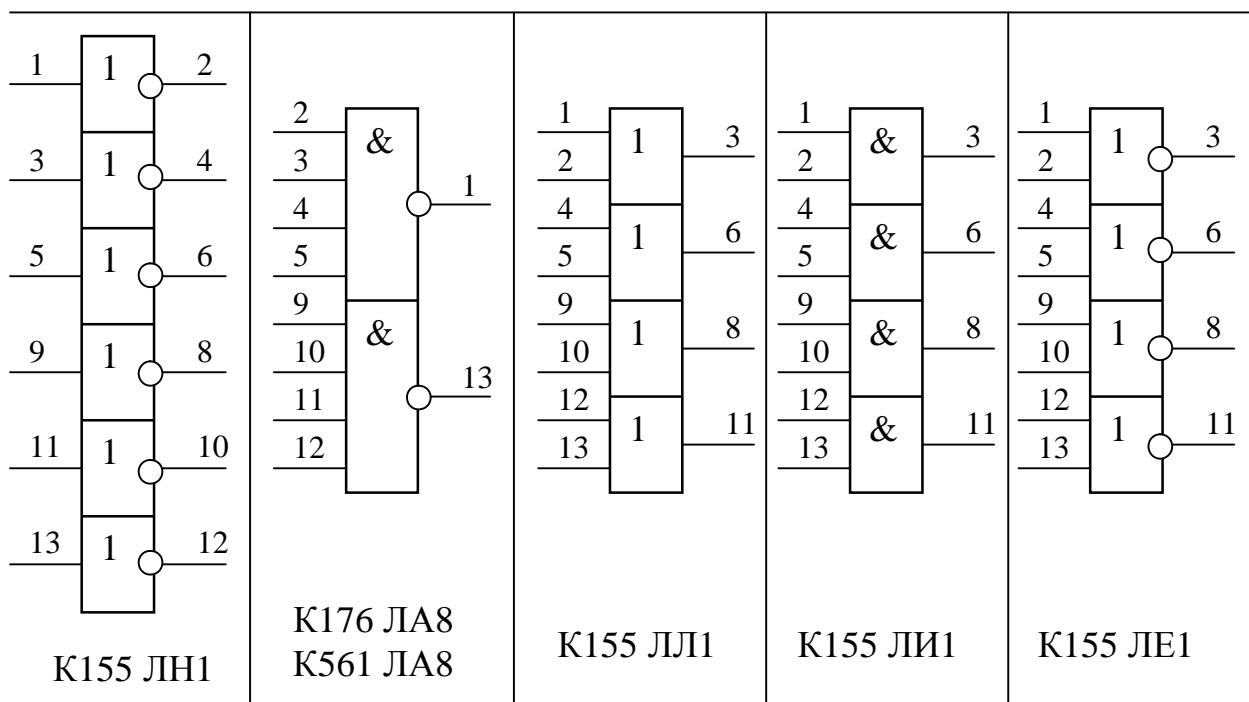


Рис. 1.26. Графическое обозначение серий К155, К176, К561.

Как видно различные ИС могут содержать различное число ЛЭ, различающиеся их видом, количеством входов и выходов, напряжением питания и др. На рис. 1.26 с правой и с левой стороны каждого элемента обозначены номера контактов, соответствующие определенному входу или выходу. Как видно из рис. 1.26 в нумерацию представленных элементов не включены номера ножек 7 и 14. Для таких серий микросхем эти контакты используются для подключения питания элементов, включенных в состав микросхемы. Для

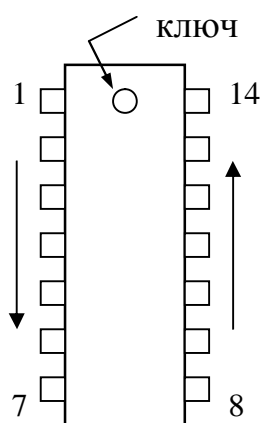


Рис. 1.27

указанных серий элементов ножка 7 – (- Упит.), а 14 – (+ Упит.), где Упит. – напряжение питания. Например, для серии К155 Упит. = 5 В. Напряжение логического нуля соответствует интервалу значений $\Delta U_n = 0 \div 0,4$ В, а логической единице $\Delta U_n = 2,4 \div 5$ В. Для других серий ЛЭ номера контактов для подключения питания, значения Упит., напряжения логического нуля и единицы могут отличаться. Для определения соответствующих номеров контактов, а также ряда других характеристик элементов необходимо использовать специальные радиотехнические справочники.

Номера контактов отсчитываются относительно ключа, расположенного на верхней части корпуса элемента, по направлению против часовой стрелки. Это видно из рис. 1.27, на котором приведен вид корпуса микросхемы (вид сверху) и соответствующее направление отсчета контактов. Отметим, что различные элементы могут отличаться и количеством контактов, число которых может достигать нескольких десятков. Это определяется технологией производства, а также количеством ЛЭ, включенных в состав ИС.

Глава 2. УЗЛЫ ЦИФРОВЫХ УСТРОЙСТВ

2.1. ЦИФРОВЫЕ УЗЛЫ КОМБИНАЦИОННОГО ТИПА

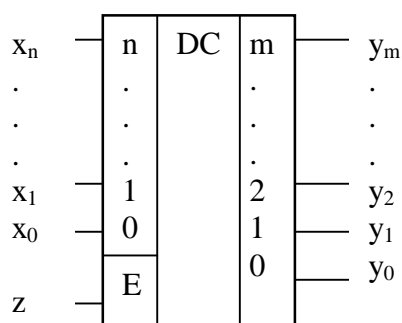
2.1.1. Преобразователи кодов

К преобразователям кодов относятся такие ЦУ как дешифраторы и шифраторы, которые производят преобразование входного кода в выходной по определенному закону.

Дешифратор (decoder) - КС, выполняющая преобразование $(n+1)$ -разрядного кода $X = x_n x_{n-1} \dots x_0$ (представляющего собою натуральное число) в $(m+1)$ -разрядный унитарный код Y (**унитарный код** $Y = y_m y_{m-1} \dots y_0$, - код, у которого во всех разрядах стоят нули, а в разряде номер X записана единица). Таким образом, число $x_n x_{n-1} \dots x_0$ - является входным, а число $y_m y_{m-1} \dots y_0$ - выходным кодом.

Дешифратор, у которого $m + 1 = 2^{n+1}$, называют **полным**. Говорят, что полный дешифратор выполняет преобразование $(n + 1) \rightarrow (m + 1)$. Например, $3 \rightarrow 8$, когда трехразрядный двоичный позиционный код трансформируется в восьмиразрядный унитарный. Дешифратор, у которого $m + 1 < 2^{n+1}$, называют **неполным**. Таким является, к примеру, двоично-десятичный дешифратор.

На рис. 2.1 показано УГО для дешифратора. Здесь $x_n x_{n-1} \dots x_0$ - преобразуемый



код, $y_m y_{m-1} \dots y_0$ - результат преобразования. Вход E - разрешающий (стробирующий). На него подается сигнал так, что при $z = 1$ дешифратор реализует свою функцию, а при $z = 0$ на всех выходах дешифратора $y_i = 0$, где $i = 0, 2^{n+1} - 1$, независимо от входного набора.

Аналитически функция полного дешифратора задается так:

Рис. 2.1

$$y_i = EC_i(x_n x_{n-1} \dots x_0), \quad i = 0, 2^{n+1} - 1 \quad (2.1)$$

Рассмотрим дешифратор $3 \rightarrow 8$. Такой дешифратор называется линейным, поскольку он представляет собою линейку конъюнкторов, а каждый конъюнктор вычисляет один из минтермов системы (2.1). ТИ дешифратора $3 \rightarrow 8$ представляется следующей таблицей:

α	x_2	x_1	x_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Из приведенной ТИ видно, что дешифратор $3 \rightarrow 8$ содержит 8 конъюнкторов (ЛЭ «И»). Аналитические выражения, описывающие такой дешифратор, составим по приведенной ТИ.

$$\begin{aligned} y_0 &= \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0; y_1 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0; y_2 = \bar{x}_2 \cdot x_1 \cdot \bar{x}_0; y_3 = \bar{x}_2 \cdot x_1 \cdot x_0; \\ y_4 &= x_2 \cdot \bar{x}_1 \cdot \bar{x}_0; y_5 = x_2 \cdot \bar{x}_1 \cdot x_0; y_6 = x_2 \cdot x_1 \cdot \bar{x}_0; y_7 = x_2 \cdot x_1 \cdot x_0 \end{aligned} \quad (2.2)$$

На рис. 2.2 представлена ФС дешифратора, построенного по формулам (2.2). Из анализа соотношений (2.2) следует, что рассматриваемый дешифратор

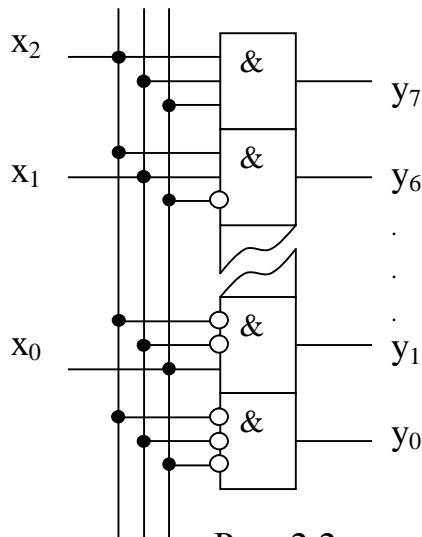


Рис. 2.2

преобразовывает каждое двоичное трехразрядное число в одну и только одну логическую единицу на соответствующем выходе, помеченном десятичным числом. Такие дешифраторы широко используются в устройствах вывода цифровой информации, для индикации двоичного числа в десятичном виде, для определения адресной шины в микросхемах памяти и т.д.

На рис. 2.3 приведен один из возможных способов стробирования дешифратора. Из этого рисунка видно, что сигнал z поступает на все конъюнкторы дешифратора. Такой способ стробирования не вносит дополнительной задержки, однако требует увеличения числа входов конъюнкторов.

Для устранения подобного недостатка существуют другие способы стробирования дешифраторов, например, основанные на включении в состав дешифратора дополнительных ЛЭ.

В номенклатуру ИС той или иной серии наряду с другими узлами входят и дешифраторы.

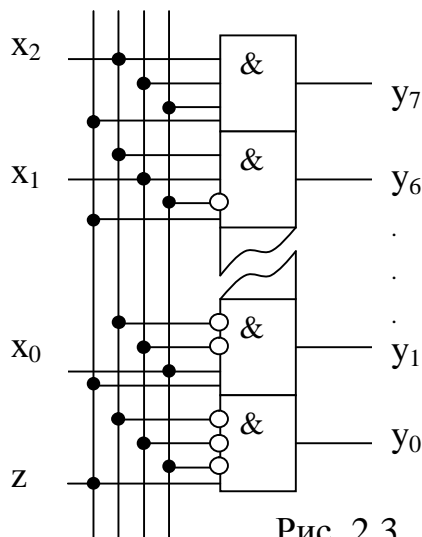


Рис. 2.3

В сериях ТТЛШ дешифраторы обычно имеют инверсные выходы, в сериях КМДП - прямые. ИС дешифраторов обычно имеют малую разрядность и выполняют преобразования $2 \rightarrow 4$ (два таких дешифратора в одном корпусе ИС), $3 \rightarrow 8$ и редко $4 \rightarrow 16$. Для получения шифраторов требуемой разрядности из таких малоразрядных схем составляют необходимые ЦУ.

Шифратор (encoder) - КС, которая реализует функцию, обратную функции дешифратора [преобразует $(m + 1)$ -разрядный унитарный код в позиционный код длиной в $(n + 1)$ разрядов]. Таким образом, можно говорить, что

шифратор выполняет преобразование $(m + 1) \rightarrow (n + 1)$.

На рис. 2.4 представлено УГО шифратора $8 \rightarrow 3$, а на рис. 2.5 - его ФС. ТИ такого шифратора следующая:

α	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

После составления формул получим:

$$y_2 = x_7 \vee x_6 \vee x_5 \vee x_4$$

$$y_1 = x_7 \vee x_6 \vee x_3 \vee x_2$$

$$y_0 = x_7 \vee x_5 \vee x_3 \vee x_1$$

то есть единица из разряда номер X входного унитарного кода Y коммутируется на дизъюнктеры тех разрядов выходного позиционного кода, которые содержат единицы в двоичном изображении X . Скажем, x_5 используется и при формировании y_2 , и при формировании y_0 . Отметим, что цифра x_0 не участвует в формировании выходного кода шифратора.

Шифраторы, как и дешифраторы, допускают каскадное построение для достижения требуемой разрядности.

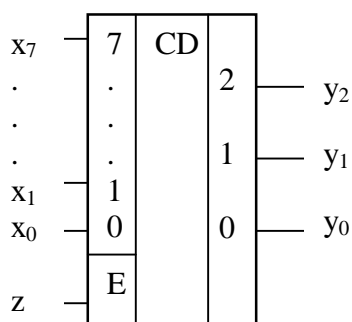


Рис. 2.4

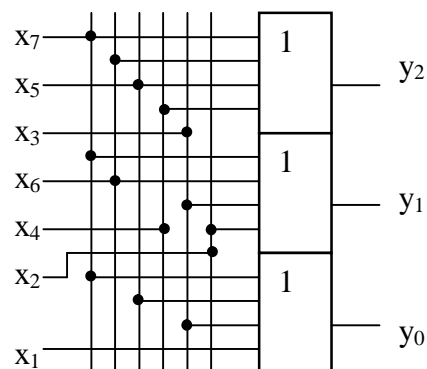


Рис. 2.5

2.1.2. Коммутаторы

К коммутаторам относятся КС такие как мультиплексоры и демультиплексоры.

Мультиплексор - КС, которая передает на свой выход значение одной из $(n + 1)$ входных логических переменных [выполняет преобразование $(n+1) \rightarrow 1$]. Выбор передаваемой переменной производится с помощью кода A , поступающего на адресные входы узла. Таким образом, мультиплексор предназначен для преобразования нескольких информационных каналов последовательно в один информационный канал под воздействием управляющего сигнала.

На рис. 2.6 представлено УГО мультиплексора $4 \rightarrow 1$, работа которого описывается табл. 2.1. Как видим, он предназначен для коммутации на выход Y одного из четырех входных сигналов x_k , т.е. $Y = x_k$ при $A = k$, где $k = 0, \dots, 3$.

a_1	a_0	Y
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

Таблица 2.1

На рис. 2.7 приводится вариант ФС такого мультиплексора. Синтез этой схемы выполнен не по классической методике, когда функция устройства задается в терминах БА, затем минимизируется и, наконец, разрабатывается ФС. В данном случае это практически невозможно, хотя бы потому, что Y зави-

сит от шести аргументов: $x_3x_2x_1x_0a_1a_0$. Здесь используется инженерный подход к синтезу КС. Он состоит в том, что заданная словесно функция устройства детализируется путем разбиения ее на подфункции. Последние, при необходимости, можно разбить на более мелкие фрагменты. Каждая подфункция связывает некоторую промежуточную величину с тем или иным подмножеством аргументов. А подфункции, как ПФ меньшей размерности, уже реализуются известными методами. В нашем случае можно выделить две подфункции. Первая состоит в коммутации четырех логических переменных на один выход. Она реализуется ЛЭ «И» и «ИЛИ». Вторая подфункция заключается в трансформации кода A , состоящего из аргументов a_1 и a_0 , в унитарный код, цифры которого будут поданы на конъюнкторы. Эта подфункция реализуется дешифратором ДС. К двум основным подфункциям также можно добавить еще и такую, как стробирование мультиплексора сигналом z , которая реализуется на выходном дополнительном конъюнкторе (на рис. 2.7 не указан).

Мультиплексоры выпускают на малое число входов. К примеру, ИС 1533КП2 - это два мультиплексора $4 \rightarrow 1$ с общими входами для двухразрядного кода A , а ИС 1554КП16 - четыре двухвходовых мультиплексора.

Мультиплексор нередко применяют как универсальную КС для реализации произвольной ПФ, число аргументов которой не превышает разрядности кода A .

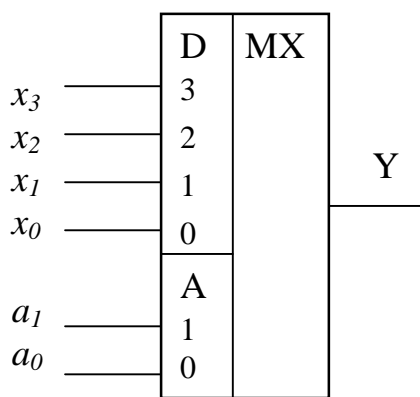


Рис. 2.6

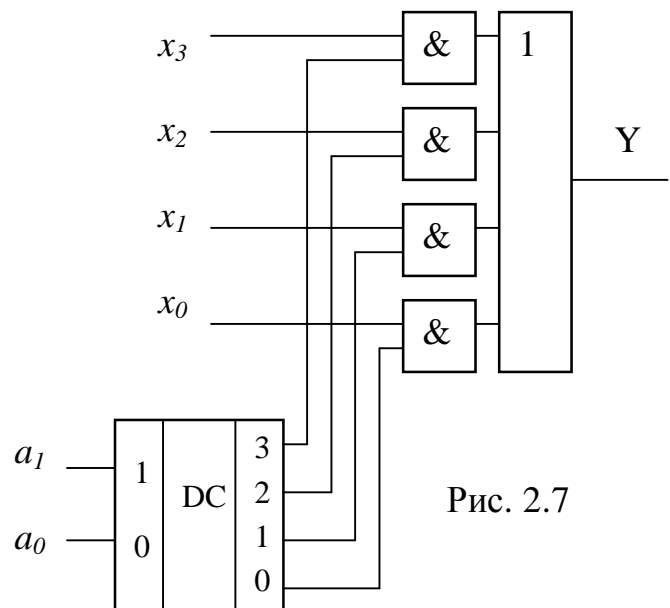


Рис. 2.7

Пример 2.1. ПФ трех переменных $x_2x_1x_0$ задана табл. 2.2. Составить ФС, реализующую данную ПФ, с использованием мультиплексора.

α	x_2	x_1	x_0	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Таблица 2.2

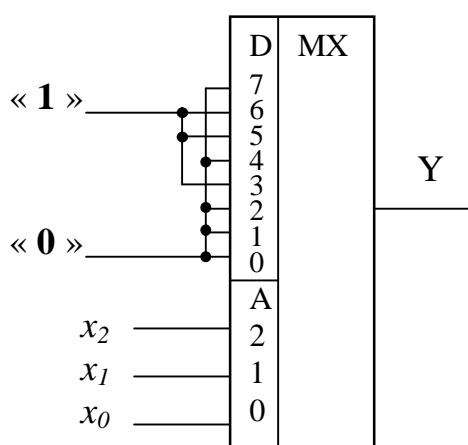


Рис. 2.8

Решение. Определим значения α при которых $Y = 0$ и $Y = 1$. В нашем случае $Y = 0$ при $\alpha = 0, 1, 2, 4, 7$, а $Y = 1$ при $\alpha = 3, 5, 6$.

На рис. 2.8 показан мультиплексор, реализующий эту ПФ. При таком способе реализации ПФ минимизировать ее не нужно. Просто на информационные входы мультиплексора подаются константы 0 и 1 в том порядке, как нули и единицы распределены в таблице истинности ПФ.

Демультимплексор - КС, передающая на один из выходов значение логической переменной, подаваемой на вход [КС, выполняющая коммутацию значения входной логической переменной на один из $(n + 1)$ выходов]. Таким образом, данная схема выполняет преобразование $1 \rightarrow (n + 1)$. Выбор выхода (направление коммутации) производится кодом А, формируемого операндами a_0 и a_1 , на адресных входах демультимплексора.

На рис. 2.9 приведено УГО демультимплексора $1 \rightarrow 4$, а на рис. 2.10 - его ФС. Нетрудно заметить, что схема на рис. 2.10 представляет собою декодер $2 \rightarrow 4$, когда x - его стробирующий сигнал.

Следовательно, дешифратор со стробированием и демультимплексор - это одна и та же схема. Поэтому для декодера с разрешающим входом используют название дешифратор-демультимплексор.

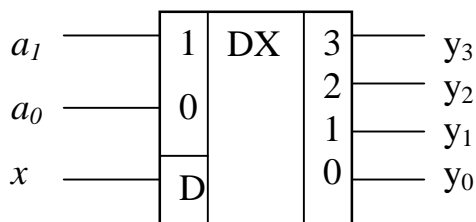


Рис. 2.9

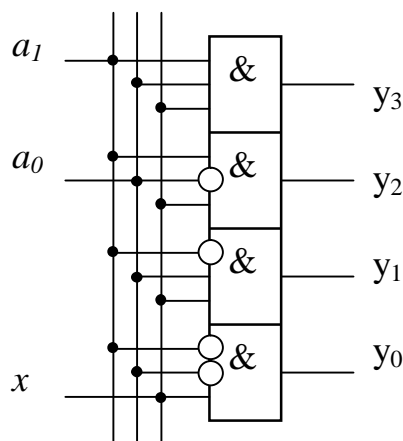


Рис. 2.10

2.1.3. Комбинационные сумматоры

Сумматор – КС, реализующая арифметическое сложение двоичных чисел (операндов). Сумматор – важнейшая часть арифметико-логического блока микропроцессора. Сумматоры характеризуются разрядностью. Как правило, многоразрядные сумматоры строятся на основе одnorазрядных.

Функцией сумматора является арифметическое сложение двоичных чисел $A = a_n a_{n-1} \dots a_0$ и $B = b_n b_{n-1} \dots b_0$ по правилам двоичной арифметики. Сумматор имеет входы A и B для $(n+1)$ -разрядных операндов, вход cr для цифры переноса p_0 в младший разряд суммы, выход C для $(n+1)$ -разрядного результата и вывод CR для цифры p_{n+1} переноса из старшего разряда суммы.

Быстродействие сумматора обычно оценивают временем **задержки распространения переноса в сумматоре**, а именно величиной

$$T = t_{cr,CR},$$

где $t_{cr,CR}$ - интервал времени от момента подачи цифры переноса p_0 на вход cr , и до момента получения на выводе CR значения p_{n+1} .

Сумматор с последовательным переносом. Операция двоичного сложения в ЦУ, как и на бумаге, выполняется поразрядно, начиная с младших разрядов.

Пусть даны операнды a_i и b_i , принимающие значения 0 или 1. Правила двоичного сложения заключаются в следующем:

$$\begin{array}{r} a_i \quad b_i \quad p_i \quad p_{i+1} \\ 0 + 0 = 0 \quad 0 \\ 0 + 1 = 1 \quad 0 \\ 1 + 0 = 1 \quad 0 \\ 1 + 1 = 0 \quad 1 \end{array}$$

Здесь p_i – сумма двоичных чисел данного разряда, p_{i+1} – цифра переноса в старший разряд суммы.

Используя указанные правила, сложим двоичные числа A и B :

$$\begin{array}{r} p_{i+1} \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \\ A \quad \quad \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \\ B \quad + \quad \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \\ \hline p_0 \quad \quad \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \end{array} \quad \text{или} \quad \begin{array}{r} 1 \quad 1 \quad 1 \quad 0 \\ 1 \quad 0 \quad 1 \quad 0 \\ + \quad 1 \quad 1 \quad 1 \quad 1 \\ \hline 1 \quad 1 \quad 0 \quad 0 \quad 1 \end{array}$$

Здесь p_0 – сумма двоичных чисел, p_{i+1} – цифра переноса в старший разряд суммы.

p_i	a_i	b_i	c_i	p_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблица 2.3

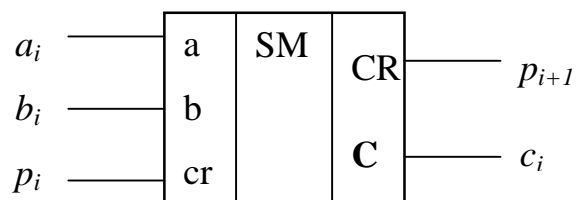


Рис. 2.11

Вычисления реализуются посредством одnorазрядного сумматора (рис. 2.11). **Одноразрядный сумматор** - сумматор, выполняющий суммирование

вание одноразрядных двоичных операндов (часть многоразрядного комбинационного сумматора). Функция одноразрядного сумматора задается таблицей 2.3, которая построена по правилам двоичной арифметики. В приведенной таблице операнды a_i, b_i, c_i - цифры слагаемых A, B и суммы S, p_i - цифра переноса в i -й разряд.

Если эту таблицу рассматривать как таблицу истинности для двух ПФ одноразрядного сумматора, то можно записать:

$$c_i = (\bar{p}_i \cdot \bar{a}_i \cdot b_i) \vee (\bar{p}_i \cdot a_i \cdot \bar{b}_i) \vee (p_i \cdot \bar{a}_i \cdot \bar{b}_i) \vee (p_i \cdot a_i \cdot b_i),$$

$$p_{i+1} = (p_i \cdot a_i) \vee (p_i \cdot b_i) \vee (a_i \cdot b_i).$$

Цена одноразрядного сумматора (по Квайну), построенного по этим формулам, составит $22 + 9 = 31$ единицу.

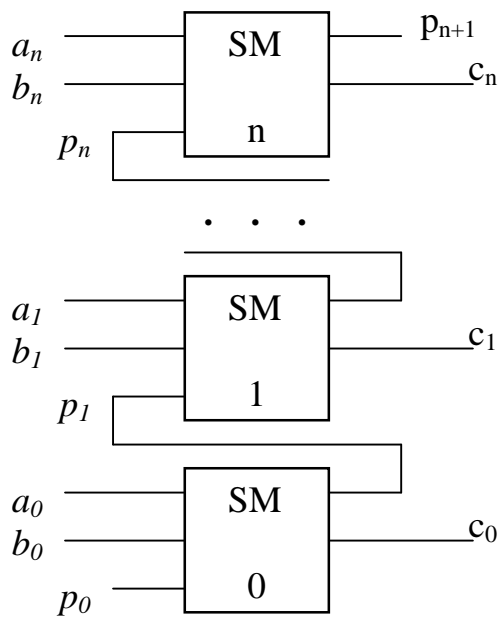


Рис. 2.12

Уменьшают цену одноразрядного сумматора неформальными методами. Например, сопоставив колонки p_{i+1} и c_i в таблице 2.3, легко заметить, что на всех наборах, кроме нулевого и седьмого, $c_i = p_{i+1}$. Поэтому для вычисления c_i можно использовать величину p_{i+1} :

$$c_i = p_{i+1} \cdot (a_i \vee b_i \vee p_i) \vee (a_i \cdot b_i \cdot p_i) \quad (2.3).$$

Цена в этом случае значительно понижается по сравнению с вышерассмотренными формулами.

Описанные одноразрядные сумматоры применяются для построения $(n+1)$ -разрядного сумматора с последовательным переносом. Их соединяют так, как показано на рис. 2.12, т.е. выход CR соединяют со входом последующего сумматора c_{i+1} .

Сумматор с последовательным переносом - сумматор, в котором цифры переноса формируются последовательно, распространяясь от младших разрядов к старшим.

В таком сумматоре прямо воспроизводится процесс ручного сложения чисел столбиком. Быстродействие сумматора с последовательным переносом пропорционально его разрядности:

$$T = 3 \tau (n + 1),$$

где τ - время задержки в одном ЛЭ.

Сумматор с параллельным переносом. При решении многих задач быстродействие сумматора с последовательным переносом оказывается недопустимо низким. Радикальный метод повышения быстродействия сумматоров - это одновременное формирование переносов во всех разрядах. **Сумматор с параллельным переносом** - сумматор, в котором цифры переноса формируются одновременно во всех разрядах.

Введем две вспомогательные ПФ: $\alpha_i = a_i \cdot b_i$ и $\beta_i = a_i \vee b_i$.

Первая из них - α_i – называется функцией генерации переноса в i -ом разряде. Когда $\alpha_i = 1$, то $p_{i+1} = 1$ не зависимо от значения p_i . Вторая - β_i – называется функцией прозрачности i -го разряда для переноса. Если $\beta_i = 1$ ($a_i \vee b_i \neq 0$), то $p_{i+1} = p_i$. Теперь для p_{i+1} можно составить таблицу, в которой каждая пара переменных a_i и b_i будет заменена соответствующими значениями вспомогательных функций α_i и β_i (таблица 2.4).

p_i	α_i	β_i	p_{i+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Таблица 2.4

Тогда

$$p_{i+1} = \alpha_i \vee \beta_i \cdot p_i. \quad (2.4)$$

Распишем выражения (2.4) для всех значений i :

$$\begin{aligned}
 p_0 &= \{0,1\}, \\
 p_1 &= \alpha_0 \vee \beta_0 \cdot p_0, \\
 p_2 &= \alpha_1 \vee \beta_1 \cdot p_1, \\
 p_3 &= \alpha_2 \vee \beta_2 \cdot p_2, \\
 &\dots\dots\dots \\
 p_{i+1} &= \alpha_i \vee \beta_i \cdot p_i
 \end{aligned}$$

Как видно, операндами в каждом выражении для p_{i+1} являются только цифры слагаемых a_i , b_i и цифра переноса p_0 . Следовательно, значения переносов во всех разрядах сумматора формируются одновременно, начиная с момента подачи на его входы слагаемых A и B и цифры p_0 . Цифры суммы c_i в таком сумматоре вычисляются одинаково – по формуле (2.3).

Быстродействие сумматора с параллельным переносом не зависит от его разрядности и составляет величину $T = 3 \tau$. Однако высокое быстродействие сумматора с параллельным переносом связано с большими материальными затратами. Например, для слова $(n + 1) = 4$ цена по Квайну равна 60. К тому же сумматор с параллельным переносом имеет нерегулярную структуру, схемы для вычисления p_{i+1} от разряда к разряду становятся все сложнее и сложнее. Поэтому ИС подобных сумматоров имеют малую разрядность.

Сигналы на выходах КС могут появляться не одновременно. Например, в сумматоре выходной сигнал переноса появляется, как правило, быстрее, чем сигнал суммы. Объясняется это тем, что сигналы проходят различное число ступеней обработки. Кроме того, в импульсном сигнале имеется временной интервал, когда значение логического сигнала не определено. Указанные причины могут вызывать в ЦУ неодновременное появление управляющих сигналов, ложные срабатывания, возникновение дополнительных паразитных импульсов и т. п. Эти нарушения в работе ЦУ называются "состязаниями". Для устранения "состязаний" осуществляют синхронизацию работы всех элементов устройства. Для этого в состав элементов включают дополнительные схемы на основе ЛЭ «И», «ИЛИ», «НЕ», с помощью которых все информационные сигналы появляются только при подаче специального, короткого сигнала синхронизации (стробирующий сигнал). В выпускаемых промышленностью КС, как правило, предусмотрен специальный вывод для подачи такого сигнала синхронизации (вход C , z). Выходные сигналы в этом случае изменяются только после подачи этого сигнала.

2.2. ЦИФРОВЫЕ УЗЛЫ НАКАПЛИВАЮЩЕГО ТИПА

2.2.1. Принцип работы триггеров и их виды

Простейшими из устройств последовательного типа являются триггеры. **Триггер** - элемент последовательного типа, обладающий двумя устойчивыми состояниями (состояниями устойчивого равновесия) и способный под воздействием внешнего управляющего сигнала скачкообразно переходить (переключаться) из одного состояния в другое. **Время переключения триггера** - параметр, характеризующий быстродействие триггера; определяется как интервал времени от подачи необходимого управляющего сигнала до переключения триггера (из 0 в 1 или из 1 в 0).

Триггеры используются для построения более сложных узлов: регистров, счетчиков и др.

Функцией триггера является фиксация значений $\{0,1\}$ логической переменной. Для этого триггер должен обладать двумя состояниями устойчивого равновесия.

Триггеры могут быть составлены с помощью простых ЛЭ, которые могут хранить («запомнить») логическое состояние (0 или 1) неопределенно долго (разумеется, пока есть напряжение питания). Такие устройства образуют память, и их выход может находиться в одном из двух устойчивых состояний: 0 или 1. Такие схемы называют бистабильными, или **триггерами**. Существует множество типов триггеров, например, **RS, JK, D, DV, T** – триггеры. Рассмотрим некоторые из них.

RS-триггер. Простейший RS-триггер реализуется на двух элементах «И-НЕ» или «ИЛИ-НЕ» (рис. 2.13). Триггер имеет прямой (Q) и инверсионный (\bar{Q}) выходы, т. е. если на прямом выходе 1 — в это время на инверсном 0 и наоборот. Вход S - (set) установка триггера в состояние 1, на выходе Q .

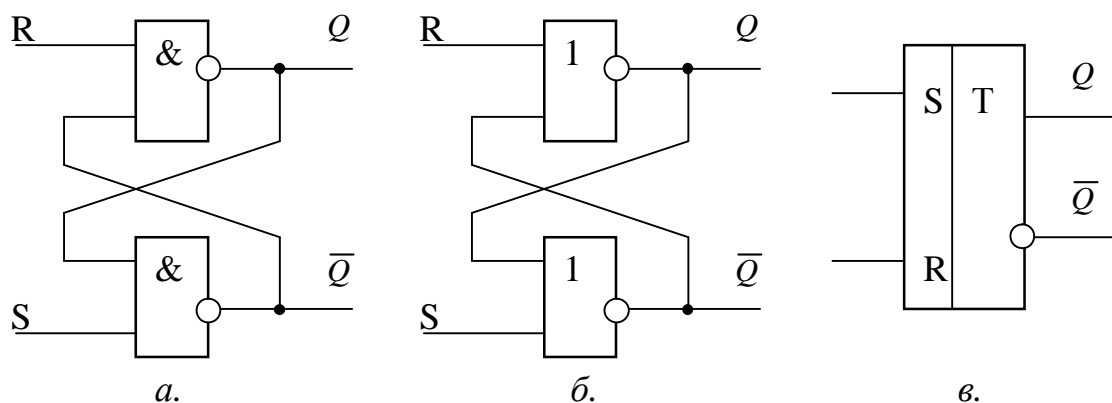


Рис. 2.13 Схемы и условное обозначение RS-триггера:

а - на элементах «И-НЕ»; б - на элементах «ИЛИ-НЕ»; в - условное обозначение RS-триггера.

Вход R – (reset) установка триггера в состояние 0, на выходе Q . Работа RS-триггера определяется работой входящих ЛЭ.

Если в схеме, изображенной на рис. 2.13 (а), на вход S поступает сигнал логической единицы, а на вход $R - 0$, то триггер устанавливается в состояние $Q = 1$, $\bar{Q} = 0$. И наоборот, если на вход S поступает сигнал логического нуля, а на вход $R - 1$, то триггер устанавливается в состояние $Q = 0$, $\bar{Q} = 1$. Таблица истинности RS-триггера на элементах «И-НЕ» приведена в табл. 2.5.

S	R	Q	\bar{Q}
0	1	0	1
1	0	1	0
0	0	H/O	
1	1	Q^*	

Таблица 2.5. Таблица истинности RS-триггера на элементах «И-НЕ». Q^* - хранение предыдущего состояния; H/O – неопределенное состояние триггера.

Сочетание $S = 0$ и $R = 0$ является запрещенным (см. табл. 2.5), так как на обоих выходах триггера устанавливаются логические единицы и после снятия входных сигналов состояние его непредсказуемо. Из сказанного ясно, что для переключения RS-триггера в состояние $Q = 1$ на его входы следует подать комбинацию $S = 1$, $R = 0$, а для переключения в состояние $Q = 0$ - комбинацию $S = 0$, $R = 1$. В случае комбинации входных сигналов $S = 1$ и $R = 1$ триггер хранит ранее установленное состояние.

Из рис. 2.13 видно, что электрическая схема относительно входов R и S является симметричной. Поэтому нет строгого соответствия между входами R и S . Например, можно заменить вход R на S , вход S на R и выход Q на \bar{Q} . Работа триггера при этом останется неизменной.

Необходимо иметь в виду, что в момент включения питания микросхем триггера, может установиться совершенно любое начальное состояние триггера – либо $Q = 1$ и $\bar{Q} = 0$, либо $Q = 0$ и $\bar{Q} = 1$. Для установки триггера в строго определенное состояние в момент включения, необходимо применять более сложные КС, основанные на включении в состав триггера так называемых нетактируемых входов (рассмотрено ниже).

Для триггера представленного на рис. 2.13 (б), который собран на ЛЭ «ИЛИ-НЕ», логические сигналы будут прямо противоположны сигналам триггера, собранного на элементах «И-НЕ». Например, комбинация $R = 1$ и $S = 1$ в этом случае будет запрещенной (на обоих выходах триггера устанавливаются логические нули и после снятия входных сигналов состояние его также непредсказуемо), а при $R = 0$ и $S = 0$ триггер хранит установленное ранее состояние.

На графике, представленном на рис. 2.14 показаны временные диаграммы, поясняющие работу RS-триггера, составленного на элементах «ИЛИ-НЕ».

Когда на RS-входы идут импульсные сигналы для перехода ЛЭ из одного состояния в другое, требуется какое-то время (хотя и очень маленькое),

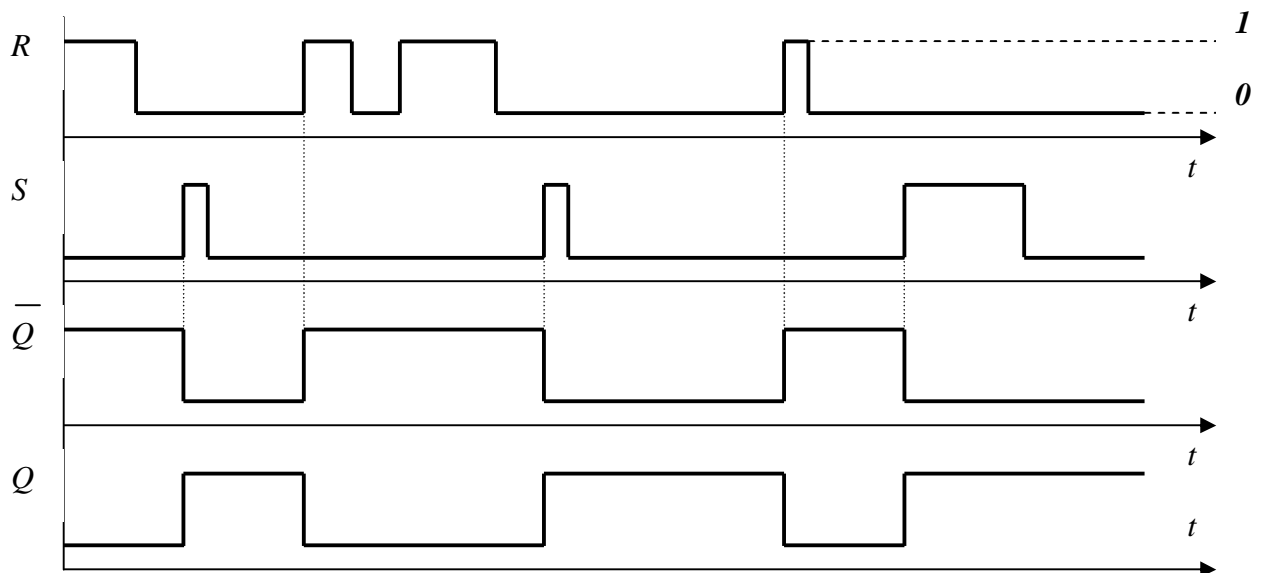


Рис. 2.14

а для триггера в целом время переключения — минимум удвоенное время ЛЭ. Если его брать с запасом, то принимают время переключения $t_{пер} = 3t_3$, где t_3 — время переключения ЛЭ (время задержки). Для надежного переключения триггера длительность входного переключающего сигнала не должна быть меньше $t_{пер}$.

Время переключения для конкретной серии триггеров — величина справочная. При ремонте и контроле устройств, длительность импульсов можно измерить осциллографом. Для промышленных триггеров время переключения очень мало и достигает несколько сотен, и даже десятков наносекунд.

Различают **асинхронные** и **синхронные** триггеры.

На рис. 2.13 (в) представлено УГО асинхронного RS триггера. Особенностью асинхронных является то, что для установки их в различные состояния, т. е. для записи информации, достаточно поступления сигналов на RS-входы. При наличии помех такой триггер может работать ненадежно. Например, короткие импульсные помехи, попадающие на R- или S-входы могут привести к самопроизвольному изменению состояния триггера.

Для повышения помехоустойчивости используют синхронные триггеры. В синхронных триггерах есть дополнительный вход С для синхронизирующего (тактового) сигнала. Запись информации производится сигналами на информационных входах (R, S), но с одновременным поступлением тактового сигнала (логической 1) на вход С, т. е. $C = 1$. Если $C = 0$, триггер находится в режиме хранения (сохраняет предыдущее состояние) и подача каких либо сигналов на входы R и S не изменяют его состояния.

На рис. 2.15 показана схема и УГО синхронного RS-триггера. Как видно, тактирование сигналов производится схемой, состоящей из двух ЛЭ «И». При поступлении сигнала на вход С ЛЭ «И» «открываются» и управляющие сигналы поступают на входы асинхронного триггера. Такая синхронизация называется **статической**. При этом, переключение триггера происходит по переднему фронту управляющего импульса (например, см. рис. 2.14).

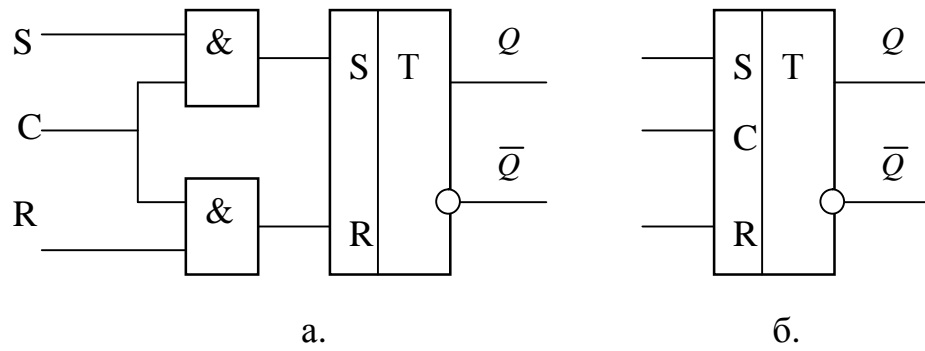


Рис. 2.15. Схема (а) и условное обозначение (б) синхронного RS-триггера.

Рассмотренные выше RS-триггеры невозможно использовать в ЦУ с обратными связями из-за неопределенностей, возникающих при работе. Действительно, выход Q триггера нельзя соединить с входами R или S , так как изменения на этих выводах происходят практически одновременно. Такие же неопределенности будут возникать в сложных устройствах с обратными связями, содержащих несколько триггеров и КС. Решением данной задачи является использование так называемого *двухступенчатого RS-триггера*.

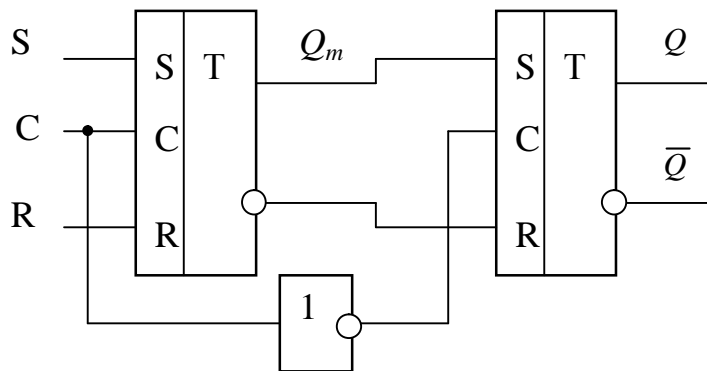


Рис. 2.16. Схема двухступенчатого RS-триггера.

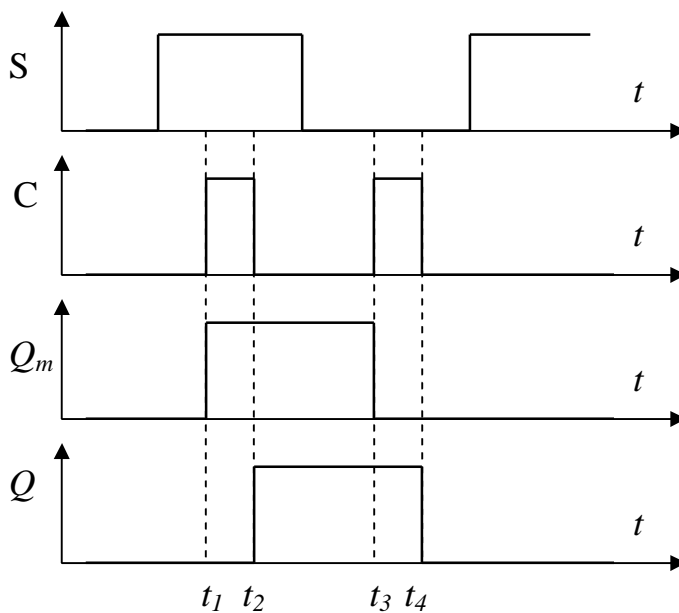


Рис. 2.17. Временные диаграммы работы двухступенчатого RS-триггера

При использовании двухступенчатого RS-триггера, схема которого приведена на рис. 2.16, допускается соединение его входов и выходов.

Двухступенчатый триггер состоит из двух синхронных RS-триггеров и дополнительного ЛЭ «НЕ». При подаче входных управляющих сигналов и синхросигнала производится запись информации в первый триггер (момент времени t_1 на рис. 2.17). При этом второй триггер не изменяет своего

состояния, так как на его синхровход С с инвертора подается логический ноль. Только по окончании записи в первый триггер при изменении значения синхросигнала с единицы до нуля производится запись во второй триггер двухступенчатой схемы (момент t_2 на рис. 2.17). Практически переключение происходит в два такта, его осуществляют две последовательности импульсов, смещенные во времени. Поэтому двухступенчатый триггер называют еще двухтактным

Временные диаграммы работы двухступенчатого триггера на рис. 2.17 получены при условии — сигнал на входе R инвертирован по отношению к сигналу на входе S. Как видим, двухступенчатый RS-триггер переключается по заднему фронту синхронизирующего сигнала С, т.е. переключение происходит в момент изменения уровней напряжений от высокого к низкому. Такая синхронизация называется **динамической**.

УГО двухступенчатого RS-триггера показано на рис. 2.18. Наличие динамической синхронизации отмечено наклонной чертой. Причем ее наклон соответствует заднему фронту синхроимпульса. В справочной и учебной литературе для обозначения динамической синхронизации могут также использоваться треугольник, звездочка, крестик и т.п. Использование двух ступеней отмечается на рис. 2.18 двумя буквами Т.

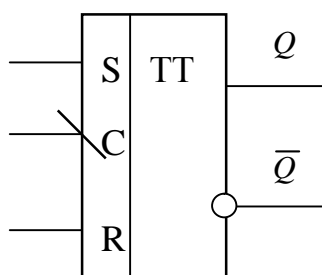


Рис. 2.18. Условное обозначение двухступенчатого RS-триггера

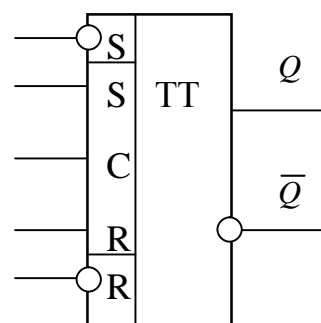


Рис. 2.19. Условное обозначение двухступенчатого RS-триггера с асинхронными входами.

Многие свойства и особенности триггеров, существенно важные при практическом применении, определяются принципом их построения. Таким образом, по этому признаку все рассмотренные триггеры делятся на одноступенчатые и двухступенчатые. Триггерные структуры, как правило, строятся на основе самого простого RS-триггера.

Одноступенчатым назван триггер, имеющий одну ступень запоминания информации (0 или 1). Запись в триггер информации, которая находится, например, на входах R и S, начинается сразу после поступления разрешающего сигнала на вход C.

В двухступенчатых триггерах имеются две ступени запоминания информации: основная и вспомогательная (вторая), каждая из которых представляет собой фактически одноступенчатый триггер (например, RS). Ступени управляются так, что с приходом тактового импульса информация записывается в первую ступень, а после его снятия — переписывается во вторую

и появляется на выходах триггера. Таким образом, в двухступенчатых триггерах процесс передачи информации двухтактный (от входов к выходам) и быстродействие его в два раза ниже.

Как было сказано выше, в момент включения питания микросхем триггера, может установиться совершенно любое начальное состояние триггера. Поэтому для установки триггера в строго определенное состояние в момент включения, в состав триггера включают дополнительные так называемые асинхронные инверсные входы \bar{S} и \bar{R} . По этим входам он независимо от сигнала на тактовом входе C переключается в состояние 1 (вход S) или 0 (вход R). Такие входы называют нетактируемыми и в тексте их указывают после информационных. К примеру, RS-триггер с асинхронными входами обозначается как RSRS. УГО тактируемого RS-триггера с инверсными асинхронными входами показано на рис. 2.19.

Входы \bar{S} и \bar{R} непосредственно воздействуют на RS-триггеры первой и второй ступени, поэтому они выполнены на трехвходовых элементах «И-НЕ». Для переключения триггера по входам \bar{S} и \bar{R} не требуется наличия синхроимпульсов на входе C . При подаче на эти входы логической 1 ($\bar{R} = \bar{S} = 1$) RSC-триггер работает как обычный синхронный RS-триггер, так как его ступени ею не переключаются. При $\bar{S} = 0$ и $\bar{R} = 1$ RSC-триггер устанавливается в состояние $Q = 1$, а при $\bar{S} = 1$ и $\bar{R} = 0$ - в состояние $Q = 0$. При этом необходимо, чтобы на тактируемом входе C присутствовала логическая единица.

Триггер задержки (D-триггер). Одним из самых широко используемых триггеров является D-триггер (триггер задержки). Чаще всего D-триггер выполняется на основе двухступенчатого RS-триггера при включении на входе дополнительного инвертора, связывающего R- и S-входы. Важное преимущество этого триггера состоит в том, что он имеет только один информационный D-вход, в то время как, например, RS-триггер - 2 информационных входа R и S. Следовательно, в RS-триггере для записи некоторой булевой переменной необходимо одновременно подавать на вход S ее прямое значение, а на вход R – инверсное, а это не всегда является удобным.

D-триггер имеет один информационный D-вход и тактовый C-вход. Схема D-триггера и его УГО приведены на рис. 2.20 (а) и (б) соответственно.

C(t)	D(t)	Q(t+1)
0	0	Q(t)
0	1	Q(t)
1	0	0
1	1	1

Таблица 2.6

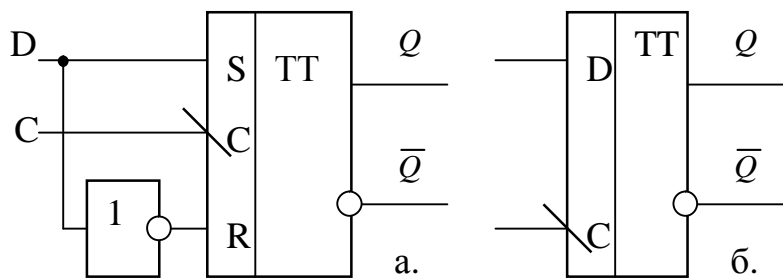


Рис. 2.20. Схема (а) и условное обозначение (б) D-триггера.

Информация в D-триггер записывается по заднему фронту синхронизирующего импульса. Поэтому сигнал на выходе Q при подаче n -го синхрои́мпульса появляется с задержкой на один такт.

Функция D-триггера задается таблицей 2.6. Как видно, при $C(t) = 1$ $Q(t+1) = D(t)$, т.е. значение сигнала на D-входе триггера транслируется на его выход Q .

Таким образом, при $C = 1$ D-триггер является повторителем: на выходе Q повторяется потенциал входа D . Однако это повторение начинается только с поступлением тактового импульса на вход C , т. е. с задержкой относительно сменившегося потенциала на D-входе. При $C = 0$ триггер хранит информацию, поступившую при $C = 1$ с D-входа.

Т-триггер (триггер со счетным входом) имеет один T-вход (асинхронный триггер) либо два входа: T-вход и синхронизирующий V-вход (синхронный T-триггер, его нередко называют TV-триггером). Функция T-триггера

T(t)	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Таблица 2.7

(при отсутствии V-входа) задается таблицей 2.7. T-триггер строится обычно на основе двухступенчатых RS- или D-триггеров путем соединения выходов со входами перекрестными связями, как это показано на рис. 2.21 (а). УГО T-триггера показано на рис. 2.21

(б). Работа асинхронного T-триггера иллюстрируется временными диаграммами на рис. 2.22. В синхронном TV- триггере смена состояния триггера происходит только при поступлении на V-вход синхронизирующего (разрешающего) импульса.

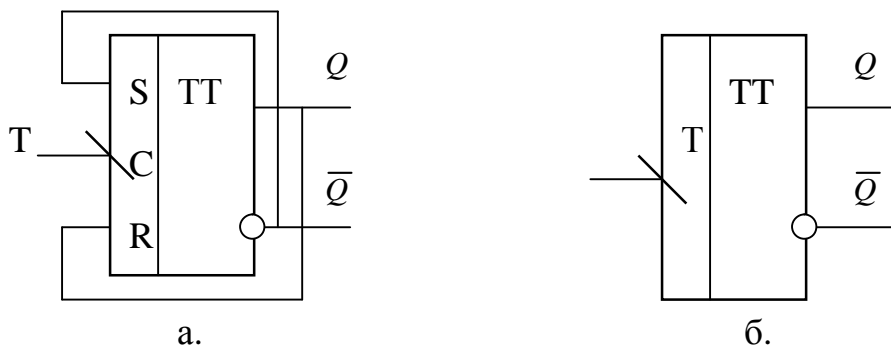


Рис. 2.21 Структурная схема T-триггера (а) и его условное обозначение (б).

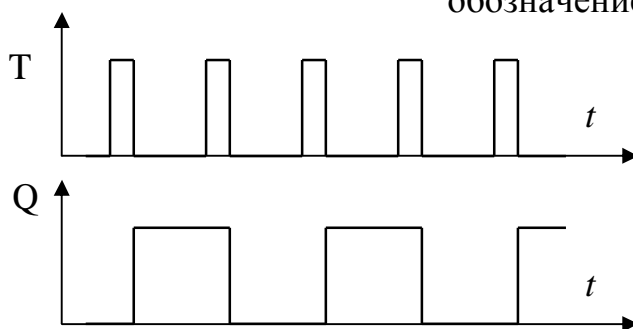


Рис. 2.22. Временные диаграммы асинхронного T-триггера.

JK-триггер. JK-триггер наиболее универсальный из рассмотренных ранее. Имеет два информационных входа - J и K, тактовый вход C и два выхода — прямой Q и инверсный. Он обладает свойствами всех других триггеров и поэтому наиболее широко представлен в современных сериях ИС. С помо-

щью определенных схем включения из JK-триггера можно построить, например, T-триггер. Функция JK-триггера задается таблицей 2.8. Схема JK-триггера и его УГО показаны на рис. 2.23, (а) и (б), соответственно.

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

Таблица 2.8

В схему включены два двухвходовых ЛЭ «И». Так как на их входы подаются выходные сигналы RS-триггера, то один из элементов «И» будет всегда закрыт для прохождения сигналов управления. По этой причине на входы JK-триггера можно одновременно подавать единичные (или нулевые) сигналы. Как известно, такая комбинация входных сигналов запрещена у RS-триггера.

Вход J триггера аналогичен входу S рассмотренного выше RS-триггера, а вход K – входу R RS-триггера. Если $J = K = 0$, то получим режим хранения записанной ранее информации. Если $J = K = 1$, то с приходом синхроимпульса триггер изменяет свое состояние на противоположное (см. таблицу 2.8).

Наряду с информационными и тактовыми входами JK-триггеры также могут иметь дополнительные входы для установки начального состояния (см. рис. 2.19). Например, JKRS-триггер имеет вход S для установки его в 1 и вход R для установки в 0. Может быть предусмотрен в микросхеме триггера только один из установочных входов. Начальное состояние триггера устанавливается коротким импульсом положительной или отрицательной полярности (прямой или инверсный вход). Часто для расширения функциональных возможностей триггеры снабжаются входной логикой, которая реализуется встроенными ЛЭ.

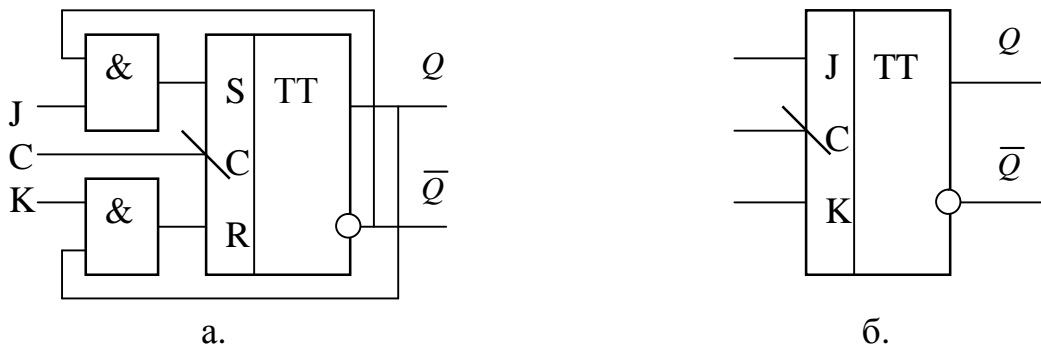


Рис. 2.23 Схема JK-триггера (а) и его условное графическое обозначение (б).

Пример 2.2. Временные диаграммы импульсов на входах J, K и C JK-триггера даны диаграммой, представленной на рис. 2.24. Построить последовательность импульсов на выходе Q. Начальное состояние $Q = 1$.

- Решение:** 1. Дополнить диаграмму временной осью Q(t).
2. Так как JK-триггер является двухступенчатым триггером, то синхронизация по входу C происходит по заднему фронту импульса.

3. В соответствии с таблицей 2.8 построить диаграмму Q(t).

Решение представлено на рис. 2.25.

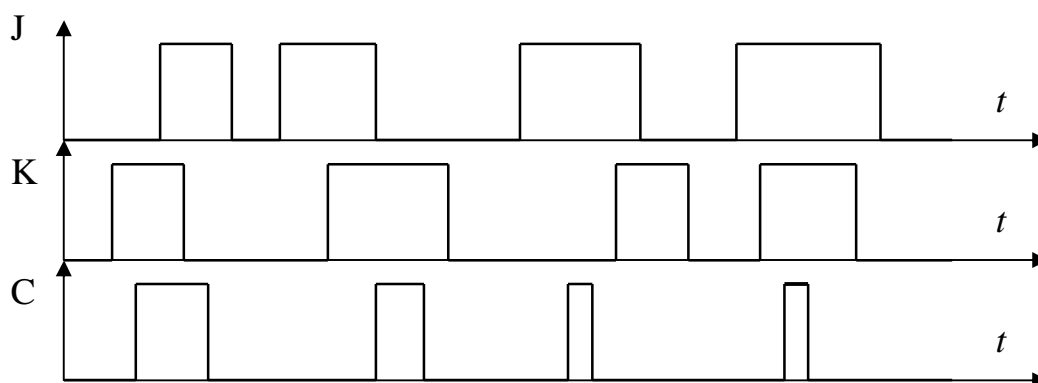


Рис. 2.24.

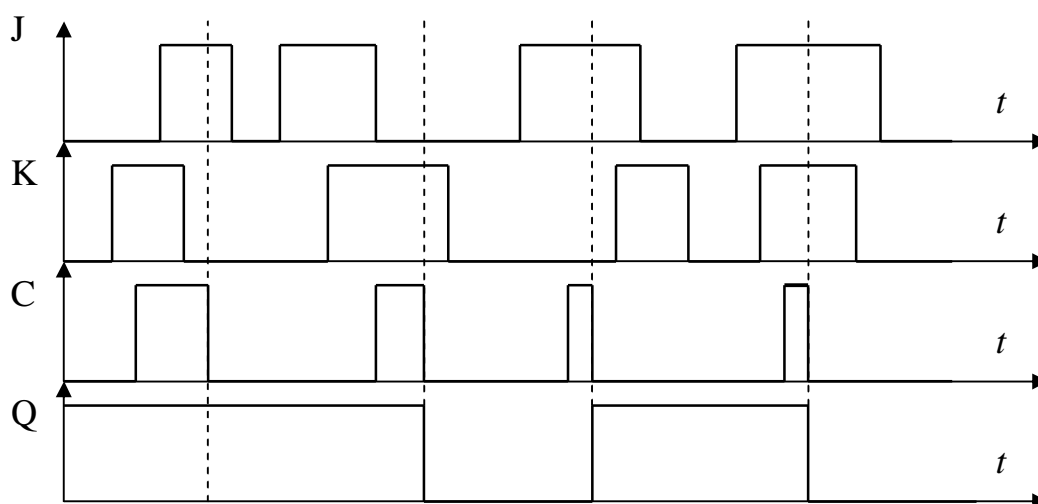


Рис. 2.25.

2.2.2. СЧЕТЧИКИ

ЦУ, содержащее несколько триггеров, состояние которых определяется числом поступивших на вход устройства импульсов, называют **счетчиком**.

Различают суммирующий, вычитающий и реверсивный счетчики.

У *суммирующего* счетчика выходное число, хранимое в счетчике, увеличивается при поступлении на его вход первого импульса. Если первый входной импульс уменьшает выходное число, то счетчик - *вычитающий*. *Реверсивный* счетчик может переключаться из режима суммирования в режим вычитания и наоборот. Максимальное число состояний счетчика называют его модулем (коэффициентом пересчета, основанием) M. Если счетные импульсы поступают одновременно на входы всех используемых триггеров, то счетчик называют *параллельным*. В *последовательном* счетчике каждый последующий триггер управляется выходным сигналом предыдущего.

Обычно в счетчиках используют JK-триггеры или триггеры типа T с установочными асинхронными R и S входами. Используя эти дополнитель-

ные входы и КС, можно осуществить начальную установку некоторого исходного заданного кода, а также осуществить сброс показаний счетчика при достижении другого заданного двоичного кода.

Счетчики широко используются в компьютерах для получения последовательности адресов команд, для деления частоты задающего кварцевого генератора, а вместе с дешифраторами - для генерации заданной последовательности логических сигналов и т. п.

Как следует из принципа работы Т-триггера, в этом случае с приходом очередного синхроимпульса триггер будет каждый раз изменять свое состояние на противоположное. Если к выходу такого триггера подключить еще один Т-триггер, последний будет подсчитывать количество единичных состояний первого. Соединив последовательно N триггеров типа Т, получим суммирующий бинарный счетчик с основанием $M=2^N$, где N - число двоичных разрядов (триггеров) счетчика.

Схема такого четырехразрядного счетчика на Т-триггерах приведена на рис. 2.26. На приведенном рисунке x_0 – вход счетчика, на который поступает последовательность электрических импульсов. Асинхронные входы R служат

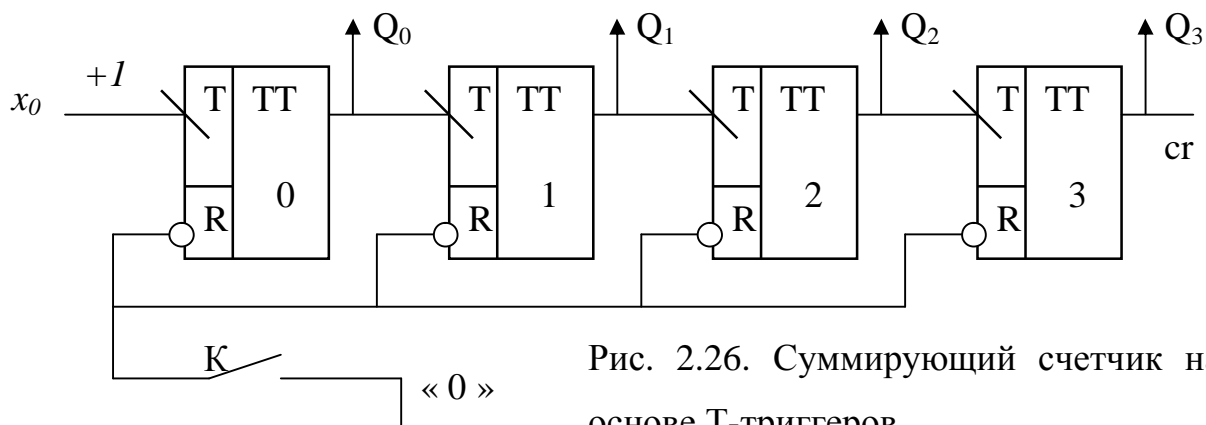


Рис. 2.26. Суммирующий счетчик на основе Т-триггеров.

для установки начального состояния триггеров, т.е. для обнуления выходов Q_n . Это происходит при подаче на установочные входы триггеров сигнала, соответствующего потенциалу логического нуля (замыкание ключа К). Временные диаграммы и таблица состояний на рис. 2.27 поясняют работу этого счетчика. Как видно, переключение происходит последовательно, с каждым поступающим импульсом. Поэтому счетчики с непосредственным соединением триггеров называют счетчиком с последовательным переносом. В рассматриваемом случае предполагается, что в исходном состоянии показания счетчика равны нулю. Как видим, M -й импульс возвращает показания четырехразрядного счетчика в исходное нулевое состояние. Достоинством таких счетчиков является предельная простота их структуры и легкость наращивания разрядности. К недостаткам последовательных счетчиков можно отнести их низкое быстродействие, которое зависит от количества триггеров в счетчике. При большой разрядности последовательного счетчика его быстродействие может оказаться очень низким.

Радикальным способом повышения быстродействия счетчика является организация одновременного переключения всех триггеров, входящих в структуру счетчика, под воздействием входного сигнала. Именно такой процесс происходит в параллельных счетчиках.

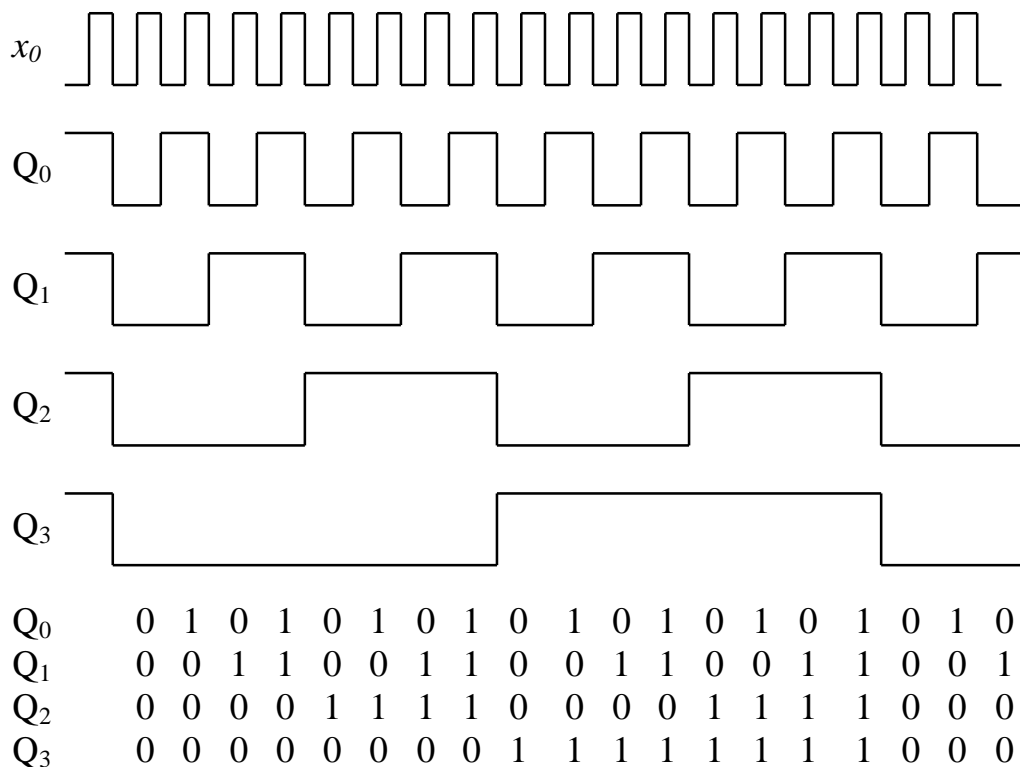


Рис. 2.27. Временные диаграммы и таблица состояний суммирующего счетчика.

В параллельных счетчиках счетные импульсы подаются на все разряды одновременно, и изменение состояния данного разряда происходит только при определенном состоянии всех предыдущих разрядов. Например, структурная схема счетчика с параллельным переносом показана на рис. 2.28. Пусть в исходном состоянии в счетчике записан код 000. После первого счетного импульса сигнал Q_0 , станет равным 1, и при этом подготовится к срабатыванию схема совпадения 1. Второй счетный импульс вернет Q_0 в ис-

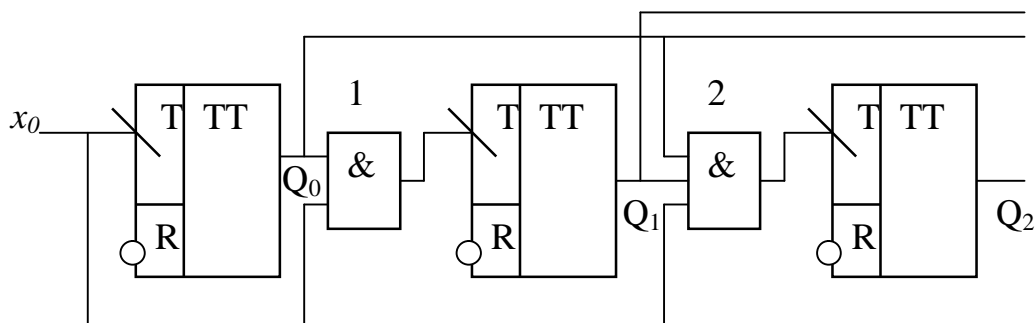


Рис. 2.28. Параллельный счетчик.

ходное состояние и через схему совпадения 1 пройдет на триггер второго разряда. Схема совпадения 2 закрыта, так как $Q_1 = 0$. Таким образом, после

второго счетного импульса $Q_0 = 0, Q_1 = 1, Q_2 = 0$. После третьего счетного импульса в счетчике будет зафиксирован код 011. Теперь в состоянии, открытом для прохождения счетных импульсов, будут находиться обе схемы совпадения. В результате четвертый импульс поступит на все три разряда и установит счетчик в состояние 100.

Счетчики с исключением старших состояний. Такими счетчиками являются счетчики с произвольным коэффициентом пересчета M , начальное состояние которых равно нулю. Кроме таких счетчиков различают также и счетчики с исключением *младших состояний*, в которых начальное состояние отлично от нуля. Применение последних в электронных схемах является ограниченным вследствие сложной схемы формирования начального состояния, а также неестественной последовательности пробегающих значений (через некоторый интервал значений). Поэтому такие счетчики применяются там, где сами показания счетчика не используются (например, в делителях частоты).

Счетчики с исключением старших состояний имеют естественную последовательность состояний. На рис. 2.29 (а) показана ФС двоично-десятичного счетчика ($M = 10$), а на рис. 2.29 (б) – временные диаграммы, поясняющие его работу.

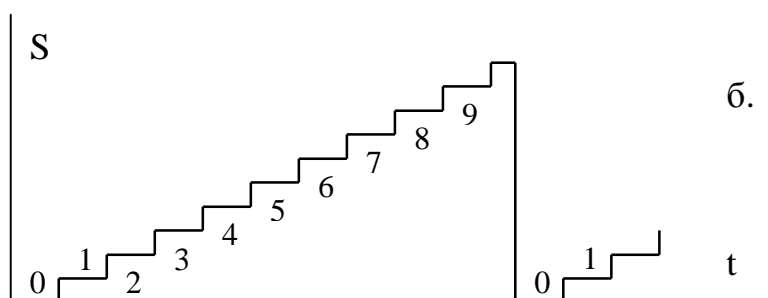
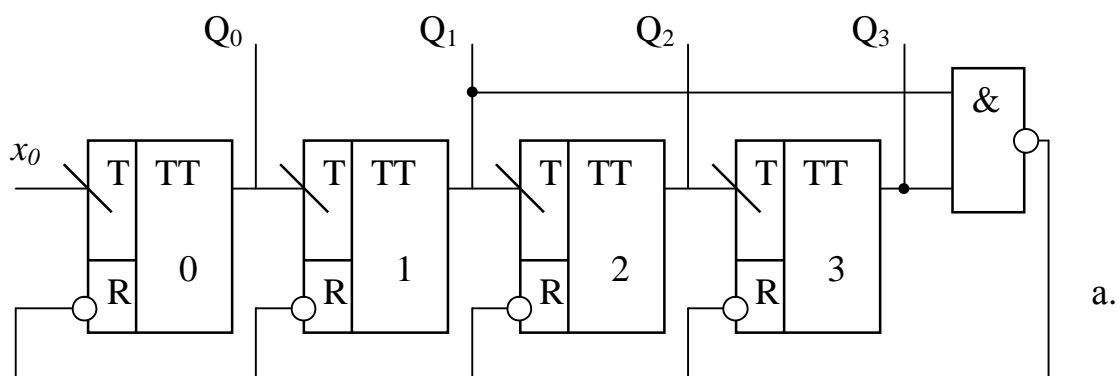


Рис. 2.29. Функциональная схема двоично-десятичного счетчика (а) и его временные диаграммы (б).

В каждом из состояний $S = (0, 1, 2, \dots, 9)$ счетчик находится в течение периода следования счетных импульсов. При $S = 10$ на выходе схемы DD формируется значение $y = 0$, которым счетчик сбрасывается в нулевое состояние. Длительность пребывания счетчика в состоянии $S = 10$ составляет

$$t_{\text{сбр.}} = \tau_{\text{DD}} + \tau_{\text{R}},$$

где τ_{DD} – задержка в схеме DD; τ_R – время срабатывания счетчика по сигналу на асинхронном установочном входе. Считается, что $t_{сбр.}$ много меньше периода следования счетных импульсов, и потребитель результатов счета не успевает среагировать на показание $S = 10$. Для реализации такого условия необходимо правильно подбирать ЛЭ (по быстродействию), составляющие счетчик, в соответствии со схемой работы ЦУ.

Пример 2.3. Используя Т-триггеры построить последовательный счетчик с коэффициентом пересчета $M = 25$.

Решение: 1. Определить минимальное количество триггеров N , входящих в состав счетчика. Для этого можно использовать следующее условие:

$$M < 2^N.$$

Из указанного условия видно, что $N = 5$ ($2^5 = 32$).

2. Используя N -разрядное двоичное число определить изображение десятичного числа M в двоичной системе счисления. В данном случае $N = 5$, $M = 25$, тогда $A_{(2)} = 11001$.

3. Определить для каждого из значений Q_n соответствующий разряд в двоичном слове $A_{(2)}$.

$A_{(2)}$	1	1	0	0	1
	Q_4	Q_3	Q_2	Q_1	Q_0

4. Соединить выходы триггеров Q_n , соответствующие единичным значениям с ЛЭ «НЕ».

Решение представлено на рис. 2.30.

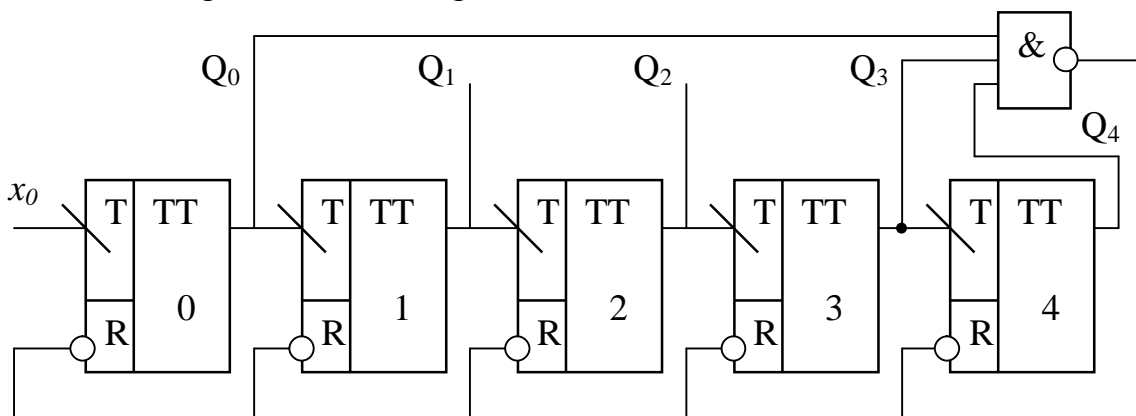


Рис. 2.30. Функциональная схема последовательного счетчика с коэффициентом $M = 25$.

Задания для самостоятельного решения.

1. Используя тождества БА, построить ФС следующих КС:

а. $Y = (x_1 \cdot x_0) \vee (x_1 \cdot \bar{x}_0) \vee (x_2 \cdot x_0)$

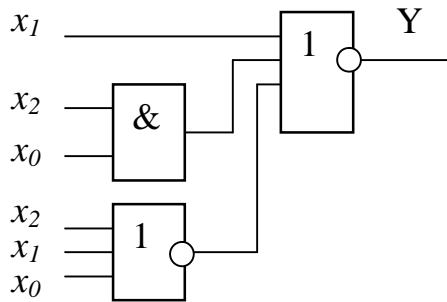
б. $Y = (x_2 \vee (x_2 \cdot \bar{x}_1)) \cdot \bar{x}_1$

в. $Y = ((x_2 \vee x_1 \vee x_0) \cdot (x_2 \vee \bar{x}_1 \vee x_0) \cdot (\bar{x}_2 \vee x_0)) \cdot x_2$

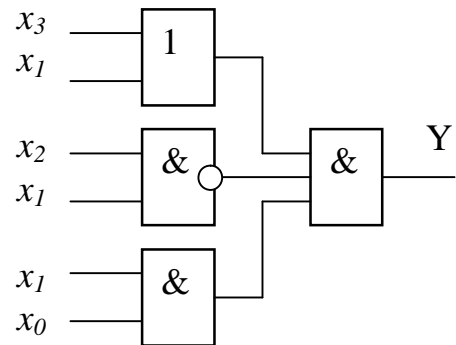
г. $Y = ((x_3 \cdot x_2 \cdot x_0) \vee (\bar{x}_1 \cdot x_0)) \cdot (x_3 \vee \bar{x}_0)$

2. Составить КС следующих ФС:

а.



б.



3. Оценить стоимость КС по шкале Квайна:

а. $Y = (x_1 \cdot x_0) \vee (x_2 \cdot \bar{x}_1) \vee x_0$

б. $Y = (x_3 \cdot (\bar{x}_2 \cdot x_0)) \cdot x_0$

в. $Y = (x_2 \vee x_0) \vee (x_1 \vee \bar{x}_0)$

г. $Y = (x_1 \vee x_0) \cdot (x_2 \vee \bar{x}_0) \cdot (\bar{x}_3 \vee x_0)$

4. Получить КС минимальной стоимости и составить их ФС, если ТИ ПФ имеют вид:

а.

α	0	1	2	3	4	5	6	7
Y	1	0	1	0	1	0	1	1

б.

α	0	1	2	3	4	5	6	7
Y	0	1	0	1	0	1	0	0

в.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1

г.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	1	0	1	1	1	0	1	0	0	1	1	0	0	0	1	1

5. Получить КС минимальной стоимости и составить их ФС с использованием указанных ЛЭ (без ограничения числа выводов), если ТИ ПФ имеют вид:

а.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

При построении ФС использовать ЛЭ Пирса.

б.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	1	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1

При построении ФС использовать ЛЭ Шеффера.

в.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	0	1	1	1	0	0	1	0	1	0	0	0	1	0	1	1

При построении ФС использовать ЛЭ Пирса.

6. Получить КС минимальной стоимости и составить их ФС с использованием указанных ЛЭ (с ограниченным числом выводов), если ТИ ПФ имеют вид:

а.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0	1

При построении ФС использовать ЛЭ «ЗИЛИ-НЕ».

б.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Υ	1	0	1	0	0	1	0	1	1	0	1	1	1	0	0	1

При построении ФС использовать ЛЭ «ЗИ-НЕ».

7. Синтезировать ФС преобразователя произвольных кодов в соответствии с заданной таблицей преобразования:

а.

α_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
α_2	5	1	2	0	4	14	13	7	6	8	3	5	11	10	15	9

В таблице α_1 – набор входного кода X, α_2 – набор входного кода Y.

При построении ФС использовать базис «И-НЕ» (1. без ограничения количества входов ЛЭ; 2. «ЗИ-НЕ»).

б.

α_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
α_2	15	0	2	1	7	13	13	7	10	8	2	5	11	3	12	6

В таблице α_1 – набор входного кода X, α_2 – набор входного кода Y.

При построении ФС использовать базис «ИЛИ-НЕ» (**1.** без ограничения количества входов ЛЭ; **2.** «ЗИЛИ-НЕ»).

в.

α_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
α_2	15	14	13	0	1	2	3	12	10	11	9	8	7	4	6	5

В таблице α_1 – набор входного кода X, α_2 – набор входного кода Y.

При построении ФС использовать базис **1.** «ЗИЛИ-НЕ»; **2.** «ЗИ-НЕ».

8. ПФ задана таблицей преобразования. Составить ФС с использованием мультиплексора, реализующую данную ПФ.

а.

α	0	1	2	3	4	5	6	7
Y	0	1	1	0	0	1	0	1

б.

α	0	1	2	3	4	5	6	7
Y	1	1	0	0	1	1	1	0

в.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y	0	1	1	0	1	0	0	0	1	0	1	1	0	0	0	1

г.

α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Y	1	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0

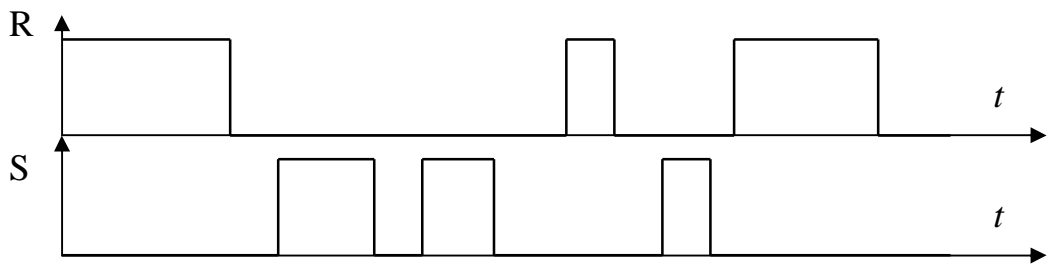
9. Составить ФС последовательного сумматора для сложения N двоичных чисел с разрядностью n.

а. N = 2, n = 2. **б.** N = 3, n = 2. **в.** N = 2, n = 3. **г.** N = 3, n = 3.

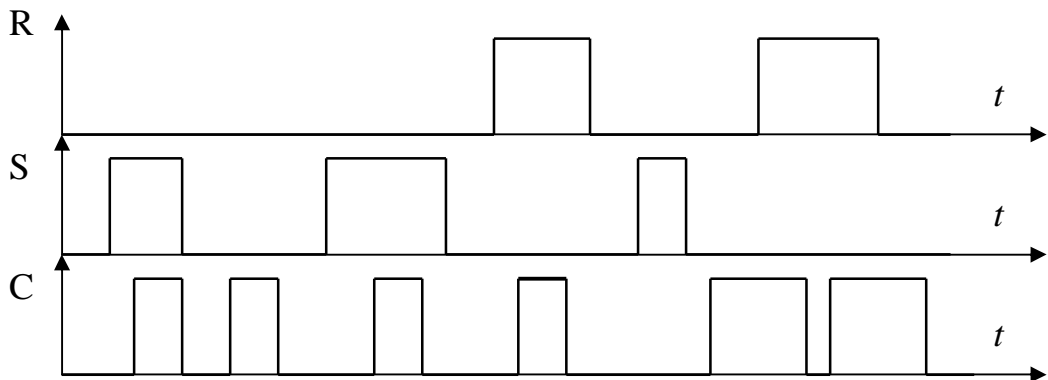
Составить для каждого из вариантов ТИ сумматора.

10. Построить временные диаграммы выходных сигналов Q и \bar{Q} для следующих входных сигналов триггеров:

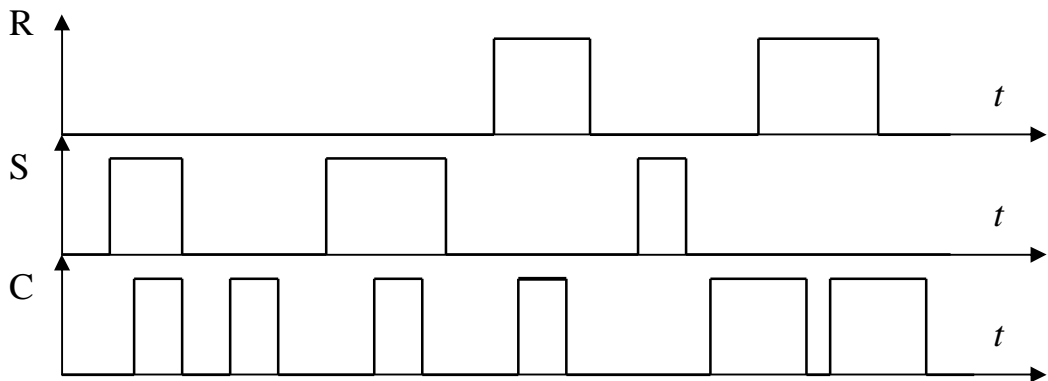
а. Асинхронный RS-триггер:



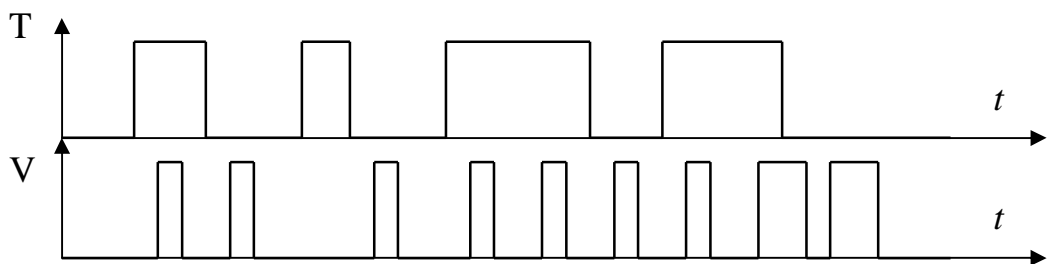
б. Синхронный одноступенчатый RS-триггер:



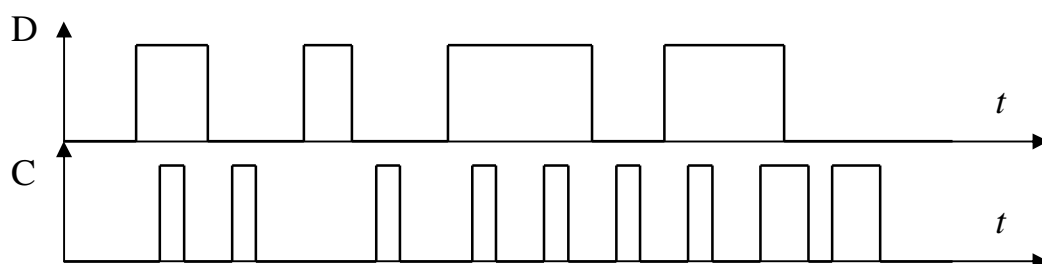
в. Синхронный двухступенчатый RS-триггер:



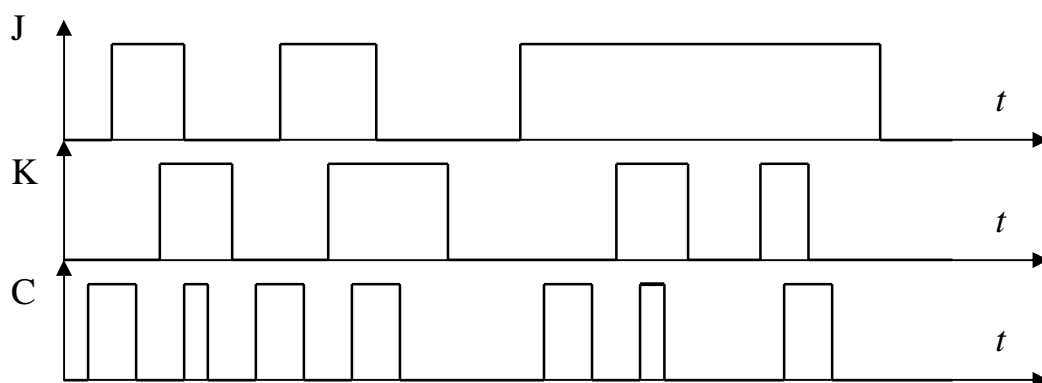
г. Синхронный T – триггер:



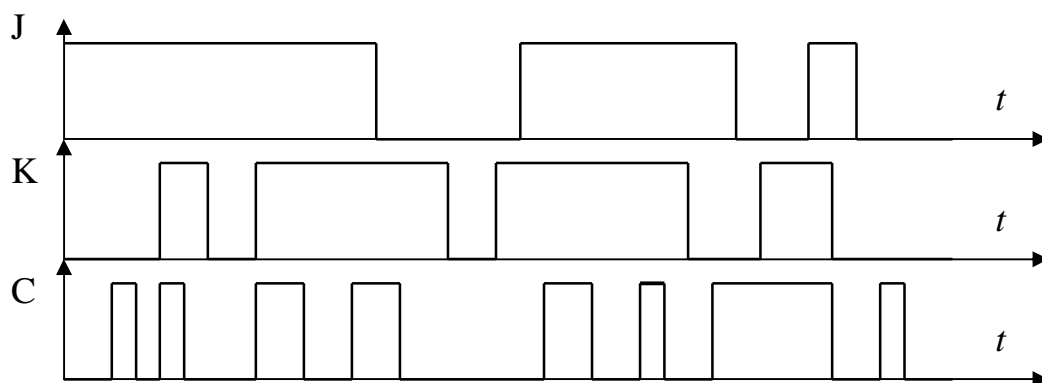
д. Синхронный D – триггер:



е. JK – триггер:



ж. JK – триггер:



11. Построить ФС последовательного счетчика с коэффициентом пересчета М:

а. $M = 6$;

б. $M = 12$;

в. $M = 17$;

г. $M = 34$.

Список использованных сокращений

БА – булева алгебра;
ДВ – диаграмма Вейча;
ИС – интегральная схема (микросхема);
КС – комбинационная схема;
ЛЭ – логический элемент;
СДНФ – совершенная дизъюнктивная нормальная форма;
СКНФ – совершенная конъюнктивная нормальная форма;
ПФ – переключательная функция;
ТИ – таблица истинности;
ФС – функциональная схема;
УГО – условное графическое обозначение;
ЦУ – цифровое устройство.

Список литературы

1. Бечева М.К. и др. Электротехника и электроника. М., Высшая школа, 1991.
2. Евреинов Э.В. и др. Цифровая и вычислительная техника. М., Радио и связь, 1991.
3. Ицхоки Я.С. Овчинников И.И. Поздняков В.Г. Импульсные и цифровые устройства. Учебник для инженерных вузов. М., ВВИА им. Н.Е. Жуковского, 1992.
4. Кучумов А. Электроника и схемотехника. М., 2002.
5. Потемкин И.С. Функциональные узлы цифровой автоматики. М., Энергоатомиздат, 1988.
6. Ямпольский В.С. Основы автоматики и электронно-вычислительной техники. М., Просвещение, 1991.