

Министерство общего и профессионального образования
Российской Федерации

Владимирский государственный университет

А.К. Бернюков А.И. Никитин

ЦИФРОВЫЕ УСТРОЙСТВА

Учебное пособие

Владимир 2000

УДК 621.3.049.77.037.372.001.63(035):681.3(076.5)

Бернюков А.К., Никитин А.И.

Цифровые устройства: Учебное пособие /Владим. гос. ун-т. Владимир, 2000. ... с. ISBN .—.....—...—.

Включены материалы по теории цифровых устройств: основы двоичной алгебры, вопросы анализа и синтеза цифровых конечных автоматов без памяти (комбинационных цифровых схем) и с памятью (последовательностных устройств). Приведены примеры практических схем цифровых устройств, имеющих широкое использование при создании систем цифровой обработки информации на современной микроэлектронной элементной базе, включая узлы микропроцессорных систем. Используются результаты научных исследований авторов и опыт чтения лекций по тематике цифровых устройств и микропроцессоров.

Пособие предназначено для студентов радиотехнических специальностей: 200700 – радиотехника, 071500 – радиофизика и электроника, 201500 – бытовая радиоэлектронная аппаратура.

Табл. ... Ил. Библиогр.: ... назв.

Печатается по решению редакционно-издательского совета Владимирского государственного университета.

Рецензенты: - заведующий кафедрой радиосистем Санкт-Петербургского государственного электротехнического университета, доктор технических наук профессор В.П. Ипатов;

- доктор технических наук профессор Г.А. Андреев (Институт радиотехники и электроники Российской Академии наук).

Предисловие

Известные преимущества цифровых устройств перед аналоговыми (точность, помехоустойчивость, многофункциональность) обусловили широчайшее распространение цифровых методов и средств практически во всех отраслях человеческой деятельности – от космических исследований, радиосвязи, радиолокации и телевидения до медицинских, бытовых приборов и детских игр. Естественно, в учебные планы подготовки специалистов различных специальностей внедряются дисциплины, связанные с разработкой и созданием цифровых приборов и систем, что побуждает развивать учебно-методическую базу в этом направлении.

Теории и практике цифровых методов и средств посвящено огромное количество работ в России и за рубежом, включая и фундаментальные, часть из которых приведена в библиографическом перечне. Однако необходимость компактного изложения материала с учетом ограниченности выделяемых на обучение часов, специфики обучаемой аудитории, “окраски” применений цифровых методов в различных областях (например, в радиолокации и медицине), а также особенности используемой учебно-лабораторной базы побуждают к необходимости издания новых пособий по данному направлению.

Данное учебное пособие имеет целью реализовать указанные отчасти противоречивые требования. Оно содержит вопросы теоретического курса, включая классические вопросы двоичной алгебры, анализа и синтеза логических устройств без памяти (комбинационных схем) и с памятью (последовательностных схем). Теоретический материал сопровождается описанием практических цифровых устройств, имеющих инженерное применение в различных отраслях, связанных с радиоэлектроникой. К их числу относятся преобразователи кодов (шифраторы и дешифраторы), элементы памяти (триггеры), регистры и счетчики, коммутаторы (мультиплексоры и демультимплексоры), устройства формирования импульсных и цифровых сигналов, элементы микропроцессорной техники (арифметические и запоминающие устройства), преобразователи информации “аналог-код” и обратно. Все рассматриваемые устройства реализованы на современной микроэлектронной базе.

Обучение предмету осуществляется комплексно на основе прослушивания лекций по теоретическому курсу, самоподготовки к лабораторным занятиям (задания и контрольные вопросы включены в пособие) и выполнения лабораторных работ во фронтальном режиме.

Пособие написано на основе многолетнего преподавания авторами дисциплин “Цифровые и импульсные устройства”, “Микропроцессоры и микроЭВМ” и “Дискретная и цифровая обработка сигналов”.

Глава 1. Общие методы представления и преобразования информации в цифровых вычислительных устройствах

1.1. Системы счисления

Совокупность приемов наименования и записи чисел называется *счислением*. Под *системой счисления* понимается способ представления любого числа с помощью ограниченного алфавита символов, называемых цифрами. Счисление представляет собой частный случай кодирования, где слово, записанное с использованием определенного алфавита и по определенным правилам, называется *кодом*. Применительно к счислению это *код числа*.

Различают:

- *непозиционные* системы счисления (например, римская), где в любом месте числа каждой цифре соответствует *одно и то же* значение;
- *позиционные* системы счисления, где одна и та же цифра имеет *разное значение*, которое определяется ее позицией в последовательности цифр, изображающей число. Количество символов S , употребляемое в позиционной системе счисления, называется ее *основанием*. Позиционной является *десятичная* (D – decimal) система счисления, для которой $S = 10$, а для записи используется алфавит из 10 цифр (0, 1, 2, ..., 9).

В позиционной системе счисления каноническая форма представления числа X_S выглядит следующим образом:

$$X_S = A_{k-1}S^{k-1} + A_{k-2}S^{k-2} + \dots + A_1S^1 + A_0S^0 + A_{-1}S^{-1} + \dots + A_{-m}S^{-m} = \sum_{i=-m}^{k-1} A_i S^i \quad (1.1)$$

где X_S – запись числа в системе счисления с основанием S ;

S – основание системы счисления;

A_i – символы, составляющие алфавит системы счисления, меньше S ;

k – число разрядов (позиций) в целой части числа;

m – число разрядов (позиций) в дробной части числа.

Изображение числа X_S в виде последовательности коэффициентов A_i полинома (1.1) является его условной сокращенной записью (кодом)

$$X_S = A_k A_{k-1} \dots A_1 A_0 . A_{-1} A_{-2} \dots A_{-m}$$

(1.2)

Точка отделяет целую часть числа от дробной и служит началом отсчета значений веса каждой позиции (разряда).

В ЦВУ также получили использование позиционные системы счисления с недесятичным основанием: *двоичная, восьмеричная и шестнадцатеричная*, для которых основание $S = 2^k$, $k = 1, 3, 4$.

В *двоичной* (B – binary) системе счисления основание $S = 2$ и алфавит $\{A_i\}$ состоит из двух символов – 0 и 1.

В *восьмеричной* (Q – octal) системе счисления алфавит $\{A_i\}$ состоит из восьми символов 0, 1, 2, ..., 7. Основание системы счисления $S = 8$.

В *шестнадцатеричной* (H - hexadecimal) системе счисления алфавит $\{A_i\}$ включает в себя 16 символов (букв и цифр) 0, 1... 9, A, B, C, D, E, F. Основание системы счисления $S = 16$.

Пример. $254D = 11111110B = 376Q = FEH$.

Из примера следует, что наиболее сжатая получается запись в шестнадцатеричной системе счисления, а наибольшего числа разрядов требует двоичное представление числа. Следует вместе с тем отметить, что ЦВУ (ЭВМ) воспринимает информацию только в виде двоичных кодов. В табл. 1 приведены коды одних и тех же чисел в разных системах счисления.

Наряду с двоичными кодами, которыми оперирует ЦВУ, для ввода и вы-

Таблица 1

X_{10}	0	1	2	3	4	5	6	7	8
X_2	0	1	10	11	100	101	110	111	1000
X_8	0	1	2	3	4	5	6	7	10
X_{16}	0	1	2	3	4	5	6	7	8
X_{10}	9	10	11	12	13	14	15	16	17
X_2	1001	1010	1011	1100	1101	1110	1111	10000	10001
X_8	11	12	13	14	15	16	17	20	21
X_{16}	9	A	B	C	D	E	F	10	11

вода десятичных чисел используют специальное *двоично-десятичное кодирование*, при котором каждая десятичная цифра заменяется тетрадой двоичных цифр, а сами тетрады записываются в последовательно в соответствии с порядком следования десятичных цифр. При обратном преобразовании двоично-десятичного кода в десятичный исходный код разбивается на тетрады вправо и влево от запятой, которые затем заменяются десятичными цифрами.

При двоично-десятичном кодировании фактически не производится перевод числа в новую систему счисления, а используется двоично-кодированная десятичная система счисления.

Пример. Десятичное число $95_{10} = 1001\ 0101_{2-10}$.

1.2. Перевод чисел из одной системы счисления в другую

ЦВУ работают с двоичными кодами, пользователю удобнее иметь дело с десятичными или шестнадцатеричными. Поэтому возникает необходимость перевода числа из одной системы счисления в другую. В общем случае перевод числа, содержащего *целую* и *дробную* часть, выполняется по универсальному алгоритму (схеме Горнера).

Пусть число X_S с основанием S необходимо представить, как число X_R с основанием R

$$X_{SQ} \cdot X_{SD} = X_{RQ} \cdot X_{RD} \text{ или } X_{SQ} = X_{RQ}, X_{SD} = X_{RD}.$$

Из канонической формы записи числа X_R с основанием R и символами в записи $\{B_j\}$ следует для целой и дробной частей:

$$X_{RC} = ((\dots(B_n R + B_{n-1})R + B_{n-2})R + \dots + B_1)R + B_0,$$

$$X_{RD} = R^{-1}(B_{-1} + R^{-1}(B_{-2} + \dots + R^{-1}(B_{-(p-1)} + R^{-1}B_{-p}))) \dots$$

Перевод целой части X_{RC} и получение символов B_j производится делением числа на основание R новой системы счисления. Преобразование дробной части – умножением X_{RD} на основание R . Эти действия необходимо производить в системе счисления с основанием S . Остатки при делении представляют символы B_j в целой части числа X_{RC} , а целые части при умножении – символы B_j в дробной части числа X_{RD} .

Пример. Перевести целое число $X_{10} = 98D$ в двоичную систему счисления.

$98 \mid 2$	
$V_0 = 0 \quad 49 \mid 2$	$X_2 = 1100010B$
$V_1 = \rightarrow 1 \quad 24 \mid 2$	
$V_2 = \rightarrow 0 \quad 12 \mid 2$	
$V_3 = \rightarrow 0 \quad 6 \mid 2$	
$V_4 = \rightarrow 0 \quad 3 \mid 2$	
$V_5 = \rightarrow 1 \quad 1 \mid 2$	
$V_6 = \rightarrow 1$	

Проверка дает:
 $2^6 + 2^5 + 2^1 = 64 + 32 + 2 = 98_{10}$

Пример. Перевести правильную дробь $X_{10} = 0.625D$ в двоичную форму.

$0.625 \mid *2$	Последовательно умножаем на $R = 2$, при этом получающиеся целые части B_j не участвуют в последующих умножениях.
$V_{-1} = 1.250 \mid *2$	
$V_{-2} = 0.500 \mid *2$	
$V_{-3} = 1.000 \mid \times$	

Проверка дает:
 $V_{-1}V_{-2}V_{-3} = 0.101B = 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 1/2 + 1/8 = 0.625D$

Перевод из двоичной в десятичную систему сложнее, т.к. надо делить (умножать) на 1010_2 в двоичной системе счисления, а результаты переводить в десятичную систему. Проще использовать представление числа в канонической форме, например, $10101B = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 = 21D$.

Переводы чисел в системах счисления с кратным основанием ($2, 8, 16$) сравнительно просты, т.к. умножение и деление на 2^i ($i = 1, 3, 4$) эквивалентны переносу точки, отделяющей целую часть от дробной, влево или вправо на i позиций (рис. 1). При этом целую часть числа можно слева дополнять незначащими нулями, а дробную часть можно справа дополнять незначащими нулями до полных триад или тетрад.

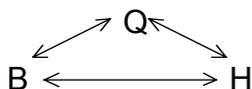


Рис. 1

Для перевода из двоичной в восьмеричную или шестнадцатеричную систему счисления необходимо в двоичной записи числа произвести разбиение на триады (тетрады) символов влево и вправо, начиная от точки между дробной и целыми частями, после чего заменить каждую триаду (тетраду) двоичных

символов ее эквивалентом в $Q(H)$ - системе счисления. В обратных преобразованиях ($Q \rightarrow B$ и $H \rightarrow B$) следует просто заменить символы в записи числа на их двоичные эквиваленты (табл. 1).

Преобразования $Q \rightarrow H$ и $H \rightarrow Q$ проще производить с промежуточным использованием двоичной записи.

Пример. Двоичное число преобразовать в восьмеричное и шестнадцатеричное

$$B \rightarrow Q(H):$$

$$\overbrace{0010111101101100100101.101011101100}^B = \\ = 1355445.5354Q = 5DB25.AECH$$

1.3. Формы представления чисел в ЦВУ

В ЦВУ используются следующие формы представления данных:

- ◇ числа с фиксированной точкой (запятой);
- ◇ числа с плавающей точкой (запятой);
- ◇ десятичные числа;
- ◇ символьные данные.

При представлении числа X с **фиксированной точкой** указываются знак числа ($sign X$) и модуль числа ($mod X$) в S -ичном коде. Место точки постоянно для всех чисел и в процессе решения задач не меняется. Знак положительного числа кодируется цифрой "0", а знак отрицательного числа – цифрой "1". В общем случае числа с фиксированной точкой выглядят следующим образом (рис. 2):

Целая часть числа со знаком					Дробная часть числа				
Знак	S^{n-1}	...	S^1	S^0	•	S^{-1}	S^{-2}	...	S^{-m}

Рис. 2

Диапазон значений X для заданных n и m определяется неравенством

$$S^m \leq |X| \leq S^n - S^m.$$

Пример. Для двоичного целого числа размерностью один байт (8 бит со знаком) диапазон представления находится в пределах ± 127 .

Форма представления числа с фиксированной точкой упрощает аппаратную реализацию ЦВУ, но при вычислениях необходимо следить за тем, чтобы не был превышен допустимый диапазон представления чисел, что влечет за собой переполнение разрядной сетки и ошибки в результатах. От этого недостатка в значительной степени можно избавиться, используя форму представления чисел с **плавающей точкой** или **нормальную форму**.

В случае с плавающей точкой числа представляются в виде $X = \pm p.\pm q$, что соответствует записи $X = S^{+p}(\pm)q$,

где p – целое число, выражающее порядок;

S – основание системы ($S = 2^i, i=1,3,4$);

q – мантисса числа (значащая часть), $|q| \leq 1$.

Разрядная сетка для размещения чисел в нормальной форме показана на рис. 3, где n – число разрядов модуля порядка, m – число разрядов модуля мантиссы.

Порядок числа со знаком				Мантисса числа со знаком					
Зн p	S^{n-1}	...	S^1	S^0	Зн q	S^{-1}	S^{-2}	...	S^{-m}

Рис. 3

Диапазон значений X для заданных n и m определяется неравенством

$$S^{-m} S^{-(S^n-1)} \leq |X| \leq (1 - S^{-m}) S^{(S^n-1)} .$$

Нормальная форма представления чисел неоднозначна, ибо взаимное изменение p и q приводит к плаванию точки. Для однозначности представления часто используют **нормализованную** запись, при которой старший разряд мантиссы отличен от 0 ($1/S \leq |q| < 1$), т.е. положение точки задается перед значащей цифрой мантиссы.

В некоторых случаях для упрощения операций над порядками, используют “смещённый” порядок, т.е. к порядку p добавляют $M = 2^k$, с тем, чтобы порядки $p+M$ были всегда целыми и положительными числами.

В ЦВУ многоразрядные **десятичные числа** чаще всего представлены в двоично-кодированной десятичной системе счисления *BCD (binary coded decimal)*, для чего в десятичной записи числа каждая цифра заменяется на тетраду (двоичный эквивалент) двоичных цифр (от 0000 до 1001 – 0÷9). Такое представление чисел характерно для задач коммерческого плана, в которых имеются большие массивы входных данных и результатов в десятичной форме. Использование *BCD*-кодирования позволит избежать больших потерь времени на перевод чисел из десятичной системы в двоичную и обратно.

В так называемом **упакованном** формате *BCD* в каждом байте хранится две цифры числа (две тетрады – старшая и младшая), например, $78D \rightarrow 0111\ 1000B$.

В **неупакованном (зонном)** формате каждая десятичная цифра занимает младшую тетраду бит, в старшей тетраде – обычно 0, т.е. $78D \rightarrow 0000\ 0111\ 0000\ 1000B$. Для обработки информации, представленной двоично-десятичными числами (*BCD*) предусмотрены специальные команды.

В ЭВМ (ЦВУ) обрабатывается не только числовая информация, но и **символьные данные**. Как и любая другая информация, символьные данные должны храниться в памяти ЭВМ в двоичном коде. Для этого каждому символу ставится в соответствие некоторое неотрицательное двоичное число, называемое кодом символа. Конкретное соответствие между символами и их кодами называется **системой кодировки**. Для кодирования символьных данных (алфавитно-цифровой информации) обычно используется *ASCII* код (*American Standard Code for Information Interchange* – американский код об-

мена информацией). В каждой стране обычно есть варианты этой кодировки (в России – КОИ-7, ДКОИ-8, альтернативная кодировка ГОСТ и др.). Отметим лишь особенности этих систем кодировок:

- ⇒ восьмой (девятый) разряд в таких кодах может служить для проверки на четность (нечетность) – *parity*;
- ⇒ код пробела меньше кода любой буквы и цифры и вообще меньше любого графического представления символа;
- ⇒ коды цифр упорядочены по возрастанию и идут без пропусков, т. е. если '0' ≤ 'код(C)' ≤ '9', то C - цифра, таким образом, если код '0' = 30H, то код 'i' = код '0' + i, i = 0÷9;
- ⇒ коды больших латинских букв также идут без пропусков, код 'A' = 41H, код 'B' = 42H и т.д.; это также верно и для малых латинских букв, 'a' = 61H, 'b' = 62H и т.д.

1.4. Выполнение арифметических операций. Специальные машинные коды

Основной операцией, которая используется в вычислениях, является операция алгебраического сложения чисел, причем, в конечном счете эта операция сводится к простому сложению, для чего используются специальные машинные коды. Для хранения чисел и выполнения различных операций над ними их представляют различными кодами: **прямым, обратным и дополнительным**. Введем следующие правила определения кодов чисел, используемых в ЦВУ.

Прямой код S-ичного числа (1.2):

$$X_{Snp} = 0 A_{k-1} A_{k-2} \dots A_1 A_0 \bullet A_{-1} \dots A_{-m}, \text{ если } X_S \geq 0;$$

$$X_{Snp} = 1 A_{k-1} A_{k-2} \dots A_1 A_0 \bullet A_{-1} \dots A_{-m}, \text{ если } X_S < 0,$$

где A_i – значение цифры в i -ом разряде исходного кода.

Здесь старший бит несет информацию о знаке числа. Если он равен 0, то число положительное, если его значение 1, то число отрицательное.

Пример. Для двоичного кода $X_2 = +101_2 = +5_{10}$ прямой код $X_{2пр} = 0101$; для $X_2 = -101_2 = -5_{10}$ прямой код $X_{2пр} = 1101_2$.

Недостаток представления чисел в прямых кодах – наличие ± 0 , а также сложность алгебраических операций в прямом коде с различными знаками. В этом случае приходится определять большее по модулю число, производить вычитание чисел и присваивать разности знак большего по модулю числа. С целью упрощения алгебраического сложения используют **специальные коды (обратный и дополнительный)** для представления **отрицательных чисел**. Специальный код положительного числа равен его прямому коду.

Обратный (инверсный) код S-ичного числа:

$X_{S \text{ об } p} = 1 \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_1 \bar{A}_0 \cdot \bar{A}_{-1} \dots \bar{A}_{-m}$, если $X_S < 0$.

Здесь инверсия цифры A_i определяется из соотношения:

$$\bar{A}_i = (S - 1) - A_i, \quad -m \leq i \leq n - 1.$$

Для двоичной системы счисления, если $A_i = 1$, то $\bar{A}_i = 0$ и наоборот. Таким образом, для преобразования прямого кода двоичного отрицательного числа в обратный код и наоборот необходимо знаковый разряд оставить без изменения, а в остальных разрядах нули заменить на единицы, а единицы на нули.

Пример. Пусть для целого ($n=5$) отрицательного числа $X_S = -18_{10} = -10010_2$ прямой код $X_{2пр} = 110010_2$. Получим обратный код числа $X_{2обр} = 101101_2$.

Дополнительный код S -ичного числа:

$$X_{S \text{ доп } n} = 1 \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_1 \bar{A}_0 \cdot \bar{A}_{-1} \dots \bar{A}_{-m} + S^{-m}, \text{ если } X_S < 0.$$

Таким образом, для преобразования прямого кода S -ичного отрицательного числа в дополнительный необходимо перевести его в обратный код и к младшему разряду добавить единицу.

Пример. Для целого отрицательного числа $X_2 = -10010_2$ получим $X_{2обр} = 101101_2$, откуда $X_{2доп} = 101110_2$.

При этом мы избавляемся от ± 0 , т.к. " -0 "_{доп} = " $+0$ "_{доп} = 000000₂

При выполнении операции сложения чисел, представленных специальными S -ичными кодами знаковые разряды участвуют в операции наряду с цифровыми разрядами. При этом цифровые разряды складываются как модули чисел по правилам S -ичной арифметики. Знаковые разряды и цифры переноса из старшего цифрового разряда при любом основании системы счисления ($S \geq 2$) складываются как одноразрядные двоичные коды. Если при этом формируется перенос из знакового разряда, то он имеет вес единицы младшего разряда S^m при использовании обратного кода и должен быть добавлен к младшему разряду результата. При использовании дополнительного кода единица переноса из знакового разряда не принимается во внимание, т.е. отбрасывается.

Признаком *положительного переполнения* разрядной сетки является наличие переноса в знаковый разряд суммы без переноса из знакового разряда; признаком *отрицательного переполнения* разрядной сетки является наличие переноса из знакового разряда без переноса в знаковый разряд. При отсутствии переносов или при наличии обоих переносов (в знаковый разряд и из знакового) переполнение разрядной сетки отсутствует.

Пример. Сложить алгебраически, используя дополнительный код:

1) $24_{10} - 10_{10} = ?$

$24D = 011000B$, $10D = 001010B$

$(-10D)_{\text{доп}} = 110110B$

$$\begin{array}{r} 011000B \\ + 110110B \\ \hline \leftarrow 001110B = 14D \end{array}$$

2) $10_{10} - 24_{10} = ?$

$(-24D)_{\text{доп}} = 101000B$

$$\begin{array}{r} 001010B \\ + 101000B \\ \hline 110010B = (-14D)_{\text{доп}} \end{array}$$

1.5. Контрольные вопросы и задания для самоподготовки

1. Пояснить на примере разницу между позиционной и непозиционной системой исчисления.

2. Привести примеры чисел в десятичной, двоичной, восьмеричной и шестнадцатеричной системах исчисления.
3. Пояснить принцип перевода чисел из одной системы исчисления в другую с учетом их целой и дробной частей.
4. Произвести перевод чисел: $38D=?B$, $56D=?B$, $74D=?B$, $122D=?B$, $178D=?B$, $184D=?B$, $194D=?B$, $214D=?B$.
5. Произвести перевод чисел: $0101B=?D$, $1011B=?D$, $11001B=?D$, $10111B=?D$, $11011B=?D$, $01011B=?D$, $01111B=?D$, $11110B=?D$.
6. Произвести перевод чисел: $5D=?Q$, $8D=?Q$, $11D=?Q$, $15D=?Q$, $23D=?Q$, $29D=?Q$, $37D=?Q$, $44D=?Q$.
7. Произвести перевод чисел: $9D=?H$, $13D=?H$, $16D=?H$, $19D=?H$, $20D=?H$, $25D=?H$, $28D=?H$, $35D=?H$.
8. Преобразовать числа: $0001 \rightarrow ?Q$, $0011 \rightarrow ?Q$, $1001 \rightarrow ?Q$, $1011 \rightarrow ?Q$, $1110 \rightarrow ?Q$, $1111 \rightarrow ?Q$, $10000 \rightarrow ?Q$, $10101 \rightarrow ?Q$.
9. Преобразовать числа: $00111 \rightarrow ?H$, $01000 \rightarrow ?H$, $01001 \rightarrow ?H$, $01111 \rightarrow ?H$, $10001 \rightarrow ?H$, $11111 \rightarrow ?H$, $100001 \rightarrow ?H$, $111000 \rightarrow ?H$.
10. Пояснить на примерах формы записи целых и дробных чисел с фиксированной и плавающей запятой.
11. Пояснить представление чисел с плавающей запятой.
12. Привести примеры записи положительных и отрицательных чисел в прямом коде.
13. Привести примеры записи целых и дробных чисел в дополнительном коде.
14. Объяснить принцип сложения чисел с использованием дополнительных кодов.
15. Объяснить сущность не упакованного и упакованного форматов двоично-десятичных чисел.

Глава 2. Логические основы построения цифровых устройств

2.1. Алгебра логики

2.1.1. Логические переменные и функции

Аппарат алгебры логики или *булевой* алгебры (Дж. Буль, 1815 – 1864 г.) используется для формального описания цифровых устройств.

Основным понятием алгебры логики является *высказывание* – некое предложение, о котором можно сказать только истинно оно ($x = 1$, true) или ложно ($x = 0$, false). В этой связи *логическими* называются *переменные*, если они принимают только два значения: 0 и 1.

Логической функцией (ЛФ) выражается более *сложное* высказывание, зависящее от нескольких переменных.

Логической называется функция $y = f(x_1, x_2, \dots, x_n)$, принимающая значение 0 или 1 на наборах логических переменных x_1, x_2, \dots, x_n (*набор* – *совокупность аргументов x_i*).

Для n переменных (аргументов) число наборов конечно и равно 2^n (от 00...0 до 11...1). Поскольку на любом наборе ЛФ может принимать значение 0 или 1, то число возможных ЛФ от n аргументов будет 2^N , где $N=2^n$ - число наборов, таким образом, число возможных ЛФ для n аргументов равно 2^{2^n} .

Пример. Для $n=1$, $2^{2^1}=4$, таким образом имеется четыре ЛФ f_{0-3} ($f_0=0$, $f_1=1$, $f_2=x$, $f_3=\bar{x}$). Пусть $n = 2$ (x_2, x_1), $N = 2^2 = 4$; $2^N = 16$, т.е. для двух аргументов можно построить 16 ЛФ ($f_0, f_1, f_2, \dots, f_{15}$), как это следует из табл. 2.

Логическая функция, полностью определенная на всех наборах аргументов, *существенно зависит от x_i* , если выполняется неравенство:

$$f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots) \neq f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots).$$

В противном случае аргумент x_i – *фиктивный*, а функция, не зависящая от всех аргументов, называется *вырожденной*. Иначе говоря, если при изменении аргумента x_i значение функции не меняется, то функция от этого аргумента не зависит и этот аргумент можно опустить. Указанное свойство логических функций будет использовано при минимизации их аналитического представления. Физическая реализация ЛФ – цифровое устройство (комбинационная схема).

Таблица 2

$f(x_2, x_1)$	x_2	0	0	1	1	Наименование логической функции	Обозначение
	x_1	0	1	0	1		
f_0		0	0	0	0	Константа 0	0
f_1		0	0	0	1	Конъюнкция (функция "И")	$x_2 \wedge x_1$
f_2		0	0	1	0	Запрет по x_1	$x_2 \Delta x_1$
f_3		0	0	1	1	Повторение x_2	x_2
f_4		0	1	0	0	Запрет по x_2	$x_1 \Delta x_2$
f_5		0	1	0	1	Повторение x_1	x_1
f_6		0	1	1	0	Сумма по модулю 2	$x_2 \oplus x_1$
f_7		0	1	1	1	Дизъюнкция (функция "ИЛИ")	$x_2 \vee x_1$
f_8		1	0	0	0	Стрелка Пирса ("ИЛИ-НЕ")	$x_2 \downarrow x_1$
f_9		1	0	0	1	Логическая равнозначность	$x_2 \sim x_1$
f_{10}		1	0	1	0	Инверсия x_1	\bar{x}_1
f_{11}		1	0	1	1	Импликация от x_1 к x_2	$x_2 \rightarrow x_1$
f_{12}		1	1	0	0	Инверсия x_2	\bar{x}_2
f_{13}		1	1	0	1	Импликация от x_2 к x_1	$x_1 \rightarrow x_2$
f_{14}		1	1	1	0	Штрих Шеффера ("И-НЕ")	x_2 / x_1
f_{15}		1	1	1	1	Константа 1	1

Второе название логической функции ИЛИ (дизъюнкция) – логическое сложение, ЛФ И (конъюнкция) – логическое умножение, а ЛФ f_6 (сумма по mod2) – исключающее ИЛИ. В инженерной практике нашли наибольшее применение *логические элементы* (ЛЭ), реализующие следующие ЛФ: f_1 (И, AND), f_7 (ИЛИ, OR), f_6 (Исключающее ИЛИ, XOR), функции Пирса f_8 (И-НЕ, NAND) и Шеффера f_{14} (ИЛИ-НЕ, NOR), а также вырожденные функции – инверсия f_{10} , f_{12} (НЕ, NOT) и повторение f_3 , f_5 аргументов. Остальные ЛФ (в том числе и любая ЛФ из указанных) могут быть получены посредством суперпозиции из нескольких (или даже одной) *базисных* ЛФ, как будет показано далее.

На рис. 4 показаны условные графические обозначения (УГО) основных логических элементов: повторителя, инвертора (НЕ), И, ИЛИ, И-НЕ, ИЛИ-НЕ, сумма по модулю 2 (Искл. ИЛИ).

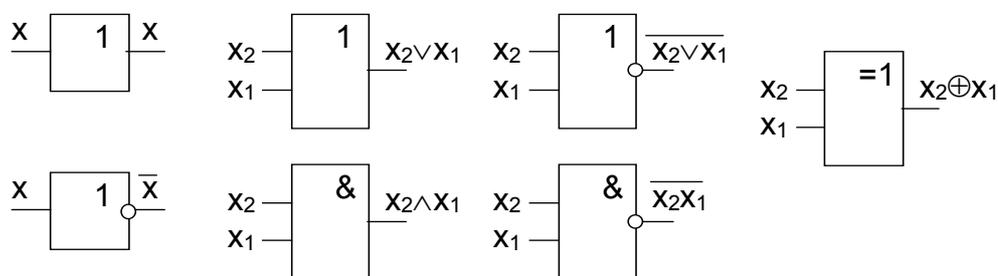


Рис. 4

2.1.2. Основные свойства логических функций

Функции И, ИЛИ, НЕ удовлетворяют следующим законам, которые легко доказываются путем подстановки наборов аргументов. Некоторые из этих законов тождественны операциям умножения и сложения, а некоторые – специфичны для алгебры логики и не имеют прямых аналогов в обычной алгебре.

1. **Идемпотентный закон:**

$$x_1 \vee x_1 = x_1 ,$$

$$x_1 x_1 = x_1 .$$

2. **Коммутативный (переместительный) закон:**

$$x_1 \vee x_2 = x_2 \vee x_1 ,$$

$$x_1 x_2 = x_2 x_1 .$$

3. **Ассоциативный (сочетательный) закон:**

$$(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3) ,$$

$$x_1 (x_2 x_3) = (x_1 x_2) x_3 .$$

4. **Дистрибутивный (распределительный) закон** (нет аналога в алгебре):

$$x_1 (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3 ,$$

$$x_1 \vee (x_2 x_3) = (x_1 \vee x_2) (x_1 \vee x_3) .$$

5. **Закон отрицания:**

$$x_1 \vee \bar{x}_1 = 1 ,$$

$$x_1 \bar{x}_1 = 0 .$$

6. **Закон двойного отрицания:**

$$\bar{\bar{x}} = x .$$

7. **Правило (закон) де Моргана:**

$$\overline{x_1 x_2 \dots x_n} = \bar{x}_1 \vee \bar{x}_2 \vee \dots \vee \bar{x}_n ,$$

$$\overline{x_1 \vee x_2 \vee \dots \vee x_n} = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n .$$

8. **Закон поглощения:**

$$x_1 \vee x_1 x_2 = x_1 ,$$

$$x_1 (x_1 \vee x_2) = x_1 .$$

9. **Закон склеивания** (закон с очень важными последствиями – см. п. 2.2.4):

$$x_1 x_2 \vee x_1 \bar{x}_2 = x_1 ,$$

$$(x_1 \vee x_2) (x_1 \vee \bar{x}_2) = x_1 .$$

2.2. Задание логических функций

2.2.1. Задание логических функций таблицей истинности

Из существующего множества способов задания ЛФ наиболее употребим способ задания с помощью **таблицы истинности**, где каждому набору аргументов соответствует значение ЛФ. В приведенной табл. 2 ЛФ $y_1(x_2, x_1, x_0)$ полностью определена на всех наборах аргументов, а ЛФ $y_2(x_2, x_1, x_0)$ не полностью определена, т.е. для некоторых наборов значение функции безразлично (отмечено звездочкой). Доопределение ЛФ на этих наборах нулем или единицей может быть впоследствии использовано при минимизации (см. п. 2.2.4).

Т а б л и ц а 3

x_2	x_1	x_0	y_1	y_2
0	0	0	0	0
0	0	1	1	*
0	1	0	1	*
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	*

2.2.2. Аналитическое представление логических функций с помощью совершенных нормальных форм

ЛФ может быть представлена аналитически в виде **СКНФ** и **СДНФ** — **совершенных конъюнктивных и дизъюнктивных нормальных форм**.

Введем определения:

- ♦ **Конъюнкция** называется **элементарной** (ЭК), если она содержит любое количество попарно различных между собой переменных со знаками отрицания или без них, например, ЭК: x_1x_2 , $x_1\bar{x}_3x_4$.
- ♦ **Дизъюнкция** называется **элементарной**, если она содержит любое количество попарно различных между собой переменных со знаками отрицания или без них, например, ЭД: $x_1 \vee \bar{x}_2$, $x_1 \vee \bar{x}_3 \vee x_4$.

Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция элементарных конъюнкций.

Конъюнктивной нормальной формой (КНФ) называется конъюнкция элементарных дизъюнкций.

Пример. $y = x_1x_2 \vee \bar{x}_2x_3 \vee \bar{x}_1x_2x_3$ (ДНФ);
 $y = (x_1 \vee x_2 \vee \bar{x}_3)(x_1 \vee \bar{x}_2)$ (КНФ).

Преобразование ДНФ \Leftrightarrow КНФ совершается по правилам, изложенным выше (2.1.2).

Пример. Преобразовать ДНФ в КНФ.

Используя дистрибутивный закон и закон отрицания, получим:

$$x_1x_2 \vee \bar{x}_1x_3 = (x_1x_2 \vee \bar{x}_1)(x_1x_2 \vee x_3) =$$

$$= (\bar{x}_1 \vee x_1)(\bar{x}_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3) = (\bar{x}_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3).$$

Для определения *совершенных нормальных форм* (СДНФ и СКНФ) введем понятия *конституент единицы и нуля* (КЕ и КН).

Конституентой единицы (K_i) называется ЭК n аргументов, принимающая значение 1 лишь на одном наборе.

Конституентой нуля (M_i) называется ЭД n аргументов, равная 0 лишь на одном наборе.

Общее число K_i и M_i , определенных на всех наборах равно 2^n (для ЛФ n аргументов).

Совершенной ДНФ называется дизъюнкция всех КЕ этой функции $\bigcup_i^m K_i$.

Совершенной КНФ называется конъюнкция всех КН этой функции $\bigcap_j^k M_j$.

Согласно этим определениям из таблицы истинности (табл. 3) для функции $y_1(x_2, x_1, x_0)$, имеем:

$$\text{СДНФ } y_1 = K_1 \vee K_2 \vee K_4 \vee K_5 \vee K_7$$

$$\text{СКНФ } y_1 = M_0 M_3 M_6.$$

Сформулируем *правило записи K_i и M_i* .

В наборе значений аргументов составляющих элементарную конъюнкцию конституенты единицы каждый аргумент x_i записывается в прямой форме, если его значение на этом наборе 1, и в инверсном виде, если его значение на данном наборе равно 0. Для записи КН *правило обратное*: аргумент записывается в КН в прямой форме, если на данном наборе его значение равно нулю и в инверсной форме, если его значение на наборе равно единице.

$$\text{Тогда СДНФ } y_1 = \bar{x}_2 \bar{x}_1 x_0 \vee \bar{x}_2 x_1 \bar{x}_0 \vee x_2 \bar{x}_1 \bar{x}_0 \vee x_2 \bar{x}_1 x_0 \vee x_2 x_1 x_0,$$

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

Переход от СДНФ к СКНФ (и обратно) может быть совершен по формальным правилам:

- ♦ Записать новую СДНФ (СКНФ) из КЕ (КН), не входящих в исходную, например, из табл. 2 следует $y^*_1 = \bar{x}_2 \bar{x}_1 \bar{x}_0 \vee \bar{x}_2 x_1 x_0 \vee x_2 x_1 \bar{x}_0$.
- ♦ В новой СДНФ (СКНФ) операцию ИЛИ заменить на И, а операцию И – на ИЛИ, т.е. $y^{**}_1 = (\bar{x}_2 \vee \bar{x}_1 \vee \bar{x}_0) (\bar{x}_2 \vee x_1 \vee x_0) (x_2 \vee x_1 \vee \bar{x}_0)$.
- ♦ Инвертировать аргументы для завершения преобразования:
 $y^{***}_1 = (x_2 \vee x_1 \vee x_0) (x_2 \vee \bar{x}_1 \vee \bar{x}_0) (\bar{x}_2 \vee \bar{x}_1 \vee x_0) = y_1$.

В общем случае предпочтение отдается той совершенной форме (СДНФ, СКНФ), в которой меньше составляющих – произведений (сумм).

2.2.3. Функционально полные системы логических функций

Система ЛФ $\{y_1, \dots, y_N\}$ называется **функционально полной** (или **базисом**), если любая ЛФ из системы может быть определена суперпозицией функций $\{y_1, \dots, y_N\}$.

Минимальным назовем такой базис, для которого удаление хотя бы одной функции y_i превращает указанную систему ЛФ в неполную.

Функционально полной является система функций И, ИЛИ, НЕ, но эта система не минимальна. Функция штрих Шеффера (И-НЕ) или стрелка Пирса (ИЛИ-НЕ) образует минимальный базис. *Возникает вопрос* – что для логической функции считать минимальным представлением.

Примеры функционально-полных наборов.

1. Набор И, ИЛИ, НЕ:

$$\text{СДНФ } y(x_1 x_2 \dots x_n) = \bigcup_i^m K_i, \quad \text{СКНФ } y(x_1 x_2 \dots x_n) = \bigcap_j^k M_j.$$

2. Наборы И-НЕ, ИЛИ-НЕ:

$$\text{И-НЕ } y(x_1 x_2 \dots x_n) = \overline{\bigcup_i^m K_i} = \overline{\bigcap_i^m \overline{K_i}}, \quad \text{ИЛИ-НЕ } y(x_1 x_2 \dots x_n) = \overline{\bigcap_j^k M_j} = \overline{\bigcup_j^k \overline{M_j}}.$$

2.2.4. Минимизация логических функций

ДНФ (КНФ), содержащая наименьшее число логических переменных по сравнению со всеми другими представлениями данной функции, называется минимальной ДНФ (КНФ) – МДНФ (МКНФ).

Основной задачей проектирования логических схем (ЛС) и структур является реализация ЛФ в заданном базисе при наименьшем числе логических элементов (ЛЭ) (корпусов ИМС и т.д.).

Известны аналитические методы: Квайна, Мак-Класки, Рота и другие. В инженерной практике более распространен формальный **метод минимизации с помощью диаграмм Вейча (карт Карно)**. Диаграмма Вейча (ДВ) представляет собой квадратную или прямоугольную таблицу с числом клеток 2^n (n - число переменных) – по числу всех конституент функции.

Если минимизация идет в ДНФ, то диаграмма Вейча заполняется единицами (конституентами 1), если в КНФ – то ДВ заполняется нулями (конституентами 0), таким образом ДВ представляет собой иную форму записи таблицы истинности.

Диаграммы Вейча для $n = 2, 3, 4$, показаны на рис. 5

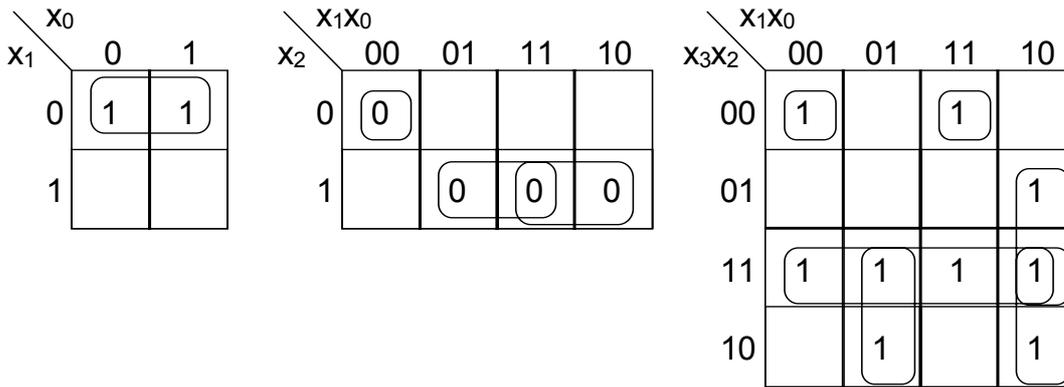


Рис. 5

Диаграммы построены так, что соседние клетки соответствуют наборам аргументов, различающимся значением одного аргумента. Клетки на противоположных краях диаграммы также являются соседними.

Нахождение МДНФ или МКНФ сводятся к отысканию варианта, при котором все единицы в СДНФ (нули в СКНФ) покрываются наименьшим числом наиболее коротких произведений или сумм, для чего в ДВ обводят прямоугольными контурами клетки с единицами (нулями) по правилам:

- ◆ контур должен быть прямоугольным с числом клеток 2^i , $i = 0, 1, 2, \dots$;
- ◆ одни и те же клетки с 1 или 0 могут входить в несколько контуров;
- ◆ число контуров должно быть как можно меньше, а размеры каждого контура – как можно больше;
- ◆ объединение начинают с клеток, которые могут войти в контур единственным образом;
- ◆ клетки с неопределенными значениями могут произвольным образом доопределяться и входить или не входить в контур.

Для каждого контура в записи соответствующей ДНФ (КНФ) исключаются те аргументы, от которых функция не зависит. Иначе говоря, если для данного контура изменения значений x_i не приводят к изменению значения функции, то из записи этот аргумент опускается.

Минимизируем ЛФ y_1 из табл. 3 в ДНФ (рис. 6).

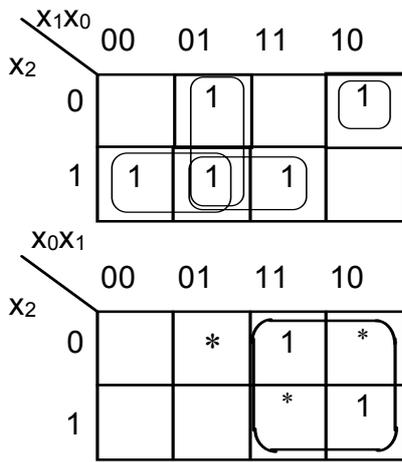


Рис. 6

Для y_1 диаграмма Вейча содержит 5 КЕ, которые входят в 4 контура:

$$\text{МНДФ } y_1 = x_2 \bar{x}_1 \vee x_2 x_0 \vee \bar{x}_1 x_0 \vee \bar{x}_2 x_1 \bar{x}_0.$$

Для y_2 диаграмма Вейча содержит 2 КЕ и три неопределенных значения; неопределенные значения (*), которые вошли в единственный контур, считаются допределенными единицами.

$$\text{МДНФ: } y_2 = x_1$$

2.2.5. Минимизация систем логических функций

Рассмотрим цифровую схему S , имеющую N логических входов и L выходов, представляющих функции от входных аргументов (рис. 7).

Если каждая из L функций может быть представлена как ЛФ входных переменных x_1, x_2, \dots, x_N , то такая схема называется **комбинационной** (КС) и описывается системой уравнений (2.1).

Комбинационную схему часто называют логическим (N, L) – полюсником, а систему (2.1) – системой собственных функций (N, L) – полюсника. Эквивалентными считаются КС, у которых собственные функции y_i ($i=1..L$) равны.

$$\begin{cases} y_1 = y_1(x_1 \dots x_N) \\ y_2 = y_2(x_1 \dots x_N) \\ \dots \dots \dots \dots \dots \dots \\ y_L = y_L(x_1 \dots x_N) \end{cases} \quad (2.1)$$

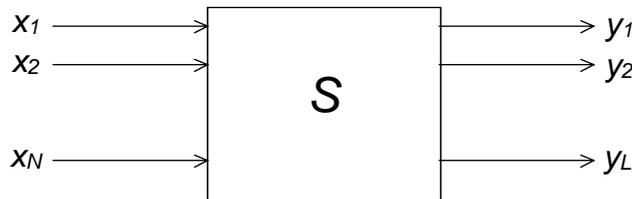


Рис. 7

Синтез КС состоит из двух этапов.

На первом этапе синтеза составляется система собственных функций, адекватно отражающая некоторое физическое описание работы будущего устройства.

На втором этапе система собственных функций реализуется некоторой комбинационной схемой, для построения которой используют набор логических элементов.

Для одной и той же системы собственных функций может быть построено несколько КС различной сложности. Минимизация ЛФ, составляющих КС может быть отдельная, но может быть и совместная. Идея совместной мини-

мизации состоит в покрытии диаграмм Вейча одинаковыми контурами, что дает одни и те же схемные блоки для различных ЛФ.

2.3. Комбинационные цифровые устройства

2.3.1. Общие положения

Комбинационное цифровое устройство (КЦУ) — это преобразователь дискретной информации X в Y , причем входному двоичному слову X в данный момент времени однозначно и независимо от предыдущего состояния соответствует выходное слово Y в тот же момент времени. У КЦУ нет «памяти» о предистории процесса. ЦУ с «памятью» называются автоматами и будут рассмотрены далее.

Элементарными КЦУ являются логические элементы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ и т. п. Более сложными являются КЦУ, состоящие из элементарных и удовлетворяющие двум условиям:

- входы последующих ЛЭ связаны с выходами предыдущих (рис. 8,а);
- отсутствуют обратные связи с выходов последующих ЛЭ на входы предыдущих (замкнутые пути для сигналов) (рис. 8,б).

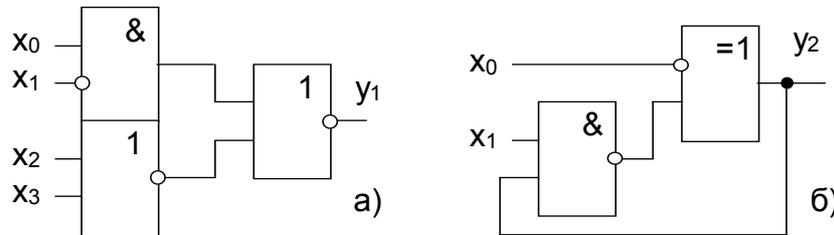


Рис. 8

Для КЦУ характерен **принцип дуальности** (следует из правила *де Моргана*). Если в КЦУ₁, реализующей ЛФ $f(x_1x_2\dots x_n)$, заменить И на ИЛИ, ИЛИ на И, проинвертировать входы x_i , то полученная КЦУ₂ реализует $\overline{f(x_1x_2\dots x_n)}$.

2.3.2. Базовые интегральные схемы и их основные параметры

Логические **интегральные схемы (ИС)**, выпускаемые промышленностью, представлены широкой номенклатурой интегральных серий на основе биполярных и полевых транзисторов.

Серии ИМС отличаются по целому ряду параметров в зависимости от технологий их изготовления (ТТЛ, ТТЛШ, ЭСЛ, КМОП, n -МОП, p -МОП и другие).

К числу типовых параметров ИМС относятся:

1) **Коэффициент разветвления по выходу** $K_{раз}(N)$ — число входов однотипных ИС, которые одновременно могут быть подключены к выходу данной ИС (нагрузочная способность ИМС)

2) **Коэффициент объединения по входу** $K_{об}(N)$ — число входных сигналов данной ИС;

3) **Статические характеристики:**

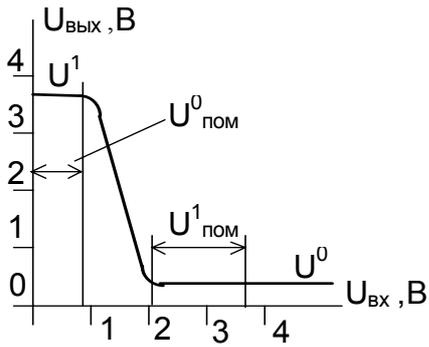
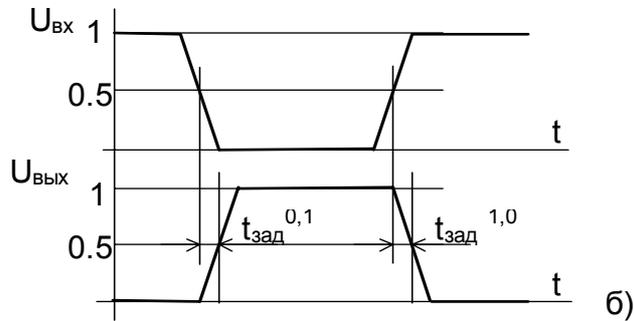
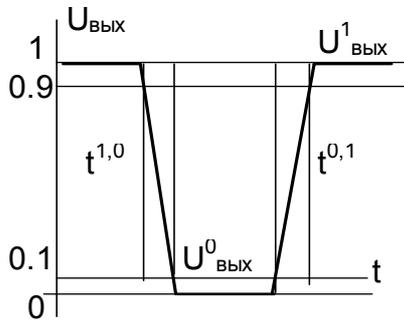


Рис. 9

- входная $I_{вх} = f_1(U_{ВХ})$
- выходная $U_{вых} = f_2(I_{ВЫХ})$
- передаточная $U_{вых} = f_3(U_{ВХ})$.

Типовая передаточная характеристика ТТЛ ИМС приведена на рис. 9, где показаны уровни помех на входе $U^1_{пом}$ и $U^0_{пом}$, вызывающие изменение сигнала на выходе (“срабатывание” элемента). Эти значения характеризуют помехоустойчивость схемы. Уровень логического нуля для ТТЛ ИМС $U^0 = 0...0.4$ В, уровень логической единицы $U^1 = 2.4 - 4.5$ В. Порог срабатывания по нулевому уровню составляет ≈ 0.8 В.

4) **Временные параметры** (динамические) можно оценить из рис. 10.



а)

б)

Рис. 10

Здесь $t^{0,1}$ и $t^{1,0}$ – время перехода из 1 в 0 и наоборот (рис. 10, а) и $t^{0,1}_{зад}$ и $t^{1,0}_{зад}$ – время задержки включения и выключения ЛЭ (рис. 10, б)

5) Вид реализуемой логической функции (И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ и т.д.). В качестве примера на рис. 11, а показано УГО ЛЭ 2-2-3И-ИЛИ-НЕ.

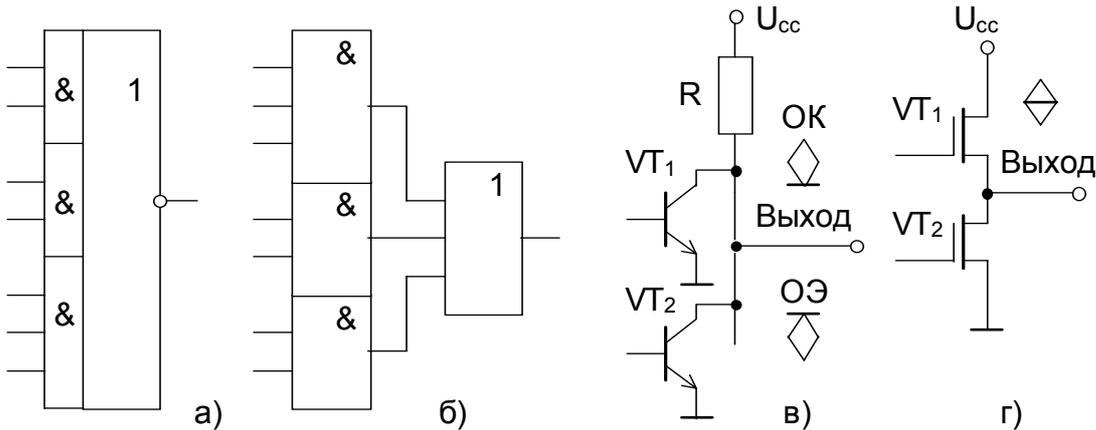


Рис. 11

б) **Выходы ИС.**

Обычные выходы ИМС ТТЛ(Ш) нельзя объединять (соединять). Это делается только логически (с помощью схем И-ИЛИ), как показано на рис. 11, б или используя монтажное ИЛИ для ИС с открытым коллектором или эмиттером (рис. 11, в).

ИС с тристабильным выходом позволяют отключить выход переводом его в высокоимпедансное состояние (ВИС, Z-состояние) (рис. 11, г).

- 1) Логический ноль на выходе U^0 , $VT1$ закрыт, $VT2$ - открыт;
- 2) Логическая единица на выходе U^1 , $VT1$ открыт, $VT2$ - закрыт;
- 3) ВИС состояние, $VT1$ и $VT2$ закрыты.

Выходы нескольких ИМС с тристабильным выходом могут быть объединены, при этом только одна из этих ИМС включена (“выбрана”) сигналом

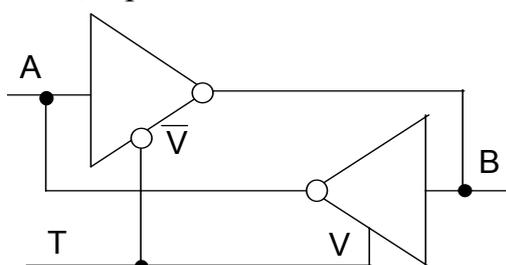


Рис. 12

разрешения CS (*chip select*), а выходы остальных переведены в ВИС. На рис. 12 показан управляемый двунаправленный инвертирующий буфер, позволяющий передавать информацию в том или ином направлении в зависимости от управляющего входа Т. Если $T=0$, то разрешена передача от А к В, если $T=1$, то разрешена передача от В к А. Такая

коммутация возможна, поскольку всегда выход одного из инверторов находится в высокоимпедансном состоянии.

Комбинация из нескольких подобных устройств позволяет направлять потоки информации в параллельном коде, образуя так называемую **шину передачи данных**.

Параметры интегральных микросхем серий ТТЛ, ТТЛШ, ЭСЛ, КМОП можно найти в соответствующих справочниках [4].

2.3.3. Типовые КЦУ

2.3.3.1. Дешифраторы (декодеры)

Дешифратором (декодером) называется КЦУ, у которого каждой входной комбинации сигналов $\{x_i\}$ соответствует активное состояние только одного выходного сигнала. **Полный дешифратор** (ДШ) с n входами имеет 2^n выходов. Таблица истинности и условное графическое обозначение (УГО) дешифратора К155ИДЗ (4x16, L-активные выходы, выбор кристалла) приведе-

Таблица 4

x_3	x_2	x_1	x_0	y_0	y_1	y_2	...	y_{15}
0	0	0	0	0	1	1	...	1
0	0	0	1	1	0	1	...	1
0	0	1	0	1	1	0	...	1
0	0	1	1	1	1	1	...	1
...
1	1	1	1	1	1	1	...	0

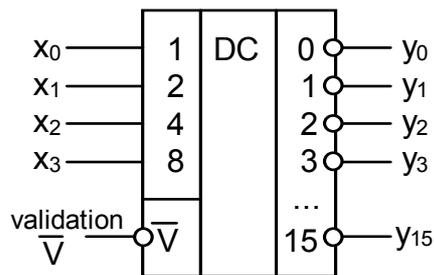


Рис. 13

ны, соответственно в табл. 4 и на рис. 13.

ЛФ, описывающие выходные сигналы ДШ выглядят следующим образом:
 $y_0 = \overline{x_3 \overline{x_2} \overline{x_1} \overline{x_0} V}$, $y_1 = \overline{x_3 \overline{x_2} \overline{x_1} x_0 V}$, ..., $y_{15} = \overline{x_3 x_2 x_1 x_0 V}$.

Неполный дешифратор при n входах имеет число выходов $N < 2^n$. В виде серийных ИС выпускаются только ДШ 4x10. Такие дешифраторы работают в двоично-десятичном коде 8-4-2-1 и используются для индикации результата в десятичном коде (К155ИД1, К561ИД1).

Каскадирование дешифраторов позволяет увеличить число входов и выходов. На рис. 14,а показан ДШ 4x16 на базе двух неполных ДШ 4x10, а на рис. 14,б – ДШ 5x32 на базе ДШ 4x16, в построении которого использованы управляющие входы V . Выходные сигналы $y_0 - y_{15}$ формируются с помощью первого ДШ при условии $x_4 = 0$, а сигналы $y_{16} - y_{31}$ формируются на выходах второго ДШ при $x_4 = 1$.

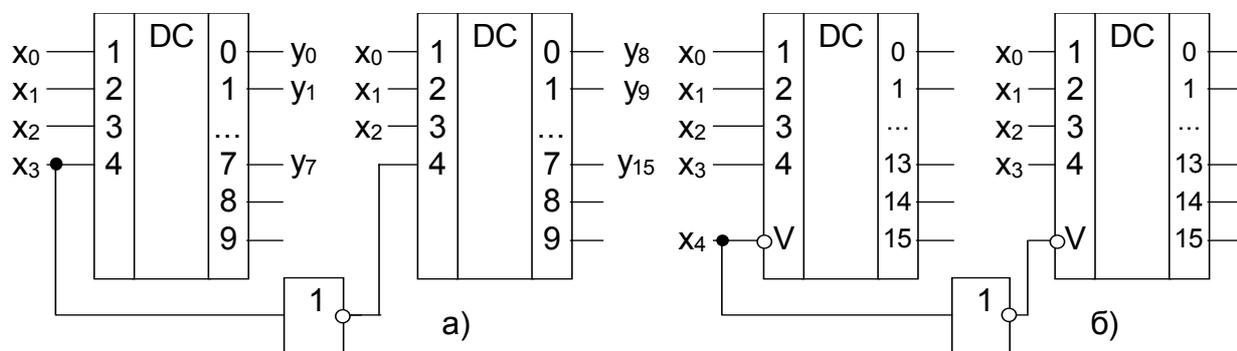


Рис. 14

2.3.3.2. Шифраторы (кодеры)

Шифраторы выполняют обратное по отношению к дешифраторам преобразование единичного (унитарного) кода на входе x_i , $i = 0, 1, \dots, 2^{n-1}$ в n -разрядный позиционный двоичный код. Например, для восьмивходового шифратора активному значению одного из входных сигналов $x_i = 0$ соответствует комбинация выходных сигналов $y_2 - y_0$ (табл. 5).

Таблица 5

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0
0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	0
1	1	0	1	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	0	0
1	1	1	1	0	1	1	1	0	1	1
1	1	1	1	1	0	1	1	0	1	0
1	1	1	1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	0	0	0	0

Для шифратора должно выполняться условие $x_i \vee x_j = 1$ при $i \neq j$ (в случае L-активных входов). Если сигналы на входы x_i поступают от независимых источников, то это условие нереализуемо. В этом случае *каждому входу назначается свой приоритет*. Будем считать, что чем выше номер входа i , тем выше его приоритет. **Приоритетный шифратор** вырабатывает двоичный код числа для активного входа с наибольшим номером [2]. Примером приоритетного ДШ 8x3 может служить К155ИВ1.

2.3.3.3. Мультиплексоры

С помощью **мультиплексоров** осуществляется коммутация на выход одного из нескольких информационных входных сигналов D_i в соответствии с заданным кодом на управляющих входах $\{x_{ij}\}$. УГО мультиплексора “4→1” приведено на рис. 15.

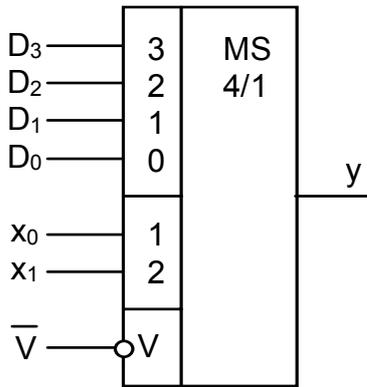


Рис. 15

Аналитическое выражение для выходного сигнала y в СДНФ выглядит следующим образом:

$$y = \bar{V} (\bar{x}_1 \bar{x}_0 D_0 \vee \bar{x}_1 x_0 D_1 \vee x_1 \bar{x}_0 D_2 \vee x_1 x_0 D_3) .$$

Сопоставляя формулы для дешифратора и мультиплексора можно отметить, что для каждого входа D_i комбинации сигналов управления $\{x_{ij}\}$ выглядят аналогично комбинациям входных сигналов в дешифраторе. Следовательно, *составной частью мультиплексора является дешифратор*.

Примерами серийных мультиплексоров являются микросхемы К155КП5 (8→1), К155КП2 (2x4→1).

2.3.3.4. Демультимплексоры

Демультимплексоры представляют собой КЦУ для коммутации информационного входного сигнала D на один из выходов в соответствии с заданным кодом на управляющих входах $\{x_{ij}\}$, решая тем самым задачу, обратную мультиплексированию. УГО демультимплексора “1→4” показано на рис. 16.

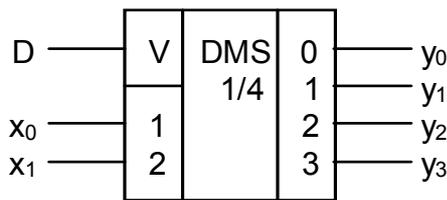


Рис. 16

Логические функции, описывающие выходные сигналы демультимплексора, имеют вид:

$$y_0 = \bar{x}_1 \bar{x}_0 D , \quad y_1 = \bar{x}_1 x_0 D , \\ y_2 = x_1 \bar{x}_0 D , \quad y_3 = x_1 x_0 D .$$

Подобными выражениями описываются выходные сигналы дешифраторов, если принять $D = \bar{V}$. В качестве примера демультимплексора “1→8” рассмотрим двоянный дешифратор К155ИД4 [1-4], как показано на рис. 17.

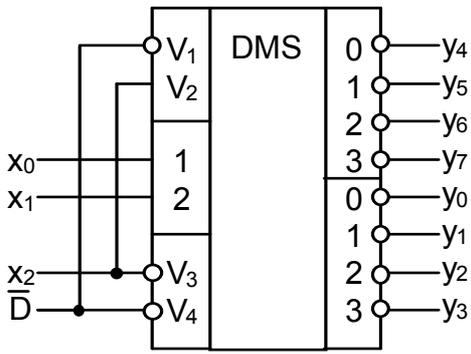


Рис. 17

Информационный сигнал \bar{D} проходит на выходы y_0-y_7 демультиплексора в зависимости от значения набора $x_2x_1x_0$. Если схему на рис. 17 рассматривать как дешифратор, то вход \bar{D} может служить сигналом синхронизации или стробирования выходов. При $\bar{D}=1$ на всех выходах схемы установится логическая единица. Каскадным включением двух ИМС К155ИД4 и ЛЭ “НЕ” можно получить дешифратор на 4 входа и 16 вы-

ходов[2, 4].

2.3.3.5. Сумматоры и АЛУ

Сумматоры представляют собой КЦУ для сложения n разрядных двоичных чисел. Сумматоры (рис. 17) являются основными вычислительными устройствами в арифметическо-логических

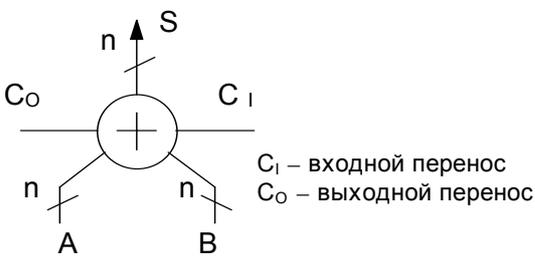


Рис. 18

блоках микропроцессоров и микроЭВМ, от их быстродействия существенно зависит скорость вычислений. Для многоразрядных чисел быстродействие сумматоров определяется в значительной мере скоростью распространения переноса при сложении.

При формировании результата в i -ом разряде необходимо учесть значения чисел в этом разряде a_i и b_i , а также перенос из предыдущего разряда c_i . Каждый разряд сумматора формирует значение суммы в этом разряде s_i и перенос в следующий разряд c_{i+1} (табл. 6).

Таблица 6

a_i	b_i	c_i	s_i	c_{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

ЛФ, описывающие работу одноразрядного двоичного сумматора, могут быть представлены в базисе И-НЕ следующим образом [1, 2]:

$$s_i = \overline{a_i a_i x a_i x x}, \text{ где } x = \overline{b_i b_i c_i b_i c_i c_i};$$

$$c_{i+1} = \overline{a_i c_i a_i b_i b_i c_i}.$$

Схема двоичного 4-разрядного сумматора, составленного из одноразрядных сумматоров SM, представлена на рис. 19.

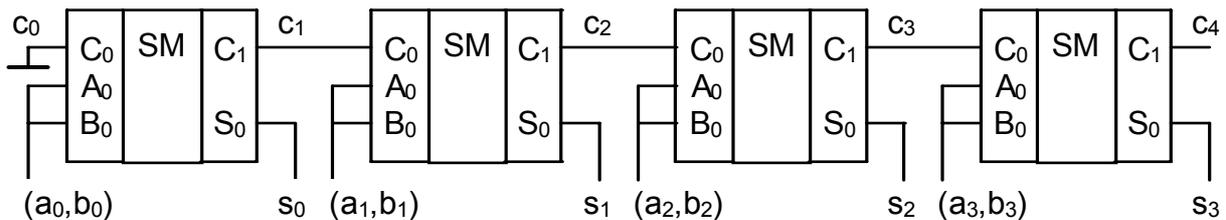


Рис. 19

Данный сумматор позволяет складывать два двоичных числа A_3-A_0 и B_3-B_0 , формируя результат S_3-S_0 и перенос C_4 в следующий разряд. Перенос между разрядами сумматора в схеме на рис. 19 последовательный, что замедляет получение результата при значительном числе разрядов. Увеличение быстродействия сумматоров достигается путем использования параллельного переноса [2] с использованием специальных схем.

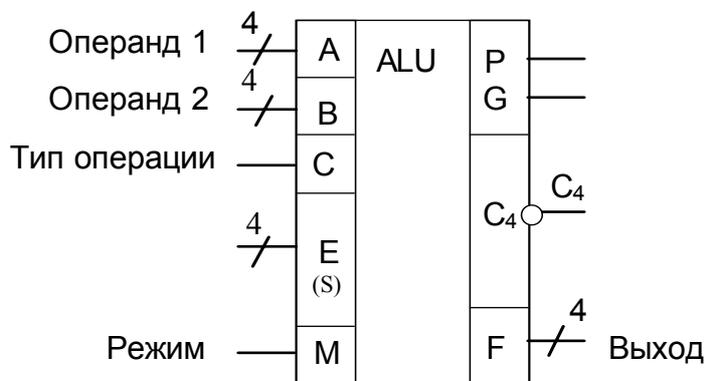


Рис. 20

Комбинационные сумматоры являются основной частью АЛУ – арифметическо-логического устройства — основного операционного узла любого МПУ. В качестве примера можно рассмотреть четырехразрядное АЛУ К155ИПЗ (рис. 20), работающая в двух возможных режимах: арифметические операции

($M = 0$) и логические операции ($M = 1$). Выбор операции задается кодом E ($E_3E_2E_1E_0$). Логические операции выполняются над операндами поразрядно ($A_3-B_3, A_2-B_2, A_1-B_1, A_0-B_0$). При выполнении арифметических операций АЛУ представляет собой четырехразрядный сумматор с параллельным переносом.

На выходах P и G формируются два сигнала $g_{i+1} = a_i b_i$ и $p_{i+1} = a_i \vee b_i$, с помощью которых получается параллельный перенос c_{i+1} , зависящий от всех предыдущих переносов сумматора

$$c_{i+1} = g_{i+1} \vee p_{i+1}g_{i+1} \vee p_{i+1}p_i g_{i-1} \vee \dots \vee p_{i+1}p_1 \dots c_0.$$

Для организации параллельного переноса в 4-м-разрядных АЛУ используются ИМС ускоренного переноса (К155ИП4, К531ИП4) [2].

2.3.3.6. Синтез комбинационных схем на мультилексорах и дешифраторах

Любую ЛФ n аргументов можно реализовать на мультилексоре $2^n \rightarrow 1$, подав на его адресные входы сигналы $(x_{n-1}, x_{n-2}, \dots, x_1, x_0)$, а на информационные входы $D_i, i=0 \dots (2^n-1)$ значения 0 и 1, соответствующие конstituентам 0 и 1 из таблицы истинности ЛФ. Это следует из вышеприведенного аналитического описания сигнала на выходе мультиплексора.

Таким образом, для ЛФ четырех аргументов потребуется мультиплексор $16 \rightarrow 1$. Однако, если подавать на информационные входы не только констан-

ты 0 и 1, но и аргументы x_i или ЛФ от некоторых аргументов, то потребуется мультиплексор с меньшим числом адресных входов.

Реализуем ЛФ у четырех аргументов, представленную диаграммой Вейча (рис. 21, а), с помощью мультиплексора $8 \rightarrow 1$.

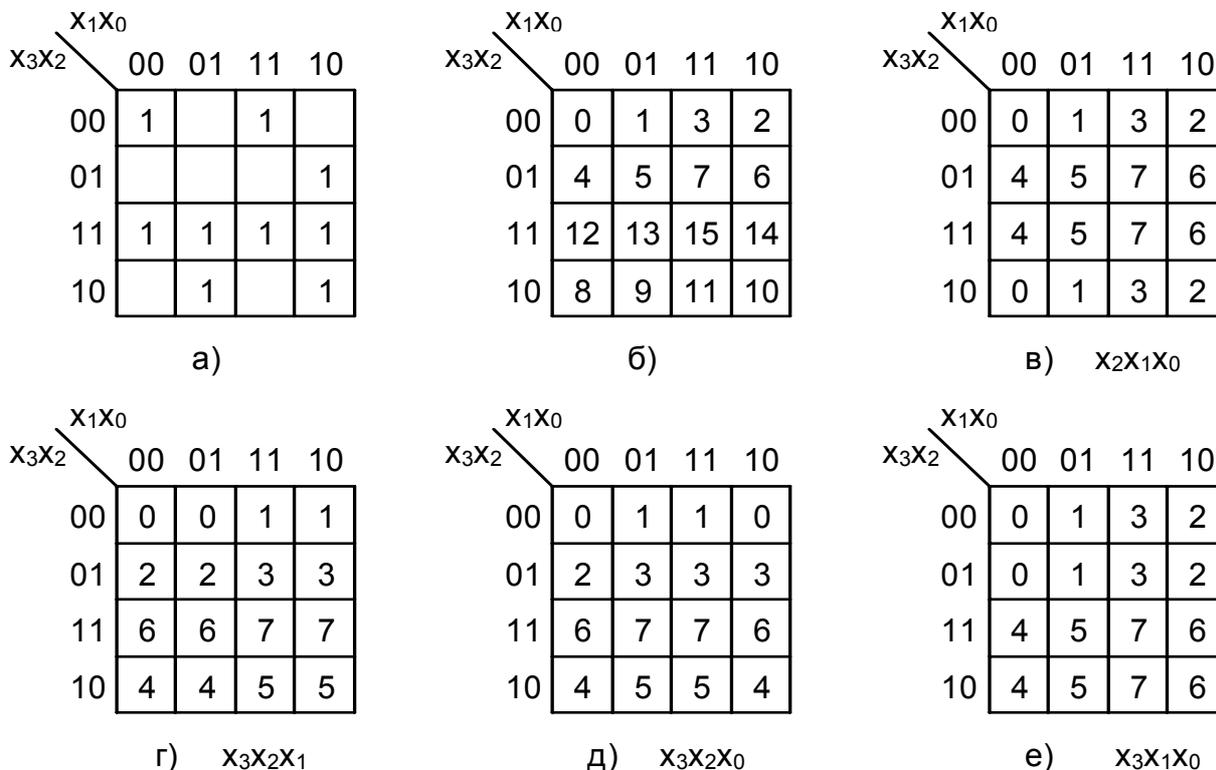


Рис. 21

На три адресных входа такого мультиплексора можно подать три из четырех переменных $x_3x_2x_1x_0$. Этими тремя переменными будет задаваться адрес входного канала мультиплексора D_i ($0 \dots 7$). Клетки диаграмм Вейча четырех переменных нумеруются числами от 0 до 15 (рис. 21, б). Аналогично можно пронумеровать клетки диаграмм Вейча адресами каналов i мультиплексора, не учитывая четвертый аргумент. Таких диаграмм Вейча будет 4 – по числу возможных сочетаний из 4 аргументов по 3 (рис. 21, в- е).

На рис. 21, в-е указаны тройки аргументов, взятых в качестве адресных. Числа i (номера каналов) разбили указанные диаграммы Вейча на 8 частей, каждая из которых представляет диаграмму Вейча из двух клеток для одного аргумента, не являющегося адресным, соответственно x_3, x_0, x_1, x_2 . Эти восемь диаграмм Вейча одной переменной определяют функции $D_i = \varphi_i(x_j)$, которые надо подать на информационные входы мультиплексора. Из рис. 21, а, в следует, что $D_0 = \bar{x}_3, D_1 = x_3, D_2 = x_3,$
 $D_3 = \bar{x}_3, D_4 = x_3, D_5 = x_3, D_6 = 1, D_7 = x_3.$

Таблица 7

Вход	Адресные входы мультиплексора			
	$x_2x_1x_0$	$x_3x_2x_1$	$x_3x_2x_0$	$x_3x_1x_0$
D_0	\bar{x}_3	\bar{x}_0	\bar{x}_1	\bar{x}_2
D_1	x_3	x_0	x_1	0
D_2	x_3	0	x_1	x_2
D_3	\bar{x}_3	\bar{x}_0	0	\bar{x}_2
D_4	x_3	x_0	x_1	x_2
D_5	x_3	\bar{x}_0	\bar{x}_1	1
D_6	1	1	1	x_2
D_7	x_3	1	1	1

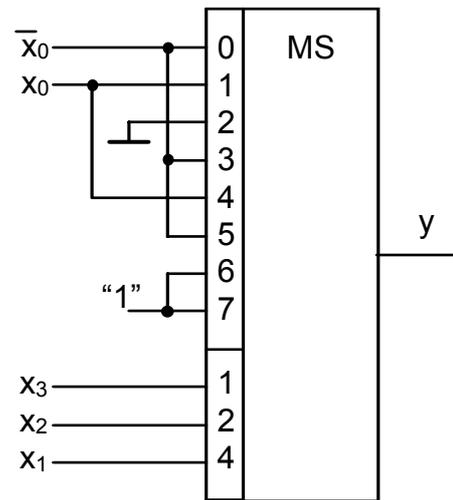


Рис. 22

В табл. 7 приведены значения D_i для всех возможных комбинаций адресных сигналов, полученных из

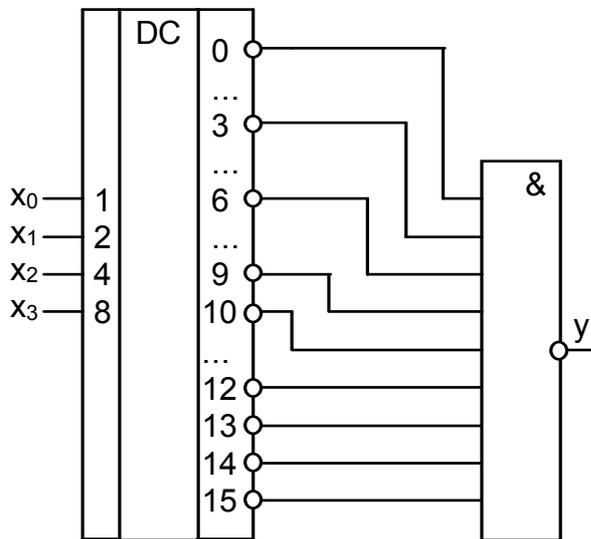


Рис. 23

рис. 21, а, в-е. Предпочтение следует отдать той схеме, на информационные входы которой подается большее число констант 0 и 1. В этом смысле варианты схемы при задании адресов аргументами $x_3x_2x_1$, $x_3x_2x_0$ и $x_3x_1x_0$ эквивалентны (рис. 22).

Для построения КЦУ можно использовать и дешифраторы. Так как активное значение сигнала на каждом выходе дешифратора соответствует одному из наборов входных сигналов, то, объединяя с помощью логических элементов выходные сигналы дешиф-

ратора, соответствующие конstituентам 0 или 1 в задании ЛФ, можно реализовать комбинационную схему с числом аргументов, не превышающем число входов дешифратора.

Для вышеприведенной функции четырех аргументов СДНФ, преобразованная по закону де Моргана в базис И-НЕ, выглядит следующим образом: $y = \overline{K_0 K_3 K_6 K_9 K_{10} K_{12} K_{13} K_{14} K_{15}}$, где K_i - конstituенты 1 функции y .

Комбинационная схема y показана на рис. 23. При построении использован дешифратор 4x16 с L-активными выходными сигналами и логический элемент И-НЕ.

2.3.3.7. Быстродействие КЦУ и эффект состязаний в них («гонки»)

Логические элементы, входящие в состав КЦУ, переключаются с задержками $t_{зад}^{0,1}$, поэтому выходные сигналы КЦУ устанавливаются (изменяются) по окончании переходных процессов в последовательно соединенных элементах. Эту задержку в появлении выходных сигналов надо учитывать в работе устройства, анализируя цепочку ЛЭ, которая дает максимальную задержку. Проанализируем схему на рис. 24.

В данном случае, как следует из временных диаграмм на рис. 24,б, неодинаковые задержки в логических цепях дают помеху на выходе, эта помеха вызвана состязаниями (“гонками”) в цепях распространения сигнала.

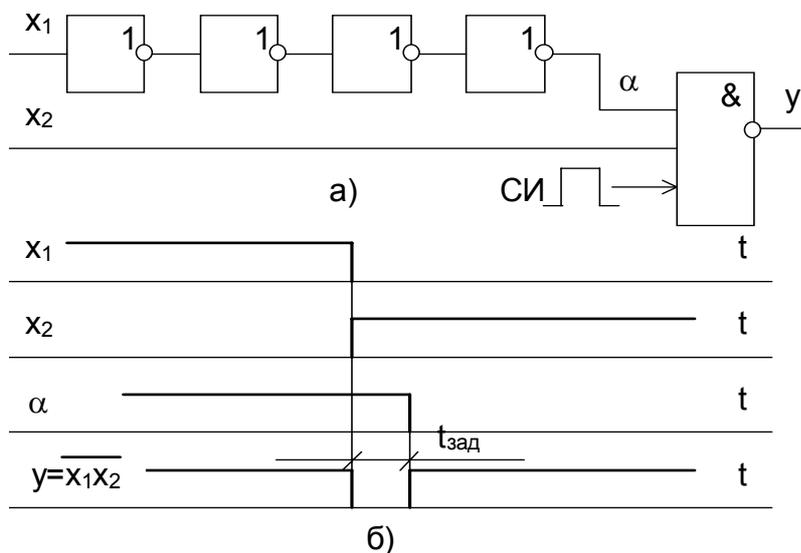


Рис. 24

С “гонками” можно бороться следующими способами:

- 1) задавая такой порядок смены входных сигналов, что помехи вообще не появляются;
- 2) уравнивая задержки в «опасных» участках;
- 3) вводя синхронизацию выходных сигналов, как показано на рис. 24; это наиболее эффективный

способ. В частности, в мультиплексорах, дешифраторах можно синхронизировать выход, блокируя помехи, вызванные “гонками”.

2.4. Контрольные вопросы и задания для самоподготовки

1. Привести примеры ЛФ для $n=1,2,3,4,5,6,7,8$.
2. Записать ЛФ ($n=2$) с номерами $i=1,14$.
3. Определить значения $K_i, i = \overline{0,15}$ для $n=2,3,4,5$.
4. Определить значения $M_i, i = \overline{0,15}$ для $n=2,3,4,5$.
5. Преобразовать ДНФ в КНФ: $x \vee \bar{x}yz, x \vee yz, xy \vee xz, x\bar{y}\bar{z} \vee \bar{x}y\bar{z}$.
6. Преобразовать КНФ в ДНФ: $(x \vee y)(x \vee \bar{y} \vee \bar{z}), (x \vee z)(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)$.
7. Преобразовать ЛФ в КНФ(ДНФ): $\bar{x}\bar{y}z \vee x\bar{z}, \overline{(x \vee y)(\bar{y} \vee z) \vee \bar{x}y}, \overline{x(y \vee \bar{z}) \vee \bar{x}yz}, \overline{\bar{x}\bar{y} \vee xz}, \overline{(x \vee y)(\bar{y} \vee \bar{z}) \vee \bar{x}y}, \overline{x(y \vee \bar{z}) \vee \bar{x}yz}$.

8. Преобразовать ЛФ в СДНФ, построить диаграмму Вейча: $x \vee y\bar{z}$, $x \vee xy \vee z$, $xy \vee \bar{x}z \vee xyz$, $x \vee y \vee z$, $x \vee \bar{x}y \vee \bar{x} \bar{y} \bar{z}$.
9. Преобразовать КНФ в СКНФ, построить диаграмму Вейча: $(x \vee y)(x \vee y \vee \bar{z})$, $(x \vee \bar{y})(y \vee z)(x \vee \bar{z})$, $(x \vee y)z$, $x(\bar{x} \vee y \vee \bar{z})$.
10. Преобразовать СДНФ в СКНФ (аналитически и с помощью диаграмм Вейча): $xyz \vee xy\bar{z}$, $\bar{x}\bar{y}\bar{z} \vee \bar{x}yz \vee \bar{x} \bar{y}z$, $xyz \vee \bar{x} \bar{y}z$, $xyz \vee \bar{x}yz$.
11. Преобразовать СКНФ в СДНФ (аналитически и с помощью диаграмм Вейча): $(x \vee y \vee z)(\bar{x} \vee \bar{y} \vee z)(x \vee y \vee z)$, $(x \vee y \vee z)(x \vee \bar{y} \vee \bar{z})$, $(x \vee y \vee z)(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee z)$, $(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee \bar{z})$.
12. Построить таблицы K_i и M_i для $n=2,3,4$.
13. Показать функциональную полноту базисов И, НЕ; ИЛИ, НЕ; И-НЕ; ИЛИ-НЕ.
14. Построить логические схемы, реализующие следующие ЛФ в базисе И-НЕ. Используя принцип дуальности построить схемы в базисе ИЛИ-НЕ: $f_1 = K_0VK_8$, $f_2 = K_1VK_3$, $f_3 = K_0VK_1VK_2$, $f_4 = K_1VK_2$, $f_5 = M_0M_1$, $f_6 = M_1M_2M_3$, $f_7 = M_0M_3$, $f_8 = M_0M_2$.
15. Найти МДНФ и МКНФ и оценить их сложность: $(x_1 \vee \bar{x}_1x_2\bar{x}_3)(x_1x_2 \vee \bar{x}_1x_3 \vee x_2x_3)$; $(x_1x_2 \vee x_1x_3 \vee \bar{x}_1\bar{x}_3)(x_1x_2 \vee \bar{x}_1x_3 \vee \bar{x}_1x_2\bar{x}_3)$; $\overline{x_1\bar{x}_3 \vee \bar{x}_1x_3} (x_1x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2x_3)$; $(x_1 \vee x_3)(x_1\bar{x}_2 \vee x_3)$; $\overline{x_1x_2x_3} (x_1x_3 \vee x_2\bar{x}_3)$.
16. Провести структурный синтез КС в базисе ИЛИ-НЕ (И-НЕ) по заданным ЛФ: $f_1 = x \vee x\bar{y} \vee xz$, $f_2 = x(x \vee \bar{y})(x \vee z)$, $f_3 = x \vee z$, $f_4 = x(y \vee z)$.
17. Пояснить принцип дуальности КС на примере структурного синтеза схемы по заданной ЛФ: $f_1 = xy \vee \bar{x}z$, $f_2 = (x \vee y)(\bar{x} \vee z)$, $f_3 = x \vee \bar{y} \vee z$, $f_4 = x\bar{y}z$.
18. Провести синтез КС, реализующей ЛФ в базисе И-НЕ: $x \vee yz$; $xyz \vee \bar{z}$; $x \vee xyz$; $xyz \vee \bar{y} \bar{z}$. Построить временные диаграммы сигналов при возбуждении КС сигналами 4-разрядного двоичного счетчика.
19. Провести синтез КС, реализующей ЛФ в базисе ИЛИ-НЕ: $(x \vee y)(x \vee \bar{y} \vee z)$; $(x \vee \bar{y})(y \vee z)$; $(\bar{x} \vee y)(x \vee \bar{z})$; $(\bar{x} \vee \bar{y})(x \vee y)$. Построить временные диаграммы сигналов при возбуждении КС сигналами 4-разрядного двоичного счетчика.
20. Записать систему функций выходов полного двоичного дешифратора при $n=2,3,4,5$. Перевести в базис И-НЕ (ИЛИ-НЕ). Построить схему дешифратора и привести условное обозначение. Нарисовать диаграммы сигналов на выходах двоичного дешифратора при периодической схеме кодовых слов на его входах.
21. Записать систему функций выходов неполного дешифратора в кодах 2-4-2-1, 4-2-2-1, 5-2-1-1, 5-4-2-1. Построить схему.

22. Синтезировать шифратор с числом входов 4,6,10,14 в базисах И-НЕ(ИЛИ-НЕ). Нарисовать схему и привести условное обозначение. Пояснить принцип приоритетности входов шифратора.
23. Записать выражение для функции выхода мультиплексора для $n= 2,3,4,5$. Нарисовать схему и привести условное обозначение.
24. Построить схему на основе мультиплексора для генерирования сигналов



25. Нарисовать структурную схему n - разрядного ПЗУ чисел, $n = 2,3,4,5,6,7,8$ на основе мультиплексора.
26. Записать выражение для функций выходов демультиплексора при $n=2,3,4,5$. Нарисовать схему и привести условное обозначение.
27. Построить дешифратор на основе демультиплексора при $n=2,3,4,5$.
28. Пояснить принцип двоичного суммирования чисел при $n=4,5,6,7$. Построить схему двоичного сумматора и привести условное обозначение.
29. Пояснить принцип каскадирования одноразрядных сумматоров при $n=2,3,4,5$.
30. Пояснить организационную структуру АЛУ.

Глава 3. Последовательностные цифровые устройства. Цифровые автоматы

3.1. Основные понятия абстрактной теории цифровых конечных автоматов

Выходные сигналы комбинационных цифровых устройств полностью определяются в любой момент времени комбинацией сигналов на их входах. В этой связи КЦУ иногда называют автоматами без памяти, т.е. КЦУ не позволяют учитывать значения логических переменных, появившихся ранее. Многие же задачи требуют учитывать предысторию процесса.

В общей теории автоматов вводится понятие *абстрактный автомат* – это математическая идеализация реального объекта или системы, перерабатывающей некоторую входную информацию. Это понятие не связано с конкретным физическим смыслом; абстрактная теория автоматов своей *главной задачей* имеет изучение общих особенностей поведения автоматов, решая в основном *вопросы анализа их внешнего функционирования*.

Прикладные разделы теории автоматов рассматривают техническую реализацию автоматов на той или иной физической базе. В прикладном, техническом смысле *цифровым автоматом* (последовательностным устройством) называется более сложный, чем КЦУ, преобразователь дискретной информации, имеющий некоторое конечное число различных внутренних состояний (так называемый *конечный цифровой автомат* – КЦА).

Под воздействием входных сигналов X (входного слова) автомат последовательно переходит из одного состояния в другое и выдает выходной сигнал (выходное слово) Y . Y определяется в общем случае поступившим *входным* словом X и *внутренним* состоянием автомата S , которое в свою очередь явилось результатом воздействия на автомат предыдущего входного сигнала.

Введем понятие *автоматного (дискретного) времени*. Для КЦА автоматное время может представлять последовательность событий, например, испытаний, номер во входной последовательности, и, наконец, реальное время t . Автоматное время удобно представлять в виде последовательности $1, 2, \dots, n$. Если интервал между двумя последовательными моментами автоматного времени постоянен, строго соответствует физическому времени, то такой автомат называется синхронным. Для асинхронного автомата нет жесткого масштабного соответствия между физическим временем и последовательностью событий в нем.

КЦА рассматривается как *абстрактный конечный автомат (АКА)*, если не исследуется его внутренняя структура. Для задания АКА вводится три конечные множества:

– входных сигналов $X = \{x_1, x_2, x_3, \dots, x_l\}$;

– выходных сигналов $Y = \{y_1, y_2, y_3, \dots, y_m\}$;

– внутренних состояний $S = \{S_1, S_2, S_3, \dots, S_p\}$.

Абстрактный автомат, как преобразователь информации, функционирует в дискретном (автоматном) времени по правилам, которые задаются двумя характеристическими функциями определенными на трех алфавитах: $X \in \{x_i\}, i = \overline{1, l}; Y \in \{y_i\}, i = \overline{1, m}; S \in \{S_k\}, k = \overline{1, p}$.

Определение **автомата Мили (Mealy)**

$$\begin{cases} Y^n = F(X^n, S^n) \\ S^{n+1} = G(X^n, S^n) \end{cases} \quad \text{– функции выходов } Y^n \text{ и переходов } S^{n+1} \quad (3.1)$$

Определение **автомата Мура (Moore)**

$$\begin{cases} Y^n = \Phi(X^n) \\ S^{n+1} = G(X^n, S^n) \end{cases} \quad \text{– функции выходов } Y^n \text{ и переходов } S^{n+1} \quad (3.2)$$

Эти функции лишь *декларируют* переходы типа $X^n, S^n \rightarrow Y^n, S^{n+1}$, но не уточняют, какими конкретно средствами задается эта функциональная связь.

3.2. Формы задания абстрактных конечных автоматов

К формам и средствам представления АКА предъявляются следующие требования:

- ⇒ **полнота представления** – исчерпывающее описание всех компонентов АКА;
- ⇒ **однозначность представления** – недопущение какой-либо неоднозначности или неопределенности в описании АКА;
- ⇒ **простота получения** – детерминированный и достаточно простой переход от описания АКА в некоторой исходной форме (словесное описание) к описанию в заданной форме;
- ⇒ **наглядность представления** – интерпретация с достаточной наглядностью всех основных свойств АКА;
- ⇒ **удобство в использовании** – применимость форм для преобразования на этапах синтеза **реальных конечных автоматов** (РКА).

3.2.1. Представление АКА с помощью таблиц переходов и выходов

Это представление одно из самых удобных и широкоупотребляемых. Таблицы переходов и выходов для АКА Мили имеют число строк, равное числу символов во входном алфавите (т. е. l) и число столбцов, равное числу символов в алфавите внутренних состояний (т.е. p). В клетках проставляются будущие внутренние состояния S^{n+1} и текущие выходные сигналы Y^n . ×àñòî àëü компактности записи таблицы переходов и выходов представляют в виде *содержимой* таблицы.

Пример. Для некоторого конкретного автомата Мили ($X \in \{x_1, x_2, x_3\}$, $Y \in \{y_1, y_2\}$, $S \in \{S_1, S_2, S_3\}$) таблицы переходов, выходов и совмещенная таблица представлены, соответственно, табл. 8, табл. 9 и табл. 10

Таблица 8

S_k	S_k		
	S_1	S_2	S_3
x_i			
x_1	S_1	S_3	S_2
x_2	S_3	S_1	S_2
x_3	S_1	S_2	S_1

Таблица 9

S_k	S_k		
	S_1	S_2	S_3
x_i			
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1
x_3	y_2	y_2	y_1

Таблица 10

S_k	S_k		
	S_1	S_2	S_3
x_i			
x_1	S_1/y_1	S_3/y_1	S_2/y_2
x_2	S_3/y_1	S_1/y_2	S_2/y_1
x_3	S_1/y_2	S_2/y_2	S_1/y_1

Для АКА Мура таблицы выходов и переходов отдельно не строятся ввиду простоты представления таблицы выходов.

Пример. Для некоторого конкретного автомата Мура ($X \in \{x_1, x_2, x_3\}$, $Y \in \{y_1, y_2\}$, $S \in \{S_1, S_2, S_3, S_4\}$) совмещенная (отмеченная) таблица представлена табл. 11

Таблица 11

Вых. сигнал	y_1	y_1	y_2	y_1
S_k				
x_i				
x_1	S_1	S_3	S_4	S_2
x_2	S_4	S_2	S_1	S_3
x_3	S_2	S_1	S_4	S_2

Все указанные таблицы (табл. 8 - 11) соответствует полностью определенным АКА Мили и Мура. При частично определенных АКА соответствующие клетки помечаются (*).

Основное достоинство табличного представления является его предельная простота составления и непосредственное отображение формул (3.1) и (3.2).

3.2.2. Представление АКА с помощью графа

Построение графов для АКА Мили и Мура сходно, за исключением отображения выходных сигналов. Внутренние состояния отображаются вершинами графа, внутренние переходы – направленными дугами. Дуги помечаются входными сигналами, которые вызывают переход. При этом для АКА Мили указывается еще выходной сигнал, сопровождающий данный переход. Выходные сигналы АКА Мура указываются у вершин графа, так как их появление не зависит от входных сигналов (в соответствии с 3.2).

Для АКА Мили граф, соответствующей табл. 10, выглядит, как показано на рис. 25, а для автомата Мура (табл. 11) граф показан на рис. 26.

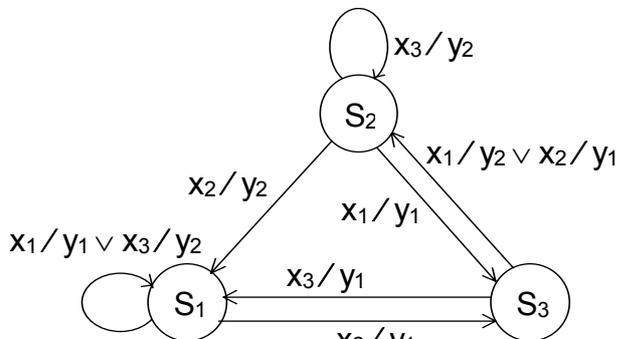


Рис. 25

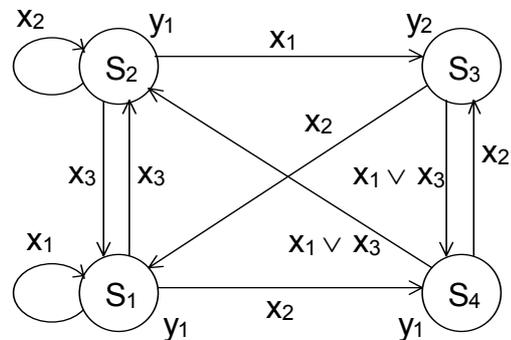


Рис. 26

Графическое представление удобно и позволяет сократить число ошибок. С помощью графа легко осуществить словарное преобразование, т. е. получить некоторую выходную последовательность символов при подаче входного слова из d символов.

Пример. Пусть для АКА Мура входное слово $X = \{x_2, x_2, x_1, x_1, x_2\}$. Считая исходным состоянием автомата S_1 , получим последовательность состояний $S = \{S_4, S_3, S_4, S_2, S_2\}$ и слово выходных символов $Y = \{y_1, y_2, y_1, y_1, y_1\}$.

3.2.3. Матричное представление АКА

Матричное представление автомата соединяет черты графического и табличного представлений. Напоминая по внешнему виду таблицу, оно, тем не менее, прежде всего отражает структурные особенности графа.

Матрица АКА Мили – квадратная, строки и столбцы соответствуют текущим и последующим состояниям. В клетках матрицы помечаются условия перехода, т. е. необходимые входные сигналы и появляющиеся при этом выходные сигналы. Матрица АКА Мура имеет дополнительный столбец сигналов выходов, зависящих от внутренних текущих состояний.

Матричное представление для вышеприведенных АКА Мили и Мура показано, соответственно, в табл. 12 и 13.

Т а б л и ц а 12

S_k^{n+1} S_k^n	S_1	S_2	S_3
S_1	$x_1/y_1 \vee x_3/y_2$	–	x_2/y_1
S_2	x_2/y_2	x_3/y_2	x_1/y_1
S_3	x_3/y_1	$x_1/y_2 \vee x_2/y_1$	–

Т а б л и ц а 13

S_k^{n+1} S_k^n	S_1	S_2	S_3	S_4	y_j^n
S_1	x_1	x_3	–	x_2	y_1
S_2	x_3	x_2	x_1	–	y_1
S_3	x_2	–	–	$x_1 \vee x_3$	y_2
S_4	–	$x_1 \vee x_3$	x_2	–	y_1

Матрица переходов позволяет с достаточной простотой обнаруживать: 1) изолированные вершины; 2) тупиковые вершины; 3) преходящие вершины (к которым нет переходов); 4) переменные (нормальные) вершины.

Пример. Для некоторого автомата Мура матрица переходов выглядит, как показано в табл. 14, из чего следует:

- S_1 – тупиковая вершина;
- S_3 – изолированная вершина;
- S_4 – преходящая вершина;
- S_2 – переменная вершина.

Т а б л и ц а 14

S_k^{n+1} S_k^n	S_1	S_2	S_3	S_4	y_j^n
S_1	$x_1 \vee x_2$	–	–	–	y_1
S_2	x_2	x_1	–	–	y_1
S_3	–	–	$x_1 \vee x_2$	–	y_2
S_4	x_2	x_1	–	–	y_1

3.3. Абстрактный синтез конечных автоматов

Основная задача любого конечного автомата – дискретное преобразование информации. Допустимо предположить наличие *эквивалентных* автоматов, которые одинаково осуществляют это преобразование.

3.3.1. Эквивалентность конечных автоматов

Два АКА будем считать эквивалентными, если выполняются следующие условия:

- ⇒ Входные и выходные алфавиты не содержат внутри себя тождественных или повторяющихся символов и могут быть взаимно-однозначно отображены друг на друга.
- ⇒ Оба автомата определены на одном и том же классе допустимых входных последовательностей символов.
- ⇒ При подаче на АКА одинаковых входных последовательностей на выходах появляются одинаковые (может быть со сдвигом на конечное число тактов автоматного времени) выходные последовательности.

Если длина входных последовательностей не ограничена, то АКА обладают полной эквивалентностью. Если длина последовательности – k символов, то автоматы k -эквивалентны. Это более слабая эквивалентность, так как на $k+1$ -ом символе может наступить неэквивалентность.

3.3.2. Эквивалентные преобразования конечных автоматов

Рассмотрим два АКА Мили и Мура, заданные с помощью своих графов (рис. 27, а,б). Положим их начальные состояния S_1 и S_1^* . Подадим на входы АКА одно и то же входное слово $X_{7,1} = \{x_1, x_2, x_2, x_1, x_1, x_2, x_2\}$. Тогда будем иметь:

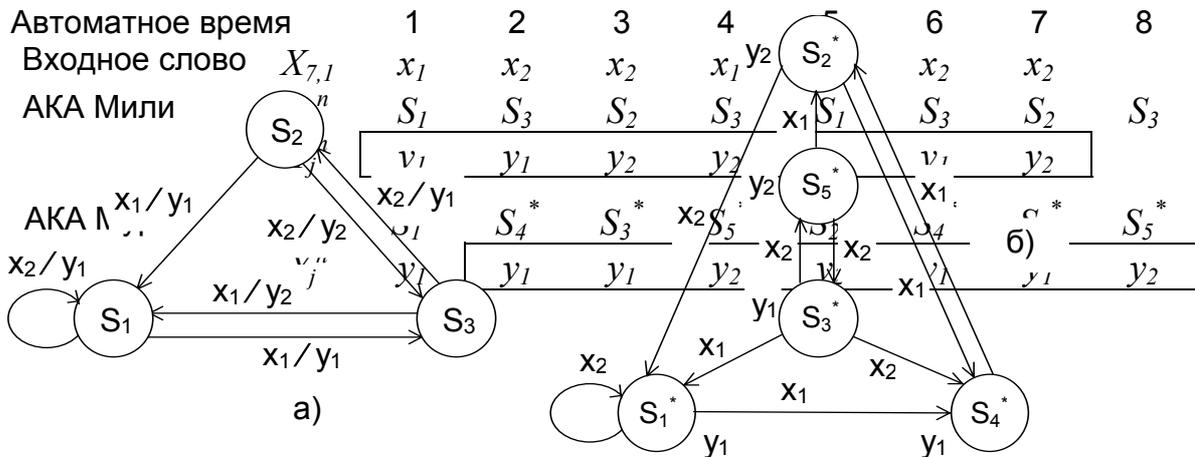


Рис. 27

Это эквивалентные АКА, они ведут себя одинаково, с той лишь разницей, что выходная последовательность АКА Мура запаздывает на один такт автоматного времени по отношению к выходной последовательности АКА Мили. Первый выходной символ АКА Мура в этой связи игнорируется.

Переход от автомата Мили к автомату Мура

С учетом того, что выходной сигнал эквивалентного АКА Мура отстает на один такт автоматного времени, выходной сигнал в клетке с координатами

$(S_k, x_i)^n$ совмещенной таблицы АКА Мили будет определять выходной сигнал для будущего внутреннего состояния S_b^{n+1} автомата Мура.

Таким образом, структура автомата Мура будет определяться парами S_b^{n+1}/y_j^n в клетках исходной таблицы с координатами $(S_k, x_i)^n$ автомата Мили. Поскольку в этой таблице одним и тем же состояниям S_b^{n+1} могут в различных клетках соответствовать различные выходные сигналы (т. е. переход в S_b^{n+1} может быть под воздействием различных входных сигналов x_i^n и при этом могут быть различные выходные сигналы y_j^n), то одному и тому же внутреннему состоянию АКА Мили могут соответствовать несколько внутренних состояний автомата Мура, каждое из которых определяется неповторяющейся парой S_b^{n+1}/y_j^n . Эти состояния можно закодировать координатами клетки $(S_k, x_i)^n \rightarrow (k, i)$.

Алгоритм перехода от АКА Мили к АКА Мура может быть представлен в следующем виде:

1. Выписать из *совмещенной* таблицы переходов и выходов АКА Мили все неповторяющиеся пары и закодировать их соответствующими состояниями АКА Мура S_b^* .
2. Соотнести каждому состоянию S_k автомата Мили множество состояний автомата Мура S_b^* .
3. Заполнить переходы в *отмеченной* таблице переходов и выходов АКА Мура следующим образом. Если состояние автомата Мура S_b^* относится к множеству S_k состояний автомата Мили, то в столбец переходов из S_b^* под воздействием x_i необходимо проставить состояния S_{bi}^* , соответствующие столбцу переходов из S_k автомата Мили.
4. Выходные сигналы для каждого состояния S_b^* находятся в клетке с координатами (k, i) , т.е. в паре S_b^{n+1}/y_j^n .

Пример. Преобразовать АКА Мили (рис.27,а, табл. 15) в эквивалентный ему АКА Мура. Следуя изложенному алгоритму, получим:

Т а б л и ц а 15

S_k	S_1	S_2	S_3
x_1	S_3/y_1	S_1/y_1	S_1/y_2
x_2	S_1/y_1	S_3/y_2	S_2/y_1

1. $S_1/y_1 \rightarrow S_{21}^* = S_{12}^* \Rightarrow S_1^*$
 $S_1/y_2 \rightarrow S_{31}^* \Rightarrow S_2^*$
 $S_2/y_1 \rightarrow S_{32}^* \Rightarrow S_3^*$
 $S_3/y_1 \rightarrow S_{11}^* \Rightarrow S_4^*$
 $S_3/y_2 \rightarrow S_{22}^* \Rightarrow S_5^*$
2. $S_1 \in \{S_1^*, S_2^*\}$
 $S_2 \in \{S_3^*\}$
 $S_3 \in \{S_4^*, S_5^*\}$

3, 4. Совмещенная таблица АКА Мура.

Т а б л и ц а 16

y_j	y_1	y_2	y_1	y_1	y_2
S_k	S_1^*	S_2^*	S_3^*	S_4^*	S_5^*
x_i	S_4^*	S_4^*	S_1^*	S_2^*	S_2^*
x_1	S_4^*	S_4^*	S_1^*	S_2^*	S_2^*
x_2	S_1^*	S_1^*	S_5^*	S_3^*	S_3^*

Например, S_4^* относится к S_3 (см. п.2), тогда в столбец переходов для S_4^* подставляются состояния для переходов из S_3 , т. е. $S_1/y_2 \rightarrow S_2^*$ и $S_2/y_1 \rightarrow S_3^*$.

Таким образом, совмещенная таблица соответствует АКА Мура, граф которого представлен на рис. 27,б.

Переход от автомата Мура к автомату Мили

Для осуществления обратного перехода от автомата Мура к автомату Мили удобно воспользоваться представлением автомата Мура в виде графа. Сущность преобразования заключается в том, что выходные сигналы, относящиеся к вершинам автомата Мура, будут помечать все дуги, идущие в эту вершину. Тем самым одновременно производится изменение момента автоматного времени, к которому относится сам выходной сигнал, на один такт вперед.

Пример. Преобразовать АКА Мура (рис. 27,б) в АКА Мили. После указанных выше преобразований с выходными сигналами, граф автомата будет выглядеть, как показано на рис. 28.

Полученный граф автомата Мили отличается по числу состояний от рассмотренного на рис. 27,а графа автомата Мили, хотя и эквивалентен ему.

Здесь возникает вопрос о минимизации числа состояний абстрактных автоматов.

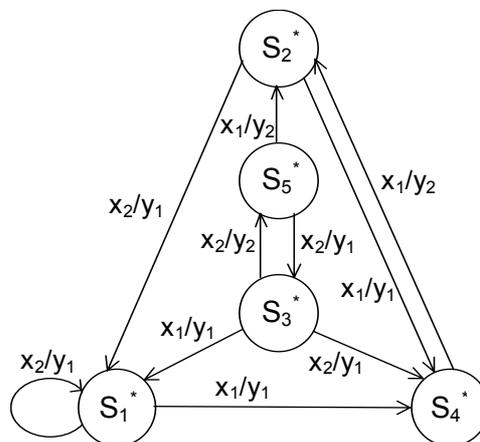


Рис. 28

3.3.3. Минимизация числа состояний абстрактных автоматов

Сущность метода минимизации заключается в разбиении всего его алфавита внутренних состояний на попарно непересекающиеся классы эквивалентных состояний с заменой далее каждого класса эквивалентности одним состоянием. Получающийся в результате минимальный автомат имеет столько же состояний, на сколько классов эквивалентности разбивается все множество внутренних состояний заданного автомата.

Эквивалентными называются такие два состояния автомата, замена которых одного на другое не изменяет результатов преобразования входной последовательности на всем множестве символов. Можно как и ранее говорить о полной эквивалентности (π), так и k -эквивалентных внутренних состояниях (π_k).

Процедура минимизации выглядит следующим образом.

1. Находятся последовательные разбиения $\pi_1, \pi_2, \dots, \pi_k$ алфавита внутренних состояний на классы $1, 2, \dots, k$ -эквивалентных состояний, до тех пор, пока на каком-то шаге $\pi_k = \pi_{k+1}$. Очевидно, при этом можно утверждать, что k -эквивалентное состояние является полностью эквивалентным. Число шагов процедуры не превышает $p-1$, где p – размер алфавита внутренних состояний.

2. В каждом классе эквивалентности выбирается по одному символу, которые и составляют новый алфавит внутренних состояний минимизированного автомата.
3. Таблицы переходов и выходов минимизированного автомата получаются из таблиц исходного автомата путем вычеркивания столбцов с состояниями, не вошедшими в минимизированный алфавит, и замены в оставшихся столбцах внутренних состояний исходного автомата эквивалентными им состояниями минимизированного автомата.
4. В качестве начального выбирается начальное состояние исходного автомата или любое, ему эквивалентное.

Пример. Минимизировать автомат Мили, заданный графом на рис. 28.

Т а б л и ц а 17

Совмещенная таблица переходов и выходов автомата представлена табл. 17.

S_k	S_1^*	S_2^*	S_3^*	S_4^*	S_5^*
x_i					
x_1	S_4^*/y_1	S_4^*/y_1	S_1^*/y_1	S_2^*/y_2	S_2^*/y_2
x_2	S_1^*/y_1	S_1^*/y_1	S_5^*/y_2	S_3^*/y_1	S_3^*/y_1

1. Проведем разбиение на класс 1-эквивалентности по реакции на слово длиной в один символ ($x_1 \vee x_2$). В понятие **реакции** относится только **выходной сигнал**, поскольку основным назначением автомата является осуществление словарного преобразования.

Для класса π_1 выполняется:

$$\pi_1 = \{S_1^1, S_2^1, S_3^1\}, \text{ где } S_1^1 = \{S_1^*, S_2^*\}, S_2^1 = \{S_3^*\}, S_3^1 = \{S_4^*, S_5^*\}.$$

Результаты разбиения отражены в табл. 18.

Т а б л и ц а 18

2. Дальнейшее разбиение на классы приводит к тому, что $\pi_2 = \pi_1 = \pi$. Перекодировав оставшиеся состояния

$S_1^1 \rightarrow S_1; S_2^1 \rightarrow S_2; S_3^1 \rightarrow S_3$ получим совмещенную таблицу автомата (табл. 15), соответствующую АКА Мили, заданного графом на рис. 27,а.

x_i	S_k				
	S_1^1		S_2^1	S_3^1	
	S_1^*	S_2^*	S_3^*	S_4^*	S_5^*
x_1	S_3^1	S_3^1	S_1^1	S_1^1	S_1^1
x_2	S_1^1	S_1^1	S_3^1	S_2^1	S_2^1

3.4. Структурный синтез конечных автоматов

Под структурным синтезом конечных автоматов будем понимать построение реальных двоичных цифровых устройств, реализующих функции заданных АКА. При этом будем полагать, что вопрос минимизации количества внутренних состояний АКА уже решен.

3.4.1. Задачи структурного синтеза конечных автоматов

Как правило, для построения цифровых устройств используются двоичные триггеры, число внутренних состояний которого равно двум. Триггеры в этой связи называют **элементарными автоматами** (ЭА) или простейшими **элементами памяти** (ЭП). Таким образом, **задача структурного синтеза на уровне логических схем сводится к построению автоматов из элементарных**

автоматов и комбинационных схем. Исходная информация для структурного синтеза может быть взята из той или иной формы задания автомата.

Структурный синтез начинается с двоичного кодирования внутренних состояний автомата, что задает взаимно-однозначное соответствие между состояниями автомата и комбинациями состояний ЭА. Двоичное кодирование позволяет поведение синтезируемого автомата, первоначально заданного, к примеру, таблицей переходов и выходов, описать в терминах двузначной логики, т. е. знакомых нам выражениях булевой алгебры. В качестве таких выражений используются **функции внешних переходов ЭА**, определяющие изменение состояний каждого из ЭА (включая рассматриваемый) и приходящих на автомат входных сигналов. Слово “внешний” подчеркивает тот факт, что функционирование ЭА в синтезируемой структуре зависит не только от его собственного состояния, но и от состояний других ЭА, входящих в эту структуру.

Функции внешних переходов зависят от выбранного способа кодирования внутренних состояний синтезируемого автомата, следовательно, можно предположить наличие *оптимального кодирования*.

Для того, чтобы каждый ЭА работал в соответствии со своей функцией внешних переходов, на его входах необходимо наличие строго определенных управляющих сигналов. Эти сигналы определяются по логическим выражениям, которые называются **функциями возбуждения ЭА**.

Функции возбуждения зависят не только от функции внешних переходов, но и от его собственного, внутреннего функционирования. Естественно, что для *разных ЭА* следует формировать *разные функции возбуждения*. Отсюда следует *задача оптимального выбора типа ЭА*, приводящего к наипростейшей форме выражений для функций возбуждения, требующей минимальных затрат для их реализации.

Двоичное кодирование внутренних состояний АКА позволяет также решить *задачу формирования выходных сигналов* синтезируемого устройства. Функции выходов, заданные таблицей выходов, преобразуются к системе комбинационных логических выражений, описывающих двоичные выходные сигналы, как функции входных сигналов автомата и состояний составляющих его элементарных автоматов. Сложность этих логических выражений зависит от способа кодирования, поэтому и здесь можно говорить об оптимальном кодировании внутренних состояний автомата.

3.4.2. Блок-схема конечного автомата

В отличие от АКА с одним входом X и одним выходом Y реальный конечный автомат имеет l входных и m выходных сигналов, каждый из которых может принимать значения 0 и 1 . Это следует из определения входных и вы-

ходных алфавитов ($X \in x_i, i = \overline{1, l}, Y \in y_j, j = \overline{1, m}$) и кроме того, для внутренних состояний $S \in S_k, k = \overline{1, p}$.

Структура реального конечного автомата выглядит как показано на рис. 29. В структуре РКА можно выделить две части: комбинационную схему, состоящую из двух КЦУ, и запоминающее устройство (ЗУ), содержащее N простейших элементов памяти -

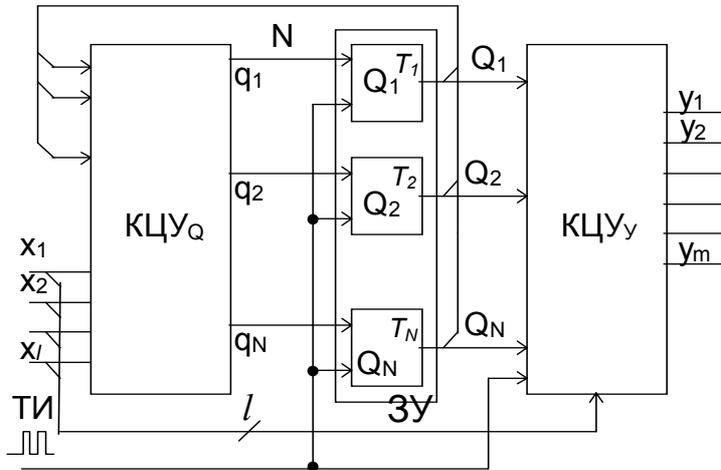


Рис. 29

триггеров T_1, T_2, \dots, T_N , на которые действуют сигналы возбуждения $\{q_i\}$.

Под воздействием q_i в моменты времени, определенные тактовыми импульсами (ТИ), триггер T_i переходит в одно из двух состояний: 0 или 1. Состояние T_i отражается сигналом Q_i , а совокупность состояний всех триггеров $\{Q_i\} = \{Q_1, Q_2, Q_3, \dots, Q_N\}$ отража-

ет внутреннее состояние РКА. Если имеется N триггеров, то возможное число внутренних состояний равно 2^N .

Комбинационные цифровые устройства КЦУ_Y и КЦУ_Q описываются логическими функциями: $Y^n = F_1(X^n, Q^n)$ и $\{q_i\}^n = \Phi(Q^n, X^n)$

РКА работает под воздействием входных сигналов X , поступающих в моменты автоматного времени $1, 2, \dots, n$, задаваемого ТИ. При этом автомат переходит из одного состояния в другое в соответствии с функцией переходов и вырабатывает выходные сигналы в соответствии с функцией выходов.

Таким образом РКА последовательно преобразовывает последовательность кодов X разрядностью l в последовательность кодов Y разрядностью m (îîëääîââðâëüîîîîâ òëòðîîâ òñòðîéîðâî).

Наибольшая трудность в синтезе РКА – получение функций возбуждения q_i^n , в зависимости от которых каждый элементарный автомат-триггер перейдет в следующее состояние Q_i^{n+1} . Трудоемкость получения функций возбуждения существенно зависит от N и l . При ручном проектировании рекомендуется $N + l \leq 8$. Увеличение разрядности m выходного кода практически автомат не усложняет, здесь ограничений нет.

Синтезируемый автомат можно проектировать как автомат Мура или как автомат Мили (с учетом связей, показанных пунктиром на рис. 29). В том и другом случае, РКА может быть построен по синхронному или асинхронному принципу. Как правило, входные сигналы задаются в логике уровней, поэтому “синхронность” или “асинхронность” автомата определяется заданием автоматного времени, т. е. характером реакции автомата на входной сигнал (рис. 30).

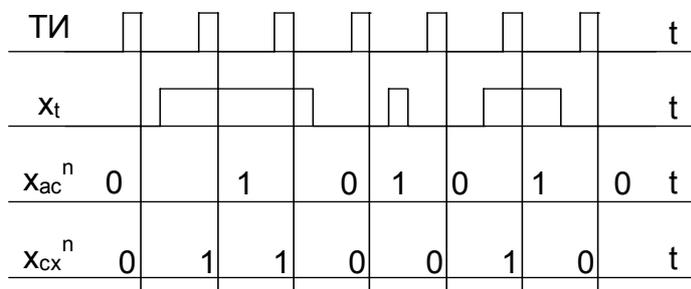


Рис. 30

(рис. 30).

Для синхронного автомата входное двоичное слово x^t интерпретируется в соответствии с тактирующими импульсами ТИ (автоматное время) как 0110010, а для асинхронного – как 0101010, в соответствии с уровнями на входе.

В асинхронных КА сигналы $Q_1, Q_2, Q_3, \dots, Q_N$ изменяются с некоторой задержкой относительно входного сигнала, в общем случае ненормируемой. Отсюда следует, что каждому изменению сигнала на входе соответствует два, в общем случае разных сигнала обратной связи $Q_1, Q_2, Q_3, \dots, Q_N$, которые должны формировать одни и те же функции возбуждения элементарных автоматов. Чтобы разброс в скорости срабатывания ЭА не приводил к нарушению детерминированности переходов, два соседних состояния РКА должны различаться лишь состоянием одного ЭА. Это влечет жесткие ограничения к характеру кодирования внутренних состояний автомата и необходимости использования значительного числа ЭА, в пределе – число внутренних состояний равно числу ЭА. Такое усложнение иногда окупается максимальным быстродействием устройства.

В синхронных КА входные сигналы ЭА изменяются в промежутках между синхронизирующими импульсами и остаются неизменными в интервалах опроса входных сигналов синхросигналами. Это обстоятельство обеспечивает устойчивую работу автомата при изменении состояний *любого* числа ЭА (не нарушается детерминированность его переходов), что снимает ограничения на кодирование внутренних состояний автомата и позволяет использовать все возможные состояния элементарных автоматов.

Таким образом, необходимое для синтеза количество ЭА N определяется числом p внутренних состояний КА $N = \text{int}[\log_2 p]$, где int – ближайшее большее целое число к $\log_2 p$. Хотя синхронные автоматы имеют меньшее быстродействие, чем асинхронные, они получили большее распространение в силу их простоты, меньшей чувствительности к скорости изменения входных сигналов и не критичности к временным характеристикам составляющих их логических элементов. Далее мы будем рассматривать синхронные КА.

3.5. Элементарные автоматы (триггеры) и их свойства

Элементарный конечный автомат (ЭКА) или **триггер** представляет собой устройство, имеющее два устойчивых состояния и обладающее *полнотой переходов и выходов*.

Автомат (триггер) обладает *полнотой переходов*, если есть хотя бы один входной сигнал, который переводит автомат из одного состояния в другое. В графе триггера нет тупиков и преходящих вершин.

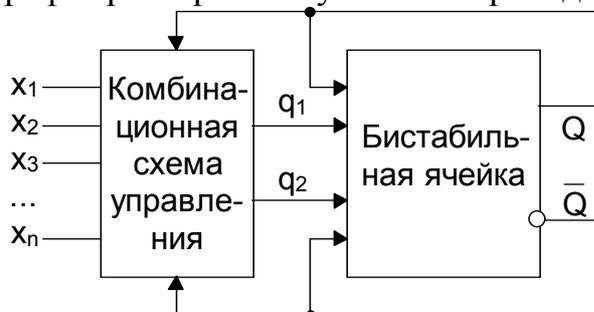


Рис. 31

Триггер обладает *полной системой выходов*, если два его внутренних состояния, различны и проявляют себя в выходах, т.е. двум различным состояниям всегда соответствуют два разных выходных сигнала. Это означает, что все **триггеры - автоматы Мура**, у которых выходной сигнал не зависит от входного, а определяется

только внутренним состоянием автомата.

Некоторые общие положения :

- ◆ Внутреннее состояние триггера обозначается $Q \in (0,1)$.
- ◆ Выходной сигнал повторяет внутреннее состояние и тоже обозначается Q .
- ◆ Триггер обычно имеет парафазный выход Q и \bar{Q} (для удобства).
- ◆ В синхронных триггерах имеется вход синхронизации, который тактирует изменение состояний триггера.
- ◆ Сигнальные входы делятся на информационные и установочные.

По установочным входам триггер устанавливается принудительно в единичное или нулевое состояния: R (*reset*) - сброс в "0", и S (*set*) - установка в "1".

Информационные сигналы q (сигналы возбуждения) подготавливают триггер к переходу в то или иное состояние по тактирующим сигналам (по уровню, фронту или спаду импульса). Триггеры, как ЭКА, описываются таблицей переходов, графом или матрицей переходов.

Наличие двух устойчивых состояний и возможность перехода из одного состояния в другое объясняется использованием в любом триггере бистабильной ячейки (схема с отрицательной обратной связью с выхода на вход) и комбинационной схемы, управляющей переключениями бистабильной ячейки в зависимости от входных сигналов $x_1, x_2, x_3, \dots, x_n$ и ее внутреннего состояния Q (рис. 31).

По способу приема информации триггеры делятся на *синхронные* и *асинхронные*. По числу информационных входов $x_1x_2x_3...x_n$ различают одно- (**D**, **T**), двух- (**RS**, **JK**, **DV**) и многовходовые триггеры.

3.5.1. Синхронные триггеры

Синхронные триггеры реагируют на информационные управляющие сигналы при наличии (изменении) сигнала на специальном входе **C** (clock - часы) синхронизации (тактирующем). Если *синхронизация* осуществляется уровнем 0 или 1 сигнала на **C**-входе, то она называется *статической*. В том случае, когда состояние триггера изменяется по фронту или спаду сигнала на входе **C** (переходу из 0 в 1 или из 1 в 0), то синхронизация называется *динамической*.

RS-триггер со статической синхронизацией. Синхронный **RS**-триггер отличается от асинхронного наличием синхросигнала **C**, который подается на входное КЦУ (рис. 32). Бистабильная ячейка этого триггера, а также входное КЦУ построены на основе ЛЭ И-НЕ. Из схемы следует, что сигналы по **RS** входам воздействуют на выходной бистабильный элемент только при условии $C=1$. Если $C=0$, то входные ЛЭ И-НЕ блокированы (их выходы принимают значения 1 и не зависят от входов **RS**). Функция переходов триггера $Q^{t+1} = Q^t(\bar{C}^t \vee \bar{R}^t) \vee C^t S^t$ может быть получена из диаграммы Вейча для $Q^{t+1} = f(Q^t, S^t, R^t, C^t)$. При $C=1$ функция переходов синхронного **RS**-триггера сводится к функции переходов асинхронного **RS**-триггера.

D-триггер с динамической синхронизацией. **D**-триггер (delay - задержка) формирует выходной сигнал, совпадающий с входным, но задержанный относительно него на один такт синхроимпульсов. Условное графическое обозначение **D**-триггера и граф переходов показаны на рис. 28. Переходы триггера из состояния Q^n в момент времени n в состояние Q^{n+1} в момент времени $n+1$ в зависимости от сигнала на **D**-входе и состояния Q^n отражены в таблице переходов (табл. 19). Матрица переходов триггера, в которой отражена зависимость переходов $Q^n \rightarrow Q^{n+1}$ в функции от D^n , приведена

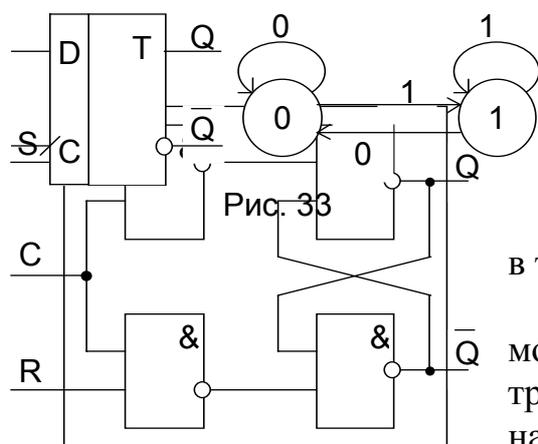


Рис. 32

Таблица 19

D^n	Q^n	Q^{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Таблица 20

$Q^n \rightarrow Q^{n+1}$	D^n
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1

в табл. 20.

Из таблицы переходов следует зависимость $Q^{n+1} = D^n$, что отражает свойство **D**-триггера как элемента задержки. Показанный на рис. 33 **D**-триггер срабатывает по фронту (переход из 0 в 1) на синхровходе; примерами

таких триггеров являются К155ТМ2, К133ТМ2 и др.

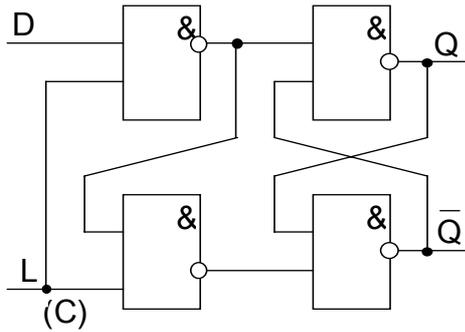


Рис. 34

DL-триггер. Схема DL-триггера (latch - защелка) показана на рис. 34. Функционирование этого триггера не отвечает табл. 19. При значении сигнала на входе синхронизации $C=L=1$ DL-триггер формирует выходной сигнал, совпадающий с входным ($Q=D$). Триггер как бы “прозрачен”, т.е. сигнал со входа D проходит на выход Q . Как только сигнал на входе L переходит из 1 в 0, то в этот момент сигнал на выходе Q фиксируется и остается неизменным, пока

$L=0$. Такие триггеры-защелки широко используются для запоминания информации по команде на синхровходе (К155ТМ5, К155ТМ7 и др.), а также в регистрах-фиксаторах (К589ИР12, К580ИР82/83).

T-триггер с динамической синхронизацией. Сигнал $T=0$ задает режим хранения информации ($Q^{n+1} = Q^n$), а сигнал $T=1$ вызывает переход триггера в противоположное состояние ($Q^{n+1} = \bar{Q}^n$). Условное графическое обозначение T-триггера (trigger - курок) и граф его переходов показаны на рис. 35. Таблица переходов и матрица переходов T-триггера приведены, соответственно, в табл. 21 и 22.

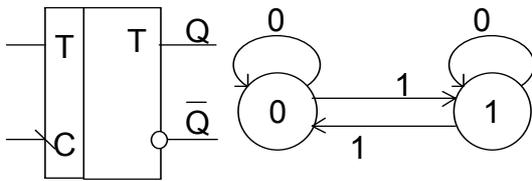


Рис. 35

Таблица 21

T^n	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Таблица 22

$Q^n \rightarrow Q^{n+1}$	T^n
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1
$1 \rightarrow 1$	0

Из таблицы переходов следует зависимость $Q^{n+1} = \bar{Q}^n T^n \vee Q^n \bar{T}^n$, что отражает свойство T-триггера работать в режимах хранения и инверсии информации. Показанный на рис. 35 T-триггер срабатывает по спаду импульса (переход из 1 в 0) на синхровходе.

JK-триггер с динамической синхронизацией. Условное графическое обозначение JK-триггера (*jump* – переход, *keep* – хранение), срабатывающего по спаду (переход из 1 в 0) сигнала на синхровходе, а также граф переходов показан на рис. 36. Таблица переходов и матрица переходов JK-триггера приведены, соответственно, в табл. 23 и 24.

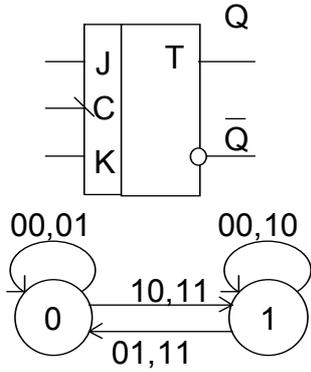


Рис. 36

Таблица 23

J^n	K^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Таблица 24

$Q^n \rightarrow Q^{n+1}$	J^n	K^n
$0 \rightarrow 0$	0	b_1
$0 \rightarrow 1$	1	b_2
$1 \rightarrow 0$	b_3	1
$1 \rightarrow 1$	b_4	0

В матрице переходов $b_1 - b_4$ - неопределенные коэффициенты, которые при минимизации могут

принимать произвольные значения (0 или 1), причем для JK-триггера эти коэффициенты взаимно независимы для любого перехода $Q^n \rightarrow Q^{n+1}$. Из таблицы переходов следует $Q^{n+1} = J^n \bar{Q}^n \vee \bar{K}^n Q^n$. В работе JK-триггера можно выделить:

- ⇒ режим хранения информации ($J=K=0$), при котором состояние триггера не меняется;
- ⇒ инверсный режим ($J=K=1$), при котором на каждый синхроимпульс состояние триггера меняется на противоположное;
- ⇒ режим перехода в 1 ($J=1, K=0$), при этом на выходе устанавливается $Q=1$;
- ⇒ режим перехода в 0 ($J=0, K=1$), когда на выходе устанавливается $Q=0$. JK-триггеры выпускаются промышленно (К155ТВ1).

RS-триггер с динамической синхронизацией. Условное графическое обозначение RS-триггера (reset - сброс в 0, set - установка в 1), срабатывающего по спаду (переход из 1 в 0) сигнала на синхровходе, а также граф переходов показан на рис. 37. Таблица переходов и матрица переходов RS-триггера приведены, соответственно, в табл. 25 и 26. В отличие от JK-триггера, для RS-

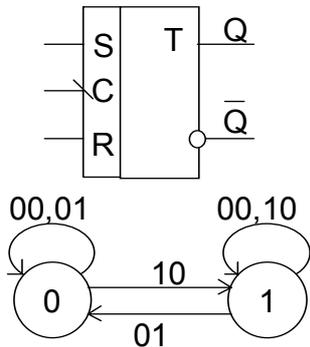


Рис. 37

Таблица 25

S^n	R^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	*
1	1	1	*

Таблица 26

$Q^n \rightarrow Q^{n+1}$	R^n	S^n
0 → 0	b_1	0
0 → 1	0	1
1 → 0	1	0
1 → 1	0	b_2

триггера недопустима комбинация $R=S=1$.

В матрице переходов b_1 и b_2 - неопределенные

независимые коэффициенты. Из таблицы переходов (табл. 25) с учетом невозможности $R=S=1$, следует $Q^{n+1} = S^n \vee \bar{R}^n Q^n$. В работе RS-триггера отсутствует режим инверсии состояния. В остальном, если принять $S=J$ и $R=K$, то RS- и JK-триггеры работают одинаково.

DV-триггер с динамической синхронизацией. DV-триггер функционирует аналогично вышеописанному D-триггеру, если сигнал на входе V равен 1, в противном случае (при $V=0$) DV-триггер находится в режиме хранения информации. Таким образом, сигнал по входу V (validation - подтверждение) играет роль сигнала переключения DV-триггера из режима хранения в режим D-триггера. Условное графическое обозначение D-триггера и граф переходов показаны на рис. 38. Таблица переходов и матрица переходов DV-триггера приведены, соответственно, в табл. 27 и 28.

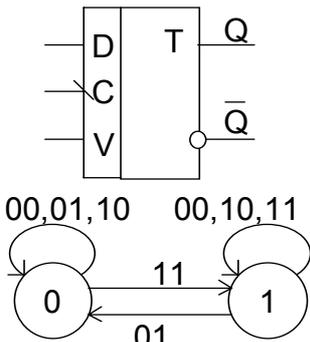


Рис. 38

В матрице переходов b_1 и

Таблица 27

V^n	D^n	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	1	1
1	1	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Таблица 28

$Q^n \rightarrow Q^{n+1}$	D^n	V^n
0 → 0	b_1	b_1^*
0 → 1	1	1
1 → 0	0	1
1 → 1	b_2	b_2^*

b_1^* , b_2 и b_2^* - неопределенные попарно зависимые коэффициенты, которые при минимизации должны отвечать условию:

$b_1 b_1^* \neq 1$, $b_2 \vee b_2^* \neq 0$. Из таблицы переходов следует $Q^{n+1} = D^n V^n \vee \bar{V}^n Q^n$.

Синхронные двухступенчатые триггеры

В *двухступенчатых* триггерах имеются две ступени (два триггера) запоминания информации (**Master** - основная, **Slave** - дополнительная). Вначале информация заносится в первую ступень, а затем переписывается во вторую и появляется на выходе.

Универсальный JK-триггер с входной логикой. Функциональная схема универсального JK-триггера с логическими элементами И по JK- входам и его условное графическое обозначение показаны на рис. 39. Входы J_1, J_2, J_3 , а также K_1, K_2, K_3 конъюнктивно связаны между собой ($J=J_1J_2J_3, K=K_1K_2K_3$), что позволяет использовать эту логику для построения схем.

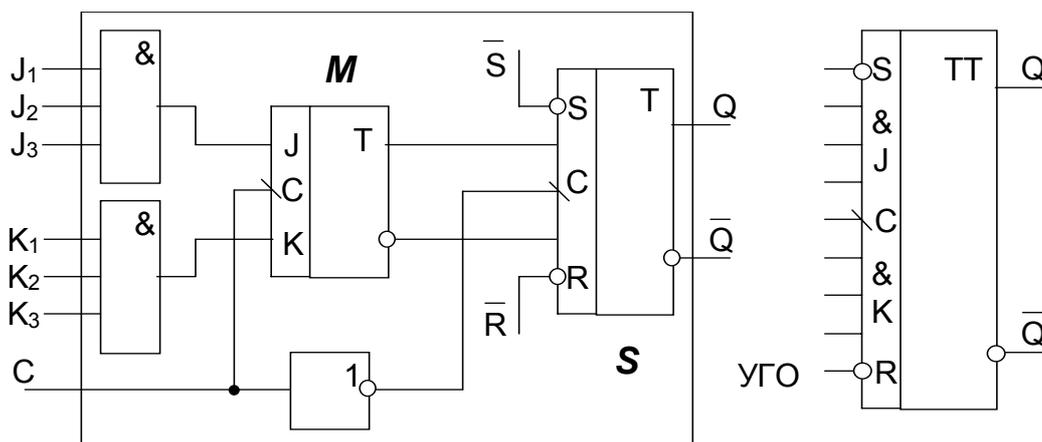


Рис. 39

Триггер может иметь асинхронные установочные входы: \bar{S} - установка в 1; \bar{R} - установка в 0. Сигналы на этих входах принудительно и независимо от всех других сигналов приводят к появлению на выходе дополнительного триггера, соответственно, 0 или 1, причем в этих состояниях триггер продолжает оставаться, не реагируя на информационные сигналы, до тех пор, пока имеется активный уровень на \bar{S} или \bar{R} входе.

Большинство триггеров с динамической синхронизацией построены по двухтактной схеме (К155ТМ2, К155ТВ1) и имеют входы асинхронной установки в исходное состояние (0 или 1). В зарубежных обозначениях это триггеры типа FF (flip-flop).

На базе универсального JK-триггера можно построить любые виды синхронных триггеров путем несложных внешних коммутаций [1, 2]. В инженерной практике часто стоит задача построения одних триггеров на базе имеющихся в распоряжении. Задача имеет простое общее решение, если рассматривать триггеры как элементарные конечные автоматы.

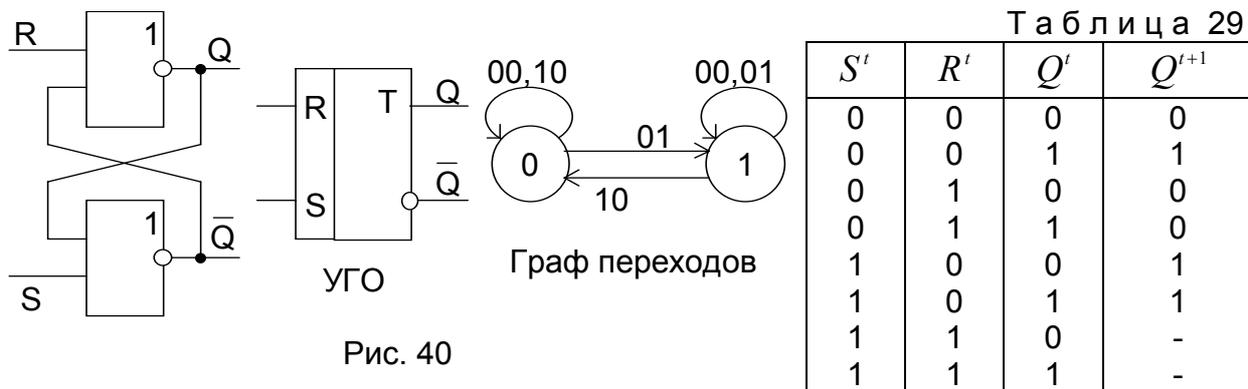
3.5.2. Асинхронные триггеры

Асинхронные триггеры воспринимают информационные сигналы и реагируют на них в момент появления на входах триггера. Сигнал на выходе

асинхронного триггера появляется при этом с задержкой, определяемой особенностями реализации триггера.

Эти триггеры существенно проще, чем синхронные, но промышленностью практически не выпускаются (106 серия). Они тем не менее часто реализуются на базе логических элементов.

RS - триггер с H-активными входами. Это простейшая бистабильная ячейка с использованием ЛЭ ИЛИ-НЕ (рис. 40). Переключение триггера из одного состояния в другое происходит входными сигналами по достижении ими уровней срабатывания ЛЭ. Потенциал логической единицы (*H-high*) по входу *S* (set) устанавливает триггер в состояние $Q=1$, а по входу *R* (reset) - в состояние $Q=0$. При $S=R=0$ триггер хранит предыдущее состояние. Подача *H*-уровней одновременно по обоим входам *S* и *R* запрещена, поскольку триггер переходит в неопределенное состояние.



Из графа переходов и таблицы переходов (табл. 29) *RS*-триггера следует, что состояние триггера в момент $t+1$ (функция переходов) $Q^{t+1} = S^t \vee \bar{R}^t \cdot Q^t$, при этом должно выполняться условие $R^t \cdot S^t \neq 1$.

$\bar{R}\bar{S}$ -триггеры с L-активными входами. Это бистабильная ячейка с использованием ЛЭ И-НЕ. Переключение триггера из одного состояния в другое происходит потенциалами логического нуля (*L-low*); по входу $\bar{S}=0$ устанавливает триггер в состояние $Q=1$, а по входу $\bar{R}=0$ - в состояние $Q=0$. При $\bar{R} = \bar{S} = 1$ триггер хранит предыдущее состояние. Запрещенной является комбинация, при которой $\bar{R} = \bar{S} = 0$. Схема, граф переходов и таблица переходов выглядят подобно рис. 40 и табл. 29.

3.6. Определение функций возбуждения триггеров

Функции возбуждения ЭА могут быть получены, если известны функции внешних переходов для каждого ЭА из таблиц переходов РКА.

3.6.1. Кодирование таблиц переходов

Кодирование внутренних состояний конечного автомата (и таблиц его переходов) заключается в установлении соответствия между состояниями авто-

мата и двоичными кодами состояний элементарных автоматов. Выбор того или иного варианта кодирования не изменяет закон внешнего функционирования автомата, но значительно влияет на сложность функций возбуждения его ЭА. Предполагается, что для всех типов триггеров простота функций возбуждения определяется простотой функций внешних переходов, поэтому *задачу оптимального кодирования определяют как задачу минимизации системы логических функций внешних переходов.*

Задача оптимального кодирования в общем виде не решена, но даже приближенные алгоритмы очень сложны и неудобны. Приведем инженерные приемы, позволяющие в некотором отношении оптимизировать кодирование таблиц переходов. Пусть таблица кодирования состояний РКА представлена табл. 30.

1. Значение b_{ij} нужно выбирать равным 0 или 1, так чтобы двоичные коды состояний РКА были различными. Необходимое число ЭА $N = \text{int}[\log_2 p]$. Если $p = 2^d$, где d – целое число, то при кодировании будут использованы все двоичные наборы $Q_1 Q_2 Q_3 \dots Q_N$, в противном случае – только часть их.

Т а б л и ц а 30

Состояние РКА	Состояние ЭА			
	Q_1	Q_2		Q_N
S_1	b_{11}	b_{12}		b_{1N}
S_2	b_{21}	b_{22}		b_{2N}
S_k				
S_p	b_{p1}	b_{p2}		b_{pN}

2. Для минимизации функций внешних переходов целесообразно, чтобы переходы конкретного ЭА из одного состояния в другое зависели бы минимального числа других ЭА. Следовательно, необходимо соседние по переходам состояния РКА кодировать минимально отличающимися друг от друга кодами, например, соседними значениями циклического кода (кода Грея).

3. Степень оптимальности кодирования оценивается по конфигурации диаграмм Вейча, описывающих его функции внешних переходов. Диаграммы можно “подправить” путем перекодировки состояний, что приведет к перемещению нулей и единиц в диаграмме и появлению возможностей объединения в большие группы. Не всякая перекодировка минимизирует логические функции: замена переменных их отрицаниями или циклическая перестановка переменных не упрощает исходные логические функции, количество и характер членов МДНФ, МКНФ остается прежним.

4. Если при кодировании задача минимизации не ставится, то для упрощения дальнейшей работы по получению функций возбуждения, *столбцы таблицы переходов кодируют по законам кодирования диаграмм Вейча*, когда первый столбец кодируется нулевым кодом, а каждый последующий – возрастающим на единицу значением N -разрядного циклического кода.

3.6.2. Определение функций внешних переходов

Целесообразно функции внешних переходов определять в минимальных нормальных формах булевых функций в следующей последовательности:

1. Расположение столбцов и строк изменяется перестановкой таким образом, чтобы входные переменные и состояния элементарных автоматов кодировали таблицу переходов, как это принято при построении диаграмм Вейча-Карно.

2. Таблица переходов “расслаивается” на N диаграмм Вейча функций внешних переходов. При этом, если код состояния РКА записывается в порядке Q_N, Q_{N-1}, \dots, Q_1 , то все правые двоичные цифры внутренних клеток таблицы переходов формируют диаграммы Вейча для первого ЭА Q_1 и т. д., и, наконец все левые – для ЭА Q_N .

3. Определяют МДНФ (МКНФ) функций внешних переходов всех ЭА

$$\begin{aligned} Q_1^{n+1} &= f_1(x_1, x_2, \dots, x_l, Q_1, Q_2, \dots, Q_N)^n \\ Q_2^{n+1} &= f_2(x_1, x_2, \dots, x_l, Q_1, Q_2, \dots, Q_N)^n \\ &\dots \\ Q_N^{n+1} &= f_N(x_1, x_2, \dots, x_l, Q_1, Q_2, \dots, Q_N)^n \end{aligned} \quad (3.3)$$

3.6.3. Определение функций возбуждения элементарных автоматов

Функции возбуждения элементарных автоматов (триггеров) описывают значения входных сигналов D, T, RS, JK, DV как функций входных сигналов x_1, x_2, \dots, x_l и выходных сигналов триггеров Q_1, Q_2, \dots, Q_N . При оптимизации структуры устройства целесообразно определить функции возбуждения для всех видов триггеров, после чего используют те из них, функции возбуждения которых требуют минимальных затрат для своей реализации. Рассмотрим три метода определения функций возбуждения.

3.6.3.1. Табличный метод определения функций возбуждения триггеров

Наиболее употребимый в инженерной практике *табличный метод* определения функций возбуждения обеспечивает кратчайший переход от диаграмм Вейча внешних переходов триггера к диаграммам Вейча функций возбуждения.

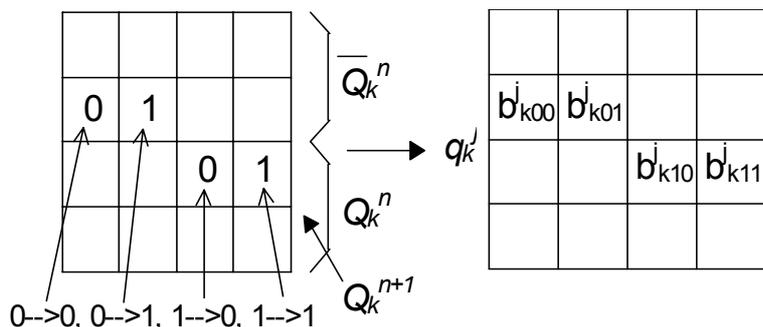


Рис. 41

Диаграмма Вейча k -ого ЭА в виде СДНФ можно условно разбить на две половины, соответствующие всем возможным переходам из единичных (Q_k) и нулевых ($\overline{Q_k}$) состояний ЭА (рис. 41).

Очевидно, что диаграмма Вейча функций возбуждения данного триггера должна быть построена так, чтобы ее значения на всех пере-

ходах вызывали требуемые из таблицы переходов автомата изменения состояний ЭА.

Отсюда следующее **правило**: *аёàãðàìà Àáé÷à функции возбуждения может быть получена для конкретного триггера по каждому из его информационных входов q_k^j (j -ый вход k -ого триггера) путем замены нулей и единиц в диаграмме Вейча функции внешних переходов значениями сигналов возбуждения $b_{k\alpha\beta}^j$ ($\alpha\beta$ – все возможные переходы) из матрицы переходов этого триггера.*

Таблица 32

$Q_k^n \rightarrow Q_k^{n+1}$	D	T	RS		JK		DV	
			R	S	J	K	D	V
$b_{k00} (0 \rightarrow 0)$	0	0	b_1	0	0	b_1	b_1	b_1^*
$b_{k01} (0 \rightarrow 1)$	1	1	0	1	1	b_2	1	1
$b_{k10} (1 \rightarrow 0)$	0	1	1	0	b_3	1	0	1
$b_{k11} (1 \rightarrow 1)$	1	0	0	b_2	b_4	0	b_2	b_2^*

Матрицы переходов рассмотренных триггеров сведены в табл. 32.

3.6.3.2. Аналитический метод определения функций возбуждения триггеров

Этот метод отражает в аналитической форме преобразование функций внешних переходов в функции возбуждения, из предыдущего раздела. Деление ДНФ диаграммы Вейча на две части эквивалентно представлению в виде

$$Q_k^{n+1} = \overline{Q_k^n} g_1^n \vee Q_k^n g_2^n, \quad (3.4)$$

где $\overline{Q_k^n} g_1^n$ – логическая функция верхней части диаграммы,

$Q_k^n g_2^n$ – логическая функция нижней части диаграммы.

Поскольку для дизъюнктивных диаграмм значения функции набирается по единицам (конституентам единицы), то нули диаграммы формируют отрицание функции. Отсюда объединение по нулям верхней части таблицы будет $\overline{Q_k^n} g_1^n$, а нижней – $Q_k^n g_2^n$. При формировании функции возбуждения q_k^j преобразованию подвергаются как единицы, так и нули функции внешних переходов, поэтому дизъюнктивная форма q_k^j собирается по единицам всех клеток диаграммы Вейча

$$q_k^j = b_{k00}^j \overline{Q_k} \overline{g_1} \vee b_{k01}^j \overline{Q_k} g_1 \vee b_{k10}^j Q_k \overline{g_2} \vee b_{k11}^j Q_k g_2, \quad (3.5)$$

здесь $\overline{Q_k} \overline{g_1}$ – нули верхней части таблицы, переход из $0 \rightarrow 0$,

$\overline{Q_k} g_1$ – единицы верхней части таблицы, переход из $0 \rightarrow 1$,

$Q_k \overline{g_2}$ – нули нижней части таблицы, переход из $1 \rightarrow 0$,

$Q_k g_2$ – единицы нижней части таблицы, переход из 1 → 1.

Правило определения аналитической формы функции возбуждения:

1. Функция переходов каждого ЭА записывается в виде (3.4).
2. По матрице переходов применяемого триггера составляется выражение по форме (3.5).

3. Производится минимизация полученного выражения. Если в функцию возбуждения входят неопределенные коэффициенты, то минимизация не полностью определенных выражений осуществляется по известным правилам, но обязательно с учетом допустимых значений неопределенных коэффициентов, связанных соотношениями, например, $b_1 b_1^* \neq 1$ для DV -триггера.

Описанная процедура дает следующие выражения для функций возбуждения триггеров, рассмотренных в табл. 32

$$\begin{aligned}
 D\text{-триггер:} & \quad D_k = Q_k; \\
 T\text{-триггер:} & \quad T_k = \overline{Q}_k g_1 \vee Q_k g_2; \\
 RS\text{-триггер} & \quad R_k = b_1 \overline{Q}_k \overline{g}_1 \vee Q_k \overline{g}_2; \\
 & \quad S_k = \overline{Q}_k g_1 \vee b_2 Q_k g_2; \\
 JK\text{-триггер} & \quad J_k = \overline{Q}_k g_1 \vee b_3 Q_k \overline{g}_2 \vee b_4 Q_k g_2; \\
 & \quad K_k = b_1 \overline{Q}_k \overline{g}_1 \vee b_2 \overline{Q}_k g_1 \vee Q_k \overline{g}_2; \\
 DV\text{-триггер} & \quad D_k = b_1 \overline{Q}_k \overline{g}_1 \vee \overline{Q}_k g_1 \vee b_2 Q_k g_2; \\
 & \quad V_k = b_1^* \overline{Q}_k \overline{g}_1 \vee \overline{Q}_k g_1 \vee Q_k \overline{g}_2 \vee b_2^* Q_k g_2.
 \end{aligned}$$

В простейшем случае значения всех неопределенных коэффициентов можно принять равными нулю, что не гарантирует минимальной формы функции возбуждения для произвольной функции внешних переходов.

3.6.3.3. Определение функций возбуждения триггеров методом сравнения

Функцию внешних переходов ЭА представим в виде

$$Q_k^{n+1} = (a_1 \vee \overline{Q}_k a_2 \vee Q_k a_3)^n \quad (3.6)$$

Если (3.6) совпадает с одним из выражений функций перехода известных нам триггеров, то логические выражения a_1, a_2, a_3 можно рассматривать как соответствующие функции возбуждения. К примеру, если функция переходов приводится к виду $Q_k^{n+1} = (\overline{Q}_k a_2 \vee Q_k a_3)^n$, то мы можем заключить, что целесообразно использовать JK -триггер с функциями возбуждения $J = a_2$ и $K = \overline{a}_3$.

Пример. Синтез D -триггера на базе JK -триггера. Таблица переходов синтезируемого D -триггера в форме диаграммы Вейча и матрица переходов JK -триггера используются для получения функций возбуждения J^n и K^n , как показано на рис.42. Полученные выражения определяют построение комбинационной схемы управления. При этом D -триггер будет тактироваться по тому же перепаду на входе C , как и базовый JK -триггер.

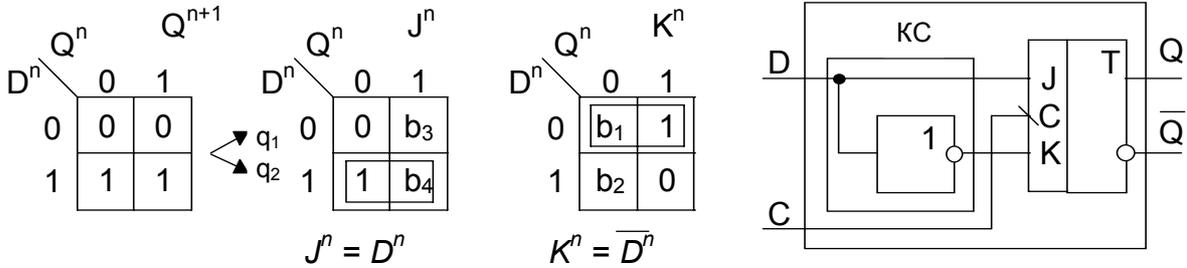


Рис. 42

Пример. Синтез генератора (формирователя) одиночных импульсов (ГОИ), как автомата Мура. ГОИ должен вырабатывать одиночный импульс, равный по длительности периоду тактовых импульсов (ТИ) T_c , по фронту входных импульсов, длительность каждого из которых больше T_c .

Как следует из временных диаграмм на рис. 43,а, выходной сигнал y_1 вызывает входная последовательность x_0x_1 , на основании чего построен граф переходов автомата (рис.43,б), где жирной линией показан “правильный” путь, приводящий к формированию сигнала y_1 .

Для кодирования состояния автомата необходимо 2 триггера ($\log_2 3$), причем комбинация $\{Q_1Q_0\}=\{10\}$ в работе автомата отсутствует. С учетом $S_0=01$, $S_1=00$, $S_2=11$, $S_0=**$, получена таблица переходов ГОИ (табл. 33) и кодированная таблица переходов (табл. 34)

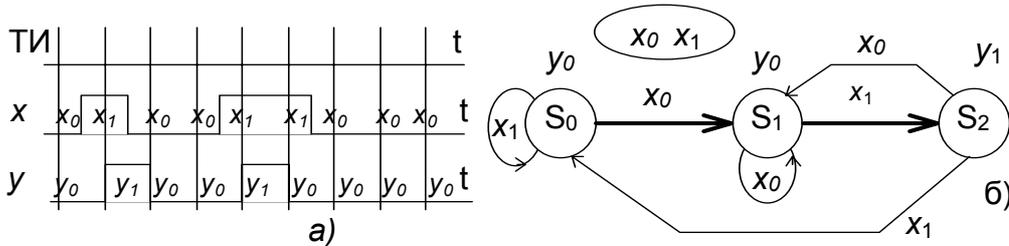


Рис. 43

Таблица 33

	y	y ₀	y ₀	y ₁	*
x \ S	S ₁	S ₁	S ₀	S ₂	S ₃
x ₀	S ₁	S ₁	S ₁	S ₁	*
x ₁	S ₂	S ₀	S ₀	S ₀	*

Таблица 34

y	0	0	1	*
Триггер	Q ₁ Q ₀	Q ₁ Q ₀	Q ₁ Q ₀	*
Код сост.	00	01	11	10
x \ S	S ₁	S ₀	S ₂	S ₃
0	00	00	00	*
1	11	01	01	*

Из кодированной таблицы переходов получаем две таблицы переходов для Q_1

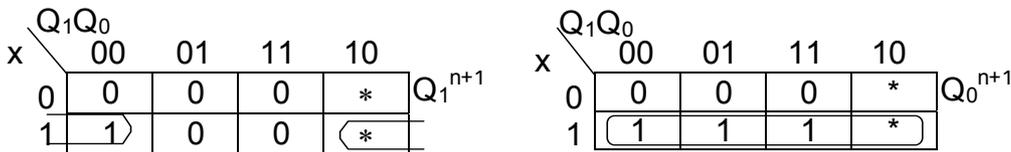


Рис. 44

и Q_0 в форме диаграмм Вейча (рис. 44), откуда $Q_1^{n+1} = (x\bar{Q}_0)^n$ и $Q_0^{n+1} = x^n$.

Из кодированной таблицы переходов (табл. 34) можно отметить, что сигнал y повторяет состояние тригера Q_1 , т.е. $y^n = Q_1^n$.

Реализуем ГОИ на JK -триггерах. Найдем функции возбуждения табличным способом. Для Q_1 получим J_1^n и K_1^n (рис. 45,а), а для $Q_0 - J_0^n, K_0^n$ (рис. 45,б)

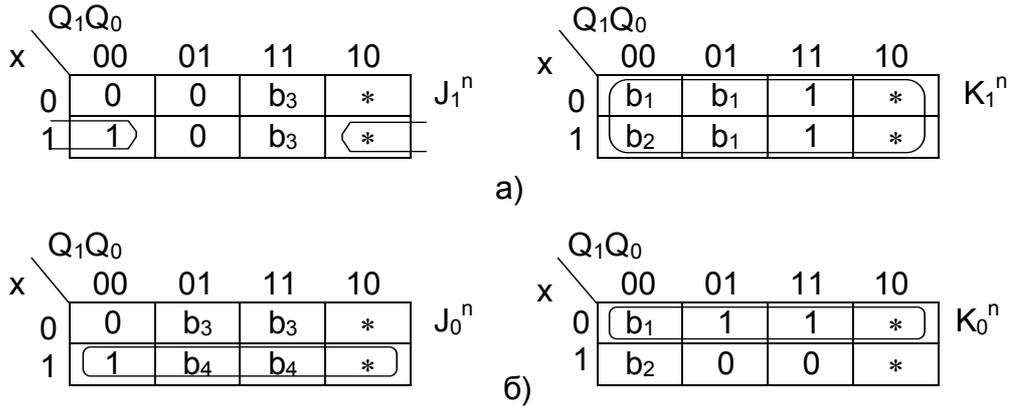


Рис. 45

В результате минимизации получим $J_1^n = x \overline{Q_0}$, $K_1^n = 1$, $J_0^n = x$, $K_0^n = \overline{x}$.

В соответствии с полученными функциями возбуждения $J_1^n, K_1^n, J_0^n, K_0^n$ построим схему ГОИ и временную диаграмму его работы (рис. 46, а, б).

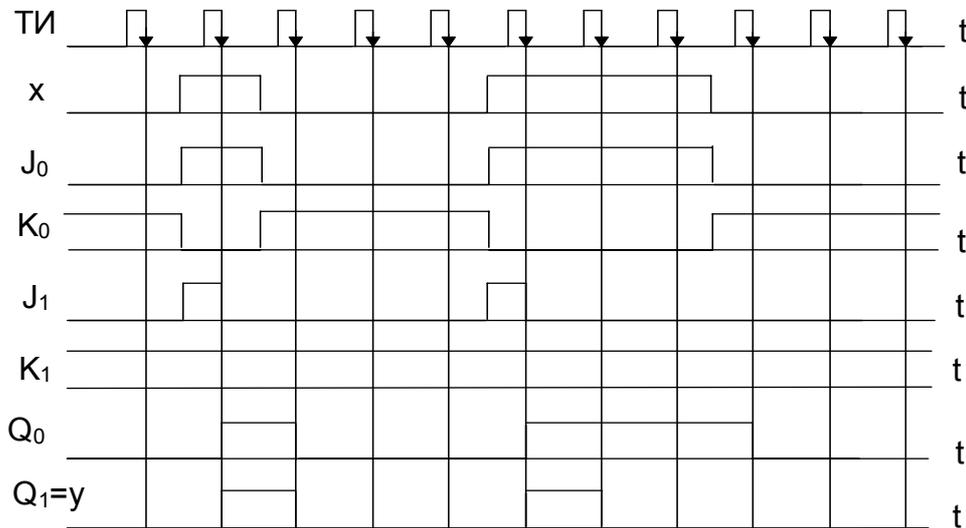
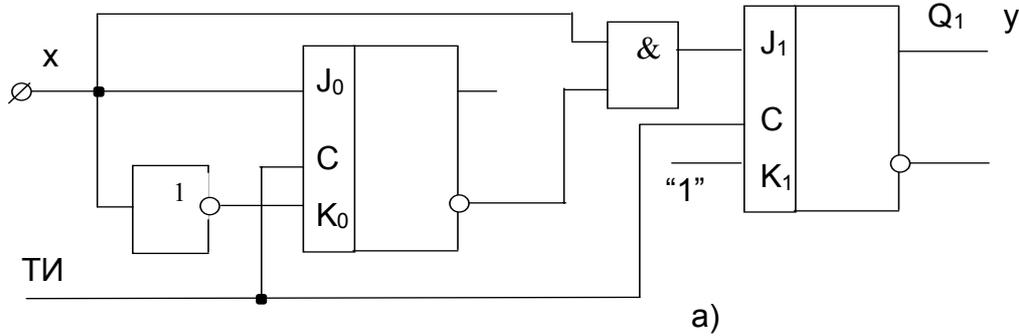


Рис. 46

3.7. Типовые последовательные цифровые устройства

3.7.1 Регистры

3.7.1.1. Классификация регистров и особенности их схемотехнических решений

Регистром называется последовательное цифровое устройство, используемое для хранения и выполнения логических преобразований над n -разрядным двоичным словом. Регистр представляет собой упорядоченную совокупность триггеров, число которых соответствует числу разрядов в слове, и вспомогательных логических элементов, с помощью которых осуществляется выполнение логических *микроопераций* (МО) над двоичными словами в ПЦУ.

В регистре могут выполняться следующие МО: ввод-вывод информации в параллельном и/или последовательном коде; сдвиг слова вправо или влево на заданное число разрядов; преобразование числа из параллельного кода в последовательный и обратно; установка регистра в исходное состояние; поразрядные логические операции.

Регистр памяти. Регистр с параллельным вводом-выводом информации называется *регистром памяти*. Прием информации в регистр может быть однофазным с использованием D -триггеров или парафазным с помощью RS -триггеров (рис. 47). Сброс регистра в нулевое состояние производится по установочным входам сигналом "Уст. 0".

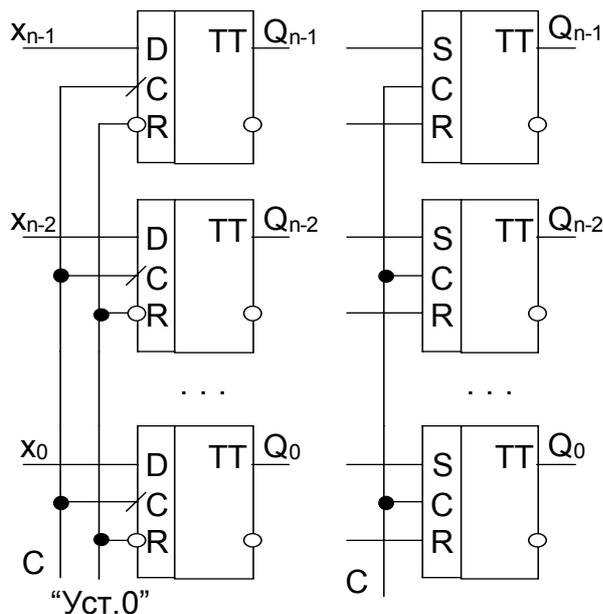


Рис. 47

однофазным с использованием D -триггеров или парафазным с помощью RS -триггеров (рис. 47). Сброс регистра в нулевое состояние производится по установочным входам сигналом "Уст. 0". В случае парафазной записи числа в регистр $RS \neq 1$. Можно обойтись без дополнительного сброса в 0, если записать в регистр ноль по информационным входам (К155ТМ5, К155ТМ7).

Регистры сдвига. Регистры с последовательным вводом-выводом информации получили название *регистров сдвига*. Слово может сдвигаться вправо, влево, в обоих направлениях (реверсивный сдвиг) и в кольцо (циклический сдвиг).

Наиболее употребимы в регистрах сдвига D -триггеры (рис. 48), где ввод информации и сдвиг по сигналам синхронизации C осуществляется в последовательном коде по входу x , а вывод может быть как в последовательном коде (Q_0), так и в параллельном — с выходов триггеров. Пунктиром показана

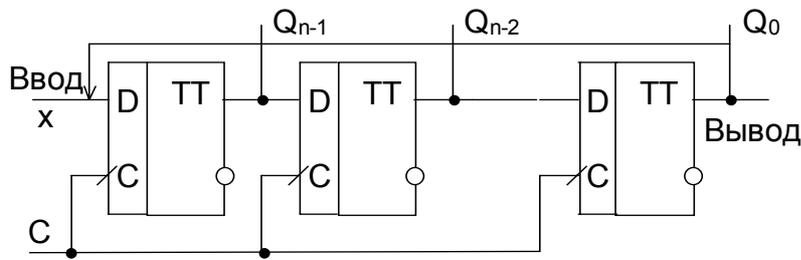


Рис. 48

связь для реализации циклического сдвига, для чего следует предусмотреть независимый от сдвига ввод слова в регистр.

Универсальные регистры. В универсальных

регистрах сочетаются функции регистров сдвига и регистров памяти: ввод-вывод информации в параллельном и/или последовательном коде, реверсивный сдвиг, в том числе циклический. На основе универсальных могут быть реализованы схемы любых регистров [1] (ИМС К155ИР1, К155ИР13).

3.7.1.2. Применение регистров для преобразования информации

Регистры сдвига используются для **умножения и деления**: сдвиг числа влево или вправо на один разряд эквивалентен умножению или делению числа на основание системы счисления. Последовательный сдвиг может быть использован для задержки на n тактов.

Регистры памяти используются для **реализации микрооперации перезаписи числа $A := B$** . Передача числа производится в однофазном или парафазном параллельном коде (рис. 47). **Микрооперация инвертирования числа $A := \bar{A}$** выполняется при использовании в регистре счетных триггеров, на информационные входы которых подается число A .

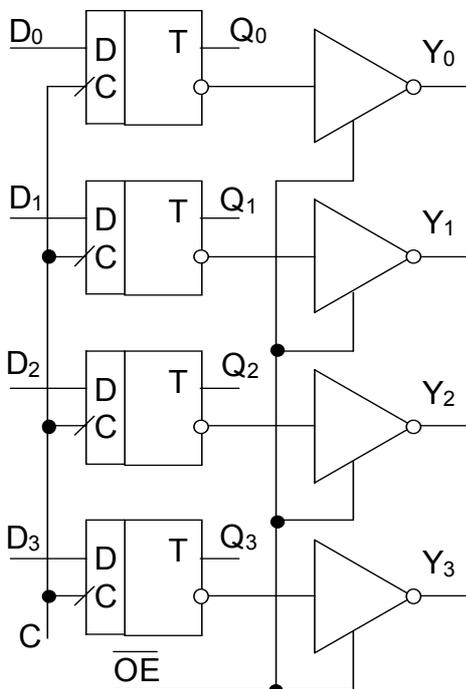


Рис. 49

Для двунаправленной передачи информации по шинам микропроцессорных вычислительных устройств используются регистры с тристабильными выходами. Схема регистра К1804ИР1 представлена на рис. 49, где $Q_3 - Q_0$ - выходы триггеров с TTL-уровнями сигналов, $Y_3 - Y_0$ - тристабильные выходы буферных инверторов. Сигнал управления $\overline{OE} = 1$ переводит выходы буферов в высокоимпедансное состояние, отключая регистр от внешних устройств по выходам $Y_3 - Y_0$. Наличие выходов с тремя состояниями позволяет использовать такие схемы для подключения к общей шине, где в каждый момент времени может быть только один источник и один приемник информации (КР580ВА86, КР580ИР82/83, КР1810ИР82, К589ИР12, К555ИР22, К1533ИР33).

Пример. Накапливающий сумматор. Это сумматор, содержащий в своем составе регистр, в который перед началом микрооперации (МО) сложения хранится слагаемое (первый операнд), а после выполнения МО сумма $S:=S+Y+C_1$, где Y —второй операнд, C_1 —входной перенос. Структурная схема накапливающего сумматора показана на рис. 50.

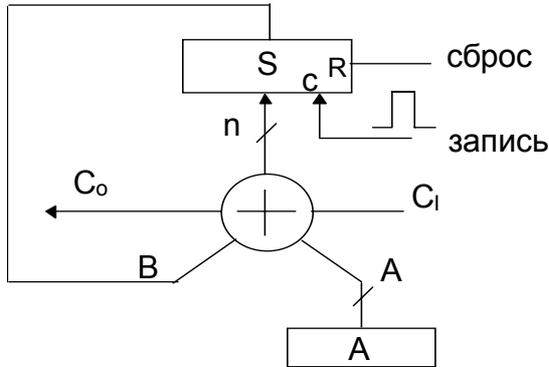


Рис. 50

Алгоритм сложения двух операндов выглядит следующим образом :

- вызов в A первого операнда;
- сброс S в ноль (обнуление частичной суммы);
- микрооперация $S:=S+A$ при $C_1=0$ (загрузка первого операнда в S);
- вызов в A второго операнда;
- микрооперация $S:=S+A+C_1$, после чего результат сложения операндов находится в S.

3.7.2. Счетчики

3.7.2.1. Общие сведения

Счетчики - это ПЦУ, обеспечивающие хранение слова информации C и выполнения над ним микроопераций счета, т.е. изменения его содержимого на единицу. В зависимости от реализуемой микрооперации различают суммирующие $(C+1)$, вычитающие $(C-1)$ и реверсивные $(C\pm 1)$ счетчики. Функционирование счетчика по модулю N можно представить графом (рис. 51).

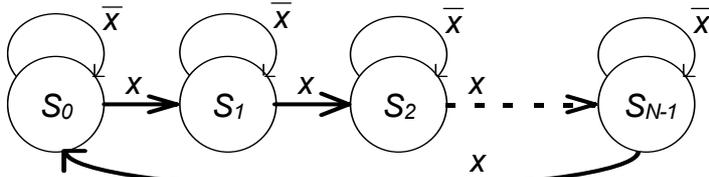


Рис. 51

графом (рис. 51).

Микрооперация счета выполняется по сигналу x . В случае отсутствия этого сигнала счетчик остается в текущем состоянии. Модуль счета счетчика K_c определяется количеством N его внутренних состояний. Двоичный счетчик, содержащий n разрядов (триггеров), имеет $N = 2^n$ состояний, т.е. $K_c = 2^n$, а емкость его равна $K_c - 1$, т.е. $2^n - 1$. По способу организации переноса между разрядами различают счетчики с *последовательным*, *параллельным* и *комбинированным* переносом. Различают *синхронные* и *асинхронные* счетчики в зависимости от способа организации счета. Синхронные счетчики имеют общий для всех разрядов источник тактирующих сигналов.

3.7.2.2. Суммирующие двоичные счетчики

В суммирующем счетчике, содержащем n триггеров, каждому состоянию соответствует число в счетной последовательности от 0 до $2^n - 1$, отображаемое двоичной комбинацией на выходах триггеров – $Q_{n-1}Q_{n-2} \dots Q_1Q_0$.

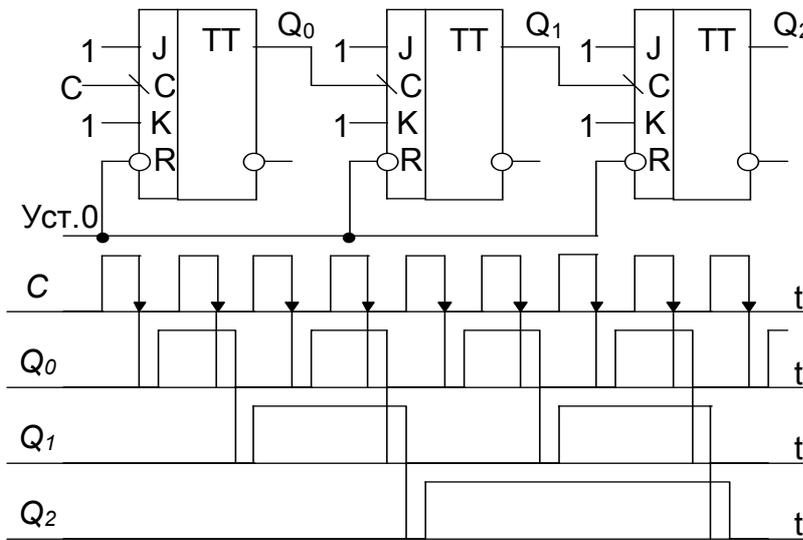


Рис. 52

В простейшем случае суммирующий счетчик может быть построен последовательным соединением любых триггеров, включенных по счетной схеме, т. е. изменяющих свое состояние на инверсное на каждый тактовый импульс. Схема

трехразрядного асинхронного суммирующего счетчика на JK -триггерах и временные диаграммы его работы приведены на рис. 52. Данный счетчик построен на последовательно соединенных JK -триггерах ($J=K=1$). В такой схеме с *последовательным переносом*, вследствие суммирования задержек срабатывания триггеров, новое состояние счетчика устанавливается в худшем случае с задержкой nt_d , где t_d - время переключения триггера. При этом в счетчике появляются на короткое время состояния, не соответствующие счетной последовательности,

что может вызвать срабатывание соединенных с ним схем. Для повышения

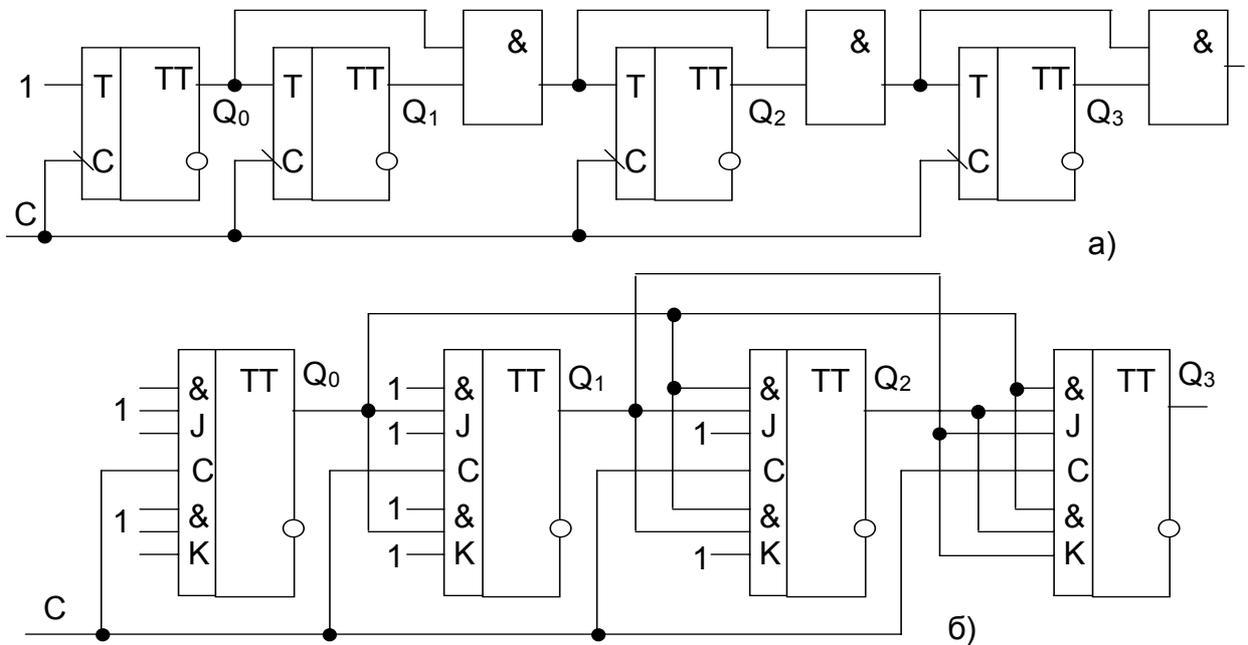


Рис. 53

что может вызвать срабатывание соединенных с ним схем. Для повышения

быстродействия используют синхронизацию триггеров, а также *сквозной* или *параллельный* перенос по информативным входам (рис. 53).

Входные импульсы, число которых надо подсчитать, подаются на синхровходы триггеров. В счетчиках со сквозным переносом (рис. 53,а) процесс переноса также является последовательным, что ограничивает максимальное значение входной частоты. При параллельном переносе задержка определяется временем срабатывания только одной схемы И (внутренний логический элемент триггера), а с учетом того, что триггеры тактируются от одного источника сигналов и срабатывают практически одновременно, параллельный перенос (рис. 53,б) обеспечивает максимальное быстродействие.

3.7.2.3. Вычитающие и реверсивные двоичные счетчики

В вычитающих счетчиках на каждый счетный сигнал предыдущий результат уменьшается на единицу, т. е. содержимое счетчика изменяется от исходного значения $2^n - 1$ до 0, после чего вновь становится равным $N_{max} = 2^n - 1$. Триггер каждого последующего разряда переходит в другое состояние после сигнала заема, обратном сигналу переноса в суммирующих счетчиках. Для построения вычитающих счетчиков необходимо в схемах на рис. 52 и 53 использовать сигналы с инверсных выходов триггеров \bar{Q}_i вместо прямых выходов Q_i , а исходное состояние задавать сигналом по входам “Уст. 1” (\bar{S}).

В реверсивных счетчиках объединяются схемы суммирующего и вычитающего счетчиков с возможностью управления направлением счета (сложения или вычитания). Реверсивный двоичный счетчик с последовательным переносом (заемом) показан на рис. 54.

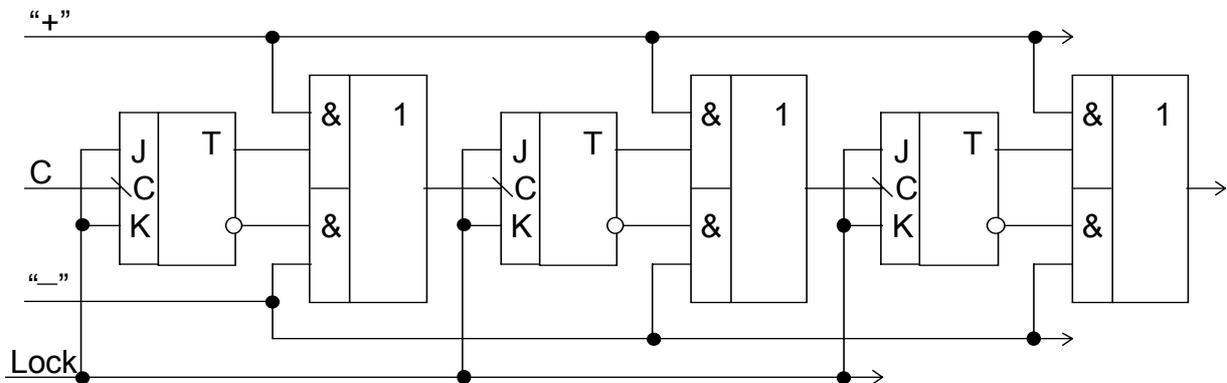


Рис. 54

Направление счета задается логической единицей по входу “+”(сложения) либо “-”(вычитание). Сигнал логического нуля по входу “Lock” блокирует возможные ложные срабатывания *JK*-триггеров счетчика в момент изменения направления счета или при записи информации по установочным входам триггеров.

3.7.2.4. Двоично-десятичные синхронные счетчики

Двоично-десятичные счетчики представляют практически важный случай построения счетчиков по произвольному модулю $2^n \leq K_c \leq 2^{n+1}$ для $K_c=10$. Десятичный счетчик может быть построен на базе 4-разрядного двоичного

Т а б л и ц а 35

Состояние счетчика	Двоично-десятичные эквиваленты $Q_3Q_2Q_1Q_0$ в различных системах кодирования				
	8-4-2-1	2-4-2-1	4-2-2-1	5-2-1-1	5-4-2-1
0	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0001
2	0010	0010	0010	0011	0010
3	0011	0011	0011	0101	0011
4	0100	0100	0110	0111	0100
5	0101	0101	0111	1000	1000
6	0110	0110	1010	1001	1001
7	0111	0111	1011	1011	1010
8	1000	1110	1110	1101	1011
9	1001	1111	1111	1111	1100

счетчика, из графа работы которого необходимо исключить 6 состояний из 16, что возможно сделать значительным числом способов в зависимости от принятого способа кодирования состояний (табл. 35).

Согласно графу своих переходов де-

сятичный счетчик последовательно проходит состояния от нулевого до девятого, после чего вновь устанавливается в нулевое. Из табл. 35 следует, что состояние любого из триггеров счетчика в будущий момент времени зависит только от состояний всех триггеров в настоящий момент времени. Входного информационного сигнала в счетчиках нет, а переходы совершаются под воздействием синхросигналов.

Таким образом, для двоично-десятичного счетчика в коде 8-4-2-1, диаграммы Вейча для функций переходов триггеров выглядят, как показано на рис. 55.

Звездочками на диаграммах Вейча помечены шесть исключенных состоя-

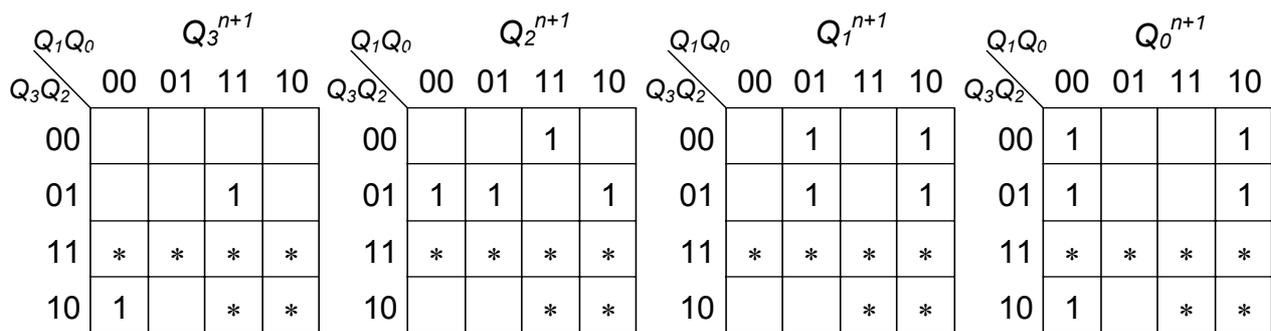


Рис. 55

ний. Используя методику синтеза конечных автоматов, построим десятичный счетчик на базе *JK*-триггеров, для которых диаграммы Вейча функций возбуждения (*JK*-входы) приведены на рис. 56.

		J_3						K_3						J_2						K_2									
		Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0							
Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10
00						00		b_1	b_1	b_1	b_1	00				1		00		b_1	b_1	b_2	b_1	00		b_1	b_1	b_2	b_1
01				1		01		b_1	b_1	b_2	b_1	01		b_4	b_4	b_3	b_4	01				1		01				1	
11		*	*	*	*	11		*	*	*	*	11		*	*	*	*	11		*	*	*	*	11		*	*	*	*
10		b_4	b_3	*	*	10			1	*	*	10				*	*	10		b_1	b_1	*	*	10		b_1	b_1	*	*

		J_1						K_1						J_0						K_0									
		Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0				Q_1Q_0							
Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10	Q_3Q_2		00	01	11	10
00			1	b_3	b_4	00		b_1	b_2	1		00		1	b_3	b_3	1	00		b_2	1	1	b_2	00		b_2	1	1	b_2
01			1	b_3	b_4	01		b_1	b_2	1		01		1	b_3	b_3	1	01		b_2	1	1	b_2	01		b_2	1	1	b_2
11		*	*	*	*	11		*	*	*	*	11		*	*	*	*	11		*	*	*	*	11		*	*	*	*
10				*	*	10		b_1	b_1	*	*	10		1	b_3	*	*	10		b_2	1	*	*	10		b_2	1	*	*

Рис. 56

В результате минимизации получим выражения для информационных входов: $J_3 = Q_2Q_1Q_0$, $K_3 = Q_0$, $J_2 = Q_1Q_0$, $K_2 = Q_1Q_0$, $J_1 = \overline{Q_3}Q_0$, $K_1 = Q_0$, $J_0 = K_0 = 1$. Схема синхронного двоично-десятичного счетчика на JK-триггерах с входной логикой приведена на рис. 57.

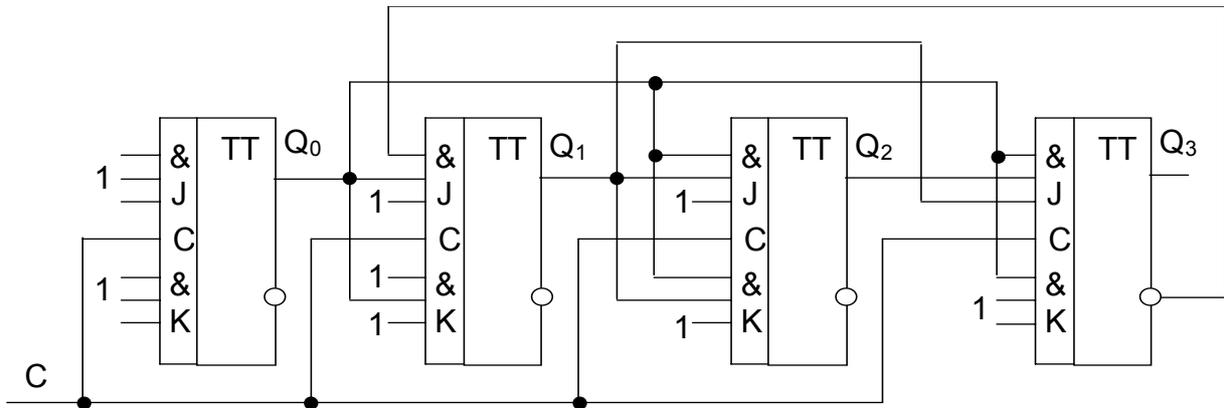


Рис. 57

Десятичный счетчик (рис. 57) работает в двоичном коде 8-4-2-1 до состояния 1001. Обратная связь (сигнал логического нуля) с инверсного выхода $\overline{Q_3}$ блокирует срабатывание, т.е. переход в единичное состояние триггера Q_1 а, вследствие этого, и триггера Q_2 на девятый и десятый синхроимпульсы, так что счетчик приходит в исходное нулевое состояние по окончании десяти импульсов синхронизации.

3.7.2.5. Метод синтеза синхронно-асинхронных десятичных счетчиков

Классический метод синтеза синхронных счетчиков применим для любого кодирования состояний счетчика и дает в общем случае усложненные решения. В инженерной практике употребим метод, который позволяет получить более простые решения за счет сочетания асинхронного и синхронного режимов работы триггеров. Для иллюстрации метода проведем синтез десятичного счетчика в коде 8-4-2-1 на базе *JK*-триггеров.

На первом этапе синтеза для большей наглядности целесообразно построить временные диаграммы работы счетчика, пользуясь кодированием из табл. 35 (рис. 58).

На втором этапе находятся триггеры, работающие в счетном режиме - инверсии. Как следует из рис. 58, в счетном режиме ($J=K=1$) работает триггер Q_0 от сигналов на синхровходе C и триггер Q_2 , если на его вход синхронизации подать Q_1 .

На третьем этапе определяются источники тактовых сигналов для оставшихся триггеров по следующему правилу: множество изменений состояний на выходе данного триггера должно включаться полностью во множество рабочих фронтов источника тактовых сигналов для данного триггера.

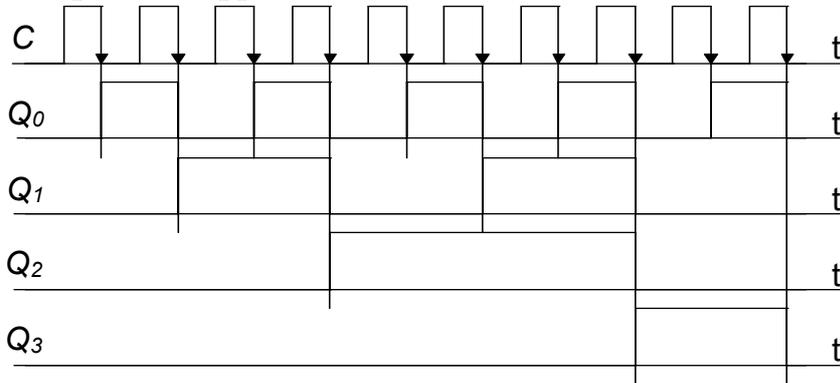


Рис. 58

Можно заметить, что все изменения состояний триггеров Q_1 и Q_3 приходятся на перепады из 1 в 0 (рабочие фронты для *JK*-триггера) на выходе триггера Q_0 . Следовательно, для указанных триггеров в качестве

источника синхросигналов можно взять Q_0 .

На четвертом этапе с помощью диаграмм Вейча определяются функции возбуждения триггеров, работающих не в счетном режиме. В рассматриваемом случае состояния триггеров Q_1 и Q_3 определяются как функция Q_1 , Q_2 , Q_3 и не будут зависеть от Q_0 , поскольку для этих триггеров Q_0 является синхросигналом. Диаграммы Вейча для триггеров Q_1^{n+1} и Q_3^{n+1} , приведенные на рис. 59, отражают переходы триггеров при изменении сигналов Q_0 из 1 в 0.

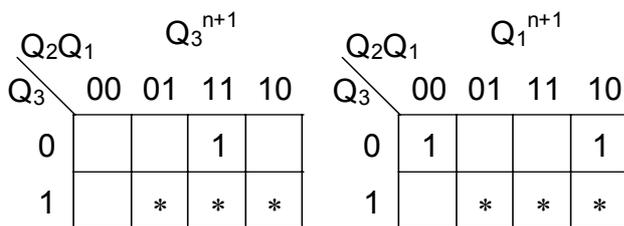


Рис. 59

Следуя вышеизложенной методике синтеза, построим диаграммы Вейча для определения функций возбуждения триггеров Q_1 и Q_3 (рис. 60).

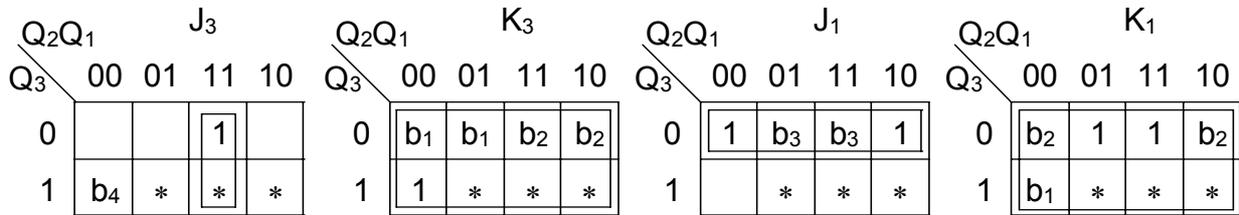


Рис. 60

Получим из приведенных диаграмм выражения для JK -входов: $J_3 = Q_2Q_1$, $K_3 = 1$, $J_1 = \bar{Q}_3$, $K_1 = 1$. Используя полученные выражения, построим схему десятичного счетчика (рис. 61), который содержит меньшее число связей и логических элементов по сравнению с решением, полученным классическим способом (рис. 57).

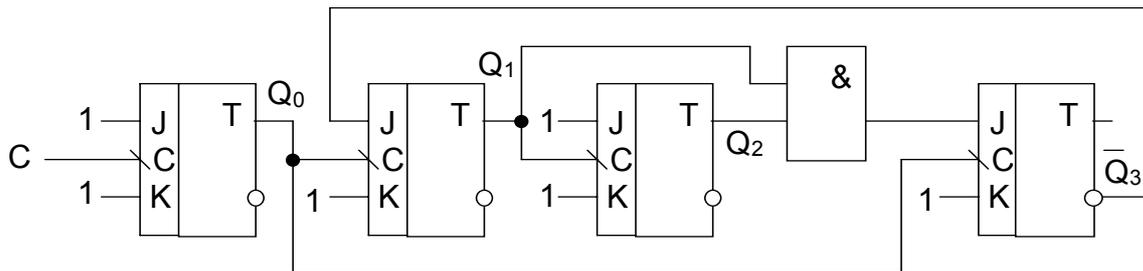


Рис. 61

3.7.3. Цифровые делители частоты

Цифровой делитель частоты (ЦДЧ) – последовательностное цифровое устройство, в котором входная периодическая последовательность импульсов формирует на выходе периодическую последовательность с частотой, в K_δ раз меньшую. Последовательность состояний делителя частоты может быть любой, необходимо лишь обеспечить заданный коэффициент деления частоты K_δ . ЦДЧ могут иметь фиксированный и программируемый коэффициент деления. Практически важным является случай, когда K_δ представляет собой простое число, поскольку любой коэффициент деления может быть разложен на простые сомножители и степени двух (2^k). Делитель частоты с $K_\delta = 2^k$ может быть легко реализован k -разрядным двоичным счетчиком. Реализация ЦДЧ с коэффициентом деления, выраженным простым числом, имеет практические инженерные решения на основе счетчиков по произвольному модулю счета и регистров сдвига.

3.7.3.1. ЦДЧ на основе пересчетных схем

Коэффициент деления, выраженный простым или нечетным числом, может быть записан в виде $K_\delta = 2N + 1$, где N – произвольное, в свою очередь, простое число. В общем случае $K_\delta = 2(2(2 \dots 2(N_0 + 1) + 1) + \dots + 1) + 1$. Здесь $N_0 = 2^j$, т.е. число, являющееся степенью двух. Например, $K_\delta = 17 = 2 * 2^3 + 1$, $K_\delta = 23 = 2(2(2^2 + 1) + 1) + 1$. Функциональная схема ЦДЧ, реализующего формулу

$K_d=2N+1$ с использованием пересчетных схем и временные диаграммы его работы, представлены на рис. 62.

Пересчетная схема с коэффициентом деления $2N$ (на рис. 62 обведена пунктиром) состоит из последовательно соединенных JK -триггера и счетчика

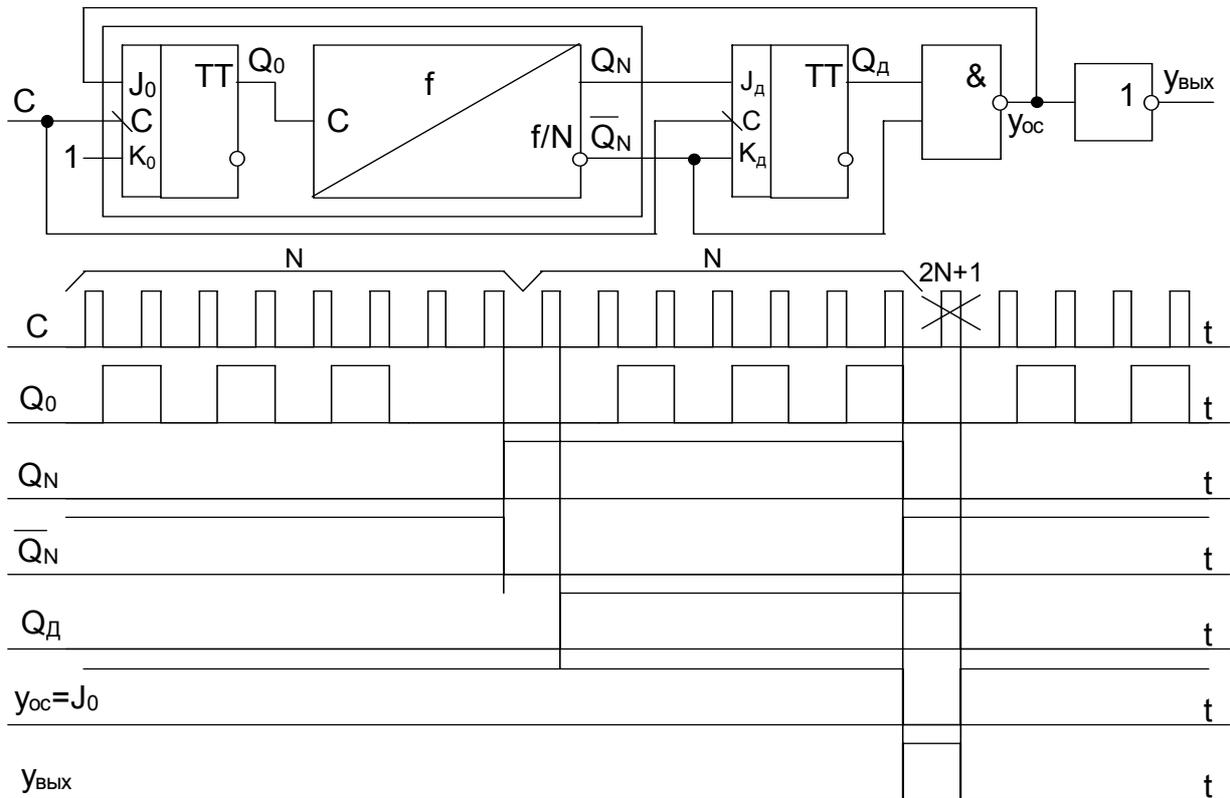


Рис. 62

по модулю N . В течение $2N$ тактов сигнала на входе схемы этот JK -триггер работает в счетном режиме ($J_0=K_0=1$). На выходе счетчика Q_N формируется импульс по окончании $2N$ периодов входного сигнала. С помощью дополнительного триггера Q_d выходной импульс пересчетной схемы по модулю $2N$ задерживается на один период входной последовательности, при этом на выходе логического элемента И-НЕ формируется импульс инверсной полярности длительностью один период сигнала на входе, блокируя срабатывание триггера Q_0 по входу $J_0=y_{oc}=0$ при поступлении $2N+1$ -ого импульса. После перехода триггера Q_d в нулевое состояние цикл работы делителя частоты повторяется. На выходе $y_{вых}$ формируется импульсная последовательность с частотой в $2N+1$ раз меньшей частоты сигнала на входе. Построение ЦДЧ по описанной схеме целесообразно при $K_d>16$, поскольку синтез синхронного счетчика по вышеприведенной методике при числе триггеров свыше четырех вручную достаточно сложен и приводит, как правило, к более сложным решениям.

3.7.3.2. ЦДЧ на основе регистров

Схема ЦДЧ с коэффициентом деления $K_{\delta}=2N+1$ (рис. 62) фактически представляет собой сочетание счетчика по модулю $2N$ и одноразрядного регистра сдвига. Обобщая полученное решение, можно построить ЦДЧ с $K_{\delta}=K_c+K_p$, объединяя последовательно счетчик по модулю K_c и регистр сдвига с числом разрядов K_p . В предельном случае при $K_c=0$ с помощью регистра сдвига, работающего циклически, можно получить ЦДЧ с $K_{\delta}=K_p$. В таком делителе частоты один из разрядов должен устанавливаться в единицу при начальной установке, а остальные - в 0 (рис. 63). Занесенная таким образом в регистр "1" передвигается в кольце триггеров.

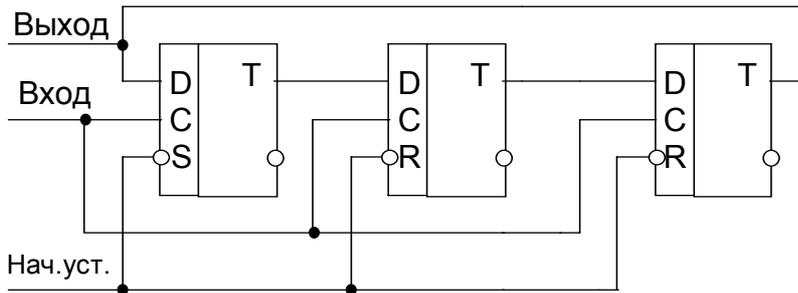


Рис. 63

К недостаткам схемы относится то, что в случае сбоя, т.е. при исчезновении единицы в кольце или появлении нескольких единиц, следует повторно производить начальную установку. Для самовосстановления после сбоя необходимо использовать дополнительные логические элементы [3].

Период цикла ЦДЧ на кольцевом регистре сдвига определяется числом триггеров в регистре и при больших значениях K_{δ} такие схемы требуют значительного числа триггеров, т.е. становятся неэкономичными. Увеличить коэффициент деления при том же числе триггеров в регистре можно, используя перекрестные связи (счетчики Джонсона). Четный коэффициент деления $K_{\delta}=2N$ можно получить в схеме, представленной на рис. 64.

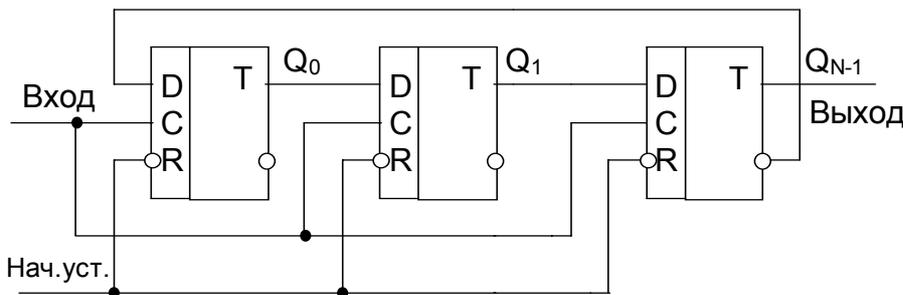


Рис. 64

В исходном состоянии все триггеры регистра установлены в нулевое состояние. В течение N тактов входной импульсной последовательности триггеры регистра поочередно, начиная с Q_0 , заполняются единицами, а в течение следующих N тактов в той же очередности триггеры заполняются нулями, после чего процесс повторяется (табл. 36).

В делителях частоты с перекрестными связями могут быть использованы любые триггеры, включенные по схеме D-триггера. Достоинством таких счетчиков является простота декодирования внутренних состояний – требуются

Таблица 36

Номер такта	Состояние триггеров					
	Q_0	Q_1	Q_2	...	Q_{N-2}	Q_{N-1}
0	0	0	0	...	0	0
1	1	0	0	...	0	0
2	1	1	0	...	0	0
3	1	1	1	...	0	0
...
N-1	1	1	1	...	1	0
N	1	1	1	...	1	1
N+1	0	1	1	...	1	1
N+2	0	0	1	...	1	1
...
2N-2	0	0	0	...	1	1
2N-1	0	0	0	...	0	1
2N	0	0	0	...	0	0

только двухвходовые ЛЭ И-НЕ. Так, например, $\overline{Q_{k+1}}Q_k = 1$ только при поступлении $k+1$ тактовых импульсов. Кроме того, в каждый момент изменяется состояние только одного триггера, что исключает ложные сигналы на выходе дешифратора.

На рис. 65 приведена схема счетчика Джонсона, позволяющая получить нечетный коэффициент деления $K_0 = 2N - 1$. Уменьшение числа состояний на 1 в цикле модифицированного счетчика Джонсона возникает за

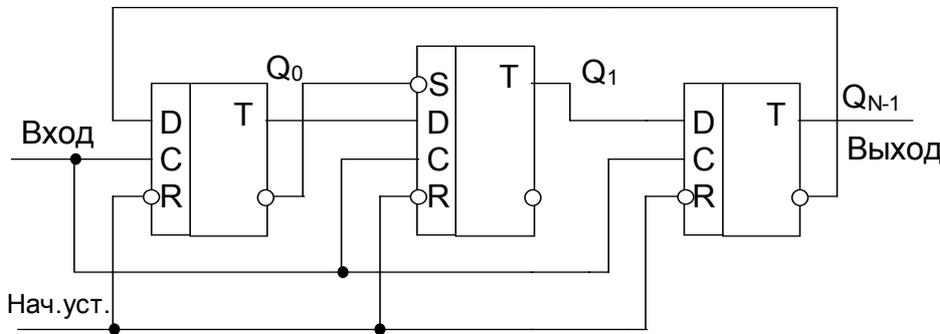


Рис. 65

счет асинхронной установки в единичное состояние триггера Q_1 на первый же тактовый импульс. После этого цикл работы счетчика соответствует табл. 36.

3.8. Контрольные вопросы и задания для самоподготовки .

1. Дать определение конечного цифрового автомата. В чем его отличие от комбинационной схемы.
2. В чем отличие автоматного времени от реального дискретного?
3. Назвать и пояснить требования к абстрактным конечным автоматам.
4. Дать определения автоматам Мили и Мура и представить их таблицами истинности и графами.
5. Пояснить на примерах матричное представление абстрактного конечного автомата (АКА).
6. При каких условиях два АКА эквивалентны?
7. Определить алгоритм перехода от АКА Мили к АКА Мура и обратно.
8. Объяснить сущность метода минимизации АКА.
9. Сформулировать задачи структурного синтеза автоматов.
10. Нарисовать структурную схему реального конечного автомата.
11. Пояснить сущность синхронных и асинхронных автоматов.
12. Провести анализ синхронного RS-триггера на элементах И-НЕ (ИЛИ-НЕ). Найти функцию переходов, построить граф состояний и временные диаграммы сигналов, определить отличие триггеров, реализованных на элементах И-НЕ и ИЛИ-НЕ. Определить характеристики быстродействия синхронных триггеров на логических элементах.
13. Провести анализ синхронных JK-, D-, DL-, T- триггеров. Определить функции переходов, построить графы состояний и временные диаграммы сигналов.
14. Пояснить принцип работы универсального JK/RS- триггера и возможные режимы его работы.
15. Провести анализ асинхронного RS-триггера на элементах И-НЕ(ИЛИ-НЕ). Найти функцию переходов, построить граф состояний и временные диаграммы сигналов. Выявить отличия схем триггеров на элементах И-НЕ и ИЛИ-НЕ. Определить характеристики быстродействия асинхронных триггеров на логических элементах.
16. Пояснить методы определения функций возбуждения триггеров.
17. Дать определение счетчика как цифрового автомата.
18. Провести классификацию счетчиков. Объяснить принцип организации переноса в разрядах.
19. Определить параметры двоичного n-разрядного счетчика при $n=2,3,4,\dots,10$.
20. Нарисовать схему и провести анализ асинхронного счетчика на JK(D)-триггерах при $n=2,3,4,\dots,10$.
21. Оценить разрешающее время и быстродействие асинхронных n- разрядных счетчиков при $n=2,3,4,\dots,10$.

22. Провести синтез n - разрядного синхронного счетчика с параллельных переносом. Пояснить эффект повышения быстродействия по сравнению с асинхронным режимом.
 23. Объяснить принцип организации суммирования и вычитания в счетчиках. Нарисовать схему реверсивного счетчика.
 24. Пояснить работу реверсивного счетчика в режиме вычитания (сложения) из (от) числа 5,6,7,8,9,10,11,12.
 25. Пояснить принцип двоично-десятичного счета в кодах 8-4-2-1, 2-4-2-1, 4-2-2-1.
 26. Пояснить принцип цифрового деления частоты, дать определение параметрам ЦДЧ и провести их классификацию.
 27. Пояснить, на конкретных примерах, принципы изменения коэффициента цифрового деления частоты.
 28. Предложить реализацию ЦДЧ с коэффициентом деления $K_d = 5, 7, 10, 13, 15, 17, 21, 25, 27, 30$. Нарисовать схему и поясняющие временные диаграммы сигналов.
 29. Пояснить принцип ЦДЧ на кольцевых сдвигающих регистрах и счетчиках Джонсона. Нарисовать схему и поясняющие временные диаграммы сигналов.
 30. Привести схему и пояснить режимы работы n - разрядного регистра памяти ($n=3, 4, 5, 6, 7, 8$).
- Привести схему и пояснить режимы работы n - разрядного регистра сдвига ($n=4, 8, 16$).

Глава 4. Элементы синтеза цифровых операционных устройств

4.1. Понятие операционного устройства.

Общая структура операционного устройства

Операционное устройство (ОУ) является важнейшей частью любой аппаратуры для обработки дискретной информации. Именно в ОУ реализуется основная функция цифровых вычислительных устройств – операционное преобразование информации. На основе общего принципа программного управления вычислительными устройствами эта обработка осуществляется путем выполнения на каждом шаге реализуемого алгоритма некоторой операции из фиксированного набора – системы операций или команд данного вычислительного устройства.

В состав выполняемых ОУ операций могут входить операции, различающиеся содержанием (сложение, вычитание, умножение и др.), типом операндов (операнды фиксированной и плавающей точкой, десятичные операнды, операнды с различным числом разрядов и др.), направлением перемещений результатов операций, типом признаков результата (больше - меньше нуля, равно нулю, переполнения и др.).

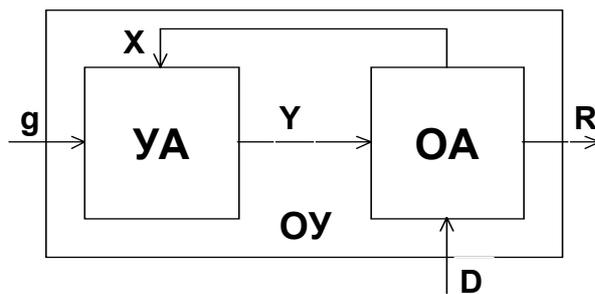


Рис. 66

В настоящее время классической является структура представления операционного устройства в виде композиции *операционного (ОА) и управляющего (УА) автоматов*, как показано на рис. 66, где g – код операции, D – входная оперативная информация, Y – управляющие сигналы, X – осведомительные сигналы, R – результат операции.

ции.

Функцией операционного устройства является выполнение заданного множества операций $F = \{f_1, f_2, f_3, \dots, f_g, \dots, f_G\}$ над входными словами $D = \{d_1, d_2, \dots, d_N\}$ с целью вычисления слов $R = \{r_1, r_2, \dots, r_G\}$, представляющих результаты операций $R = f_g\{D\}$, $g = 1, 2, \dots, G$.

Процесс функционирования во времени операционного устройства состоит из последовательности тактовых интервалов, в которых **операционный автомат производит элементарные (микро) операции преобразования слов под программным воздействием сигналов, поступающих из управляющего автомата.**

4.2. Принцип микропрограммного управления

Операционный автомат состоит из регистров, сумматоров и других узлов, производящих прием из внешней среды и хранения кодов слов, их преобра-

зование под воздействием сигналов Y управляющего автомата, выдачу во внешнюю среду результата преобразования R , а также выдачу в управляющий автомат и внешнюю среду осведомительных сигналов $X = \{x_1, x_2, \dots, x_L\}$ о знаках и особых значениях операндов, их отдельных разрядов, их отдельных разрядов, например, равенства нулю и т. д.

ОА выполняет некоторый набор элементарных преобразований информации, например, таких, как передача слова из одного регистра в другой, взятие обратного кода, сдвиг и др.

Выполнение этих элементарных операций инициируется поступлением в ОА из УА соответствующих управляющих сигналов $Y = \{y_1, y_2, \dots, y_M\}$.

Элементарная функциональная операция, выполняемая за один тактовый интервал и приводимая в действие одним управляющим сигналом y_i называется **микрооперацией** (МО).

Пусть $Y = \{y_1, y_2, \dots, y_M\}$ – множество микроопераций, реализуемое в ОА. Они возбуждаются сигналами y_1, y_2, \dots, y_M из УА, причем, если должна быть выполнена МО y_i , то $y_i = 1$, если МО y_i не выполняется, то $y_i = 0$ ($i = 1, 2, \dots, M$).

Последовательность управляющих сигналов определяется кодом операции g и сигналами X . В этой связи эти сигналы (g и X) относятся к одному классу – к классу осведомительных сигналов (логических условий), т.к. по отношению к УА они играют одинаковую роль: они влияют на порядок выработки управляющих сигналов.

Совокупность МО, выполняемых одновременно за один такт автоматного времени, образует **микрокоманду** (МК). В частности, микрокоманда может состоять из одной микрооперации.

Последовательность микрокоманд, обеспечивающих выполнение данной операции, называется **микропрограммой** данной операции. Функционирование операционного устройства может быть описано совокупностью реализуемых в нем микропрограмм.

Следует отметить, что время не является аргументом функции операционного автомата, для которого устанавливается список действий (микроопераций) Y автомата и вырабатываемые при этих действиях логические условия X , но никак не определяется порядок следования этих действий во времени, и, соответственно, появление логических условий. Управляющий автомат вырабатывает распределенную во времени последовательность управляющих сигналов $y_{t_1}, y_{t_2}, \dots, y_{t_k}$, порождающих в ОА нужную последовательность микроопераций.

4.3. Язык описания операционных устройств

Микропрограмма, представляющая функцию операционного устройства безотносительно к средствам, которые могут быть использованы для реализа-

ции заданной функции, называется *функциональной микропрограммой*. Для записи функциональных микропрограмм используется язык *функционального микропрограммирования*.

Функциональная микропрограмма состоит из двух частей:

- ⇒ описание слов и массивов, с которыми оперирует функциональная микропрограмма;
- ⇒ содержательный граф микропрограммы, который содержит описание микроопераций и логических условий.

4.3.1. Описание слов и массивов

Слова, которыми оперирует микропрограмма, делятся на входные (*I*), внутренние (*L*) и выходные (*O*), где в формате слова

$$IdentW(n_2:n_1),$$

IdentW – идентификатор или имя, символический указатель слова,

n_2 – номер старшего двоичного разряда,

n_1 – номер младшего двоичного разряда.

Массив слов описывается в виде $IdentM[m_2:m_1](n_2:n_1)$,

где *IdentM* – идентификатор массива слов,

m_2 – номер старшего слова массива,

m_1 – номер младшего слова массива.

Пример. $PROM[1023:0](7:0)$ – массив из 1024 восьмиразрядных слов (K573PФ1),

$RAM[4095:0](15:0)$ – массив из 4096 шестнадцатиразрядных слов,

$PaK(31:16)$ – слово, разрядностью 16 бит (содержимое регистра PгК).

У слов можно выделять *поля*. Действия в микропрограмме могут производиться как над целыми словами, так и над частями слов – полями.

Пример. Если $R(15:0)$ – слово в 16 бит, тогда имеем

$D(7:0) = R(15:8)$ – восьмиразрядное поле слова R,

$K(3:0) = R(3:0)$ – четырехразрядное поле слова R,

$F = R(5)$ – одноразрядное поле слова R.

Если скобки у разрядов опущены, то именуется все слово целиком. Для обращения к словам, входящим в массив, указываются их номера (адреса).

Пример. $PROM[18](3:0)$ – четыре младших разряда слова с номером 18.

Слова и массивы слов физически могут представлять собой содержимое как отдельных регистров и их наборов, так и ячеек памяти и их совокупности, что показано в приведенных выше примерах.

4.3.2. Двоичные выражения

Двоичное выражение определяет правило вычисления двоичного значения путем выполнения операций над первичными двоичными значениями, в качестве которых используются константы, значения слов, элементов массивов и полей.

Константы представляют собой числа, выраженные в одной из систем счисления (двоичной, восьмеричной, шестнадцатеричной, десятичной).

Объединение в двоичное выражение осуществляется с помощью двоичных операций:

$\bar{}$ – операция инверсии, $\bar{}A$ – поразрядная инверсия, “НЕ-А”;

\cdot – конкатенация (составление) слов и полей ($A=101$, $B=110$, $A.B=101110$);

\wedge , \vee – конъюнкция (поразрядная) и дизъюнкция;

\oplus – сумма по модулю 2;

$-$, $+$ – арифметическое сложение и вычитание кодов.

Пример двоичных выражений: $C+A(15:1)$; $B+11.\bar{}A(15:0)+1$.

4.3.3. Микрооперации и их классификация

В основе всех микроопераций лежит *МО присваивания*, посредством которой слову присваивается значение двоичного выражения

$$A:=B,$$

где B – двоичное выражение.

При присваивании слова выравниваются по правой границе.

Пример. Если $A=1011$, $B=101$, тогда после *МО* $A:=B$, $A=0101$, а после *МО* $B:=A$, $B=011$.

Классы микроопераций

- *МО* установки (инициализации): $B:=A7h$, $A:=11011001b$, $C(7:0):=137Q$;
- *МО* передачи кода: $C:=11.A(15:0).0$;
- *МО* логические поразрядные ($\bar{}$, \wedge , \vee , \oplus);
- *МО* счета: $A:=A+C$, где C – константа, в том числе $C=1$ ($A:=A+1$);
- *МО* алгебраического сложения: $A:=A\pm B$;
- *МО* сдвига, используются для выделения отдельных битов слов или переносу запятой вправо/влево на n позиций.

Различают *МО* сдвига: арифметические, логические и циклические.

Сдвиг слова A влево на k двоичных позиций: $Lk(A).d$,

Сдвиг слова A вправо на k двоичных позиций: $d.Rk(A)$,

где d – двоичное поле (k бит), заполняющее позиции слова A , освобождающиеся при сдвиге.

Пример. Циклический сдвиг влево на 1 бит: $A:=L1(A).A(n-1)$,

где $A(n-1:0)$ – n -разрядное слово.

Сдвиг вправо на 2 бита (с потерей двух младших разрядов): $R2(A)$.

Сдвиг влево на 1 бит (с потерей одного старшего разряда): $L1(A)$.

Комбинированная *МО*: $A:=L1(\bar{}).1$ – сдвиг влево на 1 бит с одновременной инверсией слова A .

Микрооперации, которые могут выполняться одновременно (параллельно), называют совместимыми. Различают функциональную и структурную совместимость. Функциональная совместимость может быть ограничена

структурой операционного устройства, допускающей или исключающей возможность параллельного выполнения нескольких операций (структурная совместимость).

4.3.4. Вычисление логических условий

Для задания логических условий (ЛУ) или осведомительных сигналов можно использовать любые логические выражения, дающие однобитовый результат, а также выражения в форме логических соотношений

$\langle \text{Логич. выражение} \rangle [\text{операция соотношения}] \langle \text{Логич. выражение} \rangle$,

где в качестве операций соотношения могут быть взяты EQ (=), NE (\neq), GT ($>$), LT ($<$), GE (\geq), LE (\leq).

Пример. Выражение, дающее однобитовый результат: $c = p(n-1) \oplus p(n-2)$.

Логическое соотношение $A=0$ истинно, т.е. дает ЛУ=1, при $A=0$.

Логическое соотношение $A=0$ ложно, т.е. дает ЛУ=0, при $A \neq 0$.

4.3.5. Содержательный граф микропрограммы

Для описания микропрограмм в инженерной практике используется *содержательный граф микропрограммы* (граф-схема алгоритма).

Это ориентированный связный граф, включающий вершины четырех типов: начальную, конечную, операторную и условную, как показано на рис. 67.

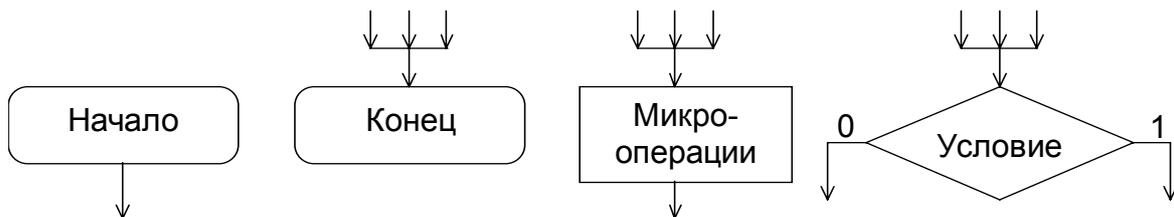


Рис. 67

Конечная, операторная и условная вершины имеют по одному входу; начальная вершина входов не имеет. У начальной и операторной вершины по одному выходу; у условной – два выхода, помеченные символами 0 и 1. Конечная вершина выходов не имеет.

Микрооперации, принадлежащие к одному операторному блоку, должны быть функционально совместимыми.

Содержательный граф микропрограммы удовлетворяет следующим условиям:

- ♦ входы и выходы вершин соединяются друг с другом с помощью дуг, направленных всегда от выхода ко входу;
- ♦ каждый выход соединен только с одним входом;
- ♦ любой вход соединяется по крайней мере с одним выходом;
- ♦ любая вершина графа лежит по крайней мере на одном пути из начальной вершины к конечной;

- ♦ в каждой условной вершине записывается один из элементов множества логических условий $X=\{x_1, x_2, \dots, x_L\}$;
- ♦ один из выходов условной вершины может соединяться с ее входом, что недопустимо для операторной вершины;
- ♦ в каждой операторной вершине записывается оператор y_i из множества МО $Y=\{y_1, y_2, \dots, y_M\}$.

Пример. Функциональная микропрограмма умножения двух целых чисел без знака.

Задача реализации любого процесса автоматического преобразования информации сводится к разработке функциональной микропрограммы, т.е. алгоритма преобразования. Разработка алгоритма представляет собой наиболее творческую часть задачи автоматизации процесса обработки информации и управления.

Рассмотрим пример построения функциональной микропрограммы умножения двух целых чисел без знака

$$C := A \times B,$$

где C – произведение (результат);

A – множимое;

B – множитель.

Т а б л и ц а 37

Формат слова, ЛУ	Назначение и тип слова, содержание ЛУ
$A(n-1:0)$	Множимое; IL
$B(n-1:0)$	Множитель, младший байт результата; LO
$C(n:0)$	Старший байт результата; LO
$Cч(int \log_2 n-1:0)$	Счетчик циклов; LO
$B(0) / x_1$	Младший бит множителя, если $B(0)=1$, то $x_1=1$, иначе $x_1=0$
$Cч=n / x_2$	Содержимое счетчика циклов, если $Cч=n$, то $x_2=1$, иначе $x_2=0$

Описание слов и логических условий показано в таб.37.

Умножение реализуется столбиком, со сдвигом частичной суммы (C - B) вправо. Как показано на рис. 68, на каждом цикле проверяется значение младшего бита множителя $B(0)$. Если $B(0)=1$, то к частичной сумме C добавляется значение множимого. В случае $B(0)=0$ сложение не производится. Особенностью алгоритма является использование только одного дополнительного регистра (ячейки памяти) C разрядностью $n+1$ бит для получения результата умножения разрядностью $2n$ бит. Содержимое множителя B в ходе умножения сдвигается вправо и теряется, а на освобождающиеся слева позиции записываются поочередно младшие биты частичной суммы $C(0)$. Результат умножения представляет собой конкатенацию $R=C.B$ по окончании n циклов умножения.

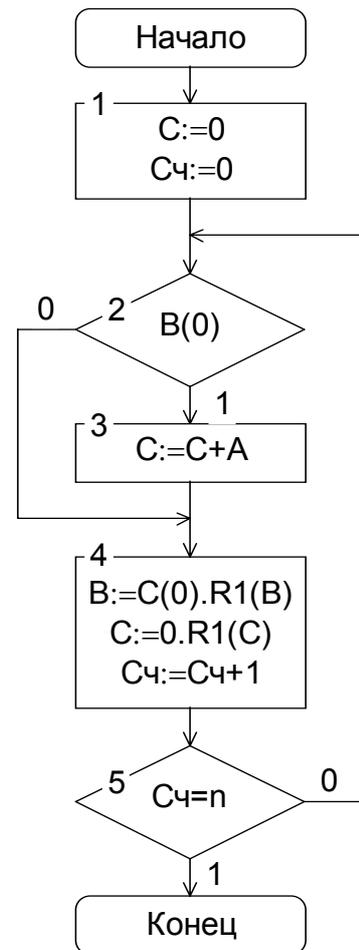


Рис. 68

4.4. Структурный базис операционных устройств

Структурный базис – это набор элементов, на основе которых могут строиться структуры с определенными свойствами. Наиболее широко используются

- ⇒ *шины*, обеспечивающие передачу слов;
- ⇒ *регистры*, обеспечивающие хранение слов;
- ⇒ *комбинационные устройства*, посредством которых вычисляются значения функций.

4.4.1. Шины

Совокупность линий (цепей), предназначенных для передачи слов или, в более общем случае, кодов и сигналов, *объединенных общим функциональ-*

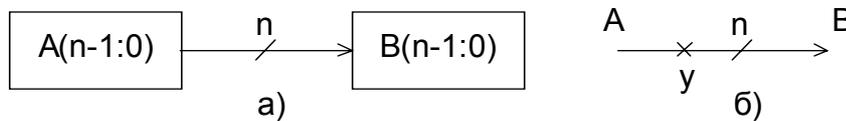


Рис. 69

ном назначением, называется шиной (рис. 69, а,б).

Шина может быть управляемой (бистабильной). На рис. 69, б управляющий сигнал y имеет смысл $B:=A$. В качестве элементов управления целесообразно использовать схемы “И”, по числу линий в шине. Стремление уменьшить число контактов микропроцессорных БИС привело к появлению двунаправленных тристабильных шин. Появилась возможность подключения всех устройств к одной, общей шине и управления направлением передачи данных от источника к приемнику (рис. 11,12 гл. 2).

Совокупность управляемых шин может иметь общий выход при использовании *мультиплексора*. Это более общее назначение мультиплексора по сравнению с использованием, данным в 2.3.2.3. На рис. 70 показано условное обозначение шинного мультиплексора. Чтобы избежать конфликта на шине, в каждый данный момент времени только один управляющий сигнал y_j ($j=1,2,\dots,k$) активен, т.е. разрешает $B:=A_j$.

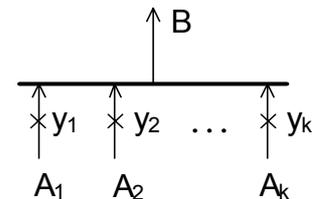


Рис. 70

4.4.2. Регистры

В соответствии с данным в 3.7.1.1 определением регистром называется совокупность элементов памяти, используемая для хранения и выполнения логических преобразований над n -разрядным двоичным словом.

В качестве запоминающих элементов в регистрах операционных устройств обычно используют D и DV -триггеры. Как следует из рис. 71, в регистре R возможен доступ как ко всему слову в целом, так и к отдельным полям слова, а также к отдельным битам.

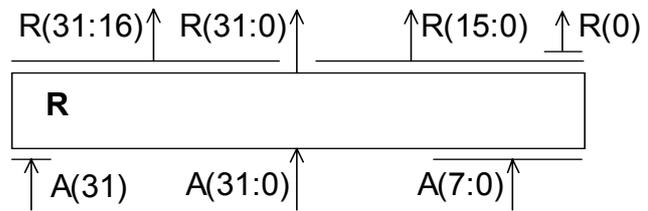


Рис. 70

4.4.3. Комбинационные устройства

В операционных устройствах используются дешифраторы, шифраторы, сумматоры, арифметическо-логические устройства (АЛУ), которые достаточно полно описаны в 2.3.3.

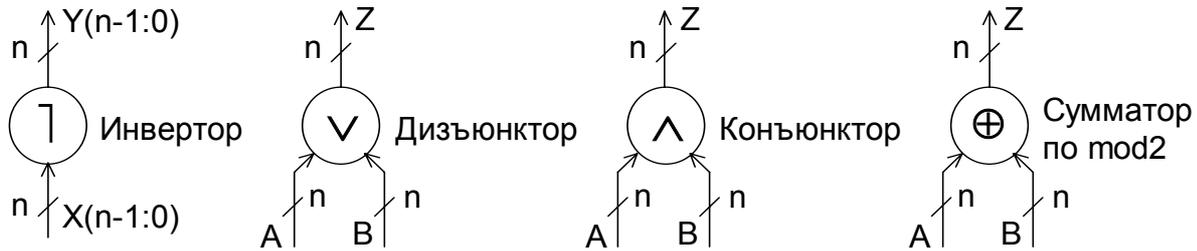


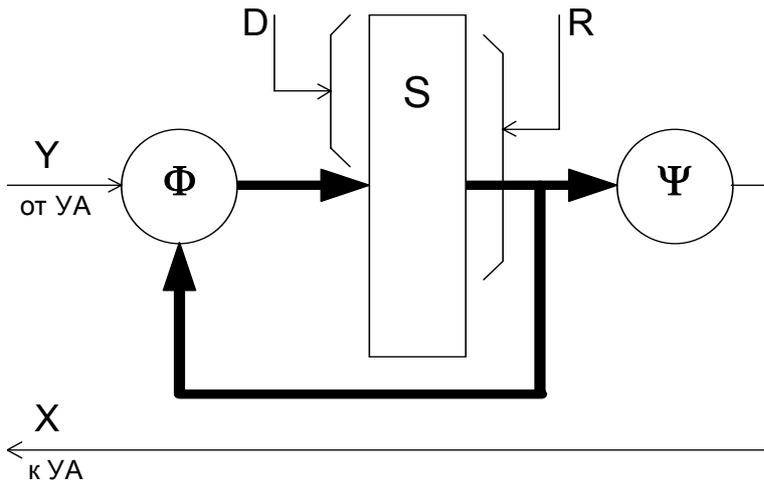
Рис. 71

Кроме этих схем, для поразрядных логических преобразований, например, для вычисления ЛУ, используются инверторы, дизъюнкторы, конъюнкторы и сумматоры по модулю 2 (рис. 71).

4.5. Построение операционного автомата

4.5.1. Каноническое представление ОА

Операционный автомат может быть представлен совокупностью трех ос-



новных блоков: памяти S , комбинационной схемы Φ , реализующей микрооперации, и комбинационной схемы Ψ , вычисляющей значения логических условий (рис. 72).

Функции, реализуемые ОА, определяются описанием множеств слов (D и R), микроопераций Y и логических условий X .

Рис. 72

Память S обеспечивает

хранение слов S_1, S_2, \dots, S_N , которые представляют значения операндов D , промежуточные значения и конечные результаты R . Управляющие сигналы Y , формируемые УА, *инициируют* выполнение необходимых микроопераций. При поступлении сигнала y_i комбинационное цифровое устройство Φ выполнит МО $S_i := \varphi_i\{S_j\}$, что сводится к вычислению $\varphi_i\{S_j\}$ и присваиванию вычисленного значения слову S_i . Для вычисления ЛУ (осведомительных сигналов) служит КЦУ Ψ , реализующее систему булевых функций $X = \{x_l\}, x_l := \psi_l\{S_k\}$.

Пример. Для ОУ умножения двух целых чисел без знака в соответствии с вышеприведенной функциональной микропрограммой МО У представлены в табл. 38. В числе логических

Т а б л и ц а 38

Обозначение	Микрооперация
y_1	$C:=0$
y_2	$Cч:=0$
y_3	$C:=C+A$
y_4	$B:=C(0).R1(B)$
y_5	$C:=0.R1(C)$
y_6	$Cч:=Cч+1$

условий – $x_1\{B(0)\}$ и $x_2\{Cч = n\}$.

На основании описанных функций ОА на рис. 73 представлена его каноническая структура. Выходные цепи регистров, хранящих слова $A, B, C, Cч$ объединены в шину M , по которой сигналы поступают на входы КЦУ Φ и Ψ , реализующих МО и вычисляющих ЛУ.

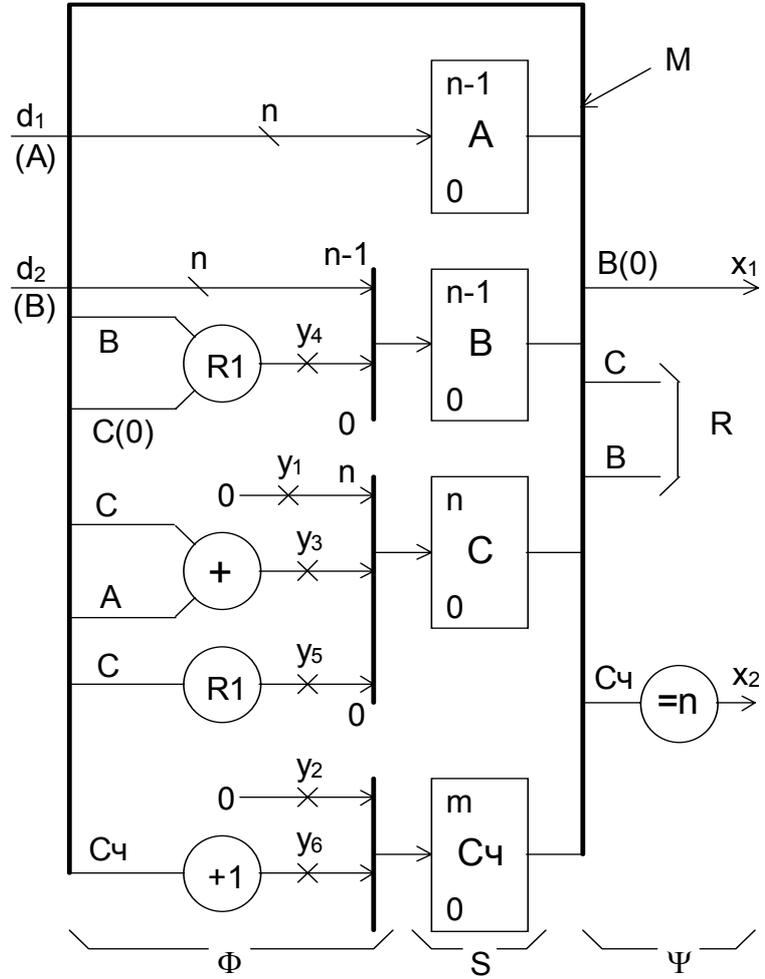


Рис. 73

4.5.2. Структурный синтез ОА

Операционный автомат может быть представлен совокупностью *операционных элементов* (ОЭ), которые выполняют соответствующие микрооперации.

Операционным элементом называется *элементарный ОА*. В общем случае число ОЭ, составляющих операционный автомат, определяется количеством внутренних слов, обрабатываемых микропрограммой. Если в операционном элементе процесс обработки информации носит итерационный, последовательный характер, то их называют *ОЭ накапливающего типа*. Структура операционного элемента накапливающего типа показана на рис. 74. Для синтеза ОЭ наиболее широко используются сдвигающие регистры, сумматоры и счетчики. Сдвигающие регистры обеспечивают сдвиг хранимого в регистре слова на заданное число разрядов. Накапливающие сумматоры (рис. 50) служат для выполнения МО типа $S:=S+A$, где A —операнд, в том числе и константа. Если

$A=\pm 1$, то выполняется МО счета $S:=S\pm 1$, где предыдущий результат используется в качестве операнда в следующей МО.

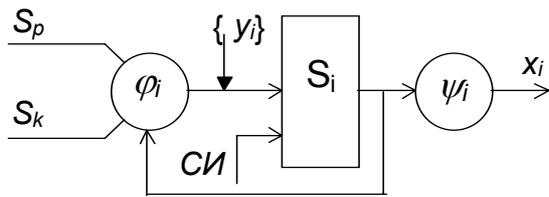


Рис. 74

Обычно ОЭ накапливающего типа строятся на базе синхронных схем. При этом момент переключения элемента памяти (регистра, счетчика) в очередное состояние определяется синхронизирующим импульсом СИ, рабочий фронт которого должен приходиться на стабильное активное состояние сигнала y_i , инициирующего данную микрооперацию.

В качестве элементов памяти чаще всего используются синхронные триггеры JK, RS и DV типов.

Пример. Рассмотрим синтез операционного элемента для внутреннего слова C. В этом ОЭ реализуются три МО: y_1, y_3, y_5 . Микрооперация y_1 – тривиальная, это запись “0” во все ячейки регистра C. Микрооперация y_3 (сложение C и A) реализуется посредством комбинационного сумматора. Для реализации y_5 (сдвиг слова C вправо на один разряд с записью “0” в старший разряд) необходимо, очевидно, чтобы регистр S_i (C) был сдвигающим.

Построим регистр C на DV-триггерах. При выполнении МО y_1 $C:=0$ на все D-входы триггеров регистра подается сигнал логического нуля и одновременно сигнал логической единицы (“разрешение”) на V-входы.

При выполнении МО y_3 $C:=C+A$ на D-входы триггеров подаются результаты поразрядного сложения с S-выходов комбинационного сумматора.

При выполнении МО y_5 $C:=0.R1(C)$ на D-вход старшего разряда подается “0”, а на D-входы последующих разрядов подаются C-выходы предыдущих.

Т а б л и ц а 39

y_i	Микрооперации	D_n	V_n	D_{n-1}	V_{n-1}	D_i	V_i	D_0	V_0
y_1	$C:=0$	0	1	0	1	0	1	0	1
y_3	$C:=A+C$	S_n	1	S_{n-1}	1	S_i	1	S_0	1
y_5	$C:=0.R1(C)$	0	1	C_n	1	C_{i+1}	1	C_1	1

В табл.39 приведены сигналы на DV-входах триггеров регистра в зависимости от реализуемой МО.

Сигналы на выходах комбинационного сумматора описываются следующими выражениями: $S_i = a_i \oplus c_i \oplus q_i$ и $p_i = a_i c_i \vee a_i q_i \vee c_i q_i$,

где q_i – перенос из предыдущего разряда, p_i – перенос в последующий разряд.

Значения функций возбуждения триггеров D_i определяется как

$$\varphi_i = D_i = \bigcup_{m=1}^M y_{mi} U_{mi}$$
, где U_{mi} – значение сигнала возбуждения в строке y_{mi} и столбце i .

Таким образом, из табл. 39 следует

$$\left\{ \begin{array}{l} D_0 = y_3 S_0 \vee y_5 C_1, \\ D_i = y_3 S_i \vee y_5 C_{i+1}, \\ D_n = y_3 S_n, \\ V_0 = V_i = V = y_1 \vee y_3 \vee y_5. \end{array} \right.$$

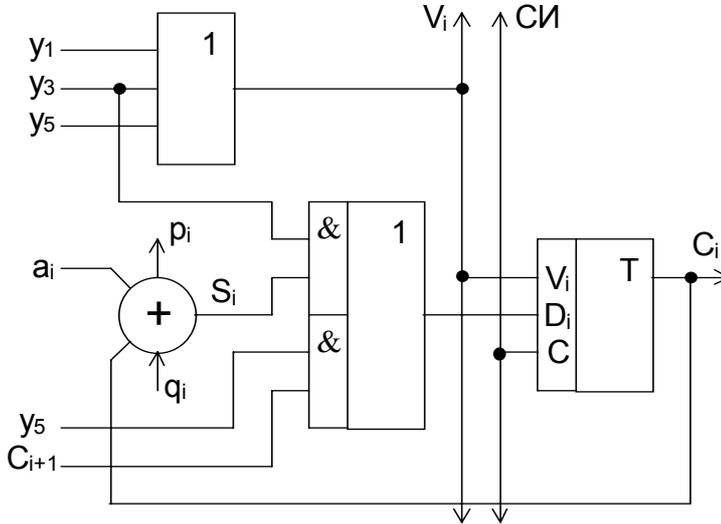


Рис. 75

Схема i -ого разряда операционного элемента приведена на рис. 75.

Подобным образом могут быть синтезированы операционные элементы, реализующие МО y_2, y_4, y_6 .

4.6. Построение управляющего автомата с жесткой и программируемой логикой

Управляющий автомат, который строится по канонической схеме автомата Мили или Мура (рис. 29, 72) называется УА с жесткой логикой, в отличие от автомата с программируемой логикой, который строится на основе хранимой в ПЗУ микропрограммы.

4.6.1. Интерпретация микропрограмм автоматами Мили и Мура

Задача управляющего автомата состоит в преобразовании входной последовательности $X = \{x_1, x_2, \dots, x_L\}$ в выходную управляющую последовательность $Y = \{y_1, y_2, \dots, y_M\}$.

Начало преобразования и одновременно начало работы ОУ задается сигналом g ("Пуск"), который может принимать только два значения (0 и 1), если данное ОУ реализует только одну операцию. Это обстоятельство отражается "ждушей" условной вершиной на граф-схеме функциональной микропрограммы (рис. 76, а).

Необходимый набор состояний автомата определяется путем отметки графа микропрограммы. Рассмотрим переход от графа микропрограммы к графу УА типа Мили (рис. 76, б) и Мура (рис. 76, в) на вышеприведенном примере

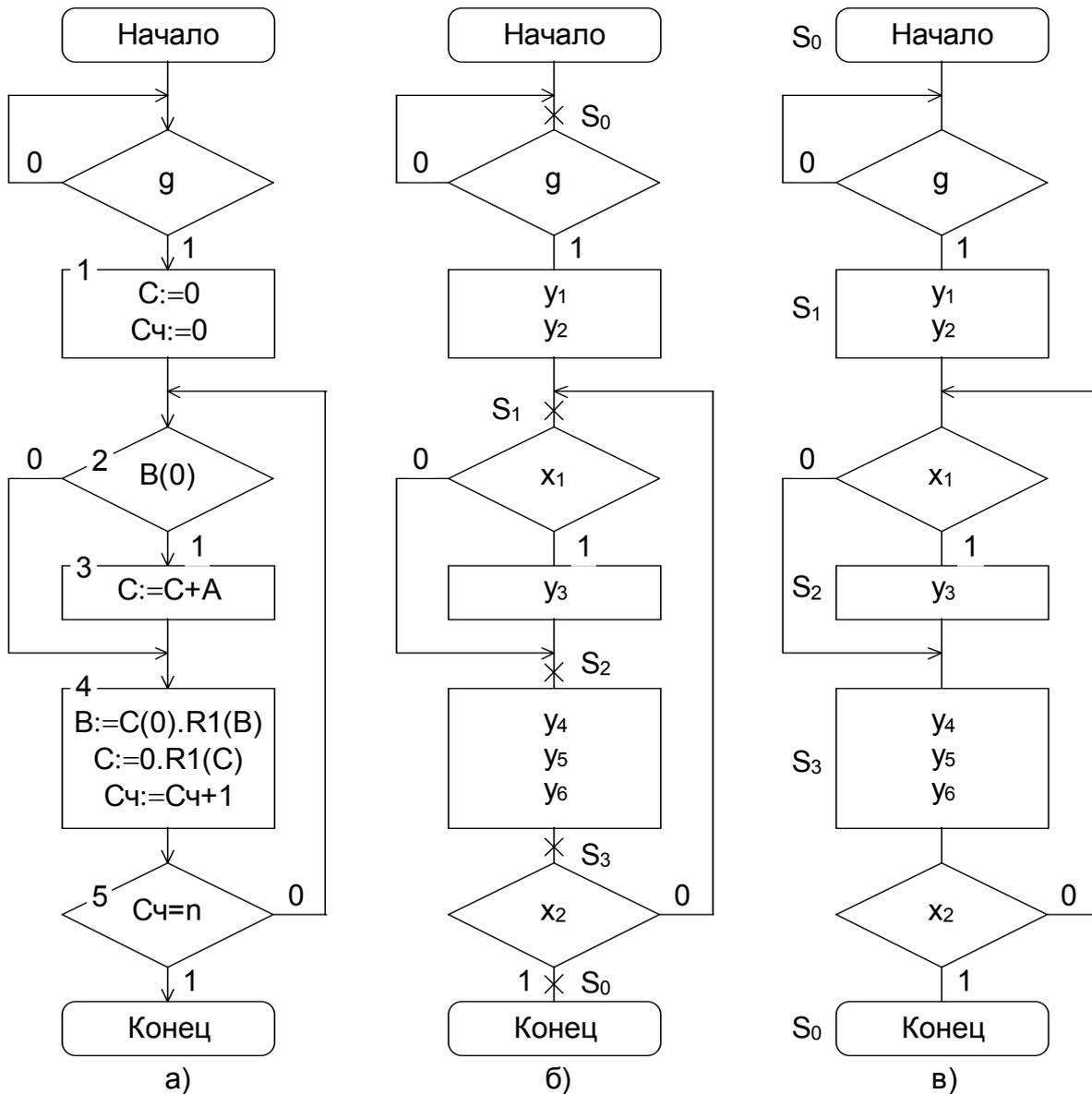


Рис. 76

алгоритма умножения двух целых чисел без знака.

Переход от графа микропрограммы к УА Мили выполняется по следующим правилам:

1) символом S_0 отмечается вход первой вершины, следующей за вершиной “Начало”, а также вход вершины “Конец”, это начальное состояние УА;

2) входы вершин, *следующих за операторными вершинами*, отмечаются символами S_1, S_2, \dots, S_N , при этом каждый вход отмечается единственным символом, который не должен дублироваться, за исключением S_0 .

Переход от графа микропрограммы к УА Мура выполняется по следующим правилам:

1) символом S_0 отмечаются вершины “Начало” и “Конец”, это начальное состояние автомата;

2) каждая операторная вершина отмечается единственным символом S_1, S_2, \dots, S_N , при этом обозначения вершин не должны дублироваться.

Полученные по приведенным правилам микропрограммы УА Мили и Мура (рис. 76, б,в) можно представить в виде графов переходов и выходов, как

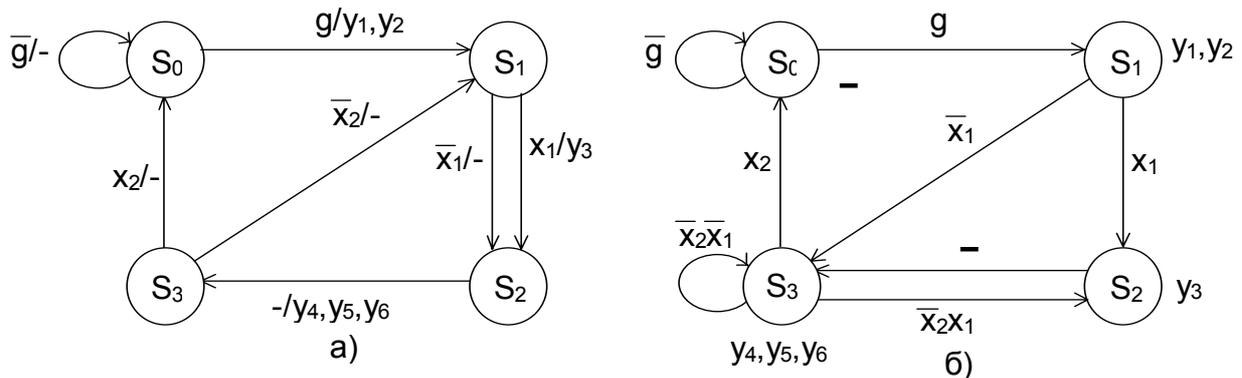


Рис. 76

показано на рис.77, а,б, соответственно.

Дальнейший синтез управляющих автоматов с жесткой логикой производится по методике, изложенной в главе 3.

В общем случае автомат Мура имеет большее число состояний, чем автомат Мили. Обычно использование автомата Мили приводит к экономии оборудования, следовательно, автомат Мили может оказаться предпочтительней.

4.6.2. Построение управляющего автомата с программируемой логикой

Управляющий автомат с программируемой логикой используются при создании ОУ достаточно высокой сложности, решающих, как правило, несколько задач. Синтез УА по приведенным инженерным методикам вручную представляет значительные трудности и не исключает ошибок.

Управляющие сигналы в УА с программируемой логикой генерируются по значениям специальных управляющих кодов в течение одного такта, которые называются микрокомандой (МК). Совокупность микрокоманд образует массив $МК[P:0]$ – микропрограмму. В построении микропрограмм используется *принудительная* или *естественная* адресация МК.

Каждая микрокоманда, относящаяся к классу МК с *принудительной адресацией*, содержит ответы на два вопроса:

⇒ какие управляющие сигналы должны генерироваться в текущем тактереботы операционного устройства, иначе говоря, какие микрооперации должны выполняться в данной МК;

⇒ какая микрокоманда должна выполняться следующей, иначе говоря, каков адрес следующей МК.

В формате микрокоманды с принудительной адресацией (рис. 77) можно выделить две компоненты: *операционную* и *адресную*. Операционная компонента, занимающая *поле операции*, содержит информацию о текущих МО. В адресной части микрокоманды содержится *однозначно интерпретируемая информация* о направлении выполнения микропрограммы в зависимости от

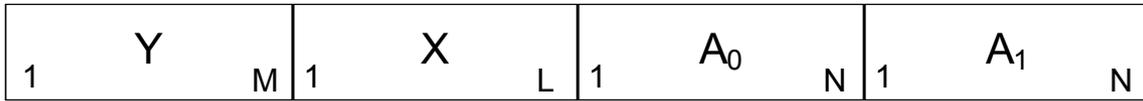


Рис. 77

анализируемых логических условий.

Поле Y включает в себя в простейшем случае столько бит, сколько формируется разных (неэквивалентных) управляющих сигналов. Поле X определяет номер $1, 2, \dots, L$ осведомительного сигнала x_1, x_2, \dots, x_L , значение которого анализируется микрокомандой. Размеры полей Y и X можно сократить, если

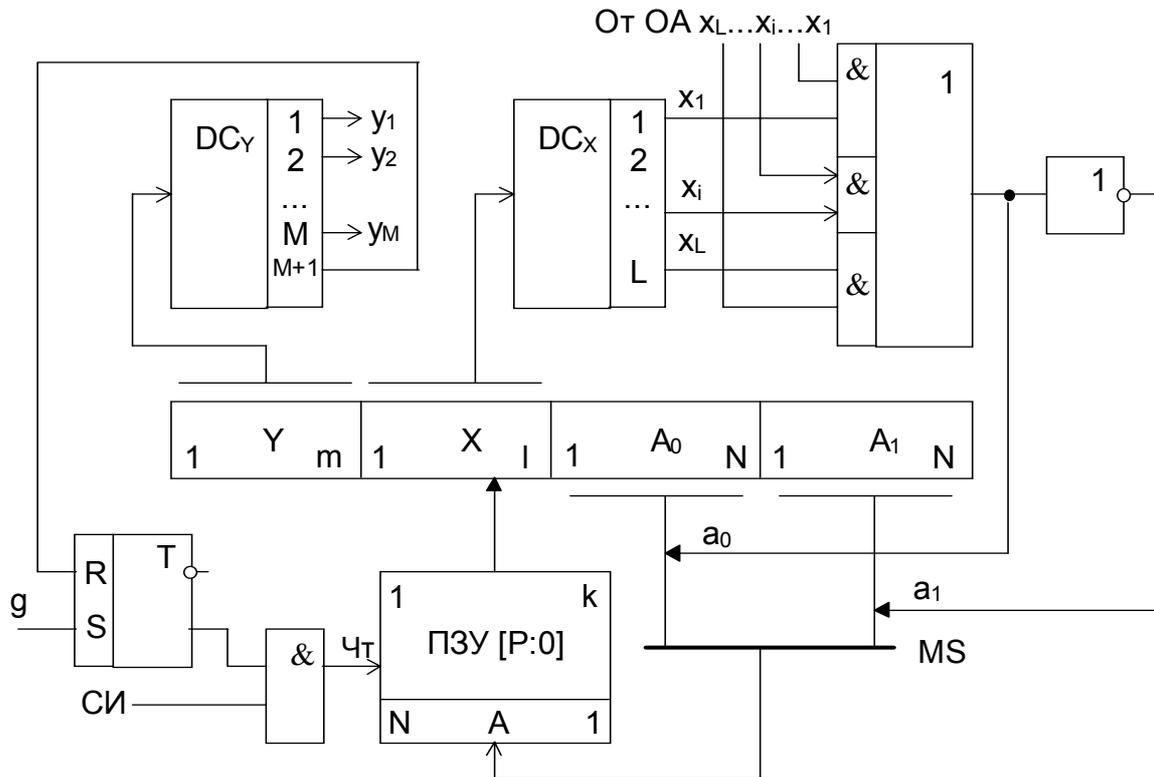


Рис. 78

управляющие сигналы и признаки закодировать (рис. 78).

Микрокоманда с адресным полем N бит адресует 2^N ячеек ПЗУ, в котором хранятся все МК из микропрограммы разрядностью k бит. Если выбранный признак $x_i=1$, то выполняется микрокоманда с адресом A_1 , а если выбранный

признак $x_i=0$, то выполняется МК с адресом A_0 . В случае нулевого значения кода в поле признаков X (отсутствие проверяемых ЛУ), также выполняется МК с адресом A_0 .

В начальном состоянии в регистре микрокоманды содержится “пустая” МК, которая не генерируя никаких управляющих сигналов, передает управление самой себе. Запуск УА осуществляется сигналом g (“Пуск”), а останов – по команде y_{M+1} , которая интерпретируется как дополнительный управляющий сигнал, останавливающий работу УА. Сигналы a_0 и a_1 инициируют передачи $A:=A_0$ или $A:=A_1$. Управляющий сигнал $Чт$ заносит в регистр микрокоманд слово из ПЗУ по адресу A , т.е. $МК:=ПЗУ[A]$. В каждом такте УА приведенной структуры может выполнять не более одной МО $Y=\{y_1, y_2, \dots, y_M\}$, дешифрируемой DC_Y .

Кроме принудительной адресации МК, используется *естественная* адресация команд, при этом уменьшается разрядность МК и объем ПЗУ. При естественной адресации все МК в зависимости от признака z разделяются на два



Рис. 79

типа: *операционные* ($z=0$) и *управляющие* ($z=1$)(рис. 79).

Управляющая МК содержит адресное поле D , по которому передается управление в случае изменения естественного (в порядке возрастания адресов) следования команд, что зависит от анализируемых признаков X .

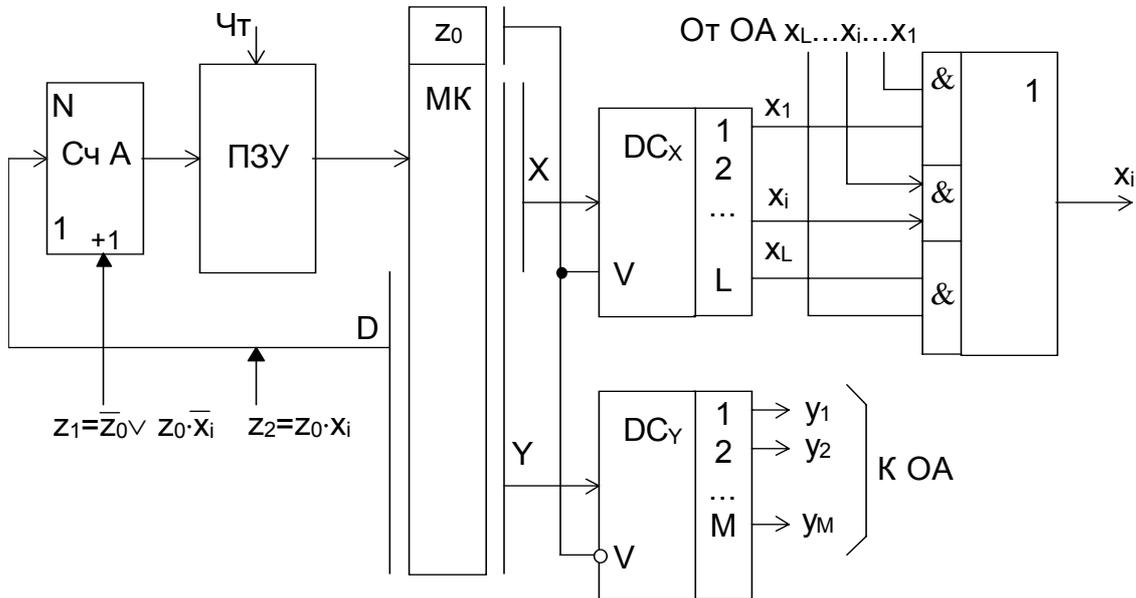


Рис. 80

Структура УА с естественной адресацией приведена на рис. 80.

Дешифратор управляющих сигналов DC_Y и дешифратор признаков DC_X стробируются в зависимости от бита признака микрокоманды z_0 , т.е. в зави-

симости от того, операционная эта МК или управляющая. Адрес МК хранится и преобразуется в счетчике $CчA$, который модифицируется в зависимости от формируемых признаков z_1 ($CчA:=CчA+1$) и z_2 ($CчA:=D$). Алгоритм функционирования УА показан на рис. 81.

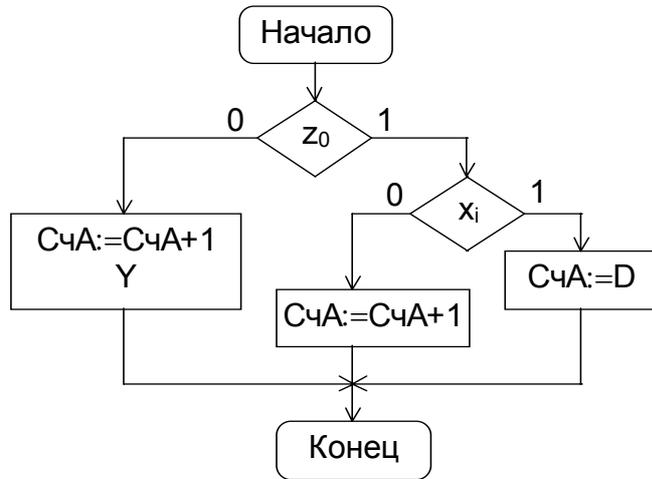


Рис. 81

Функционирование УА разделяется на последовательность микротактов, в течение которых определяются действия, определяемые микропрограммой. Если $z_0=0$, то выполняется операционная МК. При этом совместно с МО $CчA:=CчA+1$, выполняемой в УА, выполняются МО $Y=\{y_1, y_2, \dots, y_M\}$ в операционном автомате. За счет этого процесс формирования адреса следующей микрокоманды в УА

совмещается с выполнением микроопераций в ОА (конвейеризация команд).

Пример. Микропрограмма с естественной адресацией для умножения двух целых чисел без знака.

Управляющие сигналы в микропрограмме – $y_1, y_2, y_3, y_4, y_5, y_6$. В качестве признаков необходимо взять инверсные значения X и g . Действительно, из граф-схемы алгоритма умножения (рис. 76) следует, что все передачи управления (переходы к МК с адресом D) совершаются, если ЛУ принимает значение 0, иначе выполняется следующая микрокоманда. В граф-схеме на рис. 81 логика действий обратная, следовательно, надо вести анализ условий в инверсной форме.

Допустим, $\bar{g} = 1$, т.е. $g = 0$; УА должен находиться в состоянии ожидания, что имеет место, поскольку при этом управление будет передаваться МК с адресом ждущей вершины.

Словесное описание процедуры выглядит следующим образом:

Начало
 М1: Если \bar{g} , то М1
 y_1 : $C:=0$
 y_2 : $Cч:=0$
 М2: Если \bar{x}_1 , то М3
 y_3 : $C:=C+A$
 М3: y_4 : $V:=C(0).R1(B)$
 y_5 : $C:=0.R1(C)$
 y_6 : $Cч:=Cч+1$
 Если \bar{x}_2 , то М2
 Конец

Кодовое выражение микропрограммы приведено в табл. 40, где УП – условный переход, БП – безусловный переход.

Т а б л и ц а 40

Метка	Адрес	Разряды микрокоманд							Примечание
		z_0	\bar{g}/y_1	\bar{x}_1/y_2	\bar{x}_2/y_3	D_2/y_4	D_1/y_5	D_0/y_6	
M1	0	1	1	0	0	0	0	0	УП к ждущ. верш.
	1	0	1	1	0	0	0	0	
M2	2	1	0	1	0	1	0	0	УП к M3
	3	0	0	0	1	0	0	0	
M3	4	0	0	0	0	1	1	1	УП к M2
	5	1	0	0	1	0	1	0	
	6	1	0	0	0	0	0	0	БП к M1

Приведенные в табл. 40 коды микрокоманд (семь семиразрядных слов) должны быть записаны в ПЗУ, откуда они будут считываться в порядке, определяемом микропрограммой.

4.7. Контрольные вопросы и задания для самоподготовки.

1. Описать функциональное назначение в структуре операционного устройства его составляющих компонентов: операционного и управляющего автоматов.
2. Дать определение микрооперации, микрокоманды и микропрограммы.
3. Дать понятие функциональной микропрограммы.
4. Привести примеры описаний слов, массивов, двоичных выражений.
5. Классифицировать микрооперации и привести их примеры.
6. Привести правила построения содержательного графа микропрограммы.
7. Составить функциональную микропрограмму умножения двух целых чисел со знаком.
8. Описать структурный базис операционных устройств и привести примеры составляющих компонентов.
9. Привести структуру операционного автомата и описать его функционирование.
10. Привести пример операционного автомата для устройства умножения двух целых чисел со знаком.
11. По структуре операционного автомата для устройства умножения двух целых чисел со знаком синтезировать операционные элементы для всех микроопераций.
12. Дать примеры графов автоматов типа Мили и типа Мура для микропрограмм умножения двух целых чисел со знаком.
13. Представить полученные в п.12 графы управляющих автоматов в виде графов переходов и выходов.
14. Описать функционирование управляющих автоматов с естественной и принудительной адресацией микрокоманд.
15. Привести кодовое выражение микропрограммы с естественной адресацией для умножения двух целых чисел со знаком.
- 16*. Синтезировать операционное устройство умножения двух целых восьмиразрядных чисел без знака (задание для курсовой работы).

Глава 5. Запоминающие устройства

5.1. Классификация и основные характеристики запоминающих устройств

В настоящее время для хранения информации в цифровых, в том числе микропроцессорных вычислительных устройствах (МПВУ), используются различные запоминающие устройства (ЗУ) – от отдельных триггеров и регистров до внешних ЗУ большой емкости. Так, например, при построении главной памяти (ГП) используются полупроводниковые запоминающие устройства, а в устройствах внешней памяти применяются накопители на магнитных носителях (ленты – НМЛ, диски – НМД), накопители на лазерных компакт-дисках (CD). В рамках курса будет рассмотрена организация и характеристики памяти на полупроводниковых элементах, т.е. построение ГП.

Память ЭВМ представляет собой *одномерную матрицу*, состоящую из двоичных *n-битных* ($n=8, 16\dots$) *ячеек*, причем для хранения каждого бита информации требуется один *запоминающий элемент* (ЗЭ), к примеру, триггер. При обращении (чтение, запись) ячейка памяти селектируется *адресом* или номером этой ячейки в запоминающем устройстве.

ЗУ могут быть классифицированы:

- *по назначению* (сверхоперативная память (СОЗУ, кэш) для временного хранения информации в составе самого процессора; оперативная память для хранения часто используемых программ и данных, которая в свою очередь делится на оперативные ЗУ или RAM, предназначенные для записи, хранения и чтения информации, а также постоянные ЗУ или ROM, предназначенные только для хранения и чтения информации; *буферные* ЗУ для хранения больших массивов информации);
- *по быстродействию* (предъявляются высокие требования по быстродействию для СОЗУ(ОЗУ), которые реализуются на основе полупроводниковых ИМС, требования по быстродействию для внешних ЗУ существенно ниже, но они должны иметь высокую емкость и низкую стоимость на единицу хранения информации);
- *по способу хранения информации в ЗЭ* (статические ЗУ и динамические; в статических ЗУ хранение информации производится в триггерах, а динамической – на конденсаторах, при этом время хранения информации в триггерах не ограничено, тогда как в динамических ЗУ время хранения информации на конденсаторах определяется их разрядом через сопротивления утечки, поэтому в целях сохранения информации заряды конденсаторов необходимо периодически регенерировать);

- по виду доступа к информации (ЗУ с произвольной выборкой (адресацией) – ЗУПВ, и с последовательным доступом, т. е. в порядке убывания или возрастания адресов, что характерно для внешних ЗУ);
- по способу занесения информации в ПЗУ (масочные ПЗУ, программируемые заводом-изготовителем; однократно программируемые потребителем ПЗУ пережиганием перемычек; многократно программируемые (репрограммируемые) потребителем ПЗУ со стиранием ранее записанной информации ультрафиолетовым излучением; перепрограммируемые потребителем ПЗУ с электрическим стиранием ранее записанной информации);
- по технологии изготовления (КМОП, n-МОП, р-МОП, ТТЛ, ЭСЛ, И²Л и др.);
- по виду выхода (выход с открытым коллектором, выход с тремя состояниями).

Важнейшие характеристики ЗУ:

- ◆ тип ЗУ и его структурные особенности (характер хранения информации, вид доступа и т.д.);
- ◆ общая емкость ЗУ в Кбитах-Кбайтах и внутренняя организация, задающая число слов M и их разрядность n ($M \times n$);
- ◆ временные параметры (время обращения к ЗУ);
- ◆ управление режимами работы ЗУ (соответствие между сигналами управления и самими режимами);
- ◆ электрические данные (напряжение питания U , ток потребления I и потребляемая мощность $P_{\text{потр.}}$);
- ◆ технологическое и конструкторское исполнение (технология, тип корпуса, выводы);
- ◆ технико-экономические показатели (стоимость 1 бита хранимой информации, параметры надежности – наработка на отказ).

5.2. Оперативные запоминающие устройства

5.2.1. Построение запоминающих устройств с произвольной выборкой

На кристалле ИМС запоминающего устройства формируется регулярная структура из ЗЭ, называемая *накопителем*, и логическое обрамление накопителя (дешифраторы выбора адресов ЗЭ, буферные регистры, усилители записи-считывания, элементы управления). Структура накопителя определяется формулой $M \times n$, где M - число адресуемых слов, n - число бит в слове. Если $n = 1$, то такой накопитель называется *матрицей*, в нем есть доступ к любому ЗЭ. Тип микросхем ЗУПВ обозначается буквами РУ.

В накопителе отдельные ЗЭ подключены к адресным и разрядным линиям (АЛ и РЛ). При словарной адресации накопителя ($n \geq 2$) обращение (адреса-

ция) может быть только одновременно к нескольким ЗЭ, объединенным в слово (рис. 82,а).

При матричной организации накопителя можно обратиться к единственному биту, независимо от других. Выбор нужного ЗЭ задается пересечением соответствующих адресных линий X и Y (рис. 82,б).

Пример. Рассмотрим структуру ОЗУ матричного типа (рис. 83,а). Временные диаграммы цикла запись-чтение показаны на рис. 83,б. Адресная m -разрядная шина A подключена к дешифратору X (P разрядов) и к дешифратору Y (L разрядов),

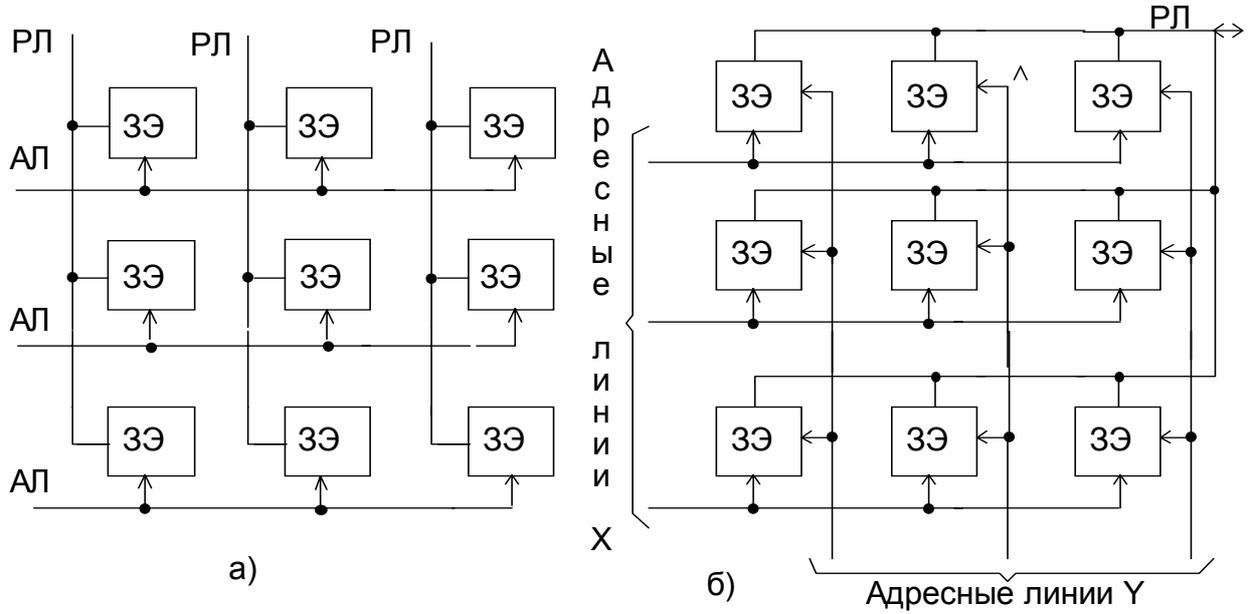


Рис. 82

$P+L=m$, что дает $2^P * 2^L = 2^{P+L} = M$ адресуемых запоминающих элементов.

Сигнал выбора кристалла $/CS$ здесь играет роль синхроимпульсов записи-чтения. К моменту появления $/CS = 0$ должны быть стабильны сигналы на шине адреса A, сигналы записи-чтения $/WR(RD)$ и входа-выхода DIO . Усилители (буф-

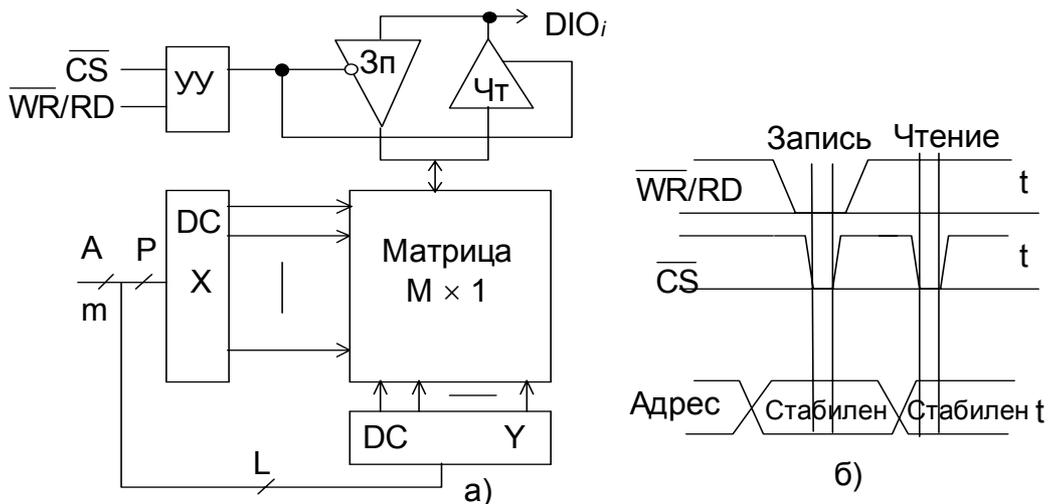


Рис. 83

ры) чтения-записи должны иметь тристабильный выход. Устройство управления (УУ) в зависимости от режима записи-чтения разрешает прохождение сигнала только через один из буферов, выход второго буфера переводится в высокоимпедансное состояние.

5.2.2. Особенности динамического ОЗУ.

В отличие от статических ОЗУ, в которых однажды записанная информация может храниться бесконечно долго при сохранении питания, в динамических ЗУ информация сохраняется в течении непродолжительного времени (1...2 мс), в связи с чем возникает необходимость ее периодического восстановления (регенерации, “освежения” – refresh).

В динамическом ЗЭ хранение информации осуществляется с помощью конденсатора (паразитной емкости $C_{зи}$ полевого транзистора). Рассмотрим ДЗУ (DRAM), в котором ЗЭ реализован с помощью всего лишь трех МОП-транзисторов (рис. 84). Такие ЗУ обладают более высокой степенью интеграции на кристалл и имеют меньшую стоимость по сравнению со статическими ОЗУ (SRAM). В данном случае используются отдельные адресные линии при записи информации и считывании.

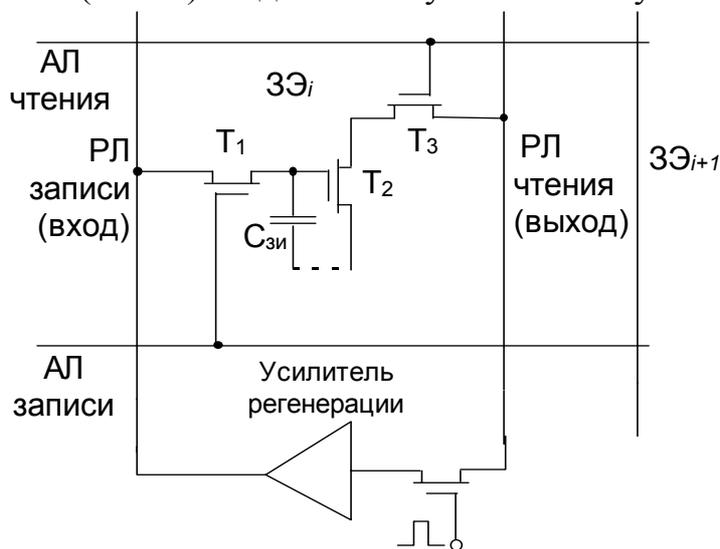


Рис. 84

записи информации и считывании.

Запись в ЗЭ происходит следующим образом: T_1 - открыт (выбрана адресная линия записи $АЛ_{зп}$) и на разрядную линию записи $РЛ_{зп}$ подается сигнал “1”. Паразитная емкость $C_{зи}$ заряжается через открытый транзистор T_1 , после чего T_2 будет открыт (насыщен), что соответствует единичному состоянию ЗЭ.

При *чтении*, когда T_3 - открыт (i -ый запоминающий элемент $ЗЭ_i$ селектирован по адресной линии чтения $АЛ_{чт}$), в зависимости от состояния T_2 на выходе (разрядной линии чтения $РЛ_{чт}$) будет либо 0, либо 1.

Как уже было сказано, разряд емкости $C_{зи}$ через сопротивление утечки обуславливает необходимость регенерации записанной информации с необходимой периодичностью (1...0,5 кГц).

Регенерация – это перезапись читаемого слова информации с использованием усилителей регенерации, входящих в состав ДЗУ, при этом читаемое слово на выходе ДЗУ не появляется; имеет место внутренний цикл “чтение-запись”.

Если перезаписываемое слово имеет разрядность n бит, то необходимо n усилителей регенерации. Цикл регенерации включает в себя сканирование всех слов по линиям чтения-записи. Это может быть сделано либо с помощью внешних схем, либо с использованием внутренних устройств регенерации ДЗУ.

5.2.3. Построение ЗУ большой емкости

При построении оперативной памяти информационно-вычислительных систем используют ЗУ большой емкости (десятки и сотни Кбит). Как правило, при этом используются более дешевые динамические ЗУ (ДЗУ). Из-за ограничений по числу выводов в корпусе ИМС в ряде случаев адрес передается в ИМС в два приема, по частям. Вначале обычно следует адрес строки, затем адрес столбца для ОЗУ матричного типа.

На рис. 85 (а,б) представлена структура типичного ДЗУ, предназначенного для хранения $M \times N$ одноразрядных чисел и временные диаграммы адресации.

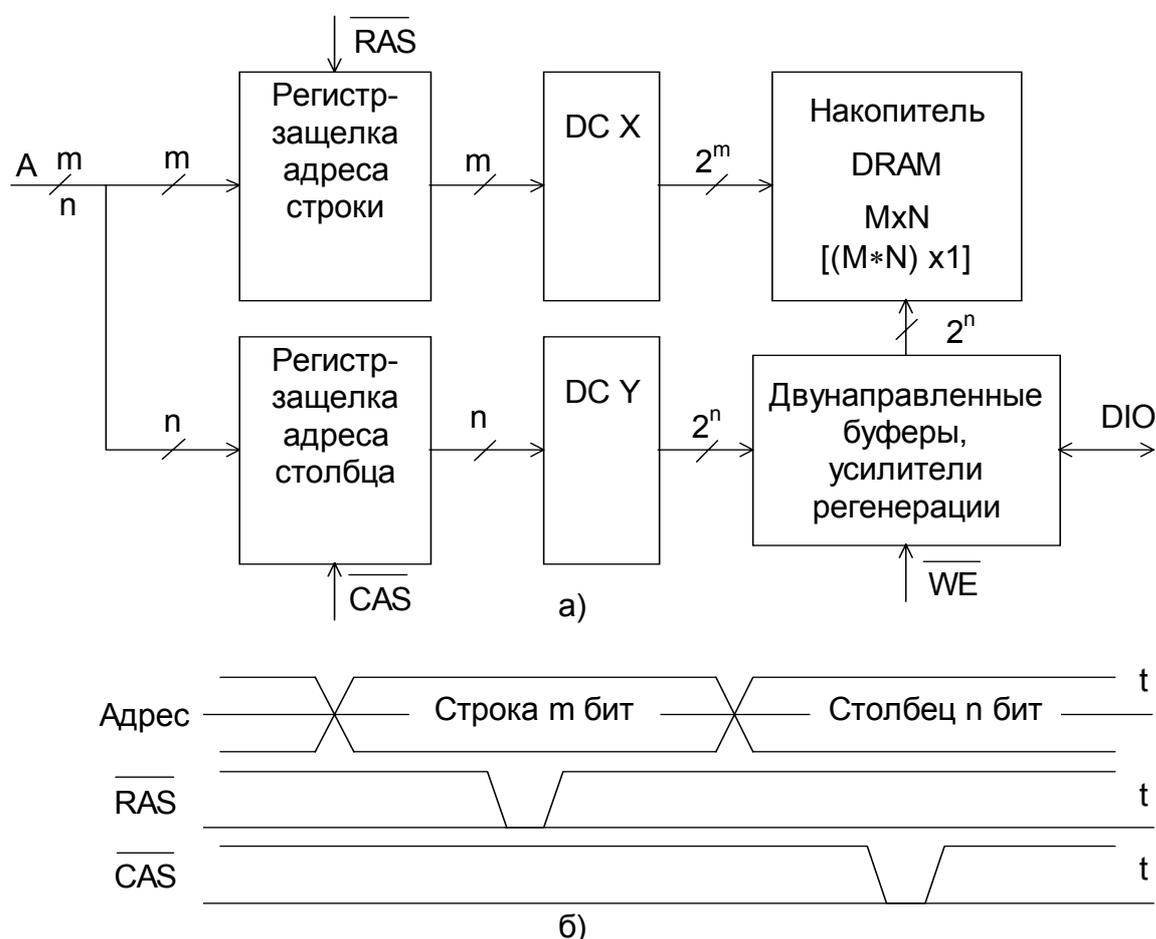


Рис. 85

Адрес запоминающего элемента задается $(m+n)$ -разрядным кодом, передаваемым по одним и тем же адресным линиям в два приема. Режимы работы

Для увеличения разрядности хранимых слов данных, используя однотипные ЗУ меньшей разрядности, их адресные и управляющие входы объединяются параллельно.

На рис. 86 показана структура ЗУ $M \times n$ с использованием n матричных ЗУ емкостью $M \times 1$, при этом число проводников в адресной шине $m = \log_2 M$.

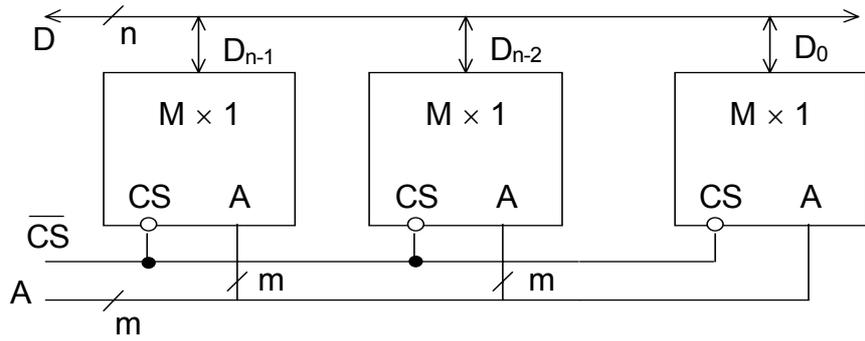


Рис. 86

Сигнал выбора кристалла \overline{CS} является разрешающим для подключения ЗУ к шине данных D ($D_{n-1} - D_0$).

Для увеличения емкости ЗУ (адресного пространства) с сохранением разрядности хранимых слов необходимо объединить тристабильные выходы ИМС, а также младшие адресные линии; старшие адресные линии используются с помощью дешифратора для селекции отдельных микросхем ЗУ.

На рис. 87 показана структура ЗУ емкостью $M \times 2^k$ n -разрядных слов на базе ИМС памяти $M \times n$, при этом младшие $m = \log_2 M$ адресных линий ИМС объединены параллельно, а с помощью кода на k старших линиях шины адреса

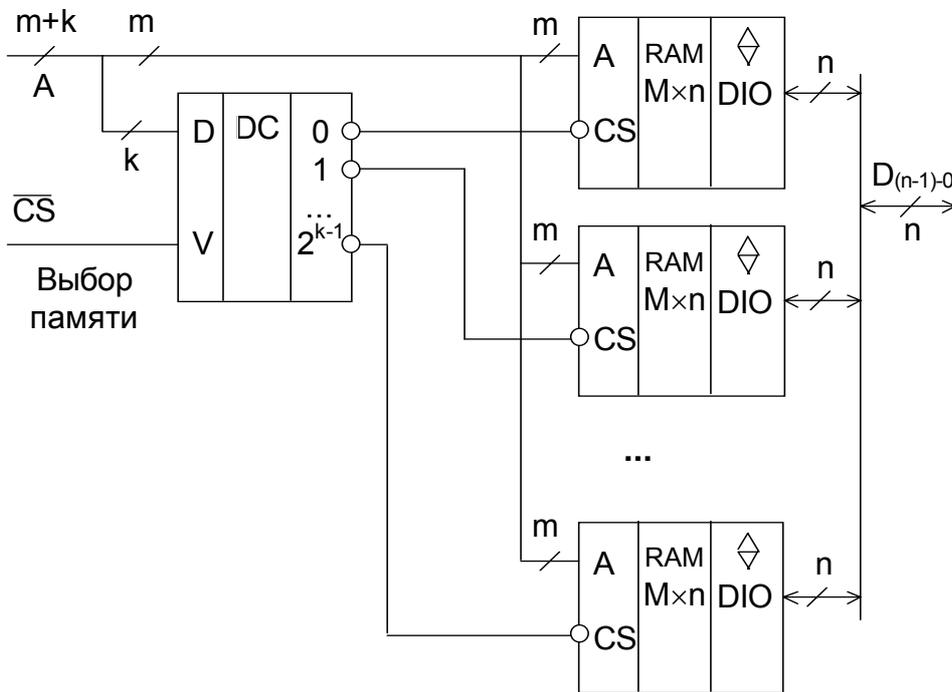


Рис. 87

через дешифратор DC с L – активными выходами включаются по входам \overline{CS}

однотипные ИМС памяти с организацией $M \times n$. Полученное ЗУ представляет собой совокупность 2^k смежных банков памяти, каждый емкостью $M=2^m$ n -разрядных слов.

На практике реализация памяти заданной структуры требует комбинации объединения ИМС с целью увеличения как разрядности, так и адресного пространства.

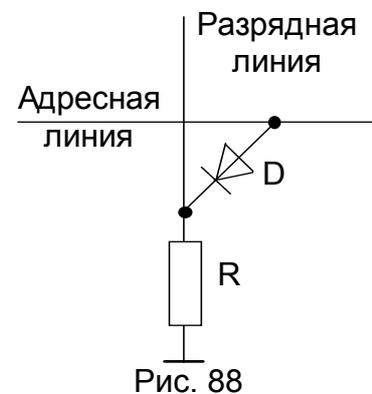
5.3. Постоянные запоминающие устройства

Как и ОЗУ, ПЗУ состоит из ячеек, обратившись к которым можно прочитать их содержимое. Отличие ПЗУ от ОЗУ состоит в том, что информация в ячейки ПЗУ записывается либо однократно, либо может быть модифицирована в результате специальной процедуры перепрограммирования, после чего в штатном процессе эксплуатации используется лишь режим чтения (read only memory - ROM).

5.3.1. Масочные ПЗУ

По способу занесения информации имеются *масочные* ПЗУ, программируемые предприятием изготовителем, и ПЗУ, программируемые пользователем.

Программирование масочных ПЗУ производится в процессе их изготовления. Тип таких микросхем обозначается буквами РЕ. Вначале формируются на кристалле все ЗЭ, затем по фотошаблону (маске) наносится требуемая сеть соединений по информации, предоставляемой заказчиком. Данная сеть соединений может быть реализована, например, с помощью диодной матрицы-решетки, где наличие диода (соединения) приводит к появлению на выходе РЛ логической единицы при селекции данной строки с помощью АЛ (рис. 88). Для масочных ПЗУ характерна самая высокая степень интеграции и при большой серийности изделий это самые дешевые запоминающие устройства. Их достоинством является также простая регулярная структура, и, следовательно, более высокая надежность.



5.3.2. Однократно программируемые ПЗУ

Занесение информации в однократно программируемые ПЗУ производится пользователем. Использование программируемых ПЗУ (ППЗУ) характерно для мелкосерийного производства и на этапах изготовления образцов изделий (микросхемы имеют буквенное обозначение – РТ). ППЗУ по сравнению с ПЗУ имеют более сложную структуру, в связи с наличием *плавких нихромо-вых или поликремниевых перемычек*, прожигание которых обеспечивает за-

пись информации, а также из-за дополнительных элементов, через которые осуществляется программирование. Программирование производится с помощью специальных устройств – программаторов. Прожигание переключки L ведет к записи в ЗЭ логического нуля (рис. 89), т.е. до процедуры программирования все ЗЭ хранят логические единицы. В программировании достаточно много проблем (сложный цикл программирования, высокий процент брака). Следует отметить необходимость специальных циклов термотренировки для устранения возможности восстановления некоторых переключек после программирования. Для программирования используются те же выводы, которые служат для чтения информации.

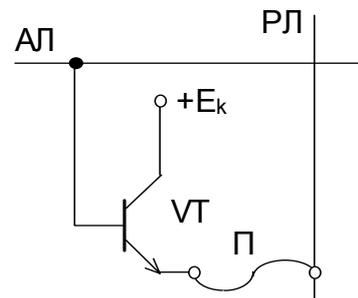


Рис. 89

5.3.3. Перепрограммируемые ПЗУ

Для многих применений, как в условиях массового производства, так и мелкосерийного, удобными являются **репрограммируемые** постоянные ЗУ, допускающие выполнение многих циклов перепрограммирования с предшествующим стиранием ненужной информации.

Число циклов перепрограммирования у различных типов ППЗУ составляет от нескольких десятков до нескольких тысяч. Гарантированное время хранения информации доходит до нескольких лет.

В ЗЭ таких устройств чаще всего используются МОП-транзисторы с двухслойной структурой диэлектрика или с дополнительными плавающими затворами, позволяющие за счет введения избыточного заряда изменять пороговое напряжение, при котором отпираются транзисторы. Благодаря исключительно низкой проводимости используемых диэлектриков избыточные заряды могут существовать без значительного уменьшения своей величины очень длительное время.

Перепрограммируемые ПЗУ имеются двух типов: с *электрическим* (микросхемы имеют буквенное обозначение – РР) и *ультрафиолетовым* (буквенное обозначение – РФ) стиранием информации. При электрическом стирании возможно избирательное (побайтовое) стирание информации с последующим выполнением байтовой записи. Стирание с помощью ультрафиолетового облучения приводит к полному разрушению первоначально хранящейся информации.

Элемент памяти с ультрафиолетовым стиранием и электрической записью показан на рис. 90. При подаче достаточно большого напряжения к $p-n$ переходу истока либо стока МОП – транзистора T_2 происходит инжекция электронов в его изолированный затвор (рис.93,б), где этот заряд может храниться

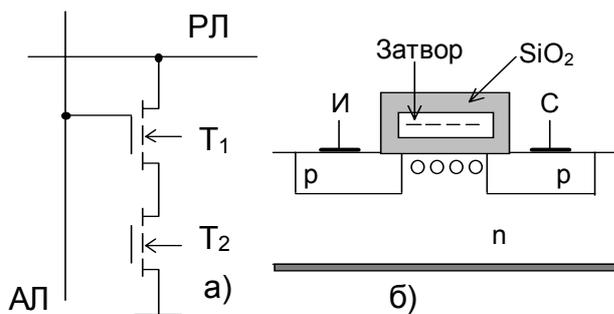


Рис. 90

достаточно долго. Это запись логического “0” в ЗЭ, т.к. канал транзистора T_2 в n -области подложки становится проводящим (заряд затвора притягивает дырки). Транзистор T_2 при этом открыт (сопротивление сток – исток мало). Таким образом, в исходном состоянии (до программирования) все запоминающие элементы

такой структуры хранят логические “1”. Поток квантов ультрафиолетового излучения приводит к исчезновению заряда в затворе, восстановлению непроводящего состояния транзистора T_2 и, тем самым, появлению в ЗЭ логической единицы.

Сложная структура ЗЭ, необходимость введения элементов перепрограммирования, значительно повышают стоимость ПЗУ по сравнению с другими разновидностями ПЗУ. Поэтому они применяются в первую очередь там, где свойство перепрограммирования является определяющим, например, для хранения информации определяющей конфигурацию компьютера и его основные параметры (*FLASH*-память в микросхеме CMOS).

Постоянные ЗУ можно рассматривать не только как элементы памяти, хранящие произвольные записанные в них коды, но и как универсальные элементы – преобразователи информации, например, m -разрядного входного кода (адресная шина) в n -разрядный выходной код (шина данных). Решение подобного логического m - n -полюсника рассматривалось в гл. 2 с помощью КЦУ. Использование ПЗУ в случае $m \gg n$ может оказаться нерациональным, так как при этом число различных на выходе кодовых комбинаций 2^m будет много меньше числа n -разрядных элементов памяти ЗУ (2^n), т.е. решение задачи указанным способом характеризуется большой аппаратной избыточностью.

5.3.4. Программируемые логические интегральные схемы

Компромиссным вариантом является использование такого решения, при котором, с одной стороны, сохраняется принцип построения КЦУ, основанный на применении ограниченного числа логических элементов с минимальной избыточностью (минимизированное представление КЦУ), с другой стороны, сохраняются преимущества постоянных ЗУ, связанные с регулярной структурой элементов и программируемостью связей между ними для решения конкретных задач. В целом все программно-аппаратное разнообразие подобных решений можно объединить под общим названием – **программируемые логические интегральные схемы (ПЛИС)**. В зарубежной и отечествен-

ной литературе такие устройства получили несколько названий – *программируемые логические устройства (PLD – Programmable Logic Device)* или *программируемые логические матрицы – ПЛМ (PLA – Programmable Logic Array)*.

Основой ПЛМ (PLD) является набор нескольких уровней логических элементов заданного базиса (И, ИЛИ, И-НЕ и т.д.) На каждом уровне используются однотипные элементы с одинаковым числом входов, что создает регулярную структуру. Между уровнями элементов вводятся системы (матрицы) горизонтальных и вертикальных линий, на пересечении которых в нужных местах при программировании выполняются электрические соединения. Это можно сделать либо на этапе производства ИМС (как в случае масочных ПЗУ), либо введением в ИМС элементов программирования (как в случае репрограммируемых ПЗУ). На рис. 91 приведен пример структуры, иллюстрирующий основные принципы построения PLD (ПЛМ).

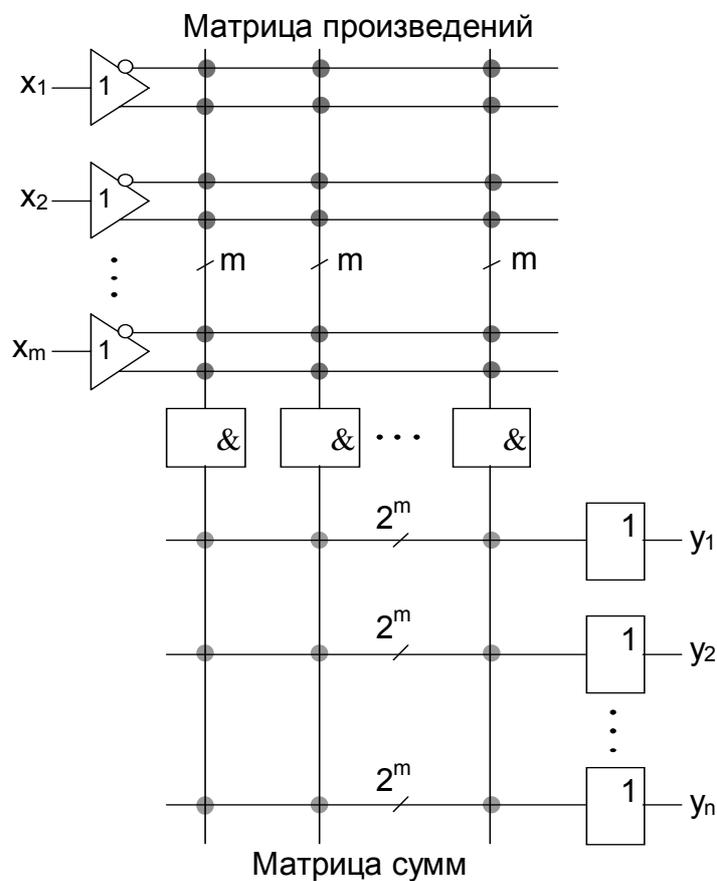


Рис. 91

соединений в матрице сумм. Очевидно, что PLD реализует систему n логических функций m аргументов в базисе И, ИЛИ, НЕ в соответствии с совершенной дизъюнктивной нормальной формой (СДНФ).

рующей основные принципы построения PLD (ПЛМ).

Буферные элементы *первого уровня* используются для получения парафазных значений (прямые и инверсные) m -разрядного входного кода $x_m \dots x_1$ и обеспечивают необходимую нагрузочную способность.

На выходе каждого логического элемента И *второго уровня* формируется произведение входных сигналов или их инверсий в зависимости от их конкретных соединений к m входам ЛЭ И в матрице произведений.

Выходные сигналы устройства формируются 2^m -входными ЛЭ ИЛИ *третьего уровня* в зависимости от со-

В зависимости от того, структура какой из матриц является фиксированной, а какой – программируемой, PLD делятся на PROM (ППЗУ), PAL (*Programmable Array Logic*) и PLA (*Programmable Logic Array*).

В ППЗУ матрица произведений и 2^m схем И образуют дешифратор адреса и потому соединения в матрице произведений фиксированы, а соединения в матрице сумм формируют требуемые кодовые комбинации на выходах n ЛЭ ИЛИ. В PAL программируемой является матрица произведений, матрица сумм – фиксированная. В устройствах PLA обе логические матрицы произведений и сумм являются программируемыми, что делает эти устройства наиболее гибкими. В то же время PLA имеют некоторые недостатки по сравнению с PROM и PAL. Во-первых, за счет прохождения сигнала по двум матрицам уменьшается быстродействие, и во-вторых, программирование двух матриц PLA более сложное, чем одной матрицы в PAL или PLD.

Рассмотренная выше структура PLD относится к комбинационным цифровым устройствам. Имеются программируемые логические интегральные схемы, содержащие кроме ЛЭ еще и элементы памяти (триггеры и регистры), что позволяет на их основе строить последовательностные цифровые устройства. Например, последовательностный PLD 16R4 (фирма AMD) имеет 16 входов и четырехразрядный регистр в своем составе. Структура последовательностного PLD соответствует реальному конечному автомату (рис. 29). При построении устройств на базе ПЛИС используются специализированные программно-аппаратные комплексы, в настоящее время, как правило, на платформе персонального компьютера. Вопросы проектирования цифровых устройств на основе ПЛИС выходят за рамки пособия и здесь не рассматриваются.

5.4. Контрольные вопросы и задания для самоподготовки.

1. Классифицировать запоминающие устройства и привести их основные технические характеристики.
2. Описать функционирование ЗУПВ матричного типа по приведенной схеме.
3. Описать принципы построения ДЗУ. Регенерация ДЗУ.
4. Построить ОЗУ 64К×8 на базе:
 - БИС ЗУ 4К×4;
 - БИС ЗУ 8К×2;
 - БИС ЗУ 8К×1;
 - БИС ЗУ 2К×8.
5. Пояснить отличия в реализации ПЗУ (масочные, однократно-программируемые и перепрограммируемые)
6. Описать особенности построения ПЛИС типа PROM, PAL и PLA.
7. Синтезировать на базе ПЛИС логическую функцию (по индивидуальному заданию).

Глава 6. Формирователи, генераторы и преобразователи импульсных и цифровых сигналов

6.1. Формирователи импульсов на основе элементов задержки

6.1.1. Элемент задержки с интегрирующей RC-цепью и логическим элементом ТТЛ

Принципиальная схема элемента задержки (ЭЗ), эквивалентные схемы и временные диаграммы сигналов, поясняющие его работу на отдельных этапах представлены на рис. 93, а-д.

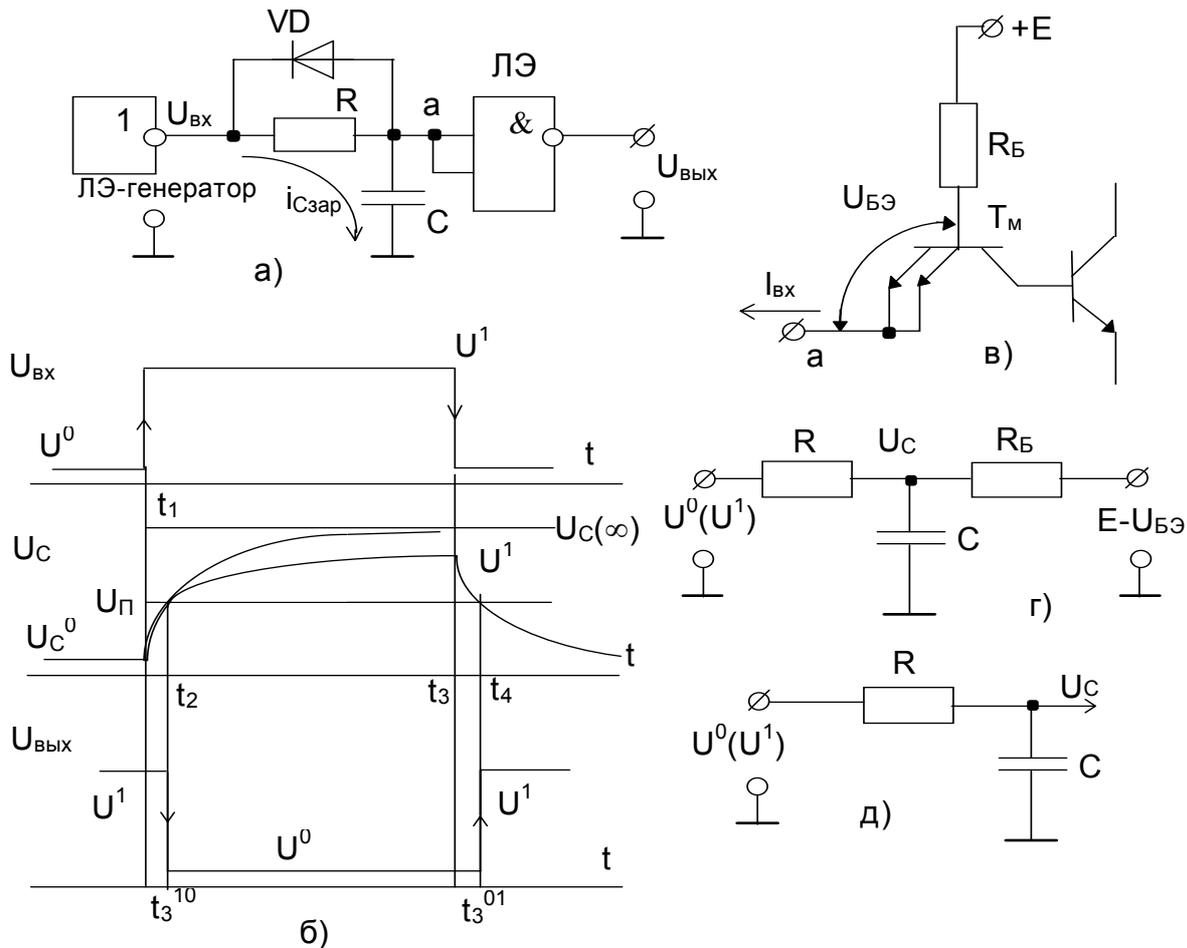


Рис. 93

À èñòàííí òîòòîýèè ($t < t_1$) ËÝ-ãàíáäàòîð ìèèèùò, $U_{\hat{A}\hat{O}} = U^0$, èíááíñàòîð C ðàçðÿ-
 æåí, ñàïðÿæåíèå ñà ñåì ìèíèìàëüíî è ñå òðåâûøàåò óðîâíÿ U_{Π} , òðè êîòîðîì
 çàïèðàåòñÿ âõîäíÿÿ ìíîãîýìèòòåðíÿÿ òðàíçèñòîð T_M ËÝ. Òàêèì îáðàçîì,
 òðàíçèñòîð T_M ñàñûùåí (ðèñ. 93, в), ðàáî÷èé ËÝ çàïåðò, $U_{\hat{A}\hat{O}} = U^1$.

Начальное напряжение $U_C^0 = U_C(t_1)$ согласно эквивалентной схеме на рис. 93,г

$$U_C^0 = U^0 + U_R = U^0 + \frac{E - U_{БЭ} - U^0}{R + R_B} R = \frac{E - U_{БЭ}}{R + R_B} R + \frac{U^0}{R + R_B} R_B.$$

В момент времени t_1 конденсатор C заряжается по цепи $U_{АЭ} \rightarrow R \rightarrow C \rightarrow \perp$ при ещё насыщенном транзисторе T_M .

Напряжение на конденсаторе растет по экспоненциальному закону, стремясь в пределе к $U_C(\infty)$, определяется по принципу суперпозиции из рис. 93, г

$$U_C(\infty) = \frac{E - U_{БЭ}}{R + R_B} R + \frac{U^1}{R + R_B} R_B > U^1.$$

Рост $U_C(t)$ продолжается с постоянной времени заряда $\tau_{C3} = C \frac{R \cdot R_B}{R + R_B}$ до

значения порогового напряжения $U_{П}$.

При достижении порогового уровня $U_{П}$ в момент t_2 транзистор T_M запирается, а ЛЭ открывается, $U_{ВЫХ} = U^0$. Конденсатор C начинает разряжаться согласно эквивалентной схеме на рис. 93, д с постоянной времени $\tau_{Cp} \approx CR$, так как при $t_3 \leq t \leq t_4$ транзистор T_M заперт.

$$t_3^{10} = t_2 - t_1 = \tau_{C3} \ln \frac{U_C(\infty) - U_C^0}{U_C(\infty) - U_{П}}$$

До момента t_4 конденсатор C разряжается медленнее, так как постоянная времени $\tau'_{C3} = RC > \tau_{C3}$ (рис. 93, д), и $U_{ВЫХ}(t)$ растет до уровня U^1 . В момент t_3 конденсатор C начинает разряжаться согласно эквивалентной схеме на рис. 93, д с постоянной времени $\tau_{Cp} \approx CR$, так как при $t_3 \leq t \leq t_4$ транзистор T_M заперт.

До момента t_4 конденсатор C разряжается медленнее, так как постоянная времени $\tau'_{C3} = RC > \tau_{C3}$ (рис. 93, д), и $U_{ВЫХ}(t)$ растет до уровня U^1 . В момент t_3 конденсатор C начинает разряжаться согласно эквивалентной схеме на рис. 93, д с постоянной времени $\tau_{Cp} \approx CR$, так как при $t_3 \leq t \leq t_4$ транзистор T_M заперт.

$$t_3^{01} = t_4 - t_3 = \tau_{Cp} \ln \frac{U^1 - U^0}{U_{П} - U^0}.$$

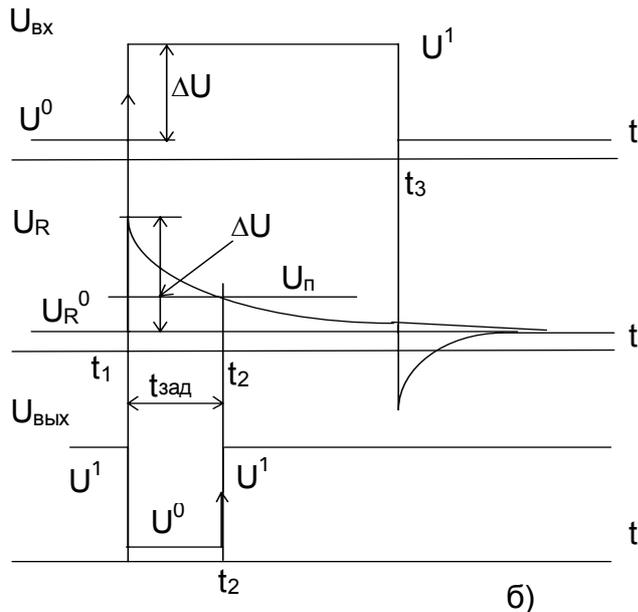
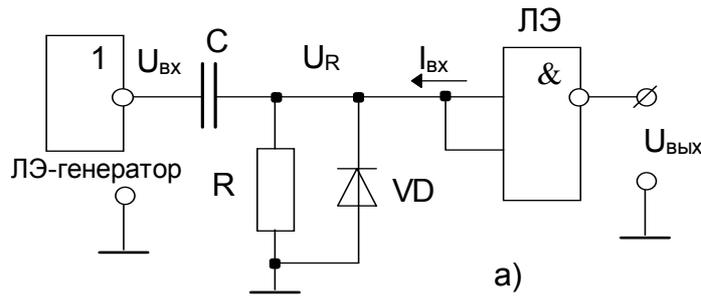
При разряде конденсатора $U_C(\infty)$ стремится к уровню U^0 . Задержка t_3^{10} вызвана запирающим ЛЭ, а t_3^{01} — отпирающим ЛЭ и в общем случае $t_3^{10} < t_3^{01}$, при $R = 1 \text{ кОм}$, $C = 1000 \text{ нФ}$, $t_3^{01} = 1 \text{ мкс}$, $t_3^{10} = 0.22 \text{ мкс}$. Для выравнивания задержек используют диод VD , шунтирующий резистор R .

Напряжение на конденсаторе $U_C(t)$ стремится к уровню U^0 . Задержка t_3^{10} вызвана запирающим ЛЭ, а t_3^{01} — отпирающим ЛЭ и в общем случае $t_3^{10} < t_3^{01}$, при $R = 1 \text{ кОм}$, $C = 1000 \text{ нФ}$, $t_3^{01} = 1 \text{ мкс}$, $t_3^{10} = 0.22 \text{ мкс}$. Для выравнивания задержек используют диод VD , шунтирующий резистор R .

6.1.2. Элемент задержки с дифференцирующей RC-цепью и ЛЭ ТТЛ

Принципиальная схема устройства и временные диаграммы его работы показаны на рис. 94, а,б.

В исходном состоянии ($t < t_1$) ЛЭ-генератор открыт, $U_{BX} = U^0$, ЛЭ заперт, $U_{BIX} = U^1$, поскольку $U_R^0 = I_{BX}R < U_{\Pi}$ и многоэмиттерный транзистор T_M ЛЭ насыщен.



Перепад входного напряжения в момент t_1 $\Delta U = U^1 - U^0$ через конденсатор C практически без изменения подается на вход ЛЭ, T_M запирается, а ЛЭ переключается в открытое состояние, $U_{BIX} = U^0$. В момент t_1 , $U_R(t) = U_R^0 + \Delta U > U_{\Pi}$.

Далее ($t > t_1$) конденсатор C заряжается ($U_c(t) \uparrow$, $U_R(t) \downarrow$) по экспоненциальному закону с постоянной времени $\tau_{C3} = RC$.

В момент t_2 $U_R(t) = U_{\Pi}$, ЛЭ закрывается и $U_{BIX} = U^1$. Величина задержки выходного перепада $t_2 - t_1 = t_{зад}$ определяется по формуле:

Рис. 94

$$t_{зад} = \tau_{C3} \ln \frac{\Delta U}{U_{\Pi} - U_R^0} = RC \ln \frac{U^1 - U^0}{U_{\Pi} - U_R^0}.$$

Диод VD служит для уменьшения отрицательного выброса напряжения на входе ЛЭ при разряде конденсатора, когда $U_{BX} = U^0$. В настоящее время такие "антизвонные" диоды стоят на входах ЛЭ в большинстве серий ИМС. Рассмотренный элемент задержки перепады сигнала на выходе не инвертирует.

6.1.3. Формирователь прямоугольных импульсов на основе элементов задержки

Функциональная схема формирователя и временные диаграммы работы в характерных точках приведены на рис. 95, а,б.

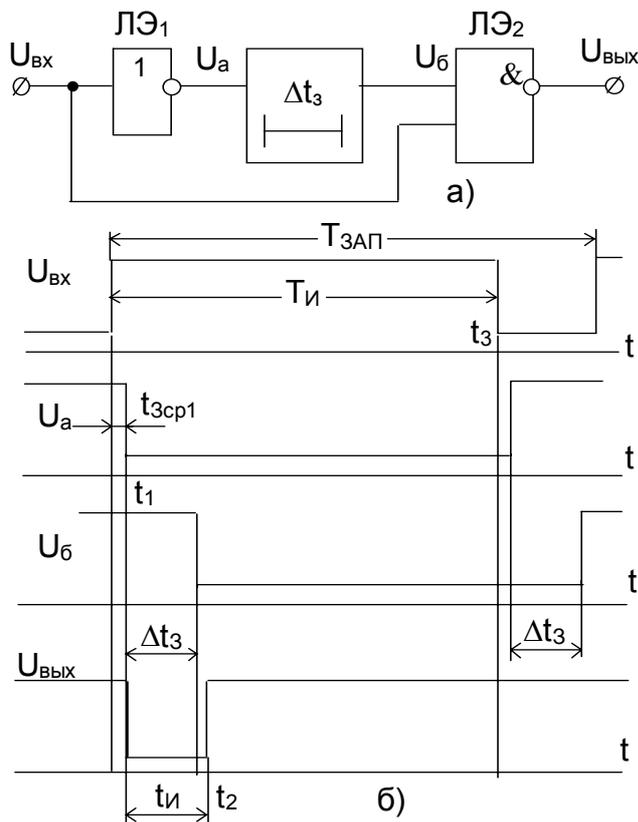


Рис. 95

В исходном состоянии ($t < t_1$) $U_{ВХ} = U^0$, $\ddot{E}\ddot{Y}_1$ è $\ddot{E}\ddot{Y}_2$ çàìáðòù, $U_{ВЫХ} = U_a = U_б = U^1$. Входной импульс инвертируется и подается на элемент задержки. С задержкой Δt_3 (элемент задержки неинвертирующий) этот импульс подается на выходной ЛЭ.

На выходе ЛЭ появляется инверсный импульс длительностью $t_{И} = \Delta t_3 + t_{3cp1} \approx \Delta t_3$, если пренебречь задержкой входного инвертора ($t_{3cp1} \approx 0$).

Восстановление исходного состояния схемы после переключения в момент t_3 происходит за время $t_{восст} = \Delta t_3 + t_{3cp1} \approx \Delta t_3$. Необходимо выполнить условие $t_{восст} < T_{ЗАП} - T_{И}$.

Элемент задержки может быть выполнен в виде цепочки из четного

числа последовательно включенных инверторов. В этом случае возможно формирование достаточно коротких импульсов. Для получения импульсов с крутыми фронтами ЛЭ₁ и ЛЭ₂ должны быть как можно более быстродействующими (ТТЛ КР1533 – $t_{3cp} = 4$ нс, К155 – $t_{3cp} = 10$ нс).

6.2. Генераторы импульсов прямоугольной формы

К числу генераторов импульсов прямоугольной формы, нашедших применение в ЦУ, относят:

- задающие генераторы импульсов тактовой частоты (ГТИ);
- генераторы серий импульсов (ГСИ);
- генераторы (формирователи) одиночных импульсов (ГОИ, ФОИ).

Их можно построить с использованием комбинационных логических устройств, триггеров, операционных усилителей (ОУ) и специализированных интегральных микросхем.

6.2.1. Автоколебательный мультивибратор на логических элементах ТТЛ

В цифровой схемотехнике широко используются автоколебательные и ждущие (заторможенные) *мультивибраторы* (МВ) – генераторы релаксационных колебаний с время-задающими RC-цепями. Автоколебательный мультивибратор (АМВ) – это релаксатор с емкостной связью, не имеющий ни од-

ного устойчивого состояния. В простейшем случае АМВ имеет два квазиустойчивых состояния, в которых он находится поочередно. Принципиальная схема автоколебательного МВ с использованием ЛЭ ТТЛ и временные диаграммы его работы показаны на рис. 96, а,б.

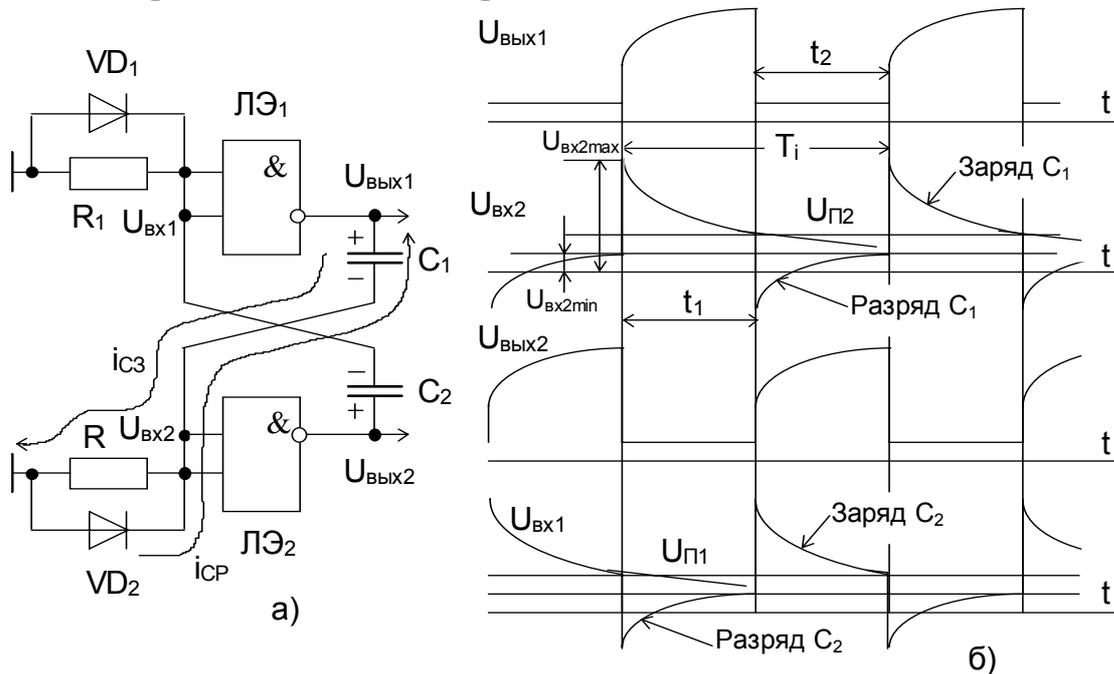


Рис. 96

АМВ обладает двумя автоматически сменяемыми состояниями квазиравновесия. При включении питания даже в симметричной схеме из-за флуктуаций возникает асимметрия, вызывающая лавинообразный регенеративный процесс. Пусть, например, $U_{ВЫХ1} > U_{ВЫХ2}$. Тогда рост $U_{ВЫХ1}$ через C_1 передается на вход ЛЭ₂. При достижении $U_{ВХ2}$ порогового уровня $U_{П2}$ ЛЭ₂ отпирается, $U_{ВЫХ2}$ уменьшается. Через C_2 уменьшение $U_{ВЫХ2}$ передается на вход ЛЭ₁, понижая его потенциал. Уменьшение $U_{ВХ1}$ вызывает повышение $U_{ВЫХ1}$ и т. д. При условии, если произведение коэффициентов усиления каскадов ЛЭ₁ и ЛЭ₂ $K_1 K_2 > 1$ (выполняется автоматически в работоспособной микросхеме) замыкается цепь положительной обратной связи (ПОС) (баланс фаз обеспечен структурой схемы) и протекает лавинообразный регенеративный процесс, усугубляющий первоначальные изменения. Скорость лавинообразного процесса обуславливается паразитными емкостями цепи. Этот процесс заканчивается полным запираем ЛЭ₁ и отпираем ЛЭ₂.

На первой стадии квазиравновесия в течении интервала t_1 ЛЭ₁ заперт, ЛЭ₂ – открыт; конденсатор C_1 заряжается по цепи: выход ЛЭ₁ → C_1 → R_2 → ⊥. При этом диод VD_2 заперт падением напряжения на R_2 от тока заряда $i_{C3}(t)$. По мере заряда $U_{C1}(t)$ даётся, ÷òî âîñóâààò òíáíüòáíèà òíèà çàäóàà $i_{N3}(t)$ è íàïäóæáíèü ìà äðíàá ËÝ₂ $U_{ВХ2}(t)$. В момент, когда сравниваются с порогом напряжения $U_{R2} = U_{ВХ2} = U_{П2}$, ЛЭ₂ начнет запирается, перейдя в активный режим. Напряже-

ние на выходе ЛЭ₂ скачком возрастает. Перепад $U_{ВЫХ2}(t)$ через C_2 передается на вход ЛЭ₁, $U_{ВХ1}$ и при $U_{ВХ1}=U_{П1}$ отпирается ЛЭ₁. При этом $U_{ВЫХ1}$ скачком уменьшается, что через конденсатор C_1 также скачком передается на вход ЛЭ₂, $U_{ВХ2}$ также уменьшается, что ведет к резкому возрастанию $U_{ВЫХ2}$. Таким образом, замыкается петля ПОС. Протекает регенеративный лавинообразный процесс, который заканчивается запирающим ЛЭ₂ и отпиранием ЛЭ₁. Первая стадия квазиравновесия на этом заканчивается.

На стадии t_1 $\perp \rightarrow$ диод $VD_1 \rightarrow C_2 \rightarrow$ выход ЛЭ₂. Поскольку $\tau_{3C1} \approx C_1 R_2 \gg \tau_{PC2} \approx C_2 (r_{Д1} + r_{ВЫХ, ЛЭ})$ длительность стадии t_1 $\approx C_1 R_2 \ln \frac{U_{ВХ2 MAX}}{U_{П2}}$.

Длительность t_1 определяется по формуле:

$$t_1 = t_{3C1} = \tau_{3C1} \ln \frac{U_{ВХ2 MAX}}{U_{П2}} \approx C_1 R_2 \ln \frac{U_{ВХ2 MAX}}{U_{П2}}.$$

Длительность разряда конденсатора C_2 на этой стадии определяет этап восстановления, который должен закончиться в течение интервала t_1 :

$$t_{B1} = t_{P C_2} = 3C_2 (r_{Д} + r_{ВЫХ, ЛЭ2}^{\alpha}).$$

Длительность второй стадии квазиравновесия t_2 определяется зарядом конденсатора C_2 , аналогично тому, как происходит заряд C_1 в течении t_1 . На этой же стадии C_1 разряжается по цепи: выход ЛЭ₁ $\rightarrow C_1 \rightarrow VD_2 \rightarrow \perp$.

Длительность t_2 и t_{B2} определяется аналогично t_1 и t_{B1} . Если $C_1=C_2$, $R_1=R_2$, то $t_1=t_2$. АМВ симметричен и генерирует импульсы со скважностью $Q=T_i/t_1=(t_1+t_2)/t_1=2$.

Диоды VD_1 и VD_2 служат для уменьшения отрицательных выбросов напряжения на входах ЛЭ (“антизвонные” диоды).

6.2.2. Ждущий мультивибратор на логических элементах ТТЛ

Ждущий (заторможенный) мультивибратор (ЖМВ) имеет одно устойчивое состояние, в котором он может находиться сколь угодно долго и одно состояние равновесия квазиустойчивое (временно-устойчивое). Переход из устойчивого во временно-устойчивое состояние осуществляется после воздействия импульса запуска, вызывающего скоротечный регенеративный процесс, заканчивающийся квазиравновесным режимом. Длительность последнего, как и длительность выходного импульса ЖМВ, определяется процессом заряда или разряда конденсатора во время задающей цепи.

Принципиальная схема и временные диаграммы устройства даны на рис. 97, а, б. В исходном (ждущем) состоянии ЛЭ₁ открыт и ЛЭ₂ – закрыт, поскольку $U_{3АП}=U^1$. Потенциал на выходе ЛЭ₂ $U_{ВЫХ2}=U^1$ подтверждает открытое состояние ЛЭ₁ ($U_{ВЫХ1}=U^0$). Конденсатор C при этом разряжен.

При поступлении запускающего импульса $U_{3АП}$ инверсной полярности (L -активный сигнал), мультивибратор лавинообразно переключается в состояние квазиустойчивого равновесия.

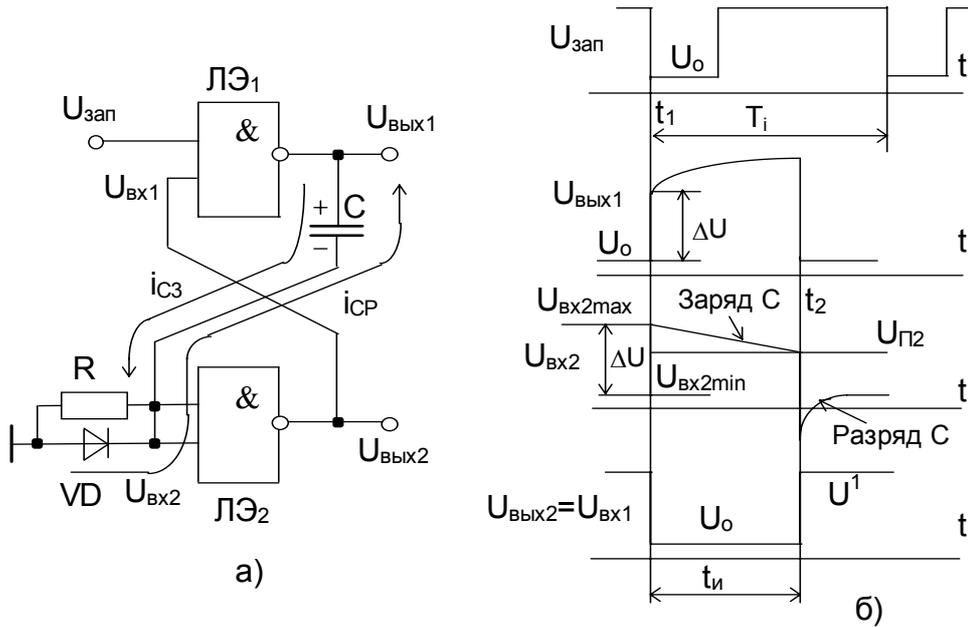


Рис. 97

Процесс переключения протекает следующим образом. Под воздействием $U_{3АП} = U^0$ ЛЭ1 запирается, перепад напряжения ΔU с выхода ЛЭ1 передается через конденсатор C без изменения на вход ЛЭ2, отпирая его. Замыкается петля положительной обратной связи и протекает регенеративный процесс ($U_{ВЫХ1} \uparrow \rightarrow U_{ВХ2} \uparrow \rightarrow U_{ВЫХ2} \downarrow = U_{ВХ1} \downarrow \rightarrow U_{ВЫХ1} \uparrow$), который заканчивается полным открыванием ЛЭ2. Выходное напряжение $U_{ВЫХ2} = U^0$ поддерживает запертое состояние ЛЭ1 после окончания запускающего импульса.

На стадии квазиравновесия конденсатор C заряжается по цепи: выход ЛЭ1 $\rightarrow C \rightarrow R \rightarrow \perp$, поскольку диод VD при этом закрыт падением напряжения на сопротивлении R . По мере заряда конденсатора ток $i_{C3}(t)$ уменьшается, вызывая уменьшение $U_{ВХ2}(t)$ и при достижении $U_{ВХ2}(t)$ порогового уровня $U_{П2}$ начинает закрываться ЛЭ2. Мультивибратор переключается в устойчивое состояние.

Второй регенеративный процесс при запираии ЛЭ2 протекает следующим образом: $U_{ВЫХ2} \uparrow \rightarrow U_{ВХ1} \uparrow \rightarrow$ ЛЭ1 открывается $\rightarrow U_{ВЫХ1} \downarrow \rightarrow U_{ВХ2} \downarrow \rightarrow U_{ВЫХ2} \uparrow$. Здесь, как и в первом случае, замыкается петля ПОС.

Длительность импульса $t_{И}$ на выходе ЛЭ2 определяется этапом квазиравновесия (зарядом C):

$$t_{И} = \tau_{Э} \ln \frac{U_{ВХ2 MAX}}{U_{П2}} = C(R + r_{ВЫХ}^{3АП}) \ln \frac{U_{ВХ2 MAX}}{U_{П2}}.$$

После открывания ЛЭ₁ конденсатор С разряжается по цепи: выход ЛЭ₁→VD→⊥. Время восстановления $t_B = 3C(r_{ВЫХ ЛЭ}^{ОКР} + r_D)$ должно быть меньше $T_i - t_{и}$, где T_i - период повторения запускающих импульсов.

Диод VD служит для уменьшения амплитуды отрицательного выброса напряжения на входе ЛЭ₂.

6.2.3. Автоколебательный мультивибратор на операционном усилителе

Принципиальная схема и временные диаграммы работы АМВ на операционном усилителе (ОУ) представлены на рис. 98, а,б.

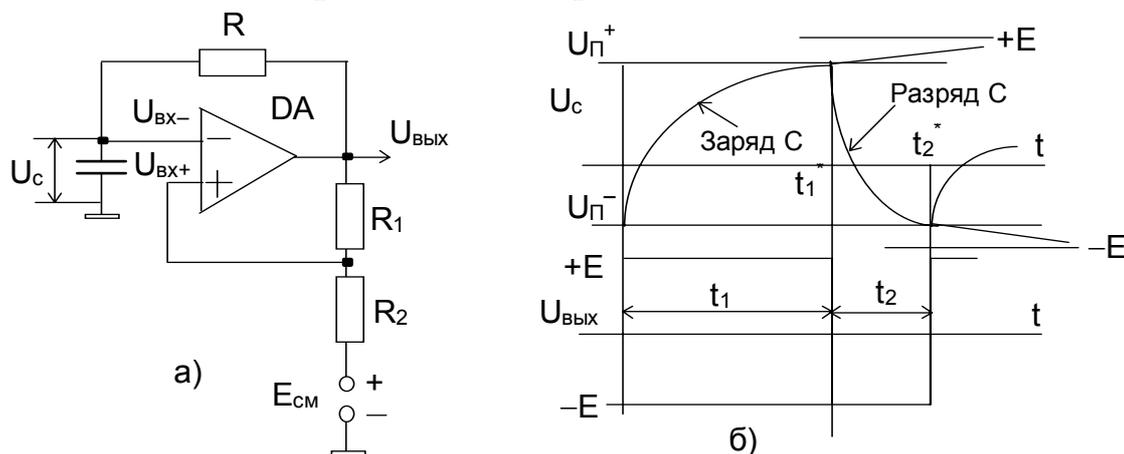


Рис. 98

В данном мультивибраторе имеется две петли обратной связи – положительная (ПОС) и отрицательная (ООС). Моменты переключения t_1^* и t_2^* мультивибратора из одного состояния в другое определяются сравнением $U_c(t) = U_{ВХ-}$ с пороговыми уровнями $U_{П+}$ и $U_{П-}$, задаваемыми как части $U_{ВЫХ}$ с делителя R_1, R_2 .

В течении интервала t_1 напряжение на неинвертирующем входе ОУ $U_{П+} = \frac{E - E_{СМ}}{R_1 + R_2} R_2 + E_{СМ} > U_c(t)$, за счет чего на выходе ОУ $U_{ВЫХ} = +E$.

Конденсатор С заряжается по цепи: $r_{ВЫХ ОУ} \rightarrow R \rightarrow C \rightarrow \perp$, с постоянной времени $\tau_{СЗ} = C(R + r_{ВЫХ ОУ}) \approx CR$, поскольку $r_{ВЫХ ОУ} \ll R$. Напряжение $U_c(t)$ стремится в пределе к $+E$. При достижении равенства $U_c(t) = U_{П+}$, в момент t_1^* ОУ переходит в активный режим ($K_0 \gg 1$), замыкается петля ПОС, возникает регенеративный процесс, который заканчивается переключением ОУ в состояние $U_{ВЫХ} = -E$.

Длительность t_1 определяется по формуле:
$$t_1 = CR \ln \frac{E + |U_{П-}|}{E - U_{П+}}$$

Во время второй стадии t_2 квазиравновесия напряжение на инвертирующем входе

$$U_C(t) = U_{BX}^- > U_{\Pi}^- = \frac{-E - E_{CM}}{R_1 + R_2} R_2 + E_{CM} .$$

При этом на выходе ОУ напряжение $U_{ВЫХ} = -E$. Конденсатор C перезаряжается до тех пор, пока $U_C(t) > U_{\Pi}^-$. В момент t_2^* $U_{ВХ}^-(t) = U_{\Pi}^-$, возникает второй регенеративный процесс, поскольку ОУ переходит в активный режим ($K_0 \gg 1$). Восстановление усилительных свойств ОУ приводит к его переключению вновь в состояние, когда на выходе $U_{ВЫХ} = +E$.

Длительность $t_2 = CR \ln \frac{|-E| + U_{\Pi}^+}{|-E| - |U_{\Pi}^-|}$, период переключения ОУ $T_i = t_1 + t_2$.

При $E_{CM} = 0$ $|U_{\Pi}^-| = U_{\Pi}^+ = \frac{ER}{R_1 + R_2}$, $t_1 = t_2 = CR \ln \left(1 + \frac{R_2}{R_1} \right)$. АМВ генерирует импульсы со скважностью $Q = T_i / t_i = 2$.

Регулировку периода целесообразно осуществлять с помощью резистора R , изменяя постоянную времени перезаряда $\tau = RC$. Скважность Q импульсной последовательности при этом меняться не будет. Возможны регулировки длительностей t_1 и t_2 изменением емкости C и смещения E_{CM} .

6.2.4. Ждущий мультивибратор на операционном усилителе.

Принципиальная схема ждущего мультивибратора на ОУ и временные диаграммы, поясняющие его работу, показаны на рис. 99, а, б.

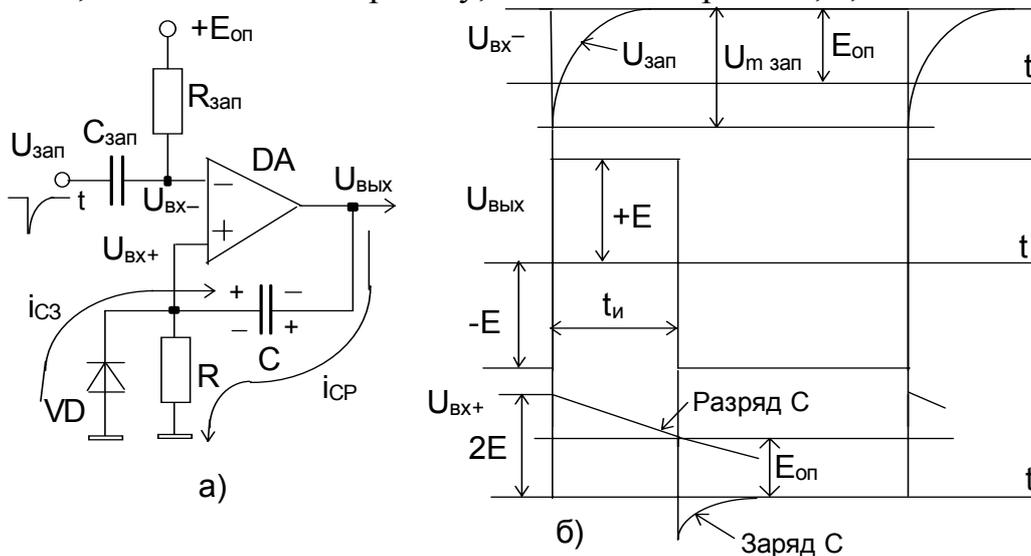


Рис. 99

В исходном заторможенном состоянии напряжение на выходе операционного усилителя DA $U_{ВЫХ} = -E$, так как $U_{ВЫХ} \approx 0$ (конденсатор C представляет собой разрыв по постоянному току), а $U_{ВХ}^- = E_{оп}$, т. е. $U_{ВХ}^- > U_{ВХ}^+$.

Запуск ЖМВ производится короткими импульсами отрицательной полярности, амплитуда которых должна превышать $E_{ОП}$ ($|U_{мзАП}| > E_{ОП}$). В момент запуска $U_{ВХ}^- < U_{ВХ}^+$, ОУ переходит в активный режим ($K_0 \gg 1$) и развивается регенеративный процесс перехода ОУ в состояние, при котором $U_{ВЫХ} = +E$ ($U_{ВЫХ} \uparrow \rightarrow U_{ВХ}^+ \uparrow \rightarrow U_{ВЫХ} \uparrow$). Скачок напряжения на выходе $\Delta U = 2E$ передается через конденсатор C без искажений на неинвертирующий вход и, поскольку при этом оказывается $U_{ВХ}^+(t) > E_{ОП}$, ЖМВ переходит в состояние квазиравновесия.

На стадии квазиравновесия конденсатор стремится перезарядиться (от $-E$ до $+E$) с постоянной времени $\tau_{CP} = C(R + r_{ВЫХ ОУ}) \approx CR$ по цепи: $r_{ВЫХ ОУ} \rightarrow C \rightarrow R \rightarrow \perp$. По мере перезаряда конденсатора $U_{ВХ}^+(t) = U_R(t)$ уменьшается в связи с уменьшением тока $i_{CP}(t) \downarrow$ и при достижении уровня $E_{ОП}$ происходит второе опрокидывание ЖМВ, который переходит снова в исходное состояние ($U_{ВЫХ} = -E$).

На стадии восстановления происходит заряд конденсатора по цепи $\perp \rightarrow VD \rightarrow C \rightarrow r_{ВЫХ ОУ} \rightarrow \perp$ с постоянной времени $\tau_{CЗ} \approx Cr_D$, причем $\tau_{CP} > \tau_{CЗ}$.

Длительность стадии квазиравновесия $t_{И} = \tau_{CP} \ln \frac{2E}{E_{ОП}}$, а длительность

стадии восстановления $t_{ВОССТ} = 3\tau_{CЗ} \approx 3Cr_D$.

6.2.5. Генераторы импульсов на специализированных микросхемах

Некоторые серии микросхем включают мультивибраторы с внешними времязадающими элементами. В частности, серия К155 содержит ждущие МВ К155АГ1 и К155АГ3.

Рассмотрим схему включения К155АГ1. Ждущий МТВ К155АГ1 имеет прямой и инверсный выходы (рис. 100). Запуск может быть по одному из трех входов в соответствии с табл. 41.

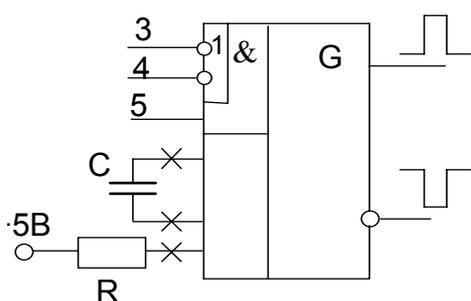


Рис. 100

Таблица 41

Запуск		
Вход 3	Вход 4	Вход 5
0	*	0 → 1
*	0	0 → 1
1 → *	1 → 0	1
1 → 0	1 → *	1

Длительность импульса $t_{И} = 0,7RC$ определяется значением внешних навесных элементов R и C . При $R \approx 2-40$ кОм, $t_{ВОССТ} = 0,1t_{И}$.

Схема К155АГ3 содержит два ждущих МВ со стробированием по входу.

6.2.6. Устройства на основе триггеров Шмитта

Отличительной особенностью триггеров Шмитта (ТШ) является наличие *гистерезиса* в передаточной характеристике (рис. 101). Переход из одного устойчивого состояния в другое в таких триггерах происходит при различных уровнях входного сигнала. Различают *инвертирующие* и *неинвертирующие* триггеры Шмитта. В инвертирующих ТШ включенному состоянию триггера соответствует низкий уровень U_0 , а выключенному – U_1 (рис.101, а); в неинвертирующих – наоборот, включенному состоянию триггера соответствует высокий уровень U_1 , а выключенному – U_0 (рис.101, б)

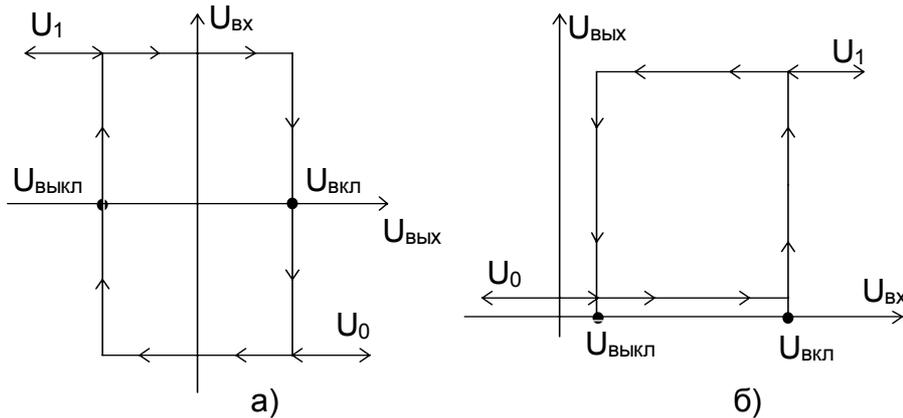


Рис. 101

Триггеры Шмитта с характеристиками, приведенными на рис. 101, могут быть реализованы, соответственно, на операционном усилителе (рис. 102, а) или на основе двух компараторов уровней и бистабильной ячейки памяти (рис. 102, б).

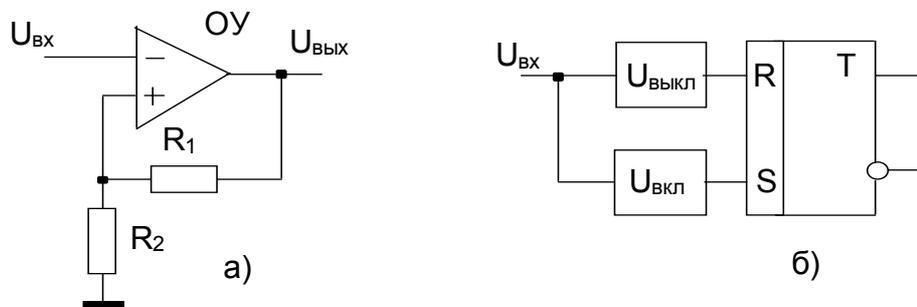


Рис. 102

В схеме ТШ на ОУ введена положительная обратная связь (через делитель R_1, R_2), действующая только в момент переключения ОУ, когда он работает в усилительном режиме. Коэффициент передачи делителя $\beta = R_1 / (R_1 + R_2)$ определяет уровни напряжений переключения на прямом входе

$$U_+ = U_{ВЫХ} \beta = U_{ВЫХ} \frac{R_1}{R_1 + R_2}.$$

При этом включение и выключение ТШ происходит при $\Delta U = U_{ВХ} - U_+ = 0$ на пороговых уровнях

$$U_{ВКЛ} = +E\beta = ER_1 / (R_1 + R_2),$$

$$U_{ВЫКЛ} = -E\beta = -ER_1 / (R_1 + R_2).$$

Напряжение гистерезиса равно разности порогов переключения:

$$U_{Г} = U_{ВКЛ} - U_{ВЫКЛ} = 2E\beta.$$

Триггер Шмитта позволяет формировать прямоугольные импульсы из сигналов произвольной формы, эффективно выделять полезный сигнал при высоком уровне помех за счет гистерезиса переключения (рис. 103, а).

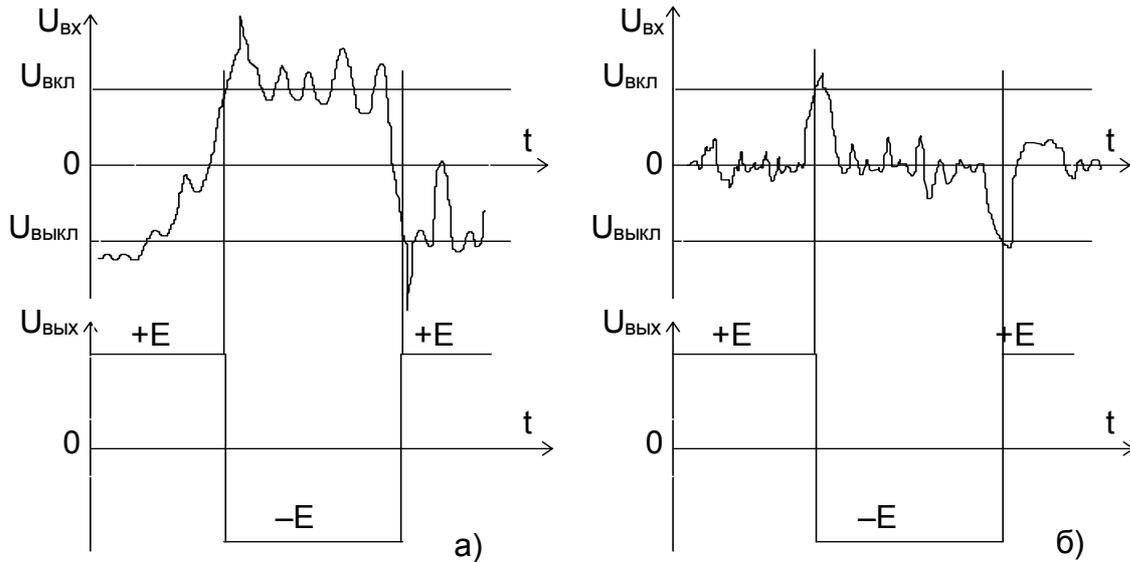


Рис. 103

На рис. 103, б показано использование ТШ для формирования импульсов прямоугольной формы из коротких разнополярных импульсов, которые получаются после фильтрации низкочастотных составляющих. Триггер Шмитта восстанавливает на выходе импульсы прямоугольной формы и фильтрует помехи.

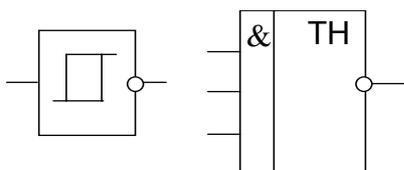


Рис. 104

Вследствие широкого применения триггеры выпускаются промышленностью в виде законченных узлов (от двух до шести триггеров в корпусе ИС) в ряде серий цифровых интегральных схем (триггеры К155ТЛ1, К155ТЛ2, К155ТЛ4). На рис. 104 приведены УГО инвертирующих триггеров.

6.3. Генераторы линейно-изменяющегося напряжения

6.3.1. Параметры линейно-изменяющегося напряжения

Линейно-изменяющееся напряжение (ЛИН) описывается формулой $U_p(t) = U_p t$, где $U_p = const$ (рис. 105).

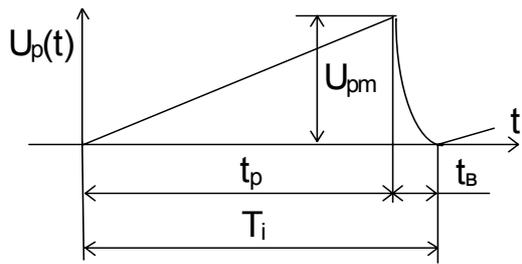


Рис. 105

К числу **параметров ЛИН** относятся:

- 1) t_p – длительность развертки (прямой ход луча), от долей мкс до десятков секунд;
- 2) t_B – длительность обратного хода ЛИН (восстановление), $t_B = (1 \div 50)\%$ от t_p ;
- 3) U_{pm} – амплитуда ЛИН, от единиц до тысяч вольт;
- 4) скорость изменения ЛИН:

$$v = U_p^*(t) = \frac{dU_p(t)}{dt}, \text{ от } 10^{12} \text{ до } 10^{-2} \text{ В/с};$$

5) коэффициент использования источника питания E_{II} : $\xi = \frac{U_{pm}}{E_{II}}$;

6) коэффициент нелинейности ЛИН:

$$\varepsilon = \frac{U_{pn} - U_{pk}}{U_{pn}} = 1 - \frac{U_{pk}}{U_{pn}} = \frac{U_p^*(0) - U_p^*(t_p)}{U_p^*(0)},$$

коэффициент линейности ЛИН: $\alpha = 1 - \varepsilon$.

На практике стремятся реализовать $\xi \rightarrow 1$, а $\varepsilon \rightarrow 0$, хотя для аналоговых ГЛИН эти требования противоречивы.

Основные применения ЛИН:

- ⇒ развертка электронного луча в видеоиндикаторах;
- ⇒ измерение времени и временная модуляция параметров сигналов.

Способы получения ЛИН.

Практически все способы получения ЛИН основаны на заряде или разряде конденсатора током, близким к постоянному $i_c(t) = I_0 = const$:

$$U_C(t) = \frac{1}{C_0} \int i_C(t) dt = \frac{1}{C_0} \int I_0 dt = \frac{I_0}{C} t = U_0(t).$$

В состав устройства, формирующего ЛИН, должны входить интегрирующие звенья и узлы токостабилизации.

Аналоговые генераторы ЛИН реализуются на основе:

- ◇ времязадающей RC-цепи заряда или разряда конденсатора и использование начального участка экспоненциальной функции;
- ◇ нелинейного токостабилизирующего элемента;
- ◇ компенсационных генераторов ЛИН с отрицательной или положительной обратной связью.

Аналоговые ГЛИН обладают :

- ◆ невысокой точностью и стабильностью ЛИН из-за наличия нестабильных дискретных элементов (конденсаторы, транзисторы), особенно при больших значениях длительности развертки t_p ;
- ◆ значительным временем восстановления t_e (до 50% от длительности развертки);
- ◆ трудностью реализации в микросхемном исполнении конденсаторов.

6.3.2. Цифровые ГЛИН

Цифровые ГЛИН свободны от этих недостатков. Они основаны на счете импульсов стабильной частоты $F_0=1/T_0$ и цифро-аналоговом преобразовании результата счета в линейно-ступенчатое напряжение (ЛСН). При уменьшении единичного приращения напряжения ΔU ЛСН приближается к ЛИН.

Число, выражаемое n -разрядным двоичным кодом, можно представить в виде:

$$N_X = \{Q_{n-1}Q_{n-2}\dots Q_2Q_1Q_0\} = Q_{n-1}2^{n-1} + \dots + Q_12^1 + Q_0 = \sum_{i=0}^{n-1} Q_i 2^i, Q_i \in (0, 1)$$

При использовании n эталонов напряжений ($E_{\text{Э}} / 2^1, E_{\text{Э}} / 2^2, \dots, E_{\text{Э}} / 2^n$) напряжение на выходе цифроаналогового преобразователя (ЦАП) можно представить как сумму произведений этих эталонов на значения разрядов кода числа N_X :

$$U_N = \frac{E_{\text{Э}}}{2^1} N_{n-1} + \frac{E_{\text{Э}}}{2^2} N_{n-1} + \dots + \frac{E_{\text{Э}}}{2^n} N_{n-1} = \frac{E_{\text{Э}}}{2^n} N_X = \frac{E_{\text{Э}}}{N_{\text{max}}} N_X = m_U N_X,$$

$$\text{где } N_{\text{max}} = 2^n, m_U = \frac{E_{\text{Э}}}{2^n} = \Delta U.$$

Таким образом, в составе генератора ЛСН (рис. 106, а) должны быть генератор счетных импульсов (ГСИ) стабильной частоты, счетчик импульсов (СТ2) для получения N_X , источник эталонного напряжения $E_{\text{Э}}$, разрядная сетка (РС), суммирующая разрядные напряжения с соответствующими весовыми коэффициентами и электронные ключи (ЭК), коммутирующие подачу эталонных напряжений в зависимости от двоичного кода числа N_X .

По мере увеличения содержимого счетчика СТ2 N_X (пропорционально во времени, если частота импульсов ГСИ стабильна) напряжение на выходе разрядной сетки $U_{\text{ВЫХ}}$ будет представлять собой пропорционально растущее во времени линейно-ступенчатое напряжение (рис. 106, б).

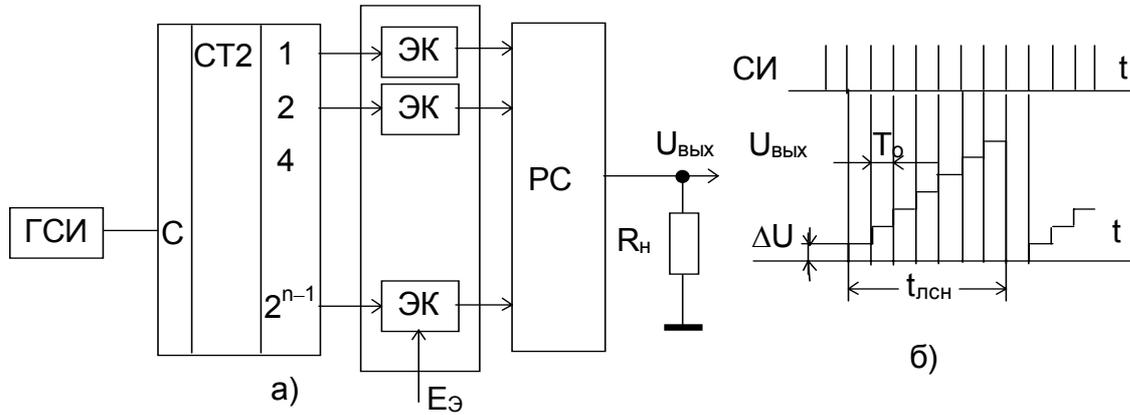


Рис. 106

Параметры ЛСН (рис. 106, б) определяются следующим образом:

- 1) длительность развертки $t_{ЛСН} = N_{max} T_0 = 2^n T_0 = t_P$, может быть “неограниченно” большой и зависит только от разрядности счетчика СТ2;
- 2) время восстановления $t_B \ll t_{ЛСН}$ и определяется переходным процессом в электронных ключах и счетчике;
- 3) амплитуда ЛСН $U_{ЛСН}$ определяется величиной E_3 и периодом счетных импульсов, т.е. значением m_U : $U_{ЛСН max} = m_U N_{Xmax}$;
- 4) точность генератора ЛСН определяется стабильностью ГСИ и погрешностями, вносимыми элементами ЦАП (E_3 , РС, ЭК).

6.4. Цифро-аналоговые и аналого-цифровые преобразователи

Цифро-аналоговые и аналого-цифровые преобразователи (ЦАП и АЦП) используются соответственно для преобразования цифрового кода в мгновенное значение аналогового сигнала и обратно.

6.4.1. Цифро-аналоговые преобразователи

Структура ЦАП определяется видом разрядной сетки. Существует два вида разрядных сеток, на которые могут быть поданы коды в общем случае от счетчиков, регистров и других источников цифровой информации.

Разрядная сетка в виде матрицы двоично-взвешенных резисторов

Цифро-аналоговый преобразователь с использованием разрядной сетки в виде матрицы двоично-взвешенных резисторов показан на рис. 107.

Напряжение на выходе операционного усилителя, включенного по схеме усилителя-инвертора, определяется как сумма парциальных напряжений

$$|U_{Вых}| = \left| \sum_{i=0}^{n-1} k_i E_3 \right| = \left| -E_3 \left(\frac{R/2}{R \cdot 2^{n-1}} Q_0 + \frac{R/2}{R \cdot 2^{n-2}} Q_1 + \dots + \frac{R/2}{R \cdot 2^0} Q_{n-1} \right) \right| =$$

$$= \frac{E_3}{2^n} (Q_{n-1} 2^{n-1} + \dots + 2^1 Q_1 + 2^0 Q_0) = \frac{E}{N_{max}} N_X$$

где $k_i = \frac{R_{oc}}{R \cdot 2^i}$, $N_{max} = 2^n$, $N_X = Q_0 2^0 + Q_1 2^1 + Q_2 2^2 + \dots + Q_{n-1} 2^{n-1}$

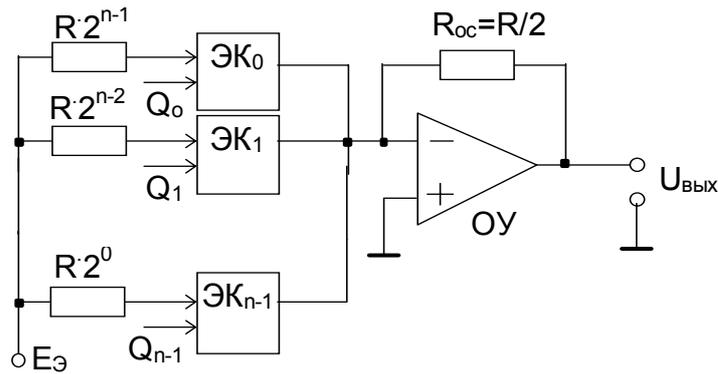


Рис. 107

Недостаток такого ЦАП – большой диапазон номиналов весовых разрядов (при $n=10$ от $1R$ до $512R$), что затрудняет их технологическое исполнение. Серийный ЦАП такого вида – К594ПА1 ($n=12$, время преобразования $t_{np}=3.5$ мкс).

Разрядная сетка в виде матрицы типа R-2R.

Цифро-аналоговый преобразователь с использованием матрицы $R-2R$ показан на рис. 108.

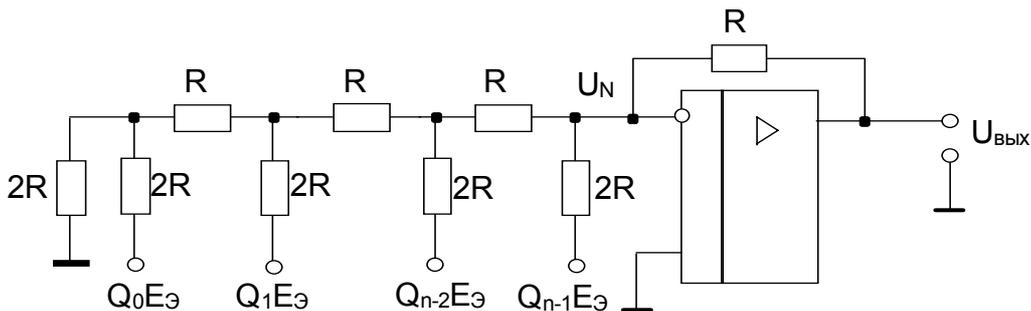


Рис. 108

Выходное сопротивление матрицы (сетки) $R-2R$ не зависит от кода $\{Q_i\}$, числа разрядов (при наличии идеальных ЭК и $E_{\text{Э}}$) и равно R . Таким образом, свойства разрядной сетки $R-2R$ не зависят от числа секций, что позволяет вводить дополнительные секции $R-2R$.

Определим парциальные напряжения, вносимые подключаемыми разрядами сетки.

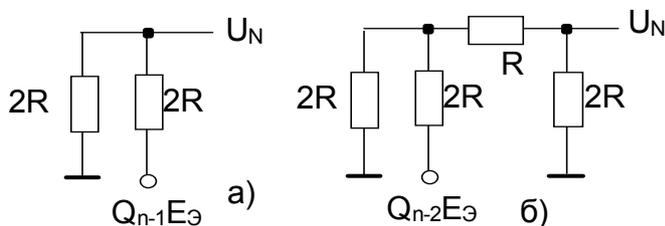


Рис. 109

В случае подключения только одного старшего разряда с номером $n-1$ (рис. 109, а) $Q_{n-1}=1$, $Q_i=0$, $i \neq n-1$, $U_N = E_{\text{Э}}/2$.

При подключении разряда с номером $n-2$ (рис. 109, б) $Q_{n-2}=1$, $Q_i=0$, $i \neq n-2$, $U_N = E_{\text{Э}}/4$.

В общем случае

$$U_N^S = \frac{E_{\Delta}}{2} Q_{n-1} + \frac{E_{\Delta}}{4} Q_{n-2} + \dots + \frac{E_{\Delta}}{2^n} Q_0 = \frac{E_{\Delta}}{2^n} (Q_{n-1} 2^{n-1} + Q_{n-2} 2^{n-2} + \dots + Q_1 2^1 + Q_0 2^0) =$$

$$= \frac{E_{\Delta}}{N_{\max}} N_X, \quad U_N^S = m_U N_X,$$

где $m_U = \frac{E_{\Delta}}{N_{\max}}$ - коэффициент преобразования.

Время установления кода определяется переходным процессом в электронных ключах ЭК и быстродействием триггеров Q_i . Точность любого ЦАП определяется погрешностью резисторов сетки, погрешностью и стабильностью E_{Δ} и параметрами ЭК (сопротивлением в открытом $R_{\text{ПР}}$ и закрытом $R_{\text{ОБР}}$ состояниях). Число разрядов ЦАП $n=10-12$, при больших значениях n увеличиваются требования к точности резисторов ($\Delta R/R \leq 2^{-n}$, т.е. при $n=10$ необходимо выполнить резисторы с погрешностью $\Delta R/R < 0.1\%$).

6.4.2. Аналого-цифровые преобразователи

АЦП основаны на сопоставлении аналоговой величины A с набором дискретных эталонных значений в заданной сетке уровней квантования и определение номера N ближайшего к A уровню в момент измерения (рис. 110).

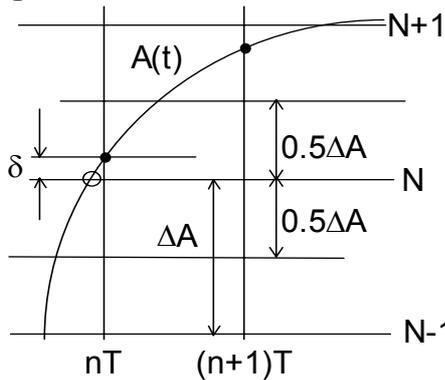


Рис. 110

В момент измерения:
 $A = m_a N \pm \delta = m_a N \pm \eta \Delta A, \quad \eta = 0 \div 0.5,$
 $N = \frac{A \text{ m} \delta}{m_a} = m_N (A \text{ m} \delta) = m_N (A \text{ m} \eta \Delta A) \rightarrow$
 $\rightarrow \{Q_{n-1} Q_{n-2} \dots Q_1 Q_0\}_N, \quad Q \in [0, 1],$

где ΔA - шаг квантования по уровню, m_a, m_N - масштабные коэффициенты, $\delta = \pm \eta \Delta A$ - погрешность квантования по уровню.

Разновидности АЦП

В технических применениях используются:

- ⇒ преобразователи напряжения в код $U \rightarrow N$;
- ⇒ преобразователи тока в код $I \rightarrow N$;
- ⇒ преобразователи угла в код $\alpha \rightarrow N$;
- ⇒ преобразователи расстояния в код $L \rightarrow N$;
- ⇒ преобразователи времени в код $t \rightarrow N$;
- ⇒ преобразователи частоты в код $f \rightarrow N$ и другие.

Все многообразие схем АЦП напряжения основано на трех методах:

- АЦП последовательного счета единичных приращений;
- АЦП поразрядного уравнивания;
- АЦП параллельного действия (считывания).

6.4.2.1. АЦП последовательного счета

Структура АЦП последовательного счета и временные диаграммы, поясняющие его работу, показаны на рис. 111, а,б. В основе преобразование “аналог-код” лежит использование генератора ЛСН. Эталонном служит напряжение ΔU , пропорциональное стабильному периоду тактирующих импульсов T_0

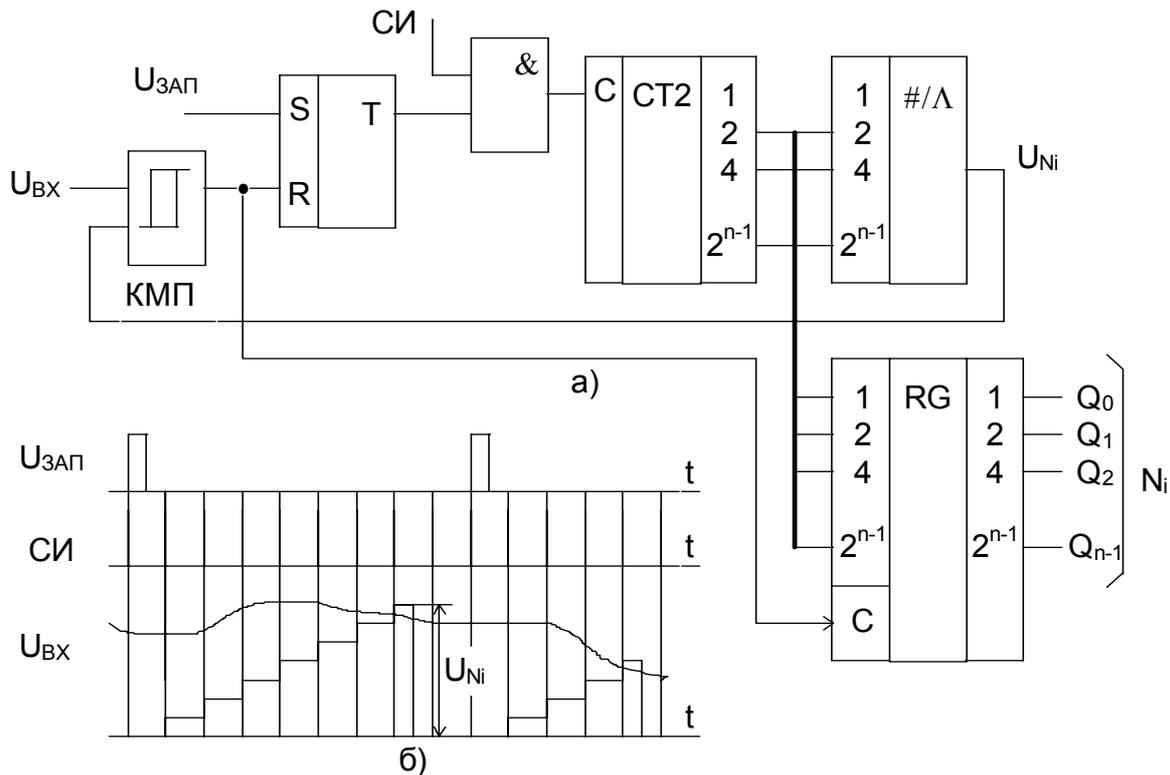


Рис. 111

Цикл аналого-цифрового преобразования начинается в момент действия импульса запуска $U_{ЗАП}$, который переключает управляющий триггер в единичное состояние. С целью уменьшения погрешности преобразования импульс запуска должен быть синхронизирован с частотой счетной последовательности, как показано на рис. 111, б.

Импульсы СИ, поступающие через открытую схему И, подсчитываются счетчиком СТ2. Текущее значения содержимого счетчика N_i преобразуется в линейно-ступенчатое напряжение U_{Ni} . В момент сравнения в компараторе КМП входного напряжения и ЛСН вырабатывается импульс, устанавливающий триггер в нулевое состояние. При этом запирается схема И и прекраща-

ется поступление через нее импульсов в счетчик, а значение кода N_i на момент равенства напряжений фиксируется в регистре-защелке RG:

$$U_{BX} = U_{N_i} = E_{\mathcal{E}} \frac{N_i}{N_{\max}}, \text{ т.е. } N_i = \frac{N_{\max}}{E_{\mathcal{E}}} U_{BX} = m_N U_{BX}, N_{\max} = \frac{U_{BX \max}}{\Delta U} = 2^n,$$

где ΔU – шаг квантования (эталон). Таким образом, в АЦП последовательного счета используется один эталон $\Delta U = E_{\mathcal{E}}/N_{\max} = E_{\mathcal{E}}/2^n$.

Быстродействие АЦП определяется максимальной длительностью преобразования $T_{IP \max} = N_{\max} T_0 = \frac{U_{BX \max}}{\Delta U} T_0 = 2^n T_0$ (до момента сравнения на-

пряжений на входах компаратора проходит 2^n тактов СИ). Максимальная частота импульсов запуска в предельном случае, т.е. частота преобразования

$$F_{\max} < \frac{1}{T_{IP \max}} = (N_{\max} T_0)^{-1} = \frac{F_0}{2^n}, \text{ где } F_0 \text{ – частота СИ. АЦП последовательного счета обладают самым низким быстродействием.}$$

Точность преобразования определяется погрешностями СИ, ЦАП и компаратора: $\gamma_{АЦП} = \gamma_{СИ} + \gamma_{ЦАП} + \gamma_{КМП}$. При $n = 10-11$, $\gamma_{АЦП} \approx 0.5-1\%$, $F_{\max} < 1-2$ кГц.

6.4.2.2. АЦП поразрядного уравнивания

В АЦП данного типа используется n эталонов напряжений ($E_{\mathcal{E}}/2, E_{\mathcal{E}}/4, \dots, E_{\mathcal{E}}/2^n$), которые формируются кодом на выходе n -разрядного регистра сдвига. Уравнивание входного напряжения происходит за n тактов, в течение которых ЦАП вырабатывает сумму эталонов, соизмеримую с уровнем преобразуемого напряжения с точностью до шага квантования ΔU .

Функциональная схема АЦП поразрядного уравнивания и временные диаграммы, иллюстрирующие метод преобразования, показаны на рис. 112, а, б.

В первом такте после запуска на выходе ЦАП устанавливается $U_{ЦАП} = E_{\mathcal{E}}/2$, поскольку в старший разряд Q_{n-1} регистра последовательных приближений (РПП) по фронту ($0 \rightarrow 1$) первого тактового импульса записывается 1. На выходе компаратора КМП при этом появляется 0, так как напряжение на входе АЦП U_{BX} меньше напряжения с выхода ЦАП. С выхода компаратора сигнал 0 проходит на вход D РПП и по спаду ($1 \rightarrow 0$) тактовых импульсов переписывается с этого входа в старший разряд РПП. При этом напряжение $U_{ЦАП}$ уменьшается до нуля, и на выходе компаратора вновь появляется сигнал 1. При появлении второго тактового импульса это значение записывается во второй по старшинству разряд Q_{n-2} РПП. На выходе ЦАП при этом появляется напряжение $U_{ЦАП} = E_{\mathcal{E}}/4$. Так как $U_{BX} > U_{ЦАП}$, то на выходе компаратора сохранится 1, и это значение по спаду ТИ присваивается разряду Q_{n-2} .

Таким образом осуществляется поразрядное уравнивание кода РПП с уровнем входного сигнала. После n тактовых импульсов (n – число разрядов

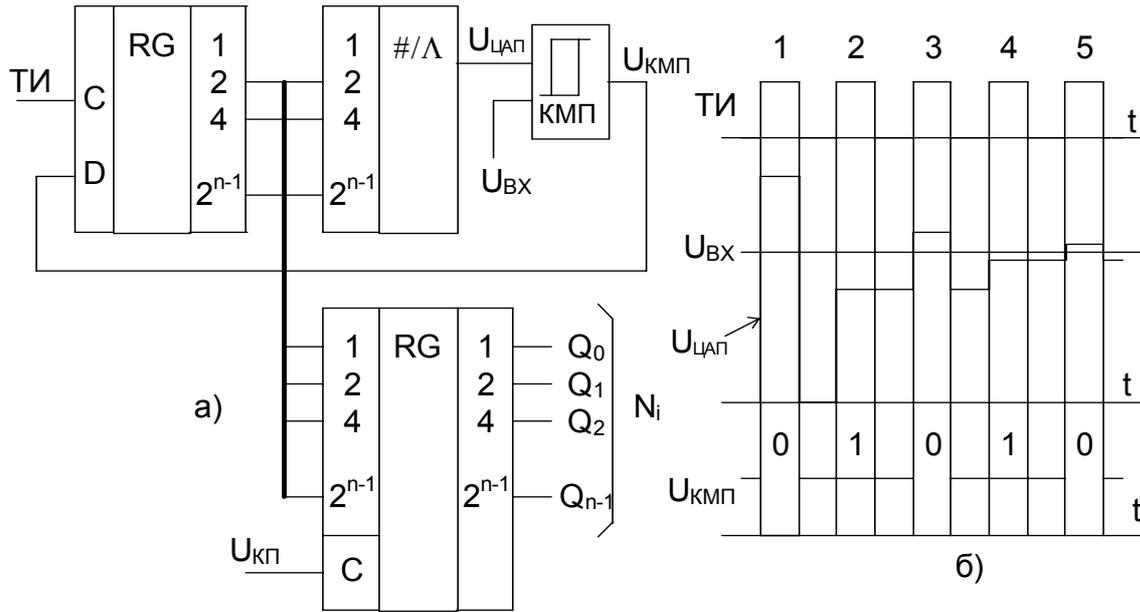


Рис. 112

РПП) произойдет запоминание самого младшего разряда Q_0 РПП и содержимое этого регистра переписывается в выходной регистр RG сигналом “Конец преобразования” $U_{КП}$ с выхода РПП.

Быстродействие АЦП с РПП значительно выше, чем у АЦП с последовательным счетом, так как $T_{IP \max} = nT_0 = T_0 \log_2 N_{\max} < T_0 N_{\max}$. Точность данного АЦП определяется погрешностями $\gamma_{АЦП} = \gamma_{СИ} + \gamma_{ЦАП} + \gamma_{КМП}$ и соизмерима с точностью АЦП последовательного счета.

6.4.2.3. АЦП параллельного действия

В n -разрядном АЦП параллельного действия используется $2^n - 1$ эталонов: $\Delta U = E_{\Delta} / 2^n, 2 \cdot \Delta U, 3 \cdot \Delta U, \dots, (2^n - 1) \cdot \Delta U$, формируемых либо от $2^n - 1$ источников эталонных напряжений (ИЭН), либо от одного ИЭН, а также $2^n - 1$ компараторов и шифратор кода (CD).

Структурная схема АЦП параллельного действия показана на рис. 113, где ИЭН – источник эталонных напряжений, КМП₁ - КМП_М – компараторы напряжения ($M = 2^n - 1$). Шифратор преобразует единичный код с выходов компараторов (типа 00...01...111) в двоичный код, который сохраняется в выходном регистре RG.

При некотором значении входного напряжения $(2^n - 1) \Delta U \geq U_{ВХ} \geq \Delta U$ на выходах компараторов (при условии подачи разрешающего сигнала $U_{ТИ} = 1$) появляются сигналы: 1 для компараторов КМП₁...КМП_i; 0 для компараторов КМП_{i+1}...КМП_М.

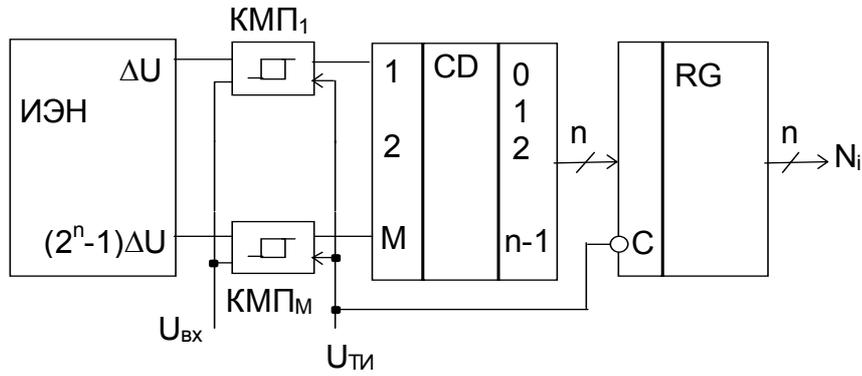


Рис. 113

Сигналы с компараторов поступают в шифратор, с выходов которого n -разрядный двоичный код $N_i=i$ переписывается в выходной регистр при условии $U_{ти}=0$. Таким образом, частота кодовых отсчетов на выходе АЦП равна тактовой частоте сигнала $U_{ти}$.

АЦП параллельного действия обладают наивысшим быстродействием (преобразование совершается за один период тактовых импульсов) и наилучшими точностными характеристиками. Сложность АЦП - наивысшая среди трех типов АЦП, что объясняется наличием значительного числа компараторов. Сравнительно небольшое число разрядов в параллельных АЦП ($n=6-8$) не ограничивает области их применения, поскольку имеется возможность наращивания разрядности параллельным включением таких преобразователей.

Идея $\text{òäåçäüâíâí òäââíââøèââíèÿ$ ÀÖÏ:

- $\text{òäåçäüâíâí òäââíââøèââíèÿ}$
- Ê572ÏÄ1 ($n=12, \text{ò}_{\text{ID}}=120 \text{ ìñ}$),
- Ê1113ÏÄ1 ($n=10, \text{ò}_{\text{ID}}=30 \text{ ìñ}$);
- $\text{ìäðäèèâèüíâí ääèñðäèÿ}$:
- Ê1107ÏÄ1 ($n=6, \text{ò}_{\text{ID}}=0.1 \text{ ìñ}$).

6.5. Контрольные вопросы и задания для самоподготовки.

1. Нарисовать схему элемента задержки с интегрирующей цепью и временные диаграммы сигналов, поясняющие работу. Определить и обосновать способ изменения задержки и нарисовать график зависимости величины задержки от величины регулирующего параметра.
2. Выполнить задание по п.1 для элемента задержки с дифференцирующей цепью.
3. Привести схему формирователя импульсов на базе элементов задержки, сопутствующие временные диаграммы сигналов и пояснить работу. Предложить и обосновать способ изменения длительности импульсов.
4. Нарисовать схему автоколебательного мультивибратора на элементах ТТЛ, сопутствующие временные диаграммы сигналов. Показать на временных диаграммах влияние время-задающих элементов на работу АМВ и предложить механизмы регулировки длительности стадий квазиравновесия
5. Выполнить на задание по п.4 для ждущего мультивибратора на элементах ТТЛ.
6. Выполнить задания по п.4 и 5 для АМВ и ЖМВ на операционных усилителях.
7. Привести примеры мультивибраторов на специализированных микросхемах.
8. Пояснить принцип формирования импульсов на базе триггеров Шмитта.
9. Пояснить принцип цифрового генератора линейно-изменяющегося напряжения. Обосновать выбор параметров ЛИН (ЛСН). Нарисовать схему ГЛСН и временные диаграммы сигналов.
10. Дать определение цифро-аналоговому преобразователю. Пояснить состав типового ЦАП и природу его погрешностей.
11. Привести схему и провести анализ ЦАП с матрицей двоично-весовых резисторов.
12. Привести схему и провести анализ ЦАП с разрядной сеткой из сопротивлений двух номиналов.
13. Привести методы аналого-цифрового преобразования и дать их сравнительную оценку.
14. Провести анализ АЦП последовательного счета. Оценить быстродействие и источники погрешностей.
15. Провести анализ АЦП считывания. Сравнить его с АЦП последовательного счета и поразрядного уравнивания по параметрам быстродействия и точности.
16. Пояснить принцип преобразования уровня $U = 0,5; 1; 1,5; 2; 2,5; 3; 4; 5В$ в двоичный код, если разрешающая способность АЦП $\Delta U = 5мВ$.

Глава 7. Микроэлектронная элементная база цифровых устройств

7.1. Классификация цифровых микросхем

Цифровые микросхемы (ЦМС) предназначены для обработки дискретной информации, кодируемой логической “единицей” и “нулем”.

Существуют системы *потенциальных* и *импульсных* ЦМС. В современной технике доминирует система потенциальных элементов (СПЭ), в которой логическим значением “1” и “0” соответствуют два существенно различных уровня напряжения U^1 и U^0 . СПЭ работают в *положительной* ($U^0 \rightarrow 0, U^1 \rightarrow E > 0$) и *отрицательной* ($U^0 \rightarrow 0, U^1 \rightarrow E < 0$) логиках.

Набор СПЭ должен обеспечивать *функциональную полноту* (способность реализовать сколь угодно сложную цифровую структуру) и *совместимость* элементов (без применения согласующих звеньев).

По типу *базового элемента* ЦМС классифицируются на:

- транзисторные логические схемы с непосредственной связью (НСТЛ, англ. DCTL);
- транзисторные логические схемы с резистивной связью (РТЛ, англ. RTL);
- транзисторные логические схемы с резистивно - емкостной связью (РЕТЛ, англ. RCTL);
- схемы с диодно - транзисторной логикой (ДТЛ, англ. DTL);
- схемы с транзисторно - транзисторной логикой (ТТЛ, англ. TTL);
- схемы с эмиттерно - связанной логикой (ЭСЛ, англ. ECL);
- транзисторные схемы с непосредственной связью (НСТЛМ, англ. DCTLM) на структурах металл - диэлектрик (окисел)- полупроводник (МДП(МОП), англ. MOS);
- комплементарные (дополняющие) схемы на полевых транзисторах (КМДП (КМОП), англ. CMOS);
- ЦМС с двойной инжекционной логикой (И²Л, англ. I²L).

В современных цифровых устройствах наиболее распространены системы элементов ТТЛ, ЭСЛ и КМДП, которые и рассматриваются далее.

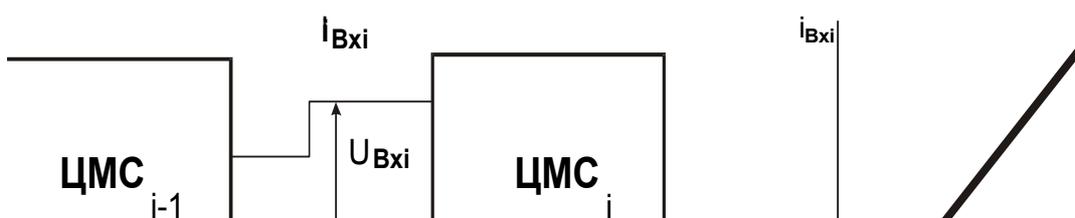
7.2. Характеристики и параметры ЦМС

7.2.1. Статические внешние характеристики ЦМС

1. Входная характеристика $i_{Вxi} = f(U_{Вxi}), i = \overline{1, M}$ (Рис.114, б)

Входная характеристика определяет ток ЦМС по i - му входу ($i = \overline{1, M}$), иначе, входное сопротивление. По этой характеристике находится так же пороговый уровень $U_{п}$ срабатывания микросхемы при непосредственной связи ее с выходом микросхемы- генератора.

Разброс входных токов ограничил применение системы НСТЛ и обусловил переход к системам РТЛ, РЕТЛ, а затем к ДТЛ и ТТЛ.



а) б)

Рис. 114

2. Выходная характеристика $i_{\text{ВЫХ}} = f(U_{\text{ВЫХ}})$ (Рис.115, б)

Выходные характеристики определяет токи нагрузки (входные токи нагружающих ЦМС) в открытом ($i_{\text{ВЫХ}}^{\text{от}} = f(U_{\text{ВЫХ}}^{\text{от}})$) и запертом ($i_{\text{ВЫХ}}^{\text{зап}} = f(U_{\text{ВЫХ}}^{\text{зап}})$) состояниях ЦМС.

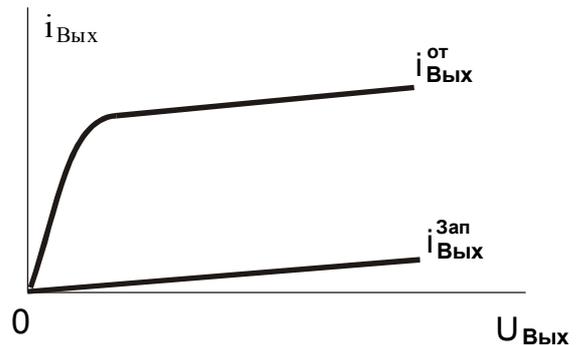
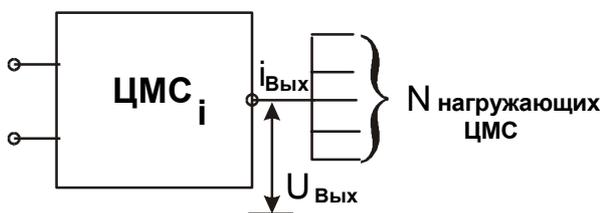


Рис. 115

3. Передаточная характеристика $U_{\text{ВЫХ}} = f(U_{\text{ВХ}})$ (Рис.116)

В передаточной характеристике для *инвертирующих* ЦМС (рис.116, а) и *неинвертирующих* ЦМС (рис.116, б) различают три состояния: I - $U_{\text{ВЫХ}} = U^0$ (уровень логического нуля), II - $U_{\text{ВЫХ}} = U^1$ (уровень логической единицы) и III - промежуточное состояние. Для надежной работы ЦМС необходимо выполнить условие: $U^0 < U_{\text{П}}^0$ и $U^1 > U_{\text{П}}^1$, где $U_{\text{П}}^0$ и $U_{\text{П}}^1$ - пороговые уровни переключения, с учетом технологического разброса передаточных характеристик. Зона неопределенности $\Delta U_{\text{П}} = U_{\text{П}}^1 - U_{\text{П}}^0$. Выходной логический перепад $U_{\text{П}} = U^1 - U^0$ должен превышать уровни статических помех.

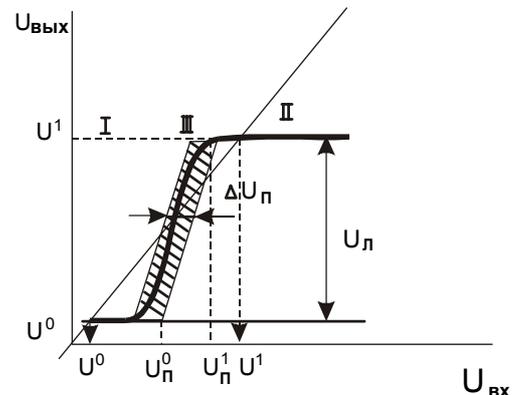
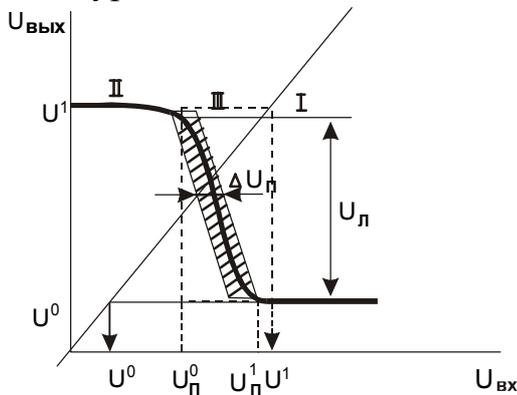


Рис.116

При последовательном соединении ЦМС общая передаточная характеристика более резко разделяет области I и II, приобретая вид гистерезиса (рис.117, а,б). При уровнях помех $U_{\text{П}}^+ > U_{\text{П}}^0 - U^0$ и $U_{\text{П}}^- > U^1 - U_{\text{П}}^1$ возможно ложное срабатывание ЦМС.



Рис.117

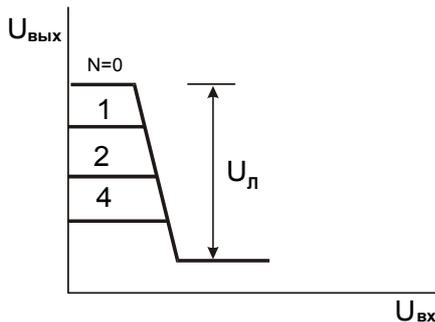
7.2.2. Параметры ЦМС

1. *Степень функциональной интеграции* $K_{И} = \lg N_{ЭЛ}$, где $N_{ЭЛ}$ - число элементов (типа И-НЕ, ИЛИ- НЕ и т.п.), расположенных в кристалле.

Для малых интегральных схем (МИС) $N_{ЭЛ} = 1-10$ и $K_{И} \approx 1$. К ним относятся простейшие логические элементы (И, ИЛИ, И - НЕ, ИЛИ - НЕ, И- ИЛИ- НЕ и т. п.). Средние интегральные схемы (СИС) содержат в составе $N_{ЭЛ} = 10 - 100$ и $K_{И} \approx 1 - 2$ (это сумматоры, счетчики, регистры). При $N_{ЭЛ} = 100 - 1000$ и более и $K_{И} \geq 2-3$ ЦМС являются большими интегральными схемами (БИС). К ним относятся арифметико - логические устройства, устройства памяти, управления и другие. Сверхбольшие интегральные схемы (СБИС) имеют $K_{И} > 4$ и являются основой микро ЭВМ.

2. *Коэффициент разветвления по выходу* N характеризует (нагрузочную способность ЦМС)

N определяет максимальное число микросхем, нагружающих данную без существенного искажения сигнала.

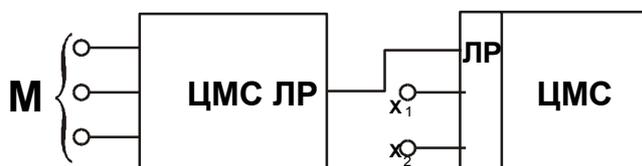


С ростом числа нагрузок N падает быстродействие ЦМС (возрастает влияние шунтирующих паразитных емкостей) и помехоустойчивость (уменьшается логический перепад $U_{\text{л}} = U^1 - U^0$) (Рис.118).

Рис.118

3. *Коэффициент объединения по входу* M (число входов ЦМС)

Обычно $M < 8$. Для увеличения числа входов без снижения быстродействия ЦМС используют логические расширители (ЛР) (Рис.119). По статистике фирмы ИВМ



для 40% ЦМС $M=2-3$, $M=3-4$ для 30% и $M > 10$ для 3%. В пределе M ограничивается числом выводов в микросхемах.

Рис.119

4. *Помехоустойчивость ЦМС* - способность сохранять работоспособность при воздействии внешних помех

Помехи могут быть статическими (длительно существующими) и динамическими (импульсными). Устойчивость ЦМС к воздействию импульсных помех выше, то есть статические помехи более опасны.

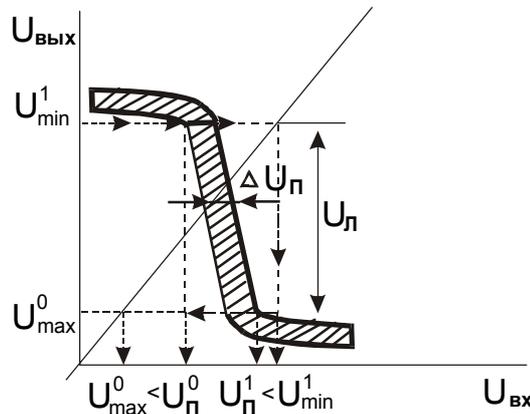


Рис. 120

Допустимый уровень по-мехи указан в справочниках. Более точное определение его производится по передаточной характеристике (Рис.120) с учетом технологического и температурного разброса (на рисунке заштрихованная область). Так, помехо-устойчивость к отпирающей по-мехе

$$\Delta U^+ = U^0_{\text{П}} - U^0_{\text{МАХ}} \text{ или } A^+ = \frac{\Delta U^+}{(U^1 - U^0)},$$

где $U_{\text{Л}} = U^1_{\text{МИН}} - U^0_{\text{МАХ}}$ - логический перепад.

Помехоустойчивость к запирающей помехе

$$\Delta U^- = U^1_{\text{МИН}} - U^1_{\text{П}} \text{ или } A^- = \frac{\Delta U^-}{(U^1 - U^0)}$$

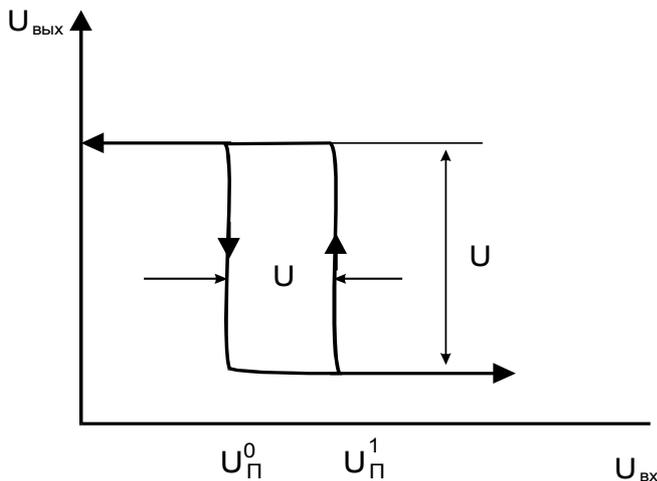


Рис. 121

ЦМС с гистерезисной передаточной характеристикой обладают повышенной помехоустойчивостью. В пределе при $U_{\Gamma} \approx U_{\text{Л}}$ помехоустойчивость $\Delta U^+_{\text{П}} \approx \Delta U^-_{\Gamma} \approx U_{\text{Л}}$ превышает вдвое величину $U_{\text{П}}$ в схемах без гистерезиса.

5. Быстродействие ЦМС

Быстродействие ЦМС определяется скоростями перехода логического элемента из состояния "0" в состояние "1". Количественно оно характеризуется сред-

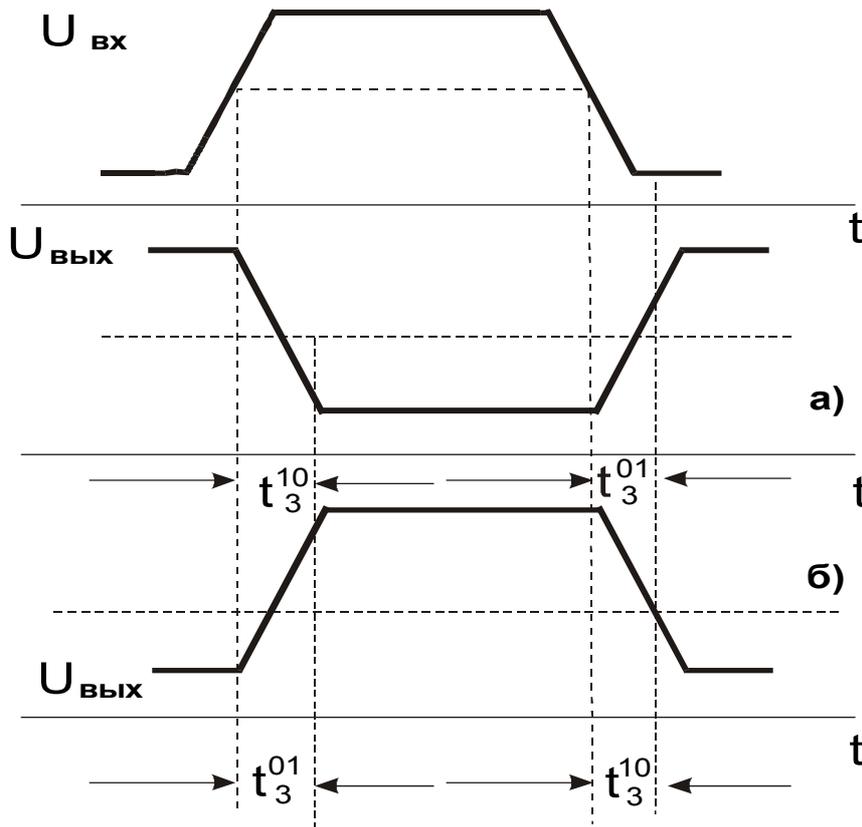


Рис. 122

ней задержкой распространения сигнала $t_{3CP} = (t_{3}^{10} + t_{3}^{01})/2$, где t_{3}^{01} и t_{3}^{10} - задержки включения и выключения элемента. На рис.122 показаны эти задержки сигнала на выходе относительно входа для инвертирующего (а) и неинвертирующего (б) элемента.

Технические условия не гарантируют точных значений t_{3}^{01} и t_{3}^{10} . Технологический разброс и старение ЦМС случайным образом (с определенным законом распределения вероятностей) определяют задержки сигнала в пределах t_{3MIN} и t_{3MAX} .

Известно лишь, что $t_3 < t_{3MAX} = t_{3ПАСП}$, что и определяет выбор микросхем по характеристике быстродействия.

Возможные последствия разброса задержек (изменения длительности, импульсов “гонки” сигналов и др.) рассматриваются в главе 2.

По величине t_{3CP} различают сверхбыстродействующие ЦМС ($t_{3CP} < 5$ нс, ЭСЛ), быстродействующие ($t_{3CP} = 5 - 10$ нс, ЭСЛ, ТТЛ, ДТЛ), микросхемы среднего ($t_{3CP} = 10 - 100$ нс, ДТЛ, ТТЛ, РЕТЛ) и низкого ($t_{3CP} > 100$ нс, РТЛ, НСТЛМ) быстродействия.

6. Потребляемая мощность ЦМС (или потребляемый ток)

$$P_{CP} = (P_{ВКЛ} + P_{ВЫКЛ})/2 \text{ или } P_{CP} = 0,5E(I_{П}^0 + I_{П}^1),$$

где $I_{П}^0$ и $I_{П}^1$ - токи, потребляемые от источника напряжения E при $U_{ВЫХ} = U^0$ и $U_{ВЫХ} \approx U^1$ соответственно.

При повышении P_{CP} путем увеличения токов $I_{П}^0$ и $I_{П}^1$ возрастает быстродействие ЦМС за счет ускорения перезаряда паразитных емкостей. Так для ЦМС РТЛ $P_{CP} = 0.5-1.5$ мВт при $t_{3CP} = 100 - 1500$ нс, для ЦМС ЭСЛ $P_{CP} = 30 - 50$ мВт, при $t_{3CP} \leq 10$ нс.

Иногда рассматривают обобщенный параметр- *работу переключения* ЦМС $A_{П} = P_{CP}t_{3CP}$, которая определяет качество схемотехнического проектирования и конструкторско-технической реализации микросхем. Тенденции развития ЦМС - снижение $A_{П}$. В 80 - х годах $A_{П} \approx (0,01 - 1,0)10^{-12}$ Дж. Теоретический предел $A_{П} = 10^{-15} - 10^{-18}$ Дж.

7. Температурный диапазон

Температурный диапазон серийных ЦМС простирается от -55 до $+125^0$ С.

Наименее устойчивы к воздействию температуры ЦМС типов РТЛ, РЕТЛ, ЭСЛ. Более устойчивы схемы ТТЛ, НСТЛМ, КМОП.

Ниже в таблице приведены параметры основных систем элементов ЦМС.

Параметры основных систем элементов цифровых микросхем

названия	Е, В	Р _{СР} , мВт	t _{ЗСР} , нс	F _{МАХ} , МГц	U ⁰ _{ВХМА} , В	U ¹ _{ВХМІN} , В	U ⁰ _{ВЫХМА} , В	U ¹ _{ВЫХМАХ} , В
РТЛ(RTL)	3,6	20	10	-	0,50	0,88	0,3	-
ДТЛ(DTL)	5	8	30	5	-	-	-	-
ТТЛ(TTL)	5	10	10	35	0,8	2,0	0,4	2,4
Быстр.ТТЛ (HTTL)	5	22	6	50	0,8	2,0	0,4	2,4
Маломощные ТТЛ(LPTTL)	5	1	33	3	0,8	2,0	0,4	2,4
ТТЛШ(STTL)	5	16	4	75	0,8	2,0	0,5	2,7
Маломощные ТТЛШ(LSTTL)	5	2	10	40	0,8	2,0	0,5	2,7
Усовершенствованные маломощные ТТЛШ (ALSTTL)	5	1	4	50	0,8	2,0	0,5	2,5
Усоверш. ТТЛ (ASTTL)	5	8	2,5	100	0,8	2,0	0,5	3,0
ЭСЛ (ECL)	-5,2	25	2	-	-1,48	-1,13	-1,6	-0,98
p-МДП (PMOS)	-9	≈1		0,1	-4,0	-1,2	-8,5	-1,0
n-МДП (NMOS)	+5	≈0,1	≈100	3	0,8	2,4	0,4	2,4
КМОП(CMOS)	3-15	0,5	100	3	1,5	3,5	0,5	4,5
Быстродейс. КМОП (HCMOS)	5	0,5	10	30	1,0	3,5	0,05	4,95

7.3. Система элементов ТТЛ (ТТЛШ)

ЦМС ТТЛ (серии 131, 133,155) и транзисторно-транзисторные схемы с переходами Шоттки (ТТЛШ) (серии 531, 533,555, КР1533) обеспечивают среднее (t_{ЗСР} = 10 – 20 нс, Р_{СР} = 10 – 25 мВт) и высокое (t_{ЗСР} = 5 – 7 нс, Р_{СР} = 20 – 40 мВт) быстродействие. На рис.123, а) показана принципиальная схема ячейки ЦМС ТТЛ, на рис.123, б) временные диаграммы сигналов.

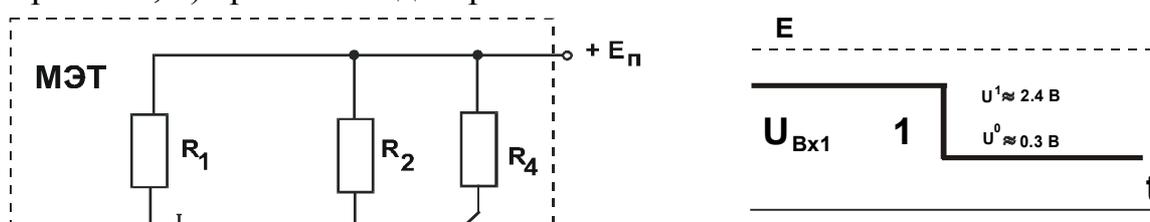


Рис. 123

В составе схемы ТТЛ (рис. а) многоэмиттерный транзистор (МЭТ) T_M и сложный инвертор (T_1, T_2, T_3). Состояние ЦМС ТТЛ определяется по уровню выходного сигнала напряжения – *открытое*, если $U_{ВХ} = U^0 \approx 0,3$ В или *закрытое*, если $U_{ВХ} = U^1 \approx 2,4$ В.

Открытое состояние ЦМС ТТЛ

Если на всех входах $U_{ВХi} = U^1 \approx 2,4$ В, то все переходы “база – эмиттер” МЭТ смещены в обратном направлении ($U_{БЭ} < 0$, $i = \overline{1, m}$), а переход “база – коллектор” в прямом ($U_{БК} > 0$). МЭТ работает в *активном инверсном* режиме. При этом коэффициент передачи тока $\beta_i \ll 1$. Из базы в цепи эмиттера передается ничтожная часть тока I_R от источника питания. Основной ток втекает в базу транзистора T_1 и далее в T_2 , насыщая оба транзистора. Выходное напряжение определяется напряжением на коллекторе насыщенного транзистора T_2 , то есть $U_{ВЫХ} = U_{КН} \approx 0,3$ В.

Транзистор T_3 заперт благодаря диоду смещения, так как $U_{К2} \approx 0,3$ В, $U_{К1\perp} = U_{БН2} + U_{КН1} = 0,8 + 0,3 = 1,1$ В, $U_D = 0,7$ В; $U_{БЭ3} = U_{К1\perp} - U_D - U_{КН2} = (1,1 - 0,7 - 0,3) = 0,1$ В $< U_{БП} \approx 0,6$ В

Выходное сопротивление ТТЛ - элемента в открытом состоянии определяется сопротивлением насыщенного перехода коллектор – эмиттер T_2 , то есть мало.

Закрытое состояние

Если на одном, нескольких или всех входах сигнал логического нуля $U_{ВХi} = U^0 = U_{КН ГЕН} \approx 0,3$ В, соответствующие переходы “база – эмиттер” МЭТ прямосмещены ($U_{БЭi} > 0$, $U_{БК} > 0$). Ток I_R течет в цепи источников логических нулей. Малые напряжения насыщения выходных транзисторов источников через насыщенные переходы T_M прикладываются к базе T_1 , запирая его. Это вызывает запирающие T_2 , возрастание $U_{К1}$ и отпирание T_3 . При запирающей T_1 избыточный заряд в базе быстро рассасывается через насыщенные переходы T_M . Открытый транзистор T_3 работает в режиме эмиттерного повторителя с активной нагрузкой в виде транзистора T_2 , обеспечивая малое выходное сопротивление ТТЛ – элемента. При этом $U_{ВЫХ} = U^1 \approx 2,4$ В.

Примером ЦМС с мощным инвертором является микросхема 155ЛА6 (4И-НЕ×2) (рис.126, б)

Схема со свободным коллектором

На (рис.127, а) показана принципиальная схема ячейки ТТЛ со свободным коллектором, а на (рис.127, б) ее условное обозначение.

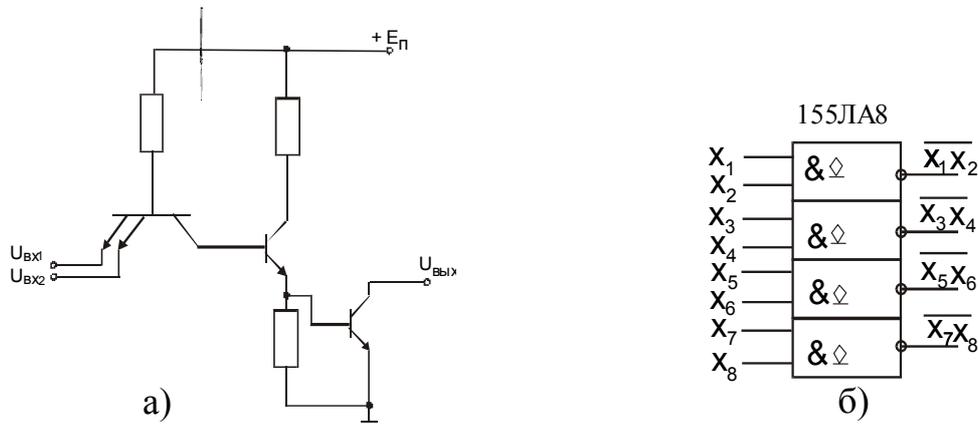


Рис. 127

Такая схема позволяет :

- реализовать “монтажные” И, ИЛИ;
- подключать различные нагрузки (индикаторы, реле и т. п.);
- согласовывать ТТЛ ЦМС с другими системами элементов (например с ЦМС КМОП).

Примером ЦМС со свободным коллектором является микросхема 155ЛА8 (2И-НЕ×4).

Схема с двухступенчатой логикой и логическим расширением

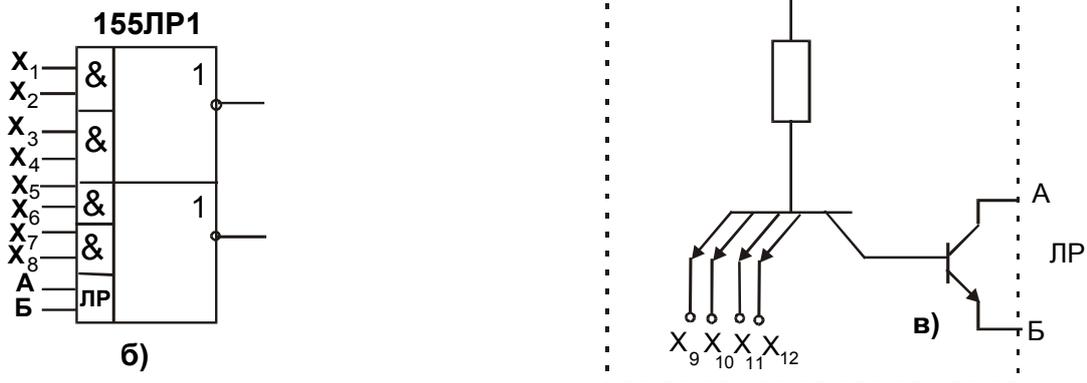


Рис.128

Такие элементы позволяют расширить функциональные возможности основных элементов ТТЛ.

На (рис.128, а) показана схема элемента ТТЛ с двухступенчатой логикой И-ИЛИ-НЕ, на (рис.128, б) его условное обозначение, (микросхемы 155ЛР1), а

на (рис.128, в) схема логического расширителя, позволяющего, увеличить количество входов в основной ЦМС.

Быстродействующая система ТТЛШ (серии 555, 533, 531, КР1533) с не насыщающимся переходом Шоттки (металл – полупроводник).

ИМС ТТЛ с малым потреблением ($E = 1,2 \text{ В}$, $P_{CP} = 0,5 \text{ мВт}$), что позволяет уменьшить габариты микросхемы, тем самым увеличить степень интеграции. Система ЦМС ТТЛ очень распространена. На ней построены ЭВМ серии “Ряд” ЕС1020, ЕС1030, ЕС1040.

7.4. Система элементов КМОП (КМДП)

КМДП – комплементарные (дополняющие) структуры МДП (металл - диэлектрик - полупроводник)

Характеризуются малым потреблением, высокими термостабильностью и помехоустойчивостью, а так же наивысшей степенью интеграции. Эта технология является базовой для многих фирм производящих микропроцессоры (например INTEL 8086) Система элементов КМДП включают серии микросхем 164, 176, 561, 564, КР1561, а также БИС четвертого поколения (к 586, 587, 588).

Разновидности ЦМС КМДП элементов

Ключ-инвертор (Рис.129, а).

Два полевых транзистора разной проводимости последовательно соединены в схеме ключа стоками. Входное напряжение подается на объединенные затворы и управляет транзисторами (с n- (Т1) и р- (Т2) каналами). Если $U_{ВХ}$ растет, то $U_{ЗИ1}$ повышается, а $U_{ЗИ2} = |E - U_{ВХ}|$ уменьшается и наоборот (Рис.129, б).

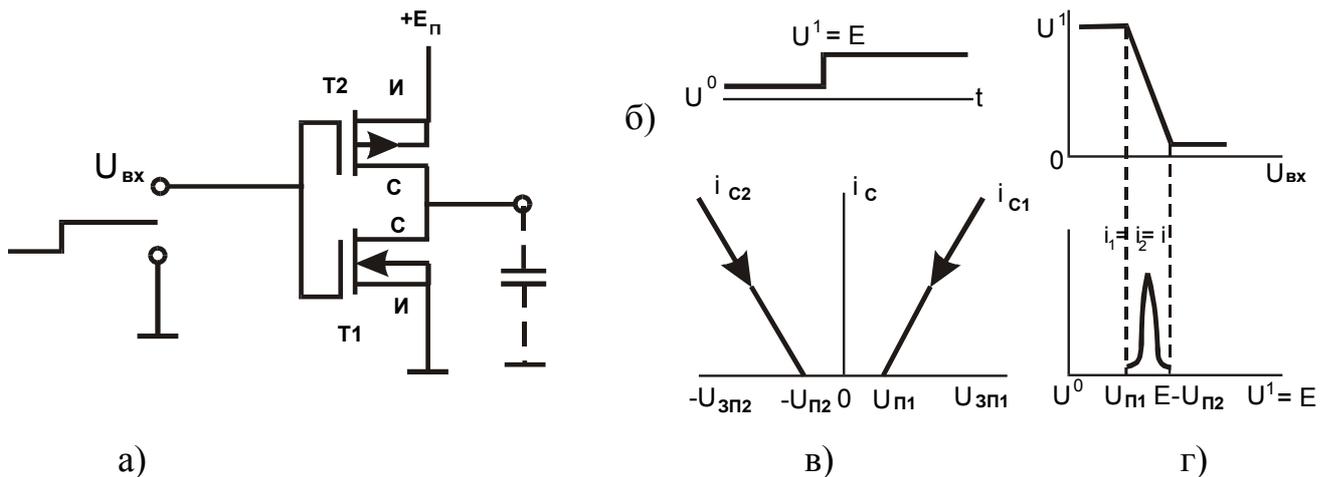
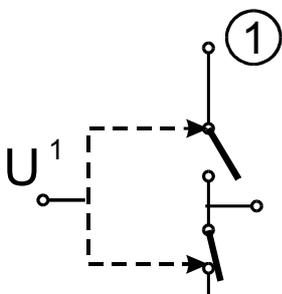


Рис. 129

Два состояния ключа определяют его логические свойства:

1. Если $U_{ВХ} = U_{ЗИ1} = U^1 = E > U_{П1}$ и Т1 (с n-каналом) проводит, Т2 (с р-каналом) заперт, так как $U_{ЗИ2} = E - E = 0 > -U_{П2}$. На выходе напряжение $U_{ВЫХ} = U^0 \approx 0$. Отличие от нуля определяется ничтожным



падением напряжения на сопротивлении проводящего Т1 от тока запертого Т2 (наноамперы). Выходное состояние элемента соответствует логическому нулю.

Рис.130

2. Если $U_{ВХ} = U^0 \approx 0$, то заперт Т1, Т2 проводит, так как $U_{зИ2} = (0 - E) < -U_{П2}$. Напряжение на входе $U_{ВЫХ} = U^1 = E$. Выходное состояние элемента соответствует логической единице.

Свойства ЦМС КМДП определяются достоинствами собственно МДП-структур:

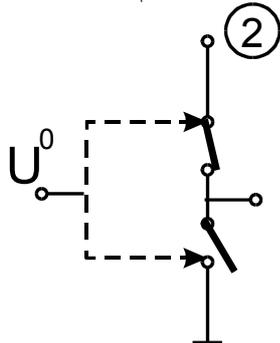


Рис.131

- высоким входным сопротивлением $R_{ВХ} = 10^{10} - 10^{15}$ Ом;
- использованием в качестве компонентов только МДП- транзисторов;
- униполярным управлением и высокой степенью развязки между управляющими и входной цепями;

Соединение двух МДП – структур разной проводимости в комплементарную (до-

полняющую) обуславливают дополнительные достоинства электронных ключей КМДП:

1. Экономичность, так как в статических состояниях инвертор тока не потребляет (токи транзисторов возникают только во время переключения; причем один возрастает, другой убывает и наоборот (Рис.129, б, г)).

2. Высокая помехоустойчивость, так как логический перепад близок к напряжению питания ($E = 3-15$ В).

$$U_{л} = U_1 - U_0 = E - 0 \approx E \begin{cases} > U_{л.ТТЛ} = 2 \text{ В} \\ >> U_{л.ЭСЛ} < 1 \text{ В} \end{cases}$$

3. Работоспособность при изменении питания в широких пределах (3-15 в). Можно использовать нестабилизированный источник питания.

4??Быстродействие КМДП выше, чем у МДП - ключей, так как сопротивление каналов снижены из-за отсутствия статического тока. Для ЦМС КМДП типа И-НЕ средняя задержка больше чем ЦМС ТТЛ ($t_{з\text{СР ТТЛ}} \approx 10 - 30$ нс $< t_{з\text{СР КМДП}} = 100 - 1000$ нс). Проигрыш КМДП (по быстродействию по сравнению с ТТЛ) объясняется медленным разрядом паразитных емкостей КМДП через большое динамическое сопротивление открывающегося канала, в то время как в ЦМС ТТЛ- разряд осуществляется через насыщенный транзистор.

5. Высокая нагрузочная способность $N = 50 - 1000$, так как $R_{ВХ} = 10^3 - 10^6$ МОм;

6. Незначительная зависимость от температуры.

7. Наивысшая степень интеграции среди всех систем элементов ЦМС.

Логическая схема ИЛИ-НЕ (Рис.132, а)

Параллельно включены транзисторы с n- каналом, нагрузкой их является последовательная цепь транзисторов с р- каналом. Каждый транзистор n- типа связи со своим транзистором р- типа.

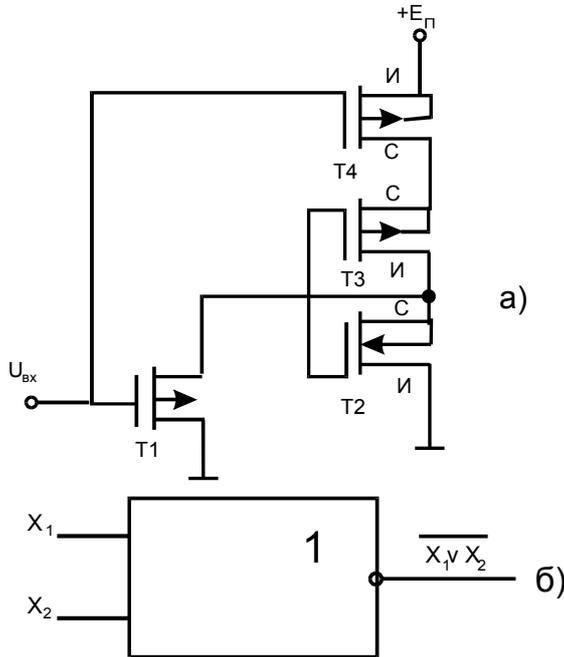


Рис. 132

Два статических состояния схемы:

1. Если на всех входах $U_{ВХ1} = U_{ВХ2} = U^0 \approx 0$, то T1 и T2 заперты, а T3 и T4 открыты $U_{ЗП3,4} = 0 - E < -U_{П}$ и $U_{ВЫХ} = U^1 = E$.
2. Если на одном или обоих входах $U_{ВХ1,2} = U^1 \approx E$, то соответствующий из T1 и T2 (или оба) n- транзисторы открыты, а соответствующие тран-зисторы р- типа закрыты. Открытый транзистор шунтирует выход и $U_{ВЫХ} = U^0 = 0$. Видно, что вся схема выполняет функцию ИЛИ- НЕ (рис.132, б).

Логическая схема И- НЕ (рис.133, а)

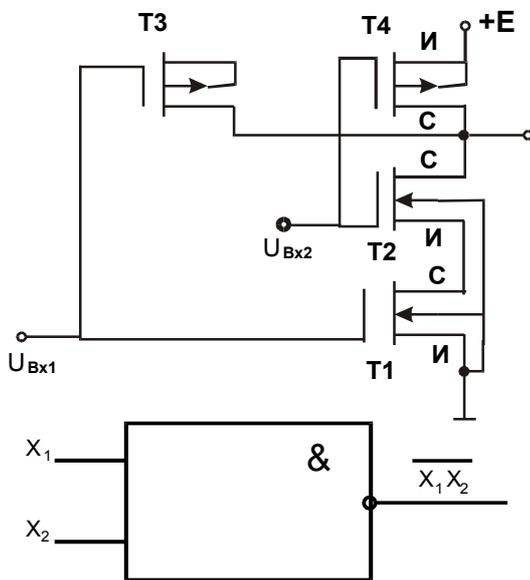


Рис. 133

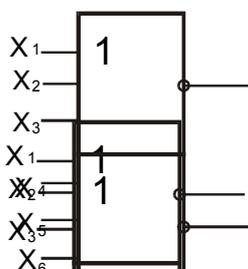
Здесь параллельны транзисторы с р- каналом (T3, T4) и последовательно соединены транзисторы с n- каналом (T1, T2).

Два статических состояния схемы:

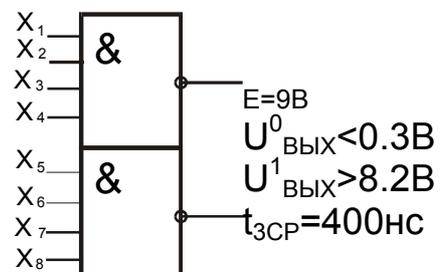
1. Если на обоих входах $U_{ВХ1} = U_{ВХ2} = U^1 = E$, то T1 и T2 открыты и $U_{ВЫХ} = U^0 = 0$ (T3 и T4 - закрыты).
2. Если на одном или обоих входах $U_{ВХ1,2} = U^0 = 0$, то T1 и T2 закрыты, T3 и T4 открыты и $U_{ВЫХ} = U^1 = E$. Закрытый транзистор разрывает последовательную цепь. Схема выполняет функцию И – НЕ (рис.133, б).

Примеры серийных ЦМС КМОП

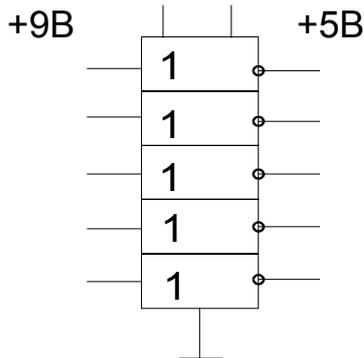
К176ЛП4 (2X3 или-НЕ/НЕ)



К176ЛП12 (2X4 и-НЕ/НЕ)



К176ПУ1 (понижающий преобразователь уровней)



К176ПУ5 (повышающий преобразователь уровней)

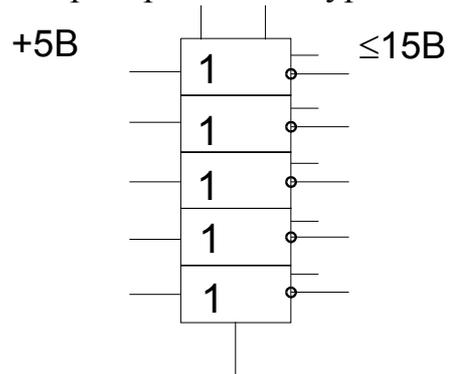
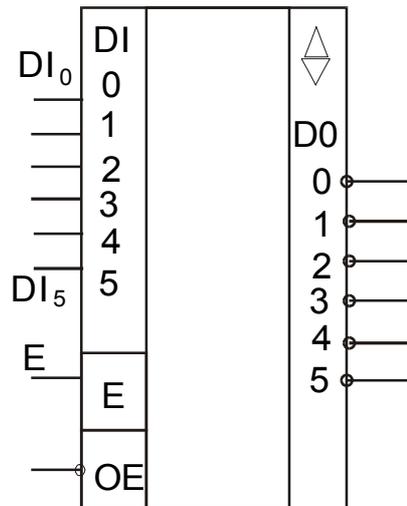


Схема с 3-мя состояниями выхода 561ЛН1



$$\overline{DO}_i = \begin{cases} \overline{DI}_i \vee \overline{E}, \overline{OE} = 0 \\ 7 - \text{СОСТ.}, \overline{OE} = 1 \end{cases}$$

7.5. Система элементов ЭСЛ

Система элементов с *эмиттерно- связанной логикой* (ЭСЛ) (на переключателях токов (ПТТЛ)) обладает наибольшим быстродействием ($t_{3CP} < 10\text{нс}$) при значительной потребляемой мощности ($P_{CP} = 40-100$ и более мВт).

Типовой логический элемент ЭСЛ (рис.134, а) состоит из переключателей токов ($T_1- T_{m+1}$) и выходных эмиттерных повторителей ($T_{ЭП1,2}$) с низкоомными выходами ($R_{ВЫХЭП1,2} \approx 30 - 50 \text{ Ом}$). Все транзисторы работают в *активном* режиме

при малых уровнях управляющих напряжений. Это обуславливает повышенное быстродействие.

При указанном способе питания ($U^1 = -0,9\text{В}$ высокий отрицательный уровень, $U^0 = -1,9\text{В}$ - низкий отрицательный уровень, ЦМС ЭСЛ работает в положительной логике).

Базовый элемент ЭСЛ выполняет операцию ИЛИ- НЕ по выходу 1 и ИЛИ по выходу 2.

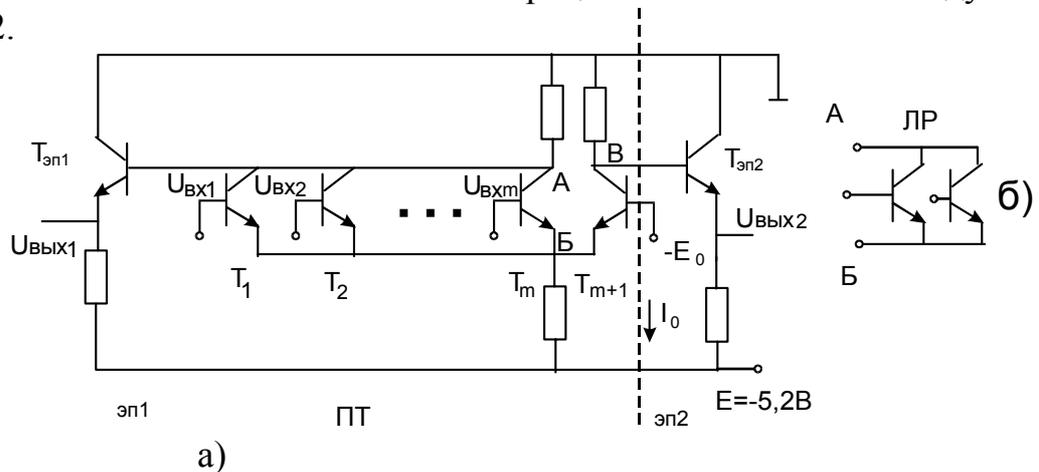


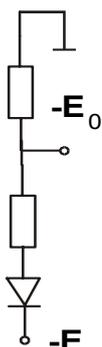
Рис.134

Рассмотрим статические состояния:

1. Если на всех входах $\{U_{ВХi}\} = U^0 = -1,9\text{В}$, транзисторы $T_1 \dots T_m$ заперты, так как $U_{ВХi}^0 = -1,9\text{В} < U_{Э}^0 = I_{Э}R_{Э}$ и все $U_{БЭi} < 0$, потенциал объединенных коллекторов U_A близок к 0, $U_{ВЫХ1} = U^1 = -0,9\text{В}$ кодирует “единицу”. Уровень E_0 на входе T_{m+1} подобран таким образом, чтобы этот транзистор был открыт ($U_{Э}^0 = I_{Э}^0 R_{Э} < E_0 = -1,2\text{В}$). Падение напряжения на резисторе $R_{Э}$ за счет тока $I_{Э}^0$ обеспечивает активный режим T_{m+1} (отрицательная обратная связь через $R_{Э}$). Потенциал U_B близок к U^0 . Эмиттерные повторители передают на выходы уровни: $U_{ВЫХ1} = U^1$, $U_{ВЫХ2} = U^0$ так, как их коэффициенты передачи $K_{ЭП1,2} \approx 1$.

2. Если на одном, нескольких или всех входах $U_{ВХi} = U^1 = -0,9\text{В}$, соответствующие T_i из $T_1 \dots T_m$ открыты и активны и через них течет суммарный ток $I_{Э}^1$. T_{m+1} заперт падением $U_{Э}^1 = I_{Э}^1 R_{Э} > E_0 = -1,2\text{В}$. На выходах: $U_{ВЫХ1} = U^0$, $U_{ВЫХ2} = U^1$. Режим подобран таким образом, чтобы при любой в данном состоянии кодовой комбинации через резистор $R_{Э}$ переключателя токов протекал примерно постоянный ток $I_{Э}^1 = I_0$.

Указанный способ питания микросхемы позволяет оставлять в ЛЭ неиспользованные входы не присоединенными. Опорное напряжение $E_0 = -1,2\text{В}$ создается интегральной термокомпенсированной схемой, питаемой от того же источника $E = -5,2\text{В}$ (рис. 135).



Несмотря на невысокую контрастность уровней “единицы” и “нуля” ($U_{ЛЭСЛ} = \Delta U_{ВЫХ} = 0,7 - 0,9\text{В} < U_{ЛТТЛ} \approx 2\text{В}$) и незначительный уровень допустимой помехи ($U_{П} = 0,15 - 0,2\text{В}$) помехоустойчивость ЦМС ЭСЛ достаточно благодаря низкоомной связи между элементами.

Эмиттеры транзисторов $T_{ЭП1,2}$ некоторых ЦМС ЭСЛ не соединены внутри микросхемы со своими эмиттерны-

Рис. 135 ми резисторами, что дает возможность присоединить выходы нескольких схем к одному эмиттерному резистору (схема ИЛИ для выходов нескольких ИМС- реализация операции “монтажное” ИЛИ).

К ЦМС ЭСЛ относятся микросхемы серий 100 (военное применение) и 500 (коммерческое).

Примеры:

1. **Основная схема** с возможным расширением по ИЛИ (ЗИЛИ - НЕ/ИЛИ) К137ЛМ1: $E = -5,0\text{В}$; $U_{\text{ВЫХ}}^1 = - (0,7 - 0,95)\text{В}$; $U_{\text{ВЫХ}}^0 = - (1,45 - 1,9)\text{В}$; $t_{\text{ЗСР}}=6\text{нс}$; $P_{\text{СР}}=40\text{мВт}$; $N=15$; $M=3$; $U_{\text{П}}=0,15 - 0,2\text{В}$. Допускается подключение внешних резисторов и расширение по ИЛИ.

2. **Элемент с повышенной нагрузочной способностью** К137ЛМ3 (ЗИЛИ - НЕ/ИЛИ) обладает параметрами: $E=-5,25\text{В}$; $U_{\text{ВЫХ}}^1 = - (0,7 - 0,95)\text{В}$; $U_{\text{ВЫХ}}^0 = - (1,45 - 1,9)\text{В}$; $P_{\text{СР}}=250\text{мВт}$; $t_{\text{ЗСР}}=6\text{нс}$; $N=100$; $M=3$.

Достоинства ЭСЛ- элементов (быстроедействие, логическая гибкость основной ячейки, не критичность к разбросу коэффициента_передачи тока β при производстве) послужили причиной широкого их использования в ЭВМ и быстродействующих цифровых измерительных приборах.

Разновидностью ЭСЛ является малосигнальная ЭСЛ (МЭСЛ) (рис.136) в которой уменьшено напряжение питания $E=2 - 3\text{В}$ и логический перепад $U_{\text{Л}}= U^{1-0} < 0,3 - 0,4\text{В}$. При этом из схемы исключаются эмиттерные повторители, потребляющие значительную мощность. Основные транзисторы входят в насыщение, но степень насыщения мала (квазинасыщение) и высокое быстроедействие МЭСЛ сохраняется. За счет малого $U_{\text{П}}$ помехоустойчивость МЭСЛ меньше ($U_{\text{П}} = 0,1-0,15\text{В}$). Из-за отсутствия ЭП нагрузочная способность МЭСЛ невелика ($N=4-5$). Среднее время задержки ЦМС ЭСЛ $t_{\text{ЗСР}}=0,5-1\text{нс}$ при потребляемой мощности $P_{\text{СР}} = 10\text{мВт}$ и величине логического перепада $U_{\text{П}} = 0,4\text{В}$.

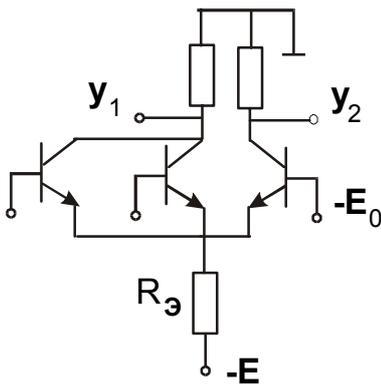


Рис. 136

Библиографический список.

1. Пухальский Г. И., Новосельцева Т. Л., Проектирование дискретных цифровых устройств на интегральных микросхемах: Справочник М.: Радио и связь, 1990.- 304 с.
2. Богданович М. И. , Грель И. И. и др. Цифровые интегральные микросхемы : Справочник. – Минск: “Беларусь”, “Полымя”, 1996 –608 с.

3. Шило В. Л. Популярныe цифровыe микросхемы: Справочник М.: Радио и связь, 1987.-352 с.
4. Алексенко А. Г. Основы микросхемотехники. Изд. 2-е. М.: Сов. Радио, 1977 - 408с.
5. Интегральные микросхемы: Справочник / Б. В. Тарабрин, Л. Ф. Лукин, Ю. Н. Смирнов и др.; Под ред. Б. В. Тарабрина.- М.: Радио и связь, 1984. – 528 с.
6. Ланцов А. М., Зворыкин Л. Н., Осипов И. Ф., Цифровые устройства на комплиментарных микросхемах . - М.: Радио и связь, 1983. – 272 с.
7. TTL. Advanced Low – Power Schottky, Date book, Vol. 2. Texas Instruments, 1983.
8. Проектирование импульсных и цифровых устройств радиотехнических систем: Учеб. пособие для радиотехнич. спец. вузов/ Гришина Ю. П., Казаринов Ю. М., Катиков В. М. И др.; Под ред. Ю. М. Казаринов. – М.: Высш. шк., 1985 – 319 с.
9. Гольденберг Л. М. , Малев В. А., Мальков Г. Б. Цифровые устройства и микропроцессорные системы: Задачи и упражнения. : Учеб. Пособие для вузов. – М.: Радио и связь, 1992. – 256 с.
10. Бернюков А. К. Цифровые и импульсные устройства: Методич. указания к лаб. работам / Владим. политехн. ин-т. Владимир, 1992. 24 с.

Библиографический список.

1. Пухальский Г.И., Новосельцева Т.Л. Проектирование дискретных цифровых устройств на интегральных микросхемах: Справочник М.: радио и связь, 1990. - 304с.
2. Богданович М. И., Грель И.И. и др. Цифровые интегральные микросхемы: Справочник. - Минск: «Беларусь», «Полымя», 1996.- 608с.
3. Шилов В.Л. Популярныe цифровые микросхемы: Справочник М.: Радио и связь, 1987. - 382с.
4. Алексенко А.Г. Основы микросхемотехники. Изд. 2-е. М.: Сов. Радио, 1977. - 408с.
5. Интегральные микросхемы: Справочник / Б.В. Тарабрин, Л.Ф. Лукин, Ю.Н. Смирнов и др.; Под ред. Б.В. Тарабрина. - М.: Радио и связь, 1984. - 528с.
6. Ланцов А.М., Зворыкин Л.Н., Осипов И.Ф., Цифровые устройства на комплементарных микросхемах. - М.: Радио и связь, 1983. - 272с.
7. TTL. Advanced Low - Power Schottky, Date book, Vol. 2. Texas Instruments, 1983.
8. Проектирование импульсных и цифровых устройств радиотехнических систем: Учеб. пособие для радиотехнич. спец. вузов / Гришина Ю.П., Казаринов Ю.М., Катиков В.М. и др.: Под ред. Ю.М. Казаринова. - М.: Высш. шк., 1985 - 319с.
9. Гольденберг Л.М., Малеев В.А., Мальков Г.Б. Цифровые устройства и микропроцессорные системы: Задачи и упражнения.: Учеб. пособие для вузов. - М.: Радио и связь, 1992. - 256с.
10. Бернюков А.К. Цифровые и импульсные устройства: Методич. Указания к лаб. работам / Владим. политехнич. ин-т. Владимир, 1992. 24с.
11. Цифровая и вычислительная техника: Учебник для вузов / Э.В. Евреинов, Ю.Г. Бутыльский и др., Под ред. Э.В. Евреинова. М.: Радио и связь, 1991. 464с.

12. Цифровая схемотехника: Метод указания к лаб. работам / Владим. гос. ун-т. Сост. А.Н. Никитин. Владимир, 1999. 48с.
13. Никитин А.И. Никитин О.Р. микропроцессоры и микроЭВМ. Учебное пособие, ВлГУ, 1999. 128с.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. ОБЩИЕ МЕТОДЫ ПРЕДСТАВЛЕНИЯ И ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ В ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВАХ. .	4
1.1. Системы счисления	4
1.2. Перевод чисел из одной системы счисления в другую	5
1.3. Формы представления чисел в ЦВУ	7
1.4. Выполнение арифметических операций. Специальные машинные коды. . . .	9
1.5. Контрольные вопросы и задания для самоподготовки	12
Глава 2. ЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ЦИФРОВЫХ УСТРОЙСТВ . . .	13
2.1. Алгебра логики	13
2.1.1. Логические переменные и функции.	13
2.1.2. Основные свойства логических функций	15
2.2. Задание логических функций	16
2.2.1. Задание логических функций таблицей истинности.	16
2.2.2. Аналитическое представление логических функций с помощью совершенных нормальных форм.	16
2.2.3. Функционально полные системы логических функций.	18
2.2.4. Минимизация логических функций	18
2.2.5. Минимизация систем логических функций	20
2.3. Комбинационные цифровые устройства	21
2.3.1. Общие положения	21
2.3.2. Базовые интегральные схемы и их основные параметры	21
2.3.3. Типовые КЦУ	24
2.3.3.1. Дешифраторы (декодеры).	24
2.3.3.2. Шифраторы (кодеры).	25

2.3.3.3. Мультиплексоры	25
2.3.3.4. Демльтиплексоры.	26
2.3.3.5. Сумматоры и АЛУ.	26
2.3.3.6. Синтез комбинационных схем на мультиплексорах и дешифраторах.	28
2.3.3.7. Быстродействие КЦУ и эффект состязаний в них («гонки»)	30
2.4. Контрольные вопросы и задания для самоподготовки.	31
Глава 3. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА. ЦИФРОВЫЕ АВТОМАТЫ	33
3.1. Основные понятия абстрактной теории цифровых конечных автоматов	33
3.2. Формы задания абстрактных конечных автоматов	34
3.2.1. Представление АКА с помощью таблиц переходов и выходов	34
3.2.2. Представление АКА с помощью графа.	35
3.2.3. Матричное представление АКА.	36
3.3. Абстрактный синтез конечных автоматов	37
3.3.1. Эквивалентность конечных автоматов	37
3.3.2. Эквивалентные преобразования конечных автоматов	37
3.3.3. Минимизация числа состояний абстрактных автоматов.	40
3.4. Структурный синтез конечных автоматов.	41
3.4.1. Задачи структурного синтеза конечных автоматов	41
3.4.2. Блок-схема конечного автомата.	42
3.5. Элементарные автоматы (триггеры) и их свойства	44
3.5.1. Синхронные триггеры	45
3.5.2. Асинхронные триггеры	50
3.6. Определение функций возбуждения триггеров	50

3.6.1. Кодирование таблиц переходов	51
3.6.2. Определение функций внешних переходов	52
3.6.3. Определение функций возбуждения элементарных автоматов	52
3.6.3.1. Табличный метод определения функций возбуждения триггеров	52
3.6.3.2. Аналитический метод определения функций возбуждения триггеров	53
3.6.3.3. Определение функций возбуждения триггеров методом сравнения	54
3.7. Типовые последовательностные цифровые устройства	57
3.7.1 Регистры	57
3.7.1.1. Классификация регистров и особенности их схемотехнических решений.	57
3.7.1.2. Применение регистров для преобразования информации	58
3.7.2. Счетчики	59
3.7.2.1. Общие сведения	59
3.7.2.2. Суммирующие двоичные счетчики	60
3.7.2.3. Вычитающие и реверсивные двоичные счетчики	61
3.7.2.4. Двоично-десятичные синхронные счетчики	62
3.7.2.5. Метод синтеза синхронно-асинхронных десятичных счетчиков	64
3.7.3. Цифровые делители частоты	65
3.7.3.1. ЦДЧ на основе пересчетных схем	65
3.7.3.2. ЦДЧ на основе регистров	67
3.8. Контрольные вопросы и задания для самоподготовки.	69
Глава 4. ЭЛЕМЕНТЫ СИНТЕЗА ЦИФРОВЫХ ОПЕРАЦИОННЫХ УСТРОЙСТВ .	71
4.1. Понятие операционного устройства. Общая структура операционного устройства	71

4.2. Принцип микропрограммного управления	72
4.3. Язык описания операционных устройств	73
4.3.1. Описание слов и массивов	73
4.3.2. Двоичные выражения	74
4.3.3. Микрооперации и их классификация	74
4.3.4. Вычисление логических условий	75
4.3.5. Содержательный граф микропрограммы	75
4.4. Структурный базис операционных устройств	77
4.4.1. Шины	77
4.4.2. Регистры	78
4.4.3. Комбинационные устройства	78
4.5. Построение операционного автомата	79
4.5.1. Каноническое представления ОА	79
4.5.2. Структурный синтез ОА	80
4.6. Построение управляющего автомата с жесткой и программируемой логикой	82
4.6.1. Интерпретация микропрограмм автоматами Мили и Мура	82
4.6.2. Построение управляющего автомата с программируемой логикой	84
4.7. Контрольные вопросы и задания для самоподготовки.	89
Глава 5. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА	90
5.1. Классификация и основные характеристики запоминающих устройств	90
5.2. Оперативные запоминающие устройства	91
5.2.1. Построение запоминающих устройств с произвольной выборкой	91
5.2.2. Особенности динамического ОЗУ	93

5.2.3. Построение ЗУ большой емкости	94
5.2.4. Память заданной структуры	95
5.3. Постоянные запоминающие устройства	97
5.3.1. Масочные ПЗУ	97
5.3.2. Однократно программируемые ПЗУ	97
5.3.3. Перепрограммируемые ПЗУ	98
5.3.4. Программируемые логические интегральные схемы	99
5.4. Контрольные вопросы и задания для самоподготовки.	102
Глава 6. ФОРМИРОВАТЕЛИ, ГЕНЕРАТОРЫ И ПРЕОБРАЗОВАТЕЛИ ИМПУЛЬСНЫХ И ЦИФРОВЫХ СИГНАЛОВ	103
6.1. Формирователи импульсов на основе элементов задержки	103
6.1.1. Элемент задержки с интегрирующей RC-цепью и логическим элементом ТТЛ	103
6.1.2. Элемент задержки с дифференцирующей RC-цепью и ЛЭ ТТЛ	105
6.1.3. Формирователь прямоугольных импульсов на основе элементов задержки	105
6.2. Генераторы импульсов прямоугольной формы	106
6.2.1. Автоколебательный мультивибратор на логических элементах ТТЛ	106
6.2.2. Ждущий мультивибратор на логических элементах ТТЛ	108
6.2.3. Автоколебательный мультивибратор на операционном усилителе	110
6.2.4. Ждущий мультивибратор на операционном усилителе	111
6.2.5. Генераторы импульсов на специализированных микросхемах	112
6.2.6. Устройства на основе триггеров Шмитта	113
6.3. Генераторы линейно-изменяющегося напряжения	115
6.3.1. Параметры линейно-изменяющегося напряжения	115
6.3.2. Цифровые ГЛИН	116
6.4. Цифро-аналоговые и аналого-цифровые преобразователи	117
6.4.1. Цифро-аналоговые преобразователи	117
6.4.2. Аналого-цифровые преобразователи	119

6.4.2.1. АЦП последовательного счета	120
6.4.2.2. АЦП поразрядного уравнивания	121
6.4.2.3. АЦП параллельного действия	122
6.5. Контрольные вопросы и задания для самоподготовки.	124
Глава 7. Микроэлектронная база цифровых устройств	125
7.1. Классификация цифровых микросхем	125
7.2. Характеристики и параметры ЦМС	125
7.2.1. Статические внешние характеристики ЦМС	125
7.2.2. Параметры ЦМС	127
7.3. Система элементов ТТЛ (ТТЛШ)	130
7.4. Система элементов КМОП (КМДП)	134
7.5. Система элементов ЭСЛ	138
Библиографический список	140

БЕРНЮКОВ Арнольд Константинович
НИКИТИН Александр Иванович

ЦИФРОВЫЕ УСТРОЙСТВА

Учебное пособие

Редактор Е.П. Викулова

Корректор В.Б. Сони́на

Изд. лиц. № 020275 от 13.11.96. Подписано в печать Формат 60x84/16.

Бумага для множит. техники. Гарнитура Таймс. Печать офсетная. Усл.печ.л. 7,44.

Уч.-изд.л. 7,92. Тираж 250 экз. С - Зак.

Владимирский государственный университет.

Подразделение оперативной полиграфии

Владимирского государственного университета.

Адрес университета и подразделения оперативной полиграфии:

600026, Владимир, ул. Горького, 87.