



**Санкт-Петербургское государственное бюджетное образовательное
учреждение среднего профессионального образования**

ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ ГОРОДСКОГО ХОЗЯЙСТВА

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Гр. РЭ-14-7с

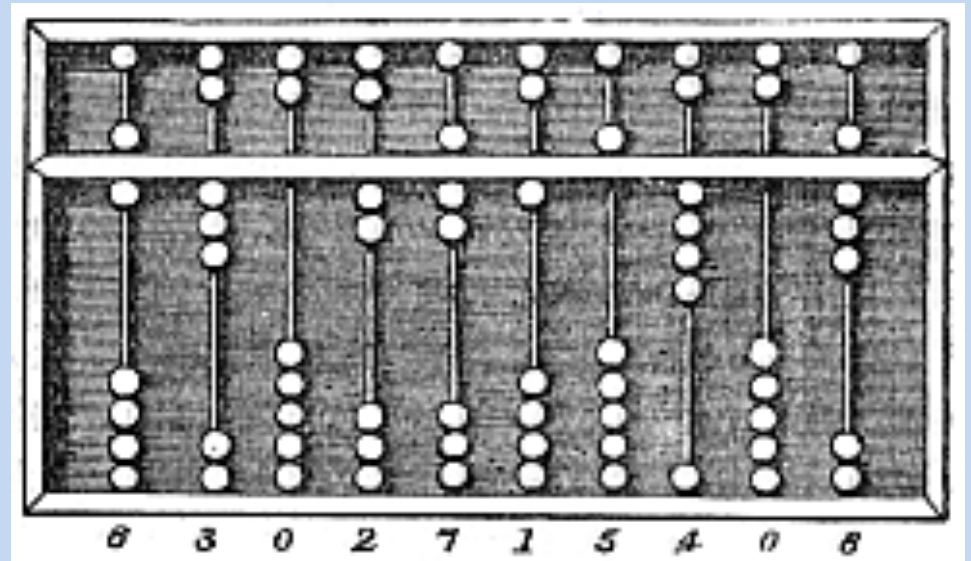
**Евгений Эдуардович Намчыл
преподаватель**

Вычислительная техника

- **Вычислительная техника** является важнейшим компонентом процесса вычислений и обработки данных.
- Слово «компьютер» означает «вычислитель».

Вычислительные приборы

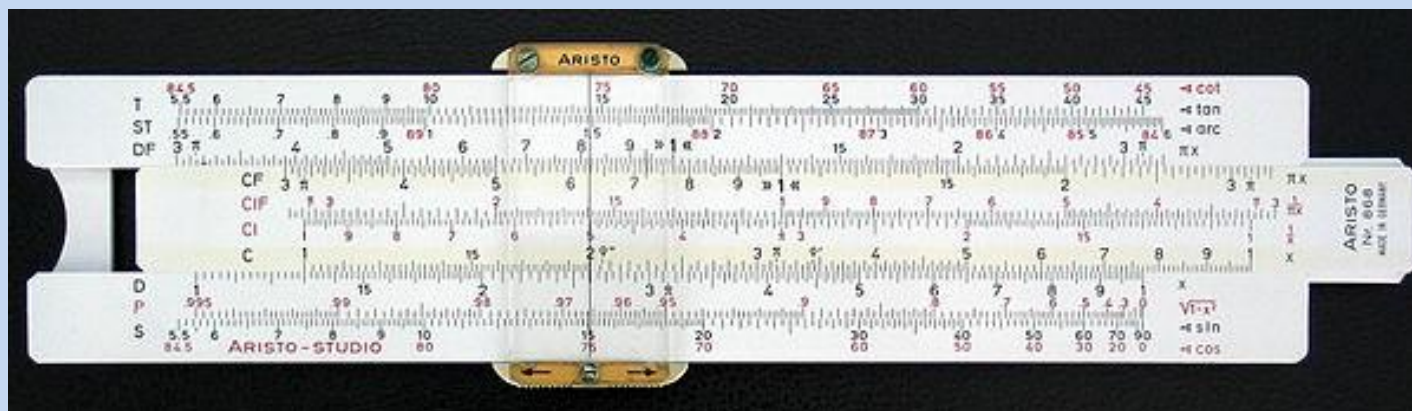
Аба́к — [счётная доска](#), применявшаяся для [арифметических вычислений](#) приблизительно с [V века до н. э.](#) в [Древней Греции](#), [Древнем Риме](#) и в Китае.





**Логарифмическая
линейка, счётная
линейка — аналоговое
вычислительное устройство,**

позволяющее выполнять
несколько математических операций,
в том числе умножение и деление
чисел, возведение в степень (чаще
всего в квадрат и куб), вычисление
квадратных и кубических корней,
вычисление
логарифмов, потенцирование,
вычисление тригонометрических и
гиперболических функций и
некоторые другие операции





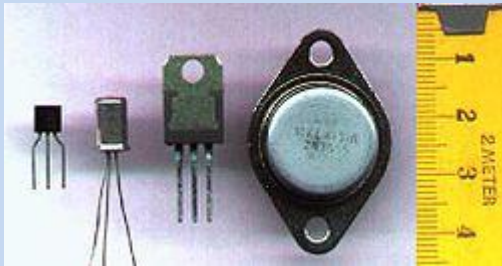
Арифмометр "Феликс"



Арифмометр — настольная или портативная механическая вычислительная машина, предназначенная для точного умножения и деления, а также для сложения и вычитания. Механическая вычислительная машина, ведущая автоматическую запись обрабатываемых чисел и результатов на особой ленте — *арифмограф*



- Первым полностью электронным настольным калькулятором был британский [ANITA Mark VII](#), который использовал дисплей на газоразрядных цифровых индикаторах и 177 миниатюрных [тиратронов](#). В июне 1963 года Friden представил EC-130 с четырьмя функциями. Он был полностью на транзисторах, имел 13-цифровое разрешение на 5-дюймовой [электронно-лучевой трубке](#) и представлялся фирмой на рынке калькуляторов по цене 2200 \$. В модель EC 132 были добавлены функция вычисления квадратного корня и обратные функции. В 1965 году [Wang Laboratories](#) произвёл LOCI-2, настольный калькулятор на транзисторах с 10 цифрами, который использовал дисплей на газоразрядных цифровых индикаторах и мог вычислять [логарифмы](#)



Структура традиционной ЭВМ

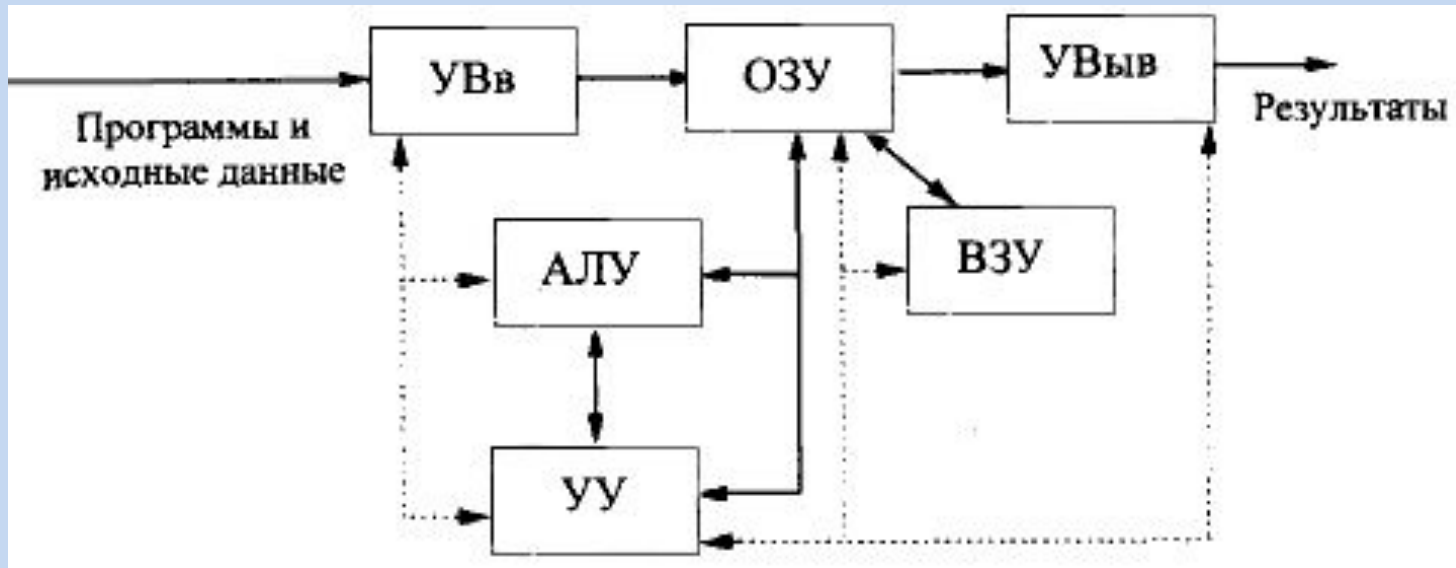
- Структура - совокупность элементов и их связей
- Электронная вычислительная машина - комплекс технических и программных средств, предназначенный для автоматизации подготовки и решения задач пользователей
- Архитектура ЭВМ - это многоуровневая иерархия аппаратно-программных средств, из которых строится ЭВМ. Каждый из уровней допускает многовариантное построение и применение. Конкретная реализация уровней определяет особенности структурного

Основные узлы компьютера

- **Центральный процессор** — это микропроцессор со всеми необходимыми вспомогательными микросхемами, включая внешнюю кэш-память и контроллер системной шины.
- **Оперативная память** (англ. Random Access Memory, память с произвольным доступом) — часть системы компьютерной памяти, в которой временно хранятся данные и команды, необходимые процессору для выполнения им операции. Обязательным условием является адресуемость (каждое машинное слово имеет индивидуальный адрес) памяти
- **Постоянная память** (ROM BIOS — Base Input/Output System) имеет небольшой объем (до 64 Кбайт), содержит программу начального запуска, описание конфигурации системы, а также драйверы (программы нижнего уровня) для взаимодействия с системными устройствами.

- **Емкость памяти** измеряется количеством структурных единиц информации, которое может одновременно находиться в памяти.
- **бит** - одна двоичная цифра. Как правило, емкость памяти оценивается в более крупных единицах измерения - **байтах** (1 байт=8 бит).
 - 1 Кбайт = 1024 байта,

устройства ввода информации (УВв),
оперативно запоминающее устройство (ОЗУ),
внешнее запоминающее устройство (ВЗУ),
устройство управления (УУ),
арифметико-логического устройства (АЛУ) ,
устройства вывода (УВыв).



- Рассмотрим пример выполнения 3-х адресной команды в ЭВМ в соответствии с принципами Дж.фон Неймана. Предположим, что некоторая i -ая команда - эта команда сложения чисел “а” и “b” и полученная сумма должна быть занесена в ЗУ по некоторому адресу.
- Пусть:
- КО «+» - 01
- Код адреса ячейки “а” – 0100
- Код адреса ячейки “ b ” – 0101
- Сумма заносится в ячейку с адресом 0110
- Тогда код команды в ячейке i – 01 0100 0101 0110
- Эта команда поступает в УУ и дешифруется КО. В результате УУ определяет, какая команда будет выполняться и всем компонентам машины, участвующим в ее реализации, по каналам связи посылается соответствующая информация. После получения от всех компонент ответной информации об их готовности к выполнению операции в УУ, выделяется код адреса первого операнда, т.е. 0100, который пересылается в ОЗУ. В ОЗУ отыскивается соответствующая ячейка и ее содержимое пересылается в АЛУ.

Все эти действия сопровождаются посылкой компонентами, участвующими в операции, соответствующей информации в УУ.

Далее также обрабатывается второй адрес. В АЛУ образуется сумма $(a+b)$, получив эту информацию, УУ выделяет код последнего адреса и посылает его в ОЗУ. Как только УУ получает информацию, что ячейка с этим адресом найдена, УУ посылает в АЛУ управляющую информацию о посылке суммы в ОЗУ. Таким образом, в ячейке **0110** оказывается “ $a+b$ ”.

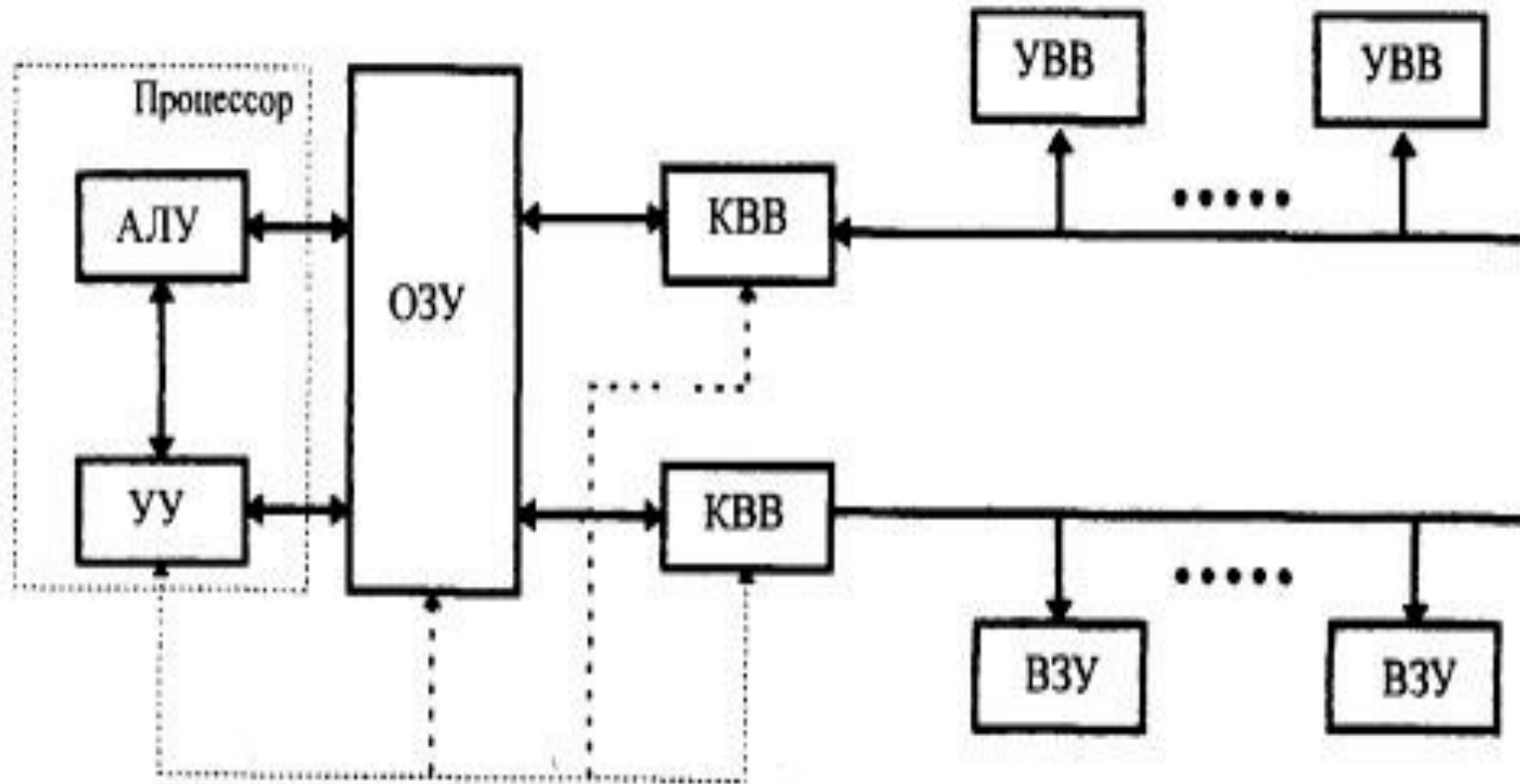
К этому моменту содержимое специального счетчика УУ – счетчика команд (СЧ) увеличивается на 1 ($i+1$).

Этот новый адрес посылается в ЗУ и начинается процесс выполнения следующей команды.

Если выполняется команда *безусловной передачи управления* другой команде программы, то в адресной части этой команды находится код адреса, который будет занесен в СЧ.

Если же выполняется команда *условной передачи управления*, то новый код адреса заносится в СЧ только при выполнении условия.

Изложенный процесс работы ЭВМ фон Неймановской архитектуры очень упрощен. На самом деле происходят более сложные процессы, все они по времени строго синхронизированы и в определенной степени совмещаются.



Основные параметры ЭВМ

- •технические и эксплуатационные характеристики ЭВМ (быстродействие и производительность, показатели надежности, достоверности, точности, емкость оперативной и внешней памяти, габаритные размеры, стоимость технических и программных средств, особенности эксплуатации и др.);
- •характеристики и состав функциональных модулей базовой конфигурации ЭВМ; возможность расширения состава технических и программных средств; возможность изменения структуры;
- • состав программного обеспечения ЭВМ и сервисных услуг (операционная система или среда, пакеты прикладных программ, средства автоматизации программирования).

СИСТЕМЫ СЧИСЛЕНИЯ

- **Система счисления** - способ наименования и представления чисел с помощью символов, имеющих определенные количественные значения. Системы счисления могут быть **непозиционные** и **позиционные**
- .
- В **непозиционной** системе счисления количественное значение символа не зависит от его позиции в ряду символов, изображающих это число. Примером такой системы является римская система счисления (вернее сказать, римская система является частично непозиционной, т. к. итоговое значение числа зависит от положения того или иного символа в числе, например: XI и IX).
- **Позиционные** системы счисления-системы, в которых количественное значение символа зависит от его позиции в ряду символов, изображающих это число. Данные системы удобны тем, что в них для записи числа требуется небольшое количество символов. Например: арабская система счисления.

- В **десятичной** системе счисления основанием системы является число 10. Для записи любого числа в этой системе используются цифры от 0 до 9.
- **Восьмеричная** система счисления применяется в ЭВМ в основном для составления программ, т. к. позволяет производить более короткую и удобную запись двоичных чисел. Основанием системы счисления является цифра 8, При указании,
 - системы счисления в этом случае можно использовать символ o: 67o.
- В **шестнадцатеричной** системе счисления используются цифры от 0 до 15. Первые десять цифр этой системы изображаются с помощью цифр от 0 до 9, а для остальных цифр, больших девяти, вводятся специальные обозначения в виде шести букв латинского алфавита A, B, C, A E,, F. При указании, системы счисления в этом случае можно использовать
 - символ h: 15Ah.

Системы счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная	Двоично-десятичная
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	00010000
11	1011	13	B	00010001
12	1100	14	C	00010010
13	1101	15	D	00010011
14	1110	16	E	00010100
15	1111	17	F	00010101
16	10000	20	10	00010110

ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

- **Пример 1.** Переведем число 976 из десятичной системы счисления в двоичную систему счисления ($976_{10} \rightarrow x_2$)

976	2																		
976	488	2																	
0	488	244	2																
	0	244	122	2															
		0	122	61	2														
			0	60	30	2													
				1	30	15	2												
					0	14	7	2											
						1	6	3	2										
							1	2	1	2									
								1	0	0	0								

$x_2 = 1111010000_2$

Пример 2.2.

Переведем число $(342_{10} \rightarrow x_8)$:

$$\begin{array}{r|l} 342 & 8 \\ \hline 336 & 42 & 8 \\ \hline 6 & 40 & 5 & 8 \\ & 2 & 0 & 0 \\ & & 5 & \end{array}$$

$$342_{10} = 526_8$$

Пример 2.3.

Переведем число $859_{10} \rightarrow x_{16}$:

$$\begin{array}{r|l} 859 & 16 \\ \hline 848 & 53 & 16 \\ \hline 11 & 48 & 3 & 16 \\ & 5 & 0 & 0 \\ & & 3 & \end{array}$$

$$859_{10} = 35B_{16}$$

Пример 2.10. Перевести число $111101,11_2$ в десятичную систему счисления.

$$111101,11_2 = (1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2})_{10} = 29 \frac{3}{4}$$

Преобразовать десятичные числа в двоичные

- а) $(52)_{10}$, в) $(298)_{10}$, д) $(321)_{10}$, з) $(46)_{10}$, к) $(71)_{10}$
- б) $(23)_{10}$, г) $(37)_{10}$, ж) $(65)_{10}$, и) $(64)_{10}$, е) $(17)_{10}$,

4. Преобразовать десятичные числа в восьмеричные: для дробных частей закончить преобразование на втором восьмеричном разряде.

а) $(15,19)_{10}$; в) $(42,24)_{10}$; д) $(6,98)_{10}$; ж) $(19,71)_{10}$, и) $(7,45)_{10}$

б) $(75,31)_{10}$; г) $(13,01)_{10}$; е) $(17,91)_{10}$, з) $(25,17)_{10}$, к) $\left(15\frac{6}{7}\right)_{10}$

8. Преобразовать двоичные числа в десятичные:

- а) 111010,01; г) 11000,011; ж) 10100,0111;
б) 1010,11; д) 10001,11; з) 10111,01.
в) 10101,101; е) 100011,1;

Пример 1.1.

Переведите число 10101_2 в десятичную систему счисления.

Пример 1.3.

Переведите число 239_{10} в пятиричную систему счисления.

Пример 1.4.

Переведите число 19_{10} в двоичную систему счисления.

Пример 1.5.

Переведите число 110101010_2 в восьмеричную систему счисления.

Пример 1.6.

Найдите сумму 1110101_2 и 1011011_2 . Ответ представьте в восьмеричной системе.

Пример 1.7.

Вычислите значение суммы в десятичной системе счисления:

$$11_4 + 11_8 + 11_{16} = ?_{10}$$

Прямой, обратный и дополнительный коды.

- Очень часто в вычислениях должны использоваться не только положительные, но и отрицательные числа. Число со знаком в вычислительной технике представляется путем представления старшего разряда числа в качестве **знакового**. Принято считать, что 0 в знаковом разряде означает знак «плюс» для данного числа, а 1 – знак «минус».
- **Прямой код** представляет собой одинаковое представление значимой части числа для положительных и отрицательных чисел и отличается только знаковым битом.
- **Обратный код** для положительных чисел имеет тот же вид, что и прямой код, а для отрицательных чисел образуется из прямого кода положительного числа путем инвертирования всех значащих разрядов прямого кода.
- **Дополнительный код** для положительных чисел имеет тот же вид, что и прямой код, а для отрицательных чисел образуется путем прибавления 1 к обратному коду. Добавление 1 к обратному коду числа 0 дает единое представление числа 0 в дополнительном коде

Число	Прямой код	Обратный код	Дополнительный код
-8	-	-	1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
00	10000000	11110000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

1. Выполнить сложение в модифицированном коде:

а) $0,101001+0,011011$

г) $0,100001+0,011111,$

б) $0,101011+0,010011$

д) $0,101010+0,000111,$

в) $0,110011+0,001111$

е) $0,101110+0,010111,$

ж) $0,100100+0,111011.$

2. Выполнить сложение, используя в каждом случае дополнительный и обратный модифицированные коды:

а) $0,110001+(-0,001011);$

г) $(-0,100111)+0,001001;$

б) $(-0,010111)+(-0,000001);$

д) $(-0,101010)+(-0,011011);$

в) $(-0,001010)+(-0,011011);$

е) $(-0,011101)+(0,001010);$

ж) $(-0,100101)+ (0,010111).$

- Что такое алгебра логики?
- **Алгебра логики** — это математический аппарат, с помощью которого
- записывают, вычисляют, упрощают и преобразовывают логические высказывания.
- Создателем алгебры логики является живший в XIX веке английский математик Джордж Буль, в честь которого эта алгебра названа булевой алгеброй высказываний.
- **Логическое высказывание** — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

- Так, например, предложение “6 — четное число” следует считать высказыванием, так как оно истинное. Предложение “Рим — столица Франции” тоже высказывание, так как оно ложное.
- Разумеется, не всякое предложение является логическим высказыванием. Высказываниями не являются, например, предложения “ученик десятого класса” и “информатика — интересный предмет”.
- Первое предложение ничего не утверждает об ученике, а второе использует слишком неопределённое понятие “интересный предмет”.
- Вопросительные и восклицательные предложения также не являются высказываниями, поскольку говорить об их истинности или ложности не имеет смысла.
- Предложения типа “в городе А более миллиона жителей”, “у него голубые глаза” не являются высказываниями, так как для выяснения их истинности или ложности нужны дополнительные сведения: о каком конкретно городе или человеке идет речь. Такие предложения называются высказывательными формами.

- **Высказывательная форма** — это повествовательное предложение,
- которое прямо или косвенно содержит хотя бы одну переменную и
- становится высказыванием, когда все переменные замещаются своими
- значениями.
- Алгебра логики рассматривает любое высказывание только с одной
- точки зрения — является ли оно истинным или ложным. Заметим, что
- зачастую трудно установить истинность высказывания. Так, например,
- высказывание “площадь поверхности Индийского океана равна 75 млн кв.
- км” в одной ситуации можно посчитать ложным, а в другой — истинным.
- Ложным — так как указанное значение неточное и вообще не является
- постоянным. Истинным — если рассматривать его как некоторое
- приближение, приемлемое на практике.
- Употребляемые в обычной речи слова и словосочетания “**не**”, “**и**”,
- “**или**”, “**если... , то**”, “**тогда и только тогда**” и другие позволяют

- Высказывания, образованные из других высказываний с помощью логических связок, называются **составными**. Высказывания, не являющиеся составными, называются **элементарными**.
- Так, например, из элементарных высказываний “Петров — врач”, “Петров — шахматист” при помощи связки “и” можно получить составное высказывание “Петров — врач и шахматист”, понимаемое как “Петров — врач, хорошо играющий в шахматы”. При помощи связки “или” из этих же высказываний можно получить составное высказывание “Петров — врач или шахматист”, понимаемое в алгебре логики как “Петров **или** врач, **или** шахматист, **или** и врач и шахматист одновременно”.

- Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.
- Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через **A** обозначено высказывание “Тимур поедет летом на море”, а через **B** — высказывание “Тимур летом отправится в горы”. Тогда составное высказывание “Тимур летом побывает и на море, и в горах” можно кратко записать как **A и B**. Здесь “и” — логическая связка, **A, B** — логические переменные, которые могут принимать только два значения — “истина”
- или “ложь”, обозначаемые, соответственно, “1” “0”

- 1. Операция, выражаемая словом **“не”**, называется **отрицанием** и обозначается чертой над высказыванием (или знаком \neg).
Высказывание истинно, когда **A** ложно, и ложно, когда **A** истинно. Пример. “Луна — спутник Земли” (**A**); “Луна — не спутник Земли” ($\neg A$).
- 2. Операция, выражаемая связкой **“и”**, называется **конъюнкцией** (лат. *conjunctio* — **соединение**) или **логическим умножением** и **обозначается точкой “•”** (может также обозначаться знаками \wedge или $\&$). Высказывание **A•B** истинно тогда и только тогда, когда оба высказывания **A** и **B** истинны.
- Например, высказывание “10 делится на 2 и 5 больше 3” истинно, а высказывания “10 делится на 2 и 5 не больше 3”, “10 не делится на 2 и 5 больше 3”, “10 не делится на 2 и 5 не больше 3” ложны.

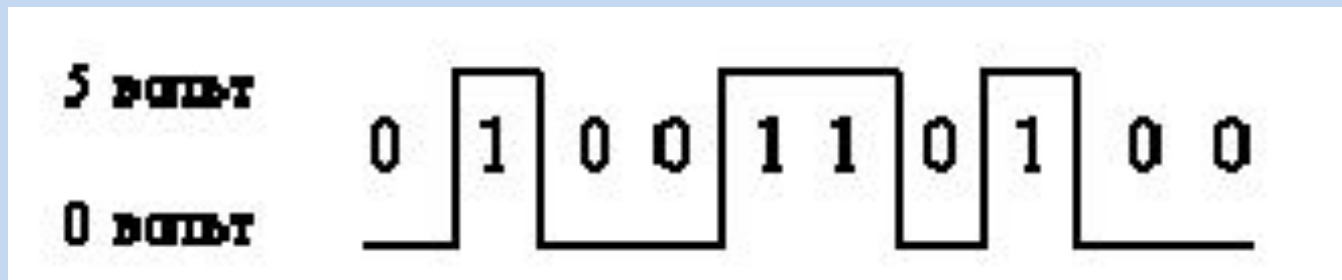
- 3. Операция, выражаемая связкой “или” (в неразделительном, неисключающем смысле этого слова), называется **дизъюнкцией** (лат. **disjunctio — разделение**) или логическим сложением и обозначается знаком \vee (**или плюсом**). Высказывание $A \vee B$ ложно тогда и только тогда, когда оба высказывания **A** и **B** ложны. Например, высказывание “10 не делится на 2 или 5 не больше 3” ложно, а высказывания “10 делится на 2 или 5 больше 3”, “10 делится на 2 или 5 не больше 3”, “10 не делится на 2 или 5 больше 3” истинны.
- 4. Операция, выражаемая связками “если ..., то”, “из ... следует”, “... влечет ...”, называется **импликацией** (лат. **implicatio — тесно связаны**) и обозначается знаком \supset . Высказывание $A \supset B$ ложно тогда и только тогда, когда **A** истинно, а **B** — ложно.

- 5. Операция, выражаемая связками “тогда и только тогда”, “необходимо и достаточно”, “... равносильно ...”, называется эквиваленцией или двойной импликацией и обозначается знаком « или \sim . Высказывание **A** « **B** истинно тогда и только тогда, когда значения **A** и **B** совпадают. Например, высказывания “24 делится на 6 тогда и только тогда, когда 24 делится на 3”, “23 делится на 6 тогда и только тогда, когда 23 делится на 3” истинны, а высказывания “24 делится на 6 тогда и только тогда, когда 24 делится на 5”, “21 делится на 6 тогда и только тогда, когда 21 делится на 3” ложны.

- **Определение логической формулы:**
- **1. Всякая логическая переменная и символы "истина" ("1") и "ложь" ("0") - формулы.**
- **2. Если A и B — формулы, то $(A \cdot B)$, $(A \vee B)$, $(A \supset B)$, $(A \ll B)$ — формулы.**
- **3. Никаких других формул в алгебре логики нет.**

- Какая связь между алгеброй логики и двоичным кодированием?
- Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: "1" и "0".
- Из этого следует два вывода:
 - 1. одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
 - 2. на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

- Данные и команды представляются в виде двоичных последовательностей различной структуры и длины. Существуют различные физические способы кодирования двоичной информации, но чаще всего единица кодируется более высоким уровнем напряжения, чем ноль (или наоборот), например:



- Что такое логический элемент компьютера?
- Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию. Логическими элементами компьютеров являются электронные схемы **И**, **ИЛИ**, **НЕ**, **И-НЕ**, **ИЛИ-НЕ** и другие (называемые также вентилями), а также триггер. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентиляей бывает от двух до восьми входов и один или два выхода.
- Чтобы представить два логических состояния — “1” и “0” в вентилях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт.

- Высокий уровень обычно соответствует значению “истина” (“1”), а низкий — значению “ложь” (“0”).
- Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.
- Работу логических элементов описывают с помощью таблиц истинности.
- Таблица истинности это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

- **Что такое схемы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ?**
- **Схема И.** Схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы И с двумя входами представлено на рисунке

Карта Карно

Карта Карно́ — графический способ минимизации переключательных (булевых) функций, обеспечивающий относительную простоту работы с большими выражениями и устранение потенциальных гонок. Представляет собой операции попарного неполного склеивания и элементарного поглощения. Карты Карно рассматриваются как перестроенная соответствующим образом таблица истинности функции. Карты Карно можно рассматривать как определенную плоскую развертку n -мерного булева куба.

Принципы минимизации

Основным методом минимизации логических функций, представленных в виде СДНФ или СКНФ, является операция попарного неполного склеивания и элементарного поглощения. Операция попарного склеивания осуществляется между двумя термами (членами), содержащими одинаковые переменные, вхождения которых (прямые и инверсные) совпадают для всех переменных, кроме одной. В этом случае все переменные, кроме одной, можно вынести за скобки, а оставшиеся в скобках прямое и инверсное вхождение одной переменной подвергнуть склейке. Например:

$$\bar{X}_1 X_2 X_3 X_4 \vee \bar{X}_1 X_2 \bar{X}_3 X_4 = \bar{X}_1 X_2 X_4 (X_3 \vee \bar{X}_3) = \bar{X}_1 X_2 X_4.$$

Аналогично для КНФ:

$$(\bar{X}_1 \vee X_2 \vee X_3 \vee X_4)(\bar{X}_1 \vee X_2 \vee \bar{X}_3 \vee X_4) = \bar{X}_1 \vee X_2 \vee X_4 \vee X_3 \bar{X}_3 = \bar{X}_1 \vee X_2 \vee X_4.$$

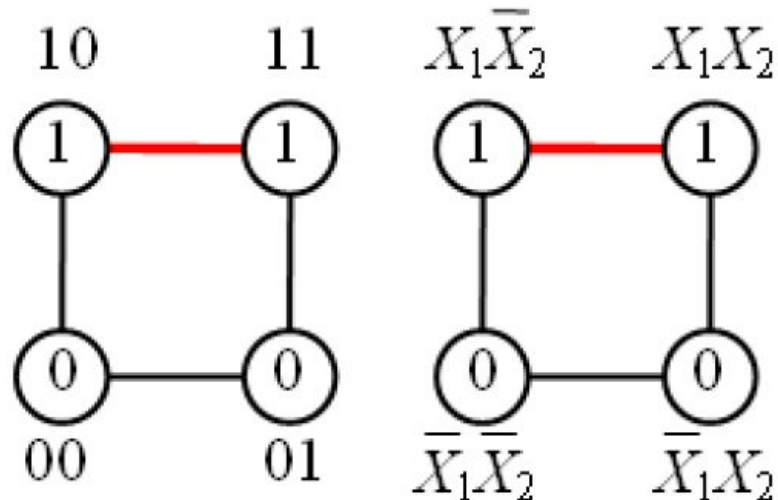
Возможность поглощения следует из очевидных равенств

$$A \vee \bar{A} = 1; A\bar{A} = 0.$$

Таким образом, главной задачей при минимизации СДНФ и СКНФ является поиск термов, пригодных к склейке с последующим поглощением, что для больших форм может оказаться достаточно сложной задачей. Карты Карно предоставляют наглядный способ отыскания таких термов.

- Как известно, булевы функции N переменных, представленные в виде СДНФ или СКНФ, могут иметь в своём составе 2^N различных термов. Все эти члены составляют некоторую структуру, топологически эквивалентную N -мерному кубу, причём любые два терма, соединённые ребром, пригодны для склейки и поглощения.
- На рисунке изображена простая таблица истинности для функции из двух переменных, соответствующий этой таблице 2-мерный куб (квадрат), а также 2-мерный куб с обозначением членов СДНФ и эквивалентная таблица для группировки термов:

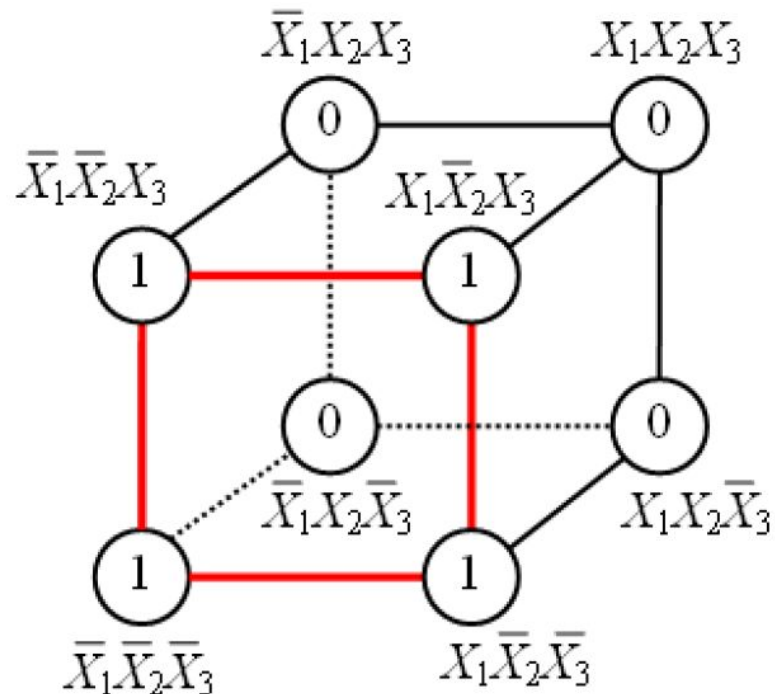
X_1	X_2	$f(X_1, X_2)$
0	0	0
0	1	0
1	0	1
1	1	1



		X_2	
		0	1
X_1	0	1	1
	1	0	0

- В случае функции трёх переменных приходится иметь дело с трёхмерным кубом. Это сложнее и менее наглядно, но технически возможно. На рисунке в качестве примера показана таблица истинности для булевой функции трёх переменных и соответствующий ей куб.

X_1	X_2	X_3	$f(X_1, X_2, X_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



- Карта Карно может быть составлена для любого количества переменных, однако удобно работать при количестве переменных не более пяти. По сути Карта Карно — это таблица истинности составленная в 2-х мерном виде. Благодаря использованию кода Грея в ней верхняя строка является соседней с нижней, а правый столбец соседний с левым, т.о. вся Карта Карно сворачивается в фигуру тор (бублик). На пересечении строки и столбца проставляется соответствующее значение из таблицы истинности. После того как Карта заполнена, можно приступать к минимизации.
- Если необходимо получить минимальную ДНФ, то в Карте рассматриваем только те клетки которые содержат единицы, если нужна КНФ, то рассматриваем те клетки, которые содержат нули. Сама минимизация производится по следующим правилам (на примере ДНФ):
 - 1. Объединяем смежные клетки, содержащие единицы, в область так, чтобы одна область содержала (целое число = $0 \dots$) клеток (помним про то, что крайние строки и столбцы являются соседними между собой), в области не должно находиться клеток, содержащих нули;
 - 2. Область должна располагаться симметрично оси(ей) (оси располагаются через каждые четыре клетки);
 - 3. Несмежные области, расположенные симметрично оси(ей), могут объединяться в одну;
 - 4. Область должна быть как можно больше, а количество областей как можно меньше;
 - 5. Области могут пересекаться;
 - 6. Возможно несколько вариантов покрытия.

- **Пример 1**
- У мальчика Коли есть мама, папа, дедушка и бабушка. Коля пойдёт гулять на улицу, если ему разрешат хотя бы двое родственников.
- Для краткости обозначим родственников Коли через буквы:
- мама — x_1
- папа — x_2
- дедушка — x_3
- бабушка — x_4
- Условимся обозначать согласие родственников единицей, несогласие - нулём. Возможность пойти погулять обозначим буквой f , Коля идёт гулять — $f = 1$, Коля гулять не идёт — $f = 0$.
- Составим таблицу истинности:

X1	X2	X3	X4	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Перерисуем таблицу истинности в 2-х мерный вид:

X1 X2 \ X3 X4		X3 X4			
		00	01	10	11
X1 X2	00				
	01				
	10				
	11				

Переставим в ней строки и столбцы в соответствии с кодом Грея. Получили Карту Карно:

		X3 X4			
		00	01	11	10
X1 X2	00				
	01				
	11				
	10				

Заполним её значениями из таблицы истинности:

		X3 X4			
		00	01	11	10
X1 X2	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

Минимизируем

в соответствии с правилами:

X1 X2 \ X3 X4		00		01		11		10	
		00	01	11	10	00	01	11	10
00	00	0	0	1	0	S3			
01	00	0	1	1	1	S6			
11	00	1	1	1	1	S2			
10	00	0	1	1	1	S5			
						S4		S1	

вариант только один.

1. Все области содержат 2^n клеток;
2. Так как Карта Карно на четыре переменные, оси располагаются на границах Карты и их не видно (подробнее смотри пример Карты на 5 переменных);
3. Так как Карта Карно на четыре переменные, все области симметрично осей — смежные между собой (подробнее смотри пример Карты на 5 переменных);
4. Области S3, S4, S5, S6 максимально большие;
5. Все области пересекаются (необязательное условие);
6. В данном случае рациональный

Пример Карты Карно на пять переменных

Имеем такую таблицу истинности:

X1	X2	X3	X4	X5	f
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	1

Карта Карно будет выглядеть следующим образом (для лучшего визуального восприятия в Карту нули не записываем):

X1 X2 \ X3 X4 X5	000	001	011	010	110	111	101	100
00	1			1	1	1	1	
01	1			1				
11			1			1		
10	1		1			1		1

(на

примере

Неправильно
ДНФ):

X1 X2 \ X3 X4 X5	000	001	011	010	110	111	101	100
00	1			1	1	1	1	
01	1			1				
11			1			1		
10	1		1			1		1

S2 (grouping 000, 001)
S3 (grouping 010, 110, 101)
S1 (grouping 010, 011)
S4 (grouping 110, 101)
S5 (grouping 101, 100)
S6 (grouping 011, 101)

Элементная база ЭВМ

- ЦВМ состоит из множества элементов, взаимодействие между которыми осуществляется путём обмена сигналами. В общем случае под сигналом можно понимать физический процесс, несущий информацию или представляющий интерес для наблюдателя. Физически сигнал в ЦВМ представляется уровнем электрического напряжения. Различают сигнал аналоговый и цифровой. Под аналоговым понимается сигнал, для которого важным считается конкретное значение напряжения, которое может принимать любую величину в заданном диапазоне значений. Под цифровым сигналом понимают электрический сигнал, для которого справедливым или допустимым являются лишь два значения: 0 и 1. Различают два способа представления цифровой информации. При импульсном представлении за 0 принимается тот сигнал, которому соответствует отсутствие импульса (нулевой уровень напряжения). Двоичная единица в этом случае, представляется одним или множеством импульсов электрического напряжения. Под импульсом можно понимать быстрое кратковременное изменение электрического напряжения. При потенциальном способе представления двоичной информации нулю и единице соответствуют два чётко различимых уровня напряжения: U^0 и U^1 . Для современных логических элементов обычно $U^0 \approx 0\text{В}$, $U^1 > 0\text{В}$. Для элементов серии ТТЛ (транзисторно-транзисторная логика): $U^1 > 2,4\text{В}$, $U^0 \leq 0,4\text{В}$

- **Под элементом понимается наименьшая функционально и конструктивно законченная часть ЦВМ, которая выполняет какую-либо логическую (булеву) функцию (логический элемент) или какое-либо другое преобразование информации. В настоящее время элементы реализуют в составе микросхем. Микросхема — это микроэлектронное изделие с повышенной степенью интеграции. Будем считать, что понятия микросхема и интегральная схема равноценны. В зависимости от используемых сигналов различают элементы и микросхемы аналоговые, аналого-цифровые и цифровые. В зависимости от степени интеграции, то есть от числа элементов в составе микросхемы, различают микросхемы малой, средней, большой и сверхбольшой степени интеграции.**

- В зависимости от технологии изготовления различают микросхемы, выполненные по **интегральной, гибридной или плёночной технологии**. Интегральная технология предполагает выполнение всех деталей в одном кристалле полупроводника (чаще всего – кремния). Гибридная технология предполагает применение интегральных элементов в бескорпусном варианте, которые монтируются на изоляторе и соединяются с помощью напылённых проводников. Плёночная технология предполагает применение плёнок даже при изготовлении активных компонентов.

- В зависимости от базовой схемы, которая используется при реализации элементов в микросхеме, различают микросхемы, выполненные по следующим технологиям:
- 1. ДТЛ - диодно-транзисторная логика;
- 2. ЭСЛ – эмиттерно-связанная логика;
- 3. ТТЛ – транзисторно-транзисторная логика;
- 4. ТТЛШ – ТТЛ с диодами Шоттки;
- 5. МОП (металл- окись- полупроводник);
- 6. МДП (металл- диэлектрик- полупроводник); 7. КМОП – комплементарные МОП;
- 8. КМДП - комплементарные МДП;

Физическая реализация логических функций

- Любое цифровое устройство можно создать, используя простейшие логические элементы, если выполняемые ими логические выражения являются функционально полными. Сколь угодно сложное устройство можно реализовать на элементах, выполняющих логическую функцию типа И-НЕ или ИЛИ-НЕ. То есть, если имеются логические двухвходовые элементы, выполняющие функцию «конъюнкция-отрицание» ($y = x_1 \& x_2$) или «дизъюнкция-отрицание» ($y = x_1 \vee x_2$), то на них можно построить сколь угодно сложное цифровое устройство. Обратим внимание на способы реализации отмеченных функций. При анализе предлагаемых ниже схем будем считать, что уровень логической единицы выше уровня логического нуля. Для физической реализации выделенных функций в простейшем случае можно использовать транзисторную логику.

- При реализации транзисторной логики используют параллельное и последовательное включение транзисторов. Наличие транзисторов приводит к тому, что кроме функции типа «И», «ИЛИ» элементы автоматически выполняют функцию «НЕ». Рассмотрим реализацию транзисторной логики на биполярных транзисторах, поскольку принцип работы элементов на МДП-транзисторах такой же. Параллельное включение транзисторов предложено на рисунке 1.5а. Только при наличии на всех входах уровня логического нуля транзисторы окажутся закрытыми, на выходе наблюдается уровень логической единицы. То есть элемент реализует функцию «2ИЛИ-НЕ», где цифра 2 указывает на число входов дизъюнктора:

Основные параметры цифровых элементов и микросхем

- Чтобы успешно использовать элементы в составе цифровых устройств необходимо учитывать хотя бы некоторые параметры используемых элементов. Выделим основные параметры, учёт значений которых позволяет создавать работоспособные цифровые устройства и определять их быстродействие и входные и выходные допустимые токи в статике.
- 1. Логические уровни - это уровни напряжения, которые соответствуют нулю и единице - U^1 и U^0 .
- 2. Пороговое напряжение - это граница раздела нуля и единицы в бесконечной цепочке элементов.
- 3. Максимальная величина входного тока определяется при нуле и единице на входе.
- 4. Максимальный выходной ток элемента определяет его нагрузочную способность. Выходные токи также определяются при уровнях логического нуля и логической единицы на выходе элемента.
- 5. Нагрузочная способность в пределах серии элементов определяется максимально допустимым количеством элементов (а точнее, входов элементов) той же серии, которые могут быть подключены к выходам элементов. 6. Быстродействие элементов определяется временем задержки информации и продолжительности фронтов сигналов на выходе.

Разновидности электрических схем

- **Электрическая схема – способ изображения цифрового изделия на плоскости с использованием условных графических обозначений (УГО) элементов и узлов.**
- **Вычислительная техника использует 3 разновидности электрических схем:**
 - 1) **схема электрическая структурная (Э1);**
 - 2) **схема электрическая функциональная (Э2);** 3) **схема электрическая принципиальная (Э3).**

2.10. Основные характеристики логических элементов

Основными характеристиками логических элементов являются: статическая характеристика передачи, статическая помехоустойчивость, быстродействие, число входов или коэффициент объединения по входу, нагрузочная способность или коэффициент разветвления по выходу, потребляемая мощность.

Статическая характеристика передачи (рис. 2.11) представляет собой зависимость напряжения $U_{\text{вых}}$ на выходе ЛЭ от напряжения $U_{\text{вх}}$ на одном из его входов ($U_{\text{вых}} = f(U_{\text{вх}})$). При этом напряжения на других входах поддерживаются постоянными и соответствующими уровню логического нуля для схем ИЛИ-НЕ и уровню логической единицы для схем И-НЕ. В этом случае ЛЭ превращается в инвертор. К выходу исследуемого элемента подключается в качестве нагрузки однотипный элемент.

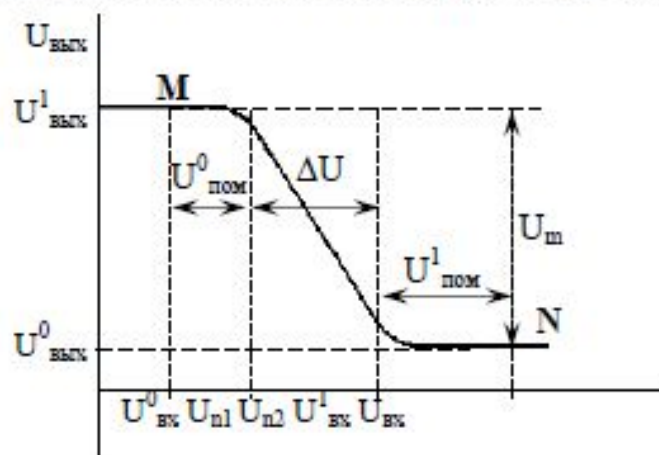


Рис. 2.11. Статическая характеристика передачи ЛЭ

Основные параметры статической характеристики передачи: уровни напряжения U^1 и U^0 , равные соответственно логической единице и нулю; логический размах

или перепад напряжения U_m ; пороговые уровни $U_{н1}$ и $U_{н2}$ и ширина активной области ΔU . Отношение $U_m/\Delta U$ представляет собой значение среднего коэффициента передачи в активной области. Точки М и N характеризуют положение рабочей точки на статической характеристике передачи при подаче на вход ЛЭ уровней напряжения U^0 и U^1 .

Статическая помехоустойчивость – определяется наибольшей величиной напряжений $U_{пом}^0$ и $U_{пом}^1$ (рис. 2.11), которые могут быть поданы на вход элемента относительно логических уровней 0 и 1 и не вызовут ложных переключений. Причиной таких помех могут быть паразитные падения напряжения на шинах питания. В практических схемах значение $U_{пом}$ колеблется от 0,1 до 0,3 В в элементах с низкой помехоустойчивостью и до 1,0 В – в элементах с высокой помехоустойчивостью.

Быстродействие в логических элементах определяется величиной задержки перепада напряжения при переходе его через ЛЭ. Эта задержка определяется наличием некоторого порога срабатывания элемента, инерционностью полупроводниковых приборов, влиянием паразитных емкостей. Она обычно измеряется на уровне, равном половине величины перепада и оказывается различной для положительного (t_3^+) и отрицательного (t_3^-) перепадов (рис. 2.12).

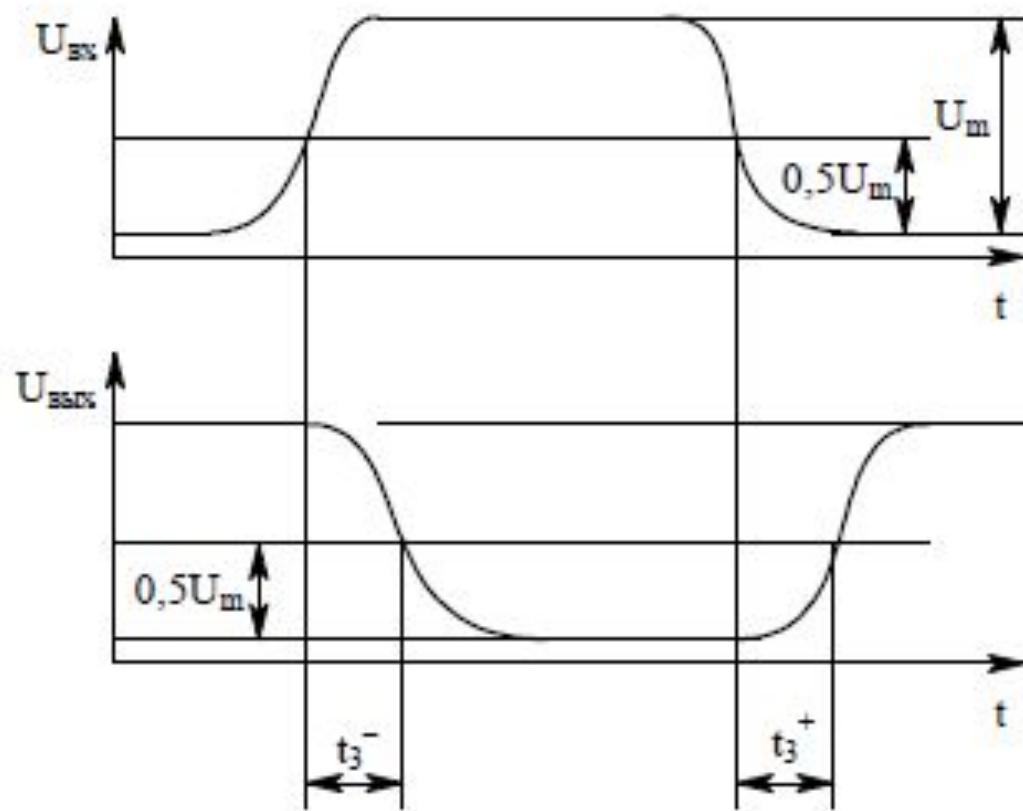


Рис. 2.12. Определение среднего времени задержки ЛЭ

Средней задержкой называют их полусумму: $t_{3,\text{cp}} = (t_3^+ + t_3^-)/2$.

Быстродействие логической схемы тем выше, чем меньше $t_{з.ср.}$. Все логические элементы по быстродействию можно разделить условно на 4 группы: сверхбыстродействующие – $t_{з.ср.} < 5$ нс; быстродействующие – $t_{з.ср.} = 5 - 10$ нс; среднебыстродействующие – $t_{з.ср.} = 11 - 15$ нс; медленнодействующие – $t_{з.ср.} > 15$ нс.

Коэффициент объединения по входу или число входов (m) логического элемента определяет максимальное количество входных сигналов, над которыми можно произвести операции ИЛИ и И, в зависимости от типа элемента $m = 2 - 12$.

Нагрузочная способность или *коэффициент разветвления по выходу* (n) характеризует число входов аналогичных элементов, которые можно подключить к выходу данного элемента без нарушения его нормального функционирования. В зависимости от типа элемента n изменяется от 3 до 100.

Потребляемая мощность рассеивания в ЛЭ определяет не только его экономичность, но, что часто более важно, – степень его разогрева. Это ограничивает габариты элемента и всего устройства в целом. Уменьшение габаритов элемента и, следовательно, величины его охлаждающей поверхности при заданной величине его рассеиваемой мощности приводит к росту температуры элемента сверх допустимого значения и к нарушению его работоспособности. Это обстоятельство оказывается особенно существенным в микроминиатюрных интегральных элементах. В сложном цифровом устройстве примерно половина входящих элементов в любой момент времени оказывается закрытой, а другая половина – открытой. Поэтому в качестве характеристики элемента пользуются понятием средней мощности P_{cp} , под которой понимается полусумма мощностей, рассеиваемых в двух статических состояниях элемента: закрытом и открытом. Величина P_{cp} достигает сотен милливатт.

2.11. Резисторно-транзисторные логические элементы

Схема базового элемента РТЛ изображена на рис. 2.13. Ее основой является простейший ключ – инвертор, а операция ИЛИ реализуется входной сборкой резисторов R1, R2, R3. Схема относительно проста, но обладает рядом недостатков, основным из которых является сильная зависимость глубины насыщения транзистора Т1 от числа логических единиц на входе, что приводит к ухудшению динамических характеристик транзистора. Эта же причина ограничивает функциональную сложность

ЛЭ – уменьшает коэффициент объединения по входу и коэффициент разветвления по выходу. Развитие интегральной схемотехники и технологии привело к тому, что в настоящее время система РТЛ практически не применяется.

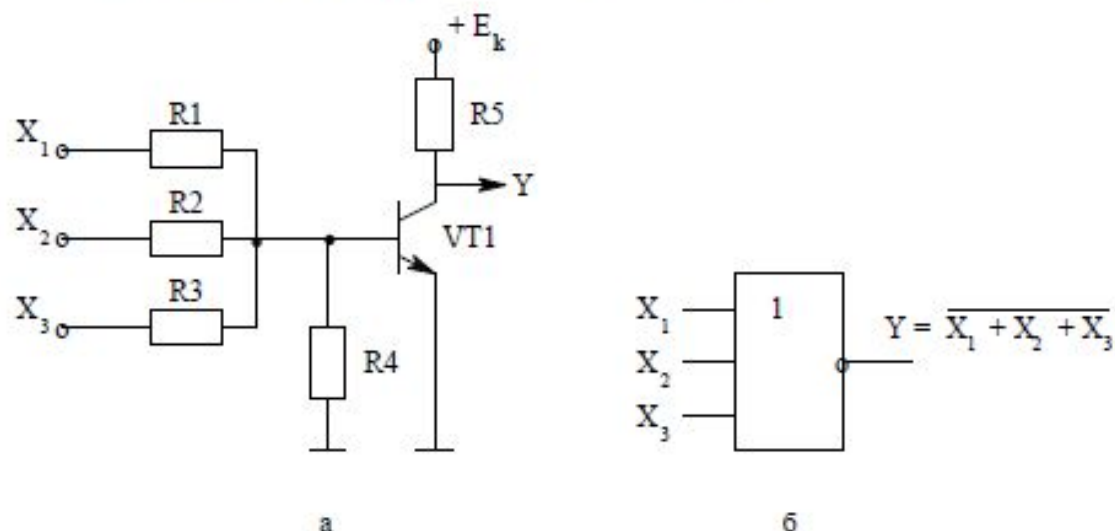


Рис. 2.13. Принципиальная схема (а) и функциональное обозначение (б)
РТЛ элемента

Диодно-транзисторные логические элементы

Диодно-транзисторные логические элементы находят широкое применение в цифровых устройствах. Они имеют повышенную помехоустойчивость, среднее быстродействие и сравнительно большой коэффициент разветвления по выходу n . Основная схема ДТЛ-элемента, выполняющая логическую функцию И-НЕ, дана на рис. 2.14.

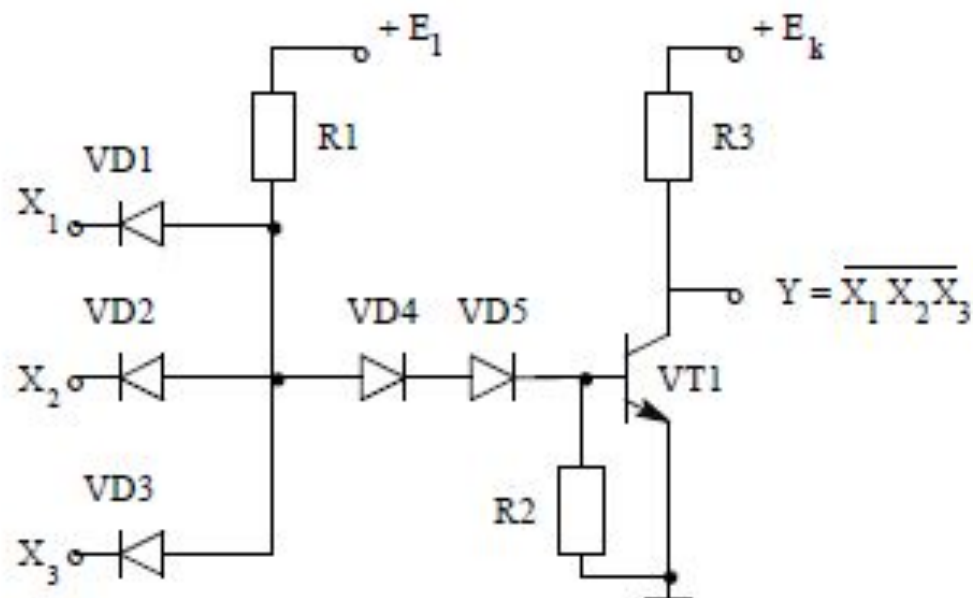


Рис. 2.14. Принципиальная схема ДТЛ-элемента

2.12. Транзисторные элементы с непосредственной связью (НСТЛ)

Схема НСТЛ-элемента приведена на рис. 2.15.

Схема ЛЭ образована соединением коллекторов нескольких (на рисунке – трех) транзисторов. Здесь применена непосредственная связь выхода предыдущего каскада со входом последующего. Если на все входы поступают низкие уровни напряжения (логический ноль), то все три транзистора VT1 – VT3 закрыты и на выходе напряжение равно $+E_k$ (без учета подключения ЛЭ к нагрузке), что соответствует уровню логической единицы.

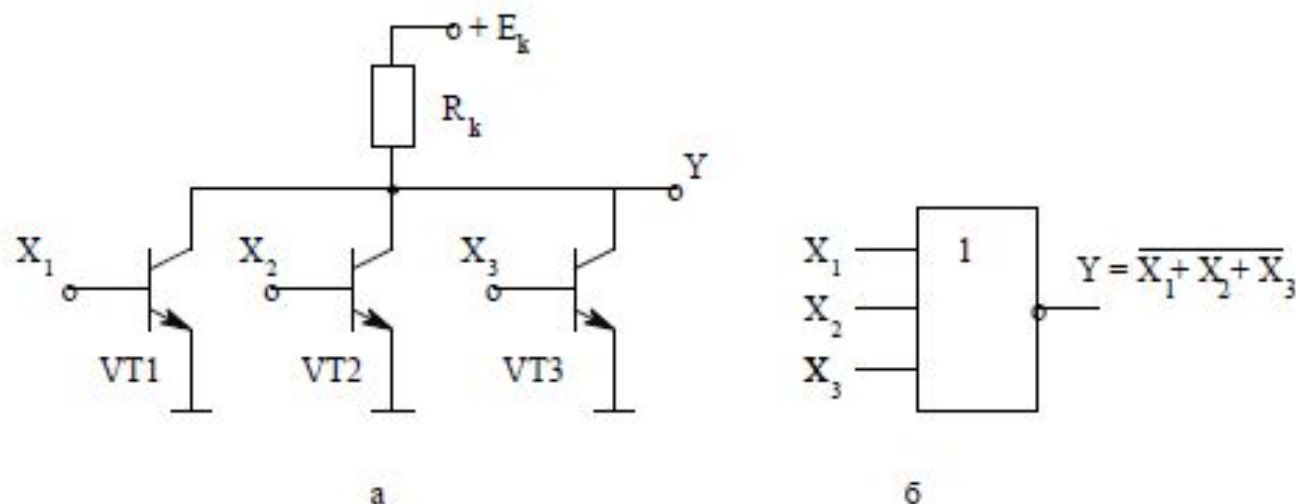


Рис. 2.15. Принципиальная схема (а) и функциональное обозначение (б)

НСТЛ-элемента

2.10. Основные характеристики логических элементов

Основными характеристиками логических элементов являются: статическая характеристика передачи, статическая помехоустойчивость, быстродействие, число входов или коэффициент объединения по входу, нагрузочная способность или коэффициент разветвления по выходу, потребляемая мощность.

Статическая характеристика передачи (рис. 2.11) представляет собой зависимость напряжения $U_{\text{ВЫХ}}$ на выходе ЛЭ от напряжения $U_{\text{ВХ}}$ на одном из его входов ($U_{\text{ВЫХ}} = f(U_{\text{ВХ}})$). При этом напряжения на других входах поддерживаются постоянными и соответствующими уровню логического нуля для схем ИЛИ-НЕ и уровню логической единицы для схем И-НЕ. В этом случае ЛЭ превращается в инвертор. К выходу исследуемого элемента подключается в качестве нагрузки однотипный элемент.

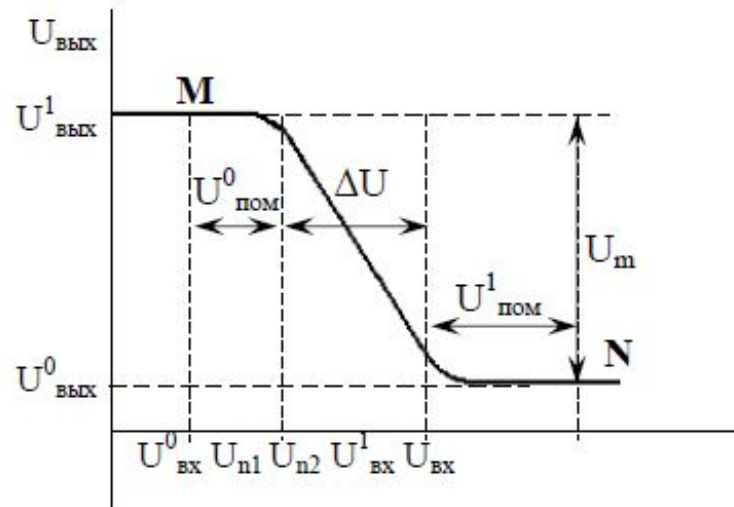


Рис. 2.11. Статическая характеристика передачи ЛЭ

- *Быстродействие* в логических элементах определяется величиной задержки перепада напряжения при переходе его через ЛЭ. Эта задержка определяется наличием некоторого порога срабатывания элемента, инерционностью полупроводниковых приборов, влиянием паразитных емкостей. Она обычно измеряется на уровне, равном половине величины перепада и оказывается различной для положительного (t_3^+) и отрицательного (t_3^-) перепадов

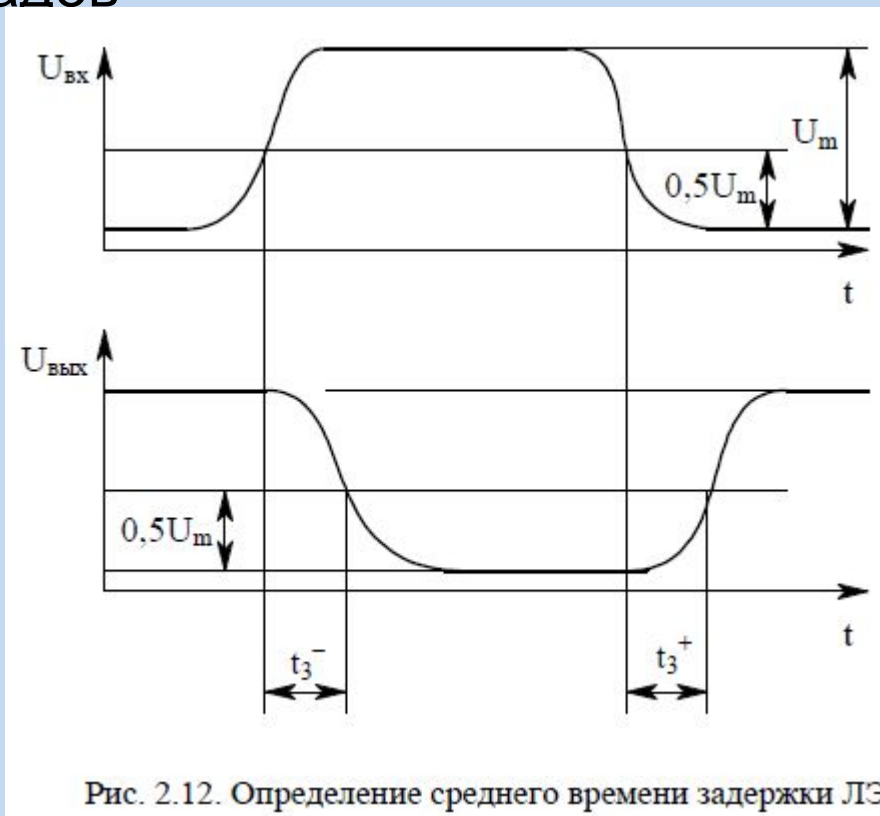


Рис. 2.12. Определение среднего времени задержки ЛЭ

Резисторно-транзисторная логика

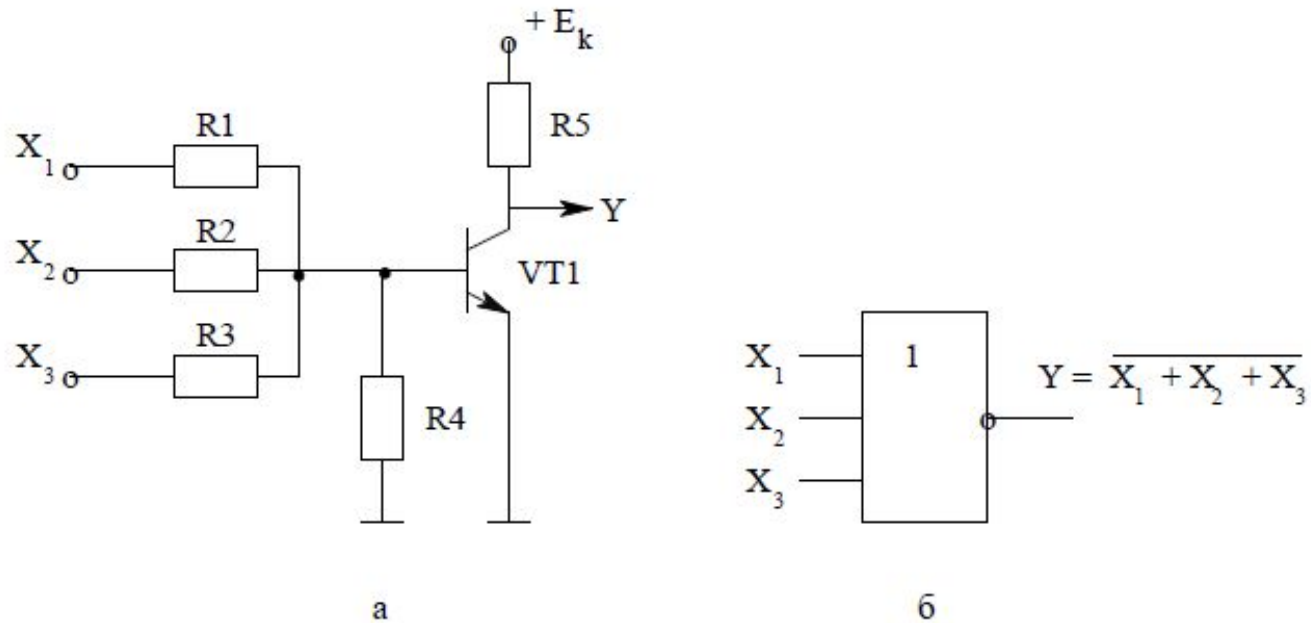


Рис. 2.13. Принципиальная схема (а) и функциональное обозначение (б)

РТЛ элемента

Диодно-транзисторные логические элементы

Диодно-транзисторные логические элементы находят широкое применение в цифровых устройствах. Они имеют повышенную помехоустойчивость, среднее быстродействие и сравнительно большой коэффициент разветвления по выходу n . Основная схема ДТЛ-элемента, выполняющая логическую функцию И-НЕ, дана на рис. 2.14.

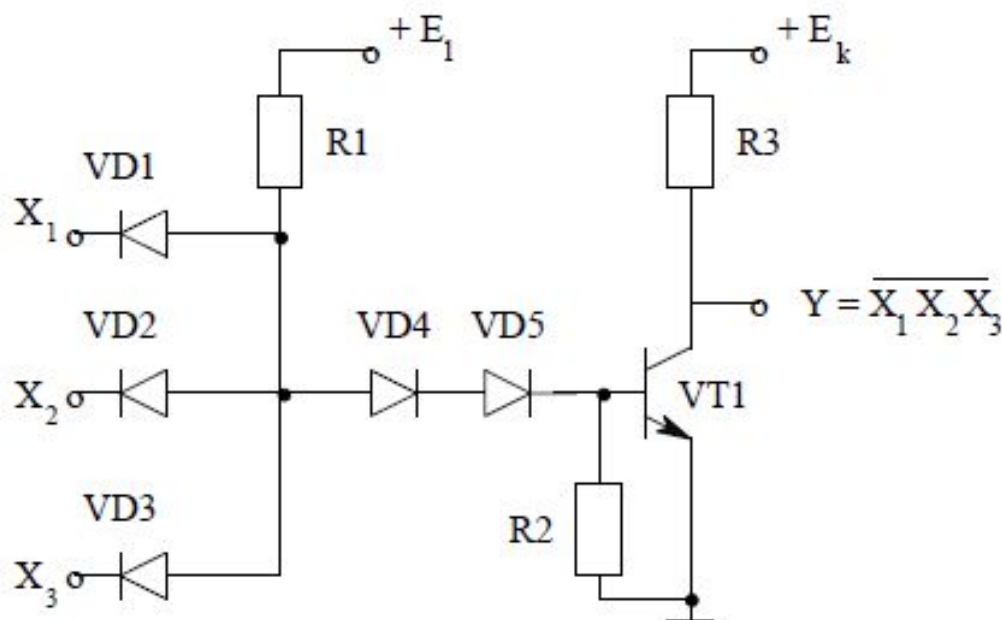


Рис. 2.14. Принципиальная схема ДТЛ-элемента

2.13. Транзисторно-транзисторные логические элементы

Схема ТТЛ ЛЭ, выполняющего логическую функцию И-НЕ, серии 155 представлена на рис. 2.16. Схема содержит входной каскад, реализующий функцию И, фазоинверсный каскад с источником тока и выходной каскад с активной нагрузкой.

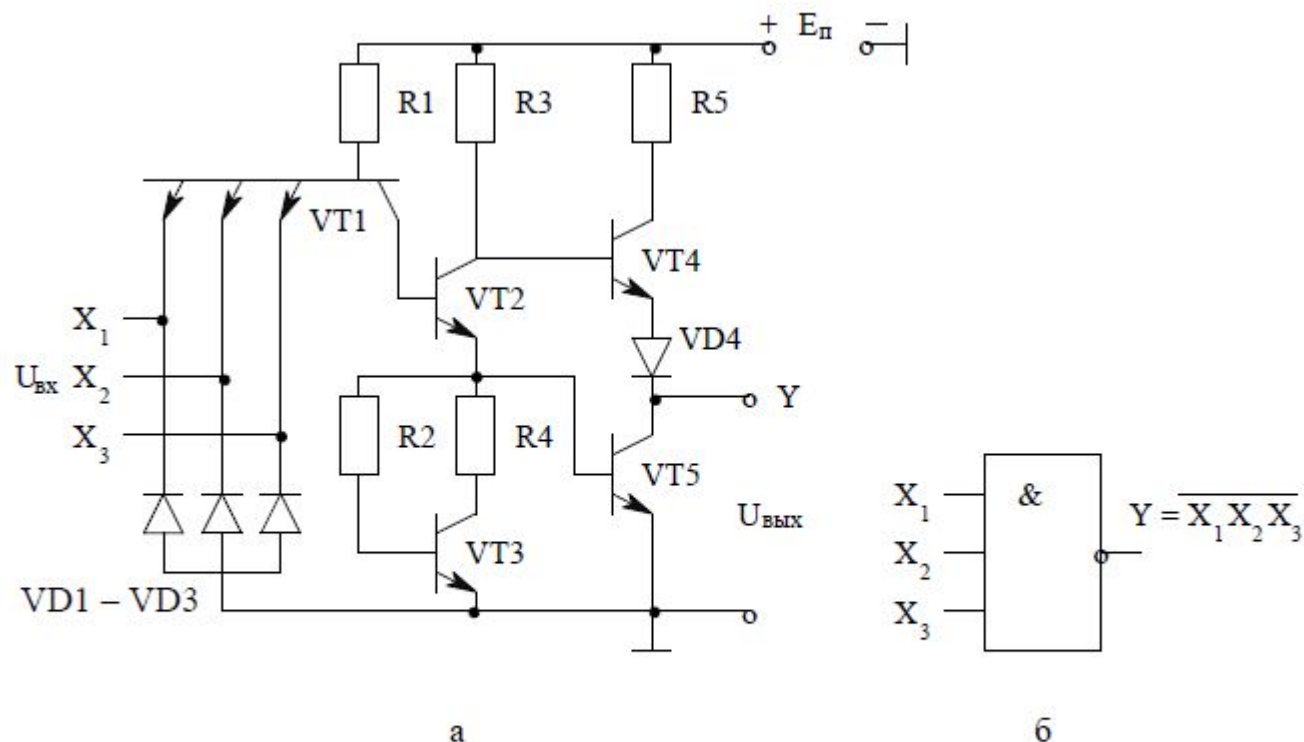


Рис. 2.16. Принципиальная схема (а) ТТЛ-элемента серии 155 и его функциональное обозначение (б)

2.13. Транзисторно-транзисторные логические элементы

Схема ТТЛ ЛЭ, выполняющего логическую функцию И-НЕ, серии 155 представлена на рис. 2.16. Схема содержит входной каскад, реализующий функцию И, фазоинверсный каскад с источником тока и выходной каскад с активной нагрузкой.

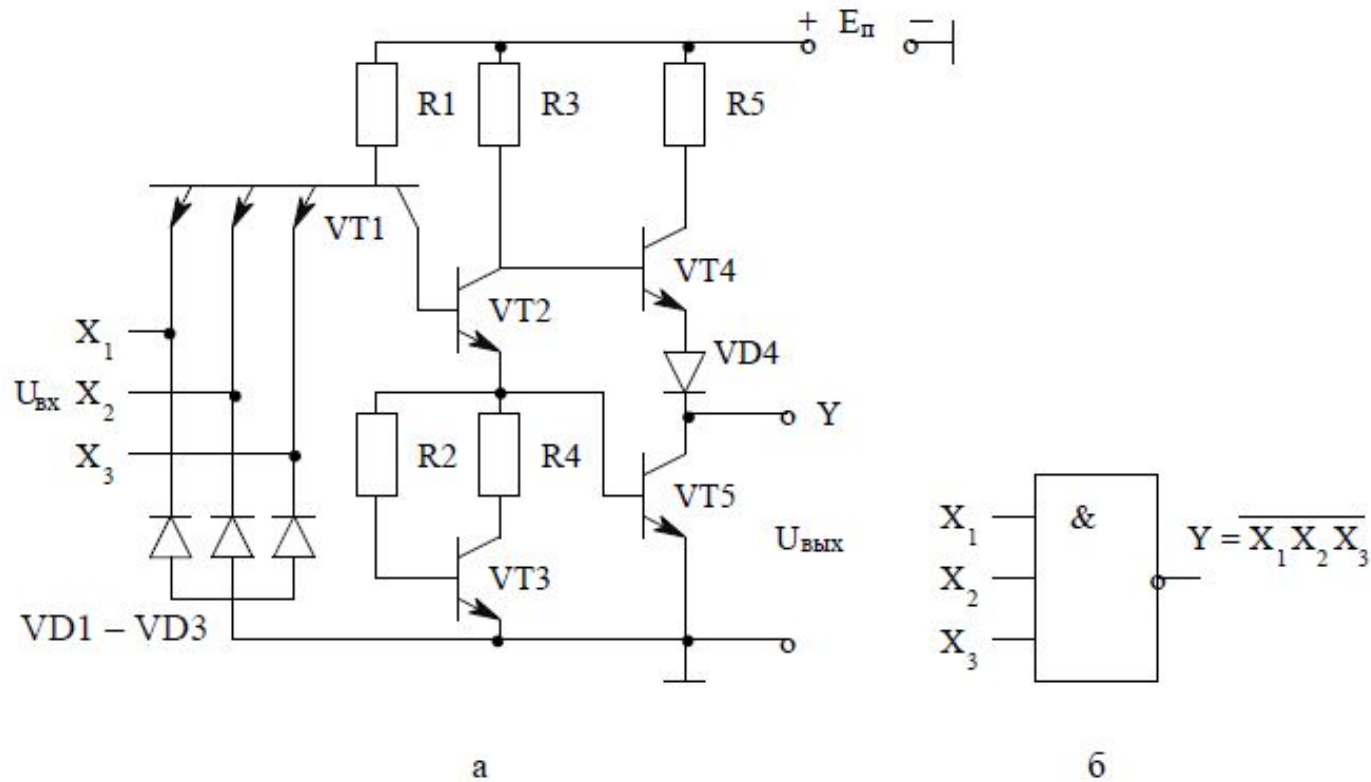


Рис. 2.16. Принципиальная схема (а) ТТЛ-элемента серии 155 и его функциональное обозначение (б)

Схема ЛЭ с открытым коллектором приведена на рис. 2.21.

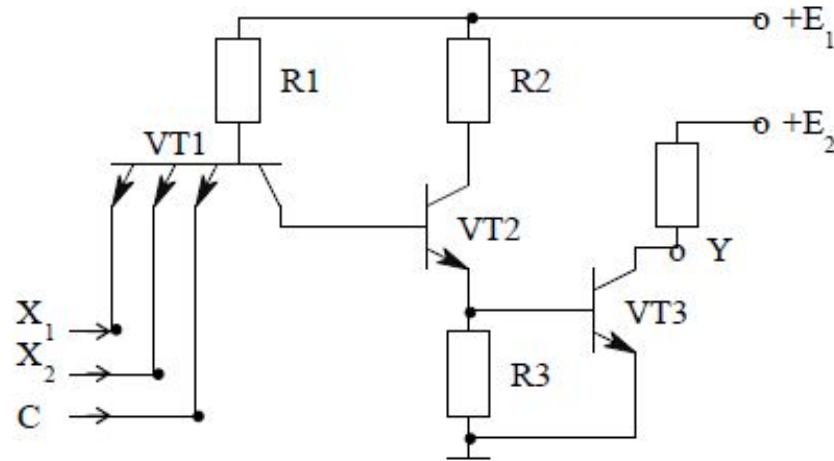


Рис. 2.21. Схема ТТЛ логического элемента с открытым коллектором

Эти элементы применяются в тех случаях, когда необходимо питать энергоемкую нагрузку. Выход с открытым коллектором позволяет иметь много параллельных выходов, что обеспечивает логическое сложение выходных функций схем при подаче напряжения питания E на группу соединенных вместе выходов через резистор соответствующей величины сопротивления, которая зависит от числа соединенных выходов и числа подключенных к ним входов последующих элементов. Кроме того, выход с открытым коллектором дает возможность в качестве нагрузки использовать лампы накаливания, светодиоды, обмотки реле, индикаторы и т. д.

Логические элементы эмиттерно-связанной логики (ЭСЛ) обладают наиболее высоким быстродействием из всех существующих ЛЭ, что обусловлено следующими факторами: открытые транзисторы работают в активном режиме, благодаря чему отсутствует задержка, связанная с рассасыванием избыточного заряда в транзисторах; внутренняя симметрия ЛЭ обеспечивает практическое отсутствие изменения потребляемого тока при переключении и связанных с ним всплесков напряжения в цепях питания.

Логические элементы ЭСЛ выполняются на базе переключателей тока (рис. 2.22). На базу транзистора VT0 подано напряжение E_0 . Пусть $U_{вх1} = E_0$, при этом оба транзистора работают в активном режиме. Ток через резистор R_3 равен: $I_3 = |E_0 - U_{бэ0}| R_3$. Транзисторы выбирают идентичными, поэтому $i_{31} = i_{30} = I_3/2$.

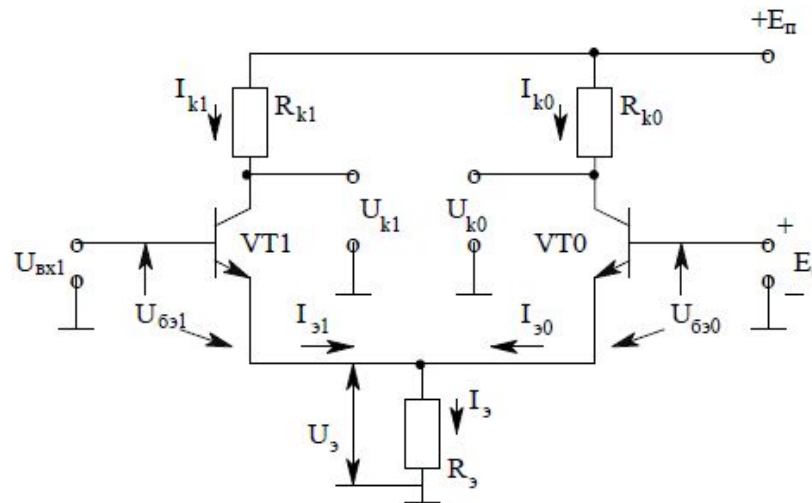


Рис. 2.22. Схема переключателя тока

$U_{k1} = U_{k0} = E_{\pi} - I_3 R_{k1} / 2$. Если входное управляющее напряжение $U_{вх1} > E_0$, то i_{3l} , i_{k1} увеличиваются, напряжение U_{k1} падает, а напряжение U_3 растет, что приводит к уменьшению $U_{бэ0}$ и, следовательно, к уменьшению i_{k0} и к увеличению U_{k0} . Дальнейшее увеличение $U_{вх}$ приводит к полному запираению транзистора VT0. При этом напряжение $U_{k0} = E_{\pi}$ (без учета неуправляемых тепловых токов), $U_{k1} = E_{\pi} - I_3 R_k$. Ток $I_3 \approx I_{k1}$ течет только через транзистор VT1.

При уменьшении напряжения $U_{вх}$ относительно E_0 уменьшается ток i_{k1} , растет напряжение U_{k1} , растет ток i_{k0} , уменьшается напряжение U_{k0} . При некотором $U_{вх} < U_0$ транзистор VT1 закрывается и ток I_3 течет только через транзистор VT0. В этом случае $U_{k0} = E_{\pi} - I_3 R_{k0}$; $U_{k1} = E_{\pi}$. Таким образом, при изменении $U_{вх}$ на некоторое значение $2\Delta U$ симметрично относительно E_0 , транзисторы переключаются (рис. 2.23).

Выходной перепад напряжения $U_{шввх} = I_3 R_k$. Отметим, что высокий входной уровень $U_{вх}^1$ соответствует уровню логической единицы E^1 , а низкий входной уровень $U_{вх}^0$ – уровню логического нуля E^0 . Обычно выбирают $E_0 = 0,5 (E^1 + E^0)$.

2.16. ЭСЛ логические элементы

Логические элементы эмиттерно-связанной логики (ЭСЛ) обладают наиболее высоким быстродействием из всех существующих ЛЭ, что обусловлено следующими факторами: открытые транзисторы работают в активном режиме, благодаря чему отсутствует задержка, связанная с рассасыванием избыточного заряда в транзисторах; внутренняя симметрия ЛЭ обеспечивает практическое отсутствие изменения потребляемого тока при переключении и связанных с ним всплесков напряжения в цепях питания.

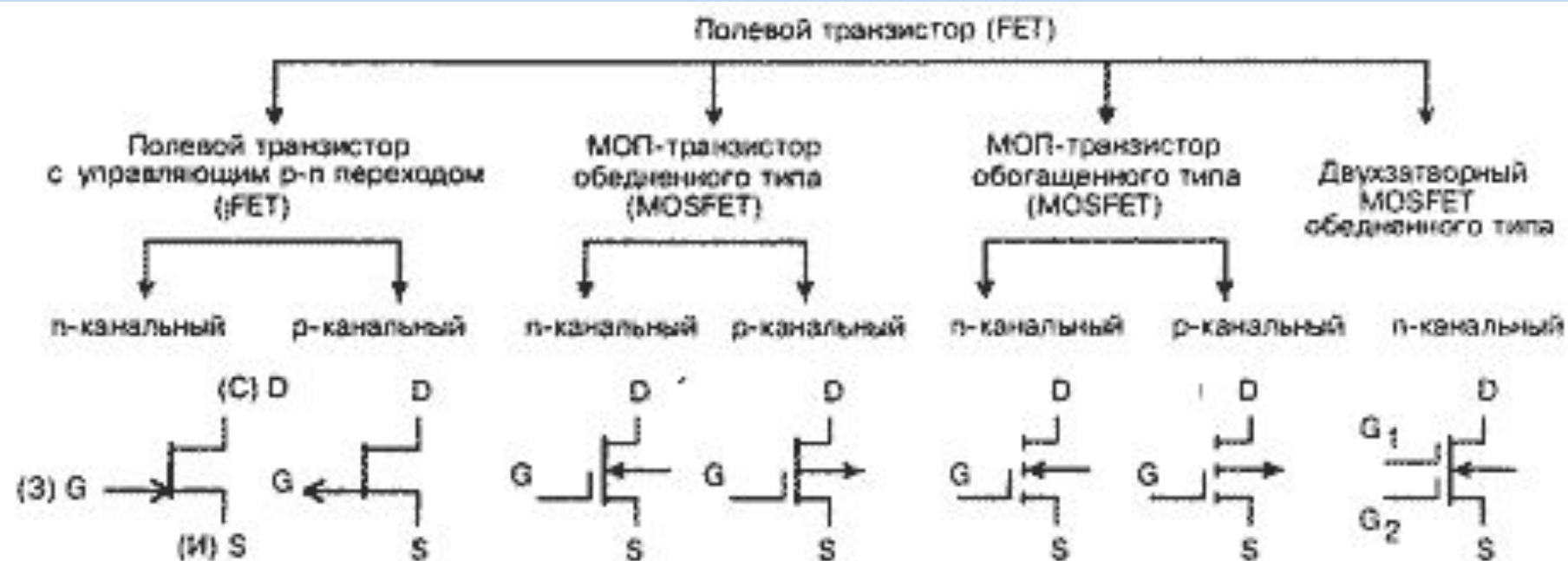


Рис. 14.1. Основные разновидности полевых транзисторов

2.17. Логические элементы на МОП-транзисторах

Логические элементы на МОП-транзисторах изготавливаются по р-МОП, п-МОП, КМОП-технологиям. В первых используются р-канальные транзисторы, во вторых п-канальные транзисторы, а в третьих – используются комплементарные МОП-схемы, где в одной схеме применяются как р-канальные, так и п-канальные транзисторы. Схемы на МОП-транзисторах имеют меньшее быстродействие, чем схемы на биполярных транзисторах, что объясняется значительными емкостями между затвором, стоком, истоком и подложкой, на перезаряд которых требуется определенное время. Схемы на МОП-транзисторах потребляют малую мощность, имеют высокую нагрузочную способность и высокую помехоустойчивость, занимают малую площадь на поверхности кристалла, они технологичны и дешевы. Их применяют в тех случаях, где нужна большая степень интеграции и небольшое быстродействие. Особенность микросхем на МОП-структурах заключается в том, что в их схемах отсутствуют резисторы, а роль нелинейных резисторов выполняют соответствующим образом включенные транзисторы.

Логические элементы на p-канальных транзисторах. В основе всех логических схем на МОП-структурах лежит инвертор (рис. 2.25). При подаче на вход X напряжения $-E$ (логическая единица), транзистор $VT2$ открывается, его сопротивление становится малым и на выходе мы получаем напряжение близкое к нулю (логический нуль). При подаче на вход напряжения логического нуля транзистор $VT2$ закрывается и на выходе мы имеем напряжение $-E$.

Для реализации функции И-НЕ ключевые транзисторы включают последовательно с нагрузочным транзистором $VT1$ (рис. 2.26). Если на каком-либо входе напряжение равно нулю (логический нуль), то соответствующий транзистор закрыт и напряжение на выходе равно $-E$ (логическая единица).

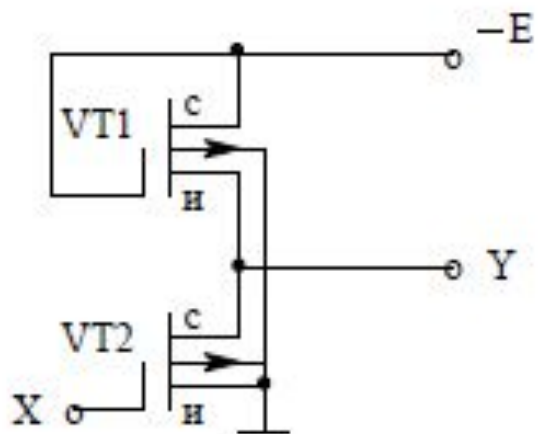


Рис. 2.25. Инвертор на р-МОП-транзисторах

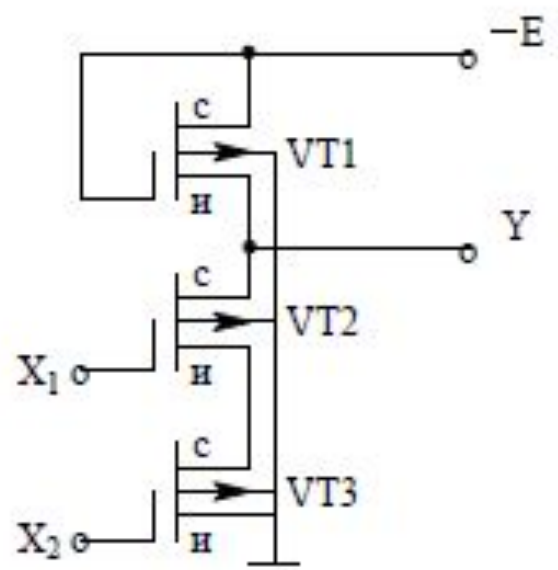


Рис. 2.26. Схема ЛЭ И-НЕ на р-МОП-транзисторах

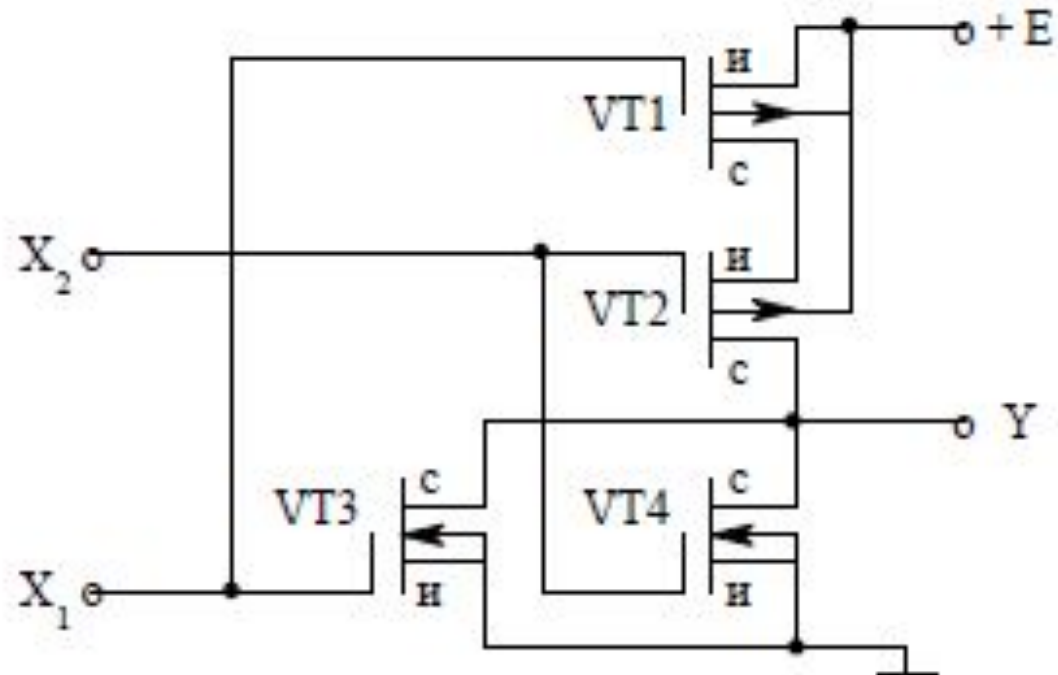


Рис. 2.28. Схема ЛЭ ИЛИ-НЕ на КМОП-структуре

3.1. Устройство равнозначности

Устройство равнозначности (УР) — это логическая схема с двумя входами X_1 и X_2 и выходом Y , условия работы которой таковы: сигнал $Y = 1$ только при совпадении информационных значений входных сигналов. В таблице 3.1 приведена таблица истинности УР.

Структурная формула схемы для наборов, где $Y = 1$, имеет вид

$$Y = X_1X_2 + \bar{X}_1\bar{X}_2.$$

Функциональная схема УР приведена на рисунке 3.1. Она содержит пять логических элементов — два инвертора, два элемента И и один элемент ИЛИ.

Таблица 3.1

X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

На основе правила де Моргана УР можно выполнить в базисе И-НЕ:

$$Y = X_1X_2 + \overline{X_1}\overline{X_2} = \overline{\overline{X_1X_2}\overline{\overline{\overline{X_1}\overline{X_2}}}}$$

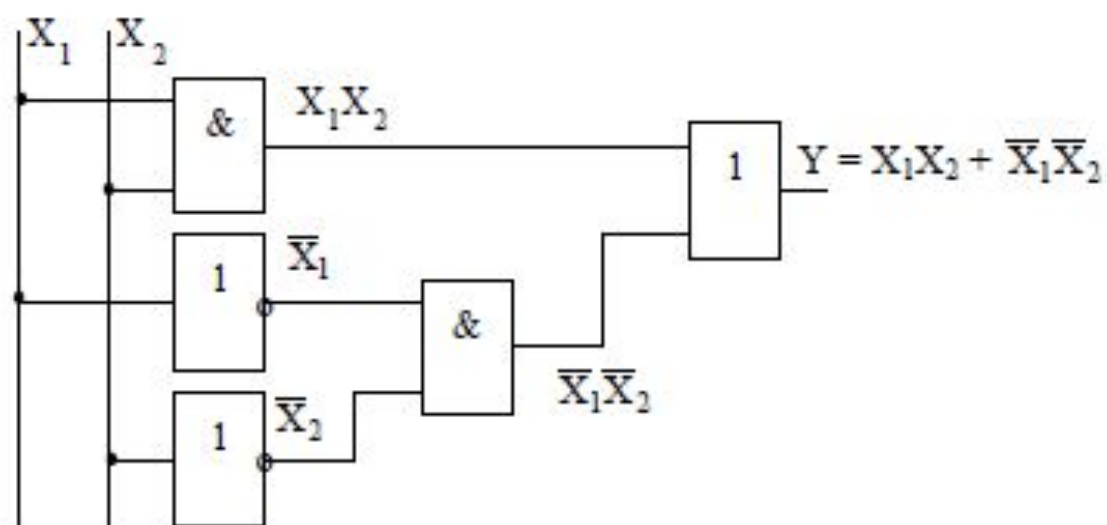


Рис. 3.1. Функциональная схема устройства равнозначности

Эту структурную формулу можно реализовать на пяти элементах И-НЕ (рис. 3.2).

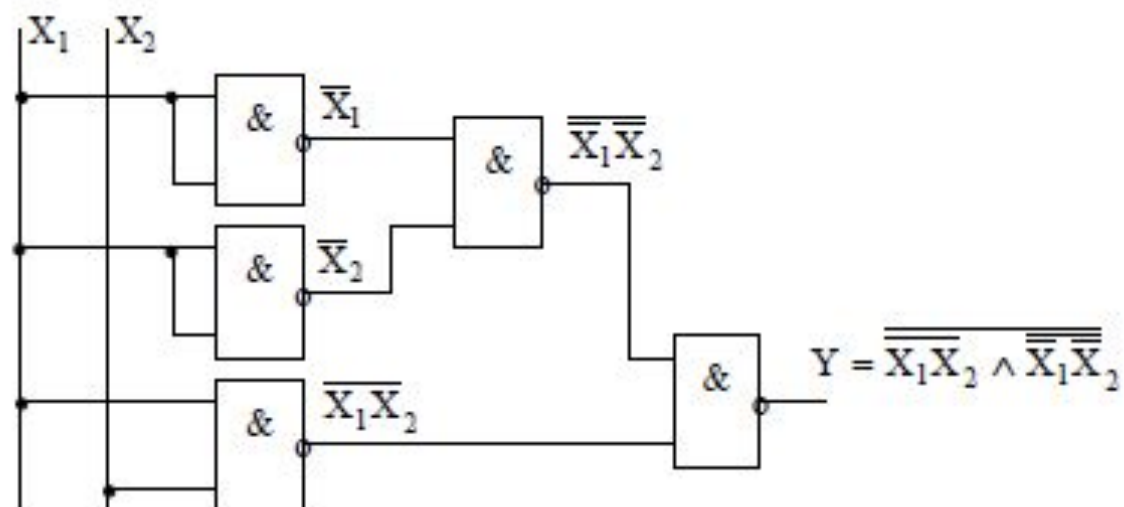


Рис. 3.2. Функциональная схема УР на элементах И-НЕ

Устройства равнозначности применяются в основном в схемах сравнения цифровых кодов.

3.2. Устройство неравнозначности

Устройство неравнозначности — это устройство с двумя входами X_1 и X_2 и выходом Y , реализующее логическую функцию, называемую «исключающее ИЛИ»: $Y = 1$ только при несовпадении информативных значений входных сигналов. Таблица истинности устройства неравнозначности представлена в таблице 3.2.

Таблица 3.2

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Согласно таблице истинности можно записать структурную формулу

$$Y = X_1\bar{X}_2 + \bar{X}_1X_2 = X_1 \oplus X_2.$$

Символ \oplus означает операцию «исключающее ИЛИ». Устройство может быть реализовано схемой, содержащей пять логических элементов: два инвертора, два элемента И и один элемент ИЛИ (рис. 3.3).

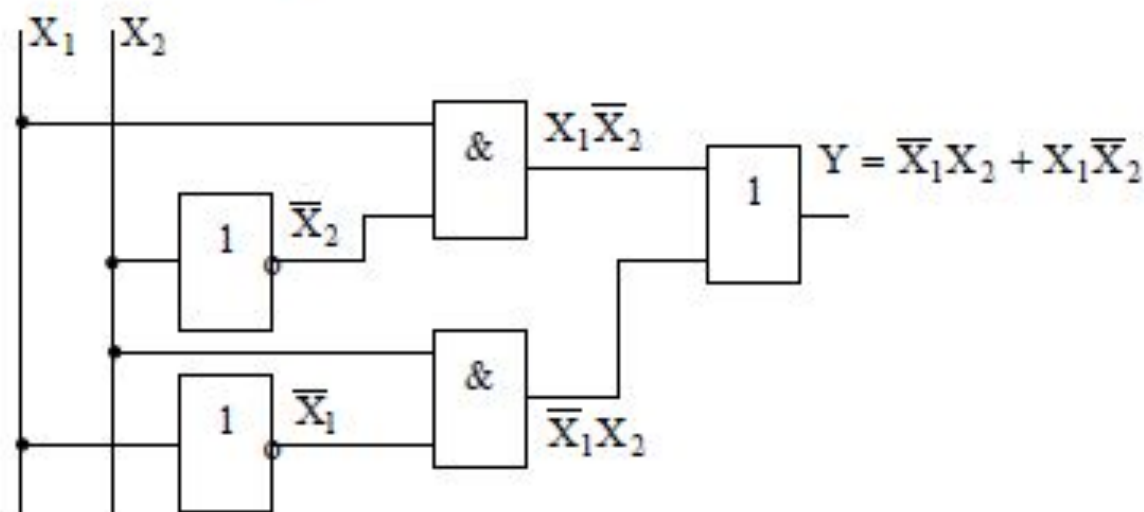


Рис. 3.3. Функциональная схема устройства неравнозначности

Устройства неравнозначности являются базовыми для построения комбинационных полусумматоров и сумматоров.

3.3. Комбинационные сумматоры

Рассмотрим логическую схему полусумматора, предназначенную для суммирования двух одноразрядных двоичных чисел. Логическая функция полусумматора задается следующей таблицей истинности (табл. 3.3).

Таблица 3.3

X_1	X_2	S'	P'
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Здесь X_1 и X_2 — переменные, S' — частичная сумма (сумма по модулю два), P' — перенос в старший разряд. Из таблицы истинности видно, что

$$S' = X_1 \oplus X_2, P' = X_1 X_2,$$

т. е. полусумматор имеет два выхода (S' , P') и реализуется с помощью устройства неравнозначности и схемы И. Функциональная схема полусумматора представлена на рис. 3.4 (а, б).

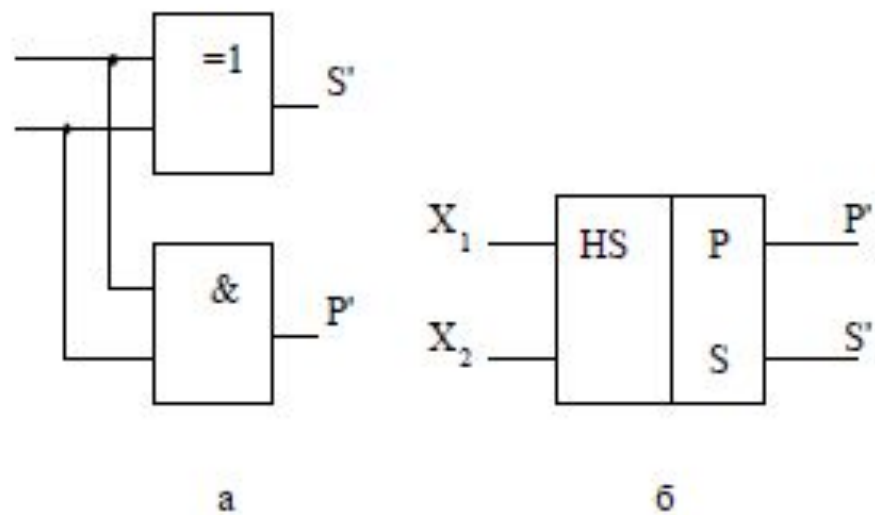


Рис. 3.4. Функциональная схема полусумматора (а)
и его условное графическое обозначение (б)

Два одноразрядных полусумматора образуют одноразрядный сумматор. Сумматор — логическая схема, имеющая три входа (входы разрядов X_1 и X_2 суммируемых чисел и вход X_3 для возможного переноса, образованного при сложении младших разрядов). Сумматор может быть образован с помощью двух полусумматоров и элемента ИЛИ. Таблица истинности (табл. 3.4) и функциональная схема сумматора показаны на рис. 3.5. В суммирующем устройстве последовательного действия для суммирования многоразрядных чисел перенос P подается на вход X_3 с задержкой на один такт сложения. Для ускорения процесса сложения двух n -разрядных чисел можно построить суммирующее устройство параллельного действия, в которых используются n сумматоров одноразрядных чисел. В настоящее время одно-, двух- и четырехразрядные двоичные сумматоры выпускаются в виде отдельных ИМС (К155ИМ1, К155ИМ2, К155ИМ3).

Таблица 3.4

X_1	X_2	X_3	S	P
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

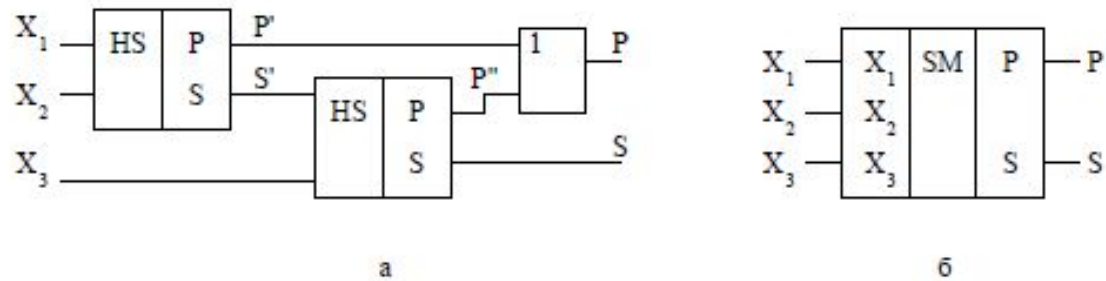


Рис. 3.5. Функциональная схема (а)

и условное графическое обозначение сумматора (б)

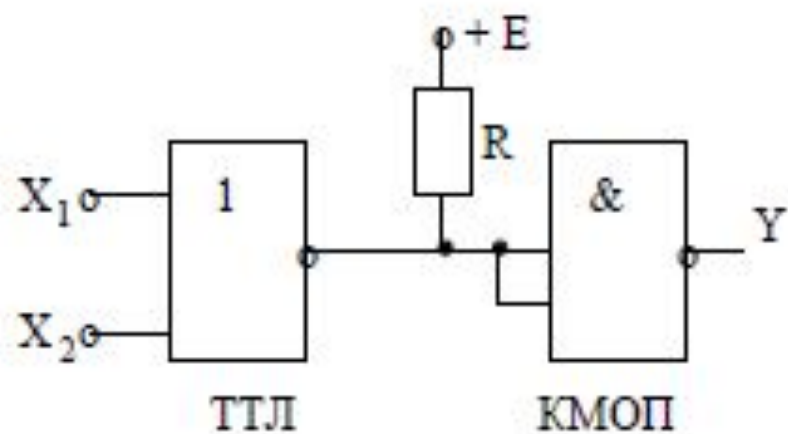


Рис. 2.30. Схема согласования ТТЛ и КМОП-логических элементов

Триггером называется устройство, обладающее двумя состояниями устойчивого равновесия (1 и 0) и способное под воздействием внешнего управляющего сигнала переходить скачком из одного состояния в другое.

Асинхронный RS- триггер. В основе схем триггеров лежит основной (базовый) асинхронный RS-триггер, который может быть построен на двух логических элементах И-НЕ или ИЛИ-НЕ. Элементы охвачены цепями обратных связей, для чего выход каждого элемента подключен к одному из входов другого элемента.

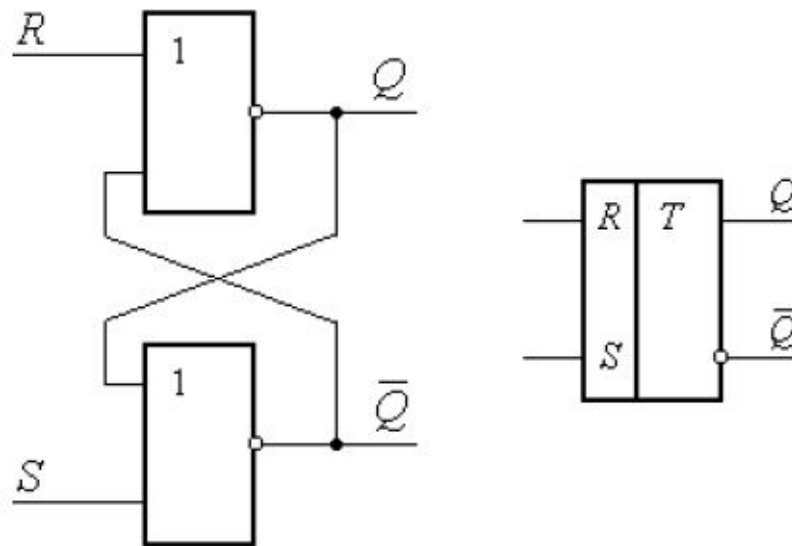


Рис. 1. Асинхронный RS-триггер на элементах ИЛИ-НЕ: *a* – логическая структура; *б* – условное обозначение

Из табл. 1 видно, что RS -триггер сохраняет одно из устойчивых состояний независимо от многократного изменения информационного сигнала на одном входе при нулевом значении информационного сигнала на другом входе.

Это свойство блокирования — основное функциональное свойство RS -триггера, и именно оно делает его элементарной запоминающей ячейкой.

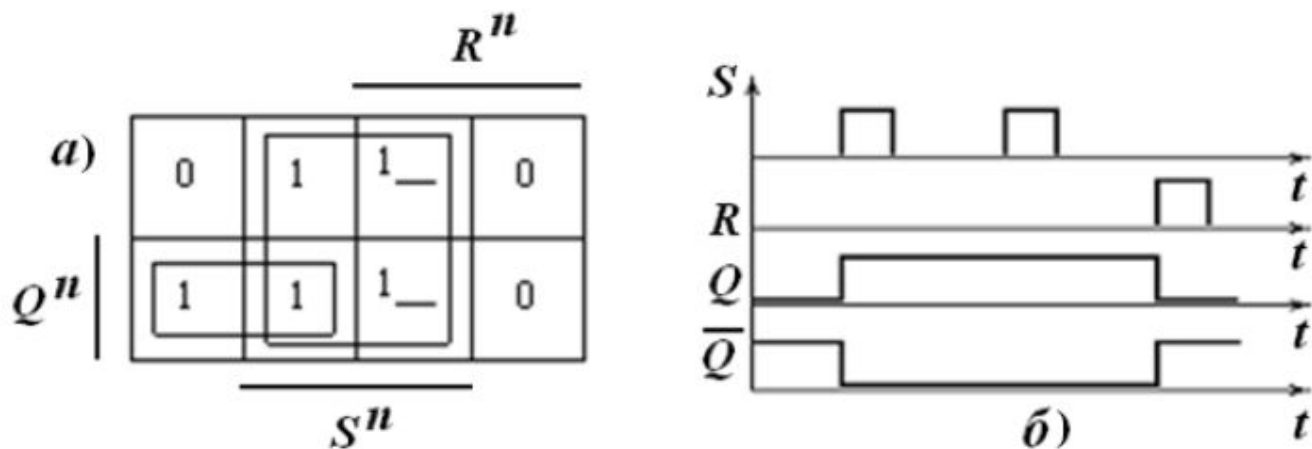
Таблица 1

Таблица состояний RS -триггера на элементах ИЛИ–НЕ

Номер набора	R	S	Q^n	Q^{n+1}	Режим
0	0	0	0	(0)	Хранение
1	0	0	1	(1)	
2	0	1	0	(1)	Установка 1
3	0	1	1	(1)	
4	1	0	0	(0)	Установка 0
5	1	0	1	0	
6	1	1	0	X	Неопределенность
7	1	1	1	X	

$$Q^{n+1} = S^n \vee \overline{R^n} Q^n, \text{ при } R \cdot S = 0 \quad (7.1)$$

Полученная формула описывает работу RS триггера.



e)

ТАКТ t^n		ТАКТ t^{n+1}
R^n	S^n	Q^{n+1}
0	0	Q^n
0	1	1
1	0	0
1	1	н/о

Рис. 2. Карта Карно функции переходов RS-триггера (a); временная диаграмма его работы (б) и минимизированная таблица состояний RS-триггера (e)

Асинхронный RS - триггер с инверсными входами. Схема асинхронного RS -триггера, выполненная на элементах И-НЕ, таблица переходов, обозначение и временные диаграммы представлены на рис. 3. Такой вариант триггера называют \overline{RS} -триггером или RS -триггером с инверсным управлением. Триггер на элементах И-НЕ нашел наибольшее применение.

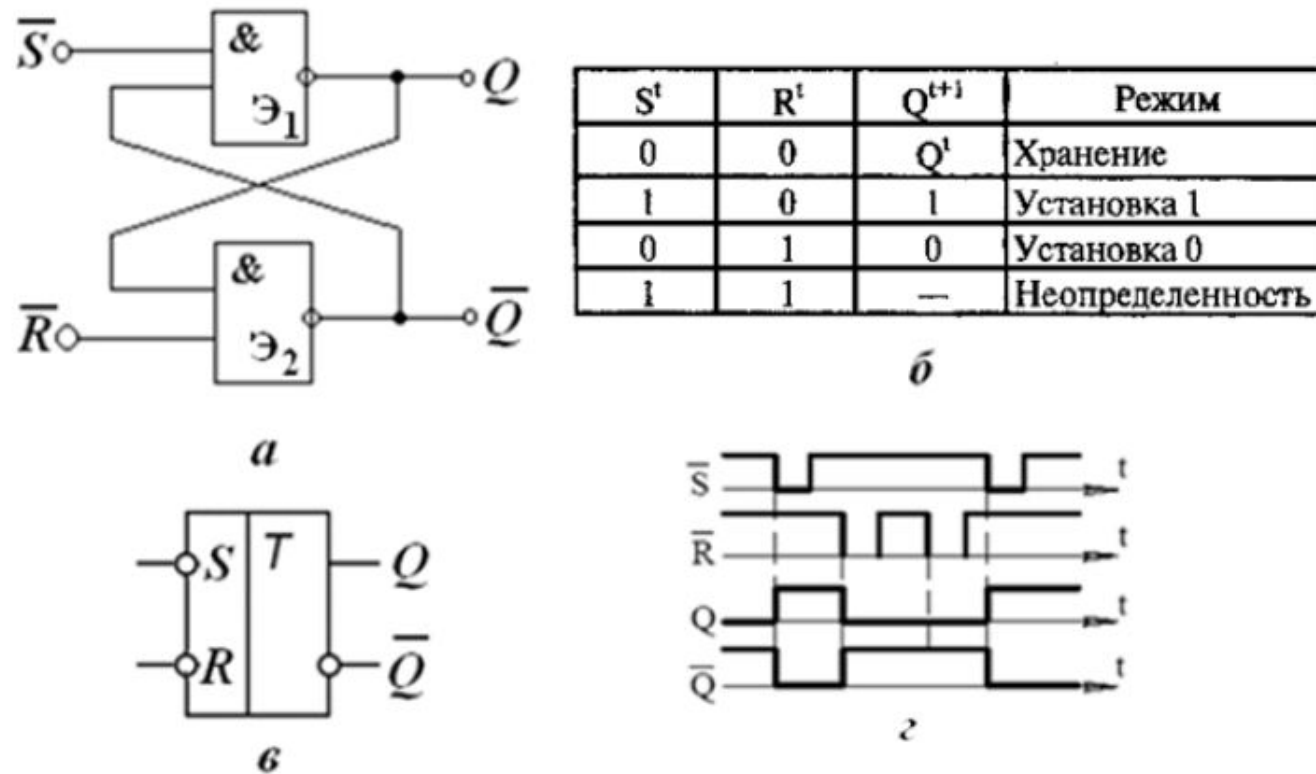


Рис. 3. Структурная схема (а); сокращенная таблица переходов (б) условное графическое обозначение (в) и временные диаграммы работы (г) асинхронного RS -триггера на элементах И-НЕ

Синхронный RS-триггер (RST-, RSC-триггер). Состояние RST- триггера может меняться только при поступлении специальных тактовых импульсов. В промежутках между тактовыми импульсами изменения уровней на входах S и R не вызывают изменения состояния триггера, а лишь программируют то состояние, которое он примет при поступлении очередного тактового импульса. Тактовый вход на схемах обозначен C . Структура и временные диаграммы работы синхронного RS-триггера представлены на рис. 4.

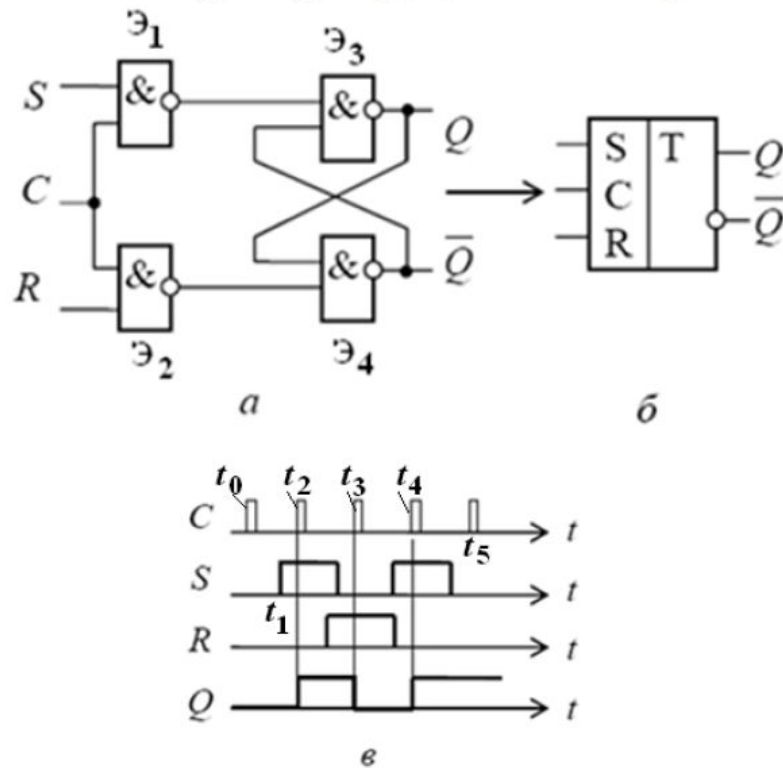


Рис. 4. – Тактируемый RS-триггер: а – построение триггера на логических элементах И-НЕ; б – обозначение триггера на функциональных схемах; в – временные диаграммы

D-триггеры (от английского слова *delay* - задержка) имеют единственный входной сигнал D . Значение сигнала на выходе Q такого триггера в такте $(t+1)$ равно значению сигнала на выходе D в предыдущем такте t . Таким образом, входной сигнал передается на выход с задержкой на один такт.

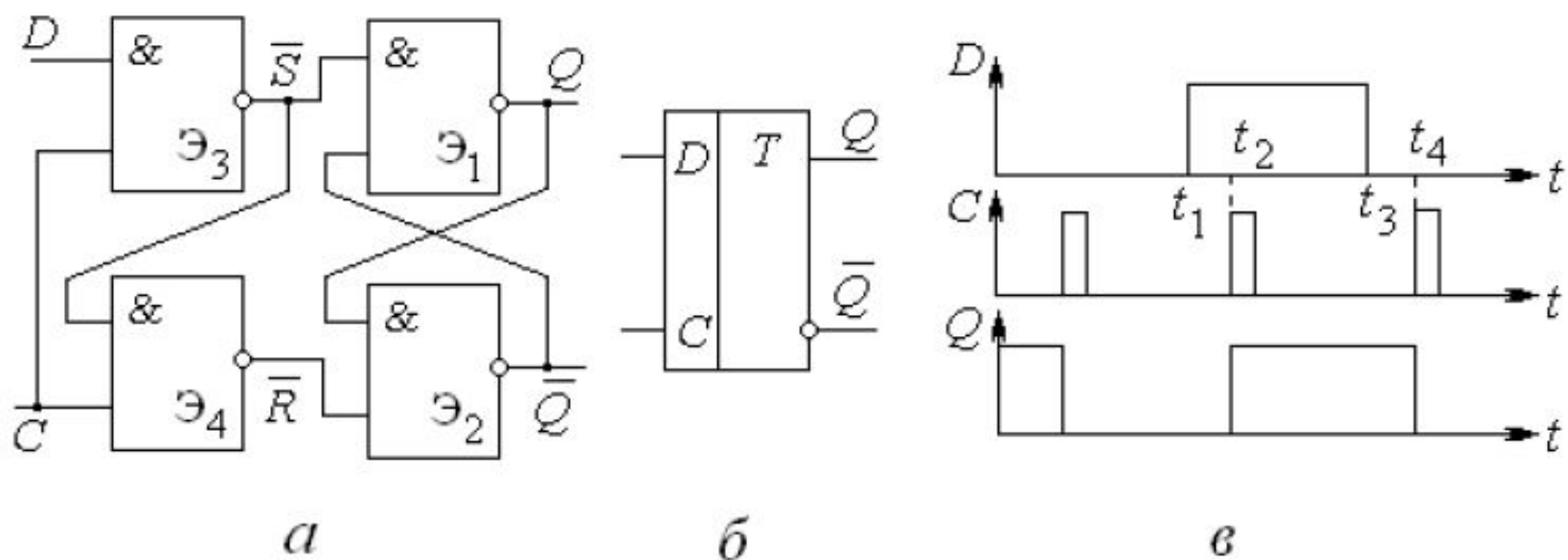


Рис. 5. Структурная схема (а), условное графическое обозначение (б) и временные диаграммы работы (в) тактируемого D-триггера

***T*-триггер**, или счетный триггер, имеет один информационный вход и переходит в противоположное состояние в результате воздействия на его вход каждого очередного сигнала. Название «счетный» (или со «счетным запуском») связано с широким применением *T*-триггеров в счетчиках импульсов. Смена состояний здесь происходит всякий раз, когда входной сигнал меняет своё значение в определенном направлении.

T-триггер – единственный вид триггера, текущее состояние которого определяется не информацией на входах, а состоянием его в предыдущем такте.

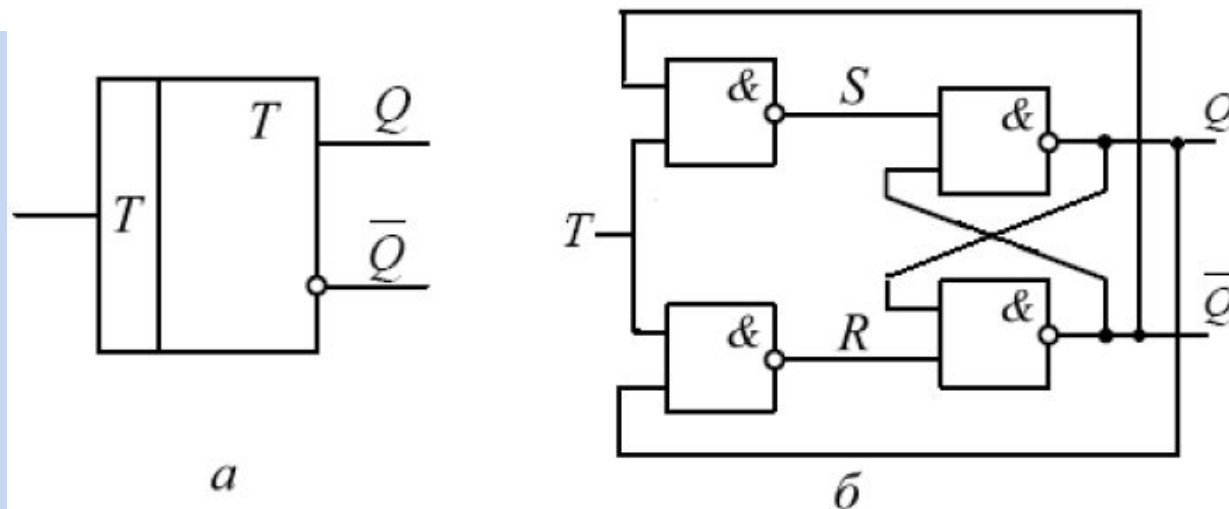
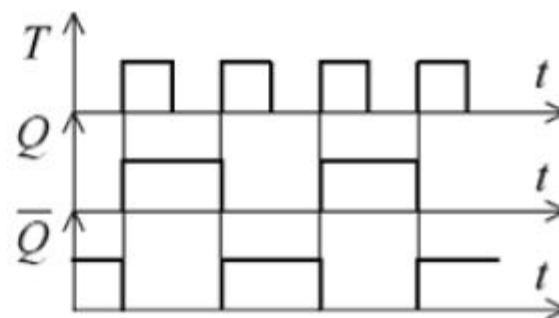


Рис. 6. *T*-триггер на элементах И-НЕ: *a* – условное обозначение;
б – схема

На рис. 7 приведена таблица истинности и временные диаграммы работы динамического (по фронту информационного сигнала) T -триггера.

T	Q	\bar{Q}
0	0	1
1	1	0
0	1	0
1	0	1
0	0	1
1	1	0

a



б

Рис. 7. Динамический (по фронту информационного сигнала) T -триггер: *a* – таблица истинности; *б* – временные диаграммы

JK-триггер является наиболее универсальным. В JK-триггере входы J и K (подобно входам R и S) задают желательное состояние, но в отличие от RS-триггера допустим набор $J = 1, K = 1$. То есть JK-триггер отличается от двухступенчатого RS-триггера тем, что в нём отсутствует запрещённая комбинация входных сигналов. Когда $J = K = 1$, он превращается в T-триггер, на остальных комбинациях ведёт себя подобно RS-триггеру.

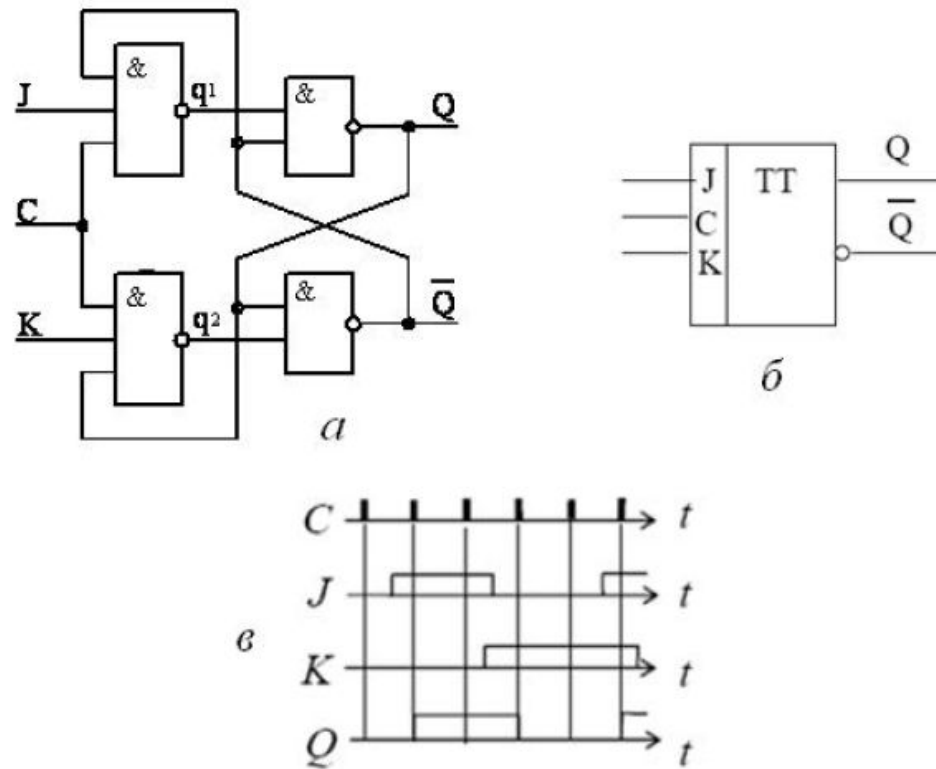


Рис. 8. Структурная схема JK-триггера (а), его условное обозначение (б) и временные диаграммы (в)

Логическое уравнение JK -триггера имеет вид:

$$Q^{n+1} = J \cdot Q + K \cdot Q^n \quad (7.6)$$

В частности, при $J = 1$, $K = 1$ получаем:

$$Q^{n+1} = Q^n$$

JK -триггер, также как и все типы триггеров, получаемые на его основе, является триггером с внутренней задержкой: в момент действия импульса T информация записывается в основной триггер, а после его окончания состояние основного триггера передается во вспомогательный. Наличие в работе триггера внутренней временной задержки, определяемой длительностью тактовых импульсов, часто отражается индексом t в обозначении JK -триггера и триггеров, выполненных на его основе: JKt , RSt , Dt , Tt .

4.11. Регистры

Регистром называется устройство, предназначенное для выполнения операций приема, хранения и передачи слов в двоичном коде. Наиболее распространены статические регистры. Каждому разряду слова, записанному в такой регистр соответствует свой разряд регистра, выполненный на основе статического триггера. Разряды регистра нумеруются в порядке нумерации разрядов в слове. Находят также применение динамические регистры, в которых функции запоминания значения двоичной переменной реализуются конденсатором (иногда индуктивностью). Ввиду того, что время хранения заряда на конденсаторе и тока в индуктивности ограничено, в динамических регистрах предусматривается операция регенерации (восстановления) хранимой информации. По способу записи информации регистры подразделяются на параллельные, последовательные и параллельно-последовательные. В параллельном регистре запись и выдача слова осуществляется в параллельной форме одновременно во всех разрядах регистра. Последовательный регистр характеризуется последовательной записью и выдачей слова, начиная со старшего или младшего разряда. Параллельно-последовательный регистр имеет входы и выходы как для параллельной, так и для последовательной формы приема и передачи слова. На основе таких регистров осуществляются операции преобразования последовательного кода в параллельный и наоборот. В регистре могут выполняться следующие операции: прием слова из другого ПУ, передача слова из регистра в другие ПУ, поразрядные логические операции, сдвиг слова вправо или влево на заданное число разрядов, обращение кода.

4.11.1. Регистры памяти

Регистр с параллельным приемом и выдачей информации называется регистром памяти. Он позволяет записывать, хранить и в нужный момент выдавать информацию в прямом или обратном коде. Регистры памяти могут быть построены на RS-, D-, или JK-триггерах. Схема регистра памяти на RS-триггерах со счетным входом представлена на рис. 4.16.

При подаче управляющего импульса на шину «Сброс», все триггеры устанавливаются в нулевое состояние. Ввод новой информации в регистр осуществляется через ячейки И, связанные с входными шинами. Для записи информации, подведенной к входным шинам, подается управляющий импульс на шину «Ввод». При этом срабатывают те ячейки И, на входных шинах которых действует сигнал 1. Под действием импульсов, появляющихся на выходах ячеек И, соответствующие триггеры будут установлены в состояние 1. Вывод информации из регистра также осуществляется через элементы И, связанные с выходами триггеров.

Для вывода информации (считывания) управляющий сигнал подается на шину «Вывод». При этом срабатывают ячейки И, соединенные с триггерами, в которых записана 1 и информация в параллельном коде передается на выходные шины.

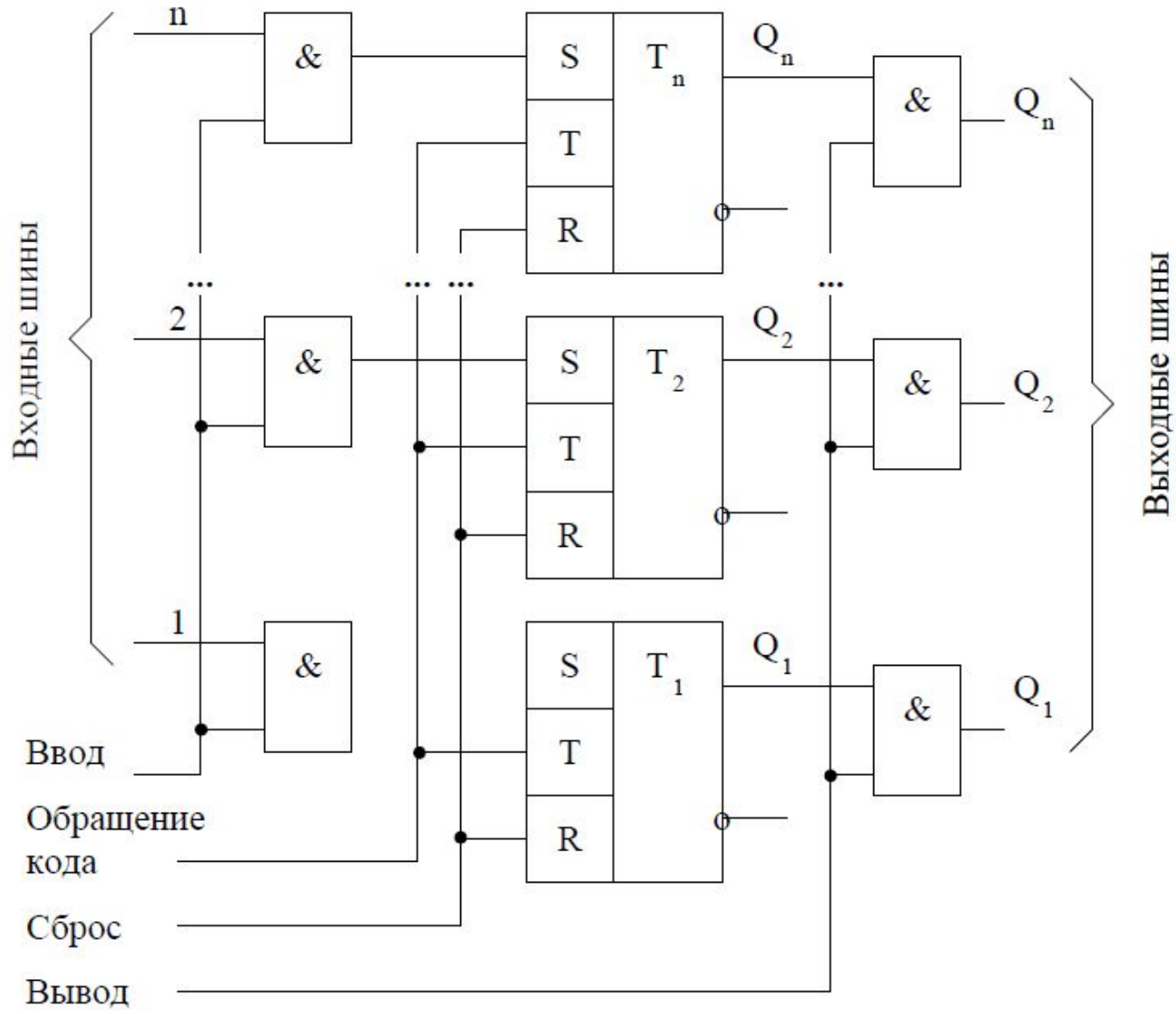


Рис. 4.16. Схема регистра памяти на RS-триггерах

Для выдачи информации в инвертированном (обратном) коде, когда все единицы заменяются нулями, а нули – единицами, необходимо подать управляющий импульс на шину «Обращение кода», соединенную со счетными входами триггеров. При этом состояние всех триггеров меняется на противоположное, и если теперь подать сигнал на шину «Вывод», то на выходные шины будет передана информация в обратном коде. Вторичное воздействие импульса «Обращение кода», если это необходимо, преобразует информацию снова в прямой код.

4.11.2. Сдвигающие регистры

Регистры с последовательной записью и выдачей информации называют сдвигающими регистрами. Сдвигающие регистры строятся на D-триггерах или JK-триггерах, включенных по схеме D-триггера. Они предназначены для сдвига слова, записанного в регистр, вправо или влево на один или несколько разрядов. Для ввода информации в регистр, записываемое слово в виде последовательности импульсов, разряд за разрядом поступает на вход регистра в моменты действия тактовых импульсов. Для выдачи записанной информации необходимо снова подать продвигающие импульсы. При этом на выходе регистра, разряд за разрядом, начиная с младшего, в момент действия тактовых импульсов будут появляться сигналы, соответствующие кодовой комбинации, хранящейся в регистре. По мере вывода информации из регистра, старшие разряды регистра будут освобождаться. Поэтому вывод информации из регистра можно совместить с записью новой информации. Рассмотрим схему регистра на универсальных JK-триггерах (рис. 4.17).

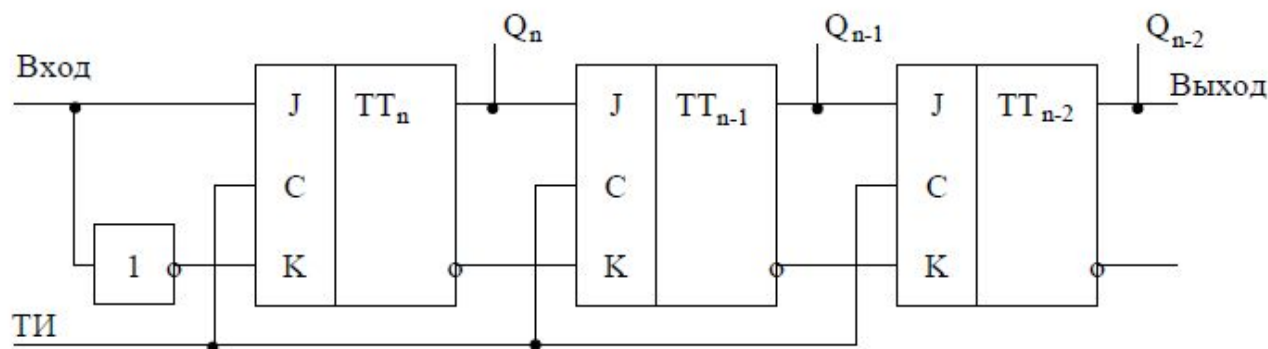


Рис. 4.17. Схема сдвигающего регистра на JK-триггерах

В момент поступления тактового импульса (ТИ) на синхронизирующие входы триггеров они принимают информацию от соседних слева триггеров, т. е. информация сдвигается на один разряд вправо. Если сдвиг информации возможен в обе стороны: и вправо, и влево – такие регистры называют реверсивными. Рассмотрим работу схемы реверсивного регистра на D-триггерах (рис. 4.18).

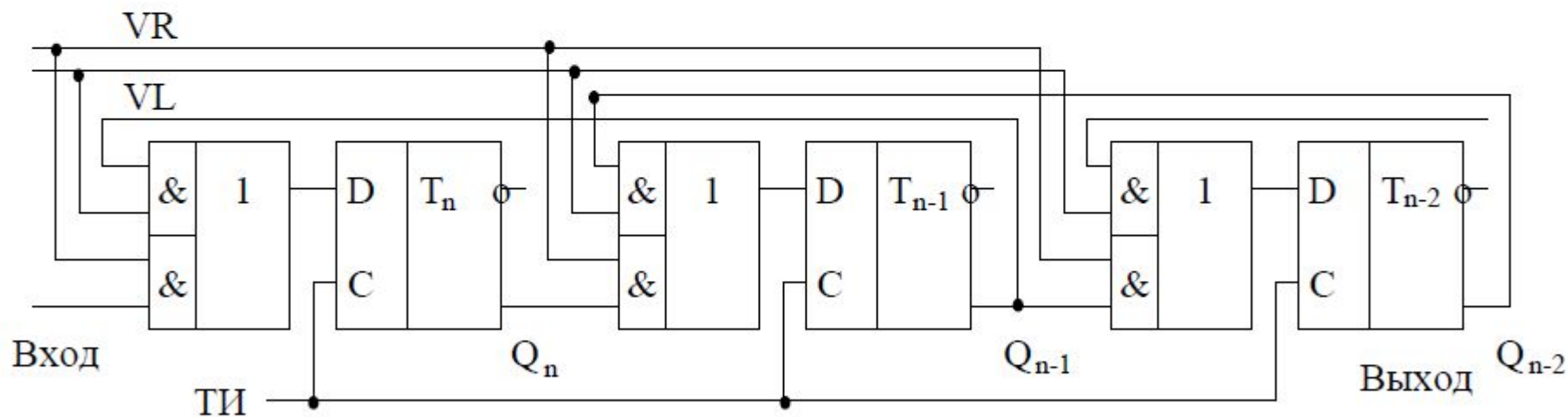


Рис. 4.18. Схема реверсивного регистра на D-триггерах

При подаче разрешающего сигнала на управляющий вход VR включается схема сдвига вправо. Реверсивный регистр при этом превращается в регистр сдвига вправо. При подаче разрешающего сигнала на управляющий вход VL включается схема сдвига влево. В регистрах сдвига влево и вправо разряды двоичного кода выходят за пределы разрядности регистра. Если соединить выход крайнего правого разряда регистра со входом крайнего левого разряда, то получим схему кольцевого регистра сдвига. Возможно также совмещение в одной схеме памяти и регистра сдвига. На рис. 4.19 представлена схема 4-разрядного регистра памяти и 4-разрядного регистра сдвига влево.

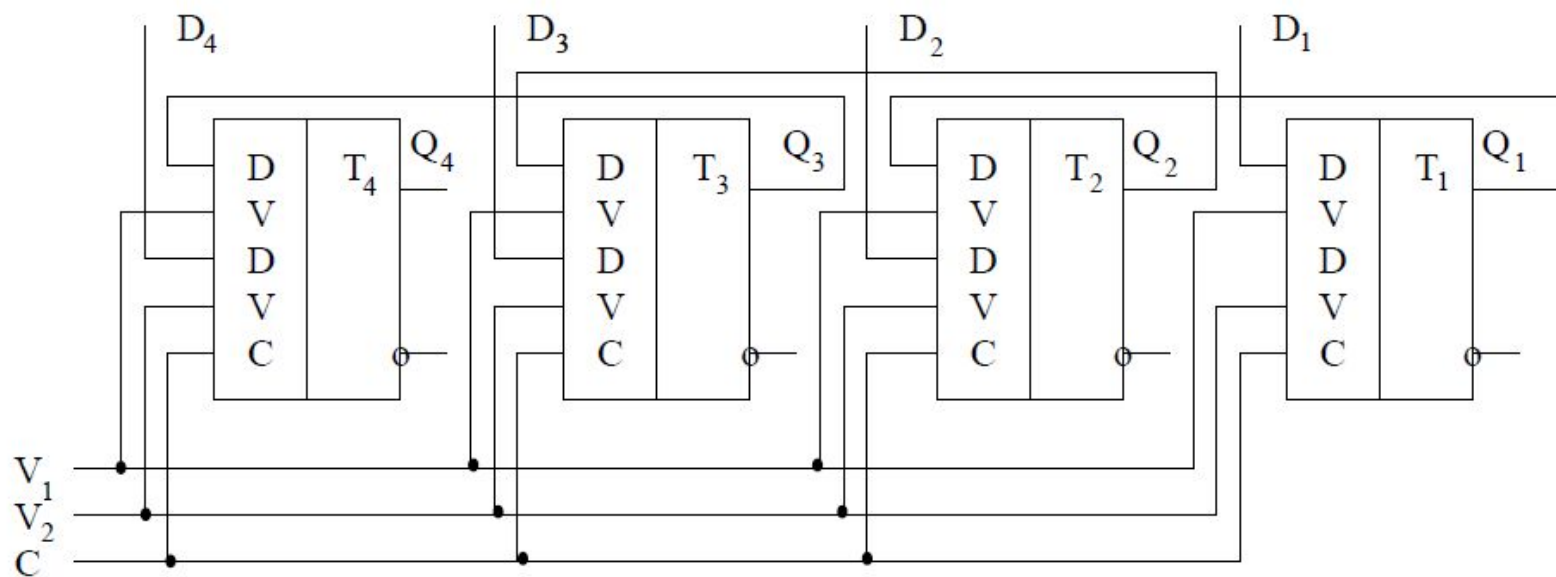


Рис. 4.19. Совмещенная схема регистра памяти и регистра сдвига

4.12. Счетчики

Цифровым счетчиком называется устройство, осуществляющее счет числа входных импульсов и фиксирующие это число в каком-либо коде. После определенного числа импульсов счетчик обычно сбрасывается в исходное состояние и счет повторяется. Если на вход подается серия импульсов и выходной сигнал образуется только при появлении в счетчике одной заданной кодовой комбинации, то счетчик функционирует как делитель числа импульсов, т. к. эта кодовая комбинация периодически повторяется. В цифровой технике применяются суммирующие, вычитающие и реверсивные счетчики. По виду связи между разрядами различают счетчики с непосредственными связями, с параллельным переносом и комбинированными связями, асинхронные и синхронные. По коэффициенту счета счетчики разделяют на двоичные (бинарные) и с произвольным коэффициентом счета.

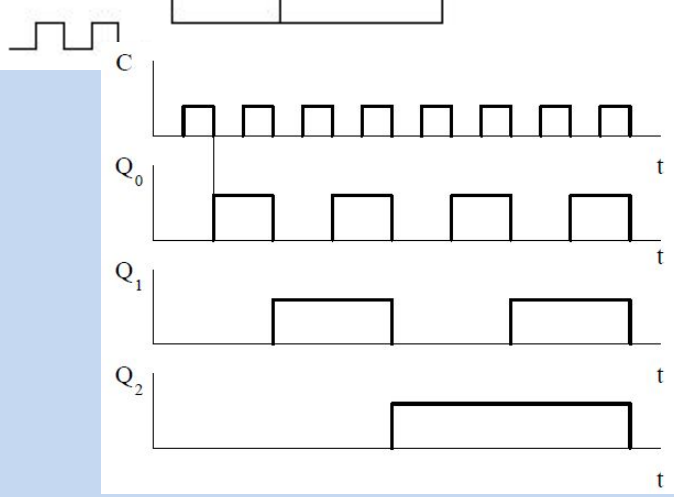
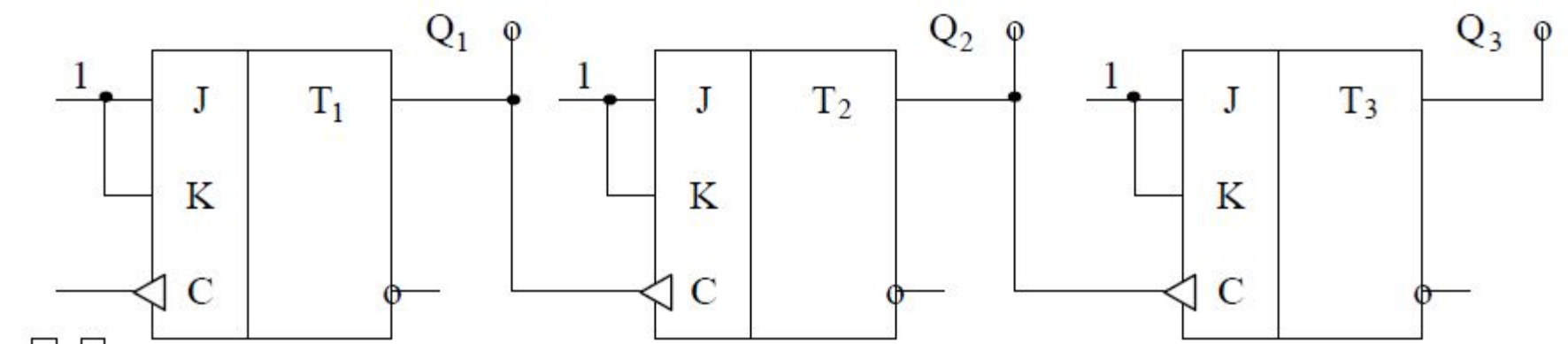


Рис. 4.20. Схема суммирующего двоичного счетчика (а) и временная диаграмма его работы (б)

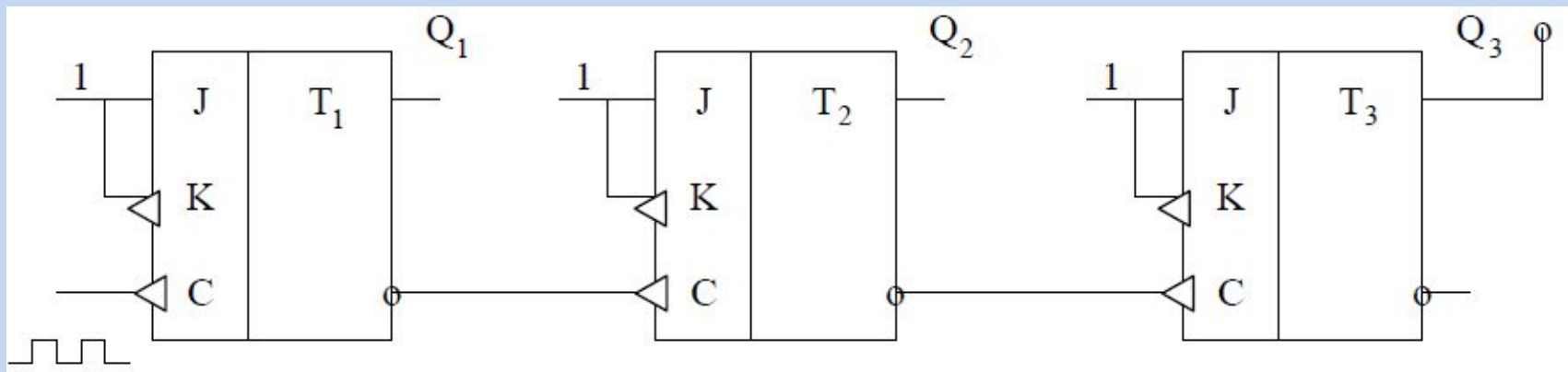


Рис. 4.21. Схема вычитающего двоичного счетчика

4.12.1. Параллельные счетчики

На рис. 4.20, а и 4.21 показаны схемы двоичных последовательных счетчиков, т. е. таких счетчиков, в которых при изменении состояния определенного триггера возбуждается последующий триггер, причем триггеры меняют свои состояния не одновременно, а последовательно. Если в данной ситуации должны изменить свои состояния n триггеров, то для завершения этого процесса потребуется n интервалов времени, соответствующих времени изменения состояния каждого из триггеров. Такой последовательный характер работы является причиной двух недостатков последовательного счетчика: меньшая скорость счета по сравнению с параллельными счетчиками и возможность появления ложных сигналов на выходе схемы. В параллельных счетчиках синхронизирующие сигналы поступают на все триггеры одновременно.

Последовательный характер переходов триггеров счетчика является источником ложных сигналов на его выходах. Например, в счетчике, ведущем счет в четырехразрядном двоичном коде с «весами» 8421, при переходе от числа $7_{10} = 0111_2$ к числу $8_{10} = 1000_2$ на выходе появится следующая последовательность сигналов:

$$0111 \Rightarrow 0110 \Rightarrow 0100 \Rightarrow 0000 \Rightarrow 1000.$$

Это означает, что при переходе из состояния 7 в состояние 8 на входах счетчика на короткое время появятся состояния 6; 4; 0. Эти дополнительные состояния могут вызвать ложную работу других устройств.

С целью уменьшения времени протекания переходных процессов, схему, приведенную на рис. 4.20, а, можно реализовать в варианте с подачей входных импульсов одновременно на все триггеры. В этом случае получим параллельный суммирующий счетчик (рис. 4.22).

Здесь на информационные входы триггеров подаются сигналы, являющиеся логической функцией состояния счетчика и определяющие конкретные триггеры, которые изменяют свое состояние при данном входном импульсе. Принцип стробирования сводится к следующему: триггер меняет свое состояние при поступлении очередного импульса синхронизации, если все предыдущие триггеры находились в состоянии логической единицы.

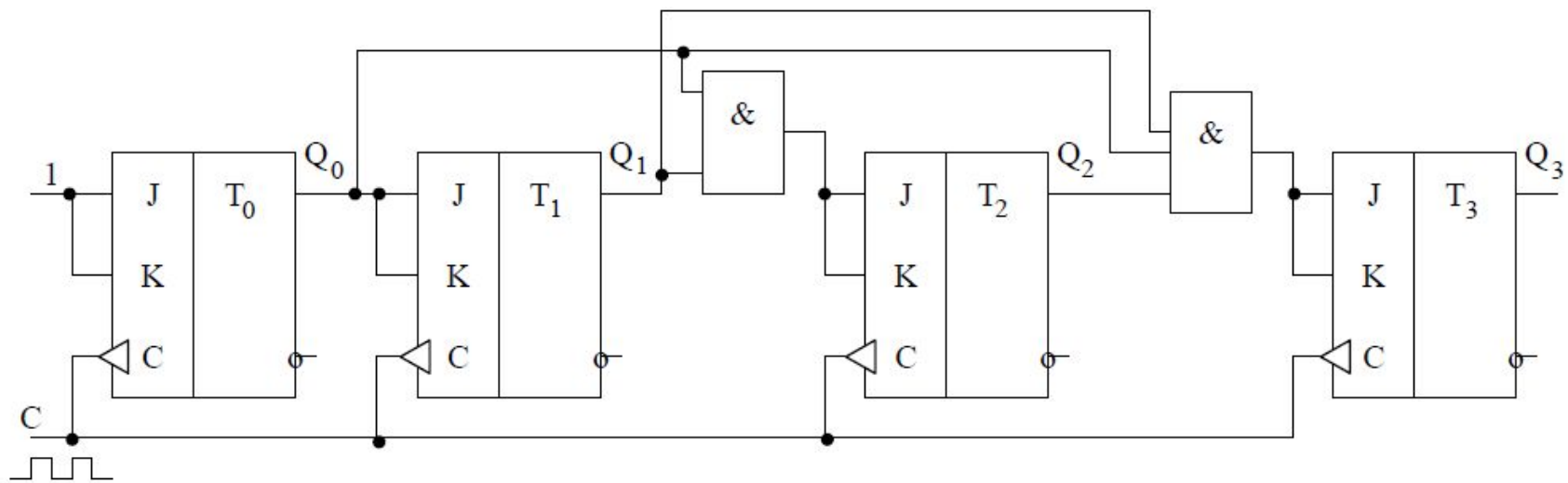


Рис. 4.22. Схема параллельного суммирующего счетчика

Параллельные счетчики имеют более высокое быстродействие по сравнению с последовательными, поскольку синхронизирующие импульсы поступают на все триггеры одновременно.

4.12.2. Реверсивные счетчики

Реверсивные счетчики позволяют суммировать или вычитать входные импульсы в зависимости от значений сигналов управления. Он реализуется путем объединения схем управления входами суммирующего и вычитающего счетчиков.

Реверсивные счетчики выпускаются в виде отдельных ИМС. Например, ИМС типа 155 ИЕ7 представляет собой реверсивный четырехразрядный двоичный счетчик. На рис. 4.23 представлена схема реверсивного восьмиразрядного двоичного счетчика, построенного на двух ИМС типа 155 ИЕ7. Разряд Q_8 является знаковым. Счет ведется в дополнительных кодах.

Каждая ИМС имеет по два счетных входа C_1 и C_2 , обеспечивающих соответственно работу в режиме суммирования и вычитания. Изменения состояния счетчика происходит в момент поступления переднего фронта импульса на вход C_1 или C_2 . Если счетчик работает в режиме суммирования, то изменяется сигнал на входе C_1 . Сигнал на входе C_2 должен иметь при этом высокий логический уровень. При работе в режиме вычитания изменяется сигнал на входе C_2 , а вход C_1 находится в состоянии логической единицы.

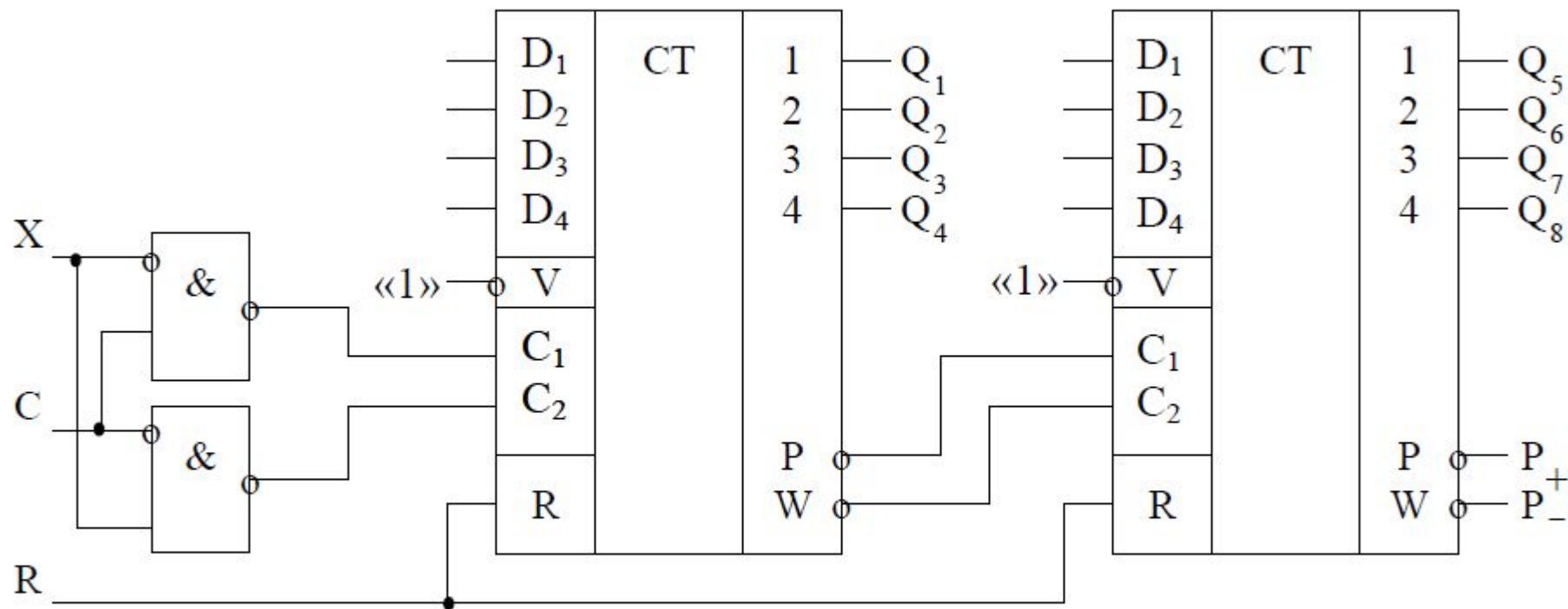


Рис. 4.23. Схема реверсивного восьмиразрядного счетчика на микросхемах 155IE7

ИМС 155ИЕ7 имеет два выхода переполнения: выход Р (переноса при суммировании) и выход W (заема при вычитании). Сигналы Р и W формируются аналогичным образом. На выходе переноса Р отрицательный спад импульса образуется при переполнении счетчика, т. е. при появлении на выходах $Q_1 - Q_4$ максимального числа и условии, что тактирующий сигнал на входе С имеет нулевое состояние. На выходе заема W импульс формируется при появлении на выходах всех разрядов $Q_1 - Q_4$ счетчика нулевого логического уровня, при этом вход C_2 должен находиться в состоянии логического нуля. Микросхема имеет входы $D_1 - D_4$, по которым возможна параллельная запись четырехразрядных кодов при условии, что на входы V и R поданы сигналы нулевого логического уровня.

Контрольные вопросы

1. Какое принципиальное отличие последовательных устройств от комбинационных?
2. В чем отличие асинхронных и синхронных триггеров?
3. Какие входы может иметь регистр памяти?
4. Какие функции может выполнять регистр сдвига?
5. Какие способы используются для построения счетчиков по модулю не кратному степени два?
6. Какие основные недостатки счетчиков последовательного типа?

3.3. Комбинационные сумматоры

Рассмотрим логическую схему полусумматора, предназначенную для суммирования двух одноразрядных двоичных чисел. Логическая функция полусумматора задается следующей таблицей истинности (табл. 3.3).

Таблица 3.3

X_1	X_2	S'	P'
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Здесь X_1 и X_2 — переменные, S' — частичная сумма (сумма по модулю два), P' — перенос в старший разряд. Из таблицы истинности видно, что

$$S' = X_1 \oplus X_2, P' = X_1 X_2,$$

т. е. полусумматор имеет два выхода (S' , P') и реализуется с помощью устройства неравнозначности и схемы И. Функциональная схема полусумматора представлена на рис. 3.4 (а, б).

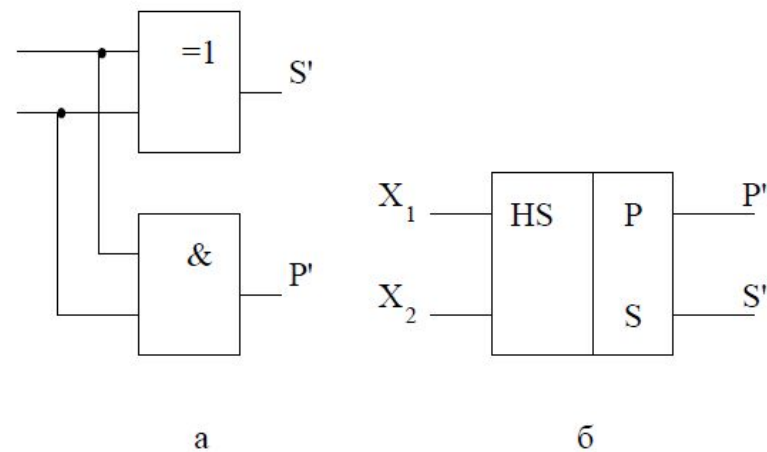
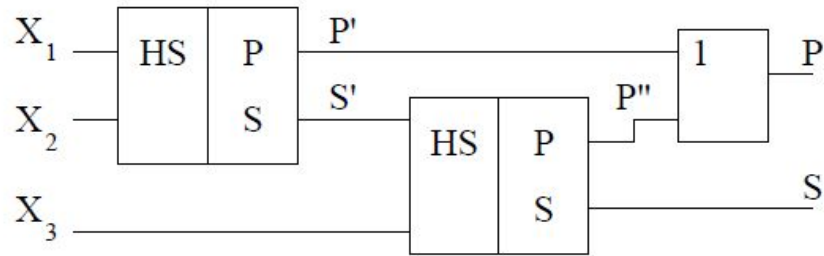


Рис. 3.4. Функциональная схема полусумматора (а)
и его условное графическое обозначение (б)

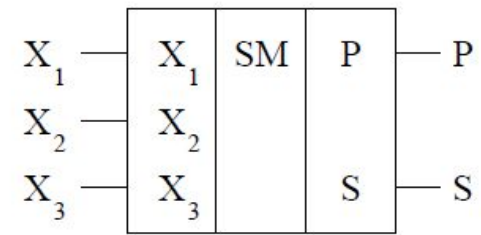
Два одноразрядных полусумматора образуют одноразрядный сумматор. Сумматор — логическая схема, имеющая три входа (входы разрядов X_1 и X_2 суммируемых чисел и вход X_3 для возможного переноса, образованного при сложении младших разрядов). Сумматор может быть образован с помощью двух полусумматоров и элемента ИЛИ. Таблица истинности (табл. 3.4) и функциональная схема сумматора показаны на рис. 3.5. В суммирующем устройстве последовательного действия для суммирования многоразрядных чисел перенос P подается на вход X_3 с задержкой на один такт сложения. Для ускорения процесса сложения двух n -разрядных чисел можно построить суммирующее устройство параллельного действия, в которых используются n сумматоров одноразрядных чисел. В настоящее время одно-, двух- и четырехразрядные двоичные сумматоры выпускаются в виде отдельных ИМС

Таблица 3.4

X_1	X_2	X_3	S	P
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



а



б

Рис. 3.5. Функциональная схема (а)

и условное графическое обозначение сумматора (б)

Контрольные вопросы

1. Составить таблицу истинности переключательных функций:

$$y_1 = x_1 + x_2, \quad y_2 = x_1 \cdot x_2.$$

2. Построить логические схемы на элементах И, ИЛИ, НЕ, реализующих переключательные функции $y_1 = x_1 \cdot x_2 + x_1 \cdot \bar{x}_2$, $y_2 = x_1 + x_2 + x_1 \cdot x_2$.

3. Составить таблицу истинности переключательных функций:

$$y_1 = x_1 + x_2, \quad y_2 = x_1 \cdot x_2.$$

4. Как по статической передаточной характеристике ЛЭ определить запас помехоустойчивости?

5. Перечислить факторы, обеспечивающие высокое быстродействие ЭСЛ-логических элементов. Объяснить назначение эмиттерных повторителей.

5. ЦИФРО-АНАЛОГОВЫЕ И АНАЛОГОВО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ

5.1. Цифро-аналоговые преобразователи

Цифро-аналоговые преобразователи (ЦАП) – это преобразователи код-напряжение (ПКН). Задача рассматриваемого преобразователя состоит в выработке напряжения U на выходе ПКН, пропорционального входному двоичному числу N . Используя известную форму представления двоичных чисел

$$N = a_0 2^0 + a_1 2^1 + \dots + a_{n-1} 2^{n-1} = \sum_{i=0}^{n-1} a_i 2^i,$$

запишем операцию, выполняемую ПКН, в виде

$$U(N) = \frac{U_{\max}}{N_{\max}} N = U_{\max} \sum_{i=0}^{n-1} \frac{2^i}{2^{n-1}} a_i = \sum_{i=0}^{n-1} U_i a_i, \quad (5.1)$$

где отношение максимального выходного напряжения U_{\max} к максимальному входному числу $N_{\max} = 2^n - 1$ играет роль масштабного коэффициента, определяемого как напряжение, соответствующее единице входного числа, т. е. $U(1) = U_{\max} / N_{\max}$.

Из выражения (5.1) следует, что операция преобразования код-напряжение сводится к суммированию элементарных напряжений:

$$U_i = U_{\max} 2^i / 2^n - 1 \approx U_{\max} / 2^{n-i} = U_{\max} K_i, \quad (5.2)$$

которые образуются путем деления некоторого эталонного напряжения U_{\max} с помощью резистивных делителей с коэффициентом передачи $K_i = 1/2^{n-i}$. При этом из операции суммирования исключаются те слагаемые U_i , которые соответствуют нулевым значениям элементов a_i , составляющих входное двоичное число ($a_{n-1} a_{n-2} \dots a_1 a_0$).

Для реализации этого принципа преобразования на практике используются две разновидности схем, представленные на рис. 5.1 а, б. Первую схему (рис. 5.1, а) называют ПКН с двоично взвешенными резистивными цепями (или схемой с суммированием напряжений), а вторую (рис. 5.1, б) ПКН с многозвенной резистивной цепью типа R-2R. В обеих схемах при появлении единицы в i -м разряде двоичного числа эталонное напряжение U_{\max} проходит через соответствующий замкнутый ключ и резистивную цепь на выход. Различие этих схем в том, как формируется требуемый коэффициент деления эталонного напряжения с помощью резистивной матрицы.

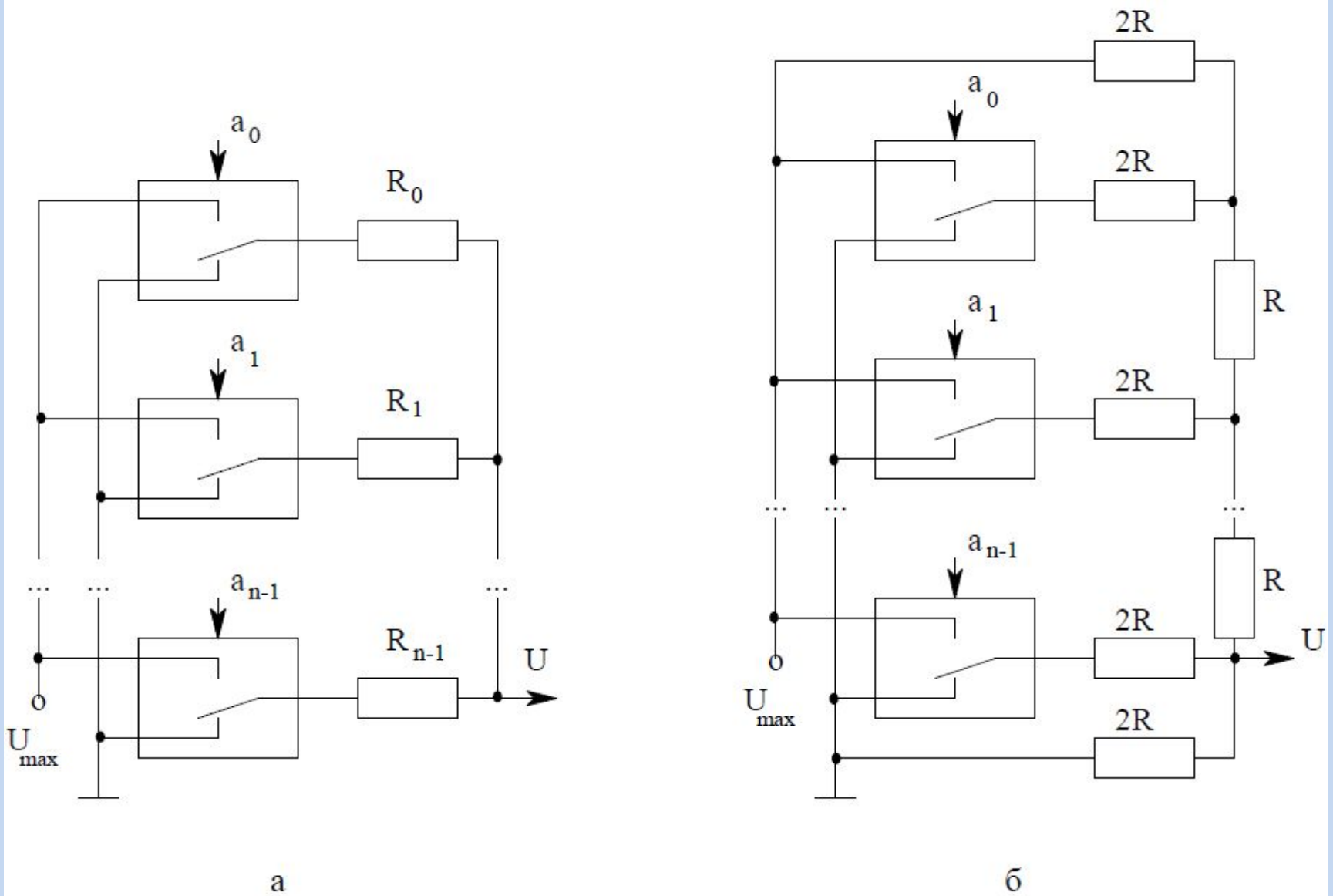


Рис. 5.1. Схемы ПКН с двоично взвешенными резистивными цепями (а) и многозвенной резистивной цепью типа R-2R (б)

В первой схеме (рис. 5.1, а) при $a_i = 1$ коэффициент передачи напряжения $K_i = U_i / U_{\max}$ определяется отношением проводимости $Y_i = 1 / R_i$ к суммарной проводимости всей цепи $Y_{\Sigma} = \sum_{i=0}^{n-1} Y_i$. Значения сопротивлений R_i в этой схеме задаются следующим образом: $R_{n-1} = R$, $R_{n-2} = 2R$, $R_i = 2^{n-i-1}R$, ..., $R_0 = 2^{n-1}R$.

Тогда

$$Y_{\Sigma} = \frac{1}{R} \sum_{i=0}^{n-1} 2^{-(n-i-1)} \approx \frac{2}{R}$$

и, следовательно, $K_i = U_i / U_{\max} = Y_i / Y_{\Sigma} = 1/2^{n-i}$. Это значит, что заданная совокупность сопротивлений удовлетворяет соотношению (5.2), которое определяет правило формирования коэффициентов передачи эталонного напряжения на выход ПКН.

По второй схеме ПКН (рис. 5.1, б) необходимый коэффициент передачи эталонного напряжения образуется с помощью многозвенного делителя напряжения.

В этой схеме коэффициент передачи напряжения от любого звена к следующему равен $1/2$, за исключением последнего звена, где коэффициент равен $1/3$.

Результирующий коэффициент передачи от i -го звена на выход

$$K_i = U_i / U_{\max} = \frac{1}{3} \left(\frac{1}{2} \right)^{n-i-1} = \frac{2}{3} \left(\frac{1}{2} \right)^{n-i}. \quad (5.3)$$

Таким образом, схема (рис. 5.1, б) также обеспечивает требуемое правило формирования коэффициентов, определяемое соотношением (5.3). Такая схема имеет коэффициент использования эталонного напряжения $2/3$, в отличие от схемы (рис. 5.13, а), в которой этот коэффициент равен 1.

Однако, несмотря на этот недостаток и на большое число элементов схемы, преимуществом схемы ПКН с резистивной цепью типа R-2R является то, что для выполнения такой схемы требуется всего два номинала резисторов. Это существенно упрощает практическую реализацию таких схем, особенно при большой разрядности схем ПКН, когда предъявляются высокие требования к точности подбора номиналов резисторов. Поэтому схемы ПКН, использующие резистивные цепи типа R-2R, широко применяются на практике.

Одной из важнейших задач, связанных с разработкой и применением ПКН, является обеспечение требуемой скорости преобразования, или быстродействия схемы. Это важно тогда, когда ПКН является элементом более сложных устройств, например, преобразователей напряжения в код, характеристики которых в значительной степени зависят от характеристик используемых в них ПКН.

В рассмотренных схемах ПКН время выполнения операции преобразования определяется быстродействием ключевых схем и переходными процессами в резистивных цепях, обусловленными наличием паразитных емкостей. Второй фактор для этих схем является основным, так как значения сопротивлений обычно выбирают довольно большими, чтобы пренебречь погрешностями, вносимыми конечным сопротивлением электронных ключей. Следует отметить, что схема (рис. 5.1, б) обладает более низким быстродействием, чем схема (рис. 5.1, а), так как содержит больше паразитных емкостей и в ней используется многозвенный принцип передачи напряжения. Этот недостаток схемы с резистивной цепью типа R-2R является причиной того, что схеме с двоично взвешенной резистивной цепью нередко отдается предпочтение в практических применениях.

Для обеспечения более высокой скорости преобразования на практике используются схемы ПКН, основанные на принципе суммирования токов (рис. 5. 2).

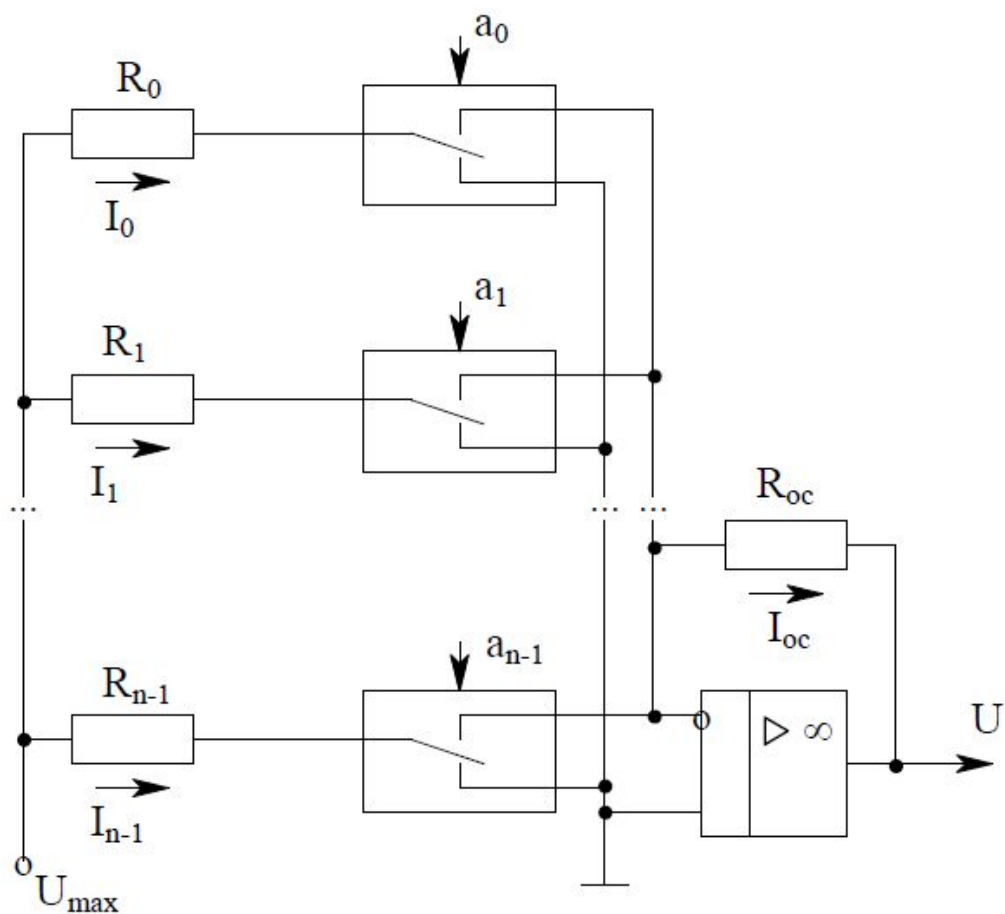


Рис. 5.2. Схема ПКН с суммированием токов

Такая схема отличается от рассмотренных расположением ключевых элементов и наличием ОУ с отрицательной обратной связью. Благодаря большим значениям коэффициента усиления и входного сопротивления ОУ, можно считать, что напряжение на его входе и входной ток ОУ близки к нулю. Следовательно все токи i_k , текущие через элементы резистивной цепи, уравниваются током I_{oc} , текущем в цепи обратной связи, и выходное напряжение ПКН равно

$$U(N) = R_{oc} I_{oc} = R_{oc} \sum_{k=0}^{n-1} a_k i_k, \quad (5.4)$$

т. е. операционный усилитель выполняет операцию суммирования токов, которые определяются значениями сопротивлений в тех разрядах ПКН, где $a = 1$. Значения сопротивлений в схеме выбирают согласно правилу

$$R_k = R_{n-1} 2^{n-k-1}.$$

При этом

$$U(N) = R_{oc} \sum_{k=0}^{n-1} a_k \frac{U_{max}}{R_k} = \frac{2R_{oc}}{R_{n-1}} U_{max} \sum_{k=0}^{n-1} a_k \frac{1}{2^{n-k}}, \quad (5.5)$$

Преимуществом схемы ПКН, использующей принцип суммирования токов, является ее более высокое быстродействие. Это объясняется тем, что в этой схеме токи, проходящие через резисторы, практически не меняются при изменении состояния ключей, так как входные напряжения и ток операционного усилителя близки к нулю, а это означает, что паразитные емкости при переключениях ключей не перезаряжаются.

5.2. Аналого-цифровые преобразователи

В основе построения преобразователей напряжение-код (ПНК) лежат в основном три известных принципа преобразования, определяющих алгоритм функционирования и структуру соответствующих устройств последовательного счета, поразрядного кодирования, считывания. Согласно этой классификации рассмотрим три основные разновидности схем ПНК.

Принцип последовательного счета реализуется с помощью схемы, представленной на рис. 5.3.

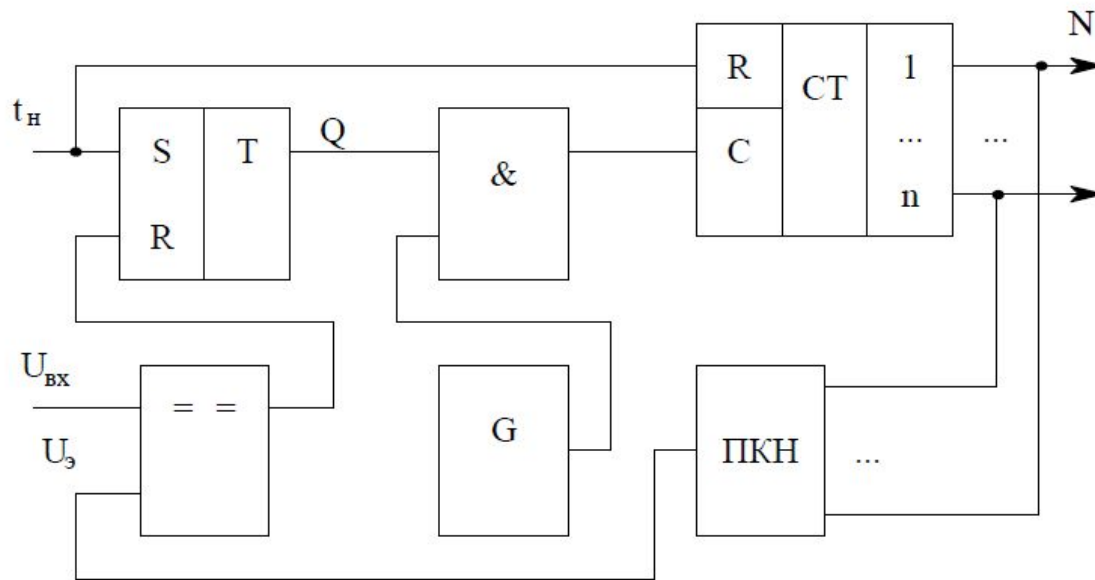


Рис. 5.3. Функциональная схема АЦП последовательного счета

1.4. Основные обозначения на схемах

Для изображения электронных устройств и их узлов применяется три основных типа схем:

- принципиальная схема;
- структурная схема;
- функциональная схема.

Различаются они своим назначением и, самое главное, степенью детализации изображения устройств.

Принципиальная схема — наиболее подробная. Она обязательно показывает все использованные в устройстве элементы и все связи между ними. Если схема строится на основе микросхем, то должны быть показаны номера выводов всех входов и выходов этих микросхем. Принципиальная схема должна позволять полностью воспроизвести устройство. Обозначения **принципиальной** схемы наиболее жестко стандартизованы, отклонения от стандартов не рекомендуются.

Структурная схема — наименее подробная. Она предназначена для отображения общей структуры устройства, то есть его основных блоков, узлов, частей и главных связей между ними. Из структурной схемы должно быть понятно, зачем нужно данное устройство и что оно делает в основных режимах работы, как взаимодействуют его части. Обозначения структурной схемы могут быть довольно произвольными, хотя некоторые общепринятые правила все-таки лучше выполнять.

Функциональная схема представляет собой гибрид структурной и принципиальной. Некоторые наиболее простые блоки, узлы, части устройства отображаются на ней, как на структурной схеме, а остальные — как на принципиальной схеме. Функциональная схема дает возможность понять всю логику работы устройства, все его отличия от других подоб-

ных устройств, но не позволяет без дополнительной самостоятельной работы воспроизвести это устройство. Что касается обозначений, используемых на функциональных схемах, то в части, показанной как структура, они не стандартизованы, а в части, показанной как принципиальная схема, — стандартизованы.

Положительный сигнал (сигнал положительной полярности) - это сигнал, активный уровень которого — логическая единица. То есть нуль - это отсутствие сигнала, единица — сигнал пришел (рис. 1.13).



Рис. 1.13. Элементы цифрового сигнала

Отрицательный сигнал (сигнал отрицательной полярности) — это сигнал, активный уровень которого — логический нуль. То есть единица — это отсутствие сигнала, нуль — сигнал пришел (рис. 1.13).

Активный уровень сигнала — это уровень, соответствующий приходу сигнала, то есть выполнению этим сигналом соответствующей ему функции.

Пассивный уровень сигнала — это уровень, в котором сигнал не выполняет никакой функции.

Инвертирование или инверсия сигнала — это изменение его полярности.

Инверсный выход — это выход, выдающий сигнал инверсной полярности по сравнению с входным сигналом.

Прямой выход — это выход, выдающий сигнал такой же полярности, какую имеет входной сигнал.

Положительный фронт сигнала - это переход сигнала из нуля в единицу.

Отрицательный фронт сигнала (спад) — это переход сигнала из единицы в ноль.

Передний фронт сигнала — это переход сигнала из пассивного уровня в активный.

Задний фронт сигнала — это переход сигнала из активного уровня в пассивный.

Тактовый сигнал (или строб) — управляющий сигнал, который определяет момент выполнения элементом или узлом его функции.

Шина — группа сигналов, объединенных по какому-то принципу, например, шиной называют сигналы, соответствующие всем разрядам какого-то двоичного кода.

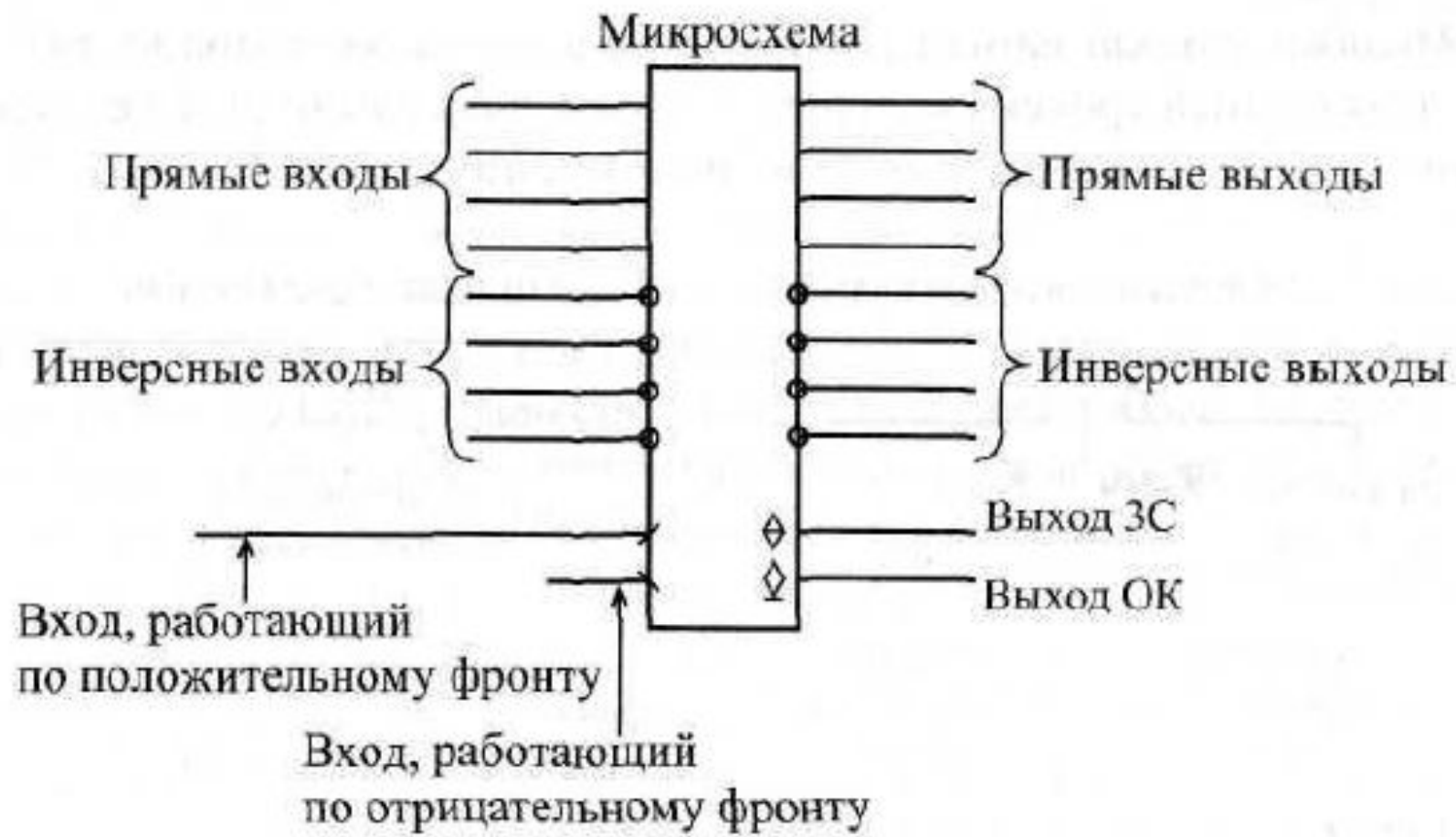


Рис. 1.14. Обозначение входов и выходов

Наконец, если у микросхемы необходимо показать неинформационные выводы, то есть выводы, не являющиеся ни логическими входами, ни логическими выходами, то такой вывод помечается косым крестом (две перпендикулярные линии под углом 45°). Это могут быть, например, выводы для подключения внешних элементов (резисторов, конденсаторов) или выводы питания (рис. 1.15).

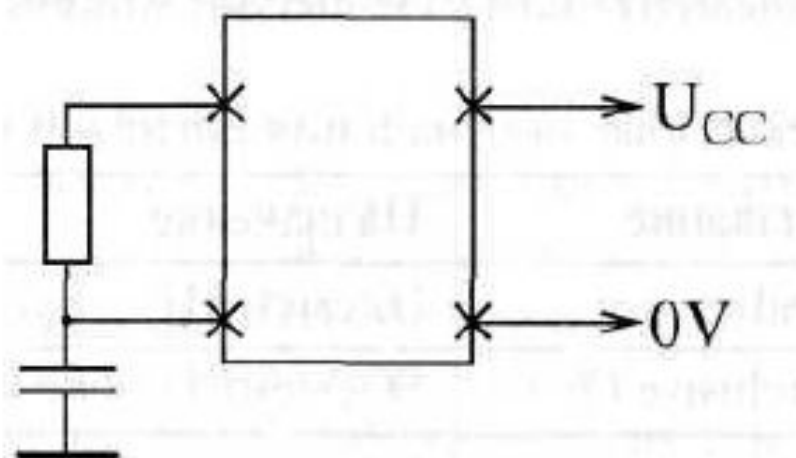


Рис. 1.15. Обозначение неинформационных выводов

Таблица 1.2. Некоторые обозначения сигналов и микросхем

Обозначение	Название	Назначение
&	And	Элемент И
=1	Exclusive Or	Элемент Исключающее ИЛИ
1	Or	Элемент ИЛИ
A	Address	Адресные разряды
BF	Buffer	Буфер
C	Clock	Тактовый сигнал (строб)
CE	Clock Enable	Разрешение тактового сигнала
CT	Counter	Счетчик
CS	Chip Select	Выбор микросхемы
D	Data	Разряды данных, данные
DC	Decoder	Дешифратор
EZ	Enable Z-state	Разрешение третьего состояния
G	Generator	Генератор
I	Input	Вход
I/O	Input/Output	Вход/Выход
OE	Output Enable	Разрешение выхода
MS	Multiplexer	Мультиплексор
Q	Quit	Выход
R	Reset	Сброс (установка в нуль)
RG	Register	Регистр
S	Set	Установка в единицу
SUM	Summator	Сумматор
T	Trigger	Триггер
TC	Terminal Count	Окончание счета
Z	Z-state	Третье состояние выхода

1.8. Функции цифровых устройств

Любое цифровое устройство от самого простейшего до самого сложного всегда действует по одному и тому же принципу (рис. 1.23). Оно принимает входные сигналы, выполняет их обработку, передачу, хранение и выдает выходные сигналы. При этом совсем не обязательно любое изменение входных сигналов приводит к немедленному и однозначному изменению выходных сигналов. Реакция устройства может быть очень сложной, отложенной по времени, неочевидной, но суть от этого не меняется.

В качестве входных сигналов нашего устройства могут выступать сигналы с выходов других цифровых устройств, с тумблеров и клавиш или с датчиков физических величин. Причем в последнем случае, как правило, необходимо преобразование аналоговых сигналов с датчиков в потоки цифровых кодов (рис. 1.24) с помощью аналого-цифровых преобразователей (АЦП). Например, в случае персонального компьютера входными сигналами являются сигналы с клавиатуры, с датчиков перемещения мыши, с микрофона (давление воздуха, то есть звук, преобразуется в аналоговый электрический сигнал, а затем — в цифровые коды), из кабеля локальной сети и т. д.

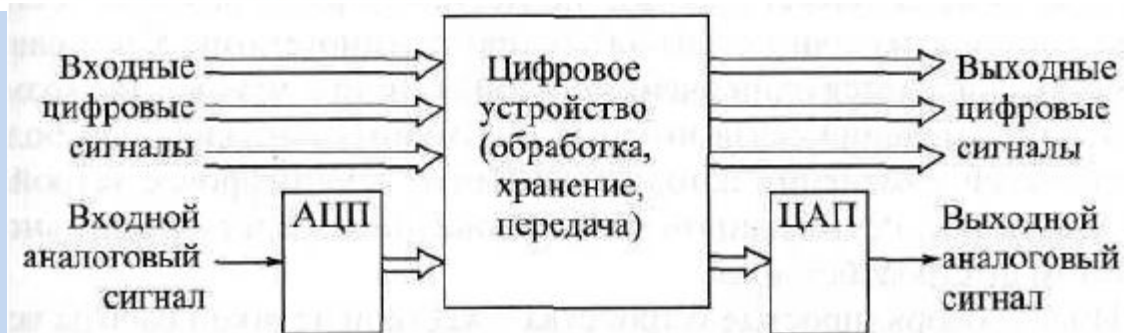


Рис. 1.23. Включение цифрового устройства

6.2. Принципы фон Неймана

Со времени появления в 40-х гг. XX в. первых электронных цифровых вычислительных машин технология их производства была значительно усовершенствована. В последние годы благодаря развитию интегральной технологии существенно улучшились их характеристики. Однако несмотря на успехи, достигнутые в области технологии, существенных изменений в базовой структуре и принципах работы вычислительных машин не произошло. Так, в основу построения подавляющего большинства современных компьютеров положены общие принципы функционирования универсальных вычислительных устройств, сформулированные американским ученым *Джоном фон Нейманом* в 1945 году.

Согласно фон Нейману, для того чтобы ЭВМ была универсальным и эффективным устройством обработки информации, она должна строиться в соответствии со следующими принципами:

1. Информация кодируется в двоичной форме и разделяется на единицы (элементы) информации, называемые *словами*.

Использование в ЭВМ двоичных кодов продиктовано в первую очередь спецификой электронных схем, применяемых для передачи, хранения и преобразования информации. Как уже отмечалось, в этом случае конструкция ЭВМ предельно упрощается и ЭВМ работает наиболее надежно (устойчиво). Совокупности нулей и единиц (битов информации), используемые для представления отдельных чисел, команд и т. п., рассматриваются как самостоятельные информационные объекты и называются словами. Слово обрабатывается в ЭВМ как одно целое — как машинный элемент информации.

2. Разнотипные слова информации хранятся в одной и той же памяти и различаются по способу использования, но не по способу кодирования.

Все слова, представляющие числа, команды и прочие объекты, выглядят в ЭВМ совершенно одинаково и сами по себе неразличимы. Только порядок использования слов в программе вносит различия в них. Благодаря такому «однообразию» слов оказывается возможным использовать одни и те же операции для обработки слов различной природы, например для обработки и чисел, и команд, т. е. команды программы становятся в такой же степени доступными для отработки, как и числа.

Перечисленные принципы функционирования ЭВМ предполагают, что компьютер должен иметь следующие устройства:

- *арифметико-логическое устройство (АЛУ)*, выполняющее арифметические и логические операции;

- *устройство управления (УУ)*, которое организует процесс выполнения программы;

- *запоминающее устройство (ЗУ)*, или память для хранения программ и данных;

- *внешние устройства* для ввода (устройства ввода) и вывода (устройства вывода) информации.

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Под *архитектурой ЭВМ* понимают ее логическую организацию, состав и назначение ее функциональных средств, принципы кодирования и т. п., т. е. все то, что однозначно определяет процесс обработки информации на данной ЭВМ. ЭВМ, построенные в соответствии с принципами фон Неймана, называют фоннеймановскими, или компьютерами фоннеймановской (классической) архитектуры.

Структура ЭВМ – совокупность элементов компьютера и связей между ними.

Ввиду большой сложности современных ЭВМ принято представлять их структуру иерархически, т. е. понятие «элемент» жестко не фиксируется. Так, на самом высоком уровне сама ЭВМ может считаться элементом. На следующем (программном) уровне иерархии элементами структуры ЭВМ являются память, процессор, устройства ввода-вывода и т. д. На более низком уровне (микропрограммном) элементами служат узлы и блоки, из которых строятся память, процессор и т. д. Наконец, на самых низких уровнях элементами выступают интегральные логические микросхемы и электронные приборы.

Для увеличения функциональных возможностей устройств часто предусматривается использование нескольких сигналов управления. На рис. 3.6, б представлено графическое изображение микросхемы К155ИД4, содержащей двойной двухвходовой дешифратор с активными нулевыми выходными сигналами.

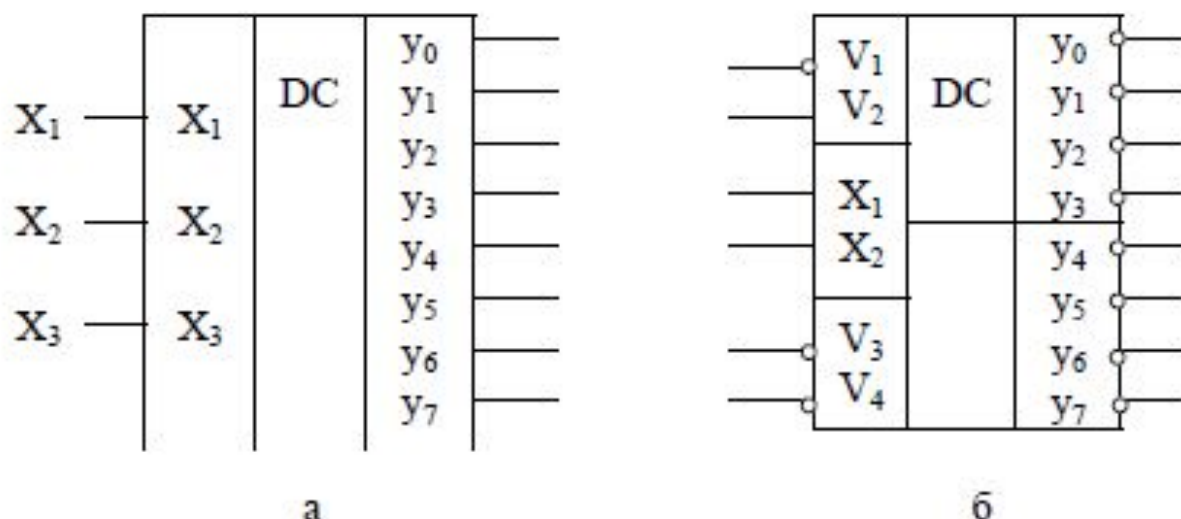


Рис. 3.6. Условное графическое обозначение дешифратора (а)
и условное графическое обозначение ИМС К155ИД4 (б)

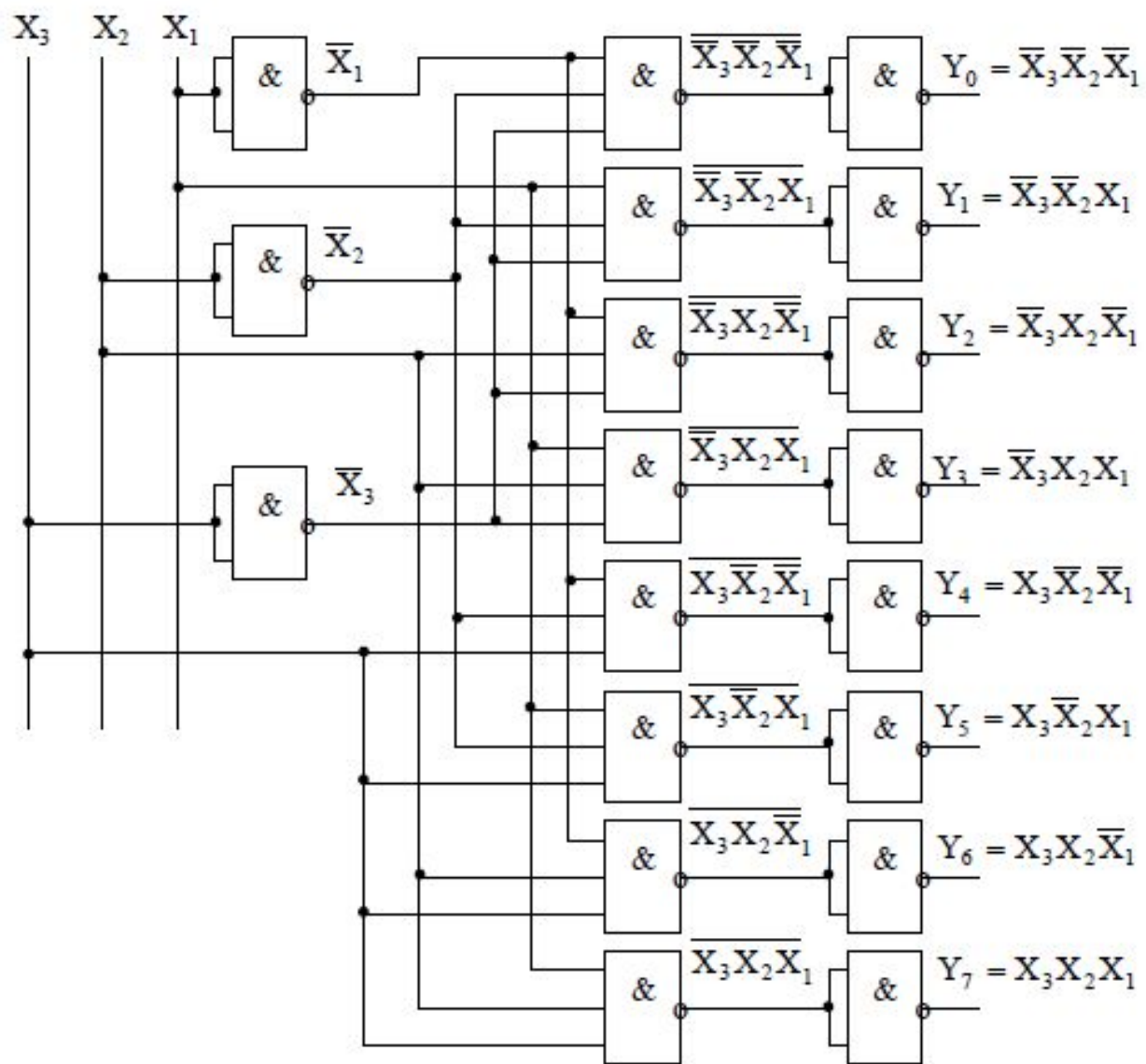


Рис. 3.7. Принципиальная схема линейного дешифратора на элементах И-НЕ

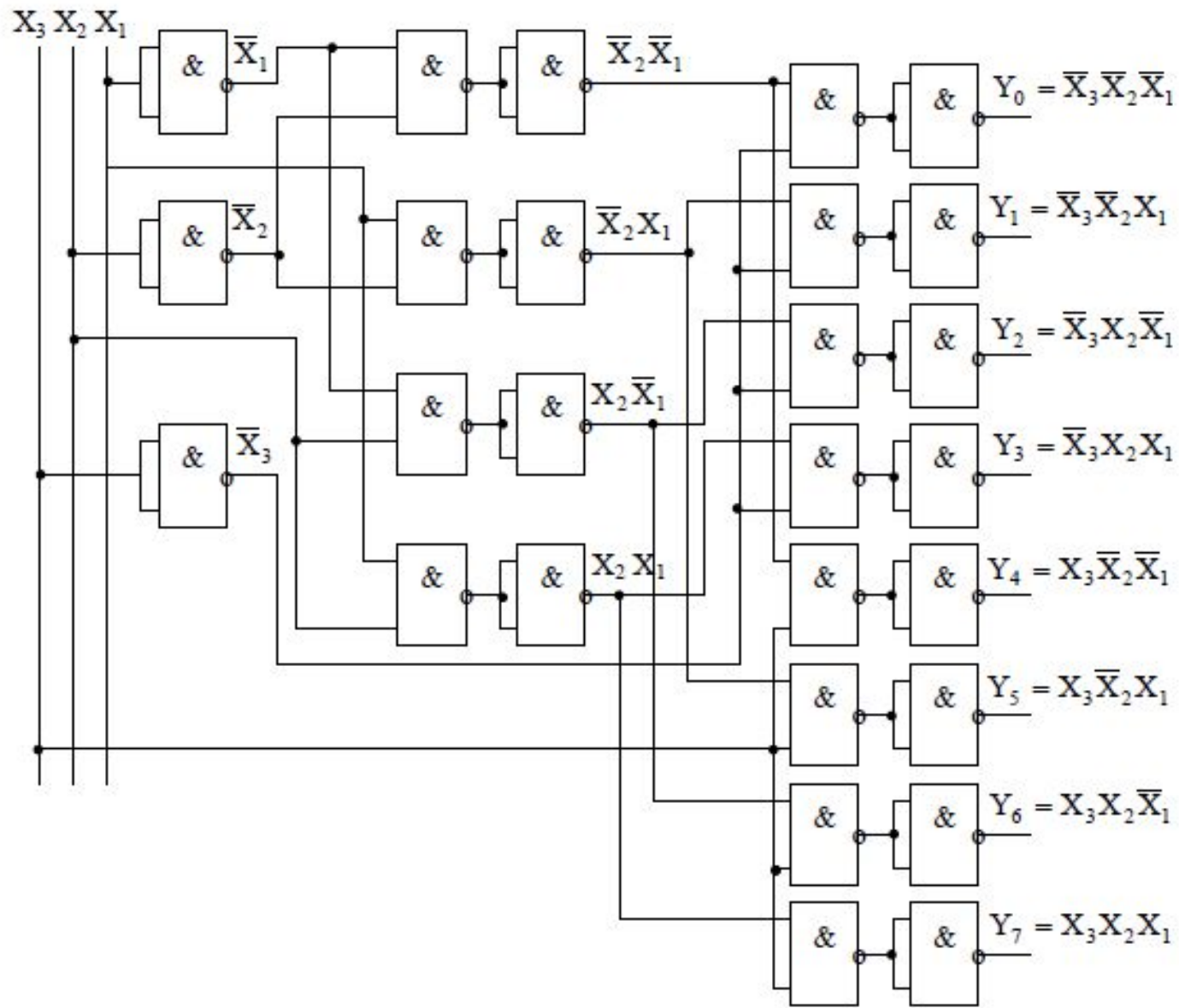


Рис. 3.8. Принципиальная схема пирамидального дешифратора

3.5. Шифраторы

Шифраторы выполняют задачи обратные дешифраторам. С их помощью активным значением определяемых входных сигналов можно поставить в соответствие заданные выходные коды (комбинации выходных сигналов). Полный шифратор имеет n выходов и $m = 2^n$ входов (рис. 3.10, а). Выходной сигнал шифратора в числовой интерпретации соответствует номеру возбужденного входа. Таблица истинности одного

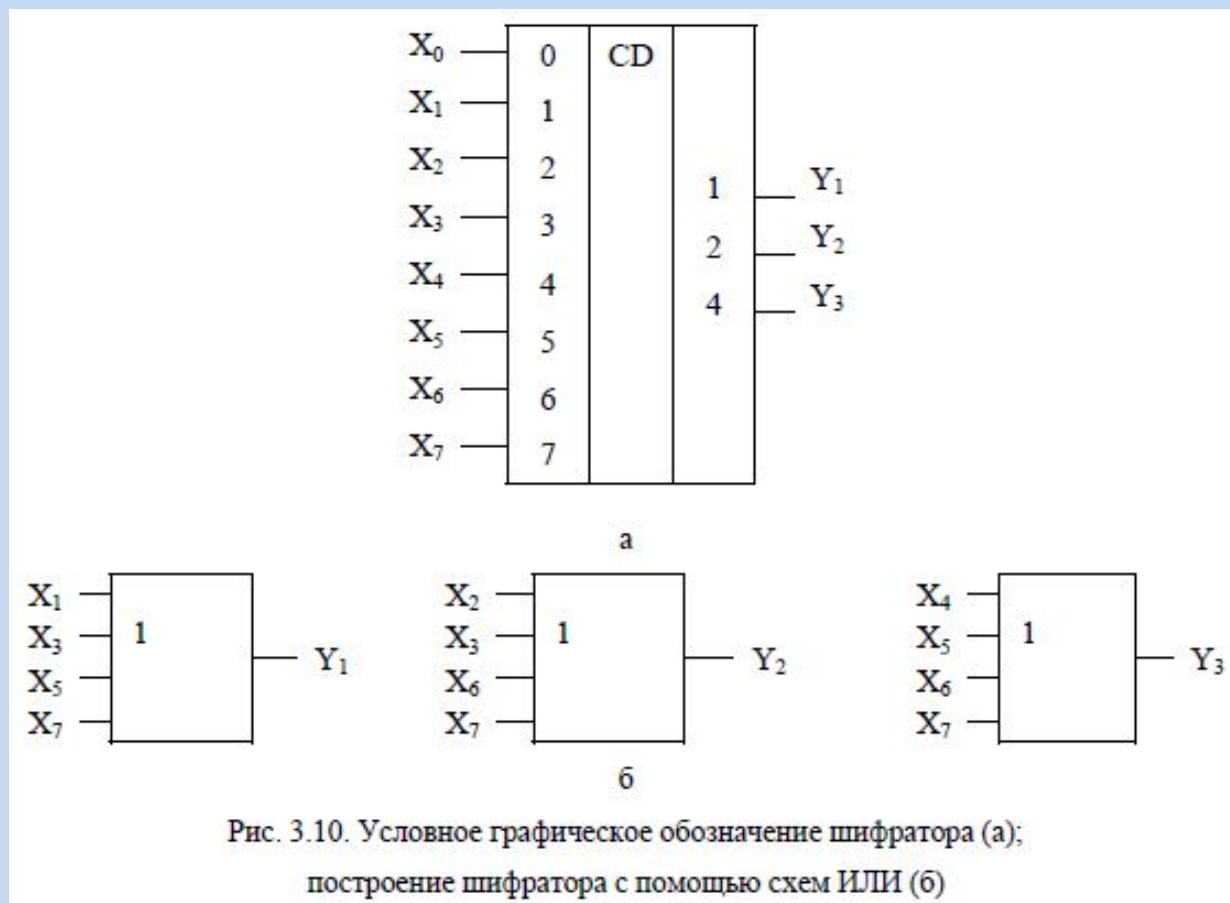


Рис. 3.10. Условное графическое обозначение шифратора (а);
построение шифратора с помощью схем ИЛИ (б)

Таблица 3.6

X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	Y ₃	Y ₂	Y ₁
1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0

При практической реализации шифратора может оказаться выгоднее использовать элементы И-НЕ. Тогда, в соответствии с двойственностью операций конъюнкции и дизъюнкции, входные переменные нужно брать с инверсией.

Рассмотренный шифратор обладает рядом недостатков:

- 1) невозможно наращивать информационную емкость (число входов и выходов), соединяя между собой шифраторы меньшей емкости;
- 2) для правильной работы шифратора необходимо, чтобы возбужден был всегда строго один из входов. Во всех других случаях получаемый на выходе сигнал не отражает реальной ситуации на входе.

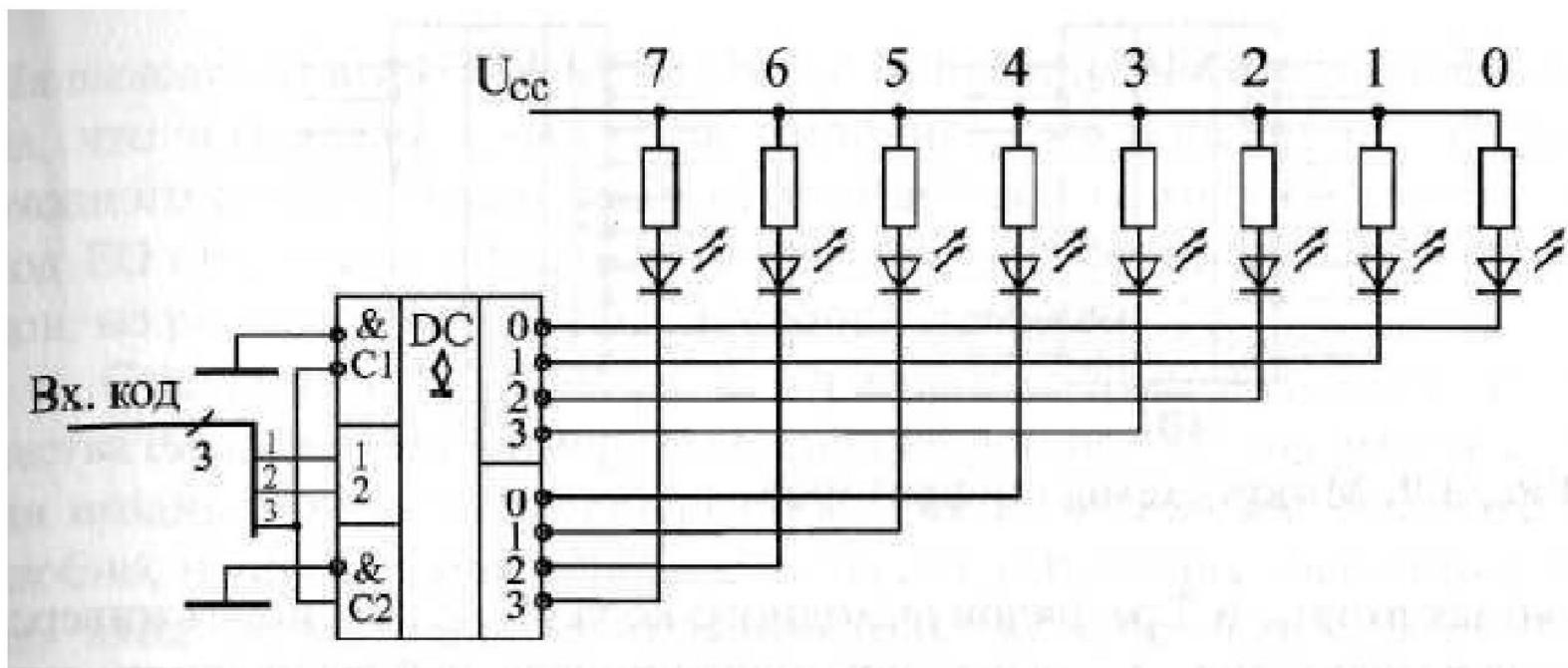


Рис. 3.7. Позиционная индикация на дешифраторе с выходами ОК

3.3. Компараторы кодов

Микросхемы компараторов кодов (английское Comparator) применяются для сравнения двух входных кодов и выдачи на выходы сигналов о результатах этого сравнения (о равенстве или неравенстве кодов). На схемах компараторы кодов обозначаются двумя символами равенства: " $=$ =". Код типа микросхемы компаратора кода в отечественных сериях - **СП**.

Примером такой микросхемы может служить СП1 — 4-х разрядный компаратор кодов, сравнивающий величины кодов и выдающий информацию о том, какой код больше, или о равенстве кодов (рис. 3.16).

Помимо восьми входов для сравниваемых кодов (два 4-х разрядных кода, обозначаемых $A_0 \dots A_3$ и $B_0 \dots B_3$), компаратор СП1 имеет три управляющих входа для наращивания разрядности ($A > B$, $A < B$, $A = B$) и три

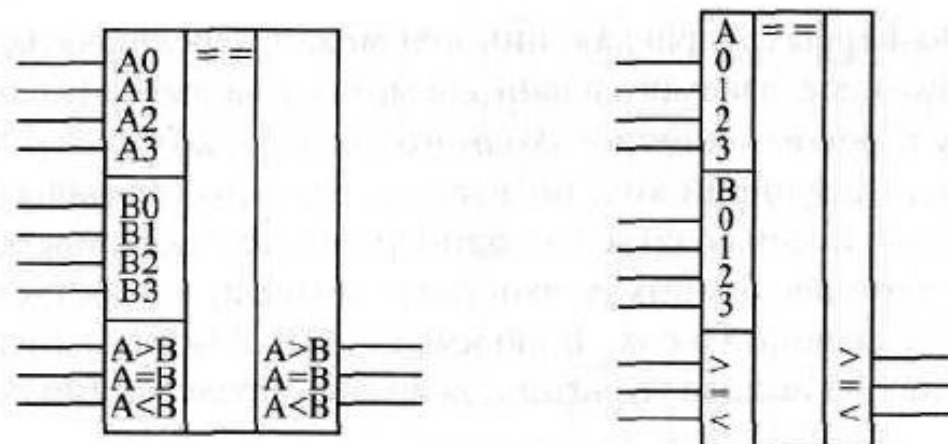


Рис. 3.16. 4-х разрядный компаратор кодов СП1 (два варианта обозначения)

Таблица 3.4. Таблица истинности компаратора СП1

A3, B3	Входы			Исходы нарастающая			Выходы		
	A2, B2	A1, B1	A0, B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	1	0	0
A3<B3	X	X	X	X	X	X	0	1	0
A3=B3	A2>B2	X	X	X	X	X	1	0	0
A3=B3	A2<B2	X	X	X	X	X	0	1	0
A3=B3	A2=B2	A1>B1	X	X	X	X	1	0	0
A3=B3	A2=B2	A1<B1	X	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	1	0	0	1	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	1	0	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	X	X	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	0	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	0	1	1	0

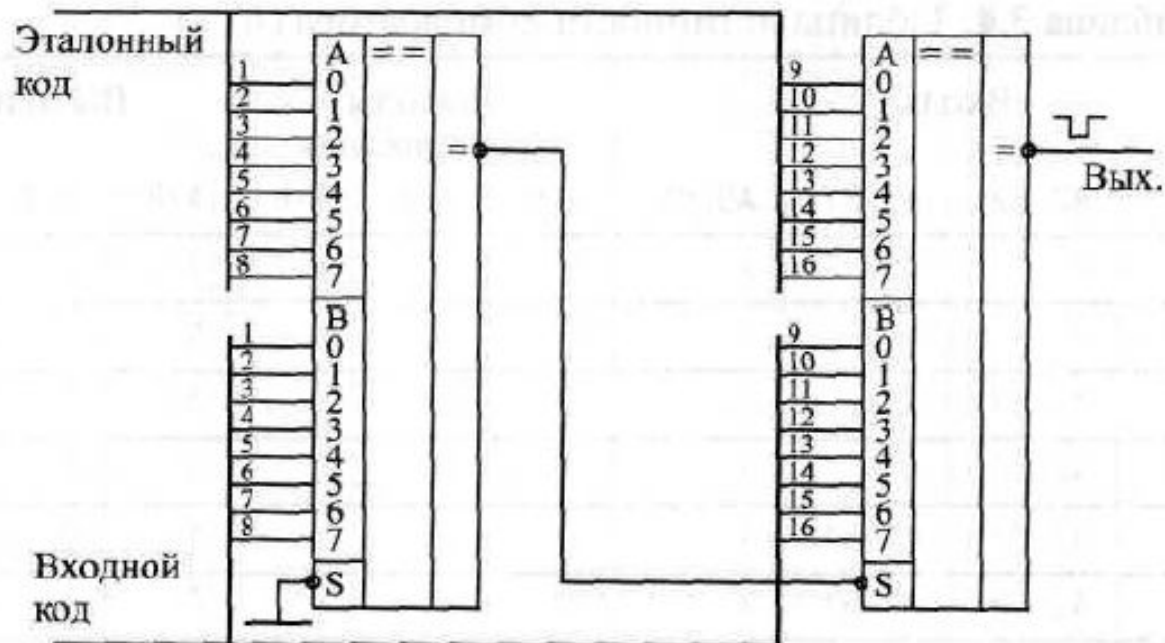


Рис. 3.18. Селектирование 16-разрядных кодов

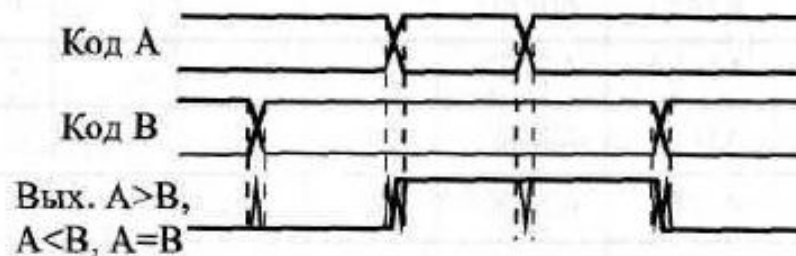


Рис. 3.19. Неопределенные состояния на выходах компаратора при изменении входных кодов

3.6. Мультиплексоры

Мультиплексоры обеспечивают коммутацию на выходе одного из нескольких информационных входных сигналов в соответствии с заданным кодом на управляющих входах. Условное графическое обозначение одного из вариантов мультиплексора представлено на рис. 3.12. В зависимости от комбинации сигналов управления X_1, X_2, X_3 он обеспечивает коммутацию одного из восьми информационных входных сигналов $D_0 - D_7$ на выход Y .

Сигнал синхронизации V в данном случае имеет нулевое активное значение, разрешающее передачу информации с одного из входов на выход.

Структурная формула, определяющая функционирование рассматриваемого восьмивходового мультиплексора, имеет вид:

$$Y = \bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{V} D_0 + \bar{X}_3 \bar{X}_2 X_1 \bar{V} D_1 + \bar{X}_3 X_2 \bar{X}_1 \bar{V} D_2 + \bar{X}_3 X_2 X_1 \bar{V} D_3 + X_3 \bar{X}_2 X_1 \bar{V} D_4 + X_3 \bar{X}_2 \bar{X}_1 \bar{V} D_5 + X_3 X_2 \bar{X}_1 \bar{V} D_6 + X_3 X_2 X_1 \bar{V} D_7. \quad (3.2)$$

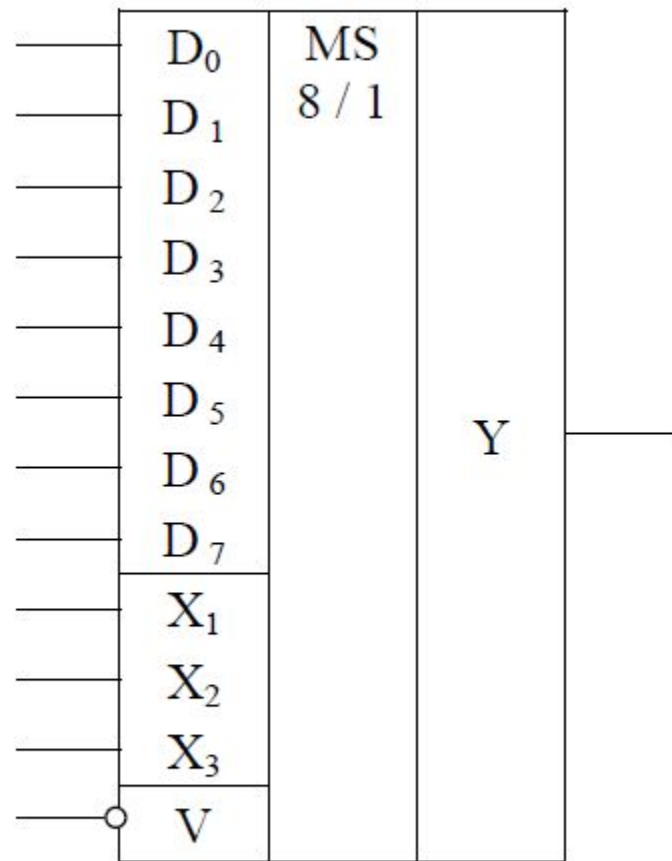


Рис. 3.12. Условное графическое обозначение мультиплексора

Мультиплексор можно реализовать с помощью ЛЭ заданного базиса. В его структуру можно ввести и более сложные цифровые устройства. Сопоставляя формулы (3.1) и (3.2), можно заметить, что для каждого входа D комбинации сигналов управления X_1, X_2, X_3 в мультиплексоре такие же, как и в дешифраторе. Следовательно, составной частью мультиплексора является дешифратор.

Структурная схема одного из возможных вариантов мультиплексора с четырьмя информационными входами $D_0 - D_3$ приведена на рис. 3.13.

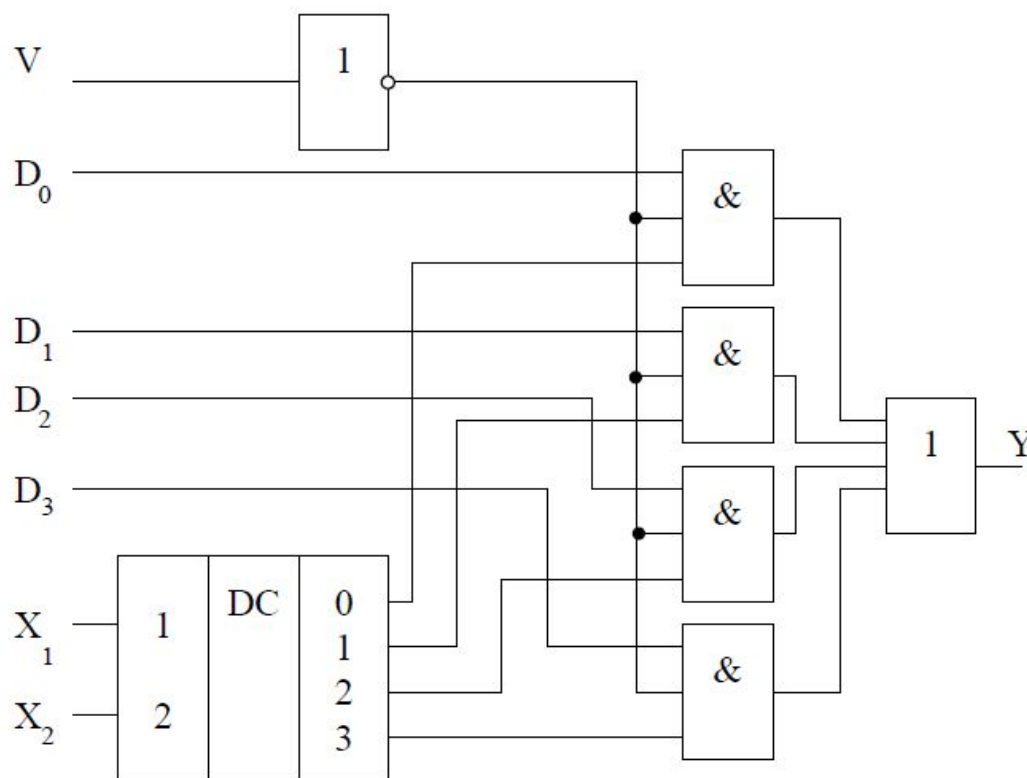


Рис. 3.13. Схема мультиплексора с четырьмя информационными входами

3.7. Демультимплексоры

Демультимплексоры представляют собой цифровые устройства для коммутации информационного входного сигнала в одном из нескольких выходов в соответствии с заданным кодом на управляющих входах. Иными словами, демультимплексоры решают задачи обратные мультиплексированию. Условное графическое обозначение демультимплексора с четырьмя информационными выходами представлено на рис. 3.14. Структурные формулы, определяющие формирование выходных сигналов такого демультимплексора, имеют вид:

$$Y_0 = \bar{X}_2 \bar{X}_1 D; Y_1 = \bar{X}_2 X_1 D; Y_2 = X_2 \bar{X}_1 D; Y_3 = X_2 X_1 D. \quad (3.3)$$

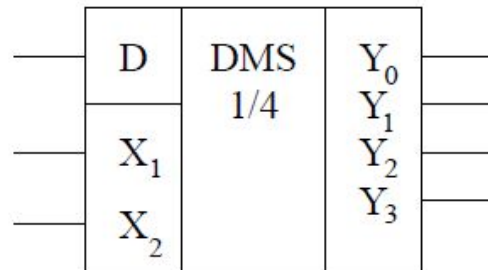


Рис. 3.14. Условное графическое обозначение демультимплексора

Точно такими же выражениями определяется формирование выходных сигналов дешифратора, использующего синхронизацию по входу D. Поэтому в качестве демультиплексора можно применять дешифратор, в котором информационный входной сигнал подан на вход синхронизации. Примером является реализация 2-разрядного демультиплексора с нулевыми активными сигналами на основе рассмотренного выше сдвоенного дешифратора К155ИД4 (см. рис. 3.6, б). Если первый разряд D_p подать на вход V_1 , а второй разряд D_p – на вход V_3 , то, в зависимости от комбинаций значений X_1, X_2 , можно обеспечить коммутацию этого 2-разрядного кода $D_{p1}D_{p2}$ на любой из четырех выходов Y . На остальные входы синхронизации V_2 и V_4 можно подать постоянные значения ($V_2 = 1, V_4 = 0$), либо использовать их для организации синхронной передачи информации на выходы (рис. 3.15).

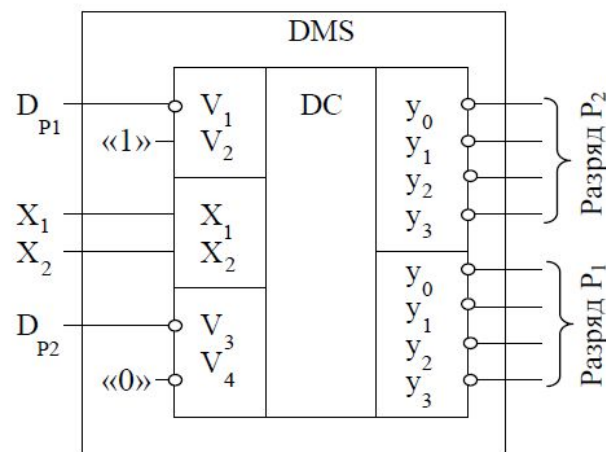


Рис. 3.15. Условное графическое обозначение демультиплексора на основе двух ИМС К155ИД4

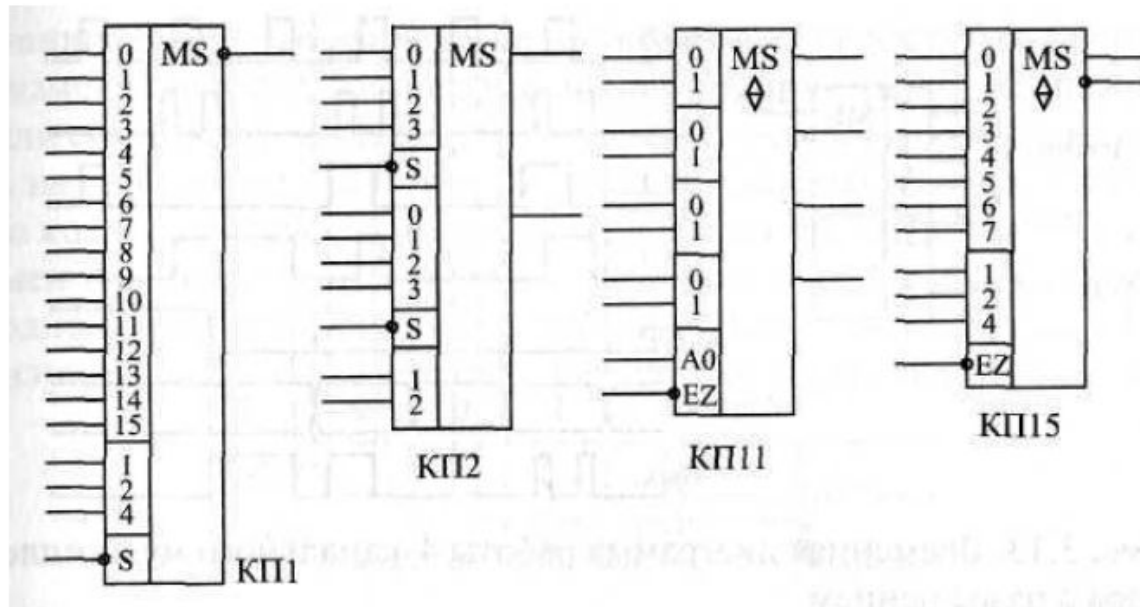


Рис. 3.12. Примеры микросхем мультиплексоров

Таблица 3.3. Таблица истинности 8-канального мультиплексора

Входы				Выходы	
4	2	1	-EZ	Q	-Q
X	X	X	1	Z	Z
0	0	0	0	D0	-D0
0	0	1	0	D1	-D1
0	1	0	0	D2	-D2
0	1	1	0	D3	-D3
1	0	0	0	D4	-D4
1	0	1	0	D5	-D5
1	1	0	0	D6	-D6
1	1	1	0	D7	-D7

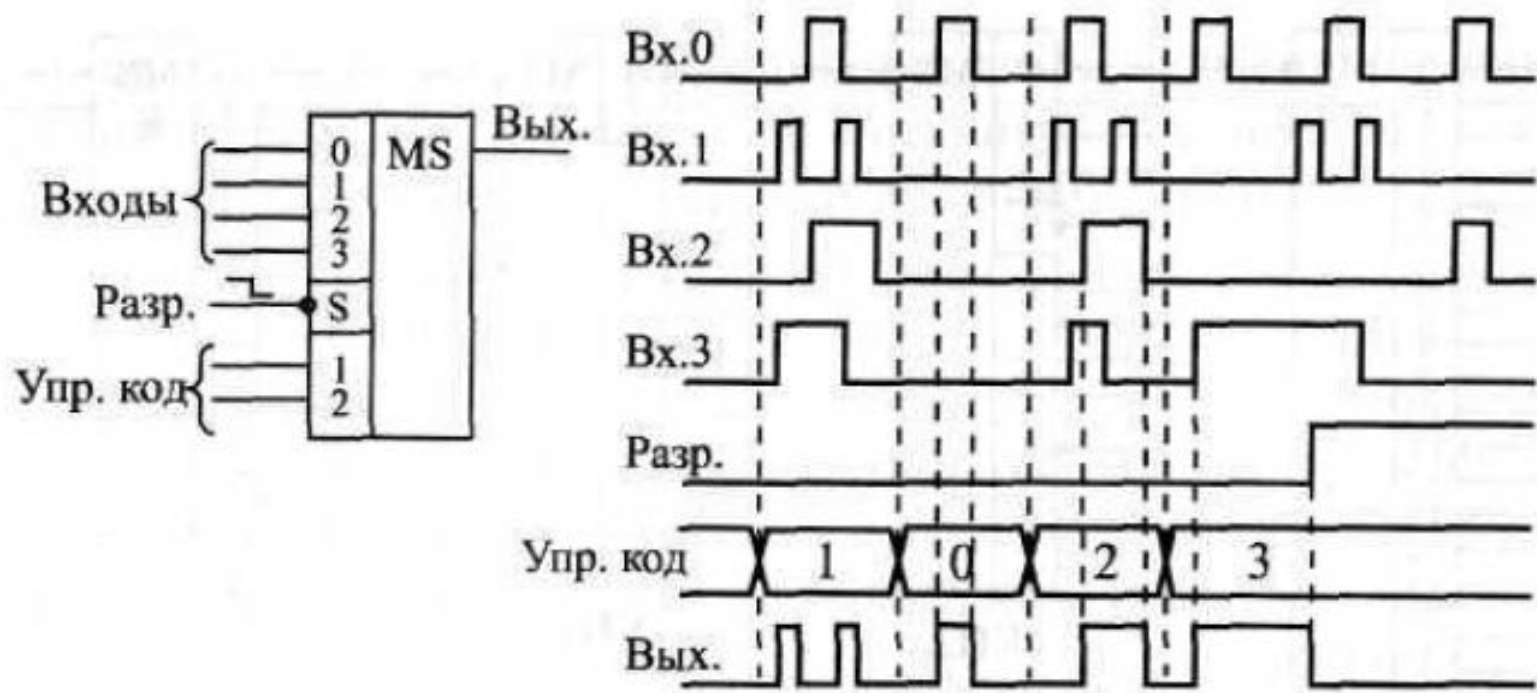


Рис. 3.13. Временная диаграмма работы 4-канального мультиплекса с разрешением

8.3. Разработка логического анализатора

Логический анализатор — это контрольно-измерительный прибор, предназначенный для запоминания (фиксации) и последующего анализа (например, просмотра на экране) временных диаграмм большого количества цифровых сигналов. Логические анализаторы используются при динамической отладке различных цифровых устройств и систем, а также при контроле их работы. Совершенно незаменимы они при разработке и отладке различных микропроцессорных систем, контроллеров, компьютеров, где используется большое количество многоразрядных шин цифровых сигналов. Именно логические анализаторы позволяют разработчику увидеть те временные диаграммы, которые он рисует на бумаге при проектировании своего устройства, причем увидеть их в реальном масштабе времени, посмотреть, как работает устройство на своей нормальной рабочей скорости.

Логический анализатор по своему назначению близок к осциллографу, так как он также позволяет наблюдать на экране временные диаграммы сигналов. Но существуют и существенные отличия логического анализатора от обычного (не цифрового) осциллографа:

- Логический анализатор работает только с цифровыми, то есть двухуровневыми (реже трехуровневыми) сигналами, а осциллограф — с аналоговыми сигналами, имеющими бесконечно большое число разрешенных уровней.
- Логический анализатор имеет большое количество входных линий (обычно от 16 до 64), то есть позволяет одновременно фиксировать

множество входных сигналов, а осциллографы обычно позволяют одновременно увидеть не более четырех входных сигналов.

- Логический анализатор работает в режиме однократного запоминания временных диаграмм (как запоминающий осциллограф). То есть анализатор запоминает состояния входных сигналов в течение заданного времени (называемого окном регистрации), а затем дает возможность анализировать зафиксированные последовательности. Осциллограф же работает обычно в режиме непрерывной развертки, то есть он не запоминает формы входного сигнала и позволяет наблюдать только повторяющиеся, периодические сигналы.
- Логический анализатор предусматривает возможность так называемой предпусковой регистрации. Эта возможность предусматривается и в цифровых осциллографах, но ее нет в аналоговых осциллографах. Рассмотрим подробнее, что такое предпусковая регистрация.

Процесс регистрации входных сигналов (или отображения их на экране в обычном осциллографе) всегда должен быть привязан к какому-то моменту времени, к какому-то внешнему событию, называемому запуском. Иначе разобраться в отображаемых сигналах будет совершенно невозможно. Например, в осциллографах моментом запуска обычно является момент превышения входным исследуемым сигналом установленного порога. Сигналом запуска может служить и специальный внешний синхронизирующий сигнал. В логических анализаторах в качестве запуска обычно используется момент появления на входах заданного уровня или заданной последовательности одного или нескольких входных сигналов.

В обычных осциллографах отображение формы входного сигнала (или входных сигналов) начинается в момент запуска, то есть на экране видно только то, что происходило со входными сигналами *после* момента запуска. Такая регистрация может быть названа послепусковой. Можно также сказать, что точка запуска всегда находится в начале окна регистрации (рис. 8.9).

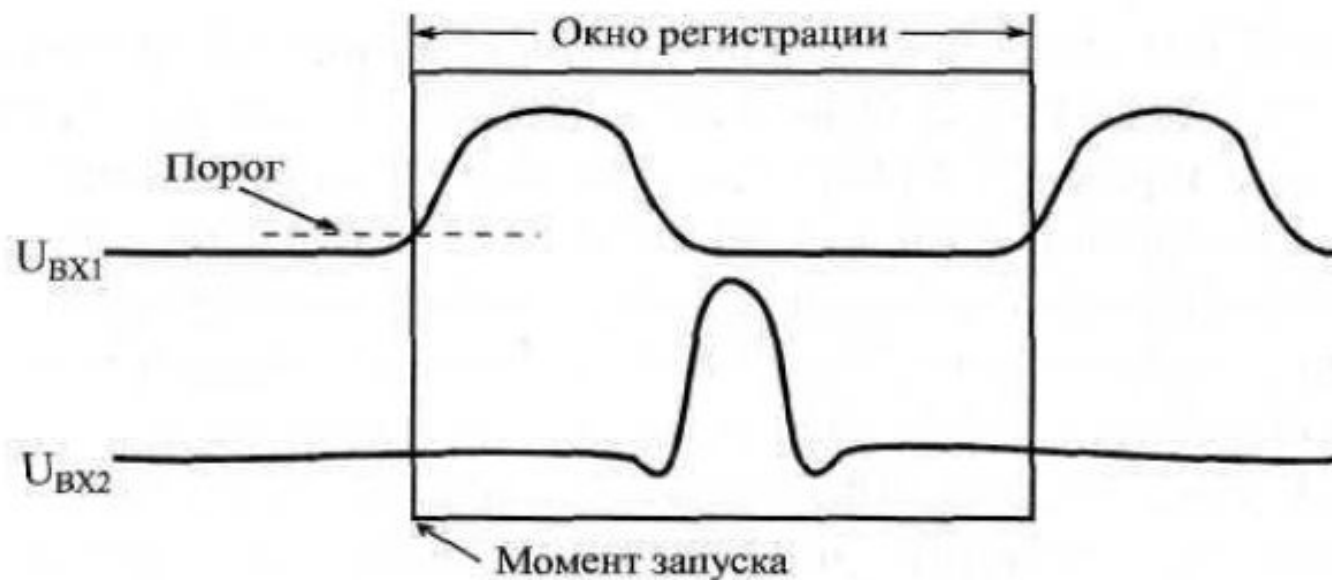


Рис. 8.9. Послепусковая регистрация в аналоговых осциллографах

В логических анализаторах (и в цифровых осциллографах) существует возможность увидеть и зафиксировать не только то, что было *после* запуска, но еще и то, что происходило в течение определенного времени *до* момента запуска. Именно эта регистрация до момента запуска и называется предпусковой регистрацией. В этом случае точка запуска может находиться и в начале, и в середине, и в конце окна регистрации (рис. 8.10). Понятно, что такая возможность очень удобна, так как, выбирая величину длительности предпусковой регистрации, можно увидеть те события, временная привязка к началу которых затруднена или попросту невозможна. Длительность (глубина) предпусковой регистрации может быть постоянной (например, равной половине длительности окна регистрации) или переменной (то есть задаваться пользователем в пределах от нуля до полной длительности окна регистрации). При переменной глубине предпусковой регистрации точка запуска может располагаться в любой точке окна регистрации — от его начала до конца.

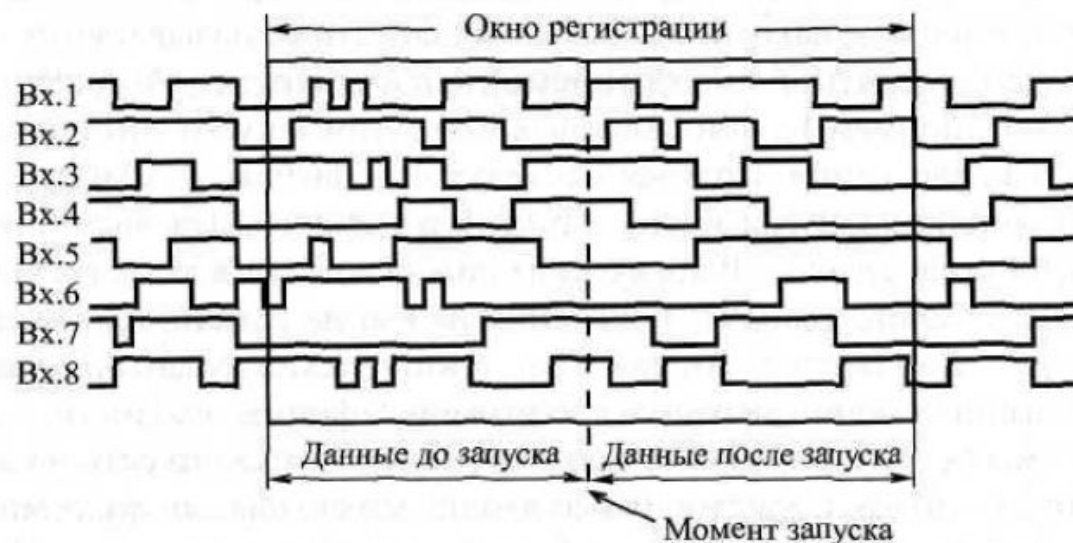


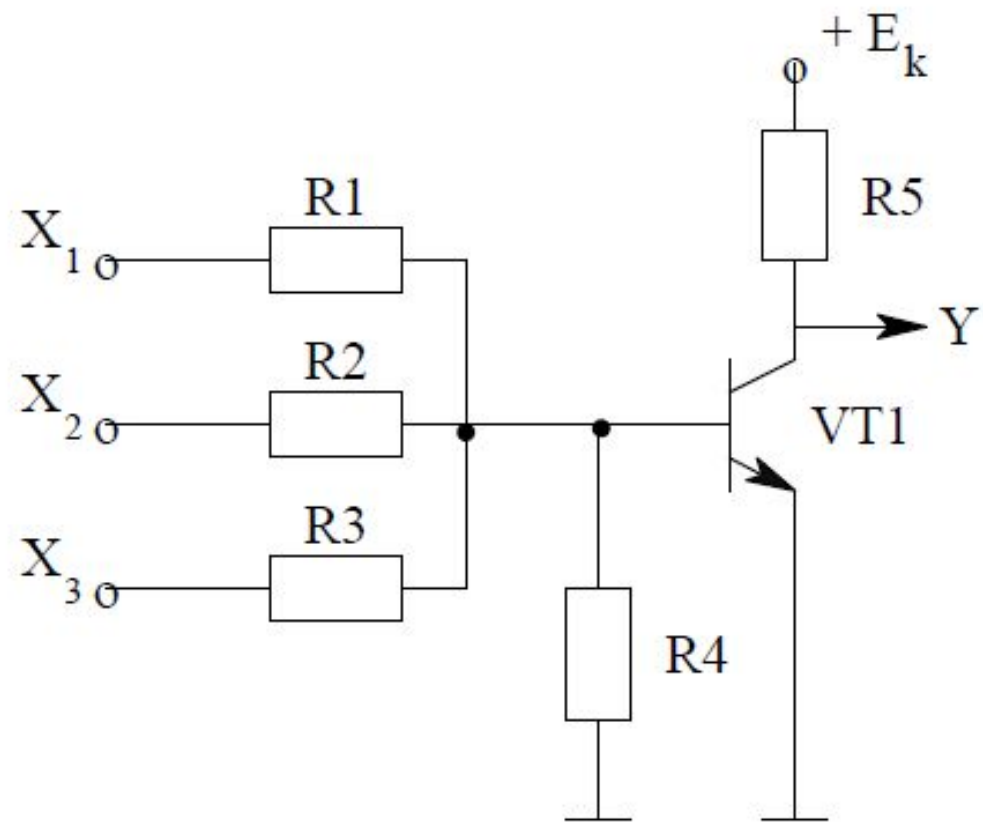
Рис. 8.10. Предпусковая регистрация в логических анализаторах и цифровых осциллографах

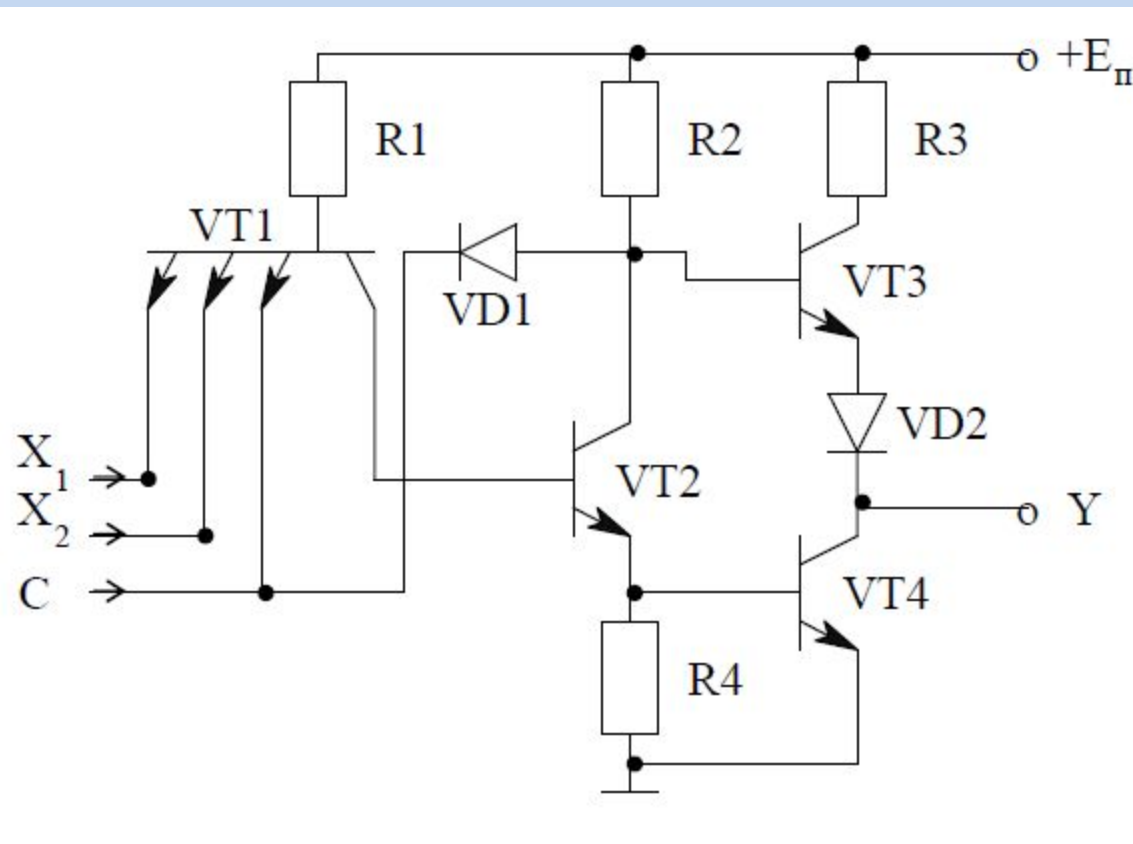
С точки зрения схемотехники, логический анализатор представляет собой быстродействующую буферную оперативную память, работающую в периодическом режиме. Буфер этот однонаправленный: сначала в буферную память с большой тактовой частотой последовательно записываются состояния нескольких входных сигналов, а затем эта информация последовательно читается из буфера. Таким образом, адреса буферной памяти могут перебираться одним и тем же счетчиком как в режиме записи, так и в режиме чтения. Структура таких буферов уже рассматривалась в разделе 6.2.2.

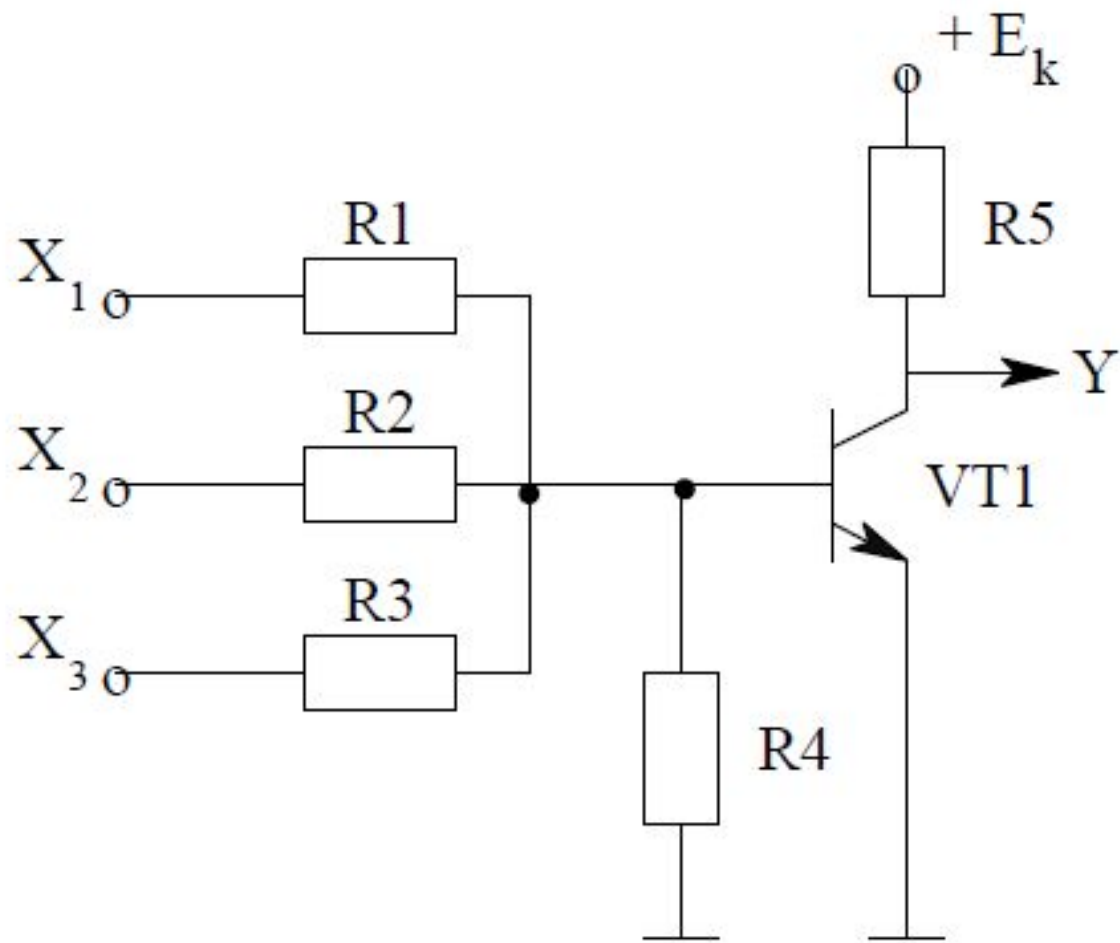
Главные особенности логического анализатора, по сравнению со стандартной структурой информационного буфера на основе оперативной памяти, следующие:

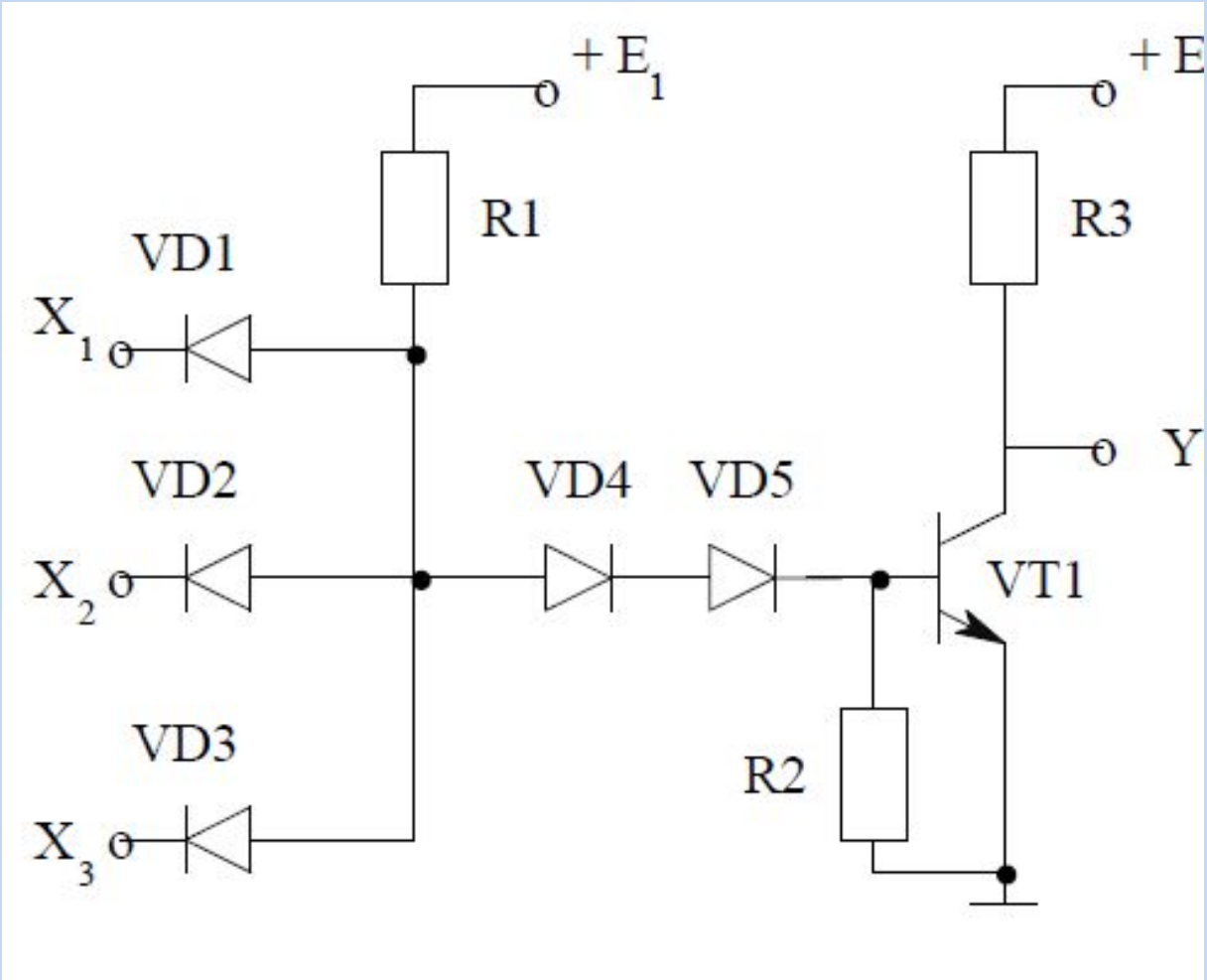
- большое число разрядов шины данных (то есть входных сигналов, каналов регистрации анализатора);
- необходимость обеспечения режима предпусковой регистрации;
- необходимость временной привязки процесса регистрации (записи в память) к состояниям входных сигналов (обеспечение запуска).

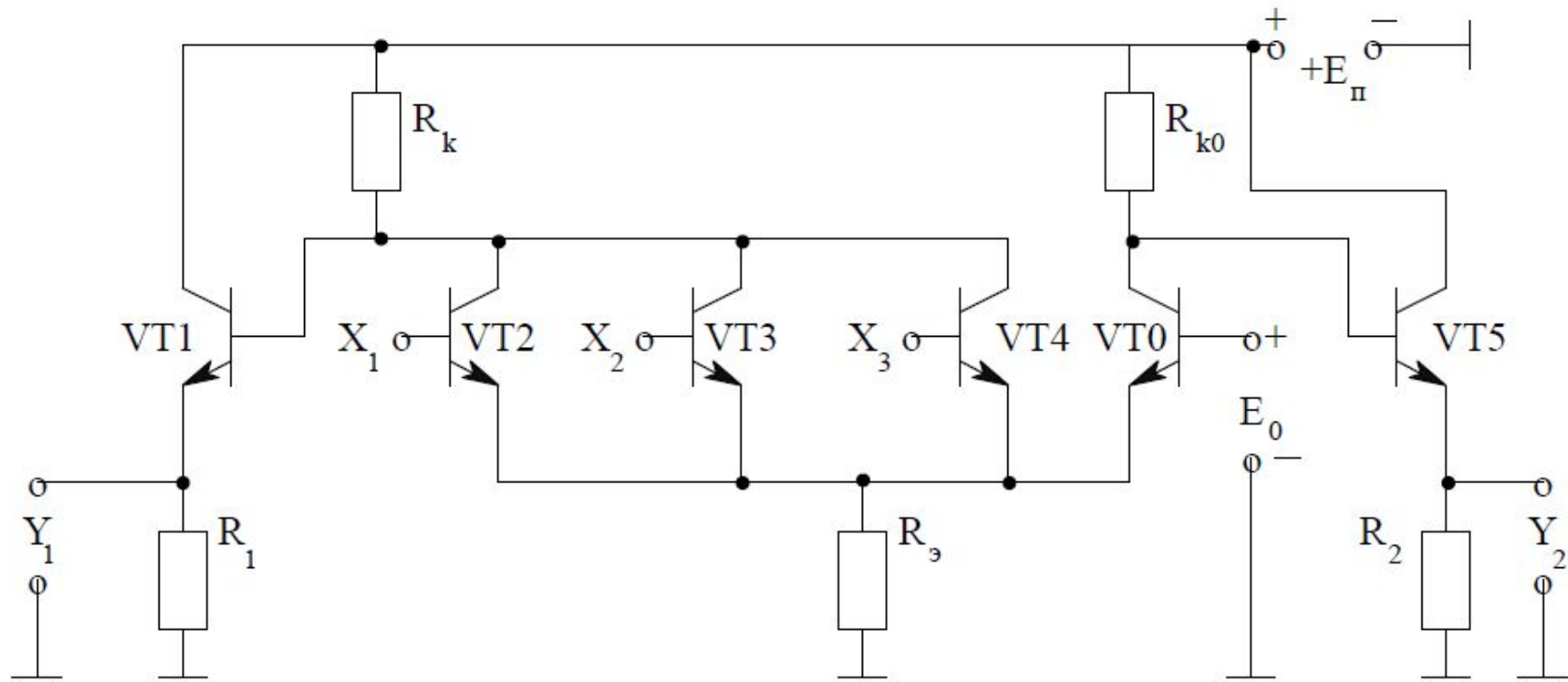
- Скоро большая проверочная работа. поэтому необходимо повторить дома :
- 1) перевод чисел из одной системы счисления в другую
- 2) повторить логические элементы (НЕ, ИЛИ, И , ИСКЛ ИЛИ, ...)
- 3) Принципиальные схемы.
- 4) Логические схемы
- 5) Обозначения цифровых устройств











3.6. Одновибраторы и генераторы

Одновибраторы и генераторы вообще-то нельзя отнести к комбинационным микросхемам. Они занимают промежуточное положение между комбинационными микросхемами и микросхемами с внутренней памятью. Их выходные сигналы не определяются однозначно входными сигналами, как у комбинационных микросхем. Но в то же время они и не хранят информацию длительное время.

Одновибраторы ("ждущие мультивибраторы", английское название "Monostable Multivibrator") представляют собой микросхемы, которые в ответ на входной сигнал (логический уровень или фронт) формируют выходной импульс заданной длительности. Длительность определяется внешними времязадающими резисторами и конденсаторами. То есть можно считать, что у одновибраторов есть внутренняя память, но эта память хранит информацию о входном сигнале строго заданное время, а потом информация исчезает. На схемах одновибраторы обозначаются буквами G1.

В стандартные серии микросхем входят одновибраторы двух основных типов (отечественное обозначение функции микросхемы — А Г):

- Одновибраторы без перезапуска (АГ1 — одиночный одновибратор, АГ4 — два одновибратора в корпусе).
- Одновибраторы с перезапуском (АГ3 — два одновибратора в корпусе).

Разница между этими двумя типами иллюстрируется рис. 3.30. Одновибратор без перезапуска не реагирует на входной сигнал до окончания своего выходного импульса. Одновибратор с перезапуском начинает отсчет нового времени выдержки T с каждым новым входным сигналом независимо от того, закончилось ли предыдущее время выдержки. В случае, когда период следования входных сигналов меньше времени выдержки T , выходной импульс одновибратора с перезапуском не прерывается. Если период следования входных запускающих импульсов больше време-

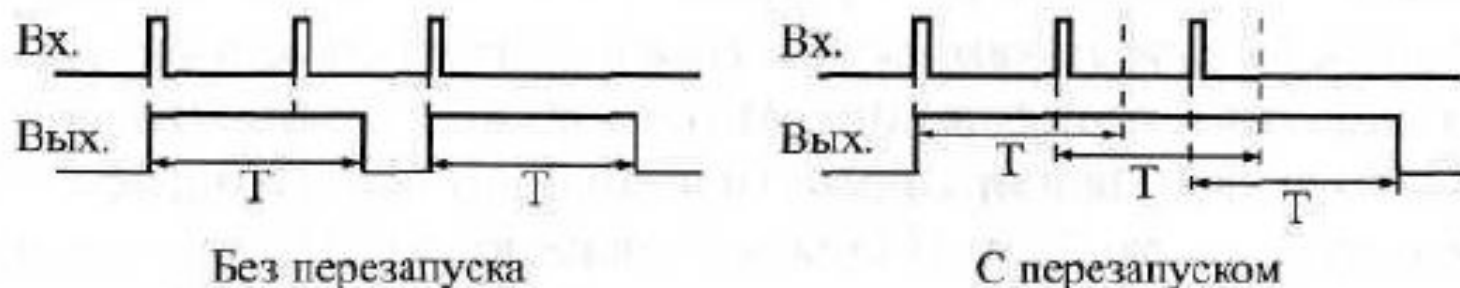


Рис. 3.30. Принцип работы одновибраторов без перезапуска и с перезапуском

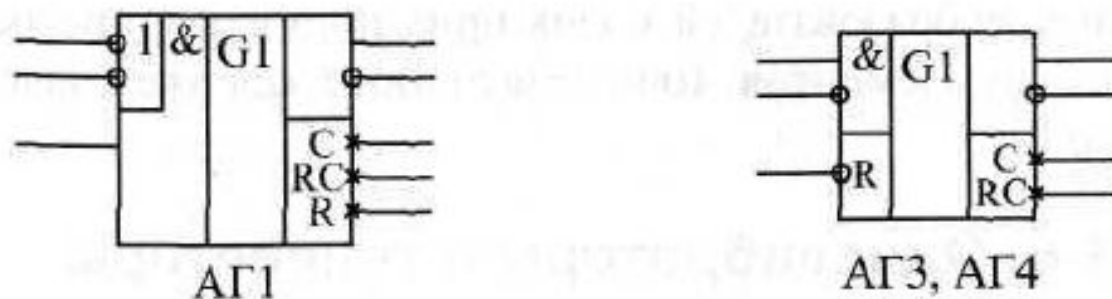


Рис. 3.31. Микросхемы одновибраторов

ни выдержки одновибратора Т, то оба типа одновибраторов работают одинаково.

На рис. 3.31 приведены обозначения микросхем одновибраторов стандартных серий. Микросхемы АГ3 и АГ4 отличаются друг от друга только тем, что АГ3 работает с перезапуском, а АГ4 — без перезапуска.

Микросхемы имеют входы запуска, объединенные по И и ИЛИ, прямые и инверсные выходы, а также выводы для подключения внешних времязадающих цепей (резисторов и конденсаторов). Запускается работа всех одновибраторов по фронту результирующего входного сигнала. Используемая логика объединения входов микросхем позволяет запустить все одновибраторы как по положительному, так и по отрицательному фронту входного сигнала (рис. 3.32 и 3.33).

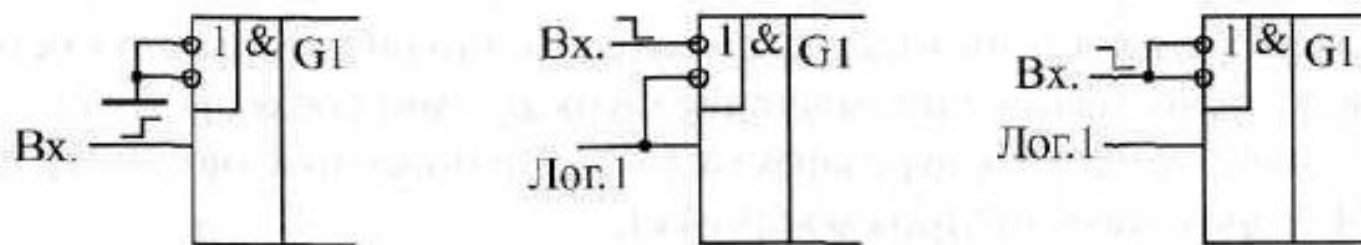


Рис. 3.32. Варианты запуска одновибратора АГ1

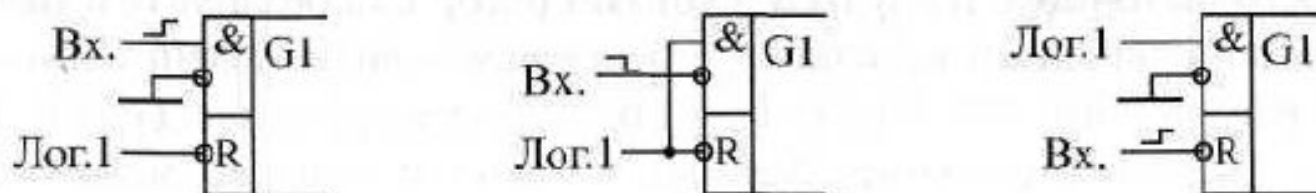


Рис. 3.33. Варианты запуска одновибраторов АГ3 и АГ4

На неиспользуемые входы при этом надо подавать сигналы логического нуля или логической единицы. Можно также использовать остающиеся входы для разрешения или запрещения входного запускающего сигнала.

Одновибраторы АГ3 и АГ4 имеют также дополнительный вход сброса -R, логический ноль на котором не только запрещает выработку выходного сигнала, но и прекращает его. Вход -R можно также использовать для запуска одновибратора.

Таблица 3.8. Таблица истинности одновибратора АГ1

Входы			Выходы	
-A1	-A2	B	Q	-Q
0	X	1	0	1
X	0	1	0	1
X	X	0	0	1
1	1	X	0	1
1	┌	1	J L	┐
┌	1	1	J L	┐┐
┌	┌	1	J L	┐┐
0	X	┐	J L	┐┐
X	0	┐	┐┐	┐┐

Таблица 3.9. Таблица истинности одновибраторов АГ3 и АГ4

Входы			Выходы	
-R	-A	B	Q	-Q
0	X	X	0	1
X	1	0	0	1
X	X	0	0	1
1	0	┐	J L	┐
1	┌	1	┐┐	┐┐
┐	0	1	J L	┐┐

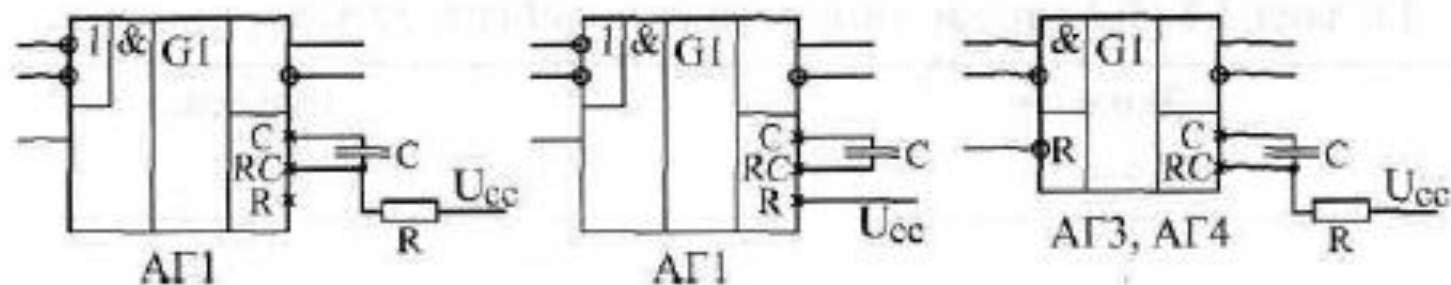


Рис. 3.34. Стандартные схемы включения одновибраторов

чая вывод R к напряжению питания. Повторный запуск одновибратора невозможен сразу после окончания выходного импульса, до повторного запуска обязательно должен пройти интервал $t = C$ (если емкость измеряется в нанофарадах, то временной интервал получается в микросекундах).

Для одновибраторов АГ3 и АГ4 длительность импульса можно оценить по формуле: $T = 0,32C (R + 0,7)$, где сопротивление резистора измеряется в килоОмах. Сопротивление резистора может находиться в пределах от 5,1 кОм до 51 кОм, емкость конденсатора — любая. Перезапуск одновибратора возможен только в том случае, когда интервал между входными запускающими импульсами больше $0,224C$ (если емкость измеряется в нанофарадах, то временной интервал — в микросекундах).

Наиболее распространенные применения одновибраторов следующие (рис. 3.35):

- а) увеличение длительности входного импульса;
- б) уменьшение длительности входного импульса;
- в) деление частоты входного сигнала в заданное число раз;
- г) формирование сигнала огибающей последовательности входных импульсов.

Для увеличения или уменьшения длительности входного сигнала (а и б) надо всего лишь выбрать сопротивление резистора и емкость конденсатора, исходя из требуемой длительности выходного сигнала. В этом

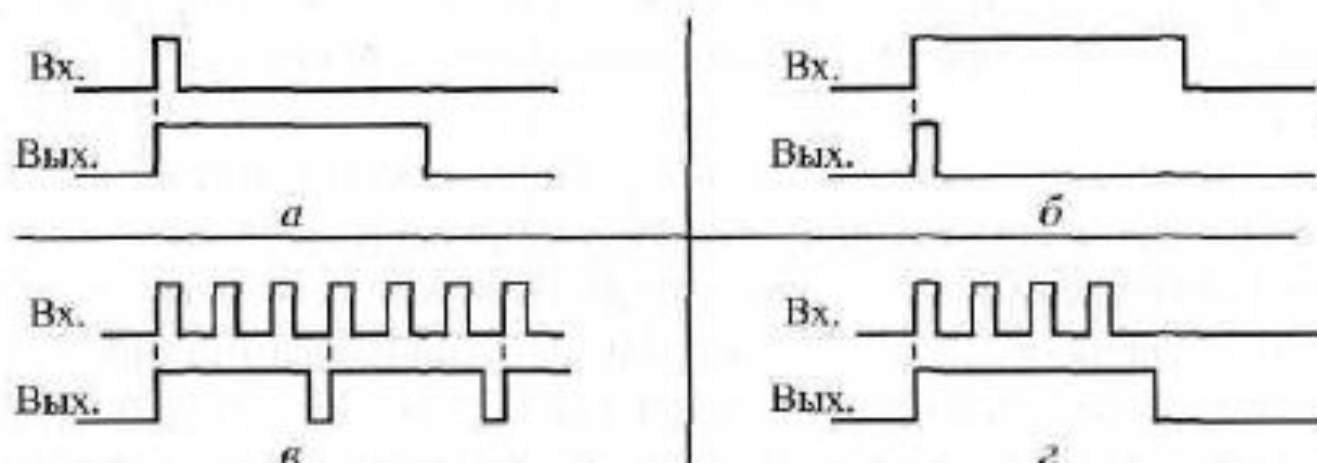


Рис. 3.35. Стандартные применения одновибраторов

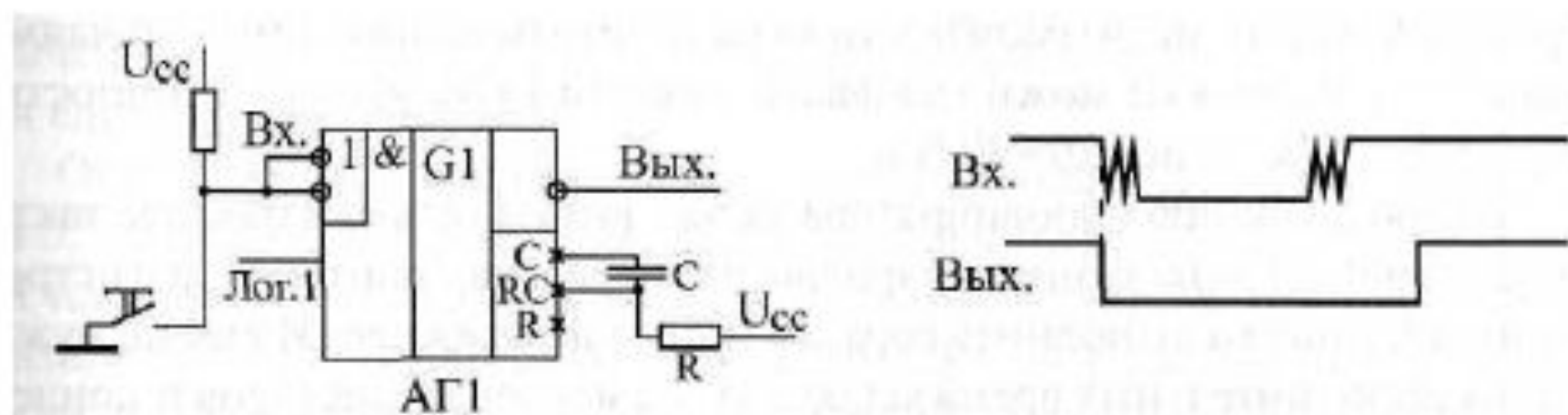


Рис. 3.36. Использование одновибратора для подавления дребезга контактов кнопки

Для этого можно использовать как одновибратор с перезапуском, так и одновибратор без перезапуска (на рисунке). Можно также подобрать время выдержки так, что одновибратор будет давать один импульс по нажатию кнопки, а другой импульс — по отпусканию кнопки. Иногда это бывает удобнее.

6.1.1. ПЗУ как универсальная комбинационная микросхема

Одно из самых распространенных применений микросхем ПЗУ — замена ими сложных комбинационных схем. Такое решение позволяет существенно упростить проектируемое устройство и снизить количество используемых комбинационных микросхем, а иногда даже уменьшить потребляемый ток и увеличить быстродействие схемы.

Суть предлагаемого подхода сводится к следующему. Если рассматривать адресные входы микросхемы ПЗУ как входы комбинационной схемы, а разряды данных — как выходы этой комбинационной схемы, то можно сформировать *любую* требуемую таблицу истинности данной комбинационной схемы. Для этого всего лишь надо составить таблицу прошивки ПЗУ, соответствующую нужной таблице истинности. В этом случае не надо ни подбирать логические элементы, ни оптимизировать их соединения, ни думать о том, можно ли вообще построить заданную комбинационную схему из стандартных микросхем. Важно только, чтобы количество требуемых входов не превышало количества адресных разрядов ПЗУ, а количество требуемых выходов не превышало разрядности шины данных ПЗУ.

В качестве примера рассмотрим довольно сложную комбинационную схему (рис. 6.5), имеющую восемь входов и четыре выхода. Функция схемы сводится к следующему. Прежде всего она распознает два различных 5-разрядных входных кода (11001 и 10011) в случае, когда на входе разрешения "-Разр." присутствует нулевой сигнал, а при приходе сигналов "-Строб 1" и "Строб 2" схема выдает на выход отрицательные импуль-

сы. Причем первый выходной сигнал вырабатывается в случае, когда входной код равен 11001 и пришел сигнал "-Строб 1", второй выходной сигнал — при том же коде, но по входному сигналу "-Строб 2". Третий и четвертый выходной сигналы вырабатываются при входном коде 10011 и при приходе соответственно управляющих сигналов "-Строб 1" и "-Строб 2". То есть логика работы довольно сложная и разнообразных логических элементов требуется немало.

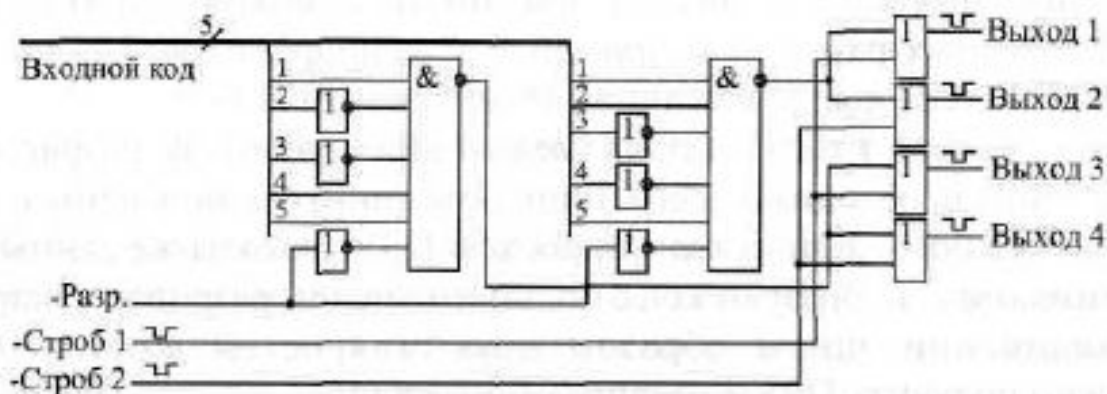


Рис. 6.5. Пример комбинационной схемы, заменяемой ПЗУ

Но всю эту схему можно заменить всего лишь одной микросхемой ПЗУ, например, типа РТ4, имеющей 8 адресных входов и 4 выхода данных (рис. 6.6). При этом пять разрядов входного кода подаются на младшие разряды адреса ПЗУ (А0... А4), входной сигнал "Разр." — на адресный вход А5, сигнал "-Строб 1" - на вход А6, сигнал "-Строб 2" - на вход А7. Младший разряд данных памяти D0 используется для первого выходного сигнала, D1 — для второго выходного сигнала, D2 — для третьего выходного сигнала, D3 — для четвертого выходного сигнала. Микросхема ПЗУ всегда выбрана (управляющие сигналы -CS1 и -CS2 - нулевые). На выходах данных памяти включены резисторы, так как тип выходов микросхемы РТ4 - ОК

Простейший пример данного применения ПЗУ — это управление знаковым семисегментным индикатором, знакомым всем по калькуляторам, кассовым аппаратам, электронным часам, весам и т. д. В семисегментных индикаторах изображение всех цифр от 0 до 9 строится всего из семи сегментов (отрезков линий) (рис. 6.9).

Чтобы отобразить в виде цифры 4-разрядный двоичный код, надо этот код преобразовать в 7-разрядный код, каждому разряду которого будет соответствовать один сегмент индикатора. То есть коду 0000 должно соответствовать изображение нуля (6 сегментов, расположенных по периметру), а коду 0001 — изображение единицы (два правых вертикальных сег-

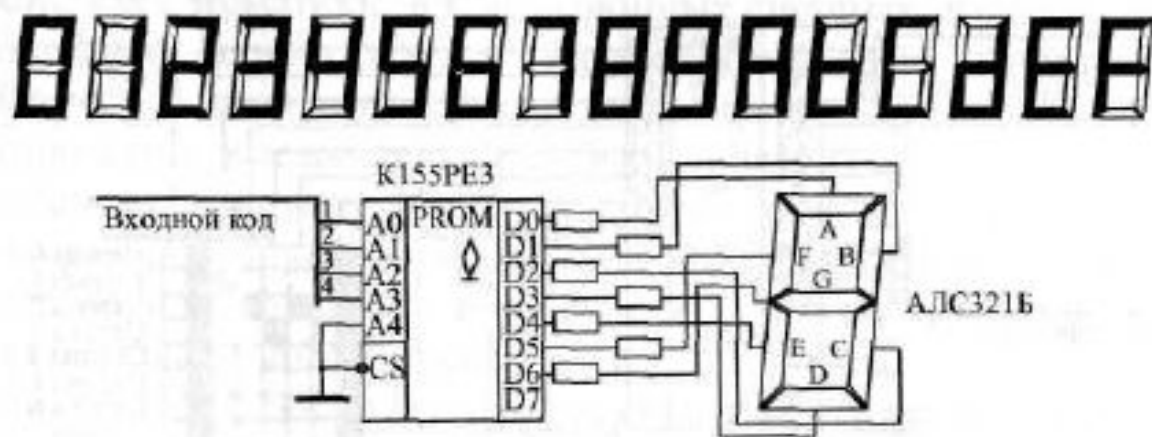


Рис. 6.9. Дешифратор знакового семисегментного индикатора на ПЗУ

мента). Для повышения универсальности индикатора удобно дополнить десять цифр еще и шестью буквами, используемыми в 16-ричном коде (А, В, С, D, Е, F). Семь сегментов индикатора позволяют сделать и это, правда, изображения букв получаются не слишком качественными.

ПЗУ позволяют также формировать и более сложные изображения букв и цифр — матричные. Такие изображения используются, например, в табло типа "бегущая строка", на экранах мониторов, в больших рекламных табло. Каждая буква, цифра, другой знак располагается в данном случае на прямоугольной матрице, называемой знакоместом и состоящей из нескольких строк и нескольких столбцов точечных элементов изображения, которые могут зажигаться независимо друг от друга. Чем больше строк и столбцов в знакоместе, тем более качественное изображение букв и цифр можно получить. Минимально возможный размер знакоместа — 5 столбцов на 7 строк, то есть всего 35 элементов изображения.

ПЗУ в данном случае содержит в себе информацию об изображениях всех возможных букв и цифр (обычно этот набор включает в себя 256 символов). Но выходной код ПЗУ имеет мало разрядов, поэтому

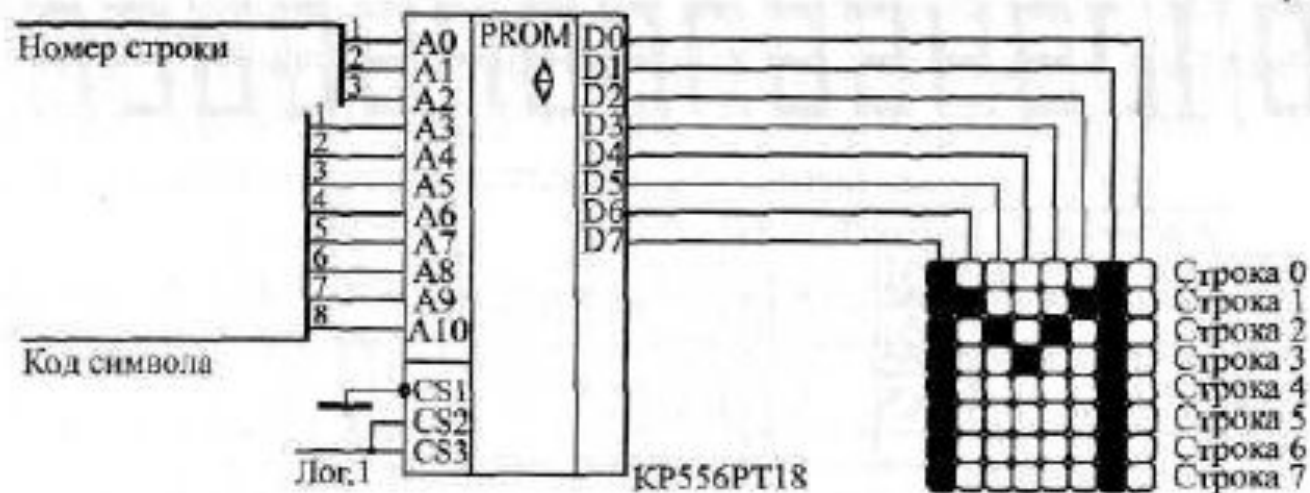


Рис. 6.10. Матричный знакогенератор на ПЗУ

каждый такой код, соответствующий одному адресу, представляет собой информацию об изображении не целого символа, а только одной его строки (или столбца). Информация о целом символе занимает в ПЗУ столько ячеек, сколько в изображении символа имеется столбцов (или строк). Пример матричного знакогенератора на ПЗУ приведен на рис. 6.10.

В данном случае используется знакоместная матрица из 8 строк и 8 столбцов. В каждую ячейку ПЗУ записывается код изображения одной из 8 строк одного из 256 символов. Изображение одного символа занимает 8 последовательно расположенных ячеек в ПЗУ. Для букв и цифр правый столбец знакоместа не используется, он служит для отделения знаков друг от друга, но для специальных символов (например, для графических) он может и использоваться. В случае матричного светодиодного индикатора перебор строк может осуществляться 3-разрядным счетчиком с дешифратором 3-8 на его выходе. В случае телевизионного монитора перебор строк осуществляется с помощью генератора вертикальной развертки изображения.

Составление карты прошивки такого ПЗУ непросто, оно обычно производится с помощью специальных программ на компьютере. Но принцип составления прост. Например, если активному (зажженному) элементу изображения соответствует единичный сигнал, то для нулевой строки символа "М", показанного на рисунке, в ПЗУ надо записать 10000010, для первой строки — код 11000110, для второй — код 10101010, для третьей - 10010010 и т. д.

Лекция 13. Применение ЦАП и АЦП

Как уже отмечалось в первой главе, цифро-аналоговые преобразователи (ЦАП, DAC — “Digital-to-Analog Converter”) и аналого-цифровые преобразователи (АЦП, ADC — “Analog-to-Digital Converter”) главным образом применяются для сопряжения цифровых устройств и систем с внешними аналоговыми сигналами, с реальным миром. При этом АЦП преобразует аналоговые сигналы во входные цифровые сигналы, поступающие на цифровые устройства для дальнейшей обработки или хранения, а ЦАП преобразует выходные цифровые сигналы цифровых устройств в аналоговые сигналы (см. рис. 1.23).

ЦАП и АЦП применяются в измерительной технике (цифровые осциллографы, вольтметры, генераторы сигналов и т. д.), в бытовой аппаратуре (телевизоры, музыкальные центры, автомобильная электроника и т. д.), в компьютерной технике (ввод и вывод звука в компьютерах, видеомониторы, принтеры и т. д.), в медицинской технике, в радиолокационных устройствах, в телефонии и во многих других областях. Применение ЦАП и АЦП постоянно расширяется по мере перехода от аналоговых к цифровым устройствам.

В качестве ЦАП и АЦП обычно применяются специализированные микросхемы, выпускаемые многими отечественными и зарубежными фирмами.

Сразу же надо отметить, что для грамотного и профессионального использования микросхем ЦАП и АЦП совершенно не достаточно знания цифровой схемотехники. Эти микросхемы относятся к аналого-цифровым, поэтому они требуют также знания аналоговой схемотехники, су-

7.1. Применение ЦАП

В общем случае микросхему ЦАП можно представить в виде блока (рис. 7.1), имеющего несколько цифровых входов и один аналоговый вход, а также аналоговый выход.

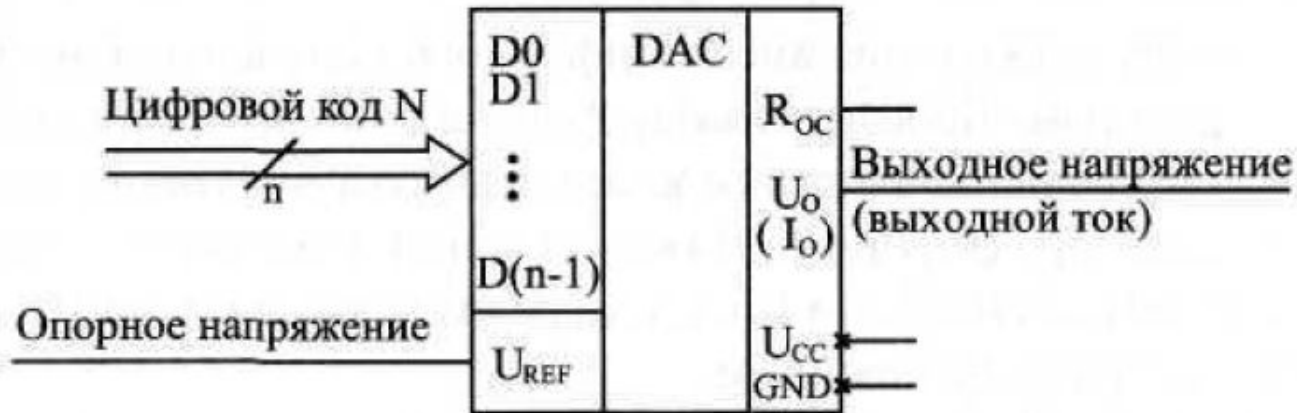


Рис. 7.1. Микросхема ЦАП

На цифровые входы ЦАП подается n -разрядный код N , на аналоговый вход — опорное напряжение $U_{0П}$ (другое распространенное обозначение — U_{REF}). Выходным сигналом является напряжение $U_{ВЫХ}$ (другое обозначение — U_0) ИЛИ ТОК $I_{ВЫХ}$ (другое обозначение — I_0). При этом выходной ток или выходное напряжение пропорциональны входному коду и опорному напряжению. Для некоторых микросхем опорное напряжение должно иметь строго заданный уровень, для других допускается менять его значение в широких пределах, в том числе и изменять его полярность (положительную на отрицательную и наоборот). ЦАП с большим диапазоном изменения опорного напряжения называется умножающим ЦАП, так как его можно легко использовать для умножения входного кода на любое опорное напряжение.

Кроме информационных сигналов, микросхемы ЦАП требуют также подключения одного или двух источников питания и общего провода. Обычно цифровые входы ЦАП обеспечивают совместимость со стандартными выходами микросхем ТТЛ.

Чаще всего в случае, если ЦАП имеет токовый выход, его выходной ток преобразуется в выходное напряжение с помощью внешнего операционного усилителя и встроенного в ЦАП резистора $R_{0С}$, ОДИН ИЗ ВЫВОДОВ которого выведен на внешний вывод микросхемы (рис. 7.2). Поэтому, если не оговорено иное, мы будем в дальнейшем считать, что выходной сигнал ЦАП — напряжение U_0 .

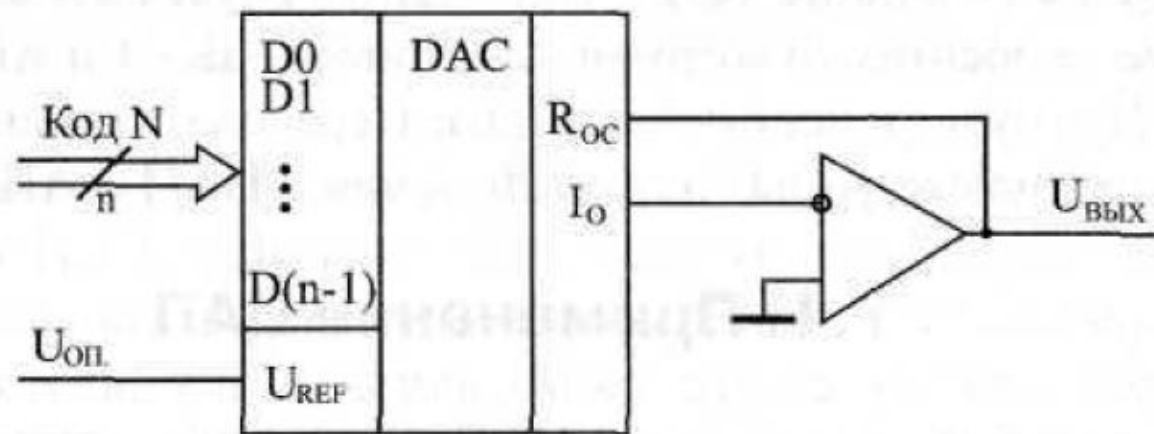


Рис. 7.2. Преобразование выходного тока ЦАП в выходное напряжение

Суть преобразования входного цифрового кода в выходной аналоговый сигнал довольно проста. Она состоит в суммировании нескольких токов (по числу разрядов входного кода), каждый последующий из которых вдвое больше предыдущего. Для получения этих токов используются или транзисторные источники тока, или резистивные матрицы, коммутируемые транзисторными ключами.

В качестве примера на рис. 7.3 показано 4-разрядное ($n = 4$) цифро-аналоговое преобразование на основе резистивной матрицы $R-2R$ и ключей (в реальности используются ключи на основе транзисторов). Правому положению ключа соответствует единица в данном разряде входного кода N (разряды $D_0 \dots D_3$). Операционный усилитель может быть как встро-

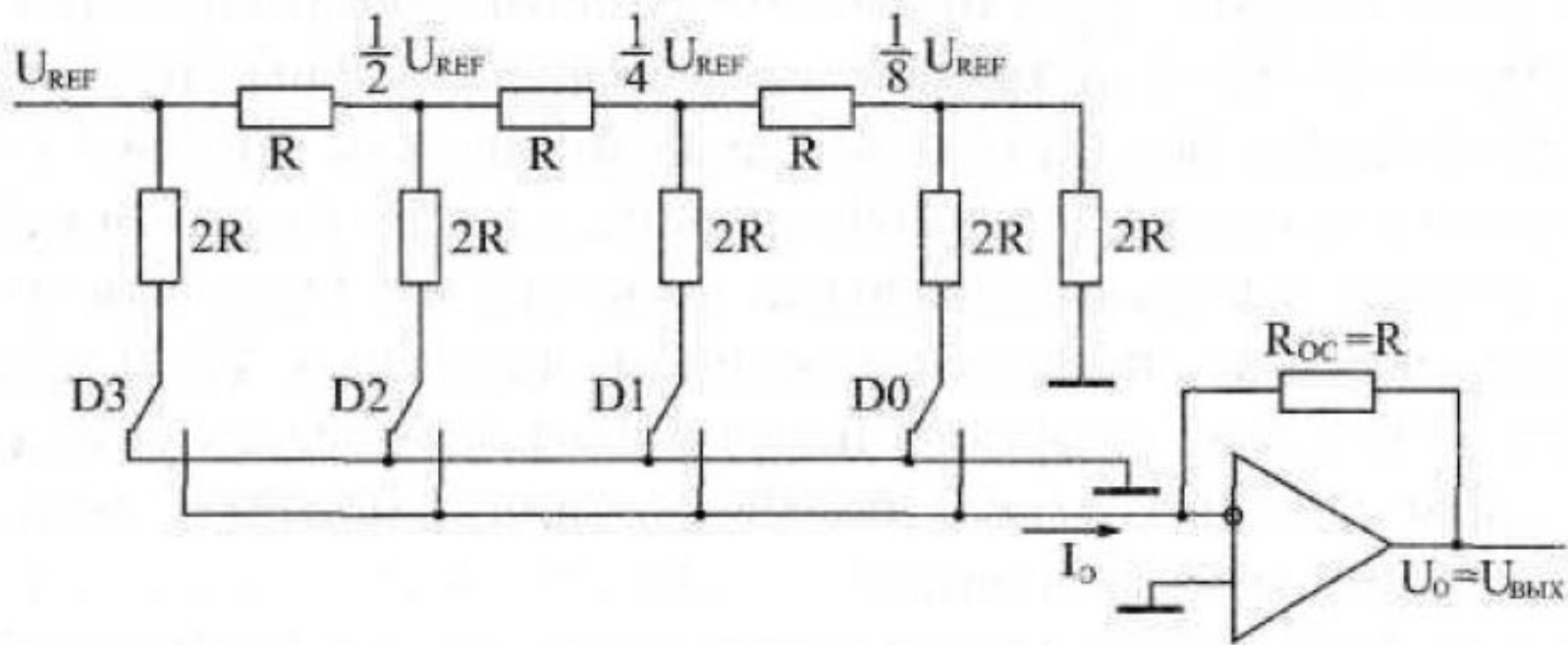


Рис. 7.3. 4-разрядное цифро-аналоговое преобразование

енным (в случае ЦАП с выходом по напряжению), так и внешним (в случае ЦАП с выходом по току).

Первым (левым по рисунку) ключом коммутируется ток величиной $U_{REF}/2R$, вторым ключом — ток $U_{REF}/4R$, третьим — ток $U_{REF}/8R$, четвертым — ток $U_{REF}/16R$. То есть токи, коммутируемые соседними ключами, различаются вдвое, как и веса разрядов двоичного кода. Токи, коммутируемые всеми ключами, суммируются и преобразуются в выходное напряжение с помощью операционного усилителя с сопротивлением $R_{OC} = R$ в цепи отрицательной обратной связи.

При правом положении каждого ключа (единица в соответствующем разряде входного кода ЦАП) ток, коммутируемый этим ключом, поступает на суммирование. При левом положении ключа (ноль в соответствующем разряде входного кода ЦАП) ток, коммутируемый этим ключом, на суммирование не поступает.

Суммарный ток I_0 от всех ключей создает на выходе операционного усилителя напряжение $U_0 = I_0 R_{OC} = I_0 R$. То есть вклад первого ключа (старшего разряда кода) в выходное напряжение составляет $U_{REF}/2$, второго — $U_{REF}/4$, третьего — $U_{REF}/8$, четвертого — $U_{REF}/16$. Таким образом, при входном коде $N = 0000$ выходное напряжение схемы будет нулевым, а при входном коде $N = 1111$ оно будет равно — $15U_{REF}/16$.

Для увеличения функциональных возможностей устройств часто предусматривается использование нескольких сигналов управления. На рис. 3.6, б представлено графическое изображение микросхемы К155ИД4, содержащей двойной двухвходовой дешифратор с активными нулевыми выходными сигналами.

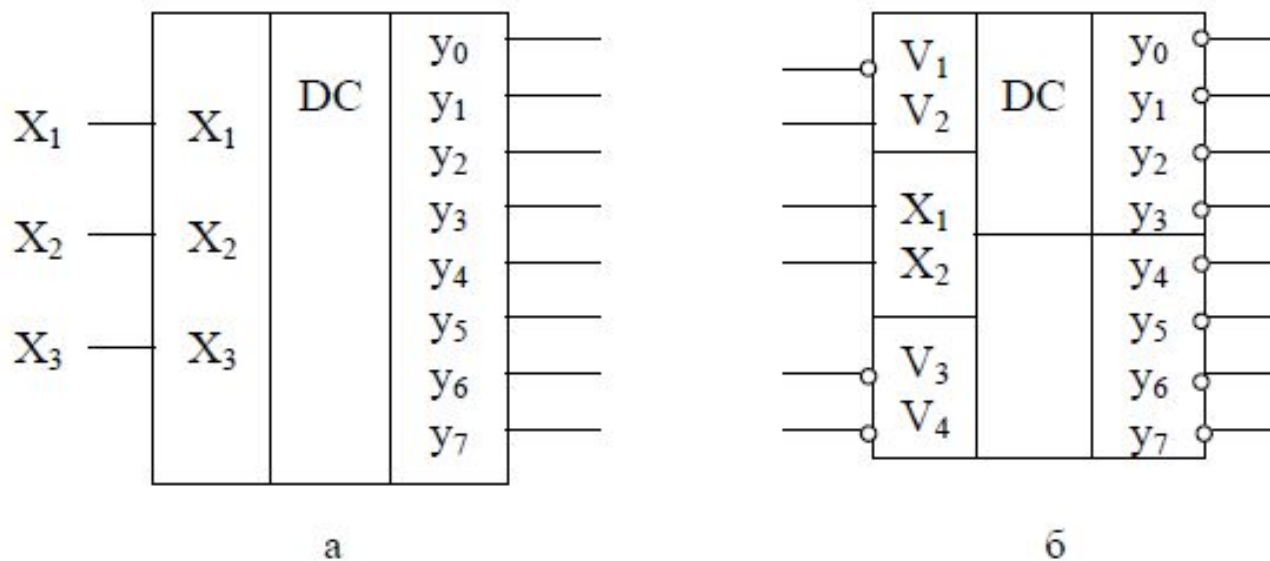


Рис. 3.6. Условное графическое обозначение дешифратора (а)
и условное графическое обозначение ИМС К155ИД4 (б)

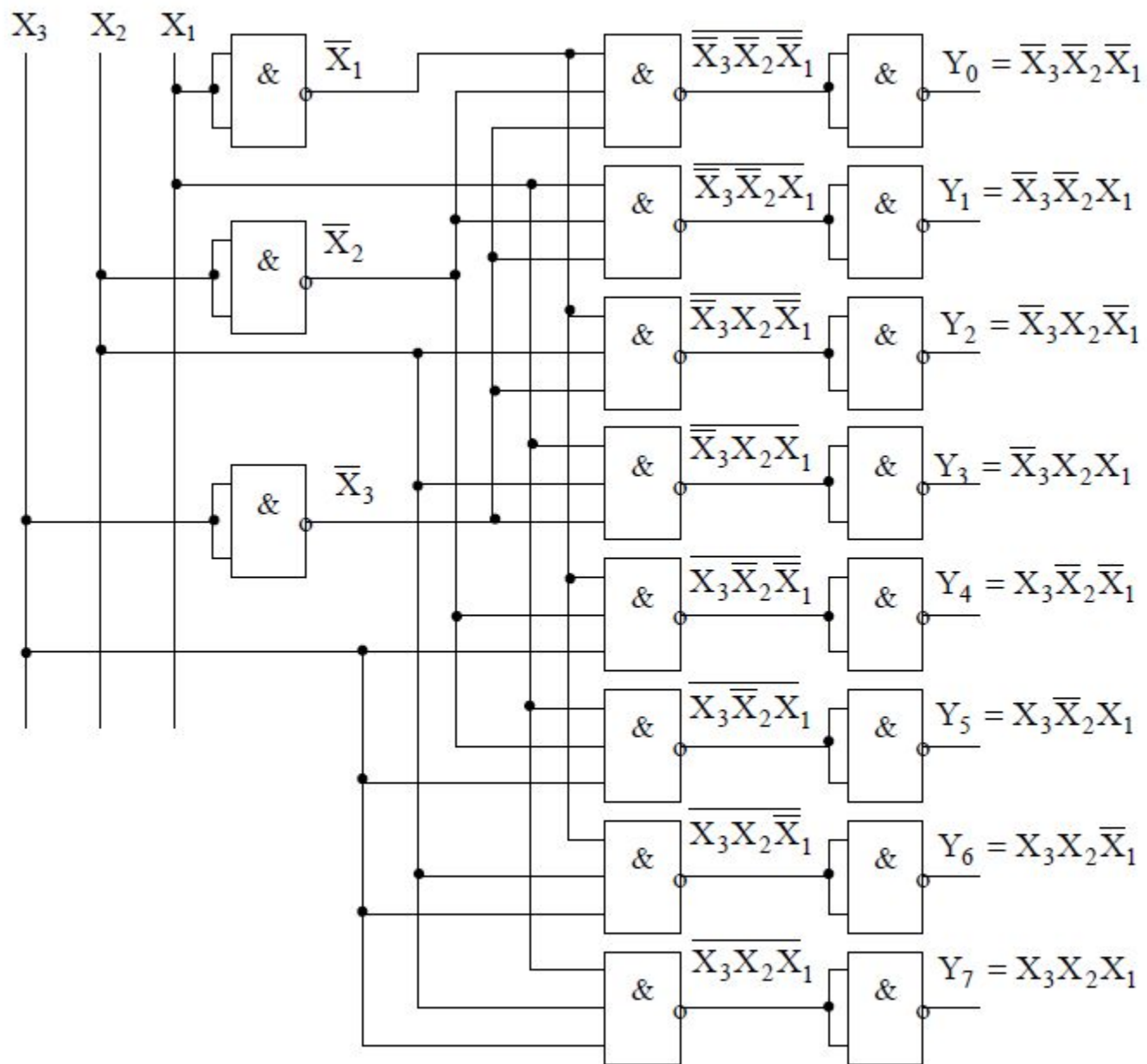


Рис. 3.7. Принципиальная схема линейного дешифратора на элементах И-НЕ

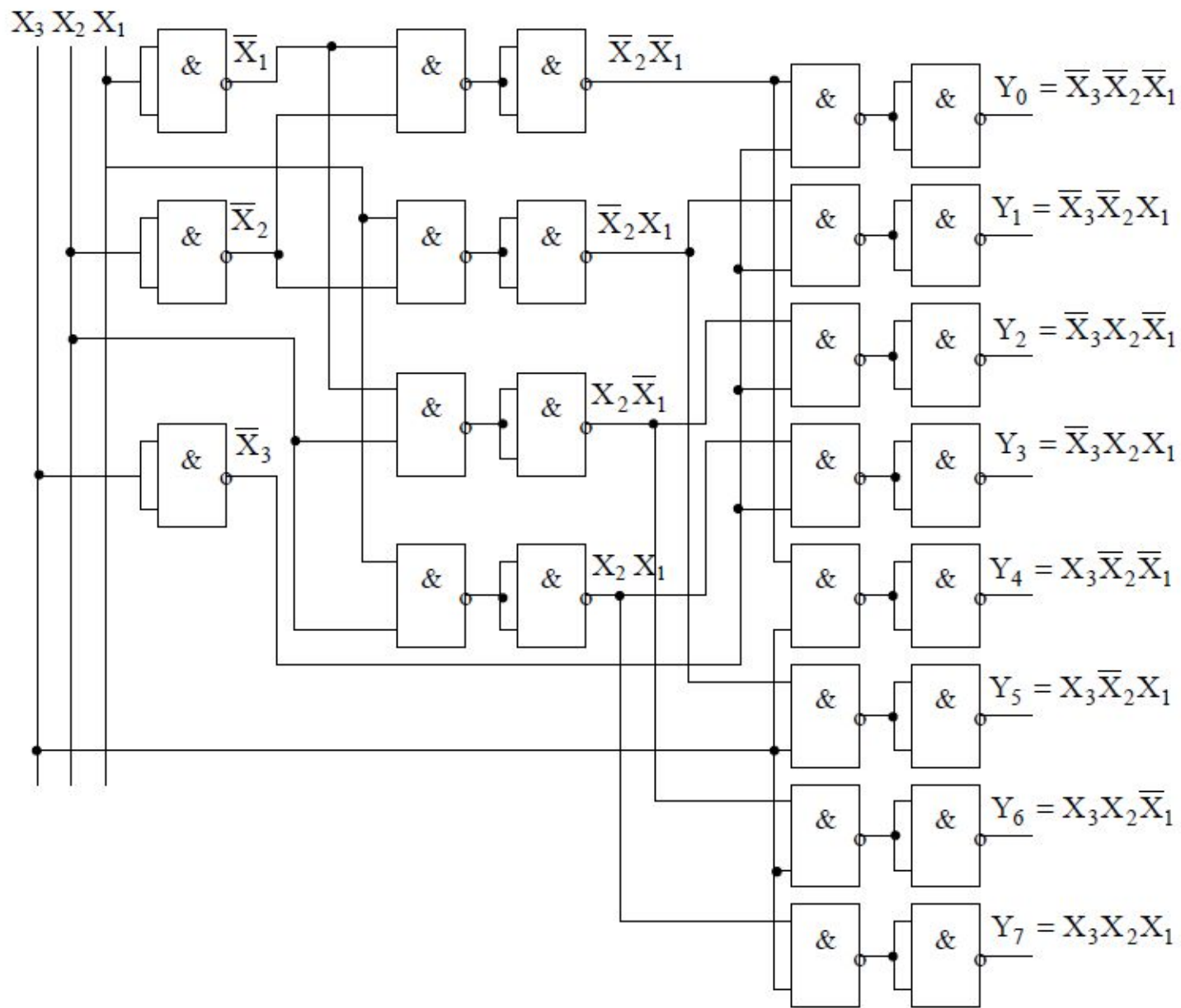


Рис. 3.8. Принципиальная схема пирамидального дешифратора

6.1.2. ПЗУ в генераторах импульсных последовательностей

Следующее важнейшее применение ПЗУ — это построение генераторов сложных последовательностей цифровых импульсов. Такие генераторы широко используются в самых разных измерительных системах,

в устройствах автоматики, в телевизионных системах, в схемах управления линейными или матричными индикаторами и т. д.

Задача в данном случае ставится следующим образом. Необходимо сформировать последовательность из нескольких сигналов различной длительности, сдвинутых относительно друг друга на различные временные интервалы. Причем последовательность эта может быть как разовой (однократно начинающейся по внешнему сигналу), так и периодической, непрерывно повторяющейся.

Наиболее распространенная структура генератора последовательностей выходных сигналов на ПЗУ включает в себя тактовый генератор нужной частоты, счетчик с требуемым числом разрядов, ПЗУ и выходной регистр (рис. 6.11). Счетчик перебирает адреса ПЗУ, ПЗУ последовательно выдает на выходы данных все записанные в него коды. Выходной регистр, тактируемый тем же тактовым сигналом, что и счетчик, служит для предотвращения появления в выходных сигналах паразитных импульсов и для обеспечения одновременного переключения всех выходных сигналов (что особенно важно в случае, когда используются несколько параллельно включенных микросхем ПЗУ для увеличения разрядности шины данных).

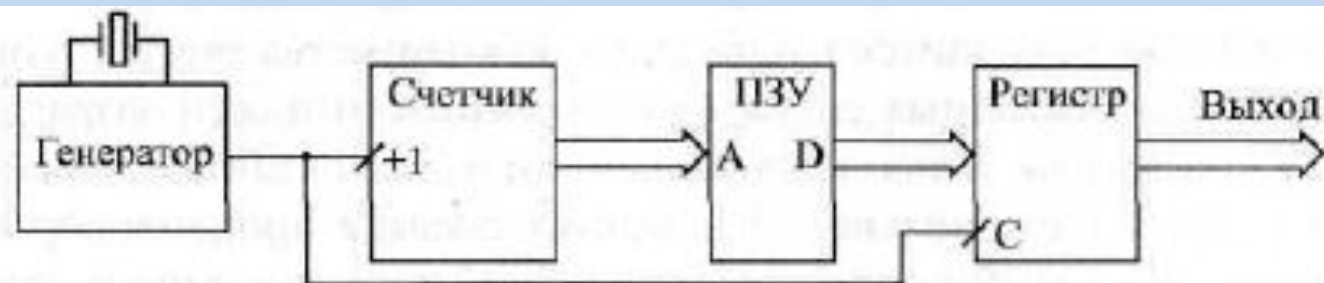


Рис. 6.11. Пример структуры генератора последовательностей сигналов на ПЗУ

Рассмотрим пример. Пусть необходимо непрерывно формировать периодическую последовательность из шести выходных сигналов в соответствии с временной диаграммой рис. 6.12. Получить такую последовательность можно, конечно, с помощью комбинационных схем, включенных на выходе счетчика, или с помощью множества одновибраторов, запускающих друг друга, но и то, и другое решение чересчур громоздко и сложно как в проектировании, так и в настройке. Применение же ПЗУ значительно упрощает задачу. Достаточно провести несложные расчеты и составить карту прошивки ПЗУ.

Расчеты сводятся к следующему.

Прежде всего определяем минимально возможную тактовую частоту (с целью минимизации требуемого объема ПЗУ). Для этого надо выделить максимальный временной интервал (дискрет времени), который укладывается целое число раз во все временные сдвиги, задержки, длительности

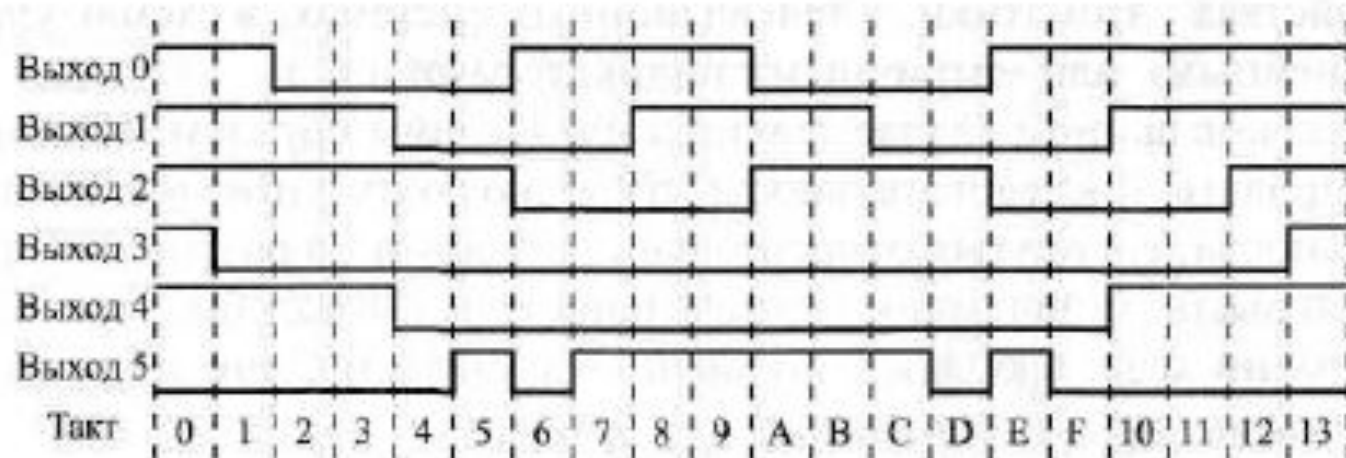


Рис. 6.12. Временная диаграмма формируемых выходных сигналов

требуемой диаграммы. В нашем случае этот дискрет равен одному делению по оси времени. Например, если длительность этого деления равна 250 нс, то и период тактового сигнала надо выбирать 250 нс, то есть тактовая частота будет равна 4 МГц. Можно, конечно, выбрать ее и кратной 4 МГц, например, 8 МГц, 12 МГц, но тогда потребуется вдвое или втрое больший объем ПЗУ. Если бы нам надо было формировать только три верхних сигнала (Выход 0, Выход 1, Выход 2), то период тактовой частоты можно было бы брать вдвое больше (в нашем примере — 500 нс), так как для этих сигналов все длительности кратны двум делениям.

Второй расчет сводится к определению количества ячеек и разрядности ПЗУ. Шесть выходных сигналов схемы требуют шести разрядов данных ПЗУ. Длительность последовательности равна 20 тактам (или 14 в 16-ричном коде), то есть не равна 2^n , поэтому счетчик придется сбрасывать в нуль через каждые 20 тактов, для чего потребуется еще один разряд данных ПЗУ. Итого потребуется 7 разрядов. А для перебора 20 тактов последовательности потребуется 5-разрядный счетчик, так как $2^4 = 16$ (недостаточно), а $2^5 = 32$ (достаточно). Значит, разрядность шины адреса ПЗУ также должна быть не менее пяти, то есть минимальные требования к организации ПЗУ — это 32x8, значит, подойдет микросхема ПЗУ типа РЕЗ.

Наконец, третий расчет касается условий правильной работы схемы. Генератор последовательности будет работать правильно, если за период тактового сигнала успеют сработать счетчик и ПЗУ. То есть сумма задержки полного переключения счетчика и задержки выборки адреса ПЗУ не должна превышать периода тактового сигнала.

Таким образом, один из возможных вариантов схемы генератора последовательности импульсов (рис. 6.13) будет включать в себя тактовый генератор, пятиразрядный счетчик на основе ИЕ19, ПЗУ типа К155РЕЗ и 8-разрядный выходной регистр ИР27. Так как счетчик срабатывает по отрицательному фронту тактового сигнала, а регистр - по положительному фронту тактового сигнала, необходимо включить инвертор. На схеме для простоты не показаны резисторы на выходах данных типа ОК микросхемы ПЗУ.

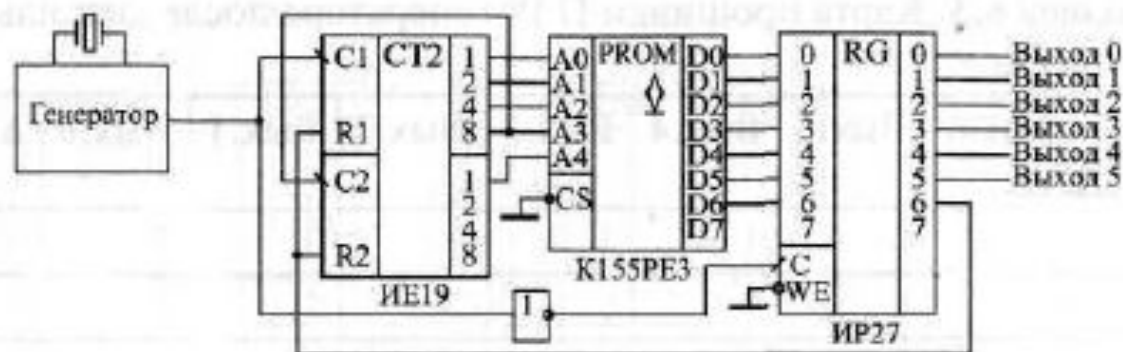


Рис. 6.13. Первый вариант схемы генератора последовательности сигналов на ПЗУ

Из схемы видно, что существует еще одно дополнительное условие правильной работы, связанное с использованием сигнала сброса. За период тактового сигнала должен успеть сработать регистр (записать в себя сигнал сброса с выхода ПЗУ), должен сброситься счетчик и должно выдать код ПЗУ. То есть сумма задержек регистра, сброса счетчика и выборки адреса ПЗУ не должна превышать периода тактового сигнала.

Теперь составим карту прошивки ПЗУ. Для этого сформируем таблицу значений всех семи выходных сигналов во всех двадцати рабочих тактах генератора последовательностей (табл. 6.5). Значения шести выходных сигналов (Выход 0... Выход 5) мы непосредственно берем из требуемой временной диаграммы (см. рис. 6.12). Хотя эти сигналы будут приходить на выход схемы с задержкой на один такт из-за наличия выходного регистра, но при периодической работе генератора последовательности это не имеет никакого значения.

Седьмой сигнал (Выход 6) используется в качестве сброса счетчика, он равен нулю от нулевого такта до 18 (в 16-ричном коде — 12) такта и равен единице в последнем девятнадцатом такте (в 16-ричном коде — 13). Этим единичным сигналом счетчик будет сбрасываться в нуль, то есть работа схемы после 19 такта будет возобновляться с нулевого такта.

Пусть нам требуется более высокое быстродействие схемы. Например, период тактового генератора должен быть равен 100 нс и даже меньше. В этом случае асинхронный счетчик типа ИЕ19 уже не подойдет из-за своего низкого быстродействия. Надо использовать 5-разрядный синхронный счетчик. Можно, конечно, включить две микросхемы 4-разрядных счетчиков типа ИЕ17, но можно обойтись и одним 4-разрядным счетчиком, если задействовать оставшийся свободным выход данных ПЗУ и оставшийся свободным разряд регистра (рис. 6.14). При этом без всякого ущерба для быстродействия можно использовать синхронный счетчик с асинхронным переносом ИЕ7, так как нужна всего одна микросхема счетчика

Таблица 6.5. Карта прошивки ПЗУ генератора последовательности сигналов

Такт (адрес)	Вых.6	Вых.5	Вых.4	Вых.3	Вых.2	Вых.1	Вых.0	Код (данные)
0	0	0	1	1	1	1	1	1F
1	0	0	1	0	1	1	1	17
2	0	0	1	0	1	1	0	16
3	0	0	1	0	1	1	0	16
4	0	0	0	0	1	0	0	04
5	0	1	0	0	1	0	0	24
6	0	0	0	0	0	0	1	01
7	0	1	0	0	0	0	1	21
8	0	1	0	0	0	1	1	23
9	0	1	0	0	0	1	1	23
A	0	1	0	0	1	1	0	26
B	0	1	0	0	1	1	0	26
C	0	1	0	0	1	0	0	24
D	0	0	0	0	1	0	0	04
E	0	1	0	0	0	0	1	21
	0	0	0	0	0	0	1	01
10	0	0	1	0	0	1	1	13
11	0	0	1	0	0	1	1	13
12	0	0	1	0	1	1	1	17
13	1	0	1	1	1	1	1	5F

Карта прошивки ПЗУ генератора последовательностей остается той же самой, что и в предыдущем случае, но добавляется один (седьмой) разряд шины данных ПЗУ. Этот разряд должен быть равен нулю в пятнадцати тактах с нулевого до 14 (в 16-ричном коде — E), в четырех тактах с 15 (F в 16-ричном коде) по 18 (12 в 16-ричном коде) он должен быть равен единице (код данных увеличится на 80), а в последнем 19 такте (13 в 16-ричном коде) он снова должен быть равен нулю. В результате адреса ПЗУ будут перебираться от 0 до 15 (старший разряд адреса равен нулю), затем от 16 до 19 (старший разряд адреса станет равен единице, а счетчик после переполнения считает с нуля), а затем снова от нулевого адреса (счетчик сбросится сигналом "Выход 6", а старший разряд адреса памяти станет ну-

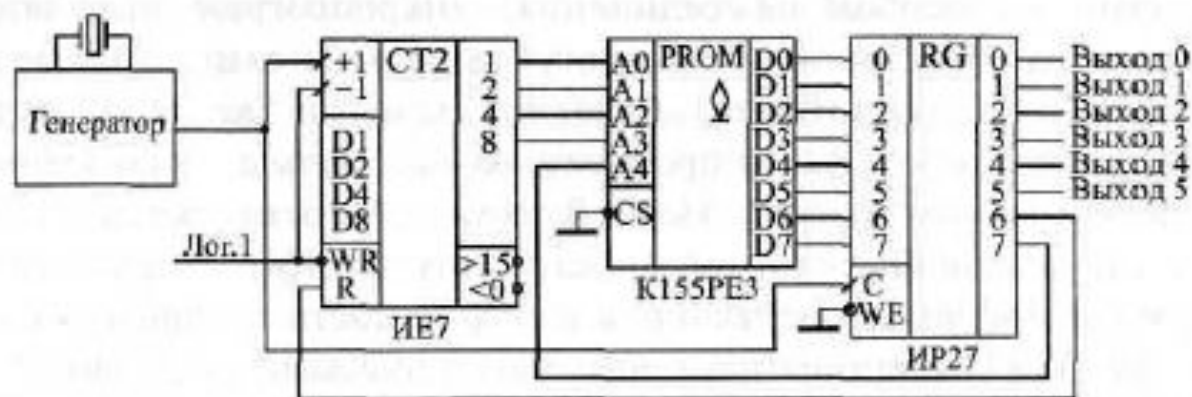


Рис. 6.14. Второй вариант схемы генератора последовательности с 4-разрядным синхронным счетчиком ИЕ7

левым). Получающаяся в результате карта прошивки ПЗУ приведена в табл. 6.6.

Таблица 6.6. Карта прошивки ПЗУ для варианта схемы генератора последовательности импульсов

Адрес	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	1F	17	16	16	04	24	01	21	23	23	26	26	24	04	21	81
10	93	93	97	5F	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

Конечно, рассмотренный пример довольно прост, при использовании ПЗУ большего объема генерируемые последовательности сигналов могут быть гораздо сложнее рассмотренной, и выходных сигналов может быть больше, но суть остается прежней: ПЗУ выдает любые последовательности выходных сигналов минимальными средствами. Причем генерируемые последовательности можно довольно просто изменять, заменяя ПЗУ на микросхемы с другой картой прошивки.

6.1.3. Микропрограммные автоматы на ПЗУ

Микропрограммные автоматы представляют собой следующий шаг по пути усложнения интеллекта цифровых схем. На основе микропрограммных автоматов можно строить устройства, которые работают по довольно сложным алгоритмам, выполняют различные функции, определяемые входными сигналами, выдают сложные последовательности выходных сигналов. При этом алгоритм работы микропрограммного автомата может быть легко изменен заменой прошивки ПЗУ.

В отличие от рассматривавшихся ранее устройств на "жесткой" логике, принцип работы которых однозначно определяется используемыми элементами и способом их соединения, микропрограммные автоматы с помощью одной и той же схемы могут выполнять самые разные функции. То есть они гораздо более гибкие, чем схемы на "жесткой" логике. К тому же проектировать микропрограммные автоматы с точки зрения схемотехники довольно просто. Недостатком любого микропрограммного автомата по сравнению со схемами на "жесткой" логике является меньшее предельное быстродействие и необходимость составления карты прошивки ПЗУ с микропрограммами, часто довольно сложными.

Наиболее распространенная структура микропрограммного автомата (рис. 6.15) включает в себя всего лишь три элемента: ПЗУ, регистр, срабатывающий по фронту, и тактовый генератор.

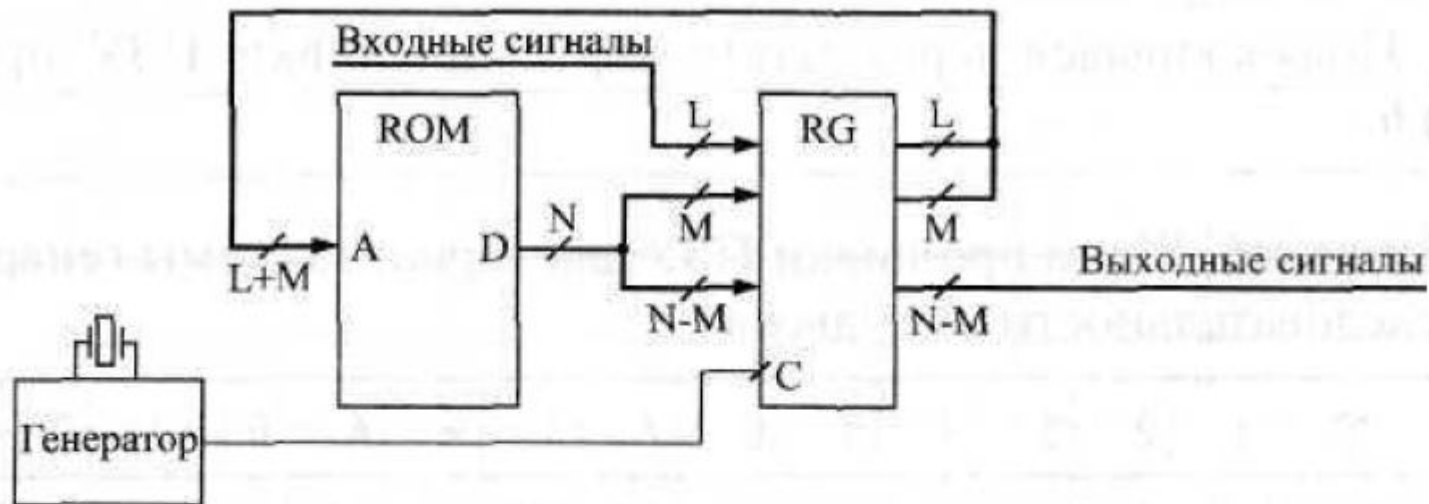


Рис. 6.15. Структура микропрограммного автомата

ПЗУ имеет $(L+M)$ адресных разрядов и N разрядов данных. Регистр применяется с количеством разрядов $(N + L)$. Разряды данных ПЗУ записываются в регистр по положительному фронту тактового сигнала с генератора. Часть этих разрядов (M) используется для образования адреса ПЗУ, другая часть ($N-M$) служит для формирования выходных сигналов. Входные сигналы (L) поступают на входы регистра и используются совместно с частью выходных разрядов ПЗУ для получения адреса ПЗУ.

Схема работает следующим образом. В каждом такте ПЗУ выдает код данных, тем самым определяя не только состояние выходных сигналов схемы, но и адрес ПЗУ, который установится в следующем такте (после следующего положительного фронта тактового сигнала). На этот следующий адрес влияют также и входные сигналы. То есть в отличие от формирователя последовательности сигналов, рассмотренного в предыдущем разделе, в данном случае адреса могут перебираться не только последовательно (с помощью счетчика), но и в произвольном порядке, который определяется прошивкой ПЗУ, называемой микропрограммой.

Условием правильной работы схемы будет следующее. За один период тактового сигнала должны успеть сработать регистр и ПЗУ. Иначе говоря, сумма задержки регистра и задержки выборки адреса ПЗУ не должна превышать периода тактового сигнала. Отметим также, что входные сигналы микропрограммного автомата нельзя подавать непосредственно на адресные входы ПЗУ (без регистра), так как их асинхронное (по отношению к тактовому сигналу) изменение может вызвать переходный процесс на выходах данных ПЗУ именно в тот момент, когда выходные сигналы ПЗУ записываются в регистр. В результате в регистр может записаться неверная информация, что нарушит работу всей схемы. Регистр же синхронизирует изменения входных сигналов с тактовым сигналом, в результате чего все разряды кода адреса ПЗУ меняется одновременно — по положительному фронту тактового сигнала. Регистр должен обязательно срабатывать по фронту, использование регистра-защелки не допускается, так как он может вызвать лавинообразный переходный процесс.

Возможности такой простой схемы оказываются очень большими. Например, микропрограммный автомат может выдавать последовательности выходных сигналов в ответ на определенное изменение входных сигналов. Он может также временно остановить выдачу выходных сигналов до прихода входных сигналов. Он может анализировать длительность входного сигнала и в зависимости от нее выдавать те или иные выходные сигналы. Он может также и многое другое.

Сформулируем несколько элементарных функций, из которых могут складываться алгоритмы работы микропрограммного автомата (рис. 6.16):

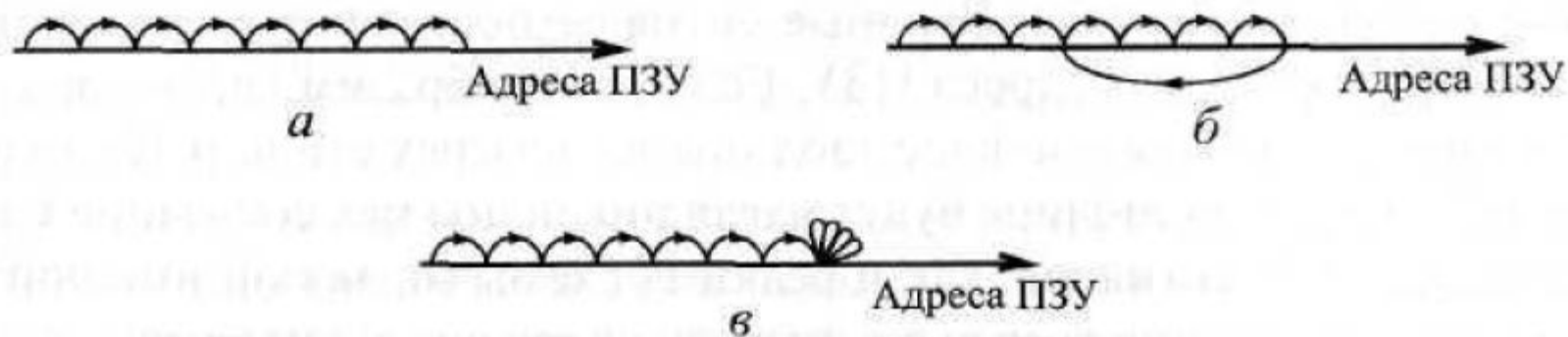


Рис. 6.16. Элементарные функции микропрограммного автомата: последовательный перебор (*a*), циклическое повторение (*б*) и остановка (*в*)

- а) Последовательный перебор адресов ПЗУ (например, для выдачи последовательности выходных сигналов).
- б) Периодическое повторение последовательности адресов ПЗУ (например, для повторения последовательности выходных сигналов).
- в) Остановка в каком-то адресе ПЗУ (например, для ожидания изменения входного сигнала).
- г) Временное отключение реакции на входные сигналы (например, для того, чтобы закончить обработку реакции на предыдущее изменение входных сигналов). Эта функция на рисунке не показана.

В качестве примера рассмотрим выполнение некоторых элементарных функций микропрограммным автоматом, показанным на рис. 6.17.

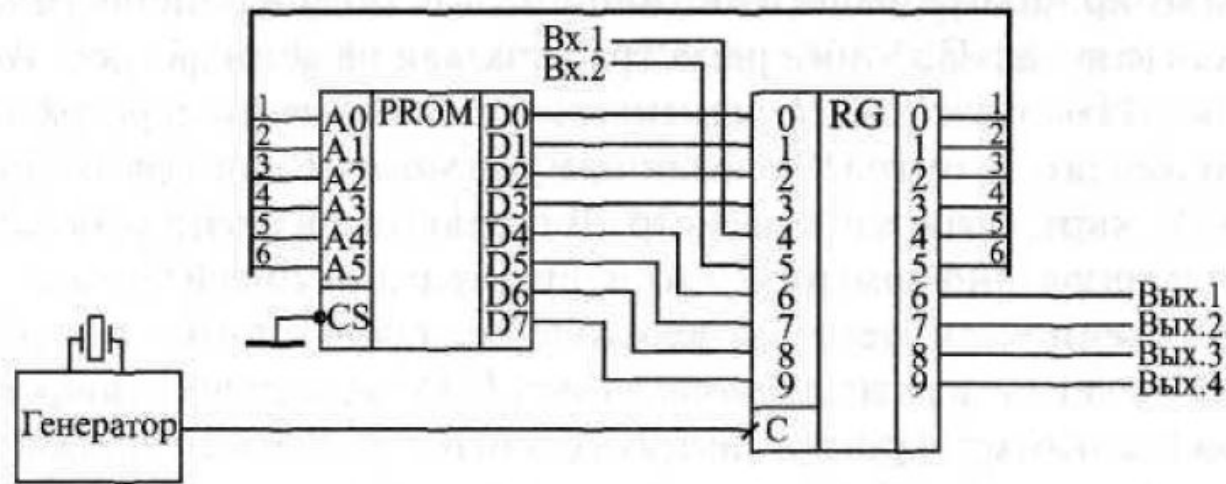


Рис. 6.17. Пример практической схемы микропрограммного автомата на ПЗУ

- **Микроконтроллер** - это специальная микросхема, предназначенная для управления различными электронными устройствами. Микроконтроллеры впервые появились в том же году, что и микропроцессоры общего назначения (1971).
- Разработчики микроконтроллеров придумали остроумную идею – объединить процессор, память, ПЗУ и периферию внутри одного корпуса, внешне похожего на обычную микросхему. С тех пор производство микроконтроллеров ежегодно во много раз превышает производство процессоров, а потребность в них не снижается.
- Микроконтроллеры выпускают десятки компаний, причем производятся не только современные 32-битные микроконтроллеры, но и 16, и даже 8-битные (как i8051 и аналоги). Внутри каждого семейства часто можно встретить почти одинаковые модели, различающиеся скоростью работы ЦПУ и объемом памяти.
- Дело в том, что микроконтроллеры применяются преимущественно во встроенных системах, в игрушках, в станках, в массовой домашней технике, в [домашней автоматике](#) – там, где нужна не мощность процессора, а, скорее, баланс между ценой и достаточной функциональностью.

- **Памятью** ЭВМ называется совокупность устройств, служащих для запоминания, хранения и выдачи информации.
- Отдельные устройства, входящие в эту совокупность, называются **запоминающими устройствами** (ЗУ) того или иного типа [7].
- Термин "запоминающее устройство" обычно используется, когда речь идет о принципе построения некоторого устройства памяти (например, полупроводниковое ЗУ, ЗУ на жестком магнитном диске и т.п.), а термин "память" - когда хотят подчеркнуть выполняемую устройством памяти логическую функцию или место расположения в составе оборудования ЭВМ (например, оперативная память - ОП, внешняя память и т.п.). В тех вопросах, где эти отличия не имеют принципиального значения, термины "память" и "запоминающее устройство" мы будем использовать как синонимы.
- *Запоминающие устройства* играют важную роль в общей структуре ЭВМ. По некоторым оценкам *производительность* компьютера на разных классах задач на 40-50% определяется характеристиками ЗУ различных типов, входящих в его состав.
- К основным параметрам, характеризующим *запоминающие устройства*, относятся *емкость* и *быстродействие*.
- **Емкость памяти** - это максимальное количество данных, которое в ней может храниться.
- **Емкость запоминающего устройства** измеряется количеством адресуемых элементов (ячеек) ЗУ и длиной ячейки в битах. В настоящее время практически все *запоминающие устройства* в качестве минимально адресуемого элемента используют 1 *байт* (1 *байт* = 8 двоичных разрядов (*бит*)). Поэтому *емкость памяти* обычно определяется в байтах, килобайтах (1Кбайт=2¹⁰ *байт*), мегабайтах (1Мбайт = 2²⁰ *байт*), гигабайтах (1Гбайт = 2³⁰ *байт*) и т.д.

- За одно обращение к *запоминающему устройству* производится считывание или запись некоторой единицы данных, называемой **словом**, различной для устройств разного типа. Это определяет разную **организацию памяти**. Например, *память* объемом 1 *мегабайт* может быть организована как 1М слов *по* 1 байту, или 512К слов *по* 2 байта каждое, или 256К слов *по* 4 байта и т.д.
- В то же время, в каждой ЭВМ используется свое понятие **машинного слова**, которое применяется при определении архитектуры компьютера, в частности при его программировании, и не зависит от размерности слова памяти, используемой для построения данной ЭВМ. Например, компьютеры с архитектурой *IBM PC* имеют машинное слово длиной 2 байта.
- **Быстродействие памяти** определяется продолжительностью *операции* обращения, то есть временем, затрачиваемым на *поиск* нужной информации в памяти и на ее считывание, или временем на *поиск* места в памяти, предназначенного для хранения данной информации, и на ее *запись*:
 - $t_{\text{обр}} = \max(t_{\text{обр сч}}, t_{\text{обр зп}})$
 - где $t_{\text{обр сч}}$ - *быстродействие ЗУ* при считывании информации; $t_{\text{обр зп}}$ - *быстродействие ЗУ* при записи.

- **Классификация запоминающих устройств**
- *Запоминающие устройства* можно классифицировать по целому ряду параметров и признаков. На [рис.5.1](#) представлена классификация по типу обращения и организации доступа к ячейкам ЗУ.



Классификация запоминающих устройств

По типу обращения ЗУ делятся на устройства, допускающие как чтение, так и запись информации, и постоянные *запоминающие устройства* (ПЗУ), предназначенные только для чтения записанных в них данных (**ROM** - **read only memory**). ЗУ первого типа используются в процессе работы процессора для хранения выполняемых программ, исходных данных, промежуточных и окончательных результатов. В ПЗУ, как правило, хранятся *системные программы*, необходимые для запуска компьютера в работу, а также *константы*. В некоторых ЭВМ, предназначенных, например, для работы в системах управления по одним и тем же неизменяемым алгоритмам, все *программное обеспечение* может храниться в ПЗУ.

В **ЗУ с произвольным доступом** (**RAM** - **random access memory**) время доступа не зависит от места расположения участка памяти (например, ОЗУ).

В **ЗУ с прямым (циклическим) доступом** благодаря непрерывному вращению носителя информации (например, магнитный диск - МД) возможность обращения к некоторому участку носителя циклически повторяется. Время доступа здесь зависит от взаимного расположения этого участка и головок чтения/записи и во многом определяется скоростью вращения носителя.

В *ЗУ с последовательным доступом* производится последовательный просмотр участков носителя информации, пока нужный участок не займет некоторое нужное положение напротив головок чтения/записи (например, магнитные ленты - МЛ).

Как отмечалось выше, основные характеристики *запоминающих устройств* - это *емкость* и *быстродействие*. Идеальное *запоминающее устройство* должно обладать бесконечно большой *емкостью* и иметь бесконечно малое время обращения. На практике эти параметры находятся в противоречии друг другу: в рамках одного типа *ЗУ* *улучшение* одного из них ведет к ухудшению значения другого. К тому же следует иметь в виду и экономическую целесообразность построения *запоминающего устройства* с теми или иными характеристиками при данном уровне развития технологии. Поэтому в настоящее время *запоминающие устройства* компьютера, как это и предполагал Нейман, строятся *по иерархическому принцип*

В общем случае ЭВМ можно классифицировать по ряду признаков.

1. По принципу действия ЭВМ делятся на три больших класса в зависимости от формы представления информации, с которой они работают:

- **АВМ** – аналоговые вычислительные машины непрерывного действия, работают с информацией, представленной в непрерывной (аналоговой) форме, то есть в виде непрерывного ряда значений какой-либо физической величины (чаще всего электрического напряжения);
- **ЦВМ** – цифровые вычислительные машины дискретного действия, работают с информацией, представленной в дискретной (цифровой) форме;
- **ГВМ** – гибридные вычислительные машины комбинированного действия работают с информацией, представленной как в цифровой, так и в аналоговой форме. ГВМ совмещают в себе достоинства АВМ и ЦВМ. Их целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.

По назначению ЭВМ можно разделить на три группы:

- **универсальные ЭВМ** предназначены для решения самых различных инженерно-технических задач: экономических, математических, информационных и других задач, отличающихся сложностью алгоритмов и большим объемом обрабатываемых данных. Они широко используются в вычислительных центрах коллективного пользования и в других мощных вычислительных комплексах. Характерными чертами универсальных ЭВМ является: высокая производительность; разнообразие форм обрабатываемых данных при большом диапазоне их изменения и высокой степени их представления; обширная номенклатура выполняемых операций, как арифметических, логических, так и специальных; большая емкость оперативной памяти; развитая организация системы ввода-вывода информации, обеспечивающая подключение разнообразных видов внешних устройств;
- **проблемно-ориентированные ЭВМ** служат для решения более узкого круга задач, связанных, как правило, с управлением технологическими процессами. Они используются для регистрации, накопления и обработки относительно небольших объемов данных, выполнения расчетов по относительно несложным алгоритмам. Проблемно-ориентированные ЭВМ обладают ограниченными по сравнению с универсальными ЭВМ аппаратными и программными ресурсами;
- **специализированные ЭВМ** используются для решения узкого круга задач или реализации строго определенной группы функций. Узкая ориентация ЭВМ позволяет четко определить их структуру, существенно снизить сложность и стоимость при сохранении высокой производительности и надежности их работы. К специализированным ЭВМ можно отнести, например, программируемые микропроцессоры специального назначения.