

**КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**  
**Институт физики**  
**Кафедра радиофизики**

**Карпов А.В., Калабанов С.А., Ишмуратов Р.А.**

**ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ**

*Учебное пособие*



**Казань**  
**2019**

**УДК 621.39**

**ББК 32.811**

**К26**

*Рекомендовано к изданию  
Учебно-методическим центром КФУ*

**Рецензенты:**

Кандидат физико-математических наук, доцент **Хузяшев Р.Г.**

Кандидат физико-математических наук, доцент **Насыров И.А.**

**Карпов А.В.**

**K26 Основы цифровой электроники: учебное пособие / А.В. Карпов, С.А. Калабанов, Р.А. Ишмуратов. – Казань: Изд-во Казан. ун-та, 2019. – 75 с.**

**ISBN**

Данное пособие написано на базе конспекта лекций по курсу «Цифровая электроника», читаемых студентам 2-го курса Института физики направления бакалавриата 03.03.03 – радиофизика. В представленном учебном пособии последовательно излагаются основные теоретические положения и содержание классических разделов цифровой электроники: аналоговые и цифровые сигналы, логические функции, принципы двоичного кодирования и булевой алгебры, основные узлы систем цифровой электроники и их схемотехническая реализация и др. Излагаемый материал сопровождается подробными графическими иллюстрациями и таблицами. Каждый раздел завершается списком контрольных вопросов, служащих для закрепления понимания учебного материала. Курс «Цифровая электроника» относится к блоку общепрофессиональных дисциплин и служит базой для дальнейшего освоения специальных дисциплин направления «радиофизика».

**УДК 621.39**

**ББК 32.811**

**ISBN**

**© Карпов А.В., Калабанов С.А., Ишмуратов Р.А., 2019**

**© Издательство Казанского университета, 2019**

## ОГЛАВЛЕНИЕ

Введение .....	4
1. Системы счисления, используемые в цифровой электронике .....	10
2. Логические функции .....	22
3. Комбинационные логические схемы.....	28
4. Кодирование .....	34
5. Последовательные логические схемы. Триггеры .....	42
6. Счетчики .....	57
7. Регистры .....	61
8. Арифметические устройства .....	65
9. Сопряжение цифровых и аналоговых схем .....	70
Литература .....	74

## **ВВЕДЕНИЕ**

### **Три базовых принципа цифровой электроники. Цифровые и аналоговые сигналы. Последовательный и параллельный код**

#### ***Три базовых принципа цифровой электроники***

Современная информатизация жизни общества достигнута во многом бурным развитием цифровой техники и электроники, которые обеспечивают наиболее совершенные принципы, технологии и средства взаимодействия человека с окружающим миром, как в науке и производстве, так и в быту. Появились новые информационные технологии и области их применения: цифровая связь, цифровое телевидение, цифровая радиолокация и навигация и др. По сравнению с аналоговыми системами применение цифровых технологий обеспечивает высокую, ранее недостижимую, точность, надежность и быстродействие. В настоящее время подавляющее большинство всех разрабатываемых технических устройств – это цифровые устройства и системы. Цифровая техника и цифровые технологии применяются в самых различных областях науки, технике, производстве, в быту. Так, например, цифровые устройства (микроконтроллеры) составляют основу современных гаджетов и различной бытовой техники, придавая им совершенно новые качества и функциональные возможности.

Целью курса «Цифровая электроника» является ознакомление с основами цифровой электроники и принципами построения цифровых устройств. Без получения знаний по основам цифровой электроники невозможно дальнейшее освоение специальных дисциплин, содержание которых связано с широким использованием в современных радиоэлектронных устройствах и технических средствах защиты информации таких цифровых узлов и систем, как микроконтроллер, цифровой сигнальный процессор (специализированный микропроцессор, предназначенный для цифровой обработки сигналов в реальном масштабе времени), ПЛИС (программируемые логические интегральные схемы) и др.

Можно выделить три основных составляющих, или раздела (образно говоря, «три кита»), на которых поконится весь фундамент цифровой электроники, это:

- Теорема Котельникова-Найквиста;
- Булева алгебра;
- Импульсная техника.

В теореме Котельникова-Найквиста обосновывается возможность получения полного цифрового эквивалента аналогового сигнала: произвольный сигнал с ограниченным спектром (с верхней граничной частотой  $f_m$ ) однозначно определяется его дискретными значениями (отсчетами), выбранными через равные промежутки времени  $\Delta t$ , которые должны удовлетворять условию (1):

$$\Delta t \leq \frac{1}{2f_m} \quad (1)$$

Величину  $\Delta t$  называют *шагом, или интервалом, дискретизации*. Соответственно, *частота дискретизации*  $f_d$  (величина, обратная  $\Delta t$ ) должна удовлетворять условию:

$$f_d \geq 2f_m$$

При выполнении условия теоремы Котельникова возможно *полное восстановление* аналогового сигнала из последовательности его дискретных отсчетов.

Математическим фундаментом цифровой электроники является алгебра логики, или булева алгебра. *Булева алгебра* позволила поставить анализ и синтез цифровых систем на строгую математическую основу.

В цифровой электронике нашли свое применение и развитие многие достижения импульсной электронной техники. *Импульсная техника* – это область радиоэлектроники, предметом которой является разработка теоретических основ, практических методов и технических средств генерирования (формирования), преобразования и измерения параметров *электрических импульсов* (которые нашли широкое применение, главным образом, в устройствах автоматики, вычислительной технике и т.д.).

### *Цифровые и аналоговые сигналы*

С точки зрения цифровой электроники, основное различие между цифровым и аналоговым сигналом заключается в следующем: аналоговый сигнал может принимать *бесконечное число состояний*, в отличие от цифрового сигнала, который может принимать *только конечное число состояний*.

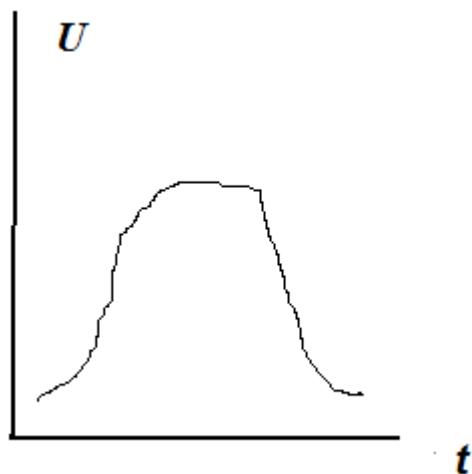


Рис.в.1. Пример аналогового сигнала

Цифровые сигналы подразделяются на два типа:

1. Цифровой эквивалент аналогового сигнала;
2. Логические сигналы.

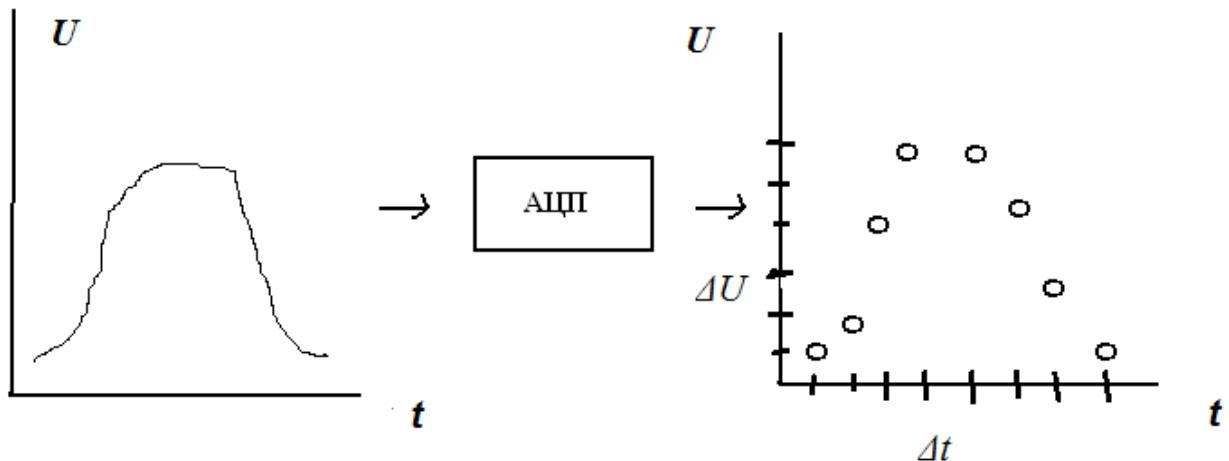


Рис. в.2. Процесс оцифровки аналогового сигнала

Первый тип цифрового сигнала является результатом дискретизации по времени и квантования по уровню аналогового сигнала (рис. в.2). Такое преобразование выполняется специальным устройством, называемым АЦП – аналого-цифровым преобразователем (в англ. терминологии, *ADC – analog digital converter*). На выходе АЦП получается цифровой эквивалент аналогового сигнала – последовательность его дискретных отсчетов:

$$x(t) \rightarrow (x_0, x_1, x_2, \dots)$$

С математической точки зрения, цифровой сигнал первого типа представляет собой временной ряд (последовательность чисел).

С технической точки зрения отдельные отсчеты дискретной последовательности (цифровые сигналы) представляют собой *двоичный код*, т.е. последовательность бит. Например, звуковые файлы представляют собой последовательность 16-битовых отсчетов.

Цифровые сигналы преобразуются в системах *цифровой обработки сигналов* (ЦОС). Современные системы ЦОС реализуется программно либо микропроцессорами (цифровыми сигнальными процессорами), либо непосредственно в компьютерах с применением соответствующего программного обеспечения.

Рассмотрим цифровые сигналы второго типа – логические сигналы. *Логические сигналы* связаны с наступлением или не наступлением того или иного события.

Таким событием, например, может быть наличие или отсутствие напряжения (тока) на выходе электрической цепи или линии связи. В цифровой электронике такие события получили названия, соответственно, логической единицы («1») и логического нуля («0»).

*Примечание.* Сигналы, которые могут принимать всего два возможных значения, получили название *бинарные сигналы*. Основное назначение бинарных сигналов – кодирование бит информации (0 или 1).

Логические единицы и логические нули могут кодироваться перепадами потенциалов напряжения  $U$  различными способами, например:

$U^1$  – высокий уровень напряжения соответствует логической «1» (уровень  $H$  (High));

$U^0$  – низкий уровень напряжения соответствует логическому «0» (уровень  $L$  (Low)).

Такая кодировка получила название «*положительной логики*». Соответственно, в «*отрицательной логике*» кодировка осуществляется наоборот:

- высокий уровень напряжения  $U^1$  соответствует логическому «0»;
- низкий уровень напряжения  $U^0$  соответствует логическому «1».

Понятие «высокий уровень» обычно соответствует напряжению 5 В (более точно, это диапазон напряжений от 2,5 до 5 В), а «низкий уровень» – 0 В (диапазон напряжений от 0 до 0,5 В).

*Примечание.* Кодирование такими значениями потенциалов соответствует так называемой «*ТТЛ логике*» (транзистор-транзисторная логика). ТТЛ логика широко применялась и применяется в настоящее время, став одним из основных стандартов цифровой электроники. Например, потенциалами, соответствующими ТТЛ логике,

кодируются биты в интерфейсе стандартного USB порта. Данный стандарт предполагает значение источника питания схем цифровой электроники +5 В. В случае использования пониженного питания электронных схем диапазоны напряжений для кодирования бит будут соответственно также снижены.

Кроме рассмотренных выше, возможны также другие физические кодировки логических «1» и «0» (соответственно, бит 1 и 0). Например, согласно широко распространенному интерфейсу стандарта RS-232, логическая «1» кодируется напряжением сигнала в диапазоне от -5 до -12 В, а логический «0» – напряжением сигнала в диапазоне от +5 до +12 В. Таким образом, в стандарте RS-232 как «1», так и «0» кодируется напряжением, одинаковым по значению, но противоположной полярности.

### ***Последовательный и параллельный двоичный код***

Цифровые логические сигналы могут быть представлены в последовательной и параллельной форме (соответственно, последовательном и параллельном двоичном коде). Определяющим фактором при выборе формы логического сигнала (кода) является количество каналов связи. Последовательный код реализуется при одном канале связи.

*Цифровая электроника основана на наименее избыточном алфавите, состоящем всего из двух символов: 0 и 1. Этот принцип цифровой электроники имеет ряд ярко выраженных достоинств, которые можно кратко описать следующим образом.*

1. Высокая помехоустойчивость при передаче сигналов – приемник цифровых сигналов осуществляет *выбор только между двумя состояниями*. Это значительно облегчает их идентификацию в условиях даже интенсивных помех.
2. Не происходит накопление искажений. Это позволяет повысить степень интеграции *до  $10^9$  элементов*.
3. Простота реализации систем цифровой электроники.
4. Реализация принципа: эквивалентность «программы» и «железа» (soft and hard):
  - а) любую программу можно реализовать аппаратным (схемным) способом;
  - б) любые аппаратные средства можно заменить посредством выполнения соответствующей программы.

*Примечание.* Все больше функций современных радиосистем реализуются посредством программного обеспечения (ПО). Эта современная тенденция развития радиотехники позволяет создавать радиооборудование, функциональность которого задается и изменяется программно.

5. Значительная дешевизна.
6. Прогресс в технологии производства микросхем позволяет все более значительно снижать энергопотребление систем цифровой электроники.

Наряду с отмеченные достоинствами есть также ряд причин, которые до сих пор сдерживали применение цифровой электроники в радиотехнике. Во-первых, использование цифровых систем требует интенсивной обработки цифровых данных, соответственно требуется высокоскоростные процессоры. Во-вторых, существует проблема синхронизации цифровых систем – необходимость правильно определять начало и конец принимаемого импульса (цифрового сигнала).

Развитие элементной базы цифровых устройств прошло большой путь от электронно-вакуумных ламп до чипов, или кристаллов БИС (больших интегральных схем), обладающих высокой степенью интеграции составных логических элементов. Это позволило решить проблему практического эффективного использования устройств цифровой электроники.

## **Контрольные вопросы**

1. Перечислите три базовых принципа цифровой электроники.
2. Сформулируйте теорему Котельникова-Найквиста и приведите условие его выполнения. Какие ограничения накладывает теорема Котельникова для работы цифровых систем обработки сигналов?
3. В чем состоит отличие цифрового сигнала от аналогового? Укажите два типа цифровых сигналов.
4. Поясните термины «положительная логика» и «отрицательная логика». В чем состоит их основное отличие?
5. Назовите основные преимущества цифровой электроники перед аналоговой электроникой.
6. Перечислите основные принципы и достоинства цифровой электроники.
7. Какие факторы сдерживали развитие цифровой электроники?

## 1. СИСТЕМЫ СЧИСЛЕНИЯ, ИСПОЛЬЗУЕМЫЕ В ЦИФРОВОЙ ЭЛЕКТРОНИКЕ

Человек в своей деятельности использует десятичную систему счисления. При этом работа компьютера и других средств вычислительной техники основана на *двоичной системе счисления*, которая в общем случае неудобна для восприятия человеком. В этом разделе мы рассмотрим подробнее системы счисления, которые используются в цифровой электронике, и покажем их связь с десятичной системой.

В процессе эволюции человечества потребность в счете всегда играла огромную роль, и не сразу человечество стало использовать десятичную систему. Древние ацтеки и майя использовали двадцатеричную систему. В древнем Шумере и древнем Вавилоне использовались двенадцатеричная и шестидесятеричная системы счисления. В Римской империи использовались так называемые римские цифры и система счисления, которая является характерным примером непозиционной системы счисления (обозначения чисел римским цифрами иногда используется и в наше время, например, для нумерации глав в книгах, обозначения годов и т.п.). После падения Римской империи в Европе начался переход к привычной нам десятичной позиционной системе счисления и записи чисел арабскими цифрами. В приведенной ниже таблице приведено соответствие основных числительных и их написание римскими цифрами.

Десятичные числа	1	2	3	5	10	50	100	500	1000
Римские числа	I	II	III	V	X	L	C	D	M

Произвольное промежуточное число между приведенными в таблице числительными получается путем простого сложения (или вычитания) базовых римских цифр (символов), например:

$$153 \Rightarrow CLIII, \quad 47 \Rightarrow XLVII \quad \text{и т.д.}$$

Очевидно, что римские цифры неудобны для записи больших чисел. Однако основной недостаток римских числительных – трудность их математических преобразований, например, умножения и деления. Это обусловлено коренным недостатком римской системы счета – тем, что она является непозиционной системой счисления. Перейдем к общему описанию систем счисления и связанной с ними терминологии.

*Система счисления* – это определенная система кодирования чисел, в которой используются специальные символы и правила их соединения в код для обозначения количества каких-либо объектов. Коротко говоря, система счисления – это совокупность символов (цифр) и правил для записи с их помощью произвольных чисел.

Существуют непозиционные и позиционные системы счисления. В *непозиционной* системе счисления значение символа, обозначающего то или иное число, не зависит от его позиции относительно других символов числа. Промежуточное число получается путем простого суммирования (соединения символов) без учета их веса, например:

$$153 = (100) + (50) + (3) = \text{CLIII}$$

*Позиционная* система счисления – система счисления, в которой значение каждого числового знака (цифры) в записи числа зависит от его места, или позиции (разряда), в этом числе (отсюда название – позиционная система счисления).

Позиционная система счисления основана на суммировании цифр с учетом их веса, зависящего от позиции в числе. Например, обратите внимание, что три числа: 153, 351, 513, записанные одинаковыми цифрами, дают разный результат.

*Основание системы счисления*,  $m$  – это количество используемых для записи чисел символов. Так, в десятичной системе счисления используется десять символов ( $m = 10$ ):

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

В *двоичной* системе счисления используется только два символа ( $m = 2$ ): 0, 1

Наряду с десятичной и двоичной системами иногда применяются другие системы – восьмеричная и шестнадцатеричная системы.

В *восьмеричной* системе счисления используется 8 символов ( $m = 8$ ):

$$0, 1, 2, 3, 4, 5, 6, 7$$

В *шестнадцатеричной* системе счисления используется 16 символов ( $m = 16$ ):

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \text{A}, \text{B}, \text{C}, \text{D}, \text{E}, \text{F}$$

Для различия записи чисел в разных системах основание системы счисления указывают в виде нижнего индекса в конце числа, например:

- $250_{10}$  – запись числа в десятичной системе счисления;

- $376_8$  – в восьмеричной системе счисления;
- $1011_2$  – в двоичной системе счисления;
- $A962_{16}$ , либо  $A962_H$ ,  $A962_h$  – в шестнадцатеричной системе счисления ( $H$ ,  $h$  – от англ. *Hexadecimal*).

### *Основные правила*

- Любое число  $N$  в системе счисления с основанием  $m$  записывается в виде последовательности символов:

$$N_m \Rightarrow k_s k_{s-1} \dots k_p \dots k_2 k_1 k_0, \quad (1)$$

где  $k_s$  – старший разряд;  $k_0$  – младший разряд (нулевой разряд)

- Число объектов в любой системе счисления равно сумме всех последовательных произведений символов и их весов:

$$\{N_m\} = k_s m^s + k_{s-1} m^{s-1} + \dots + k_p m^p + \dots + k_2 m^2 + k_1 m^1 + k_0 m^0; \quad (2)$$

после небольшого упрощения получаем формулу:

$$\{N_m\} = k_s m^s + k_{s-1} m^{s-1} + \dots + k_p m^p + \dots + k_2 m^2 + k_1 m + k_0 \quad (3)$$

- Величина  $m^p$  – называется *весом*  $p$ -го разряда;
- Коэффициент  $k_p$  – должен удовлетворять условию:

$$0 \leq k_p < m$$

### *Переход из одной системы счисления в другую*

Переход из любой системы счисления в десятичную осуществляется по формуле:

$$\{N_m\} = k_s m^s + k_{s-1} m^{s-1} + \dots + k_p m^p + \dots + k_2 m^2 + k_1 m + k_0 \quad (4)$$

Рассмотрим перевод числа из двоичной в десятичную систему счисления.

*Пример:*

$$101010101_2 = ?_{10}$$

*Решение:*

$$\begin{aligned} 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 0 \cdot 8 + 1 \cdot 16 + 0 \cdot 32 + 1 \cdot 64 + 0 \cdot 128 + 1 \cdot 256 = \\ 1 + 0 + 4 + 0 + 16 + 0 + 64 + 0 + 256 = 341_{10} \end{aligned}$$

Рассмотрим перевод числа, наоборот, из десятичной системы в двоичную систему. Этот перевод можно осуществить двумя способами.

*Первый способ:*

Обозначим через  $N$  – преобразуемое число,  $K$  – результат преобразования.

1)  $N$  целочисленно делим на 2, остаток  $k_0$  – будет младшим разрядом числа  $K$ ,

2) Результат деления (частное) снова целочисленно делим на 2, остаток  $k_1$  – будет следующим разрядом числа  $K$ ,

3) Шаги данной процедуры повторяются до тех пор, пока частное, полученное в результате очередной операции деления, не станет равным 0. Тогда остаток от последнего деления используется в качестве старшего разряда:

$$K = k_s k_{s-1} \dots k_1 k_0$$

*Пример:*

- 1)  $217_{10} = ?_2$
- 2)  $217 : 2 = 108$  (1) – младший разряд
- 3)  $108 : 2 = 54$  (0)
- 4)  $54 : 2 = 27$  (0)
- 5)  $27 : 2 = 13$  (1)
- 6)  $13 : 2 = 6$  (1)
- 7)  $6 : 2 = 3$  (0)
- 8)  $3 : 2 = 1$  (1)
- 9)  $1 : 2 = 0$  (1) !! - старший разряд
- 10)  $217_{10} = 11011001_2$

Проверка  $1 + 8 + 16 + 64 + 128 = 217$ .

*Второй способ* основан на использовании значения весов. Определяются те разряды, для которых весовые коэффициенты  $k_p$  не равны 0:

- 1) Находим максимальный разряд  $s$ , для которого выполняется:  $m^s \leq N < m^{s+1}$ , тогда  $k_s = 1$  (значит, все  $k_{i>s} = 0$ ).
- 2) Находим разность  $N = N - m^s$ .

3) В соответствие с п.1 определяем следующий ненулевой весовой коэффициент. Проделываем п.1 до тех пор, пока промежуточное  $N$  не станет равным 0.

*Переход из десятичной системы в произвольную систему с основанием « $m$ »*

Обозначим через  $N$  – преобразуемое число,  $K$  – результат преобразования

1)  $N$  целочисленно делим на « $m$ », остаток  $k_0$  – будет младшим разрядом числа  $K$ ,

2) Результат деления (частное) снова целочисленно делим на  $m$ , остаток  $k_1$  – будет следующим разрядом числа  $K$ .

3) Шаги данной процедуры повторяются до тех пор, пока частное, полученное в результате очередной операции деления, не станет равным 0. Тогда остаток от последнего деления используется в качестве старшего разряда:

$$K = K_s K_{s-1} \dots K_1 K_0$$

*Пример:*

- 1)  $151_{10} = ?_8$
- 2)  $151 : 8 = 18 (7)$  – младший разряд
- 3)  $18 : 8 = 2 (2)$
- 4)  $2 : 8 = 0 (2) !!$  – старший разряд
- 5)  $151_{10} = 227_2$

Проверка  $2 \cdot 64 + 2 \cdot 8 + 7 \cdot 1 = 151$

### *Восьмеричная система счисления*

Основание кода  $m = 8$  : 0, 1, 2, 3, 4, 5, 6, 7.

Каждый восьмеричный символ от 0 до 7 может быть представлен 3-разрядным двоичным числом (триадой) (табл. 1.1). В соответствие с приведенной таблицей удобно использовать простое правило перевода двоичных чисел в восьмеричный код: сначала двоичное число (двоичный код) разбивается на *триады*, начиная с младшего разряда (т.е. на отдельные группы из 3-х символов). Затем триады последовательно заменяются соответствующей восьмеричной цифрой. Таким образом образуется восьмеричный код.

Таблица 1.1

Соответствие восьмеричного и двоичного кода

Двоичный код (триада)			Восьмеричный код (восьмеричная цифра)
вес разряда. $2^p$			
4	2	1	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

### Шестнадцатеричная система счисления

Основание кода  $m = 16$  : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Каждый символ от 0 до F может быть представлен 4-х разрядным двоичным числом (тетрадой).

Правило перевода двоичных чисел в шестнадцатеричный код числами: двоичное число (двоичный код) разбивается на *тетрады*, начиная с младшего разряда. Затем тетрады заменяются соответствующим шестнадцатеричным кодом (шестнадцатеричной цифрой).

Таблица 1.2

Соответствие шестнадцатеричного и двоичного кода

Двоичный код				Шестнадцатеричный код (цифра)
Вес разряда				
8	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5

0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

### ***Сложение двоичных чисел***

Рассмотрим сложение двух четырехразрядных чисел  $a$  и  $b$  на следующем примере:

$$a_3a_2a_1a_0 + b_3b_2b_1b_0 = c_4c_3c_2c_1c_0$$

Правила сложения в младшем разряде:

$a_0 + b_0 = c_0$  (при этом возможен перенос 1 в следующий разряд):

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ (перенос 1 в следующий старший разряд)}$$

### ***Правила сложения в произвольном разряде***

В следующих разрядах правила такие же, но необходимо учитывать возможный перенос (обозначен символом «п») из *предыдущего* разряда:

$$0 + 0 + 1 \text{ (п)} = 1$$

$$0 + 1 + 1 \text{ (п)} = 0 \text{ (перенос «1» в следующий разряд)}$$

$$1 + 0 + 1 \text{ (п)} = 0 \text{ (перенос «1» в следующий разряд)}$$

$$1 + 1 + 1 \text{ (п)} = 1 \text{ (перенос «1» в следующий разряд)}$$

### ***Вычитание двоичных чисел***

Вычитание двоичных чисел рассмотрим на следующем примере:

$$a_3a_2a_1a_0 - b_3b_2b_1b_0 = c_3c_2c_1c_0$$

Правила вычитания в младшем разряде:  $a_0 - b_0 = c_0$  (возможен заем из следующего разряда):

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$0 - 1 = 1 \text{ (заем из следующего разряда)}$$

$$1 - 1 = 0$$

### *Правила вычитания в произвольном разряде*

В следующих разрядах правила такие же, но необходимо учитывать возможный заем из *предыдущего* разряда:

$$0 - 0 - 1 \text{ (3)} = 1 \text{ (заем «1» из следующего разряда)}$$

$$0 - 1 - 1 \text{ (3)} = 0 \text{ (заем «1» из следующего разряда)}$$

$$1 - 0 - 1 \text{ (3)} = 0$$

$$1 - 1 - 1 \text{ (3)} = 1 \text{ (заем «1» из следующего разряда)}$$

## **Умножение двоичных чисел**

Правила умножения в одном (единичном) разряде:

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

Умножение многоразрядных чисел:  $A \times B = C$

А (умножимое) –  $i$ -разрядное число.

В (умножитель) –  $k$ -разрядное число.

Операция умножения состоит из  $k$  циклов:

- В каждом цикле множимое умножается на содержимое соответствующего разряда множителя. Получаем частичное произведение.
- Каждое последующее частичное произведение сдвигается на 1 разряд влево. Результат умножения представляет собой сумму частичных произведений.

## **Преобразование дробей**

Процедура преобразования дробей состоит из нескольких этапов.

1. Определяется первый разряд после запятой: дробь умножается на 2; если произведение меньше 1, то записывается «0»; если произведение больше 1, то записывается «1», а из произведения вычитается 1.

Если произведение равно 1, записывается «1» и вычисления прекращаются.

2,3,... Аналогично определяется значение следующих разрядов.

### ***Сложение в системе счисления с произвольным основанием $m = p$***

Если  $a_0 + b_0 \geq p$ , то происходит перенос 1 в следующий разряд:

$$c_0 = a_0 + b_0, \quad \text{если } a_0 + b_0 < p$$

$$c_0 = a_0 + b_0 - p, \quad \text{если } a_0 + b_0 \geq p \text{ с переносом 1}$$

Для произвольного  $i$ -го разряда правила те же, только нужно учитывать возможный перенос из предыдущего разряда.

### ***Вычитание в системе счисления с произвольным основанием $m = p$***

Если  $a_0 < b_0$ , то происходит заем 1 в следующем разряде:

$$c_0 = a_0 - b_0, \quad \text{если } a_0 \geq b_0$$

$$c_0 = a_0 - b_0 + p, \quad \text{если } a_0 < b_0 \text{ с заемом 1 в следующем разряде.}$$

Для произвольного  $i$ -го разряда правила те же, только нужно учитывать возможный заем из предыдущего разряда.

### ***Умножение в системе счисления с произвольным основанием $m = p$***

$$A \times B = C$$

A (множимое) –  $i$ -разрядное число.

B (множитель) –  $k$ -разрядное число.

Операция умножения состоит из  $k$  циклов. В каждом цикле множимое умножается на содержимое соответствующего разряда множителя. Получаем частичное произведение. Каждое последующее частичное произведение сдвигается на 1 разряд влево. Результат умножения представляет собой сумму частичных произведений.

В следующей таблице дан пример таблицы умножения в системе счисления  $m = 8$  (умножение чисел от 1 до 7):

Таблица 1.3

Таблица умножения для системы счисления  $m = 8$ 

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>1</b>	1	2	3	4	5	6	7
<b>2</b>	2	4	6	10	12	14	16
<b>3</b>	3	6	11	14	17	22	25
<b>4</b>	4	10	14	20	24	30	34
<b>5</b>	5	12	17	24	31	36	43
<b>6</b>	6	14	22	30	36	44	52
<b>7</b>	7	16	25	34	43	52	61

### *Положительные и отрицательные числа*

Для указания знака двоичного числа используются старший разряд, который называют *знакомым разрядом*. Значение «0» в этом разряде соответствует знаку «+», «1» соответствует знаку «-».

*Пример.* Двоичное представление числа 5 есть 101. 8-разрядный двоичный код числа +5 записывается как 0000 0101.

### *Способы записи отрицательного двоичного числа*

Кроме указания знака числа, для удобства последующих преобразований с числами в цифровой электронике и целочисленной вычислительной технике широко используется запись отрицательных двоичных чисел не в обычном (прямом) коде, а в специальном коде – это так называемые «обратный код» и «дополнительный код». Рассмотрим эти способы.

*Метод «обратный код».* Запись отрицательного числа этим способом осуществляется путем последовательной замены (инверсии) каждого разряда соответствующего положительного числа: значение «1» меняется на «0», и наоборот, «0» меняется на «1».

*Пример.* Двоичное представление числа 5 есть 101, его 8-разрядное двоичное представление 0000 0101. Соответственно, обратный 8-разрядный двоичный код числа -5 есть 1111 1010.

Таким образом, для преобразования отрицательного числа в положительное применяется простая операция инвертирования. Этим обратные коды удобны в применении. В качестве недостатка следует отметить, что в обратных двоичных кодах имеются два кода записи числа 0:

«положительный нуль» 0000 0000 и «отрицательный нуль» 1111 1111 (приведены 8-разрядные обратные коды). Это приводит к некоторой неопределенности и усложнению операции суммирования. Поэтому в дальнейшем для записи знаковых целых чисел перешли к другому способу – дополнительным кодам.

Метод «*дополнительный код*» («двойное дополнение», «точное дополнение»). В этом методе сначала вычисляется обратный код числа. Затем к обратному коду прибавляется 1 (см. таблицу ниже).

Дополнительный код позволяет упростить арифметические операции над двоичными числами без дополнительных преобразований.

Таблица 1.4  
Соответствие десятичного представления числа и различных кодов записи двоичного представления.

Десятичное представление	Двоичное представление (8 бит)		
	прямой код	обратный код	дополнительный код
127	0111 1111	0111 1111	0111 1111
---	---	---	---
1	0000 0001	0000 0001	0000 0001
0	0000 0000	0000 0000	0000 0000
-0	1000 0000	1111 1111	0000 0000
-1	1000 0001	1111 1110	1111 1111
-2	1000 0010	1111 1101	1111 1110
-3	1000 0011	1111 1100	1111 1101
-4	1000 0100	1111 1011	1111 1100
-5	1000 0101	1111 1010	1111 1011
-6	1000 0110	1111 1001	1111 1010
-7	1000 0111	1111 1000	1111 1001
-8	1000 1000	1111 0111	1111 1000
-9	1000 1001	1111 0110	1111 0111
-10	1000 1010	1111 0101	1111 0110
-11	1000 1011	1111 0100	1111 0101
-127	1111 1111	1000 0000	1000 0001
-128	---	---	1000 0000

## **Контрольные вопросы**

1. Назовите основные системы счисления, дошедшие до наших времен с древности.
2. В чем отличие между позиционными и непозиционными системами счисления?
3. Что такое основание системы счисления?
4. Что такое вес определённого разряда числа?
5. Как осуществляется переход из десятичной системы в двоичную и наоборот?
6. Сколько символов в восьмеричной системе и сколько символов в шестнадцатеричной?
7. Что такое триада, тетрада?
8. Назовите правила сложения двух чисел для единичного разряда.
9. Назовите правила вычитания двух чисел для единичного разряда.
10. Назовите правила умножения двух чисел для единичного разряда.
11. Приведите правило арифметических операций с многоразрядными числами.
12. Назовите способы записи отрицательного числа. Поясните термины «прямой код», «обратный код», «дополнительный код». В чем назначение и удобство использования «обратного кода» по сравнению с прямым кодом и «дополнительного кода» по сравнению с «обратным кодом»?

## 2. ЛОГИЧЕСКИЕ ФУНКЦИИ

При изучении логических сигналов проявляется много общего с элементами формальной логики.

Формальная логика – наука о законах и формах человеческого мышления, она оперирует с событиями (высказываниями) вне зависимости от их содержания, учитывая только их истинность или ложность.

Рассмотрим элементы формальной логики на простейших примерах. Рассмотрим две основные логические конструкции.

- Если будет хорошая погода и со мной будут друзья, то я поеду на дачу;
- Если будет хорошая погода или со мной будут друзья, то я поеду на дачу.

Наличие или отсутствие логических сигналов и порождающие их условия, как правило, связаны подобными логическими конструкциями.

### *Простые и сложные высказывания*

Простые высказывания (события) содержат только один факт, не зависящий от других фактов. Сложное высказывание (событие) содержит несколько простых фактов, связанных логическими связями.

Так, в приведенных выше примерах два простых высказывания связаны логическими функциями «И» и «ИЛИ», образуя сложное высказывание (событие):

- Если *будет хорошая погода (1) и со мной будут друзья (2)*, то я поеду на дачу.
- Если *будет хорошая погода (1) или со мной будут друзья (2)*, то я поеду на дачу.

Наличие или отсутствие логических сигналов и порождающие их условия, как правило, связаны подобными логическими конструкциями. В формальной логике: ограничено число связей, допускаются только строго определенные толкования этих связей. Это позволяет однозначно представлять сложные события совокупностью простых. Введение символов, обозначающих связи между простыми и сложными событиями, позволило решать логические задачи математическими средствами булевой алгебры.

В положительной логике булевой алгебры истинному высказыванию (или наступлению события) приписывается логическая «1», а ложному

высказыванию (не наступлению события) приписывается логический «0». В отрицательной логике будет обратное соответствие.

Простые события (переменные) будем обозначать через  $x_i$ . Переменная  $x_i$  может принимать всего два значения: 0 и 1. Сложные события (функции от простых событий) будем обозначать через  $y$ . Функция  $y$  также может принимать всего два значения: 0 и 1.

### **Логическая функция ИЛИ (дизъюнкция)**

Дизъюнкция является логической функцией  $y$  преобразования переменных ( $x_1, \dots, x_n$ ) и фиксирует наступление сложного события ( $y = 1$ ), если наступило хотя бы одно простое событие ( $x_1 = 1$  или  $x_2 = 1$ , или  $x_3 = 1$  и т.д.).

Запись функции «дизъюнкция» в булевой алгебре:

$$y = x_1 \vee x_2 \vee \dots \vee x_n, \text{ где символ } \vee \text{ – обозначает знак дизъюнкции.}$$

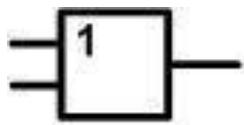
Дизъюнкцию также называют *логическим сложением* и поэтому наряду с записью алгебры логики используют и символьную запись, соответствующей обычной алгебре:

$$y = x_1 + x_2 + \dots + x_n$$

### **Обозначение логической функции ИЛИ**

Логический элемент – это базовый элемент цифровой схемы («вентиль»), выполняющий соответствующую элементарную логическую функцию.

Дизъюнктор, логический элемент «ИЛИ» обозначается на структурно-функциональных схемах по ГОСТ 2.743—72 следующим образом:



В зарубежной литературе используется значок OR и обозначение:



## *Таблица истинности и обозначение логической функции ИЛИ*

В таблице истинности («переключательная таблица») указаны все возможные комбинации сигналов на входах и соответствующие сигналы на выходе функции. Таблица истинности дает исчерпывающую характеристику работы логического элемента, т.е. полностью описывает его логическую функцию.

Таблица истинности элемента ИЛИ :

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	1

## *Логическая функция И (конъюнкция)*

Конъюнкция является логической функцией  $y$  преобразования переменных ( $x_1, \dots, x_n$ ) и фиксирует наступление сложного события ( $y = 1$ ), если имеет место наступления *всех* простых событий ( $x_1 = 1$  и  $x_2 = 1$ , и  $x_3 = 1, \dots$ ).

Если хотя бы одно из простых событий не наступило, то сложное событие не наступит – функция равна  $y = 0$ .

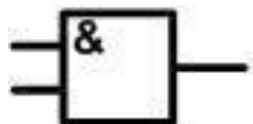
Запись функции «конъюнкция» в булевой алгебре:

$$y = x_1 \wedge x_2 \wedge \dots \wedge x_n \quad \text{где символ } \wedge \text{ – знак обозначает конъюнкцию.}$$

Конъюнкцию также называют *логическим умножением*. Символьная запись, соответствующая обычной алгебре:

$$y = x_1 \cdot x_2 \cdot \dots \cdot x_n$$

Обозначения на схемах. Конъюнктор, логический элемент «И»



В зарубежной литературе AND :



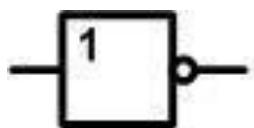
Таблица истинности элемента И:

X1	X2	y
0	0	0
0	1	0
1	0	0
1	1	1

### ***Логическое отрицание (инверсия)***

Логическая функция «НЕ» обеспечивает на выходе логический сигнал, который противоположен входному сигналу. Поэтому другое название – инвертор.

HE



NOT

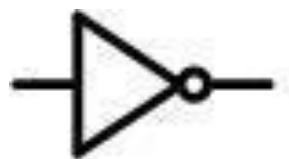


Таблица истинности

X	Y
0	1
1	0

### ***Логический базис***

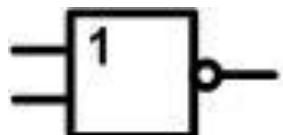
В алгебре логики доказано, что любое сложное логическое высказывание можно исчерпывающе описать, используя комбинацию всего трех логических функций: сложение (дизъюнкция), умножение (конъюнкция) и отрицание (инверсия). Этот набор логических функций называется логическим *базисом*. Отсюда следует, что любую логическую схему можно реализовать, используя комбинацию всего трех логических элементов: дизъюнктор, конъюнктор и инвертор.

Помимо этих трех логических функций, представляющих базис, в общем случае можно рассмотреть 16 логических функций. Наибольший практический интерес представляет реализация логических функций: ИЛИ-

НЕ, И-НЕ, Исключающее ИЛИ. Далее представлены логические элементы, реализующие эти функции.

### ***Логический элемент «ИЛИ-НЕ»***

Обозначение на схемах: ИЛИ-НЕ.



В зарубежной литературе: NOR

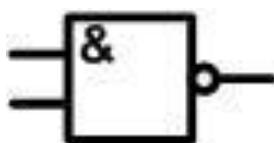


Таблица истинности элемента ИЛИ-НЕ

$x_1$	$x_2$	$y$
0	0	1
0	1	0
1	0	0
1	1	0

### ***Логический элемент «И-НЕ»***

Обозначение на схемах логического элемента И-НЕ:



В зарубежной литературе: NAND



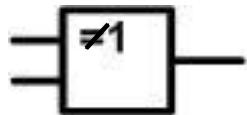
Таблица истинности элемента И-НЕ:

X1	X2	y
0	0	1
0	1	1
1	0	1

1	1	0
---	---	---

### Логический элемент «Исключающее ИЛИ»

Обозначение на схемах: Исключающее ИЛИ



- В зарубежной литературе: XOR

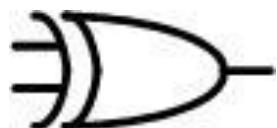


Таблица истинности элемента Исключающее ИЛИ

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0

Символьное представление:  $y = x_1 \oplus x_2$

### Контрольные вопросы

1. Поясните, что такое таблица истинности.
2. Запишите таблицу истинности дизъюнктора.
3. Запишите таблицу истинности конъюнктора.
4. Запишите таблицу истинности инвертора.
5. Какие логические элементы составляют базис?
6. Запишите таблицу истинности элемента Исключающее ИЛИ.

### 3. КОМБИНАЦИОННЫЕ ЛОГИЧЕСКИЕ СХЕМЫ

Комбинационной логической схемой называется устройство, у которого логические сигналы на выходе связаны с входными сигналами *так же*, как значения соответствующей этой схеме *логической функции* связаны со значением ее аргументов:

$$y = f(x_1, x_2, x_3, \dots, x_n)$$

Комбинационные логические схемы представляют в виде:

1. Математического алгебраического выражения (булево выражение).
2. В виде таблицы истинности (в ней указаны все возможные комбинации сигналов на входах и соответствующие сигналы на выходе).
3. В виде структурной электронной схемы (которая визуально показывает функциональную связь между логическими элементами).

Алгебраическое выражение (булево выражение) представляют в виде:

1. Дизъюнкции (суммы)

$$y = z_1 + z_2 + \dots + z_n$$

2. Конъюнкции (произведения)

$$y = z_1 \cdot z_2 \cdot z_3 \cdot \dots \cdot z_n$$

#### *ДНФ – Дизъюнктивная нормальная форма*

Дизъюнктивная нормальная форма – это функция, представляющая собой дизъюнкцию (сумму), у которой каждое слагаемое представляет собой конъюнкцию (логическое произведение) входных переменных или их дополнений (инверсий)

$$y = z_1 + z_2 + \dots + z_n$$

Каждый  $z_i$  – это произведение входных переменных или их дополнений.

*Правила построения комбинационной логической схемы:*

1. Формируем входные шины ( $x_1, x_2, x_3 \dots$ ).
- 1а. Формируем дополнения входных переменных ( $x_1, x_2, x_3 \dots$ ).
2. Формируем каждый входной элемент схемы «ИЛИ».

ДНФ в общем случае избыточна, существует минимальная форма, к ней можно свести ДНФ, при этом осуществляется минимизация электронной схемы.

## *КНФ*

Конъюнктивная нормальная форма – это функция, представляющая собой произведение, каждый член которого представляет собой сумму входных переменных или их дополнений. КНФ в общем случае избыточна, существует минимальная форма, к ней можно свести КНФ, при этом осуществляется минимизация электронной схемы.

*Правила построения комбинационной логической схемы:*

1. Формируем входные шины ( $x_1, x_2, x_3 \dots$ ).
- 1а. Формируем дополнения входных переменных.
2. Формируем каждый входной элемент схемы «И». Ими являются выходы элементов «ИЛИ», на входы которых подаются входные сигналы.

## *Упрощение комбинационных логических схем*

Исходную схему можно упростить:

- A. На основе правил булевой алгебры.
- B. На основе таблицы истинности.
- C. С помощью карты Карно.

Важно уметь осуществлять преобразование булева выражения в таблицу истинности и, наоборот, уметь по таблице истинности записывать соответствующее булево выражение.

*Преобразование булева выражения в таблицу истинности:*

1. Записываем в таблицу истинности все возможные комбинации входных сигналов.
2. В соответствие с заданной функцией вычисляем выходной сигнал.

*Преобразование таблицы истинности в булево выражение:*

1. Записываем таблицу истинности.
2. Выбираем строки, для которых функция равна «1» .
3. Записываем выражение для этой строки в виде конъюнкции. Причем переменная, равная «1», записывается в прямом виде. Переменная, равная «0», записывается в инверсном виде.
4. Булево выражение записывается как дизъюнкция выбранных строк.

Таблица 3.1

Пример некоторой таблицы истинности

$x_3$	$x_2$	$x_1$	$y$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Для таблицы истинности, представленной выше, булево выражение будет иметь вид:

$$y = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3.$$

### ***Карты Карно (диаграммы Вейча)***

Карты Карно предназначены для решения задачи минимизации булевых выражений.

Обозначим через  $n$  – число переменных в булевом выражении. Карты Карно представляют собой таблицу, состоящую из  $2^n$  клеток. В карте Карно столько клеток, сколько комбинаций можно составить из прямых и инверсных значений переменных. При  $n = 2$  будет 4 клетки, при  $n = 3$  будет 8 клеток, при  $n = 4$  будет 16 клеток и т.д. Карты Карно построены так, что в ее соседние клетки попадают смежные члены функции – члены, отличающиеся значением формы переменной: в один член эта переменная входит в прямой форме, а в другой эта переменная входит в инверсной форме. Благодаря этому получается наглядное представление о различных вариантах «склеивания» различных членов.

### ***Процедура упрощения булева выражения.***

1. Записывается булево выражение в дизъюнктивной форме конъюнкций.
2. Заносится 1 в те клетки, для которых имеются соответствующие сомножители.

3. Объединяются (соседние) единички прямоугольными контурами, охватывающими или 2, или 4, или 8, или 16 (и т.д) клеток.

4. Проводятся упрощения, путем исключения членов, дополняющих друг друга внутри контура.

Оставшиеся члены записываются в дизъюнктивной форме. Одна и та же «1» может входить в разные контуры, так как функция не меняется, если добавляется уже рассмотренный член, так как справедливо  $X+X=X$ .

*Упражнение:* Используя карты Карно, упростить следующее логическое выражение:

$$y = \overline{x_1}x_2 + x_1\overline{x_2} + x_1x_2$$

Таблица 3.2

Карта Карно для двух переменных

	$x_2$	$\overline{x_2}$
$x_1$	1	1
$\overline{x_1}$	1	

*Решение:* В горизонтальном контуре «склеивается» переменная  $x_2$

В вертикальном контуре «склеивается» переменная  $x_1$

В итоге имеем минимальную форму:

$$y = x_1 + x_2$$

**Карты Карно с тремя переменными (8 комбинаций входных переменных)**

Пример возможной реализации Карты Карно с тремя переменными разберем на следующей задаче. Требуется упростить логическое выражение:

$$y = \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1\overline{x_2}\overline{x_3}$$

Таблица 3.3

Карта Карно для трех переменных

	$x_3$	$\overline{x_3}$
$x_1\overline{x_2}$	1	1
$x_1x_2$		1
$\overline{x_1}x_2$		1
$\overline{x_1}\overline{x_2}$		

*Решение:* В горизонтальном контуре «склеивается» переменная  $x_3$  и остается  $x_1\bar{x}_2$ , в вертикальном контуре склеивается переменная  $x_1$  и остается  $x_2\bar{x}_3$ . Минимальная форма будет:  $y = x_1\bar{x}_2 + x_2\bar{x}_3$

### **Карты Карно с четырьмя переменными (16 комбинаций входных переменных)**

Пример возможной реализации карты Карно с четырьмя переменными разберем на следующем примере. Требуется упростить следующее логическое выражение:

$$y = x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2x_3x_4 + x_1\bar{x}_2\bar{x}_3x_4$$

Таблица 3.4.

Карта Карно для четырех переменных

	$\bar{x}_3\bar{x}_4$	$\bar{x}_3x$	$x_3\bar{x}_4$	$x_3\bar{x}_4$
$\bar{x}_1\bar{x}_2$		1	1	
$\bar{x}_1x_2$		1	1	
$x_1\bar{x}_2$				
$x_1\bar{x}_2$	1	1		

*Решение:* В верхнем контуре «склеивается» переменная  $x_2$  и  $x_3$ , остается  $\bar{x}_1x_4$ , в нижнем контуре «склеивается» переменная  $x_4$  и остается  $x_1\bar{x}_2\bar{x}_3$ . Минимальная форма будет равна:

$$y = \bar{x}_1x_4 + x_1\bar{x}_2\bar{x}_3.$$

### **Карты Карно. Нестандартные ситуации.**

Случай вертикальной симметрии. В картах Карно возможны случаи, когда имеются контуры, расположенные на границе карты, тогда карту нужно свернуть в вертикальный цилиндр, в котором левый край совмещается с правым.

Случай горизонтальной симметрии. В картах Карно возможны случаи, когда имеются контуры, расположенные на границе карты, тогда карту нужно свернуть в горизонтальный цилиндр, в котором верхний край совмещается с нижним.

Рассмотрим еще одну часто встречающуюся ситуацию. В картах Карно возможны случаи, когда 4 угловых ячейки карты рассматриваются как связанные друг с другом, тогда карту нужно свернуть в шар.

## **Контрольные вопросы**

1. Чем отличаются простые высказывания (события) от сложных высказываний (событий)?
2. Приведите определения логических функций ИЛИ, И, НЕ.
3. Приведите математическую запись и обозначения на схемах логических элементов: дизъюнктор, конъюнктор, инвертор.
4. Приведите таблицу истинности логической функции ИЛИ.
5. Приведите таблицу истинности логической функции И.
6. Приведите таблицу истинности логической функции НЕ.
7. Что такое и какой набор логических функций называют базисом?
8. Что такое дизъюнктивная нормальная форма записи?
9. Что такое конъюнктивная нормальная форма записи?
10. Что такое комбинационная логическая схема?
11. Как преобразовать таблицу истинности в булево выражение?
12. Опишите процедуру упрощения булева выражения с помощью карт Карно.

## 4. КОДИРОВАНИЕ

Кодирование – в широком смысле слова это правило отображения символов одного множества через символы другого множества. В узком смысле это правило отображения символов одного кода в символы другого кода.

Кодер (шифратор) – это устройство, реализующее процесс кодирования. При этом должна существовать возможность однозначного декодирования, при которой можно получить исходные символы из кодированных.

Определение кода производится с помощью таблицы истинности и булева выражения.

Слово – это кодовая комбинация нескольких символов кода. В общем случае длина слова может быть произвольной. Ограничимся технически важным случаем – все слова данного кода имеют одну длину –  $n$ .

Для кода с основанием  $m$  (основание кода – число символов) возможное число слов будет равно  $m^n$ .

Коды разделяются на два основных класса: Непомехозащищенные коды и помехозащищенные коды.

Непомехозащищенные коды – в которых искажение какого-либо разряда кодовой комбинации невозможно обнаружить. При передаче данных используются все возможные кодовые комбинации.

Помехозащищенные коды имеют в своей структуре служебные символы, при передаче данных задействована только часть возможных кодовых комбинаций. Другая часть кодовых комбинаций используется для обнаружения и исправления ошибок.

Рассмотрим некоторые простейшие коды.

### Двоичный код с весом 8.4.2.1.

Таблица 4.1

Таблица соответствия десятичного кода и кода с весом 8.4.2.1.

$m = 10$	$m = 8$	$m = 4$	$m = 2$	$m = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

В коде с весом 8.4.2.1. каждый разряд десятичного числа записывается в виде комбинации двоичного кода с весом 8.4.2.1. Например:  $9397_{10} = 1001001110010111_{8.4.2.1}$ .

### ***Самодополняющийся код***

Самодополняющийся код – это такой код, в котором инвертированный код всегда дополняет основной до числа 9. Из самодополняющихся кодов выделим:

1. Код с весовыми коэффициентами 2.4.2.1.
3. Код с избытком 3.

### ***Код с весовыми коэффициентами 2.4.2.1.***

*Таблица 4.2*

Таблица соответствия десятичного кода и кода с весом 8.4.2.1.

$m = 10$	$m = 2$	$m = 4$	$m = 2$	$m = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

Особенностью данного кода является то, что наибольшее число содержит максимальное число единиц. В коде с весом 2.4.2.1. каждый разряд

десятичного числа записывается в виде комбинации двоичного кода с весом 2.4.2.1. Например:  $9397_{10} = 1111\ 0011\ 1111\ 1101$  <sub>2.4.2.1.</sub>

### *Код с избытком 3.*

*Таблица 4.3*

Таблица соответствия десятичного кода и кода с избытком 3

$m = 10$	$m = 8$	$m = 4$	$m = 2$	$m = 1$
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Чтобы представить десятичное число в коде с избытком 3, нужно: 1) прибавить 3 к каждой цифре десятичного числа; 2) заменить полученные цифры соответствующим четырехразрядным двоичным кодом. Например,  $459 = 0111\ 1000\ 1100$ . Особенностью кода является то, что каждая комбинация, представляющая десятичное число, содержит обязательно как единицы, так и нули.

### *Код Грея*

Код Грея, рефлексный двоичный код, в котором два соседних значения различаются только в одном двоичном разряде.

Изначально предназначался для защиты от ложного срабатывания электромеханических переключателей. В настоящее время код Грея широко используется для упрощения выявления и исправления ошибок в системах связи.

В коде Грея две соседние комбинации отличаются одна от другой только в одном разряде. Уменьшается ошибка считывания неправильного результата.

## **Кодер Грэя**

Рассмотрим на примере трехразрядного кодера.

*Таблица 4.4*

Таблица соответствия кода Грэя и обычного двоичного кода.

N	Обычный двоичный код			Код Грэя		
	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

С учетом таблицы соответствия записываем булево выражение для  $y_0$ ,  $y_1$ ,  $y_2$  (для тех строк, где стоит 1 для  $y_0$ ,  $y_1$ ,  $y_2$ ):

$$y_0 = \overline{x_2}x_1x_0 + \overline{x_2}x_1\overline{x_0} + x_2\overline{x_1}x_0 + x_2x_1\overline{x_0}$$

$$y_1 = \overline{x_2}x_1\overline{x_0} + \overline{x_2}x_1x_0 + x_2\overline{x_1}\overline{x_0} + x_2\overline{x_1}x_0$$

$$y_2 = x_2\overline{x_1}\overline{x_0} + x_2\overline{x_1}x_0 + x_2x_1\overline{x_0} + x_2x_1x_0$$

После упрощения минимальная форма кодера Грэя примет следующий вид:

$$y_0 = x_0 \oplus x_1$$

$$y_1 = x_1 \oplus x_2$$

$$y_2 = x_2$$

Схемная реализация 4-разрядного кодера Грэя на элементах «исключающее ИЛИ» приведена на рис. 4.1.

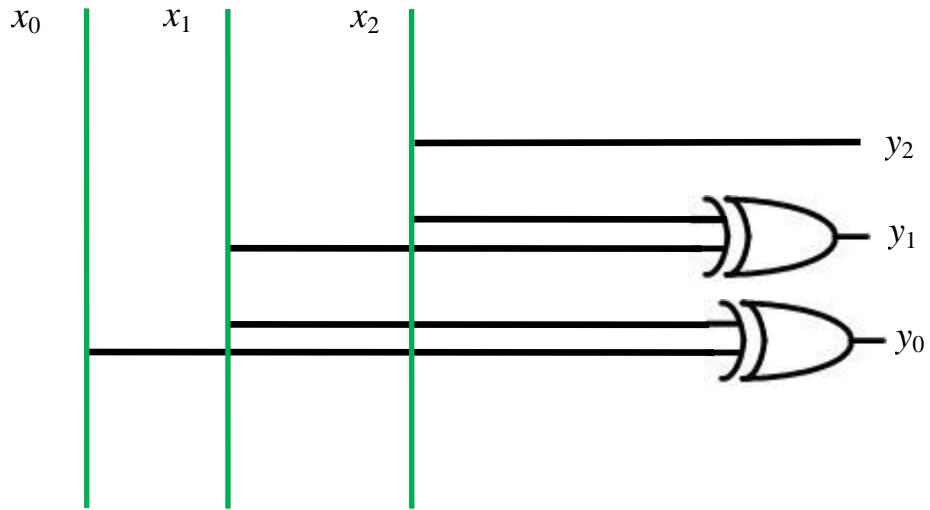


Рис. 4.1. Реализация 4-разрядного кодера Грея

Общее правило перехода из двоичного кода в код Грея:  
 ||| двоичный код преобразуется в код Грея путем поразрядного сложения  
 ||| по модулю 2 исходной кодовой комбинации и сдвинутой на один разряд  
 ||| вправо.

### *Декодер Грея*

Декодер Грея осуществляет переход из двоичного кода Грея ( $x$ ) в обычный двоичный код ( $y$ ). Таблица истинности трехразрядного декодера:

*Таблица 4.5*

Таблица соответствия перехода из кода Грея в обычный двоичный код.

N	Обычный двоичный код			Код Грея		
	$x_2$	$x_1$	$x_0$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

Записываем Булево выражение для декодера (для тех строк, где стоит 1 для  $y_0, y_1, y_2$ ):

$$y_0 = \overline{x_2}x_1x_0 + \overline{x_2}x_1\overline{x_0} + x_2x_1x_0 + x_2\overline{x_1}\overline{x_0}$$

$$y_1 = \overline{x_2}x_1x_0 + \overline{x_2}x_1\overline{x_0} + x_2\overline{x_1}x_0 + x_2\overline{x_1}\overline{x_0}$$

$$y_2 = x_2x_1\overline{x_0} + x_2x_1x_0 + x_2\overline{x_1}x_0 + x_2\overline{x_1}\overline{x_0}$$

После упрощения минимальная форма декодера принимает вид:

$$y_0 = x_2 \oplus x_1 \oplus x_0$$

$$y_1 = x_2 \oplus x_1$$

$$y_2 = x_2$$

*Правило перехода от кода Грэя к двоичному коду: старший разряд остается без изменения, каждый младший разряд представляет собой сумму по модулю 2 всех старших разрядов, включая данный.*

### **Кодер**

Кодер (шифратор) – это логическое устройство, переводящее десятичный код в двоичный. Обозначение:

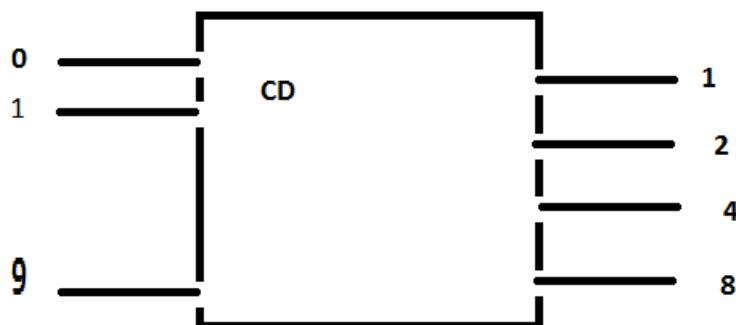


Рис. 4.2. Обозначение кодера (шифратора)

Таблица 4.6

Соответствие десятичного ( $x$ ) и двоичного ( $y$ ) кода для кодера:

$xi$	$y3(8)$	$y2(4)$	$y1(2)$	$y(0)$
$x_0$	0	0	0	0
$x_1$	0	0	0	1
$x_2$	0	0	1	0
$x_3$	0	0	1	1
$x_4$	0	1	0	0
$x_5$	0	1	0	1
$x_6$	0	1	1	0
$x_7$	0	1	1	1
$x_8$	1	0	0	0
$x_9$	1	0	0	1

Булево выражение для кодера будет:

- $y_0 = x_1 + x_3 + x_5 + x_7 + x_9$
- $y_1 = x_2 + x_3 + x_6 + x_7$
- $y_2 = x_4 + x_5 + x_6 + x_7$
- $y_3 = x_8 + x_9$

Схема кодера реализуется на дизъюнкторах.

### Декодер (десифратор)

Декодер, или десифратор – это логическое устройство, переводящее двоичный код в десятичный. Обозначение:

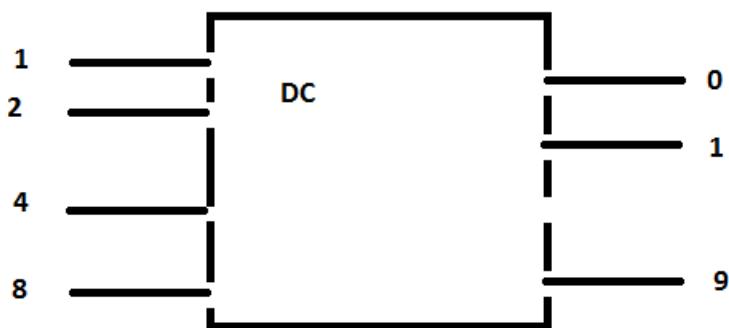


Рис. 4.3. Обозначение декодера (десифратора)

Таблица 4.7

Соответствие двоичного ( $x$ ) и десятичного ( $y$ ) кода  
для декодера:

$x_3(8)$	$x_2(4)$	$x_1(2)$	$x_0(1)$	$xi$
0	0	0	0	$y_0$
0	0	0	1	$y_1$
0	0	1	0	$y_2$
0	0	1	1	$y_3$
0	1	0	0	$y_4$
0	1	0	1	$y_5$
0	1	1	0	$y_6$
0	1	1	1	$y_7$
1	0	0	0	$y_8$
1	0	0	1	$y_9$

Булево выражение для декодера будет:

$$y_0 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0}$$

$$y_1 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0$$

.....

$$y_9 = x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0$$

Схема декодера реализуется на конъюнкторах.

### Контрольные вопросы

1. Что есть процесс кодирования информации?
2. Что такое «не помехозащищённый код»?
3. Что такое «помехозащищенный код»?
4. Приведите таблицу истинности кода с весовыми коэффициентами  
2.4.2.1.
5. Приведите таблицу истинности кода с весовыми коэффициентами  
4.2.2.1.
6. Приведите таблицу истинности кода с избытком 3.
7. В чем отличительная особенность кода Грэя?
8. Нарисуйте схему 4-разрядного кода Грэя.
9. Что такое Кодер? Приведите его обозначение на схеме.
- 10.Что такое Декодер? Приведите его обозначение на схеме.

## 5. ПОСЛЕДОВАТЕЛЬНЫЕ ЛОГИЧЕСКИЕ СХЕМЫ. ТРИГГЕРЫ

*Комбинационными логическими схемами* называются устройства, у которых логические сигналы на выходе связаны с входными сигналами также, как связаны значения логической функции со значениями ее аргументов. Сигнал на выходе вырабатывается сразу после подачи сигналов на вход, в тот же момент времени.

*Последовательные логические схемы* (последовательные цифровые устройства, цифровые автоматы) – в отличие от комбинационных логических схем имеют некоторое число различных внутренних состояний. Под воздействием входных сигналов последовательная логическая схема:

- 1) Переходит из одного состояния в другое;
- 2) Вырабатывает выходной сигнал  $y$ .

Выходной сигнал последовательной логической схемы (цифрового автомата) в конкретный момент времени определяется:

- входной последовательностью, поступившей в этот момент времени;
- внутренним состоянием цифрового автомата, которое является результатом действия на автомат входных сигналов, поступивших в предыдущие моменты времени.

Рассмотрим последовательные логические схемы на примере триггера.

*Триггер* – это цифровое логическое устройство, которое может неограниченно долго находиться в одном из двух устойчивых состояний («1» или «0») и скачкообразно переключаться из одного состояния в другое под воздействием внешнего сигнала.

В триггере выходы должны находиться в противоположных (комplementарных) состояниях. При несоблюдении этого условия триггер теряет триггерное свойство.

### *Классификация триггеров*

В зависимости от *способа управления* различают:

- RS-триггер с двумя управляемыми входами
- D-триггер с одним управляемым входом
- JK-триггер с двумя управляемыми входами, не имеет запрещенных состояний;
- T-триггер с одним счетным входом.

По способу записи информации триггеры подразделяются на асинхронные и синхронные.

Асинхронный (нетактируемый) триггер имеет только управляющие входы. Переключается сразу после поступления сигнала на один из входов.

Синхронный (тактируемый) триггер имеет управляющие и синхронизирующие (тактирующие) входы. Триггер переключается (при поступлении сигналов на информационные входы) после поступления синхросигнала. При отсутствии синхроимпульса триггер не переключается.

Триггер с прямыми входами переключается по (от) логической «1». Триггер с инверсными входами переключается по (от) логическому «0».

Работа триггеров описывается с помощью таблицы истинности и временной диаграммы. Временная диаграмма показывает уровни напряжения и временные интервалы для входных и выходных сигналов. Она соответствует той картине, которую бы мы наблюдали на экране осциллографа.

#### *Асинхронный RS-триггер с прямыми входами*

Триггер имеет два раздельных информационных входа:

S – от английского *set* – устанавливать.

R – *reset* – возвращать в исходное состояние.

В триггере выходы должны находиться в противоположных (комплементарных) состояниях.

$Q$  – прямой выход;

$\bar{Q}$  – инверсный выход.

Триггер реализуется на элементах ИЛИ-НЕ. В структуре двух элементов ИЛИ-НЕ создадим положительную обратную связь – входы одного элемента соединим с выходами другого (рис. 5.1).

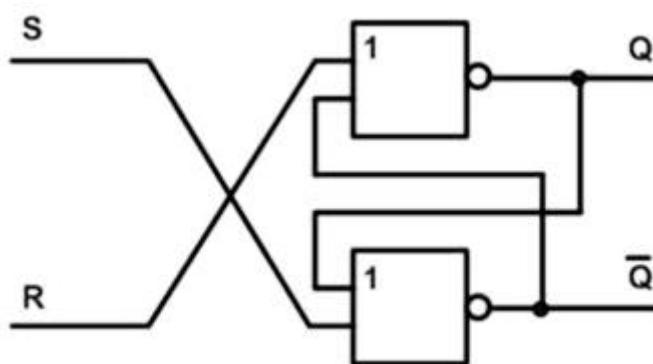


Рис. 5.1. Схема RS-триггера

Важное свойство элемента «ИЛИ-НЕ», что при подаче логической «1» на любой его вход на выходе будет всегда логический «0».

Рассмотрим все возможные комбинации на входе RS-триггера. Это соответственно состояния:

$$S = 1, R = 0$$

$$S = 0, R = 1$$

$$S = 0, R = 0$$

$$S = 1, R = 1$$

1. При  $S=1, R=0$  на инверсном выходе триггера будет логическая «0», а на верхнем элементе «ИЛИ-НЕ» на оба входа будет подан логический «0», что даст на прямом выходе триггера логическую «1».

$$Q = 1, \bar{Q} = 0 - \text{установка логической «1»}$$

2.  $S = 0, R = 1$  тогда на прямом выходе триггера будет логический «0», а на нижнем элементе «ИЛИ-НЕ» на оба входа будет подан логический «0», что даст на инверсном выходе триггера логическую «1».

$$Q = 0, \bar{Q} = 1 - \text{установка «0», сброс.}$$

3.  $S = 0, R = 0$ , тогда состояние триггера будет зависеть от его предыдущего состояния в момент времени  $t-1$ .

3.1. Допустим, его предыдущее состояние было

$$Q(t-1) = 1, \bar{Q}(t-1) = 0.$$

Тогда на оба входа верхнего элемента «ИЛИ-НЕ» будут поданы логические «0», и на прямом выходе триггера будет логическая «1». На инверсном выходе триггера будет «0».

3.2. ( $S = 0, R = 0$ )

Допустим, его предыдущее состояние было  $Q(t-1) = 0, \bar{Q}(t-1) = 1$ . Тогда на оба входа нижнего элемента «ИЛИ-НЕ» будут поданы логические «0», и на инверсном выходе триггера будет «1». На прямом выходе будет «0». Таким образом, при  $S = 0, R = 0$ , в обоих случаях, состояние выходов триггера не меняется – реализуется режим хранения.

4.  $S = 1, R = 1$ , тогда на обоих выходах установится логический «0». Триггер теряет триггерное свойство. Это состояние является неустойчивым, называется запрещенным состоянием и не используется.

Таблица 5.1.

Таблица истинности RS-триггера с прямыми входами

$S$	$R$	$Q(t)$	$\bar{Q}(t)$	Функция
1	0	1	0	Установка «1»
0	1	0	1	Сброс
0	0	$Q(t-1)$	$\bar{Q}(t-1)$	Режим хранения
1	1			Запрещенное состояние

### Асинхронный RS-триггер с инверсными входами

Триггер с инверсными входами переключается по логическому «0». Триггер реализуется на элементах И-НЕ. В структуре двух элементов И-НЕ создадим положительную обратную связь – входы одного элемента соединим с выходами другого.

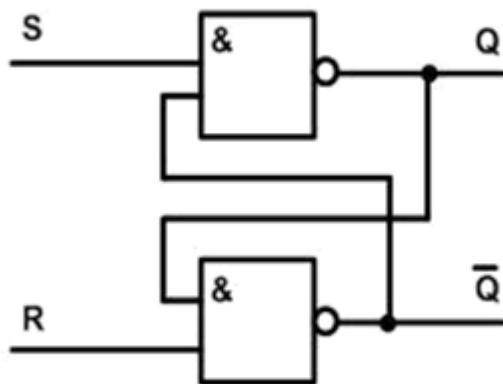


Рис. 5.2. Структурная схема асинхронного RS-триггера

Отметим важное свойство элемента «И-НЕ»: при подаче логической «0» на любой его вход на выходе всегда будет логическая «1». Рассмотрим все возможные комбинации на входе RS-триггера.

1.  $S = 1, R = 0$
2.  $S = 0, R = 1$
3.  $S = 1, R = 1$
4.  $S = 0, R = 0$

1.  $S = 1, R = 0$ . Тогда на инверсном выходе триггера будет логическая «1», а на верхнем элементе «И-НЕ» на оба входа будет подана логическая «1» (во втором случае – снизу за счет обратной связи), что даст на прямом выходе триггера логический «0».

$Q = 0, \bar{Q} = 1$  – сброс

2.  $S = 0, R = 1$ . Тогда на инверсном выходе триггера будет логический «0», а на верхнем элементе «И-НЕ» на оба входа будет подан логический «0», что даст на прямом выходе триггера логический «1».

$Q = 1, \bar{Q} = 0$  – установка «1»

3.  $S = 1, R = 1$ . В этом случае состояние триггера будет зависеть от его предыдущего состояния.

3.1. Допустим, его предыдущее состояние было  $Q = 1 \quad \bar{Q} = 0$

Тогда на оба входа нижнего элемента «И-НЕ» будут поданы логические «1», и на инверсном выходе триггера будет логический «0». На прямом выходе триггера будет «1»

3.2. Допустим, его предыдущее состояние было

$Q = 0, \bar{Q} = 1$

Тогда на оба входа верхнего элемента «И-НЕ» будут поданы логические «1», и на прямом выходе триггера будет «0». На инверсном выходе будет «1».

Таким образом, при  $S = 1, R = 1$ , в обоих случаях, состояние триггера не меняется и реализуется режим хранения.

4.  $S = 0, R = 0$ . Тогда на обоих выходах установится логическая «1». Триггер теряет триггерное свойство. Это состояние является неустойчивым, называется запрещенным состоянием и не используется.

Таблица 5.2.

Таблица истинности RS-триггера с инверсными входами

$S$	$R$	$Q(t)$	$\bar{Q}(t)$	Функция
1	0	0	1	Сброс
0	1	1	0	Установка «1»
1	1	$Q(t-1)$	$\bar{Q}(t-1)$	Режим хранения
0	0			Запрещенное состояние

## **Синтез асинхронного RS-триггера с прямыми входами**

Решение задачи синтеза логических элементов рассмотрим на примере синтеза асинхронного RS-триггера с прямыми входами.

Постановка задачи:

Необходимо синтезировать логическое устройство со следующими свойствами:

1. При  $S = 0$  и  $R = 0$  триггер сохраняет предыдущее состояние.
2. При  $S = 1$  и  $R = 0$  происходит установка «1».
3. При  $S = 0$  и  $R = 1$  происходит сброс.
4. Состояние  $S = 1$  и  $R = 1$  запрещено.

Эти требования можно записать в полной таблице истинности (табл.5.3), в которой учитываются предыдущие состояния.

Таблица 5.3. Полная таблица истинности логического устройства

$Q(t-1)$	$S(t)$	$R(t)$	Режим	$Q(t)$	$\bar{Q}(t)$
0	0	0	Режим хранения	0	1
1	0	0	Режим хранения	1	0
0	1	0	«1»	1	0
1	1	0	«1»	1	0
0	0	1	сброс	0	1
1	0	1	сброс	0	1
0	1	1	Запрещенное состояние	-	-
1	1	1	Запрещенное состояние	-	-

Инверсный выход на предыдущем такте рассматривать не нужно, так как его состояние не случайно (определенено состоянием прямого выхода).

Записываем булево выражение:

$$Q(t) = Q(t-1)\bar{S}(t)\bar{R}(t) + \bar{Q}(t-1)S(t)\bar{R}(t) + Q(t-1)S(t)\bar{R}(t) \quad (5.1)$$

$$\bar{Q}(t) = \bar{Q}(t-1)\bar{S}(t)\bar{R}(t) + \bar{Q}(t-1)\bar{S}(t)R(t) + Q(t-1)\bar{S}(t)R(t) \quad (5.2)$$

Упростим выражения, используя дистрибутивный закон, получим:

$$Q(t) = \bar{R}(t)[Q(t-1) + S(t)] \quad (5.3)$$

$$\bar{Q}(t) = \bar{S}(t)[\bar{Q}(t-1) + R(t)] \quad (5.4)$$

Умножим обе части первого уравнения на  $\bar{R}(t)$ , второго уравнения на  $\bar{S}(t)$ , и с учетом того, что  $xx = x$ , получим:

$$Q(t)\bar{R}(t) = \bar{R}(t)[Q(t-1) + S(t)] = Q(t) \quad (5.5)$$

$$\underline{\bar{Q}(t)\bar{S}(t)} = \bar{S}(t)[\bar{Q}(t-1) + R(t)] = \bar{Q}(t) \quad (5.6)$$

Далее, на основе второй теоремы Де-Моргана, согласно которой отрицание дизъюнкции есть не что иное, как конъюнкция отрицаний

$$\overline{x \cdot y} = \bar{x} + \bar{y}, \quad (5.7)$$

проинвертировав левую и правую часть, получим 5.8

$$xy = \overline{\bar{x} + \bar{y}} \quad (5.8)$$

Подставив вместо  $x$  и  $y$   $Q(t)$  и  $R(t)$ , получим булево выражение для синтезируемого устройства:

$$Q(t) = \overline{\bar{Q}(t) + R(t)} \quad (5.9)$$

$$\bar{Q}(t) = \overline{Q(t) + S(t)} \quad (5.10)$$

Схема реализуется на логических элементах ИЛИ-НЕ с учетом положительной обратной связи.

### *Синхронные (тактируемые триггеры). Гонка сигналов.*

На входы логических элементов схемы сигналы поступают не одновременно, так как перед этим они могут проходить через разное число элементов с различной задержкой. Это явление называют гонкой сигналов.

В результате, в течение некоторого времени на выходах создается непредвиденная ситуация: новые значения одних сигналов сочетаются с предыдущими значениями других сигналов. Это может вызвать ложное срабатывание устройств.

Гонку сигналов можно устраниТЬ за счет временной синхронизации, когда на логический элемент, кроме информационных сигналов, поступают тактирующие (синхронизирующие) импульсы.

Синхронный, или тактируемый, триггер кроме информационных входов имеет синхронизирующий (тактирующий, тактовый) вход С (от англ. *Clock* – часы). По тому, как осуществляется управление (синхронизация) триггера, синхронизирующие входы подразделяются на статические и динамические.

При статической синхронизации управление осуществляется *уровнем* синхросигнала. Синхровход находится в активном состоянии во время действия синхроимпульса. Статический вход не теряет своего управляющего свойства, пока на нем присутствует тактовый синхроимпульс. При динамической синхронизации управление осуществляется *фронтом* синхроимпульса (передним или задним). При управлении фронтами разрешение на запись информации дается только в момент перепада тактового импульса:

- от нуля к единице (по переднему фронту – прямой динамический синхровход);
- от единицы к нулю (по заднему фронту, или по срезу импульса – инверсный динамический синхровход).

### *Тактируемый RS-триггер со статическим управлением*

Схема триггера представлена на рис. 5.3.

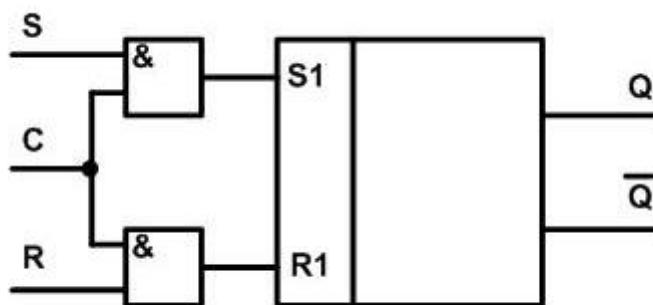


Рис. 5.3. Структурная схема тактируемого RS-триггера

Таблица 5.4.

Таблица истинности тактируемого RS-триггера

<i>C</i>	<i>S</i>	<i>R</i>	<i>Q(t)</i>	<i>Q̄(t)</i>
0	1,0	1,0	<i>Q(t-1)</i>	<i>Q̄(t-1)</i>
1	1	0	1	0
1	0	1	0	1

1	0	0	$Q(t-1)$	$\bar{Q}(t-1)$
1	1	1	-	-

Работа триггера представлена в таблице истинности (табл. 5.4), в которой необходимо учитывать состояния синхровхода  $C$ . Значение  $C = 1$  означает, что синхровход находится в активном состоянии.

В том случае, когда синхровход не находится в активном состоянии, триггер находится в состоянии хранения. Переключение триггера возможно только после поступления синхроимпульса.

### Синхронные триггеры

Синхронный двухступенчатый RS-триггер относится к классу MS-устройств.

Состоит из двух синхронных двухступенчатых RS-триггеров, соединенных через инвертор по синхронизирующему входу. Информационными сигналами для второго триггера будут выходные сигналы первого триггера.

Первый триггер носит название M (*master* – дословно: «хозяин», т.е. ведущий, основной).

Второй триггер носит название S (*slave* – дословно: «раб», т.е. ведомый, вспомогательный).

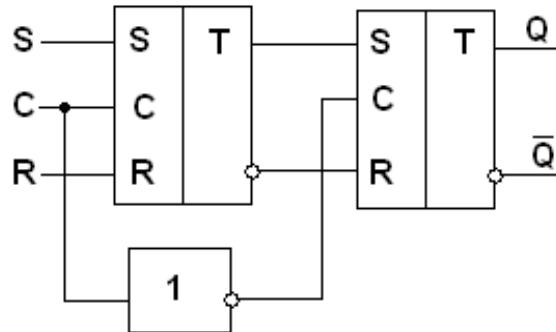


Рис. 5.4. Структурная схема двухступенчатого синхронного RS-триггера

#### Принцип действия:

1. Когда на синхровход первого триггера подана «1», то информация записывается в первую ступень по входам S и R.

Через инвертор на синхровход второй ступени подается логический «0». Вследствие этого вторая ступень заперта и не изменяет своего состояния.

2. Когда на синхровход первого триггера С подан логический «0», на входе второго триггера С будет логическая «1» и информация с первой ступени будет записана на вторую ступень, а первая ступень будет заперта.

В триггере реализуется динамический режим синхронизации. Переключение возможно по срезу синхроимпульса. Таблица истинности MS-триггера представлена в таблице ниже:

*Таблица 5.5*

Таблица истинности синхронного двухступенчатого RS-триггера.

<i>C</i> <i>по срезу синхроимпульса</i>	<i>S</i>	<i>R</i>	<i>Q(t)</i>	<i>Q̄(t)</i>
0	1, 0	1, 0	<i>Q(t-1)</i>	<i>Q̄(t-1)</i>
1	1	0	1	0
1	0	1	0	1
1	0	0	<i>Q(t-1)</i>	<i>Q̄(t-1)</i>
1	1	1	—	—

В синхронном триггере осуществляется синхронное переключение выходов. Синхронная работа очень важна для работы компьютера, где каждый шаг вычислительного процесса должен следовать в строго определенном порядке. Триггер может оставаться в неизменном состоянии даже при некоторых изменениях входных сигналов, триггер «запоминает» некоторое состояние.

В статическом RS-триггере возможно ложное срабатывание, во время активного состояния синхровхода. В динамическом RS-триггере ложное срабатывание имеет место на коротком интервале времени в окрестностях фронта или спада сигнала. Таким образом, динамический RS-триггер значительно более помехоустойчив.

Недостатком RS-триггера является наличие запрещенного состояния.

### **D-триггер**

D-триггер получил свое название от англ. слова *delay* – задержка. Рассмотрим две основные разновидности D-триггеров:

1. Статический D-триггер (триггер–зашелка), который управляется уровнем синхроимпульса.
2. Динамический D-триггер (триггер–задержка), который управляется фронтом синхроимпульса.

### **Асинхронный D-триггер**

Если в RS триггере объединить оба входа, как показано на рисунке 5.5, а на пути DR включить инвертор, мы получим схему асинхронного D – триггера. Наличие инвертора в цепи DR приводит к тому, что любой сигнал на входе D приводит к появлению на выходах RS-триггера S1 и R1 только двух комбинаций:

$$S1 = 1 \text{ и } R1 = 0$$

$$S1 = 0 \text{ и } R1 = 1$$

Таким образом, в D триггере нет запрещенных состояний.

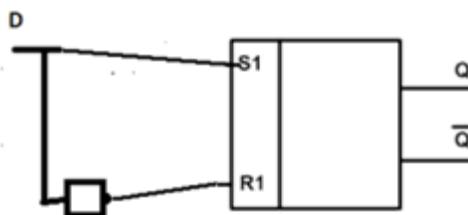


Рис. 5.5. Структурная схема асинхронного D-триггера

Синхронный триггер можно реализовать на основе асинхронного триггера, добавив синхровход и соответствующую входную логику.

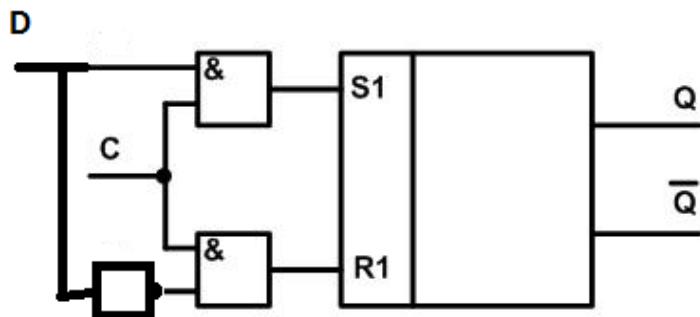


Рис. 5.6. Структурная схема синхронного D-триггера

Наличие инвертора в цепи DR приводит к тому, что любой сигнал на входе D приводит к появлению на выходах RS-триггера S1 и R1 только трех разрешенных комбинаций:

$$S1 = 1 \text{ и } R1 = 0 \text{ (} C = 1 \text{)}$$

$$S1 = 0 \text{ и } R1 = 1 \text{ (} C = 1 \text{)}$$

$$S1 = 0 \text{ и } R1 = 0 \text{ (} C = 0 \text{)}$$

Как и в случае асинхронного триггера, нет запрещенных состояний.

Реализуется статическая синхронизация по уровню синхроимпульса. D-триггер «следит» за изменением сигнала на входе во время действия синхросигнала и сохраняет ту информацию, которая имелась в момент его окончания. Он как бы защелкивает вход, поэтому и носит название триггер–зашелка. Так как вход только один, то нет гонки сигналов. Триггер используется в быстродействующих устройствах.

### *Двухступенчатый D-триггер. MS-триггер*

Двухступенчатый D-триггер состоит из двух триггеров. Первая ступень – статический D-триггер, вторая ступень – статический RS-триггер.

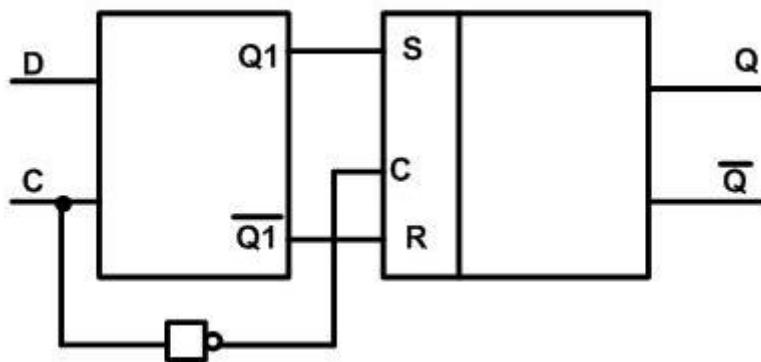


Рис. 5.7. Структурная схема двухступенчатого синхронного D-триггера

Данный триггер относится к классу MS-триггеров. Его работа аналогична работе двухступенчатого RS-триггера. Когда на синхровход первого триггера (D-триггера) подана «1», то информация записывается в первую ступень по входу D. Через инвертор на синхровход второй ступени подается логический «0». Вследствие этого вторая ступень «заперта» и не изменяет своего состояния. Когда на синхровход первого триггера С подан логический «0», на входе второго триггера С будет логическая «1» и информация с первой ступени будет записана на вторую ступень, а первая ступень будет «заперта».

В триггере реализуется динамический режим синхронизации: переключение возможно по срезу синхроимпульса, что повышает помехоустойчивость.

Так как выходной сигнал всегда задерживается на 1 такт, то этот триггер носит название «триггер-задержка».

## JK-триггер

JK-триггер имеет два информационных входа J и K и один синхронизирующий вход C:

J – *jack* – внезапное включение;

K – *kill* – аннулирование;

C – синхронизирующий вход.

По тому, как осуществляется управление (синхронизация) триггера синхронизирующими импульсами, триггеры подразделяются на *статические* (синхронизация по уровню) и *динамические* (синхронизация по срезу). Кроме того, триггеры подразделяют на *одноступенчатые* и *двухступенчатые*.

### Динамический одноступенчатый JK-триггер

Динамический одноступенчатый JK-триггер состоит из RS-триггера с инверсными входами и элементами входной логики, состоящими из элементов И-НЕ (рис. 5.8).

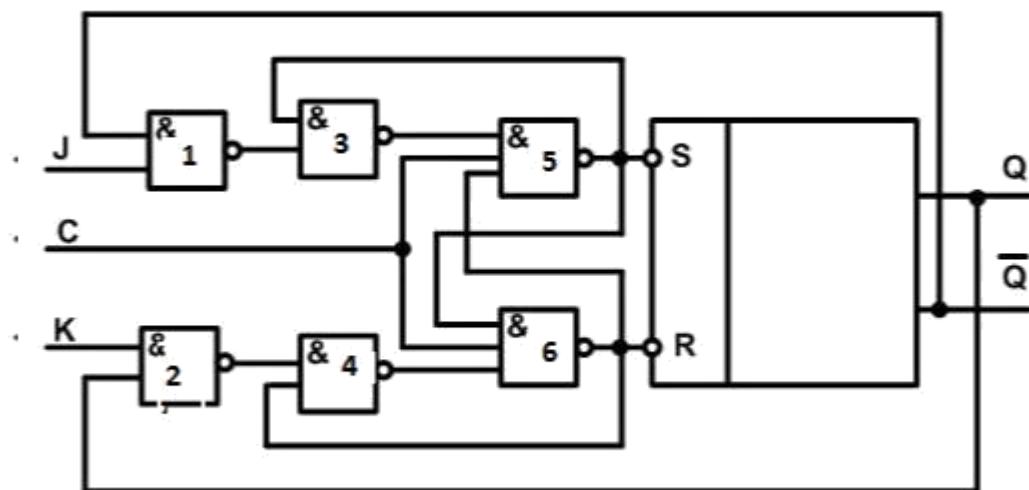


Рис. 5.8. Структурная схема одноступенчатого JK-триггера

#### Определяющие моменты работы триггера:

В такте C = 0, когда синхровход находится в неактивном состоянии, U5 = U6 = 1 (U5, U6 – это выходы элементов И-НЕ) и RS-триггер находится в режиме хранения.

Логика элементов 1-6 построена на элементах И-НЕ. На выходе элемента И-НЕ логический «0» будет только в случае, если на все входы

будет подана логическая «1». Таким образом, обратная связь элементов 5 и 6 делает невозможным «запрещенное состояние».

Триггер управляет синхроимпульсом по переднему фронту. Следовательно, обеспечивается высокая помехоустойчивость.

Нет запрещенных состояний. Обратная связь элементов 5 и 6 входной логики делает невозможным «запрещенное состояние». При одновременной подаче логической «1» на оба входа триггера ( $J$  и  $K$ ) происходит «переключение» триггера при поступлении синхроимпульса  $C$ . Состояния выходов  $Q$  и  $\bar{Q}$  меняются на противоположные. Это свойство JK-триггера используется при создании счетчиков.

В табл. 5.6 представлена таблица истинности JK-триггера.

*Таблица 5.6*

Таблица истинности JK-триггера

$C$ по переднему фронту синхроимпульса	$J$	$K$	$Q(t)$	$\bar{Q}(t)$	Режим
0	1, 0	1, 0	$Q(t-1)$	$\bar{Q}(t-1)$	Хранение
1	1	0	1	0	Уст. 1
1	0	1	0	1	Сброс
1	0	0	$Q(t-1)$	$\bar{Q}(t-1)$	Хранение
1	1	1	$\bar{Q}(t-1)$	$Q(t-1)$	Переключение

### *Динамический двухступенчатый JK-триггер*

Триггер состоит из следующих элементов: двух RS-триггеров с инверсными входами и четырех элементов входной логики, состоящей из элементов «И-НЕ».

Важной особенностью схемы «И-НЕ» является то, что логический «0» на выходе будет только в том случае, если на все входы подана логическая «1».

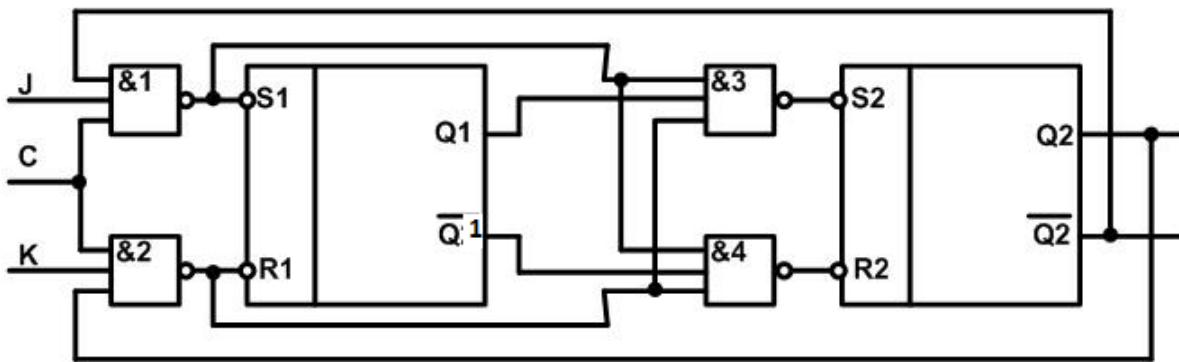


Рис. 5.9. Структурная схема двухступенчатого JK-триггера

*Определяющие моменты работы триггера:*

В такте  $C = 0$ , когда синхровход первого триггера находится в неактивном состоянии,  $U_1 = U_2 = 1$  и RS-триггер  $T_1$  находится в режиме хранения (р.х.). Второй триггер отстает на один такт от первого:

Смена состояний у  $T_1$ : р.х., «1», р.х., «0», р.х.

Смена состояний у  $T_2$ : «0», р.х., «1», р.х., «0»;

«Запрещенное состояние» для триггера  $T_1$  невозможно (для этого нужно, чтобы оба выхода второго триггера находились в состоянии «1» и «1»).

Триггер управляемся синхроимпульсом по заднему фронту (резу). Следовательно, обеспечивается высокая помехоустойчивость. Нет запрещенного состояния. При одновременной подаче логической «1» на оба входа триггера ( $J$  и  $K$ ) происходит «переключение» триггера при поступлении синхроимпульса.

### *Счетный триггер. T-триггер.*

Счетным триггером (T-триггером) называется триггер, который переключается с поступлением тактового импульса на счетный вход. Рассмотрим идею реализации подсчета импульсов на примере JK-триггера. Если на оба входа  $J$  и  $K$  подать высокий потенциал ( $J = 1, K = 1$ ), то при каждом поступлении импульсов на синхровход состояние триггера будет изменяться на противоположное. Поэтому синхровход (при  $J = 1, K = 1$ ) можно использовать как счетный вход.

## **Контрольные вопросы**

1. В чем отличие последовательной логической схемы от комбинационной логической схемы?
2. Что такое триггер?
3. Назовите разновидности триггеров по способу управления.
4. В чем отличие синхронного RS-триггера от асинхронного?
5. Приведите таблицу истинности для RS-триггера.
6. Нарисуйте схему асинхронного RS-триггера.
7. Запишите таблицу истинности асинхронного RS-триггера.
8. Что такое гонка сигналов и как ее предотвращать?
9. Нарисуйте схему тактируемого RS-триггера.
10. Запишите таблицу истинности тактируемого RS-триггера.
11. Опишите процесс записи информации в синхронный двухступенчатый RS-триггер.
12. Опишите процесс записи информации в асинхронный D-триггер.
13. Опишите процесс записи информации в синхронный двухступенчатый D-триггер.
14. Что такое MS-триггер?
15. Покажите, как в JK-триггере устраняется запрещенное состояние?
16. В чем функциональные отличия одноступенчатого JK-триггера от двухступенчатого?
17. При каких условиях JK-триггер становится счетным триггером?
18. Приведите таблицу истинности JK-триггера.

## 6. СЧЕТЧИКИ

Счетчик – это логическое устройство, выполняющее функцию подсчета числа импульсов, поступивших на его вход. Обозначается буквами СТ (*counter*).

Счетчики подразделяются (по типу выполняемых действий) на:

- Суммирующие;
- Вычитающие;
- Реверсивные (сочетающие суммирование и вычитание).

Счетчики подразделяются (по способу подачи сигналов) на:

- Синхронные (в них счетные сигналы подаются параллельно на все разряды);
- Асинхронные.

Счетчики выполняют на счетных триггерах. Выходы триггеров представляют собой выходы счетчика.  $N$  триггеров образуют  $N$ -разрядный счетчик. Выходы счетчика образуют  $N$ -разрядное двоичное число. Совокупность логических «1» и «0» на выходах триггеров представляют подсчитываемое число.

Модуль счетчика ( $M$ ) – это число различных состояний, через которые проходит счетчик в процессе полного цикла счета.

Суммирующие счетчики суммируют поступающие импульсы. Вычитающие счетчики вычитают поступающие импульсы от заранее предустановленного значения.

Рассмотрим работу асинхронного и синхронного счетчика.

*Асинхронный счетчик (счетчик прямого действия).*

В асинхронном счетчике (счетчике прямого действия) синхронизирующий вход каждого триггера связан с прямым выходом предыдущего триггера. Триггеры запускаются не одновременно, а последовательно (асинхронно).

На рис.6.1 представлена схема 4-разрядного счетчика, выполненная на четырех JK- триггерах. На входы J и K подается логическая 1, что обеспечивает работу триггера в режиме переключения. Счетные импульсы поступают на синхровход триггера (рис.6.1).

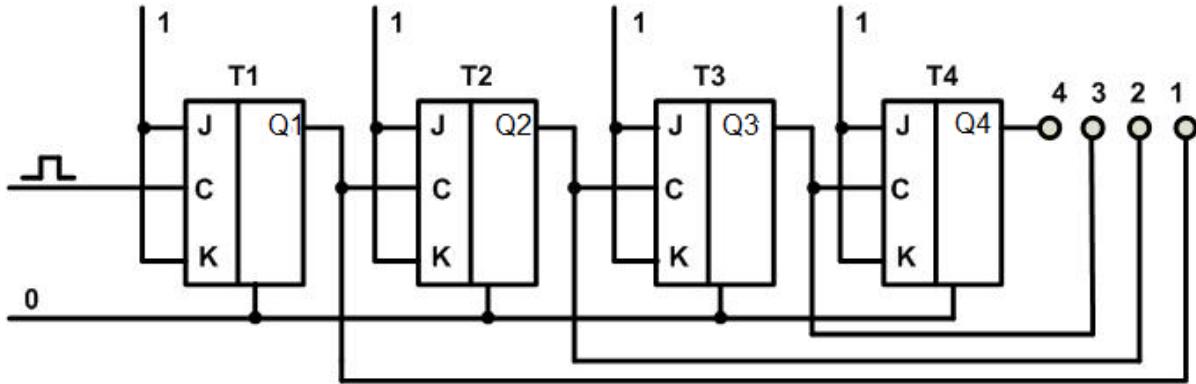


Рис. 6.1. Четырехразрядный асинхронный счетчик на JK-триггерах

### *Синхронный счетчик*

Трехразрядный синхронный счетчик, выполненный на JK-триггерах, представлен на рисунке ниже.

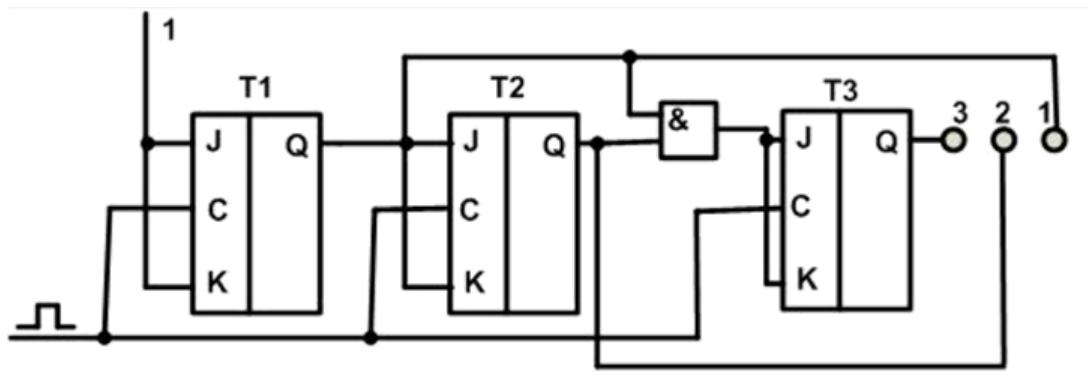


Рис. 6.2. Трехразрядный синхронный счетчик на JK-триггерах

$$N = 3, M = 8.$$

Особенности работы схемы:

1. Счетные импульсы поступают одновременно на все триггеры.
2. Триггер T1 находится в режиме переключения, так как на оба его входа поданы логические «1»:  $J_1 = K_1 = 1$ .
3. Триггер T2 переключается только в том случае, если на предыдущем такте  $Q_1=1$ . Частота переключения в два раза меньше, чем у T1.
4. Триггер T3 переключается только тогда, когда на предыдущем такте  $Q_1 = Q_2 = 1$  (из-за элемента И). Частота переключения в четыре раза меньше, чем у T1.

### **Вычитающий счетчик (счетчик обратного действия)**

В рассмотренном счетчике прямого действия синхронизирующий вход каждого триггера связан с прямым выходом предыдущего триггера. В счетчике *обратного действия* синхронизирующий вход каждого триггера связан с *инверсным* выходом предыдущего триггера. В данных триггерах предусмотрен асинхронный вход PS для введения начального состояния разряда. В отличие от суммирующего счетчика в вычитающем счетчике происходит обратный счет импульсов.

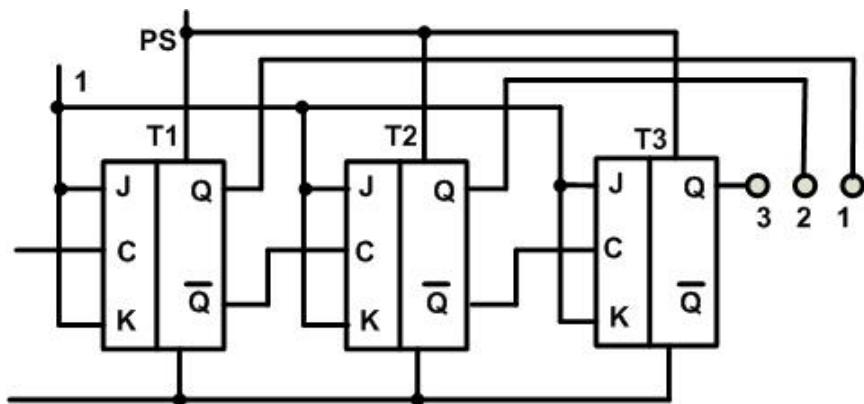


Рис. 6.3. Асинхронный вычитающий 3-разрядный счетчик на JK-триггерах

### **Контрольные вопросы**

1. Дайте определение счетчика.
2. Для чего в схемах нужен счетчик?
3. Что такое модуль счетчика?
4. Что такое разрядность счетчика?
5. В чем отличие асинхронного счетчика от синхронного?
6. На каком основном логическом элементе построены счетчики?

## 7. РЕГИСТРЫ

*Регистром* называется цифровое логическое устройство, предназначенное для приема, временного хранения, преобразования и выдачи информации  $N$ -разрядного двоичного кода (кодовой комбинации).

На схемах регистры обозначаются буквами RG (*register*).

Регистр содержит регулярный набор однотипных триггеров, в каждом из которых хранится значение одного двоичного разряда  $N$ -разрядного двоичного кода (двоичного числа).

Классификация регистров.

По составу выполняемых операций:

- Регистры записи,
- Регистры считывания,
- Поразрядные логические регистры,
- Регистры сдвига,
- Регистры преобразования последовательного кода в параллельный и наоборот.

По числу линий для записи значения одного разряда (бита информации): однофазные и парафазные:

- при *однофазном* представлении значение каждого разряда записывается по одной линии связи (рис. 7.1);
- при *парафазном* — по двум линиям (одновременно отображается прямое и инверсное значение разряда), рис. 7.2.

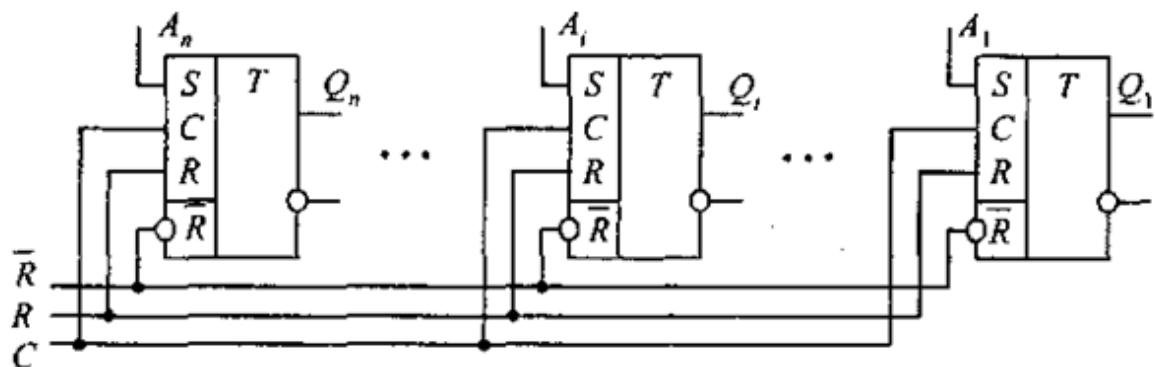


Рис. 7.1. Схема регистра с однофазной записью данных

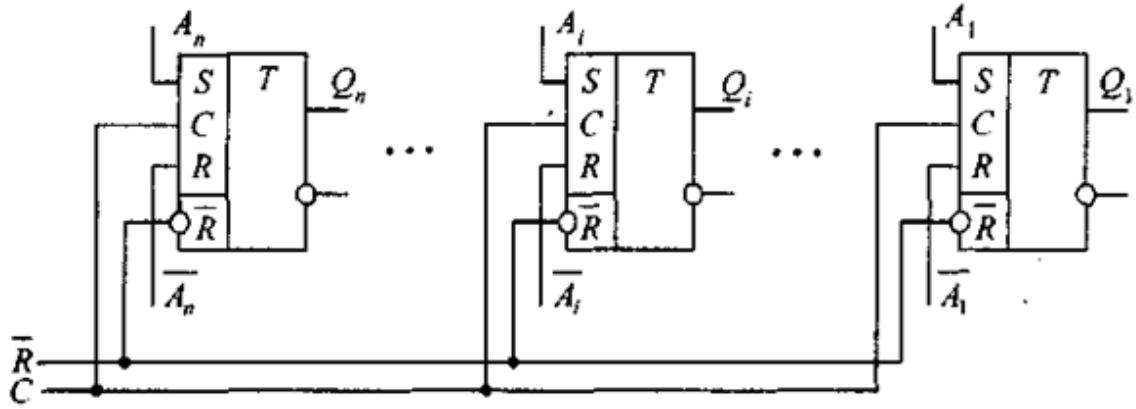


Рис. 7.2. Схема регистра с парафазной записью данных

При однофазном представлении для записи информации на входе каждого триггера ставят дополнительные комбинационные схемы, образующие входную логику регистра. При парафазной записи информации значение каждого разряда числа (кода)  $A$  записывается по двум линиям связи. При этом прямое значение  $A$  поступает на вход  $S$  (или  $J$ ) соответствующих триггеров, а инверсное значение  $A$  – на вход  $R$  (или  $K$ ).

Основным классификационным признаком, по которому различают регистры, является способ записи информации в регистр. По этому признаку можно выделить регистры следующих типов: параллельные и последовательные регистры.

В параллельные регистры запись осуществляется параллельным кодом (во все разряды регистра одновременно). Последовательные регистры характеризуются последовательной записью кода числа, начиная с младшего разряда или старшего путем последовательного сдвига кода тактирующими импульсами.

В последовательно-параллельных регистрах ввод или вывод информации может осуществляться как в параллельном, так и в последовательном кодах. Время ввода числа в регистр параллельного типа равно времени ввода одного разряда. Время ввода числа в регистр последовательного типа равно  $N \times T$ , где  $N$  – число разрядов вводимого числа, а  $T$  – период следования тактирующих сигналов, осуществляющих ввод (вывод) информации.

## *Параллельные регистры*

Параллельный регистр используется для кратковременного хранения чисел, представленных в параллельном двоичном коде. Поэтому параллельные регистры называются еще регистрами памяти.

Пример параллельного парафазного регистра на синхронных RS-триггерах приведен на Рис.7.3.

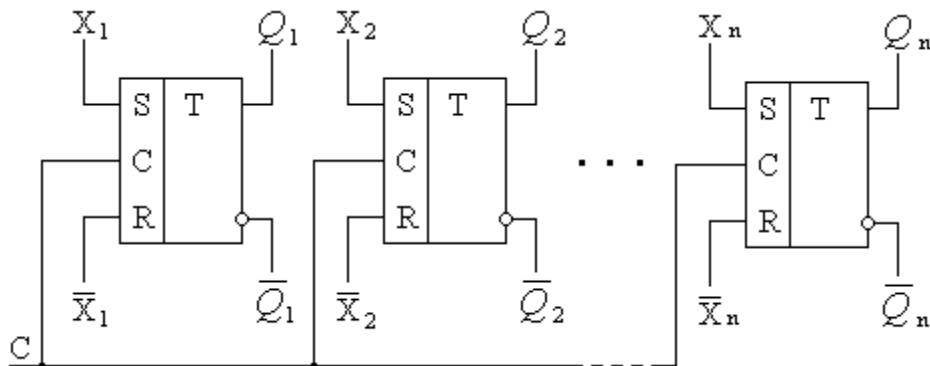


Рис. 7.3. Схема параллельного парафазного регистра  
на базе синхронного RS-триггера

В RS-триггере для записи единицы необходимо, чтобы  $S = 1$  и  $R = 0$ , а для записи нуля – наоборот,  $S = 0$ ,  $R = 1$ , т.е. информация должна поступать на оба входа RS-триггера. Полученный регистр будет парафазным, причем вход  $S$  – прямой, а вход  $R$  – инверсный. Для синхронной записи во все триггеры одновременно их тактовые входы необходимо объединить в одну шину

## *Последовательные регистры*

Последовательный регистр предназначен для кратковременного хранения информации, но, в отличие от параллельного регистра, в нем осуществляется логическая операция сдвига кода на любое количество разрядов. Ввод информации в последовательный регистр осуществляется по одному последовательному каналу.

Сдвиг кода числа происходит с помощью синхронизирующих импульсов  $C$ , в результате подачи которых осуществляется сдвиг всех разрядов со входа к выходу регистра. Последовательный регистр может быть реализован на базе триггеров RS-типа (JK-типа), но наиболее удобным для этих целей является использование D-триггеров.

Информация с первым тактирующим импульсом с входа  $V$  передается на выход первого и вход второго триггеров. С приходом второго тактирующего импульса информация из первого триггера перепишется на выход второго триггера. На выход первого триггера запишется новое значение с входа  $V$ . Таким образом, будет осуществляться сдвиг исходного кода вправо. Поэтому последовательные регистры называются регистрами сдвига. Информация выводится из триггера по одному выходу  $Q$ .

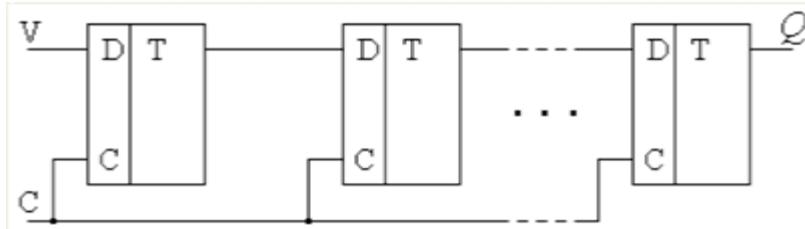


Рис. 7.4. Схема последовательного регистра сдвига на базе D-регистра

### Контрольные вопросы

1. Что называют регистром в цифровых схемах?
2. Как классифицируются регистры по составу выполняемых операций?
3. В чем отличие однофазного от парафазного регистра?
4. Как классифицируются регистры по способу записи информации?

## 8. АРИФМЕТИЧЕСКИЕ УСТРОЙСТВА

*Полусумматор* – цифровое устройство, реализующее арифметическое сложение в младшем разряде. При сложении двух чисел А и В в полусумматоре складывается содержимое младших разрядов А0 и В0 .

Результат сложения представляет собой сумму S (*sum*) и перенос T (*transfer to next digit*).

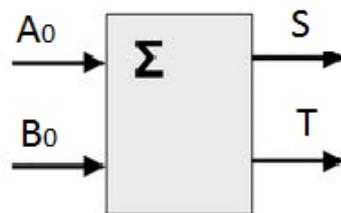


Рис. 8.1. Обозначение полусумматора на схеме

Таблица истинности полусумматора соответствует правилам сложения в младшем разряде.

Таблица 8.1

Таблица истинности полусумматора

A0	B0	S	T
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Булево выражение записываем в соответствие с таблицей истинности:

$$S = A0\bar{B}0 + \bar{A}0B0 = A0 \oplus B0$$

$$T = A0 \cdot B0$$

Схема «полусумматора» реализуется с помощью двух логических элементов: конъюнктора и «исключающего ИЛИ»

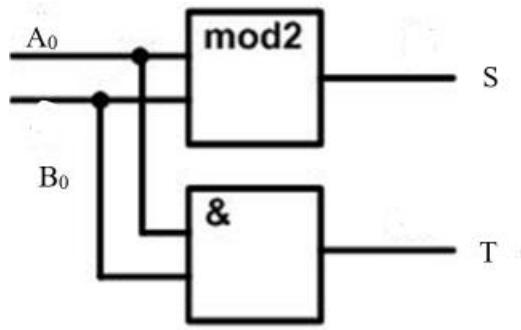


Рис. 8.2. Схема полусумматора на основе логических элементов

*Полный сумматор* – устройство, реализующее сложение в произвольном разряде. Правила сложения в произвольном разряде должны учитывать возможный перенос из предыдущего разряда ( $T_{вых}$ ). Таким образом, устройство должно иметь три входа:  $A$ ,  $B$ ,  $T_{вых}$  и два выхода:  $S$  и  $T_{вых}$

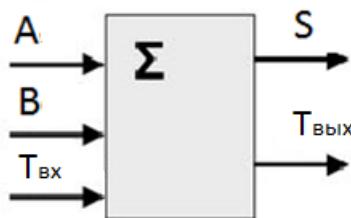


Рис. 8.3. Обозначение сумматора на схеме

Таблица 8.2  
Таблица истинности сумматора

A	B	$T_{вых}$	S	$T_{вых}$
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Булево выражение записываем в соответствие с таблицей истинности:

$$S = A\bar{B}\bar{T}_{вых} + \bar{A}B\bar{T}_{вых} + \bar{A}\bar{B}T_{вых} + ABT_{вых}$$

$$T_{вых} = AB\bar{T}_{вых} + A\bar{B}T_{вых} + \bar{A}BT_{вых} + A\bar{B}T_{вых}$$

После упрощения булево выражение будет иметь вид:

$$S = A \oplus B \oplus T_{\text{вх}}$$

$$T_{\text{вых}} = AB + (A \oplus B) \cdot T_{\text{вх}}$$

Схема «полного сумматора» реализуется с помощью логических элементов: двух элементов «конъюнктор» ( $\&$ ), двух «исключающее ИЛИ» ( $\text{mod } 2$ ) и одного элемента «дизъюнктор» (1) :

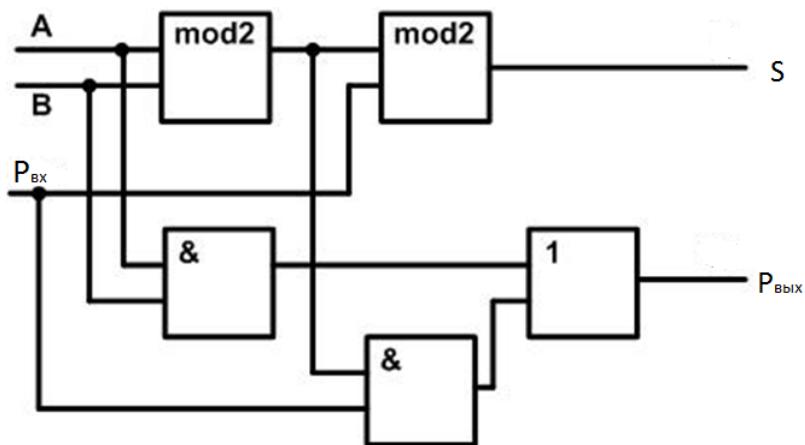


Рис. 8.4. Схема сумматора на основе логических элементов

Многоразрядный сумматор – устройство, реализующее сложение многоразрядных чисел  $A + B$ .

$$a_3\ a_2\ a_1\ a_0 + b_3\ b_2\ b_1\ b_0 = c_4\ c_3\ c_2\ c_1\ c_0$$

Многоразрядный сумматор может быть реализован в параллельной и последовательной форме. На рисунке 8.5 представлена схема параллельного сумматора. В нем значения разрядов чисел А и В одновременно поступают на входы одноразрядных сумматоров. Сумма формируется на выходах одноразрядных сумматоров:  $s_0, s_1, s_2, s_3$ .

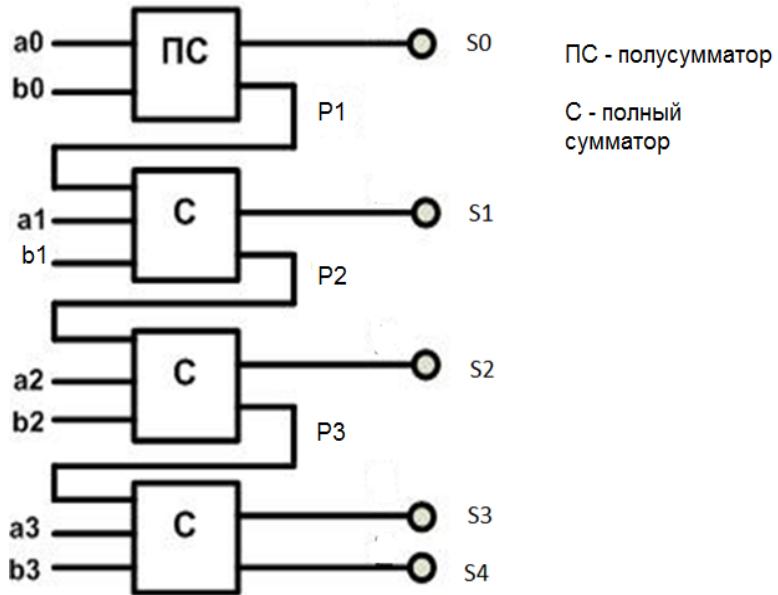


Рис. 8.5. Схема параллельного 4-х разрядного сумматора  
(ПС – полусумматор; С – сумматор)

Последовательный сумматор состоит из трех регистров сдвига: RGA, RGB, RGS, полного сумматора (реализующего сложение в одном разряде), D-триггера, осуществляющего задержку переноса на 1 такт. Регистры и D-триггер синхронизированы. Результат сложения фиксируется в регистре RGS.

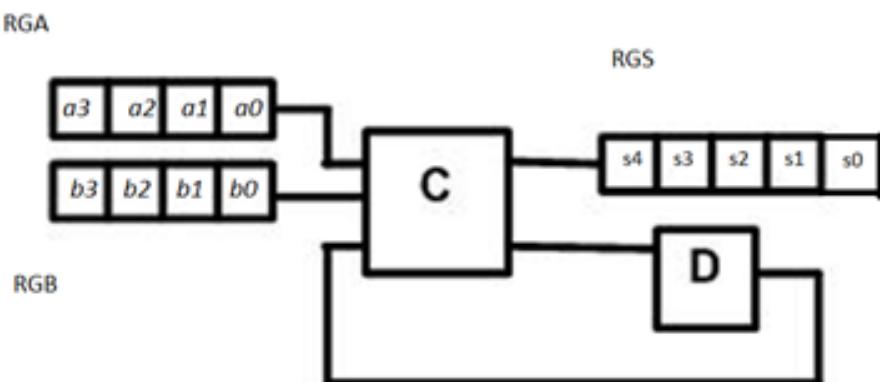


Рис. 8.6. Структурная схема последовательного 4-х разрядного сумматора

## **Контрольные вопросы**

1. Приведите таблицу истинности полусумматора и его обозначение на схеме.
2. Нарисуйте схему полусумматора на основе логических элементов.
3. Приведите таблицу истинности полного сумматора и его обозначение на схеме.
4. Нарисуйте схему полного сумматора на основе логических элементов.
5. Нарисуйте схему параллельного 4-х разрядного сумматора.
6. Нарисуйте схему последовательного 4-х разрядного сумматора.

## 9. СОПРЯЖЕНИЕ ЦИФРОВЫХ И АНАЛОГОВЫХ СХЕМ.

### Цифро-аналоговый преобразователь

*Цифро-аналоговый преобразователь* (ЦАП) – преобразователь «код–сигнал», предназначен для преобразования цифрового (двоичного) кода в аналоговый сигнал. На вход ЦАП подается цифровой код  $Q_{\text{вх}}$ , на выходе ЦАП образуется аналоговый сигнал  $U_{\text{вых}}$ .



Рис. 9.1. Структурная схема ЦАП

Основная характеристика ЦАП – разрядность: (число разрядов входного двоичного кода) –  $n$ . Считаем, что  $U_{\text{вых}}$  пропорционально зависит от  $Q_{\text{вх}}$ , тогда (длина двоичного кода зависит от разрядности ЦАП):

$U_{\text{вых}} = U_{\max}$  соответствует входному коду: 111 … 1.

$U_{\text{вых}} = U_{\min}$  соответствует входному коду: 000 … 0.

Приращение выходного напряжения на каждую единицу входного кода, равное  $\Delta U = U_{\max} / 2^n$ , называют «квантом». Очевидно, что погрешность квантования сигнала (т.е. дискретизации сигнала по уровню) не может быть меньше величины  $\Delta U$ . При этом чем выше разрядность ЦАП, тем точнее оцифровка сигнала. Схема ЦАП с двоично-взвешенными разрядами приведена рис. 9.2.

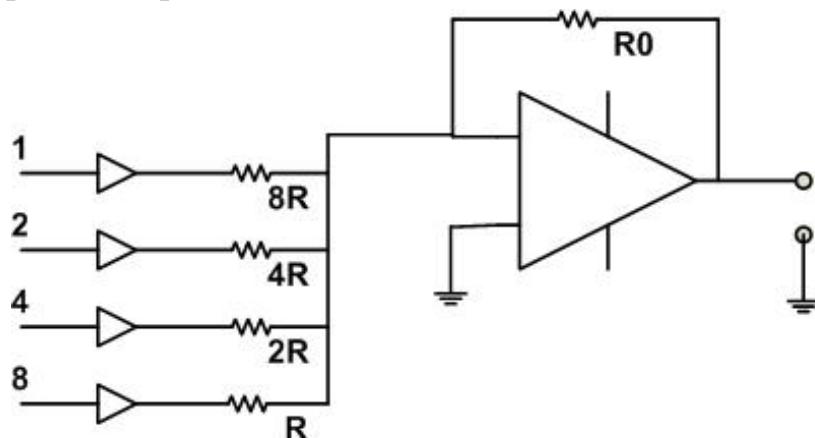


Рис. 9.2. Схема ЦАП с двоично-взвешенными разрядами

Входная кодовая комбинация в общем случае:  $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ .

Здесь  $a_k = 1$  – высокий потенциал  $U^1$ ,  $a_k = 0$  – низкий уровень  $U^0$ .

Ток  $I$  во входной цепи, обусловленный потенциалом логической «1» в  $k$ -том разряде, согласно закону Ома будет равен:

$$I_k = a_k U^1 / (2^{n-k-1} R).$$

Суммарный ток в узловой точке входа операционного усилителя будет равен сумме токов  $I_k$ :

$$\begin{aligned}\Sigma I_k &= I_{n-1} + I_{n-2} + \dots + I_1 + I_0 = \\ &= a_{n-1} U^1 / (2^0 R) + a_{n-2} U^1 / (2^1 R) + \dots + a_1 U^1 / (2^{n-2} R) + \dots + a_0 U^1 / (2^{n-1} R)\end{aligned}$$

Достоинство данного типа ЦАП – простая структура. Недостаток заключен в том, что данный ЦАП должен быть обеспечен высокой (прецизионной) точностью номиналов сопротивлений резисторов  $R$ .

### *Аналогово-цифровой преобразователь*

*Аналогово-цифровой преобразователь* (АЦП) предназначен для преобразования аналогового сигнала в цифровой (двоичный) код.



Рис. 9.3. Структурная схема АЦП

На вход АЦП подается аналоговый сигнал  $U_{вх}$ , на выходе АЦП образуется цифровой код  $Q_{вых}$ . Основная характеристика АЦП – разрядность (число разрядов выходного двоичного кода) –  $n$ .

Считаем, что  $Q_{вых}$  линейно зависит от  $U_{вх}$ , тогда (длина двоичного кода зависит от разрядности АЦП) :

$U_{вх} = U_{max}$  соответствует выходному коду  $111 \dots 1$ .

$U_{вх} = U_{min}$  соответствует выходному коду  $000 \dots 0$ .

Схема АЦП последовательного счета приведена на рис. 9.4 (на схеме приведен пример работы 4-разрядного АЦП). Данная схема состоит из компаратора (обозначен как «==»), генератора импульсов (G), двоичного счетчика (Сч), блока ЦАП, а также логических элементов «И» (&) и инвертора.

Принцип действия АЦП состоит в следующем (временные диаграммы сигналов представлены на рис. 9.5):

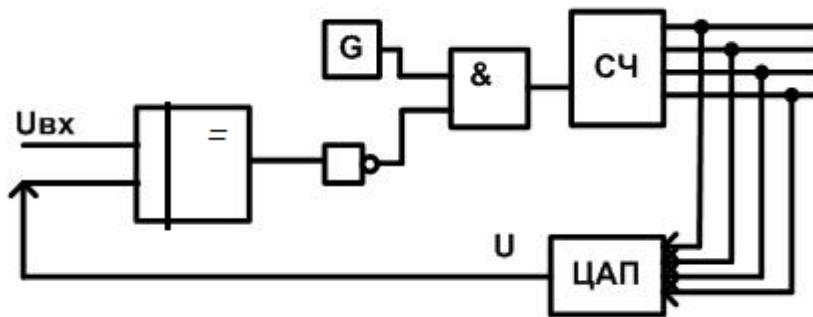


Рис. 9.4. Схема АЦП последовательного счета

1. Предварительно на один из входов компаратора подается некоторое напряжение  $U_{вх}$ , которое необходимо преобразовать в цифровой код. Далее начинается цикл счета.

2. Счетные импульсы  $U_G$  с генератора счетных импульсов (G) поступают на вход двоичного счетчика (Сч), на выходе которого формируется 4-разрядный код с нарастающим «весом».

3. Этот двоичный код поступает на ЦАП, напряжение на выходе которого последовательно увеличивается после поступления каждого очередного импульса на счетчик (цикл преобразования начинается с обнуления счетчика). Это напряжение  $U$  поступает на второй вход компаратора.

4. Когда напряжение  $U$  сравняется с  $U_{вх}$ , компаратор переключится и на его выходе появится логическая «1», которая в результате разъединит генератор G и счетчик Сч (см. схему на рис. 9.4).

5. В результате завершения данного цикла счета показания счетчика Сч (4-разрядный код) будут соответствовать цифровому эквиваленту входного сигнала  $U_{вх}$ .

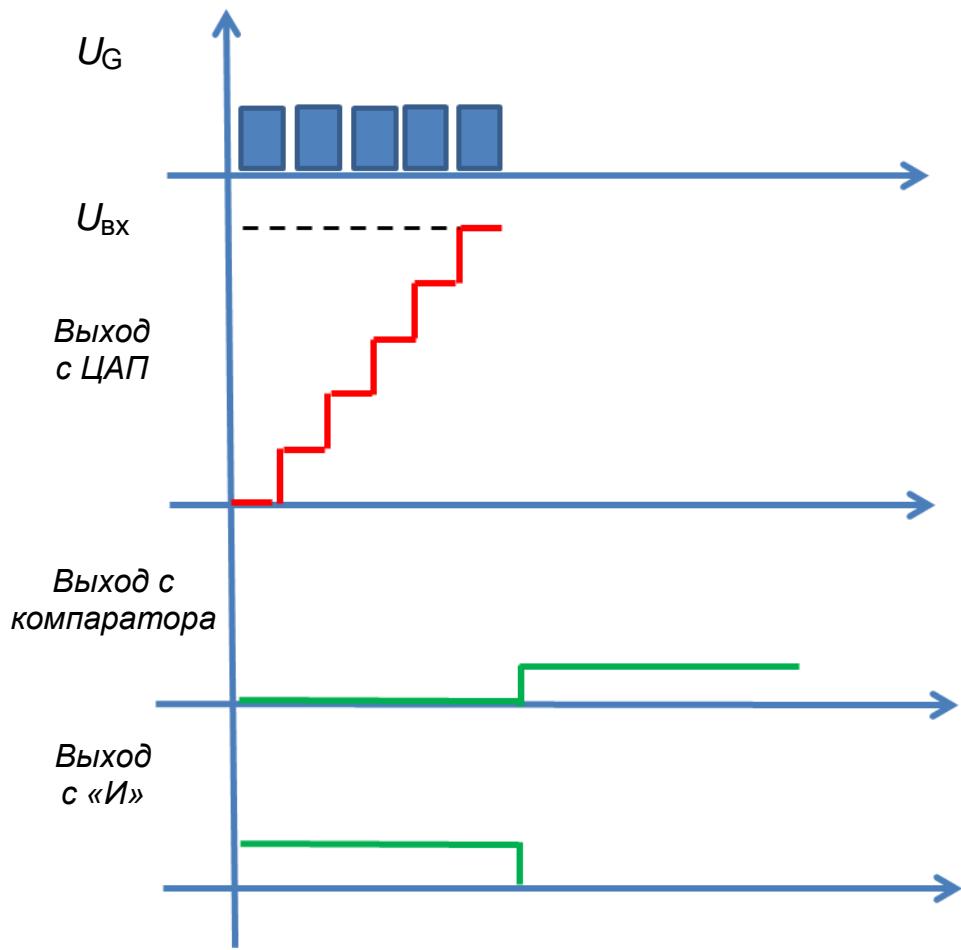


Рис. 9.5. Временная диаграмма работы АЦП  
последовательного счета

### Контрольные вопросы

1. Что такое ЦАП и в чем его назначение?
2. Приведите схему ЦАП с двоично-взвешенными разрядами.

Поясните его работу.

3. Что такое АЦП и в чем его назначение?
4. Приведите схему АЦП последовательного счета. Поясните его работу.

## **Литература**

### *Основная литература:*

1. Микушин А.В. Цифровые устройства и микропроцессоры: учеб. Пособие / А.В. Микушин. – СПб.: БХВ-Петербург, 2010. – 832 с.  
URL: <http://znanium.com/bookread.php?book=350706>
2. Угрюмов Е.П. Цифровая схемотехника : учеб. пособие для вузов / Е.П. Угрюмов. – СПб.: БХВ-Петербург, 2010. – 816с. URL: <http://znanium.com/bookread.php?book=350426>
3. Щука А.А. Электроника / А.А.Щука. – СПб.: БХВ-Петербург, 2010. – 2008. – 751с. URL: <http://znanium.com/catalog.php?bookinfo=350420>
4. Нарышкин А.К. Цифровые устройства и микропроцессоры: учебное пособие для студентов вузов радиотехнических специальностей / А.К.Нарышкин. – М: Академия, 2006. – 317с.
5. Опадчий Ю.Ф. Аналоговая и цифровая электроника: полный курс: Учеб. для студ. вузов. (под ред. О.П. Глудкина) / Ю.Ф. Опадчий, О.П.Глудкин, А.И. Гуров. – М.: Горячая линия – Телеком, 2003. – 768с.
6. Фрике К. Вводный курс цифровой электроники: Учеб. пособие для студентов в области проектирования цифровых интегральных схем / К.Фрике, – М.: Техносфера, 2003. – 428с.

### *Дополнительная литература:*

- 7.Партала О.Н. Цифровая электроника / О.Н. Партала. – СПб: Наука и техника, 2001. – 219с.
8. Хоровиц П. Искусство схемотехники: В 3-х т. Т.2 / П.Хоровиц. – М.: Мир, 1993. – 371 с.
9. Токхейм Р. Основы цифровой электроники / Р.Токхейм – Мир, 1988.
10. Янсен И. Курс цифровой электроники / И. Янсен. – Мир, 1987.
11. Roger L. Theory and problems of digital principles / L. Roger, M.S. Tokheim.–USA,1994. – P.358. URL: <http://bookre.org/reader?file=583664&pg=1>
12. Брамер Ю.А. Цифровые устройства / Ю.А. Брамер. – М.: Импульсная техника, 2005.

**Учебное издание**

Карпов Аркадий Васильевич  
Калабанов Сергей Александрович  
Ишмуратов Рашид Аминович

## **ОСНОВЫ ЦИФРОВОЙ ЭЛЕКТРОНИКИ**

**Учебное пособие**

**Редактор**  
**Н.И. Андронова**

**Компьютерная верстка**  
**А.А. Аксенова**

**Дизайн обложки**  
**М.А. Ахметов**

Подписано в печать 05.02.2015.  
Бумага офсетная. Печать цифровая.  
Формат 60x84 1/16. Гарнитура «Times New Roman». Усл. печ. л. 20,93  
Уч.-изд. л. 11,93. Тираж 100 экз. Заказ 17/2

**Отпечатано в типографии**  
**Издательства Казанского университета**

420008, г. Казань, ул. Профессора Нужина, 1/37  
тел. (843) 233-73-59, 233-73-28