MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”

ELECTRONIC SYSTEMS-1
LECTURE NOTES

Tutorial

Recommended by Methodical council of “Igor Sikorsky Kyiv Polytechnic Institute”
as a tutorial for bachelors
according to the educational program “Electronic components and systems”
specialty 171 Electronics

Compiler: K. S. Klen

Electronic online educational publication

Kyiv
“Igor Sikorsky Kyiv Polytechnic Institute”
2022

Reviewer *Naida S.A.*, doctor of technical sciences, professor
“Igor Sikorsky Kyiv Polytechnic Institute”

Responsible editor *Yamnenko Y.S.*, doctor of technical sciences, professor

The stamp was provided by Methodical council of Igor Sikorsky Kyiv Polytechnic Institute
(protocol № 3 from 01.12.2022)
at request of the Academic council of the faculty
(protocol № 09/2022-2 from 26.09.2022)

The discipline «Electronic Systems» belongs to the cycle of professional and practical training of bachelors in the educational program «Electronic Components and Systems» is read over one semester (7) and is one of the final subjects of the bachelor's degree. In the process of studying the course, students get acquainted with the informational assessments of the ES; a description of the signals used in different purposes of the ES; methods of their processing, storage and transformation; principles of construction and operation of the ES - the selection, transformation, transmission, reception, registration and display of information. The basics of device design based on programmable logic integrated circuits (FPGA) are considered. Lecture notes contain theoretical information for up to 18 lectures and a list of recommended reading.

Register № 22/23-147. Volume 10,7 author's sheet

National technical university of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Peremohy Ave., 37, Kyiv, 03056
https://kpi.ua

Certificate of inclusion in the State Register of publishers, manufacturers

and distributors of publishing products DK № 5354 dated 25.05.2017

Content

Introduction

The discipline "Electronic Systems" (ES) belongs to the cycle of professional and practical training of bachelors in the educational program "Electronic Components and Systems" is read over one semester (7) and is one of the final subjects of the bachelor's degree.

In the process of studying the course, students get acquainted with the informational assessments of the ES; a description of the signals used in different purposes of the ES; methods of their processing, storage and transformation; principles of construction and operation of the ES - the selection, transformation, transmission, reception, registration and display of information.The basics of device design based on programmable logic integrated circuits (FPGA) are considered.

Lecture notes contain theoretical information for up to 18 lectures and a list of recommended reading.

The abstract considers the issues of information evaluation of electronic systems, signaling, as well as methods of their transformation and storage, principles of construction and operation of electronic systems for selection, conversion, transmission, reception, registration and display of information based on programmable logic integrated circuits.

Lecture №1. Definitions, general, qualitative and system-creating principles. Generalized functional schemes of electronic systems and their blocks

The word *system* comes from the Greek "composed" and is explained as a group of different objects that are united in such a way that they form a single whole, function in harmony and are subject to a single form of control.

This definition can also be applied to electronic (ES) in its most general form: a set of electronic components that are interconnected and act as a whole through special control signals and functions used. In other words: an *electronic system* is any electronic university, unit, device or complex that performs information processing.

Thus, any device from a single-stage amplifier to the most complex microprocessor system can be studied as an electronic system. But there is a significant difference between a microprocessor system and a single-stage amplifier in terms of describing the details of each system. It is almost impossible to describe a microprocessor system in the form of components, capacitors, transistors, which describe the amplifier. Therefore, to describe such a system, it is necessary to group a number of components in functional blocks. Such a functional unit is called a black box or subsystem.

Electronic subsystem – is a part of a system that consists of more than one element (an intermediate element of the system division) and which has a certain functional purpose of a lower level than the system.

The generalized electronic system designed for processing input and output signals is shown in Fig. 1.1.

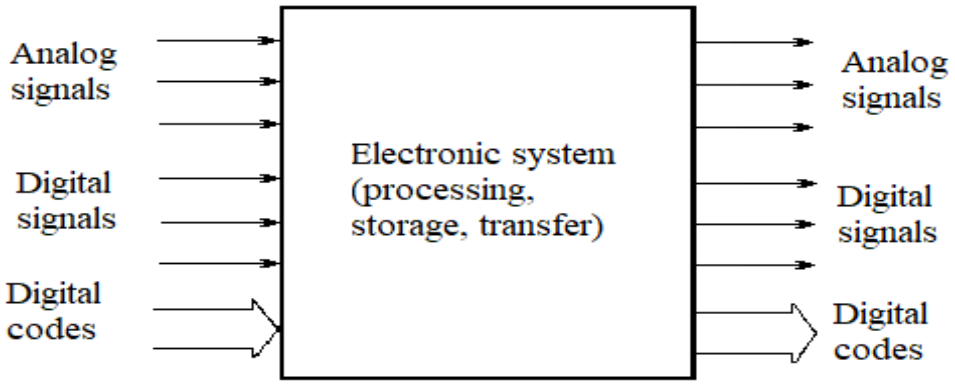


Fig. 1.1

An element of an electronic system is an integral part of the system.

It is often a problem to simultaneously have information about hundreds or thousands of homogeneous and (or) heterogeneous physical quantities, which must be obtained without human intervention, using technical means that are quite complex automatic systems.

Under automation is understood the use of a device that according to a given program without human intervention performs all operations in the receipt, conversion, transmission and distribution of energy, materials and information.

Automated electronic system – is a set of controlled object and automatic devices in which part of the control functions are performed by a person (operator).

Automated electronic system – is a collection of a controlled object and automated devices that perform a specific task according to a given program without human intervention.

By purpose, automatic systems are divided into the following **classes**:

- information and measuring;
- control;
- identification or recognition of images;
- management.

Any system is characterized by task, speed, flexibility and redundancy.

Task – is a set of functions that must be performed by an electronic system.

Speed – is an indicator of the speed of the electronic system to perform its functions.

Flexibility – is the ability of a system to adapt to different tasks.

Redundancy – is an indicator of the degree to which a system's capabilities match the system's task.

The tasks solved by the EU reflect the purpose of its operation. This goal is achieved by solving specific functional problems. Functional tasks can be conveniently divided into **four** levels.

First level: tasks of collecting and pre-processing information. These include:

- quantization of analog signals by time and level;
- preliminary digital or analog signal filtering;

- transformation and calculation of spectra;
- normalization, amplification or attenuation of signals;
- change of signal levels;
- conversion of current into voltage.

Second level:

- geometric transformations of coordinate systems;
- determination of variable parameters of measured processes;
- joint solution of linear and nonlinear systems of equations;
- solution of systems of differential equations;
- interpolation and extrapolation of calculated functions;
- digital-to-analog conversions..

Third level: solving information processing problems. Such problems include optimization problems, ie finding the extrema of a function or several variables, which are subject to certain restrictions.

Fourth level: solving problems of presenting information in a form convenient for perception by the operator.

Classification of electronic systems. Generalized functional schemes of electronic systems and their blocks

The input and output signals of the electronic system can be analog signals, single digital signals, digital codes, sequences of digital codes. Accordingly, the systems can be **analog**, **digital** or **combined**.

If the system is analog-digital, the input analog signals are converted into a sequence of sample codes using an analog-to-digital converter (ADC), and the output analog signals are generated from a sequence of sample codes using a digital-to-analog converter (DAC). Processing and storage of information are performed digitally.

By their structure, electronic systems are divided into systems on "hard logic" and microprocessor systems.

A characteristic feature of a traditional digital system, in contrast to a microprocessor system, is that the algorithms for processing and storing information

in it are strictly related to the circuitry of the system. That is, changing these algorithms is possible only by changing the structure of the system, replacing the electronic components included in the system, and / or the links between them. That is why the traditional digital system is often called a system of "rigid logic".

Any system on "hard logic" is necessarily a specialized system, configured exclusively for one task or (less often) for several close, pre-known tasks. This has its undeniable advantages.

First, a specialized system (unlike a universal one) never has hardware redundancy, ie each of its elements must work at full capacity (of course, if this system is properly designed).

Secondly, it is a specialized system that can provide the highest possible speed, because the speed of information processing algorithms is determined in it only by the speed of individual logical elements and the selected scheme of information paths. Namely, logical elements always have the maximum speed at the moment.

But at the same time, the big disadvantage of a digital system based on "rigid logic" is that for each new task it must be designed and manufactured again. This process is long, expensive and requires highly qualified performers. The way to overcome this shortcoming is quite obvious: it is necessary to build a system that could easily adapt to any task, to rebuild from one algorithm to another without changing the equipment. And set this or that algorithm by entering into the system additional control information, the program of the system. Then the system will be universal or programmable, not rigid, but flexible. This is what the microprocessor system provides.

Microprocessor – a software-controlled device designed to process digital information and control the process of this processing, made in the form of one (or more) integrated circuits with a high degree of integration of electronic elements.

Reducing the cost, power consumption and dimensions, increasing the reliability and performance of microprocessors have contributed to a significant expansion of their scope. Along with traditional computer systems, they are increasingly being used in control and processing tasks. The microprocessor was faced with the task of software control of various peripheral objects in real time.

The simplified block diagram of the microprocessor control system has the form (Fig.1.2).

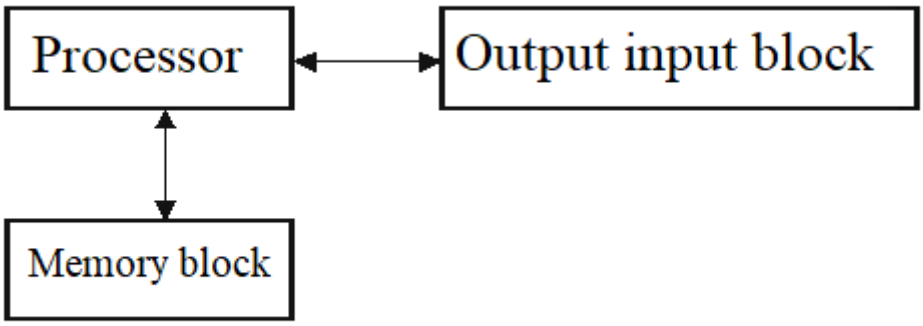


Fig. 1.2

The processor is responsible for performing all software actions required in accordance with the algorithm of the device. The memory block stores the commands of the program of the CPU operation, as well as the values of constants and variables involved in the calculations. Input output block is used to connect the microprocessor system to the control object.

Extensive use of microprocessor technology for control tasks has led to the emergence of specialized microprocessor devices focused on this type of application. The peculiarity of these chips is that in addition to the actual processor, on the same chip is the input output block system, which reduces the functional complexity and overall dimensions of the microprocessor control system. Such devices are called microcontrollers.

Microcontroller - a computing and control device designed to perform the functions of logical control and control of peripheral equipment, made in the form of a single measuring information system, which combines a microprocessor core and a set of built-in input output blocks devices.

In modern conversion technology, microcontrollers perform not only the role of direct control of the semiconductor converter due to built-in specialized peripherals, but also the role of digital controller, protection and diagnostic system, and communication system with a high-level technological network.

Recently, a number of microcontrollers have appeared, specialized for the control of semiconductor converters. Their computing core, built, as a rule, on the basis of so-called "digital signal processors", is adapted to perform recurrent polynomial digital control algorithms. Built-in peripherals include multi-channel PWM signal generators, analog-to-digital converters, vector coordinate conversion units, timers-counters, and the like. Examples of such devices are microcontrollers ADMC330 belongs to the firm Analog Devices, TMS320C240 belongs to the firm Texas Instruments, 56800 belongs to the firm Motorola, vector coprocessor ADMC200 belongs to the firm Analog Devices.

But any versatility inevitably leads to redundancy. After all, solving the most difficult problem requires much more resources than solving the simplest problem. Therefore, the complexity of the universal system should be such as to ensure the solution of the most difficult problem, and when solving a simple problem, the system will not work at full capacity, will not use all its resources. And the simpler the problem, the greater the redundancy and the less justified the universality. Redundancy leads to an increase in the cost of the system, a decrease in its reliability, an increase in power consumption, etc.

In addition, versatility tends to significantly reduce performance. It is necessary to optimize the universal system so that each new task is solved as quickly as possible. The general rule is: the greater the versatility, flexibility, the lower the speed. Moreover, for universal systems, there are no problems that they can solve with the highest possible speed.

Thus, we can draw the following conclusion. Systems on "hard logic" are effective where the problem is not changed for a long time, where the highest speed is required, where the algorithms for processing information are extremely simple. And universal, programmable systems are effective where tasks are often changed, where high speed is not too important, where information processing algorithms are quite complex.

However, in recent decades, the performance of universal (microprocessor) systems has increased significantly (by several orders of magnitude). In addition, the large volume of chip production for these systems has led to a sharp decline in their

cost. As a result, the scope of systems on "hard logic" has narrowed sharply. Moreover, programmable systems designed to solve one problem or several related problems are developing rapidly. They successfully combine the advantages of both systems with "hard logic" and programmable systems, providing a fairly high speed and the necessary flexibility.

In the scientific and technical literature, systems that use computers as part of their application, as well as the class of problems they solve, are called computer or computerized systems.

Computerized systems – are systems that mean a set of technical means, which includes a computer control, means of collecting, converting, transmitting, displaying information, as well as special mathematical and software that performs the entire complex of information processing.

It is believed that the computer should be introduced into the automatic system when the amount of information obtained becomes very large, and information processing is complicated or becomes too long in time and does not meet the requirements or performance of the system. Such systems provide information multiplexing subsystems, measurement channels and communication channels.

The generalized scheme of the computerized system is given in fig. 1.3.

Measuring channel – is a measuring instrument that is an integral part of measuring systems in the form of a set of measuring operations, means (channels or lines) of communication for sequential transformations and information transmission.

Communication channel – is a set of transmitters, receivers, and communication lines designed to transmit information from a transmitter (source) to a receiver.

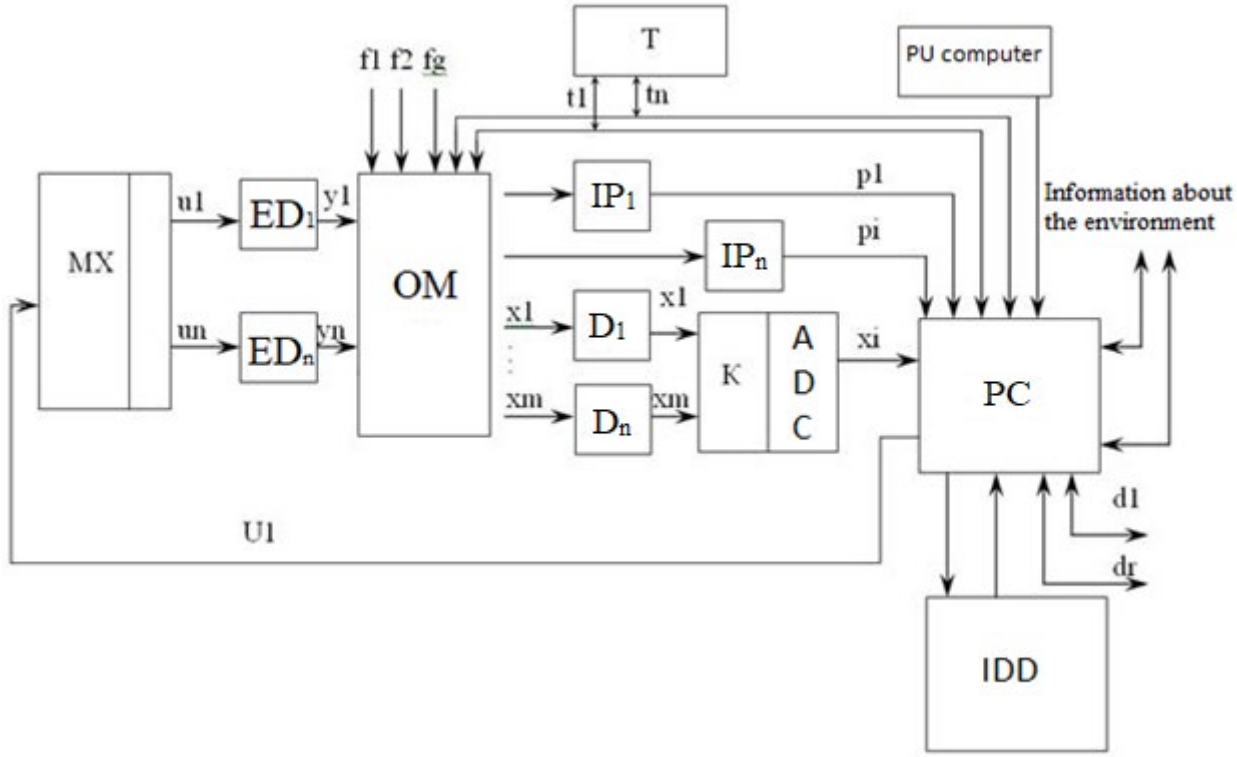


Fig. 1.3

In fig. 1.3 .: MX - multiplexer; OM - object of management (research); D1,…, Dn - sensors; Xi,…, Xm - information; ED - executive device; MI - message indicators; u1,…, un - control signal; T - timer; PU computer - computer control panel; IDD - information display device.

Communication line – is a physical medium through which signals are transmitted.

Programmable communication channel or interface – is the coherence of information exchange and information exchange rules, which means the electrical, logical and structural compatibility of the devices involved in the exchange.

Samples of electronic systems

Most modern electronic systems are built-in. A robot, such as a rover, is a built-in system. Cell phone, PDA, portable media player are built-in devices. Even an electric toothbrush is a built-in system. A small microcontroller in the toothbrush provides software speed control and battery charge status indication. Modern cars can

contain more than a hundred built-in microcontrollers. In general, embedded systems make up most of the world's microprocessor production.

As shown in Table 1.1, built-in systems can be found in a variety of products, including aircraft and military systems, biomedical systems, automobiles, communications, computer devices, electronic tools, home electronics, industrial equipment, office machines, personal devices, robots and intelligent toys. Embedded electronic systems can be found everywhere. Built-in system designers are often faced with complex design tasks because they must be reliable. Many of them cannot be repaired and cannot be rebooted. The software cannot be updated on many built-in devices. Many systems have strict design constraints on performance and energy consumption. Some systems require a long battery life. Many applications have real-time limitations, and many systems have limited memory and computing power.

Table 1.1

Aviation and military systems	Aircraft autopilots, avionics and navigation systems, automatic landing systems, guidance systems, engine control
Biomedical systems	Computed tomography and ultrasound systems, patient monitoring, pacemakers
Cars	Engine control, anti-lock brake systems, airbag control, heating and air conditioning control GPS navigation, satellite radio, system diagnostics
Communications	Communication satellites, network routers, switches, hubs
Consumer electronics	TVs, ovens, DVD players, stereos, security systems, lawn irrigation control, thermostats, cameras, answering machines, TV decoders
Devices for the computer	Keyboards, mice, printers, scanners, displays, modems, hard drives, DVD drives, graphics cards, USB devices
Electronic tools	Data collection systems, oscilloscopes, voltmeters, signal generators, logic analyzers
Industrial equipment	Elevator control, surveillance systems, robots, CNC machines, programmable logic controllers, industrial automation and control systems
Office machines	Fax machines, copiers, telephones, calculators, cash registers
Personal devices	Cell phones, portable MP3 players, personal digital assistants (PDAs), electronic watches, portable video games, digital cameras, GPS systems
Robots	Industrial works, autonomous vehicles, space research works
Toys	video game systems, toy robots such as "Aibo", "Furby", "Elmo"

Real-time systems must respond to external input parameters and create new output results in a limited time, as shown in Fig. 1.4. Response time should be limited. Very long response times can cause real-time systems to fail.

An illustrative example of a real-time system is a car airbag controller. When the airbag motion sensors (accelerometers) detect a collision, the system must respond by opening the airbag for 10 ms or the system will not operate properly. At high speeds with a delay of more than 10 ms, the driver will collide with the steering wheel before the airbag opens.

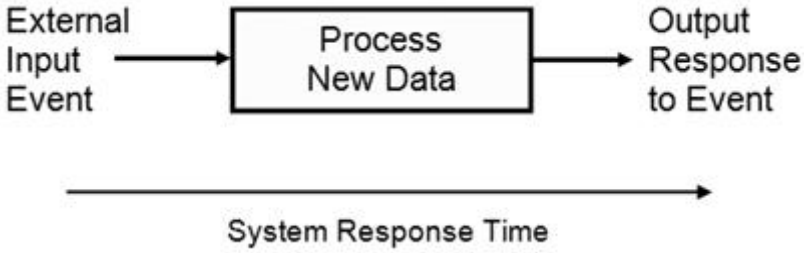


Fig. 1.4

In a "soft" real-time system, priorities are critical. A soft real-time system usually satisfies the real-time response limit. An example of a typical soft real-time system is a media player. The player may sometimes miss a video or audio and the user may not even notice.

In a "hard" real-time system, the new output must always be calculated within the specified time limits or the system will not work. As an example of a rigid real-time system, consider a remote control system (ie, computer-controlled). In an airplane flight control system, when the pilot moves the rudder, the flight controls must move very quickly in response or the aircraft will lose stability and fall. To ensure safety, the FAA constantly checks and certifies the real-time response of computer-controlled flight simulators and aircraft. Many built-in systems are real-time systems with multiple inputs and outputs. Several events occur independently of each other. Programming such systems is simplified when dividing tasks, but this requires the CPU to constantly switch between different tasks. This switching and division of CPU time between several tasks is usually provided by multitasking operating systems. They also provide the synchronization elements needed to coordinate actions between different tasks performed in.

Lecture №2. Information and its forms. Entropy as a measure of choice uncertainty. Conditional entropy. Entropy of union

Information – is an abstract concept that has different meanings depending on the context. Derived from the Latin word "*informatio*", which has several meanings:

- explanation; presentation of facts, events; interpretation;
- representations, concepts;
- acquaintance, education.

Types of information

Information can be divided into types on several grounds:

1) By way of perception:

For humans, information is divided into types depending on the type of receptors that perceive it.

- Visual - perceived by the visual organs.
- Audial - perceived by the hearing organs.
- Tactile - perceived by tactile receptors.
- Olfactory - perceived by olfactory receptors.
- Taste - perceived by taste buds.

2) In the form of submission:

- Symbolic - based on the use of symbols - letters, numbers, signs, etc., is the simplest and is practically used only to transmit signals about various events.
- Text - transmitted in the form of symbols intended to denote tokens of the language; symbols are used - letters, numbers, mathematical signs. However, the information is embedded not only in these symbols, but also in their combination, the order of passage. Due to the interconnection of symbols and the display of human language, textual information is extremely convenient and widely used in everyday life.
- Numeric - in the form of numbers and signs denoting mathematical operations;
- Graphic - in the form of images, events, objects, graphics;

– Audio - oral or in the form of a recording of the transmission of a language token by audio.

3) By purpose:

– Mass – contains trivial information and operates with a set of concepts understandable to most of society;

– Special - contains a specific set of concepts, when used there is a transfer of information that may not be clear to the majority of society, but necessary and understandable within a narrow social group, where this information is used;

– Personal - a set of information about a person that determines the social status and types of social interactions.

Quantitative measures of information

Messages transmitted in human language consist of a certain set of written (alphabet, punctuation) or real (phonetic) signs. The results of measurements are denoted by a system of signs, which consist of numbers and letters that form a certain code, ie numbers are numerical values, and letters are units of measurement. In sign theory, called semiotics, sign systems are studied at three main levels: syntactic, semantic and pragmatic.

At the syntactic level, messages are considered as symbols that are abstracted from the content or any value of information. The subject of analysis at this level is the frequency of characters, ie code characters, the relationship between them, the order of passage, the rules for constructing expressions by which messages can be formulated.

At the semantic level, the meaning of the message symbols is studied, ie the relation of the message to what it reflects. A syntactically well-constructed phrase can be completely semantically incorrect. For example, the sentence: "The height of the house is 1000 tons" is correctly grammatically composed, but have no meaning.

At the pragmatic level, message symbols are considered in relation to the recipient. This takes into account such characteristics of the message as importance, usefulness, value, relevance. For example, the message "tomorrow a wind of 11 points is expected", ie a storm that can cause great trouble, is more important than the

message that "tomorrow a strong wind is expected - a force of 3 points". Syntactically and semantically, these messages are the same.

To date, the most developed task of syntactic evaluation of information.

At the syntactic level, information analysis is defined as a measure of reducing the uncertainty of knowledge about any object (subject) in the process of its cognition. To assess the amount of information at the syntactic level, it is necessary to know the degree of uncertainty of a situation. It is intuitively clear that the amount of information that will be obtained as a result of an experiment or action is proportional to the magnitude of the difference between the uncertainty of the information before and after the study. If H_1 – the uncertainty of knowledge of a particular issue before the study (a priori), and H_2 – the uncertainty that characterizes the state of knowledge after the study (a posteriori), then the information that exists in this message is defined as. $I = H_1 - H_2$.

To assess the level of uncertainty of knowledge at the syntactic level, a large number of different mathematical measures have been developed, but the most used are two measures:

- tructural or logarithmic, proposed by Ralph Hartley in 1928;
- probabilistic, proposed by Claude Shannon in 1948.

Hartley Measure

In developing a logarithmic measure, R. Hartley proposed the following:

– signals by which messages are transmitted consist of discrete elements (symbols);

- signal elements can take a finite number of different values (*m*) (the set of all *m* values forms the alphabet);
- all symbols are statistically independent and equally probable;
- information is transmitted in the absence of interference.
- Assume that the symbols of the alphabet can take different values with equal probability of their occurrence, i.e.: $p_1 = p_2 = \ldots = p_i = \ldots = p_m,\ p_i = \frac{1}{m}$.

In the sequence of n such characters can be obtained N signals (messages). With $N = m^n$.

For example, $m = 2$, $n = 3$ then $N = 2^3 = 8$. Indeed, if in the binary alphabet only two characters "0" or "1" in a sequence of three such characters we can get 8 words (messages), namely:000, 001, 010, 011, 100, 101 and 111.

The number of possible message words increases with the number of character n. It is clear that the amount of information contained in each message is proportional to the number of characters n, and therefore R. Hartley proposed a measure of defining information as the logarithm of possible signals with the same number of characters.

$$I = \log_a N = \log_a m^n = n \log_a m . \qquad (2.1)$$

Depending on the basis of the logarithm, there are different units of information.

– The unit of quantity of information, which is a choice of two equally probable events, is called a binary unit, or *bit* (base of the logarithm $a = 2$) . The name bit is formed from the two initial and last letters of the English expression binary *unit*, which means binary unit.

– If the basis is a logarithm, then the unit of measurement of information is called bit

– And if $a = e$ (ie the basis of the natural logarithm), then the information is measured in bolts.

The logarithmic measure for the uncertainty of information is chosen because it satisfies the conditions of **additivity**, which allows you to use it to reproduce complex experiments. That is, in the presence of independent sources of information with quantity N_1 and N_2 possible messages each of N possible values of the first value can appear at the same time with any of N_2 values of another value. The number of all possible combinations of values of these two quantities is equal to the product $N = N_1 \cdot N_2$, and then: $I = \log N = \log_a N_1 N_2 = \log_a N_1 + \log N_2$.

The amount of information per message is equal to the sum of the amounts of information received from two independent sources taken separately.

Shannon measure

If the probability of any state or appearance of the alphabet symbol is different, i.e. $p_1 \neq p_2 \neq p_m$, the probabilistic measure proposed by C. Shannon is used.

Shannon measure allows you to determine the average amount of information that will be received per message:

$$I = -n\sum_{i=1}^{m} p_i \log_a p_i . \qquad (2.2)$$

where m – the number of characters in the alphabet, n – the number of alphabet characters in the message, p_i – the probability (relative frequency) of a symbol with a value *i* in the message.

It turned out that the formula proposed by R. Hartley is a special case of the more general Shannon's formula. If Shannon's formula assumes that $p_1 = p_2 = \ldots = p_i = \ldots = p_m$, then

$$I = -n\sum_{i=1}^{m} p_i \log_a p_i = -nm\left(\frac{1}{m}\log_a \frac{1}{m}\right) = -n\frac{m}{m}\log_a \frac{1}{m} = -n\left(\log_a 1 - \log_a m\right) = n\log_a m .$$

A minus sign in Shannon's formula does not mean that the amount of information in the message is negative. This is because of the probability p_i, by definition, less than one but greater than zero. It is known that the logarithm of a number less than one, i.e. $\log p_i$ – value is negative, then the product of the probability on the logarithm of the number will be positive.

Entropy as a measure of choice uncertainty

Entropy is a logarithmic measure of the disorder of the state of a message source and characterizes the average degree of uncertainty of the state of this source.

Entropy H is determined by a formula according to C. Shannon's theorem, on the basis of which, the average amount of information per symbol is determined $H = -\sum_{i=1}^{m} p_i \log p_i$. In information systems, uncertainty is reduced due to the received information, so numerically entropy H equal to the amount of information I, that is, it is a quantitative measure of information.

Basic properties of entropy:

1) Entropy is a quantity real and non-negative, because the value of probabilities p_n are in the range $0 \div 1$, value $\log p_n$ always negative, and values $-p_n \log p_n$ respectively positive.

2) The value of entropy is limited, because when $p_n \geq 0$ value $-p_n \log p_n$ also goes to zero, and when $0 < p_n < 1$ the limited sum of all terms is obvious.

3) Entropy is 0 if the probability of one of the states of the source of information is 1, and thus the state of the source is completely determined (the probabilities of other states of the source are zero, because the sum of probabilities must be 1).

4) Entropy of a source with two states u_1 and u_2 when changing the ratio of their probabilities $p(u_1)$ та $p(u_2)$ is determined by the expression $H(U) = -\left[p \log p + (1-p) \log p\right]$ and reaches a maximum with equal probabilities.

5) The entropy of the combined statistically independent sources of information is equal to the sum of their entropies.

6) Entropy is maximum with equal probability of all states of the information source: $H(U)_{\max} = -\left(\frac{1}{N}\right) \log \frac{1}{N} = \log N$.

In this particular case, Shannon's quantitative measure coincides with Hartley's. If the messages are unequal, then the average amount of information contained in one message will be less.

Entropy of union

The union entropy is used to calculate the entropy of the co-occurrence of statistically dependent messages. For example, by transmitting the number 5 a hundred times over the interference channel, it was noted that the number 5 was received 90 times, the number 6 to 8 times, and the number 4 to 2 times. The uncertainty of the occurrence of combinations of the form 5-4, 5-5, 5-6 in the transfer of the number 5 can be described by the entropy of the union.

The union entropy – is the uncertainty of what will be sent to *A* and received by *B*. For message and received message ensembles, the union entropy is the sum of the form.

$$H(A,B) = -\sum_i \sum_j p(a_i,b_j)\log_2 p(a_i,b_j). \quad (2.3)$$

The entropy of the union and the conditional entropy are related by the following relations:

$H(A,B) = H(A) + H(B/A) = H(B) + H(A/B)$,

$H(B/A) = H(A,B) - H(A)$,

$H(A/B) = H(A,B) - H(B)$.

Properties of the entropy of the union:

1) The property of symmetry: $H(A,B) = H(B,A)$.

2) In the absence of statistical dependence between the elements of ensembles *A* and *B* conditional probabilities are converted into unconditional, then $H(A,B) = H(A) + H(B)$.

3) With a full statistical relationship between the elements of ensembles *A* and *B* (for example, when the result of one event unambiguously determines the information about the second event) conditional entropies are zero, so $H(A,B) = H(A) = H(B)$.

Conditional entropy and its properties

In practice, interdependent symbols and messages are common. When conveying meaningful messages, some letters are more common, others less common, some letters and words often follow others. For example, the letter e is most often used in English; in French, the letter q is always followed by the letter u if q is not at the end of the word. Regarding interacting systems, mainly the state of one of them affects the state of the other. In such cases, the entropy cannot be determined solely on the basis of unconditional probabilities.

When calculating the average amount of information per message symbol, the interdependence is taken into account due to the conditional probabilities of occurrence of some events relative to others. And the resulting entropy is called conditional entropy.

Conditional entropy is used to determine the relationship between the characters of the primary alphabet, to determine the loss of information transmission through communication channels, when calculating the entropy of the union.

Consider a text message. If when transmitting n messages, the symbol A appears m times, the symbol B appears l times, and the symbol A together with the symbol B appears k times, then the probability of the appearance of the symbol A $p(A)=m/n$; the probability of the symbol B $p(B)=l/n$; the probability of the combined appearance of the symbols A та B $p(AB)=k/n$.

Conditional probability of occurrence of a symbol A relative to the symbol B $p(A/B)=\frac{p(AB)}{p(B)}=\frac{k}{l}$ and Conditional probability of occurrence of a symbol B relative to the symbol A $p(B/A)=\frac{p(AB)}{p(A)}=\frac{k}{m}$.

If you know the conditional probability, you can easily determine the probability of the combined appearance of the characters, using the previous formulas: $p(A,B)=p(B)p(A/B)=p(A)p(B/A)$.

Consider the formulas of conditional entropy using Shannon's formula:

$$H(b_j/a_i)=-\sum_j p(b_j/a_i)\log_2 p(b_j/a_i), \tag{2.4}$$

$$H(a_i/b_j)=-\sum_i p(a_i/b_j)\log_2 p(a_i/b_j), \tag{2.5}$$

where the index i is selected to characterize the arbitrary state of the message source A, the index j is selected to characterize the arbitrary state of the recipient B.

There are concepts of partial and general conditional entropy. Expressions (2.4) and (2.5) are partial conditional entropies.

The total conditional entropy of message B relative to message A characterizes the amount of information contained in any character of the alphabet, and is defined as the sum of the probabilities of the characters of the alphabet multiplied by the uncertainty that remains after the recipient received the signal:

$$H(B/A)=-\sum_i p(a_i)H(b_j/a_i)=-\sum_i\sum_j p(a_i)p(b_j/a_i)\log_2 p(b_j/a_i). \tag{2.6}$$

Expression (2.6) is general for determining the amount of information per message symbol for the case of unequal and interconnected characters. Because $p(a_i)p(b_j/a_i)$ is the probability of joint occurrence of two events $p(a_i,b_j)$, then formula (2.6) can be written as follows:

$$H(B/A) = -\sum_i \sum_j p(a_i,b_j)\log_2 p(b_j/a_i). \qquad (2.7)$$

Properties of conditional entropy:

1) If the message ensembles A and B are interdependent, then the conditional entropy A with respect to B is equal to the unconditional entropy A and vice versa: $H(A/B) = H(A)$; $H(B/A) = H(B)$.

2) If the ensembles of messages A and B are so rigidly statistically related that the appearance of one of them means the obligatory appearance of the other, then their conditional entropies are zero: $H(B/A) = H(A/B) = 0$.

Entropy of a continuous source of information (differential entropy) and its properties

The concept of differential entropy is used to describe the information properties of a continuous source (signal). To obtain it, we use the formula for the entropy of discrete messages. In this case, the expression for entropy can be represented as:

$$H(X) = -\sum_{i=1}^{m} p(x_i)\log_2 p(x_i) = -\sum_{i=1}^{m} f(x_i)\Delta x \log_2 \left[f(x_i)\Delta x\right] =$$

$$= -\sum_{i=1}^{m}\left[f(x_i)\log_2 f(x_i)\right]\Delta x - \log_2 \Delta x \sum_{i=1}^{m} f(x_i)\Delta x = H(\Delta x)$$

Move to the border: $H(X) = \lim_{\Delta x \to 0} H(\Delta X) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx - \lim_{\Delta x \to 0}\log_2 \Delta x$.

The total entropy of the source of continuous messages consists of two terms, one of which is determined by the distribution law, and the other is a constant value that determines the quantization step, which affects the accuracy of measurements. This member of the expression defines a constant component and is excluded from consideration.

The value of the first term is determined by the distribution law and characterizes the differential entropy of a continuous source (because $f(x)$ – probability density or differential distribution law) $h(x) = -\int_{-\infty}^{\infty} f(x)\log_2 f(x)dx$.

Differential entropy – is the part of the entropy of the source of continuous messages, which depends on the probability density of the signal emitted by the source.

Continuous distributions with maximum differential entropy

Different classes of physical phenomena and processes are subject to different laws of distribution. Continuous signals are fully characterized by the laws of distribution (integral or differential). There are certain restrictions on any real signals. Since the differential entropy depends on the probability density, we determine for which law it is maximal. That is, at what probability distribution, the signal of a given power has the maximum entropy.

The differential entropy for the normal distribution (Fig. 2.1) (signal with limited average power) does not depend on the mathematical expectation and is equal to the entropy of the centered random variable. Maximum value for entropy: $h(x) = \log_2 \sigma\sqrt{2\pi e}$.

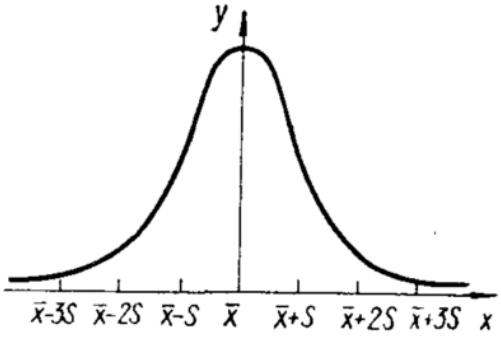


Fig. 2.1

Determine the differential entropy for uniform distribution (Fig. 2.2), ie the signal with limited peak power. If P – peak power, then $A = \sqrt{P}$ – amplitude value. The differential entropy for uniform distribution is equal to: $h(x) = \log_2 2A$.

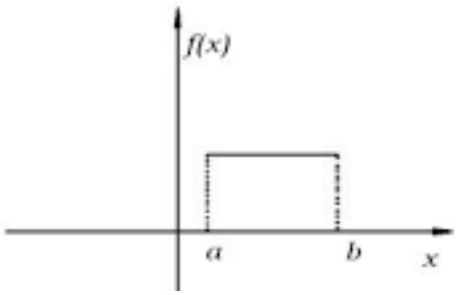


Fig. 2.2

Differential entropy for the exponential distribution (Fig. 2.3) (the distribution is widely used to determine the failure rate of electronic equipment): $f(x) = ke^{-kx}$ by $x > 0$,

$$h(x) = \int_0^\infty ke^{-kx}\left(\log_2 k - kx\log_2 e\right)dx = -\log_2 k\int_0^\infty ke^{-kx}dx + \log_2 e\int_0^\infty xke^{-kx}dx = -\log_2 k + \log_2 e = \log_2\frac{e}{k}.$$

The total entropy for the exponential distribution is equal to: $H(x) = h(x) - \log_2\Delta x = \log_2\frac{e}{k\Delta x}$.

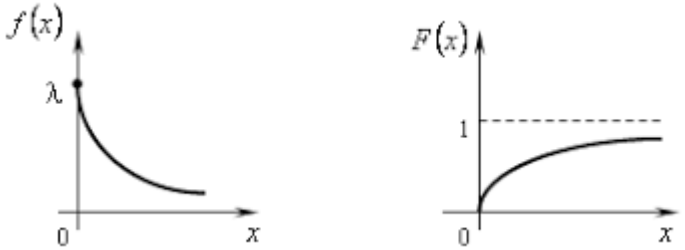


Fig. 2.3

Lecture №3. Information characteristics of discrete message sources

Discrete message – is a letter, number, or countdown that appears after a continuous signal is sampled and quantized.

Source information characteristic – is a distribution of the amount of information that falls on each message of the source alphabet. The source data for the calculation of this characteristic is the probability of occurrence of the messages of the source alphabet.

To compare sources whose alphabet has a different number of messages, the information characteristic is not convenient. Thus, the information characteristic performs an illustrative function. But like any other characteristic, the information characteristic makes it possible to determine the parameters of the source.

1) The average value of the amount of information per source message.

Consider the method of determining the average value of the amount of information that falls on one source message on the example of an information source, the alphabet of which is 2 messages "0" and "1". Let *P(1)* and *P(0)* be the probabilities of the messages "1" and "0", respectively. Then the amount of information in each occurrence of the message "1" (or "0") And (1) (or And (0)) is determined as: $I(1) = \log_2 \frac{1}{P(1)}$, $I(0) = \log_2 \frac{1}{P(0)}$.

Consider a sequence of messages, for example, 1101111000 and denote the number "1" in this sequence – *n(1)* and the number "0" – *n(0)*. Now the average amount of information per message of this sequence of messages can be calculated by the formula: $\overline{I}(a) = \frac{n(1)I(1) + n(0)I(0)}{n(1) + n(0)} = P(1)I(1) + P(0)I(0)$.

Based on this formula, we form a formula for determining the average value of the amount of information per one source message for an arbitrary number of source messages *m*: $\overline{I}(a) = \sum_{i=1}^{m} P(a_i) \log_2 \frac{1}{P(a_i)}$.

For practical use, the parameter the average value of the amount of information per source message has a shorter name - the entropy of the source.

Redundancy of messages

When the information from the source of information is issued in the form of written text and in this text there are a small number of errors in the form of missing letters, then from such text you can remove the transmitted information. From this observation we can conclude that the source of information gives significant (existing letters) and insignificant (missing letters) information. When transmitting information in the form of text, insignificant information is redundant. This effect is called **information source redundancy**.

For a source with a real distribution of message probabilities, we have the real value of the entropy of the source $H(A)$ and we can determine the maximum possible value of the entropy of the source $H_{\max}(A)$. And so we can determine the difference $H_{\max}(A) - H(A)$. This parameter is called the **redundancy of the source**. But this option is not convenient for sources with different numbers of messages ($H_{\max}(A)$) for them will be different). This inconvenience is eliminated by normalization of this indicator rather $H_{\max}(A)$: $\rho = \frac{H_{\max}(A) - H(A)}{H_{\max}(A)} = 1 - \frac{H(A)}{H_{\max}(A)}$.

Parameter ρ is called the source redundancy factor. Note that the value of the redundancy factor of the source is within $0 \le \rho \le 1$.

The redundancy factor characterizes the share of information from the source, which can not be transmitted and there will be no loss of information. In other words, the source has essential and insignificant information. Source redundancy – is actually insignificant source information. However, redundant information can be used on the receiving side of the SPI to correct errors, if such are included in the information during its transmission through the communication channel.

The source of information produces a certain amount of information that can be presented in two ways:

1) $n \cdot H(A)$, where n – the number of messages (symbols) is required to transmit a given amount of information; $H(A)$ – real source entropy.

2) $n_0 \cdot H_{\max}(A)$, where n_0 – the minimum number of messages (symbols) required to transmit a given amount of information; $H_{\max}(A)$ – maximum entropy of the source.

These two products represent the same amount of information. So on their basis we can make the equation: $n_0 \cdot H_{\max}(A) = n \cdot H(A)$. From this equation we get that $\frac{H(A)}{H_{\max}(A)} = \frac{n_0}{n}$. Then $\rho = 1 - \frac{n_0}{n} = \frac{n - n_0}{n}$.

Use can be twofold:

1) Knowing the presence of redundancy at the source, you can solve the problem of eliminating redundant information, ie compression of information.

2) Excess (insignificant) information can be used on the receiving side to correct errors, if they are included in the signal during its transmission via the communication channel.

Productivity of discrete message sources

The speed of information (similar to the concept of vehicle speed) is represented by the ratio of the amount of information contained in the sequence of messages a_T to the duration of this sequence T. Given the fact that the rate of information is a statistical value, to increase the reliability of the calculation result it is necessary to increase the value T. The mathematical expression for the rate of appearance of information based on what has been said will look like this: $R = \lim_{T \to \infty} \frac{I(a_T)}{T}$, $\left[\frac{d.o}{s}\right]$ or $\left[\frac{bit}{s}\right]$.

We will study the above expression in order to determine the conditions when the rate of information will be the maximum value. To do this, we present: $I(a_T) = nH(A)$; $T = n\overline{\tau}$, where a_T – sequence of messages at the output of the source in the time interval T; n – number of messages in the sequence a_T; $\overline{\tau}$ - the average value of the duration of one message. Then $R = \lim_{n \to \infty} \frac{nH(A)}{n\overline{\tau}} = \frac{H(A)}{\overline{\tau}}$.

If the source messages are independent $\overline{\tau}$ defined as $\overline{\tau} = \sum_{i=1}^{m} \tau_i P(a_i)$, where τ_i – duration *i*-th message; *m* – source alphabet (number of types of messages used to issue information from the source); $P(a_i)$ – probability of the *i*-th message.

If the messages have a fixed duration, then $\overline{\tau} = \tau$, then $R = \frac{H(A)}{\tau}$.

Then the performance of the source of discrete messages can be recorded as follows: $R_D = \max\left\{\frac{H(A)}{\overline{\tau}}\right\}$.

In turn, the attitude $\frac{H(A)}{\tau}$ will have the maximum value if $H(A) = H_{\max}(A)$, and $\tau = \tau_{\min}$, then $R_D = \frac{H_{\max}(A)}{\tau_{\min}}$.

All of the above about message source settings takes place provided that the source messages are independent.

If we take the letters of the Ukrainian alphabet as an example of messages, we can establish that the probability of occurrence of each letter depends on which letter was the previous one. For example, we pass a letter «ь» The next letter can't be «ь» (when transmitting meaningful text). That is, in the general case, it should be understood that the source transmits dependent messages.

The condition of independent messages is fulfilled when the messages represent samples of a continuous random signal, the interval between which Δt equal to or greater than the correlation interval τ_K ($\Delta t \geq \tau_K$). In case when $\Delta t < \tau_K$, readout are dependent messages. This condition is mandatory for the implementation of digital methods of continuous signal transmission: differential pulse code modulation, encoding of predicted values, delta modulation.

Examples

1. The probability distribution of a discrete random variable has the form $p(x_1) = 0{,}102584$, $p(x_2) = 0{,}12$, $p(x_3) = 0{,}1$, $p(x_4) = 0{,}1$, $p(x_5) = 0{,}1$, $p(x_6) = 0{,}09$, $p(x_7) = 0{,}07$, $p(x_8) = 0{,}317416$. Determine the number *n* values of a random

variable at which the entropy $H_p(X)$ the uniform distribution will be equal to the entropy $H(X)$ of the given distribution.

Determine the entropy of a given distribution. To find the entropy of a given discrete ensemble, we use the formula that determines the entropy.

Calculate the entropy of the distribution: $H(X)=\sum_{k-1}^{N}P(X_k)\log_2\frac{1}{P(X_k)}$.

$$H(X)=p(x_1)\log_2\frac{1}{p(x_1)}+p(x_2)\log_2\frac{1}{p(x_2)}+p(x_3)\log_2\frac{1}{p(x_3)}+p(x_4)\log_2\frac{1}{p(x_4)}+$$

$$+p(x_5)\log_2\frac{1}{p(x_5)}+p(x_6)\log_2\frac{1}{p(x_6)}+p(x_7)\log_2\frac{1}{p(x_7)}+p(x_8)\log_2\frac{1}{p(x_8)}=$$

$$=0{,}102584\log_2\frac{1}{0{,}102584}+0{,}12\log_2\frac{1}{0{,}12}+3\left(0{,}1\log_2\frac{1}{0{,}1}\right)+0{,}09\log_2\frac{1}{0{,}09}+$$

$$+0{,}07\log_2\frac{1}{0{,}07}+0{,}317416\log_2\frac{1}{0{,}317416}=0{,}33+0{,}36+3\cdot0{,}33+0{,}31+0{,}26+0{,}52=$$

$$=2{,}77(bit)$$

Uniform distribution implies levels of probability of all possible results, with entropy $H_p(X)=\log_2 n$.

Provided that $H(X)=H_p(X)$ we find: $\log_2 n=2{,}77$, $n=2^{2{,}77}=7$.

So, with the volume of the alphabet $n = 7$, entropy $H_p(X)$ the uniform distribution will be equal to the entropy $H(X)$ given distribution.

1. Find the entropy of noise $H(U/Z)$ in a binary symmetric channel without memory, if the entropy of the source at the input of the channel $H(U)=3400$ *(bit),* the entropy of the ensemble at the output of the channel $H(Z)=6800$ *(bit),* channel unreliability $H(U/Z)=700$ *(bit).*

We will look for the entropy of noise in a binary symmetric channel without memory by the formula $H(U/Z)=H(Z)-I(U/Z)$.

Let us express the amount of information $I(U/Z)=H(U)-H(U/Z)$. Substituting this formula into the original, we obtain the expression for finding the noise entropy in the binary symmetric channel $H(U/Z)=H(Z)-H(U)+H(U/Z)=6800-3400+700=4100(bit)$.

Noise entropy in a binary symmetric channel $H(U/Z)=4100(bit)$.

2. The receiver signal may have an amplitude A_1 (event X_1) and A_2 (event X_2), as well as phase shift φ_1 (event Y_1) or φ_2 (event Y_2) in different modes. The probabilities of joint events have the following meanings: $P(X_1,Y_1)=0,73$; $P(X_1,Y_2)=0,21$; $P(X_2,Y_1)=0,02$; $P(X_2,Y_2)=0,04$. Calculate the amount of information obtained about the phase shift of the signal, if its amplitude becomes known.

The amount of information about the phase shift at a known amplitude will be sought by the formula $I(Y/X)=H(Y)-H(Y/X)$. Find the entropy Y: $H(Y)=P(Y_1)\log_2\frac{1}{P(Y_1)}+P(Y_2)\log_2\frac{1}{P(Y_2)}$. To find the entropy, we find the probabilities of occurrence of events Y_1 і Y_2: $P(Y_1)=P(X_1,Y_1)+P(X_2,Y_1)=0,73+0,02=0,75$; 11. Find the probabilities of events X_1 і X_2: $P(X_1)=P(X_1,Y_1)+P(X_1,Y_2)=0,73+0,21=0,94$; $P(X_2)=P(X_2,Y_1)+P(X_2,Y_2)=0,02+0,04=0,06$. Substituting the values, we find the value of entropy: $H(Y)=0,75\log_2\frac{1}{0,75}+0,25\log_2\frac{1}{0,25}=0,81(bit)$.

Find:

$$H(Y/X)=\sum_{k=1}^{N}\sum_{j=1}^{M}P(Y_k/X_j)\log_2\frac{1}{P(Y_k/X_j)}=$$

$$=P(Y_1/X_1)\log_2\frac{1}{P(Y_1/X_1)}+P(Y_1/X_2)\log_2\frac{1}{P(Y_1/X_2)}+.$$

$$+P(Y_2/X_1)\log_2\frac{1}{P(Y_2/X_1)}+P(Y_2/X_2)\log_2\frac{1}{P(Y_2/X_2)}$$

$P(Y_1/X_1)=\frac{P(Y_1,X_1)}{P(X_1)}=\frac{0,73}{0,94}=0,78$; $P(Y_2/X_1)=\frac{P(Y_2,X_1)}{P(X_1)}=\frac{0,21}{0,94}=0,22$;

$P(Y_1/X_2)=\frac{P(Y_1,X_2)}{P(X_2)}=\frac{0,02}{0,06}=0,33$; $P(Y_2/X_2)=\frac{P(Y_2,X_2)}{P(X_2)}=\frac{0,04}{0,06}=0,67$.

Substituting the obtained values into the formula, we obtain that $H(Y/X)=0,78\log_2\frac{1}{0,78}+0,22\log_2\frac{1}{0,22}+0,33\log_2\frac{1}{0,33}+0,67\log_2\frac{1}{0,67}=$. Then, $=0,28+0,48+0,53+0,39=1,68$

the amount of information about the phase shift at a known amplitude will be equal to $I(Y / X) = H(Y) - H(Y / X) = 1,68 - 0,81 = 0,87$.

Lecture №4. Classification of sensors, main characteristics. Sensors of temperature, pressure, humidity, light radiation and others. Sensor devices as components of electronic systems

Today there are several definitions of " sensor ". Usually the definitions correspond to the practice of using the term by manufacturers of sensors.

Let's focus on some of the definitions of the sensor:

– sensitive element that converts the parameters of the environment (external action) into a suitable for technical use electrical signal;

– finished product based on the above sensing element, which, depending on the need, includes intermediate measuring transducers for generating an electrical signal in a form convenient for transmission, further conversion and processing and an interface for integration into control systems.

In the first case, the sensor is a small, usually monolithic electronic device, such as a thermistor, photodiode, etc. which is used to create more complex electronic devices.

In the second case, it is a complete device, which is connected via one of the known interfaces to the automatic control or registration system; in this case, the sensing element of the sensor itself may be called a sensor (from Latin *sensorium* – sense organ), which is currently a trend in automated production.

Block diagram of the sensor shown in Fig. 4.1.

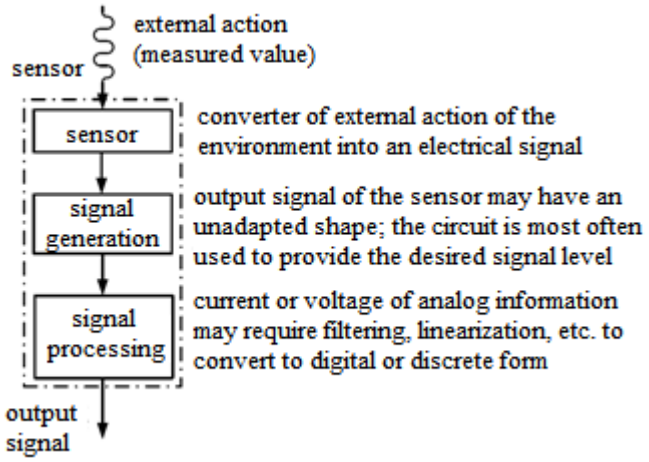


Fig. 4.1

The output signals of the sensor can be standard analog current or voltage signals, a discrete signal or a digital binary code. This set of characteristics is called the output signal format. Thus, each sensor is characterized by a set of input parameters, which can be of any physical nature and a set of output electrical parameters. In fig. 4.2 shows the principle of hardware implementation of signal converters in a certain form at the output.

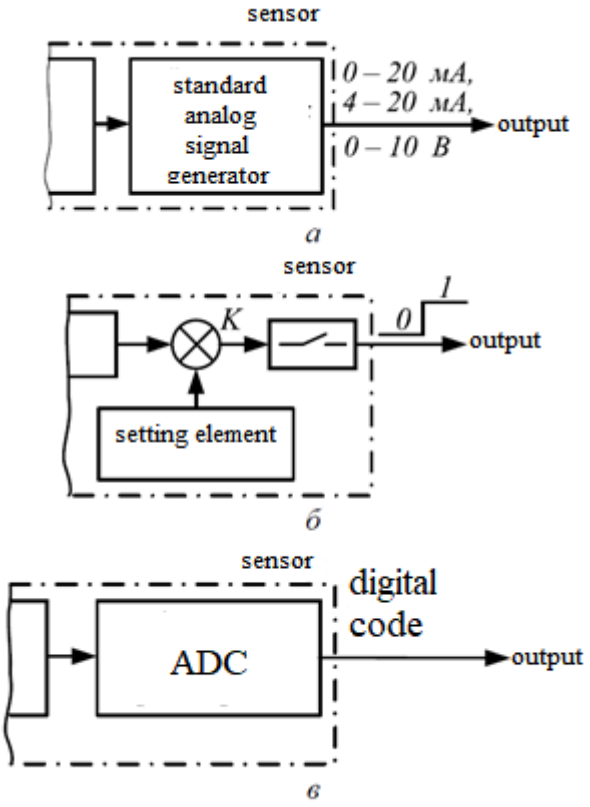


Fig.4.2

а – generating standard analog signals;

б – formation of a discrete signal;

в – digital code generation.

In fig. 4.2. *K* – comparative threshold element (comparator).

There are direct action sensors and composite sensors.

Direct-acting sensors convert external **action directly** into an electrical signal, using the appropriate physical phenomena. Examples of direct-acting sensors are a photodiode, a piezoelectric element, a thermistor, and a Hall sensor.

In **folded** sensors, several energy conversions must be performed before receiving an electrical signal at the output of the final direct-acting sensor. For example, a chemical sensor may include two transducers, one of which converts the energy of chemical reactions into heat, and the other, a thermocouple, converts the resulting heat into an electrical signal. Thus, the combination of two transducers is a chemical sensor - a device that produces an electrical signal in response to a chemical reaction. As a rule, the structure of composite sensors includes at least one direct-acting sensor and several transducers.

Historically and logically, sensors are associated with measurement techniques and measuring instruments, such as thermometers, flow meters, barometers, and so on. The generalizing term sensor was fixed in connection with the development of automatic control systems as an element of the generalized logical concept: sensor - control device - actuator (engine, valve, etc.) - control object (Fig. 4.3).

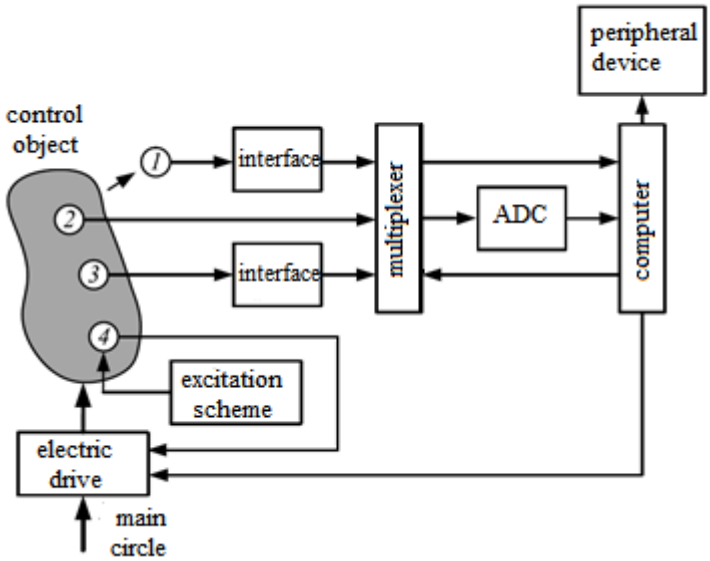


Fig. 4.3

1 – contactless sensor;

2, 3 – active sensors;

4 – passive sensor.

Classification of sensors, main characteristics

Sensor classification systems can vary from very simple to complex. The classification criterion is always chosen depending on the purpose of the classification.

Classification by **type** of sensor:

- active (generator). The **active** sensor does not require an additional power source and an electrical signal always appears at its output in response to a change in external action. Examples of active sensors are thermocouples, photodiodes, and piezoelectric sensors. Active sensors are called generators: they have a difference in electrical potentials due to external action at the output, which is directly the output signal.
- passive (parametric). In contrast to the active **passive** sensor for its operation requires external energy, which is called the excitation signal. During the formation of the output signal, the passive sensor in one way or another acts on the excitation signal. Because such sensors change their characteristics in response to a change in external action, they are sometimes called parametric. In passive sensors there is a change in their parameters, which simulate the excitation signals, and this modulation carries information about the change in external action.

Classification according to the choice of **reference point**:

- absolute. The **absolute** sensor detects the external signal in absolute units, which do not depend on the measurement conditions.
- relative. The output signal of the **relative** sensor in each case can be interpreted differently.

Classification by **external action** (measured value):

- pressure sensors:
    - absolute pressure;
    - relative pressure;
    - rarefaction;
    - pressure-vacuum;
    - pressure differences;

    - hydrostatic pressure;
- cost sensors:
    - mechanical flow meters;
    - difference meters;
    - ultrasonic flow meters;
    - electromagnetic flowmeters;
    - coriolis flowmeters;
    - vortex flowmeters;
- level sensors:
    - float;
    - capacitive;
    - radar;
    - ultrasonic;
- temperature sensors:
    - relative (sensor - thermocouple);
    - absolute (sensor - thermistor);
    - pyrometer;
- concentration sensors:
    - conductometers;
- radioactivity sensors (radioactivity or radiation detectors):
    - ionization chamber;
    - direct charge sensor;
- motion sensors:
    - absolute encoder;
    - relative encoder;
    - LVDT (adjustable linear differential converter);
- position switches:
    - contact;
    - contactless;
- angular position sensors:
    - selsin;

- converter angle - code (encoder);
- RVDT (adjustable rotary differential converter);

– vibration sensors:
  - piezoelectric;
  - vortex current;

– sensor of mechanical quantities:
  - relative expansion;
  - absolute expansion.

Classification by **characteristics**:

– sensitivity;
– stability;
– accuracy;
– hysteresis;
– speed;
– range of input values;
– output signal format;
– others.

Classification by the **number of input values**:

– one-dimensional;
– multidimensional.

Classification by **principle of action**:

– photoelectric (optical);
– magnetoelectric (based on the Hall effect);
– piezoelectric;
– strain gages;
– capacitive;
– potentiometric;
– inductive;
– induction;
– ultrasonic.

Classification according to the **principle of realization of output converters**:

- with discrete output:
  - static;
  - electromechanical;
- with analog output;
- with digital output.

Classification by **design**:

- solid, otherwise compact, or monoblock;
- composed (consists of separate structural parts), ie acts as a functional group, otherwise, for example, two-block.

Location classification:

- Sensors can be in direct contact with the object: located on the surface or inside the object or not in direct contact with it.
- Sensors that are not in direct contact with the object are called non-contact. In such sensors, the energy exchange between the sensor and the object can be carried out by means of a magnetic field (non-contact inductive proximity sensor), an ultrasonic signal (non-contact ultrasonic presence signal), etc.

Temperature sensors

Temperature sensors are designed for continuous measurement and control of temperature on the surface or in the environment of solids, in liquid and gaseous media (steam, water, bulk materials, chemical reagents, etc.).

Temperature measurement is always the transfer of a small portion of thermal energy from an object to a sensor, which must convert that energy into an electrical signal. When the sensor of the contact sensor is located inside or on the object, heat is transferred between the object and the sensor due to the phenomenon of heat transfer. The sensitive element as a component of the sensor is either heated or cooled. The same happens during heat transfer by radiation in the case of a contactless sensor.

Contactless sensors are used for quick temperature control without the need for installation on site, as well as in cases where such installation is impossible (flame, molten metal, etc.). Such sensors may be without an output signal or have it).

There are two main methods of measuring temperature: equilibrium and predictive. The **equilibrium** method of temperature measurement is performed when there is a thermal equilibrium between the measuring surface and the sensing element located in the sensor housing, ie there is no significant temperature difference between the sensor and the object of measurement. In the **prediction** method, thermal equilibrium does not occur during the measurements, and the temperature value is determined by the rate of change of the sensor temperature.

In the case of the equilibrium method of temperature measurement, it theoretically takes an infinitely long time to achieve complete thermal equilibrium between the object and the sensor. But since temperature measurements are usually performed with a given accuracy, in most cases it is assumed that in the interval from 5 to 10 time constants τ a quasi-equilibrium state occurs, taking into account, of course, that the transient temperature setting process usually corresponds to the exponential characteristic.

A typical contact sensor consists of the following components:

- sensitive element;
- terminals, which can be plates or conductors, and in the presence of a switching head, they are connected to the appropriate terminals;
- protective case - a special shell or coating that protects the sensitive element from the environment.

A typical non-contact optical temperature sensor consists of the following parts:

- sensitive element that responds to electromagnetic radiation;
- support structure with low thermal conductivity;
- sealed housing filled with inert gas (argon or nitrogen) or dry air;
- protective window transparent to the radiation of the studied wavelength range.

The temperature measured by the sensors can act as an analog line signal (current and / or voltage), as a digital code or as a logic signal ("On" / "Off"). In the case of a logic signal at the output of temperature sensors, they are usually called relay sensors and are considered as temperature control relays.

Temperature sensors can be single-block or double-block. In the case of the **two-block** sensor, which occurs more often, it consists of two functional parts: a sensor and a converter (actuator).

Depending on the measurements for which the contact sensor is intended, and how accurate the measurement result will be, in practice sensors based on three different types of sensing elements are used:

- resistance thermocouples;
- thermistors;
- thermoelectric converters.

Thermistors are essentially thermocouples of resistance. The principle of operation of both is to change the resistance of the conductor, or in the case of a thermistor - a semiconductor, from the temperature. The term thermistor was formed by combining two words: thermal and resistor.

Resistance thermocouples are made of copper or platinum; they are characterized by high accuracy (up to 0.1 ° C), stability of readings, closeness of the characteristic to linear, interchangeability.

Pressure sensors

A pressure sensor is a device whose physical parameters change depending on the pressure of the measured medium (liquid, gas, vapor). In pressure sensors, the pressure of the measured medium is converted into a logical (discrete), standard analog or digital signal.

The principle of operation of any pressure sensor is the primary conversion of the pressure acting on the sensor into an electrical signal. The design of almost all pressure sensors includes sensors that have a known surface area, the deformation or displacement of which under the action of pressure is determined in the measurement process. That is, many pressure sensors are implemented on the basis of sensors of displacement elements or force, the cause of which is also displacement (folded sensors).

In General, the pressure sensor consists of a primary pressure transducer, which contains a sensing element and a pressure receiver, a circuit for secondary signal processing (actuator), different in design housing parts and an output device.

The main difference between some sensors and others is the accuracy of pressure recording, which depends on the principle of converting pressure into an electrical signal: strain gauge, piezoresistive, capacitive, inductive, resonant, ionization, etc.

Pressure sensors are classified according to:

- type of measured pressure:
    - absolute pressure - designed to measure the absolute pressure of liquid and gaseous media. The reference pressure is a vacuum. Air is pumped out of the inner cavity of the sensor sensor. For example, a barometer (Fig. 102, a) is a special case of an absolute pressure sensor;
    - excess pressure - designed to measure the value of excess pressure (Fig. 102, b) of liquid and gaseous media. Reference pressure - atmospheric, ie one side of the membrane is connected to the atmosphere;
    - differential pressure - designed to measure the pressure difference of the medium and are used to measure the flow of liquids, gas, steam, liquid level. The pressure is applied to both sides of the membrane, and the output signal depends on the pressure difference;
- specification:
    - hydrostatic pressure - designed to measure the value of the hydrostatic pressure of the controlled medium and convert it into a level value. The pressure of a liquid column depends only on its height and the density of the liquid itself, but does not depend on the shape and volume of the tank. They can be excess or differential pressure sensors to which the pressure of the medium is applied and for comparison the other inlet is connected to the atmosphere or to the region of excess pressure in the case of a pressure tank. Hydrostatic pressure sensors are used to measure the level of homogeneous liquids in tanks without significant liquid movement and can be used for viscous liquids, suspensions and pastes. Structurally hydrostatic sensors are of two types: membrane and submersible. In the first case, the strain gauge or capacitive sensor is directly connected to the membrane and the entire sensor is located in the lower part of the tank, while the location of the sensor membrane

corresponds to the minimum level. In the case of an immersion sensor, the membrane is in the liquid and transmits pressure to the strain gauge sensor through the column. Hydrostatic sensors provide high accuracy at low cost and simplicity of design;

- vacuum gauge pressure (vacuum) - designed to measure the value of vacuum gauge pressure. The reference pressure in these sensors is atmospheric. However, unlike overpressure sensors, the measured pressure is less than atmospheric, ie there is a vacuum relative to the atmosphere;
- excess pressure-vacuum - a combination of sensors of excess and vacuum pressure, ie measure both pressure and vacuum;

– constructive execution:
  - external pressure sensors;
  - built-in pressure sensors;
– principle of action:
  - direct action - convert external influences directly into an electrical signal, using the appropriate physical phenomenon;
  - folded pressure sensors - contain several transducers of external action of the environment;
– the principle of converting pressure into an electrical signal:
  - strain gages;
  - piezoresistive;
  - capacitive;
  - resonant;
  - inductive;
  - piezoelectric;
  - ionization, etc;
– nominative measuring range:
  - high and ultra-high pressure (p> 60 MPa);
  - low and ultra-low pressure (p <0.1 MPa);
  - average pressure (0.1 <p <60 MPa);

- type of measured medium:
  - non-aggressive gases and liquids;
  - aggressive gases and liquids;
  - nutrient environments;
  - viscous environments;
  - abrasive environments;
- type of output signal:
  - with analog;
  - with digital;
  - with logical.

Sensor devices as components of electronic systems

Using the term "sensor" in the industry increasingly deviate from the term "converter" and the totality of all converters, except the sensor, which generates and processes the output signal, convenient for use in automated control and registration systems, called actuator (from the English. Actuator - actuator)). In this case, the block diagram of the sensor looks like in Fig. 4.4.

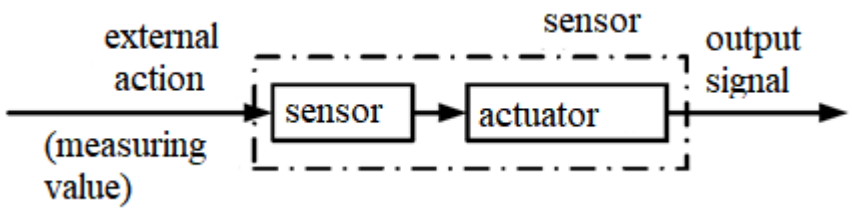


Fig. 4.4

Lecture №5. Deterministic and random signals, their time and frequency representation. Spectra of discrete signals, choice of frequency sampling

Deterministic signals – are signals that can be described by explicit mathematical dependencies and whose values at any point in time or at any point in space (or depending on any other arguments) are a priori known, or can be determined quite accurately.

Signals, the laws of change of which cannot be described by explicit mathematical dependences, because they are random, are **random**. Mostly their totality is estimated by the statistical characteristics of the process that they form, and is characterized by probability distribution laws, correlation functions, spectral energy densities. Real signals are always random. In practice, quasi-deterministic signals are mostly described by functions with unknown random parameters.

An example of a deterministic signal is a sinusoidal wave, which is graphically described by a sine wave. The sine wave together with triangular, sawtooth, rectangular and other signals belongs to periodic signals, ie to such signals which repeat the form in some constant period of time.

The **form** of signals is divided into continuous and discrete. Continuous signals can take a continuous set of values (continuum) in a certain interval (in time and level). Discrete signals are described by a finite set of numbers or discrete values of a particular function. Continuous signals (Fig. 5.1, a) are represented by a function that is continuous in time on the observation segment, and discrete (Fig. 5.1, b) arrive only at certain points in time and are represented by a discrete function.

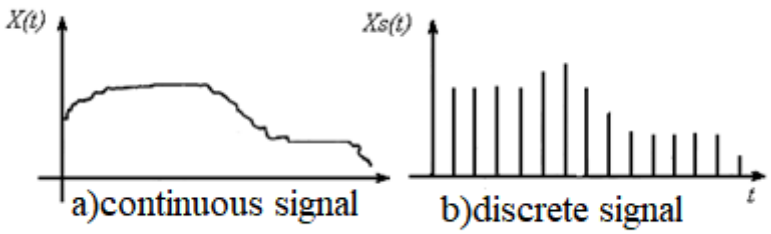


Fig. 5.1

The elementary deterministic signals include, in particular, a single function (jump).

The set of amplitudes and corresponding frequencies of harmonics is called the spectrum of amplitudes (Fig. 5.2, a). The set of initial phases and the corresponding frequencies of harmonics are called the phase spectrum (Fig. 5.2, b). The spectrum of amplitudes and the spectrum of phases unambiguously determine the signal, but for many practical problems it is enough to consider only the spectrum of amplitudes.

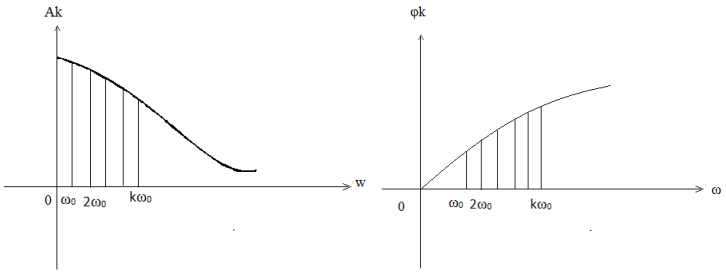


Fig. 5.2

A characteristic feature of the spectrum of the periodic signal is its discontinuity (discreteness). The distance between adjacent spectral lines is the same and equal to the fundamental harmonic frequency.

Probability theory methods are used to describe **random processes**. In the general case, the complete characteristic of a random process is its multidimensional probability density. For stationary Gaussian processes, the one-dimensional probability density is determined by the variance of the random process. The correlation function of the process is sufficient to describe Gaussian processes. One of the characteristics of a random signal is its spectral power density associated with the correlated function of the generalized Fourier transform.

The spectrum of the random process is continuous. For random processes with constant spectral density and an infinite frequency band, the power is infinite and the correlation function is a delta function. This process has infinite variance, is uncorrelated and is called white noise. In a random process with a constant spectral density in a limited frequency band, the power is finite and can be determined.

In most cases, moment characteristics of the first two orders are used to characterize random processes: mathematical expectation, variance, and correlation

function (estimating the degree of static dependence of instantaneous process values $U(t)$ at any time t_1 and t_2).

Consider the **spectral representation** of stationary **random processes**. The power spectra of random functions are determined similarly to the power spectra of deterministic signals. The average power of a random process $X(t)$, registered in the process of one implementation on the interval $0 \div T$ using Parseval equality can be calculated by the formula: $W_T = \int_0^T \left[\frac{X^2(t)}{T} \right] dt = \int_{-\infty}^{\infty} \left[\frac{X_T(f)^2}{T} \right] df$, where $X_T(f)$ – spectral density of a single implementation $x(t)$.

As the interval increases, the process energy in the interval increases indefinitely, and the average power goes to a certain limit: $W = \int_{-\infty}^{\infty} \left[\lim_{T \to \infty} \frac{1}{T} \left| X_T(f) \right|^2 \right] df$, where the subintegral function is the spectral power density of a given implementation of a random process: $W(f) = \lim_{T \to \infty} \frac{1}{T} \left| X_T(f) \right|^2$.

Quite often this expression is called simply the power spectrum. Power density is a real, integral and even function of frequency. In the general case, the power density must be averaged over multiple implementations, but for ergodic processes it is permissible to average over one long-term implementation.

Spectra of discrete signals

Any complex periodic signal can be represented by a Fourier series as the sum of simple harmonic oscillations. The set of simple harmonic oscillations into which a complex periodic signal can be decomposed is called its spectrum.

The frequency distribution of harmonic amplitudes is called the amplitude-frequency spectrum or abbreviated amplitude spectrum, and the distribution of their initial phases by frequency is called the phase frequency spectrum or phase spectrum.

Discrete spectrum lines have the dimension of the signal amplitude.

If the signal spectrum is unlimited, then when determining the width, harmonics whose amplitudes do not exceed a certain (specified) level are neglected.

The most commonly used level is 0.707 in amplitude or 0.5 in power from the maximum value.

Fourier transform – is an integral transformation of one complex-significant function of a real variable into another, closely related to the Laplace transform and similar to the Fourier series decomposition for non-periodic functions. This transformation decomposes this function into oscillatory functions. Used to calculate the frequency spectrum for time-varying signals. Fourier transforms are used to obtain the frequency spectrum of a non-periodic function, such as an electrical signal, ie to represent a signal as the sum of harmonic oscillations. This uses the convolution property. An important conclusion from this transformation is that the output spectrum is obtained from the input by simple multiplication by the response function of the system $A(\omega)$.

It is known that any periodic function that satisfies the Dirichlet condition can be represented as an infinite in the general case sum of harmonic components – Fourier series. Dirichlet's condition is that: the function must be bounded, piecewise continuous, and have a finite number of extremums during the period.

There are two forms of Fourier series decomposition: trigonometric and complex. The trigonometric form of decomposition is expressed as $x(t) = \frac{1}{2}A_0 + \sum_{k=1}^{\infty} A_k \cos(K\omega_0 t - \varphi_k)$, where $\frac{1}{2}A_0$ – constant component of the function $x(t)$; $A_k \cos(K\omega_0 t - \varphi_k)$ – k-th harmonic component; A_k, $K\omega_0$, φ_k – amplitude, frequency and initial phase k-th harmonic component; $\omega_0 = \frac{2\pi}{T}$ – frequency of the main (first) harmonic; T – function change period of $x(t)$.

Mathematically, it is more convenient to operate with a complex form of the Fourier series, presented in the form $x(t) = \frac{1}{2}\sum_{k=-\infty}^{\infty} \dot{A}_k \exp\{jK\omega_0 t\}$, where $\dot{A}_k = A_k \exp\{-j\varphi_k\}$ – complex amplitude of the harmonic component of the

frequency $\omega_k = K\omega_0$. The complex amplitude is determined by the time function $x(t)$ using the formula $\dot{A}_k = \frac{2}{T}\int_{t_1}^{t_1+T} x(t)\exp\{-jK\omega_0 t\}\,dt$.

Any non-periodic signal can be considered as a periodic one, the period of change of which is equal to infinity. In this regard, the previously considered spectral analysis of periodic processes can be generalized to a non-periodic signal.

Consider how the spectrum of a non-periodic signal will change with an unlimited increase in the period of signal change. With increasing period T the intervals between adjacent frequencies in the signal spectrum and the amplitude of the spectral components decreases and within $T \to \infty$ become infinitesimal quantities. In this case, the spectral distribution of the non-periodic signal is reflected by a Fourier series.

The complex form of the non-periodic signal has the form $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(j\omega)^* \exp(j\omega t)d\omega$, де $S(j\omega) = S(\omega)\exp(j\phi(\omega))$ – spectral density of the signal; $S(\omega) = |S(j\omega)|$ – amplitude-frequency characteristic of the signal; $\phi(\omega)$ – phase-frequency characteristic of the signal. The previous expression is called the inverse Fourier transform formula.

Thus, the spectrum of a non-periodic signal, in contrast to the spectrum of a periodic signal, is continuous and represents the sum of an infinite number of harmonic components with infinitesimal amplitudes.

Spectra of single and periodic pulse sequences

The analysis of transients in a circle under the action of a complex (inharmonic) EDF by the classical method is quite complex. More convenient in such cases are methods based on the spectral representation of the external EDF and the principle of superposition. Therefore, before proceeding to the analysis of such transients, consider the spectra of some of the most important for radio engineering periodic and non-periodic EDF.

The sum spectrum theorem and the delay theorem make it possible to calculate the spectrum of a group of identical equally lagging pulses. Let there be two identical

impulses $f_1(t)$ and $f_2(t) = f_1(t-\tau)$, separated by a time interval τ. You can write the spectral function of the second pulse through the spectral function of the first: $\overline{S}_2(\omega) = \overline{S}_1(\omega)e^{-j\omega\tau}$.

Then, based on the expression for the spectral function, we get the sum of two pulses:

$$\overline{S}(\omega) = \overline{S}_2(\omega) + \overline{S}_1(\omega) = \overline{S}_1(\omega)\left(1 + e^{-j\omega\tau}\right) = \overline{S}_1(\omega)\left[1 + \cos\omega\tau - j\sin\omega\tau\right] =$$

$$= \overline{S}_1(\omega)\sqrt{\left(1+\cos\omega\tau\right)^2 + \sin^2\omega\tau}\, e^{-j\psi(\omega)} = \overline{S}_1(\omega)2\cos\frac{\omega\tau}{2}e^{-j\psi(\omega)},$$

where $\psi(\omega) = arctg\dfrac{\sin\omega\tau}{1+\cos\omega\tau}$.

As the number of pulses increases, the spectrum of the group of pulses approaches the structure of the linear spectrum of the periodic sequence of pulses.

Virtually all communication channels have limited bandwidth. Therefore, when a signal is transmitted over a real communication channel, only part of its frequency spectrum can be transmitted.

The practical width of the signal spectrum is taken as the frequency range within which the most important part of the signal spectrum is located. The choice of the practical width of the signal spectrum is determined by two criteria: the energy criterion and the criterion of permissible distortions of the signal shape.

Consider, for example, a sequence of rectangular pulses of duration τ, amplitude h, with the passage period T (Fig. 5.3).

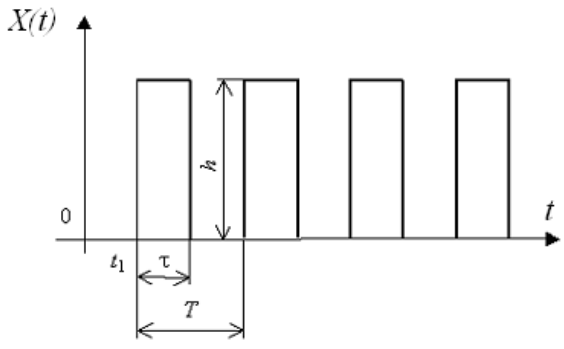


Fig. 5.3

The Fourier series of the periodic sequence of rectangular pulses is given in the form $x(t)=\frac{\tau h}{T}\left[1+2\sum_{k=1}^{\infty}\frac{\sin\frac{K\omega_0\tau}{2}}{\frac{K\omega_0\tau}{2}}\cos K\omega_0\tau\right]$.

The range of amplitudes of such a signal is shown in Fig. 5.4.

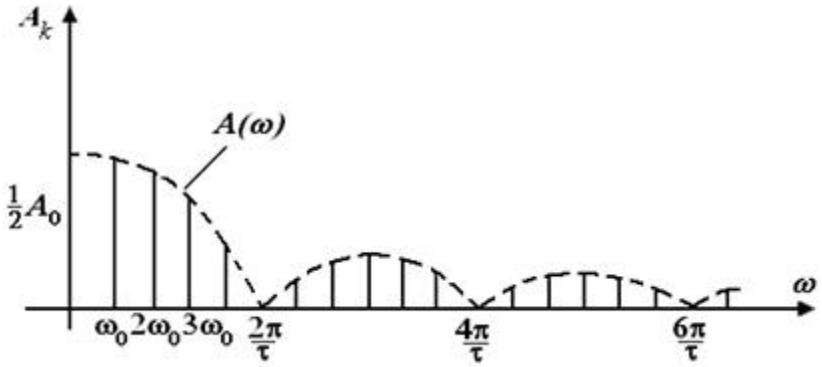


Fig. 5.4

Wrapping it is determined by the equation $A(\omega)=2\frac{\tau h}{T}\left[\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}\right]$, where $\omega=K\omega_0$ – for *k*-th harmonic.

For a periodic sequence of pulses of rectangular shape of duration $\tau=\frac{T}{2}$ it is enough to choose a practical spectrum width equal $3\omega_0=\frac{6\pi}{T}=\frac{3\pi}{\tau}$. 95% of all signal power is concentrated in this frequency range. Consider a single rectangular pulse of duration T and size h, the spectral density of such a signal is determined by the expression $S(j\omega)=\int_{-\infty}^{\infty}x(t)\exp\{-j\omega t\}dt=\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}}h\exp\{-j\omega t\}dt=\tau h\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}$.

Signal energy concentrated in the frequency band from 0 to ω_0, $W_1=\frac{1}{\pi}\int_0^{\omega_0}[S(\omega)]^2d\omega=\frac{\tau^2h^2}{\pi}\int_0^{\omega_0}\left[\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}\right]^2d\omega$.

Selection of sampling rate

Signal sampling means the transformation of functions of continuous variables into functions of discrete variables, by which the original continuous functions can be restored with a given accuracy. The role of discrete samples is performed, as a rule, by quantized values of functions in a discrete coordinate scale.

A discrete time system – is an algorithm with an input sequence $s(k)$ and the output sequence $y(k)$, which can be linear or nonlinear, invariant or time-varying. The discrete time system is linear and invariant in time if it corresponds to the principle of superposition (response to several inputs is equal to the sum of responses to each input separately), and the delay of the input signal causes the same delay of the output signal.

The essence of sampling analog signals is that the time-continuous analog function $s(t)$ are replaced by a sequence of short pulses, the amplitude values of which are determined by weight functions, or directly by sampling (readings) of the instantaneous values of the signal $s(t)$ at a time t_n. Signal presentation $s(t)$ at intervals T a set of discrete values c_n recorded as: $(c_1, c_2, \ldots, c_n) = A[s(t)]$, where A – sampling operator. Record signal recovery operation $s(t)$: $s'(t) = B[(c_1, c_2, \ldots, c_n)]$. Selection of operators A and B is determined by the required signal recovery accuracy. The simplest are linear operators. In the general case: $c_n = q_n(t)s(t)dt$, where $q_n(t)$ – system of weight functions.

The readings in the latter expression are related to the integration operation, which provides high sampling noise immunity. However, due to the complexity of the technical implementation of "weighted" integration, the latter is rarely used, with high levels of interference. Widespread methods are those in which the signal $s(t)$ is replaced by a set of its instantaneous values $s_n(t)$ at a time t_n. The role of weight functions in this case is performed by lattice functions. A period of time t between adjacent samples is called the sampling step. Sampling is called uniform with frequency $F = \frac{1}{t}$, if the value t constant over the entire signal conversion range. At

uneven sampling of value t between samples can vary according to a certain program or depending on the change of any signal parameters.

Optimal are sampling methods that provide a minimum numerical range for a given signal reproduction error. For nonorthogonal basis functions, static algebraic polynomials of the form are used: $s'(t) = c_n t_n$.

The requirement to choose the sampling rate is to introduce minimal distortions in the dynamics of changes in signal functions. It is logical to assume that the distortion of information will be less, the higher the sampling rate F. The greater the value F, the more digital data the signals will be displayed, and the more time will be spent processing them. In the optimal embodiment, the value of the sampling frequency of the signal F must be necessary and sufficient to process the information signal with a given accuracy, i.e. to allow the allowable error of recovery of the analog waveform (rms as a whole in the signal interval, or the maximum deviations from the true shape in the characteristic information points of the signals).

Kotelnikov's frequency criterion is based on **Kotelnikov's theorem**: if a continuous function $x(t)$ meets Dirichlet conditions (i.e., bounded, piecewise continuous and has a finite number of extrema) and its spectrum is limited by some frequency f_c – then it is completely determined by the sequence of its values at points that are spaced apart $T_k = \frac{1}{2f_c}$. Analytically, Kotelnikov's theorem is written by an interpolation series $x(t) = \sum_{k=-\infty}^{\infty} x(k\Delta t) \frac{\sin[\omega_c(t - k\Delta t)]}{\omega_c(t - k\Delta t)}$, where $\Delta t = \frac{\pi}{\omega_c} = \frac{1}{2f_c}$.

Obviously, a continuous function with a limited spectrum can be represented as the sum of an infinitely large number of terms, each of which is a factor of a function of the form $\frac{\sin y}{y}$ (reference function) and coefficient $x(k\Delta t)$, which determines the value of the function $x(t)$ at the time of reference.

View function $\frac{\sin \omega_c \tau}{\omega_c \tau}$ is the reaction of an ideal low pass filter with a cutoff frequency ω_c on the delta function. Then, if a time-quantized signal with a

quantization frequency is passed through such a filter $f = 2f_c = \frac{\omega_c}{\pi}$, then, finding the product of the output signals of the filter, you can get the output continuous signal of the function When using Kotelnikov's theorem for signal quantization, the real signal spectrum is conditionally limited by some frequency range from zero to ω_c, in which the main part of the energy of the signal spectrum is concentrated.

Lecture №6. Amplitude, frequency, phase modulation, comparison of signal spectra with different types of modulation. Pulse types of modulation. Code-pulse modulation, delta modulation, quadrature types of modulation

Analog modulation is used to transmit discrete data on channels with a narrow frequency band.

Analog modulation is a method of physical coding in which information is encoded by changing the amplitude, phase or frequency of the sinusoidal signal of the carrier frequency. The main methods of analog modulation are shown in Fig. 6.1. The diagram (Fig. 6.1, a) shows the sequence of bits of the source information, represented by high-level potentials for a logical unit and zero-level potential for logical zero. This method of encoding is called potential code, which is often used when transferring data between blocks of a computer.

At amplitude modulation (Fig. 6.1, b) for logical unit one level of amplitude of a sinusoid of carrier frequency is chosen, and for logical zero – another. This method is rarely used in its pure form in practice due to a number of noise immunity, but is often used in combination with another type of modulation - phase modulation.

With frequency modulation (Fig. 6.1, c) the values of 0 and 1 of the original data are transmitted by sinusoids with different frequencies – f_1 and f_2. This method of modulation does not require complex circuits in modems and is usually used in low-speed modems operating at speeds 300 or 1200 bit/s.

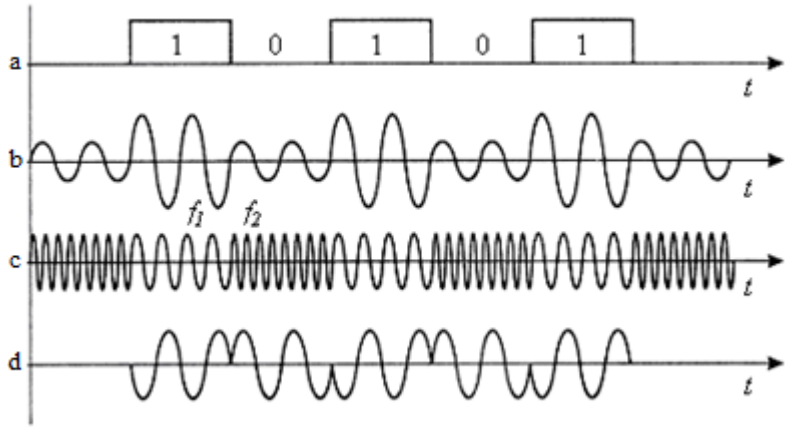


Fig. 6.1

In phase modulation (Fig. 6.1, d), the values of data 0 and 1 correspond to signals of the same frequency, but with different phases, such as 0 and 180 degrees or 0, 90, 180 and 270 degrees.

High-speed modems often use combined modulation methods, usually amplitude in combination with phase.

The spectrum of the resulting modulated signal depends on the type of modulation and the modulation rate, ie the desired bit rate of the source information.

Consider first the spectrum of the signal at potential coding. Let a logical unit be coded by a positive potential, and a logical zero by a negative potential of the same magnitude. To simplify the calculations, we assume that the information is transmitted, which consists of an infinite sequence of alternating ones and zeros, as shown in Fig. 6.2, a. Note that in this case the values of baud and bit per second coincide. For potential coding, the spectrum is directly derived from the Fourier formulas for the periodic function. If discrete data is transmitted at a bit rate N bits / s, the spectrum consists of a constant component of zero frequency and an infinite number of harmonics with frequencies f_0, $3f_0$, $5f_0$, $7f_0$, …, where $f_0 = \frac{N}{2}$. The amplitudes of these harmonics decrease rather slowly – with coefficients 1/3, 1/5,1/7, … from the amplitude of the harmonic f_0 (Fig. 6.2, a). As a result, the spectrum of potential code requires a wide bandwidth for quality transmission. In addition, keep in mind that the actual signal spectrum is constantly changing depending on what data is transmitted over the communication line. For example, the transmission of a long sequence of zeros or ones shifts the spectrum toward low frequencies, and in the extreme case, when the transmitted data consists of only ones (or only zeros), the spectrum consists of a zero frequency harmonic. When transmitting alternating units and zeros, the constant component is absent. Therefore, the spectrum of the resulting potential code signal when transmitting arbitrary data occupies a band from a value close to 0 Hz, to approximately $7f_0$ (harmonics with frequencies above $7f_0$ can be neglected due to their small contribution to the resulting signal).

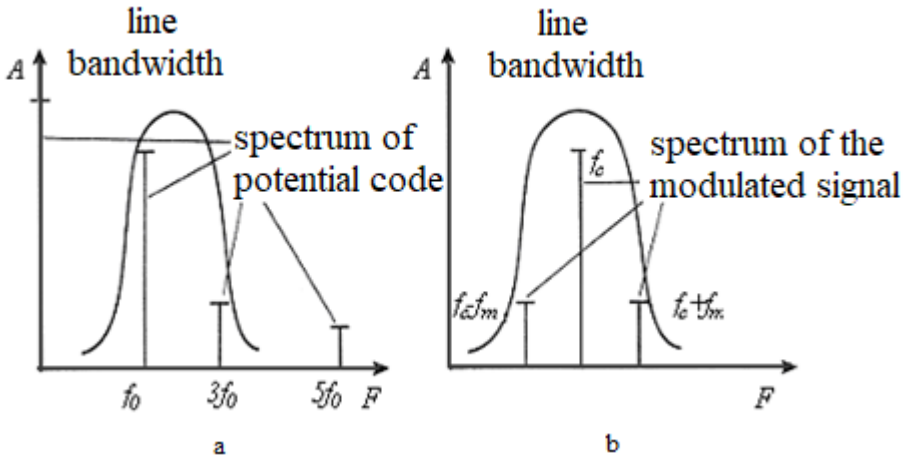


Fig. 6.2

For a tone channel, the upper limit at potential encoding is reached for a data rate of 971 bit/s, and the lower limit is unacceptable for any speed because the channel bandwidth starts at 300 Hz. As a result, potential codes on tone channels are never used.

In amplitude modulation, the spectrum consists of a carrier frequency sine wave f_c and two side harmonics: $(f_c + f_m)$ and $(f_c - f_m)$, where f_m – the frequency of change of the information parameter of the sinusoid, which coincides with the data rate when using two levels of amplitude (Fig. 6.2, b). Frequency f_m determines the bandwidth of the line in this encoding method. At low modulation frequency, the signal spectrum width will also be small (equal to $2f_m$), therefore, signals will not be distorted by a line if its bandwidth is greater than or equal to $2f_m$. For the tone channel, this method of modulation is acceptable at a data rate of not more than 3100/2 = 1550 bit/s. If 4 levels of amplitude are used to present data, the channel bandwidth is increased to 3100 bit/s.

In phase and frequency modulation, the signal spectrum is more complex than in amplitude modulation, because more than two side harmonics are formed here, but they are also symmetrically located relative to the main carrier frequency, and their amplitudes decrease rapidly. Therefore, these types of modulation are also well suited for data transmission over the tone channel.

Pulse types of modulation

In pulse modulation, a periodic sequence of pulses is used as a carrier, one of the parameters of which varies according to the law of the information signal.

The periodic sequence of pulses is characterized by the following parameters: duration τ; following period T_i or clock frequency $f = \frac{1}{T_i}$; amplitude U_0; the position (phase) of the pulses relative to the clock points t_i; gaps $Q_c = \frac{T_i}{\tau}$, which in pulsed systems may exceed 2000.

Depending on the selected modulating parameter, the following types of pulse modulation are distinguished: amplitude-pulse (APM), pulse width (PWM) and time pulse modulation, the varieties of which are phase-pulse (PPM) and pulse frequency (PFM) modulation. Complex types of pulse modulation include delta modulation (DM) and pulse code modulation (PCM).

Amplitude-pulse modulation (APM)

At APM the amplitude of pulses (Fig. 6.3, a) changes according to the law of the modulating (information) signal U_Ω (Fig. 6.3, b), and other parameters of the pulses remain unchanged.

The signal at ApM can be written in the form: $U_{APM} = [U_0 + \Delta U f(t)] \sum_i F(t - t_i - iT_i) = U_0 [1 + m_a f(t)] \sum_i F(t - t_i - iT_i)$, where $f(t)$ – function of modulating (information) signal in time; $F(t)$ – a function that describes the shape of a single pulse; $m_a = \frac{\Delta U}{U_0}$ – modulation depth factor.

There are two types of APM: continuous, the so-called APM of the 1st kind (AIM-1), in which the magnitude of the pulses changes according to the change of the modulating signal (Fig. 6.3, c), and flat - APM of the 2nd kind (APM-2).), in which the pulse amplitude is determined by the values of the modulating signal at the clock points (Fig. 6.3, d). Since the pulse duration is always much less than the period of the modulating signal, the difference between APM-1 and APM-2 is almost absent.

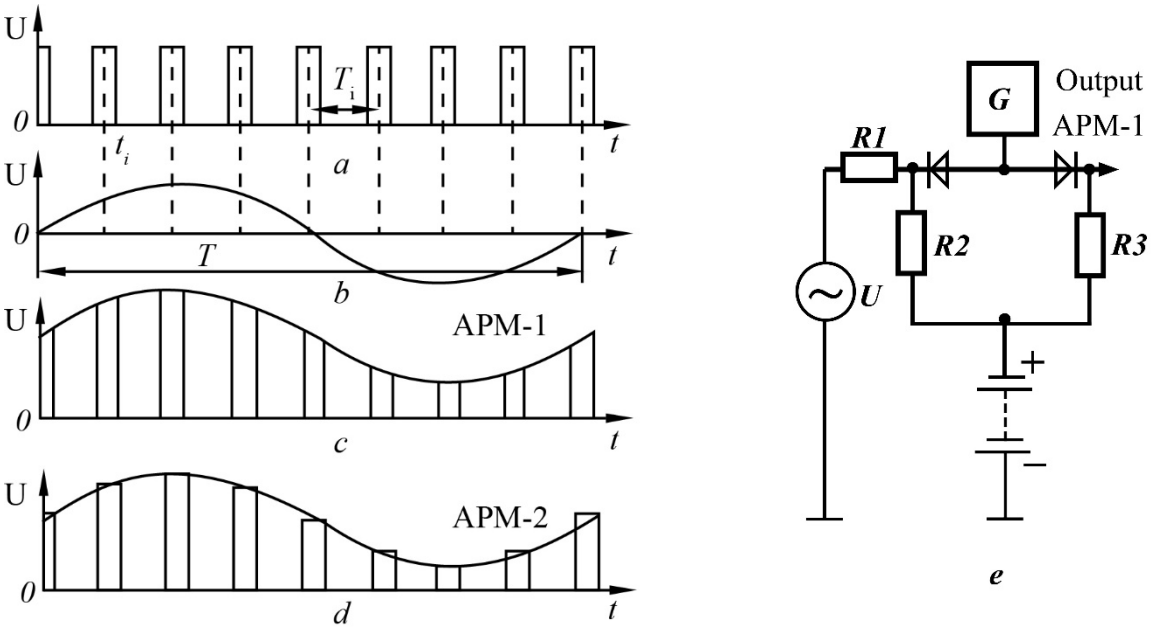


Fig. 6.3

a – carrier frequency pulses; b – information signal; c – signals APM-1; d – signals APM-2; e – modulator circuit at APM-1

In Fig. 6.3, d shows the simplest scheme of the modulator, which explains the process of APM-1. In the initial position, the diodes of the modulator are closed by the bias voltage and open only at times when the voltage of the pulses coming from the clock generator of the **CPG** is applied to them. At these moments, pulses proportional to the amplitude of the simulating voltage appear at the output of the modulator U_Ω.

The spectrum of signals in APM is generally determined by the spectrum of rectangular pulses of the carrier frequency and has an infinite number of components, the amplitude of which decreases with distance from the fundamental frequency. Due to the poor noise immunity and errors that occur when changing the transmission factor of the communication line, APM is used as an intermediate stage for other pulse modulations. At discrete character of the carrier the discrete APM (DAPM) is possible.

Pulse width modulation (PWM)

With PWM, the duration of the carrier frequency pulses (Fig. 6.4, a) varies according to the law of the information signal (Fig. 6.4, b) with constant other pulse parameters.

There are unilateral (UPWM) and bilateral (BPWM) pulse-width modulation. In UPWM, the change in pulse duration is performed by moving only one (front or rear) front (Fig. 6.4, c), and in BPWM – two fronts, symmetrically their centers, located in the clock points (Fig. 6.4, d).

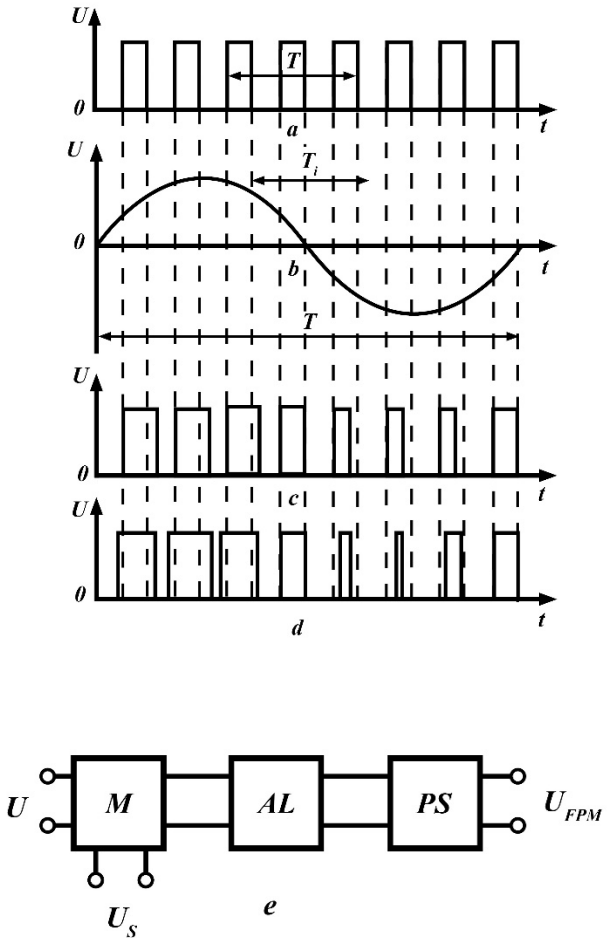


Fig. 6.4

a – carrier frequency pulses; b – information signal; c – UPWM - signals; d – BPWM – signals; e – signal generation scheme UPWM

The scheme of PWM signal generation is shown in Fig. 6.4, to UPWM signal can be obtained from the output of the modulator ***M***, if you bring it as a carrier sawtooth voltage U_S. When submitted to another entrance ***M*** information signal U_Ω,

at its output we obtain an amplitude-modulated sawtooth voltage. Limiting it with a two-way amplitude limiter ***AL***, we obtain pulses modulated by width (duration). To obtain a rectangular shape of the pulses with UPWM at the output ***OA*** include a pulse shaper ***PS***.

The spectrum of the UPWM signal can be determined from the expression: $U_{UPWM} = \frac{U_0}{T_i}\tau_n + \frac{U_0}{T_i}\Delta\tau \sin\Omega t + \frac{U_0}{n}\sum_{n=1}^{\infty}\frac{1}{n}\left[\sin n\Omega\left(t + 0{,}5\tau_n + \Delta\tau\sin\Omega t\right) - \sin\left(t - 0{,}5\tau_n\right)\right]$, where $f(t) = \sin\Omega t$; $\Delta\tau$ – the maximum deviation of the leading edge of the pulse.

The spectrum of the UPWM signal has a rather complex structure. The side frequency bands contain, in addition to components with frequency $n\omega \pm \Omega$, also combined frequencies $n\omega \pm m\Omega$, where $m = 1, 2, \ldots, n$. Bandpass filters that pass currents from one sideband are used to extract UPWM signals on the receiving side.

At discrete character of the carrier (carrier) discrete PWM – DUPWM and DBPWM is possible.

Noise immunity of PWM is much higher than noise immunity of APM, so PWM is widely used in telemetry.

Time pulse modulation

At time pulse modulation according to the law of modulating voltage the position in time of pulses concerning clock points changes. Depending on the patterns of change in the position of the pulses relative to the clock points, there are phase-pulse modulation (PPM) and frequency-pulse modulation (FPM).

Phase-pulse modulation (PPM)

Phase-pulse modulation (PPM) is characterized by the fact that the time shift of the pulses relative to the clock points, ie the deviation of the pulse repetition frequency (Fig. 6.5, a), is proportional to the amplitude of the modulating signal (Fig. 6.5, b) and does not depend on its frequency (Fig. 6.5, c). There are PPM of the first kind (PPM-1) and the second kind (PPM-2). In PPM-1, the magnitude of the time shift of the pulses relative to the clock points is proportional to the value of the modulating voltage at the time of reference of these pulses, and in PPM-2 – in the clock points. Since the pulse shift is much smaller than the period of the modulating signal, the difference between PPM-1 and PPM-2 is almost absent.

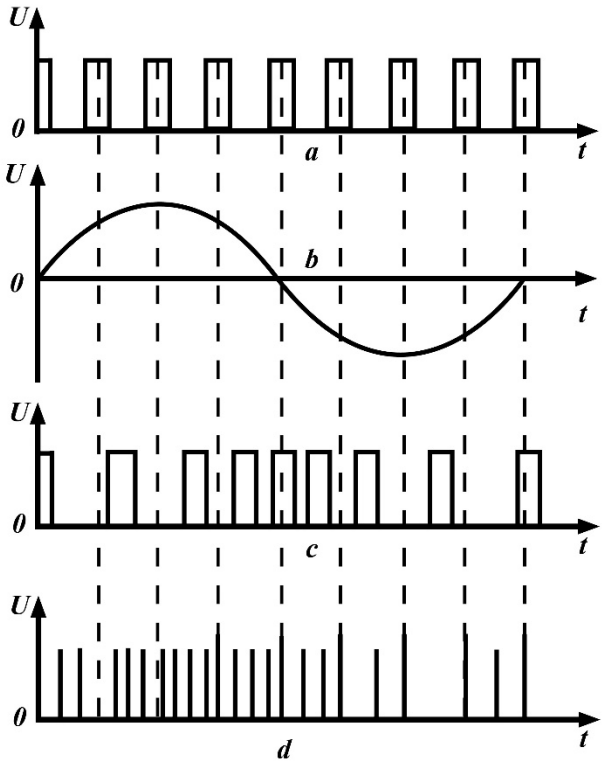


Fig. 6.5

a – pulse carrier; b – modulating signal; c – PPM signal; d – PWM signal

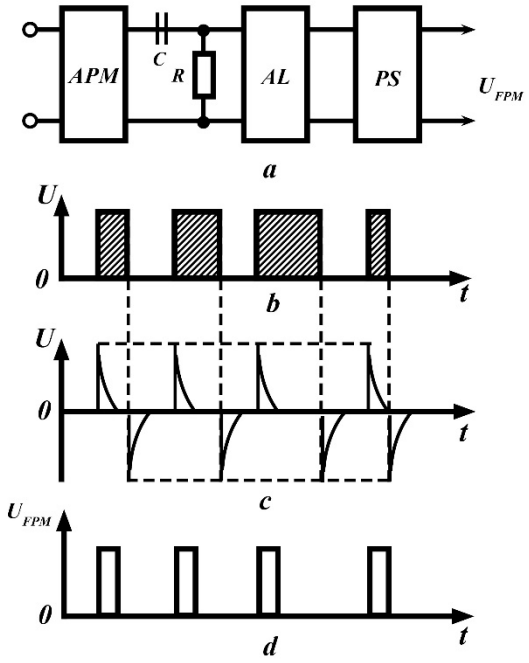


Fig. 6.6

a – the scheme of signal formation with PPM; b – PWM signal; c – signal at the output of the differentiating circuit; d – PPM signal

To obtain FIM signals, you can use the scheme shown in Fig. 6.6, a. In this scheme, after receiving the pulses modulated in width (duration) at the output of the pulse-width modulator PWM (Fig. 6.6, b), they are fed through the differential link ***RC***, at the output of which receive short sharp pulses (Fig. 6.6, c). After limiting their amplitude by the amplitude limiter ***AL*** receive pulses of the same polarity, which are fed to the forming stage ***PS*** to obtain pulses modulated by a phase of equal duration and amplitude.

The spectrum of PPM signals is similar to the spectrum in UPWM only with the difference that some components of the spectrum have a slightly smaller amplitude. In particular, the side frequencies of the spectrum in PPM have significantly smaller amplitudes than in UPWM, which complicates their selection during reception. Therefore, in systems with PPM, the obtained sequence of pulses with PPM is converted into a sequence of pulses with APM or PWM, from which the modulating signal is isolated by filtration.

Pulse frequency modulation (PWM)

PWM is characterized by the fact that the frequency of the pulses of the carrier varies according to the law of the modulating signal (Fig. 6.5, d), other parameters of the pulses remain unchanged. As the instantaneous value of the signal increases, the pulse frequency increases, and as it decreases, it decreases. In this way, pulse frequency modulation is performed, at which the pulse duration remains constant, only the interval between them changes. The bandwidth is determined by the pulse duration.

The main condition for PWM is that the duration of the information pulse (or duration of the pause) must be greater than or equal to ten periods of the carrier sequence.

There are many implementations of PWM, one of which is as follows:

- if 1 (there is a pulse) appears in the information sequence consisting of pulses, the frequency of the carrier sequence is reduced by half;
- If 0 (no pulse) appears in the information sequence, the carrier sequence frequency is doubled.Noise immunity than is approximately the same as in PPM.

Pulse code modulation (PCM)

At PCM a continuous signal $f(t)$ first quantized with time and level, and then each discrete signal level value is assigned a corresponding code combination. Thus, the original message is transmitted by code combinations (number of elements n is determined by the basis of the code and the number of quantization levels m), corresponding to the discrete that reflect the continuous signal $f(t)$ (fig. 6.7).

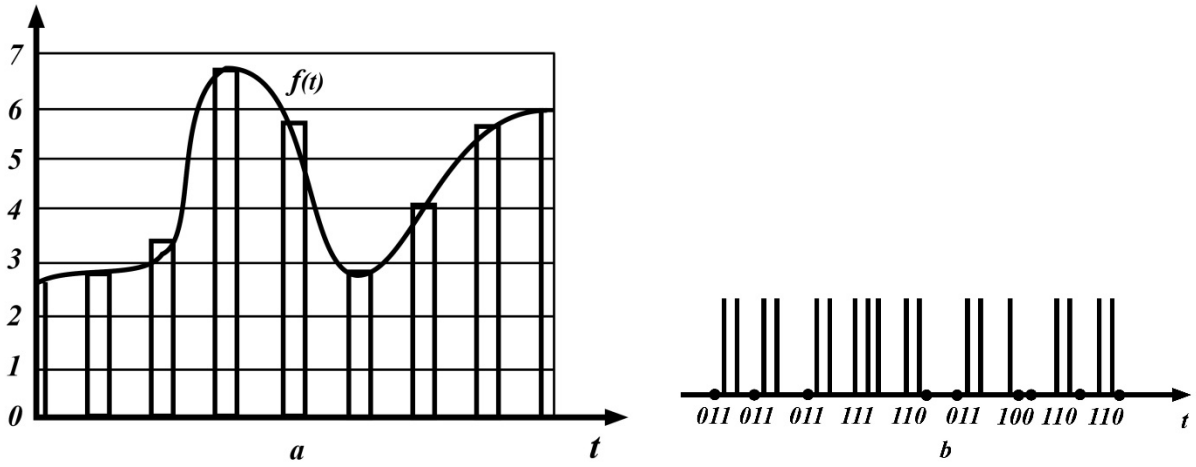


Fig. 6.7

а – output quantized signal; б – pulse sequence of code combinations

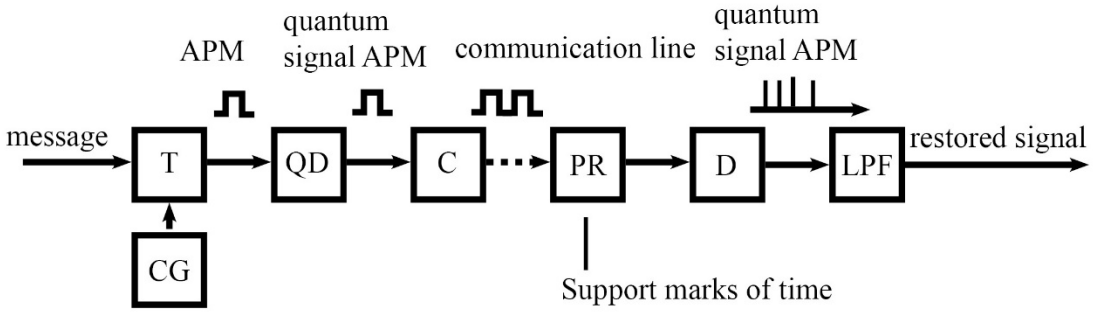


Fig. 6.8

T – transformer; CG – clock generator; QD – quantizing device; C – coder; PR – pulse regenerator; D – decoder; LPF – low pass filter

The functional diagram of the communication system is shown in Fig. 6.8. A continuous signal is applied to the system input $f(t)$, which comes to the breaker. At the same time it receives clock pulses from the clock generator with an interval Δt. At the exit a sequence of pulses is formed APM, the amplitude of which corresponds to the instantaneous values of the voltage of the continuous signal at the time of reading. From the exit the sequence of pulses from the APM enters the quantizing device, where they are quantized by level. The quantized sequence of pulses enters the coder, where each discrete pulse depending on its level is assigned a corresponding code combination containing n pulses. The sequence of pulses from the output is fed into the communication channel.

If necessary, channel signal modulation can be used to transmit code combinations through the channel.

Because the pulses are distorted during channel transmission, the pulse regenerator is usually turned on at the input of the receiving device. ***RD***, where the recovery of pulses in the form. When using modulation to transmit a communication channel before ***RD*** include a demodulator, the output of which after demodulation should be obtained a sequence of pulses with. After ***RD*** the restored sequence of pulses is fed to the decoder ***D***, where the code combinations are decoded, taking the form of a reproduced APM signal. To obtain a continuous signal from the output ***D*** APM the signal is fed to the low pass filter ***LPF*** with cutoff frequency F_m.

The advantage of the PCM system is the ability to suppress interference by regenerating pulses, which is very important when transmitting pulses over long distances. By breaking the communication line into small sections by regenerating pulses at intermediate stations, virtually undistorted message transmission can be achieved. In this case, the interference does not accumulate with increasing number of intermediate stations.

The disadvantages of PCM systems include a fairly wide required bandwidth. This is due to the fact that in the encoding process, a single pulse is converted into a code combination containing n pulses. These n pulses must occupy a time interval corresponding to one pulse they display. Therefore, the duration of each pulse during

transmission must be reduced to n times, which leads to an expansion of the corresponding number of times the frequency band.

The elements of the code combinations can be transmitted either on one channel in series or on n channels in parallel with time. In the latter case, the total frequency band required for transmission will be the same as for serial transmission. Therefore, PCM systems are broadband systems.

Delta modulation (DM)

At DM replacement of an output signal of the analog form is carried out $f(t)$ following step signal $G(t)$ (Fig. 6.9, a). Degree of conformity of functions $G(t)$ and $f(t)$ determined by the accuracy of the approximation, ie the values of the sampling steps (quantization in time) Δt and quantization (by level) ΔU. According to the function of the step signal $G(t)$ receive a modulated signal in the form of single pulses of positive and negative polarity (Fig. 6.9, b). These pulses reflect the sign of the difference between the current sample and its predicted value, which is taken as the quantized value of the previous sample. Thus, in DM, the value of the message at each point of the experiment is encoded by a single-bit binary code.

A simplified block diagram of the transmission system with DM is shown in Fig. 6.9, в. Continuous information signal $f(t)$ is fed to the input of the subtraction device ***SD***, on the second input of which simultaneously with the information from the output of the integrator ***I*** a quantized tracking signal is received $G(t)$, showing the previous time. ***SD*** performs a comparison of these signals and produces a difference signal $f(t) - G(t)$, which is fed to the input of the pulse modulator ***PM***. At another entrance ***PM*** from the output of the clock generator ***CG*** pulses are applied, the frequency of which is determined by the required degree of accuracy of reproduction of the transmitted signal at the reception, taking into account the action of interference in the channel.

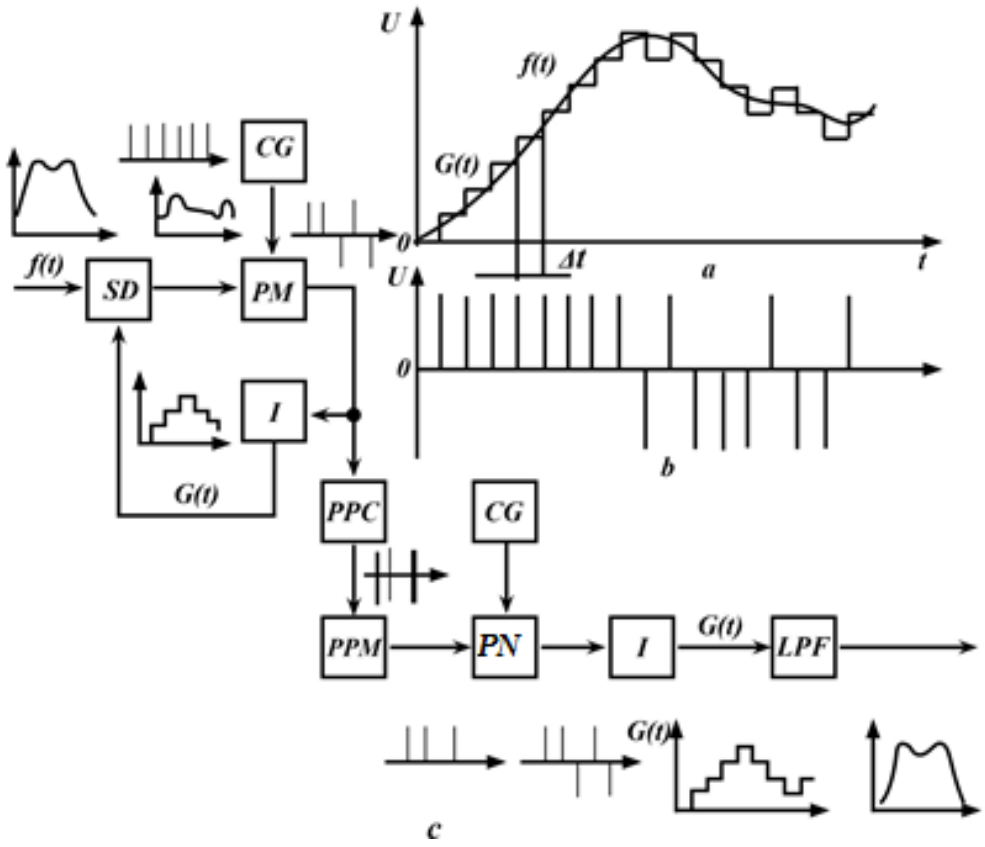


Fig. 6.9

a – replacement of the output signal $f(t)$ stepped signal $G(t)$; b – encoding the DM signal with a single binary code; c – block diagram of the transmission system with DM

At the exit ***PM*** there is a modulated signal, which is a pulse of different polarity that occurs at the time of receipt on ***PM*** clock pulses from ***CP*** (Fig. 6.9, b). The polarity of the pulses at the output ***PM*** is determined by the difference between the values of the functions $f(t)$ and $G(t)$. Yes, at $|f(t)-G(t)|>0$ a pulse of positive polarity is formed, and when $|f(t)-G(t)|<0$ - negative. If there are areas in the output signal that do not change in level, at the output ***PM*** the polarity of the pulses alternates. The modulated sequence of pulses is fed to the integrator ***I*** to generate a tracking signal. The action of the integrator is that when you enter its input from ***PM*** of the pulse of positive polarity, the voltage at its output at the clock points increases by one step, which corresponds to the quantization step by level ΔU, and upon receipt of a pulse of negative polarity - decreases by one step on the same

quantization ΔU. This tracking signal is fed to the input to generate a difference signal $f(t) - G(t)$.

From the exit ***PM*** modulated signals are fed to the transmitter, from the output of which they enter the channel. At the time of entry pulses of negative polarity at its output pauses are formed.

On the receiving side, the pulses coming from the channel are fed to the receiver, which serves to regenerate distorted interference pulses. At the exit receive pulses of positive polarity, which are fed to the pulse polarity converter ***PPC***. At another entrance ***PPC*** pulses come from ***CG*** with pulse repetition rate f_i ***CG*** transmitting device. To obtain an aligned signal $\varphi(t)$, close in shape to the original $f(t)$, the signal is passed through a low pass filter ***LPF***.

The degree of difference obtained at the output of the signal transmission system $\varphi(t)$ from the output information signal $f(t)$ will depend on the level of interference and distortion in the channel, as well as the quantization noise during transmission. The disadvantages of DM include the dependence of the dynamic range on the frequency of the modulating signal, which is explained by the increase in signal steepness with increasing frequency, as well as low noise immunity of the transmission system when exposed to impulse interference in the channel.

To improve the transmission quality in systems with DM, it is necessary to expand the bandwidth of the channel we use, as well as to apply various methods to combat interference and distortion of the transmitted pulses.

Quadrature modulation (type QAM)

Quadrature amplitude modulation algorithm (QAM, Quadrature Amplitude Modulation) is a kind of multiposition amplitude-phase modulation. This algorithm is widely used in the construction of modern modems.

When using this algorithm, the transmitted signal is encoded by simultaneous changes in the amplitude of the in-phase (***I***) and quadrature (***Q***) component of the carrier harmonic oscillation (f_c), which are shifted in phase relative to each other by $\frac{\pi}{2}$ radian. The resulting signal ***Z*** is formed by summing these oscillations. So, QAM-

the modulated discrete signal can be represented by a ratio: $Z_m(t) = I_m \cos(2\pi f_c t) + Q_m \sin(2\pi f_c t)$, where t – varies in range $\{(m-1)\Delta t \ldots m\Delta t\}$; m – serial number of the discrete countdown; Δt – the quantization step of the input signal over time; p – the quantization step of the input signal by amplitude; α_m and β_m – modulation coefficients; $I_m = \alpha_m p$, $Q_m = \beta_m p$.

The same signal can also be presented in a complex form: $Z = I + jQ$, or $Z_m = A_m \exp(2\pi f_c t + \varphi_m)$, where $A_m = \sqrt{I_m^2 + Q_m^2}$ – algorithm for changing the amplitude of the modulated signal; $\varphi_m = arctg\left(\frac{Q_m}{I_m}\right)$ – the algorithm for changing the phase of the modulated signal.

Thus, when using quadrature amplitude modulation, the information is encoded by simultaneous changes in the amplitude and phase of the carrier oscillation. Fig. 6.10 shows the principle of formation of the resulting oscillation ***Z*** by summing the vector of the quadrature component ***Q*** with the vector of the in-phase component ***I***. Vector amplitude ***Z*** determined by the ratio A_m, and the angle that this vector forms with the abscissa is determined by the ratio φ_m.

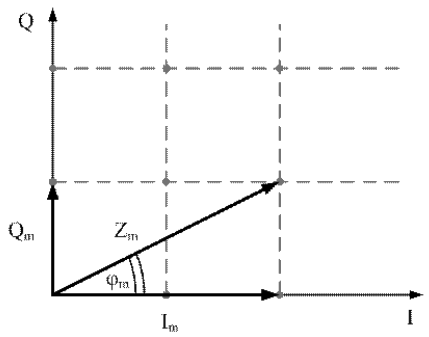


Fig. 6.10

It is important for this algorithm that when modulating the in-phase and quadrature components of the carrier oscillation, the same value of the amplitude change discrete is used. Therefore, the ends of the modulated oscillation vectors form a rectangular grid on the phase plane of the real – $\mathrm{Re}\{Z\}$ and imaginary components of the modulated signal vector – $\mathrm{Im}\{Z\}$. The number of nodes in this grid is

determined by the type of algorithm used QAM. The layout of the nodes in the phase plane of the modulated QAM oscillation is called a constellation.

To denote the type of QAM algorithm, the following notation scheme is adopted: QAM-<numeric>. «numeric» becomes type ***2N*** and corresponds to the number of nodes on the phase grid, as well as the maximum number of different values of the vector of the modulated signal. It should be noted that in this case the value ***N*** corresponds to the spectral efficiency of the algorithm used.

Fig. 6.11 shows a simplified structure of the QAM-modulated signal shaper.

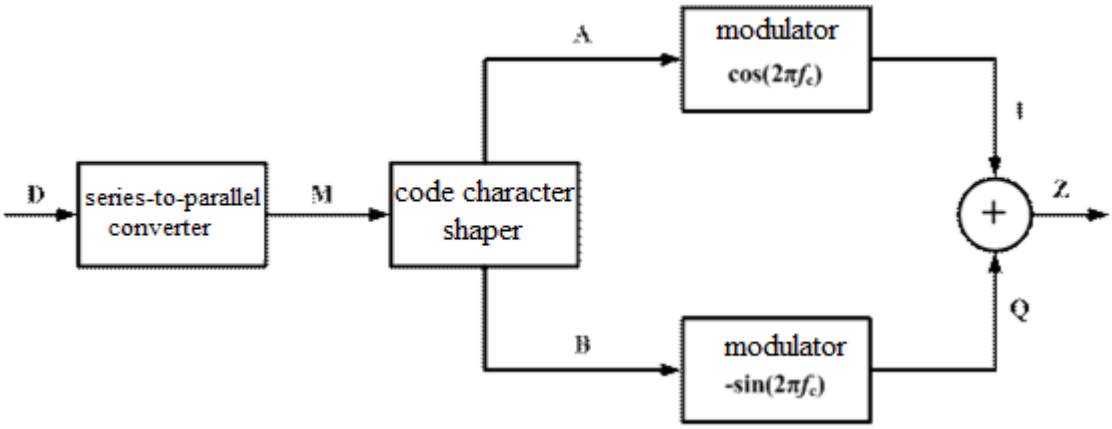


Fig. 6.11

In the first stage of conversion, a sequence of bits $D\{d_0,d_1,\ldots,d_k\}$, which comes from the signal source, will be converted into a sequence of two-dimensional modulation symbols $M\{m_0,m_1,\ldots,m_j\}$. The number of bits in this symbol is equal to the value ***N*** (for the algorithm QAM-16 $N=\log_2 16=4$).

The code character shaper converts a two-dimensional code character m_j in a pair of code characters α_j and β_j. For the algorithm QAM-16 valid values α_j and β_j belong to the plural {1, 3, –1, –3} and determine the values of the real and imaginary coordinates of the modulated oscillation vector, respectively. Generated values $A\{\alpha_j\}$ and $B\{\beta_j\}$ used for in-phase amplitude modulation ***I*** and quadrature ***Q*** components of the carrier oscillation. At the last stage of the transformation, the summation of these oscillations and the formation of the resulting signal ***Z***.

In figure 6.12 shows the location of modulated oscillation vectors - constellations for the QAM-16 algorithm. The figure shows the values of the modulation symbols, which correspond to the specified points on the phase plane of the modulated oscillation $\{m_3, m_2, m_1, m_0\}$. For the QAM-16 algorithm pair $\{m_3, m_2\}$ determines the quadrant number of the phase plane or the signs of the real and imaginary coordinates of the modulated oscillation vector:

$$
\begin{array}{ll}
00 & Sign(\mathrm{Re}\{Z\}) = 1, \quad Sign(\mathrm{Im}\{Z\}) = 1 \\
10 & Sign(\mathrm{Re}\{Z\}) = 1, \quad Sign(\mathrm{Im}\{Z\}) = -1 \\
01 & Sign(\mathrm{Re}\{Z\}) = -1, \quad Sign(\mathrm{Im}\{Z\}) = 1 \\
11 & Sign(\mathrm{Re}\{Z\}) = -1, \quad Sign(\mathrm{Im}\{Z\}) = -1
\end{array}
$$

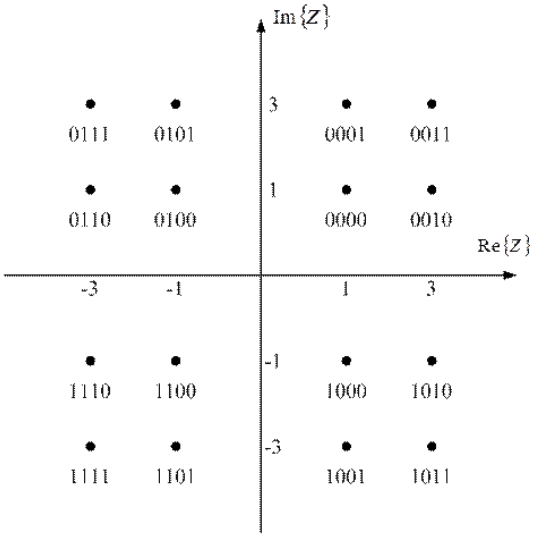


Fig. 6.12

For this algorithm pair $\{m_1, m_0\}$ determine the value of the amplitude of the real and imaginary coordinates of the vector of modulated oscillations, respectively. Table 6.1 shows the values of the code characters α_j and β_j, which correspond to the values of the lower digits of the modulation symbol $\{m_1, m_0\}$.

Table 6.1

m_1	m_0	α_j	β_j
0	0	1	1
0	1	1	3
1	0	3	1
1	1	3	3

The conversion of modulation symbols into code symbols is performed using Gray algorithms for noise-tolerant data encoding. Thus, the modulated oscillation vectors, which are close to each other on the phase plane, are matched to the values of code symbols that differ in the values of only one bit. Two vectors can be considered as an example $Z = 1 + j$ and $Z = 1 + 3j$, which correspond to the code characters {0, 0} and {0, 1}.

Currently, the most common are several variants of QAM: QAM-4 modulation algorithm, which encodes the information signal by changing the phase of the oscillation of the carrier with a step $\frac{\pi}{2}$. This modulation algorithm is called QPSK (Quadrature Phase Shift Keying). Algorithms have also become widespread QAM-16, 32, 64, 128 and 256. The quadrature amplitude modulation algorithm is essentially a variant of the harmonic amplitude modulation algorithm and therefore has the following important properties:

- the spectrum width of the QAM-modulated oscillation does not exceed the width of the modulation signal spectrum;
- the position of the spectrum of QAM-modulated oscillation in the frequency domain is determined by the carrier oscillation frequency.

The specific implementation of the QAM algorithm determines the values of such parameters:

- the dimension of the modulation symbol, defined as the number of points in the constellation ***N*** [bit];
- value of symbolic speed f_{symbol};
- central rate f_c.

The value of information speed ***V*** – data rate for the QAM algorithm – is determined by the following ratio: $V = Nf_{symbol}$.

The noise immunity of the QAM algorithm is inversely proportional to its spectral efficiency. The action of the problem causes uncontrolled changes in the amplitude and phase of the signal transmitted along the line. With a larger number of intercode points on the phase distance by them ***P*** decreases (Fig. 6.13) and,

consequently, the probability of erroneous recognition of the distorted received vector Z_m^* on the receiving side.

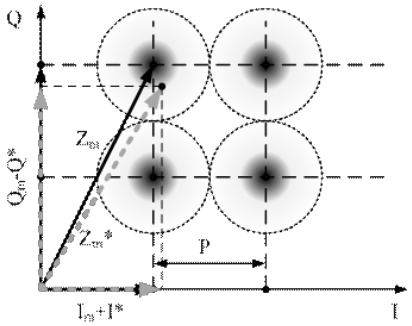


Fig. 6.13

The maximum level of allowable amplitude and phase distortion of the modulated QAM signal is a circle with a diameter ***P***. The center of this circle coincides with the node of the quadrature grid on the phase plane. The shaded areas in the figure correspond to the coordinates of the distorted vector of modulated QAM oscillation when exposed to a useful interference signal, the relative level of which is determined by the ratio 20db ≤ SNR30 ≤ db.

The disadvantages of the algorithm include the relatively low level of the useful signal in the spectrum of modulated oscillations. This disadvantage is common to harmonic amplitude modulation algorithms and is expressed in the fact that the maximum amplitude in the spectrum of modulated oscillation has a harmonic with the carrier oscillation frequency. Therefore, this algorithm in its pure form is rarely used in practice.

Lecture №7. Noise immunity of continuous and pulse types of modulation. Noise resistant coding. Correlation of corrective capacity with code distance. Heming's code, cyclic codes, BCH codes. Noise resistant encoding and decoding devices

Reliability – the degree of conformity of the received information to that transmitted. Reliability assessment – the ratio of the number of correctly received signals (k_{corr}) to the total number transferred (k_{trans}) signals for a certain period of time (with a sufficient number of transmitted messages): $R = \frac{k_{corr}}{k_{trans}}$.

The inconsistency of the received information transmitted is due to distortions, which are linear, nonlinear, random. All accidental distortions are caused by interference in the channel and communication equipment. Interference can either suppress a useful signal or create a false signal.

The effect of interference on the signal can be twofold:

1. If an obstacle $\zeta(t)$ consists of a useful signal *S(t)*, and the input of the receiver is their sum, then such a noise is called **additive**: $X(t) = S(t) + \zeta(t)$.
2. If the resulting signal is formed by the product of the noise and the signal, then such a noise is called a **multiplicative**: $X(t) = S(t) \cdot \zeta(t)$.

Multiplicative interference occurs when using radio communication. Most communication systems use wired communication, where only additive interference occurs. They are divided by type and source of interference. Types of interference – impulse, fluctuation, harmonic. Sources of interference – external and internal. The internal include thermal and background guidance. To external – industrial, atmospheric, space.

According to the form of additive interferences are divided into impulse, fluctuation and harmonic (Fig. 7.1).

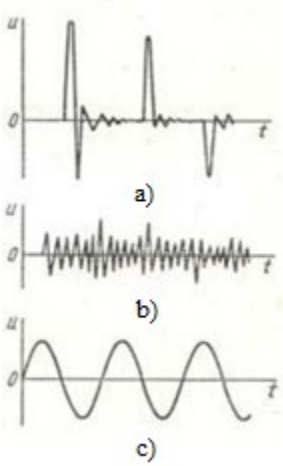


Fig. 7.1

a) – pulsed; b) – fluctuation; c) – harmonious

Impulse interference – consecutive pulses with random amplitude, duration and moment of occurrence of separate pulses.

Fluctuational interference or noise have the form of continuous randomly variable oscillations. Their most important characteristics: noise signal power, amplitude distribution law, type of energy spectrum or correlations function. The most common are fluctuation noise, the amplitudes of which are subject to the law of normal distribution. That is, for such interferences, the probability that the emission amplitude will exceed three times the effective voltage is very small. Fluctuation disturbances do not have a constant component. With a uniform frequency spectrum of interference in the communication line, the effective voltage of the fluctuating interference at the output of the receiver is proportional the square root of the bandwidth of the communication channel: $U^{ef}_{obstacle} = \sqrt{\Delta F}$, and the power is proportional to the band: $P_{obstacle} = \Delta F$.

For impulse interference, the power and amplitude of the interference is proportional to the bandwidth: $U_{obsctacle} = k_u \Delta F$ and $P_{obstacle} = k_p \Delta F$.

The amplitude and power of the interference is significantly affected by the bandwidth of the channel. It also affects the type of interference, ie it can happen that for the same interference in the communication line at the input of the narrowband receiver, the interference can be fluctuating, and for broadband - pulsed. The energy

spectrum of the interference characterizes its power distribution in the frequency range.

Noise resistance – it is the ability to perceive information correctly, despite the influence of interference. Consider the noise immunity of the elementary signal for the case of influence of fluctuation and impulse noise. The elementary signal can take two values: $A_1(t)$, $A_2(t)$. Signal transformation – it is an undetected change in a message that occurs during transmission under the influence of interference and causes the reception of a distorted signal. The probability of channel jamming is indicated by P_{jam} or P_{10} (converts 1 to 0). The probability of a distorted false obstacle P_{dist} or P_{01} (converts 0 to 1). There are two possible consequences when transmitting an elementary signal:

1. The probability of receiving a true signal (1→1 or 0→0) denote P_{11} and P_{00}.
2. The probability of receiving a false signal (1→0 or 0→1) denote P_{10} and P_{01}.

Based on the theory of the complete group of probabilities, we obtain $P_{11}+P_{10}=1$; $P_{00}+P_{01}=1$.

When creating means of communication, it is necessary to find compromise solutions, ie with sufficiently high transmission efficiency to achieve its reliability. To compare different ways of transmitting messages according to their noise immunity and efficiency, it is necessary to have a certain criterion. V. Kotelnikov developed a theory of potential noise immunity, which allows a quantitative assessment of different transmission methods under conditions of fluctuations. Based on this theory, Kotelnikov proved the possibility of the existence of an "ideal" receiver, which has the greatest potential noise immunity for this method of transmission. This receiver characterizes the noise immunity limit, which with this method of transmission can not be exceeded, it also allows you to assess the noise immunity of the real receiver and compare it with the "ideal" receiver.

The idea of the structure of this receiver (Fig. 7.2) is as follows: knowing what signals should be transmitted and having their samples, which are created by generators ***G1*** and ***G2***, compare the unknown signal with these samples and calculate the energy difference between the received signal and the sample, and then relate the received signal to the signal for which this difference is minimal. This comparison is

to determine the difference between the received signal *x(t)* with each exemplary $A_1(t)$ and $A_2(t)$ for a certain period of time *t*, where *t* – signal duration. Then find:

$$I_1 = \int_0^{\tau} [x(t) - A_1(t)]^2 \, dt \; ; \; I_2 = \int_0^{\tau} [x(t) - A_2(t)]^2 \, dt \, .$$

After that, it is stated that a signal was transmitted for which *Ii* is minimal, ie if I_2-I_1>0, then consider the received signal $A_1(t)$, and vice versa – if I_2-I_1<0 – then the signal $A_2(t)$.

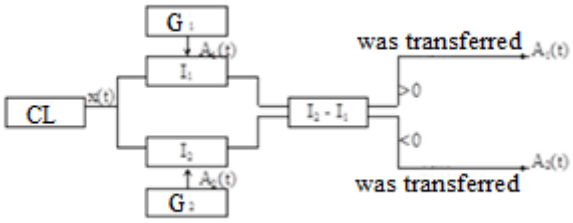


Fig. 7.2: ***CL*** – communication line, ***G*** – sample signal generators, ***I*** – integrators

Consider the **example** of the principle of operation of the ideal Kotelnikov receiver. Two elementary signals are transmitted over the communication line $A_1(t)=1$, for example – 5V and $A_2(t)=0$, in accordance – 0,5V.

1. The signal was transmitted first 1 (5V), which in the communication line was distorted, and the input of the integrators received a weakened signal - 3V, then, for $\tau = 1$ we have on integrators:

$I_1=[3-5]^2=4$ Wt;

$I_2=[3-0,5]^2=6,25$Wt.

After the adder I_2-I_1=6,25-4=2,25>0, and therefore we conclude that a signal was transmitted over the communication line $A_1(t)$=1.

2. In the second case, a signal was transmitted A2(t)=0 (0,5V), the average amplitude value of which has increased to due to interference in the communication line 1V. Then for $\tau = 1$ we have on integrators:

$I_1=[1-5]^2=16$ Wt;

$I_2=[1-0,5]^2=0,025$ Wt.

After the adder I_2-I_1=0,025-16=-15,975<0, and therefore we conclude that a signal was transmitted over the communication line $A_2(t)$=0.

To assess the noise immunity of the transmission, there is the concept of specific interference: $\sigma_0 = \frac{U_{meansquare}^{obstacle}}{\sqrt{\Delta F}}$, where $U_{meansquare}^{obstacle}$ – the root mean square value of the noise amplitude.

The value characterizing the potential noise immunity is equal to the ratio of the signal energy to the value of the specific interference: $a_0 = \frac{\sqrt{\int_0^{\tau} [A_1(t) - A_2(t)]^2 dt}}{\sigma_0}$.

Noise control methods are both methods of increasing the noise immunity of the signal and are divided into categories:

- reducing interference energy;
- based on increasing the noise immunity of transmitted signals.

The essence of the first methods is to determine the sources of interference, the place of their occurrence and reduce the power of their outflow due to:

- removal of sources of interference from communication channels (it is not allowed to lay power and information cables together);
- use of interference suppression schemes (use of filters, spark-extinguishing links, etc.).

The essence of the second methods is based either on increasing the signal energy, which increases the noise immunity, or in ensuring the noise immunity of the transmission by:

- noise-tolerant coding;
- information transfer with repetition;
- use of noise-tolerant modulation methods;
- use feedback.

The method of transmitting information with its repetition is used in the absence of feedback. The essence of this method is to send the same message several times, memorize the received messages, compare them element by element and compose a final message, which includes elements selected "by majority". For example, the same code combination was transmitted three times 1010101. In all three transmissions under the influence of interference, it was distorted as follows:

1000100

1111101

1010001

_____

1010101

Then, when using the iteration method, the receiver compares the three received characters bit by bit and puts those characters (code under the dash), the number of which in this digit is greater.

The method of transmission using feedback is that the transmission signals are compared on the forward and reverse channels, and in case of their coincidence, the message is received.

Noise-tolerant coding

Noise-tolerant codes are one of the most effective means of ensuring high fidelity both when storing and transmitting discrete information. A special theory of coding noise immunity has been created, which has been developing rapidly.

K. Shannon formulated a theorem for the case of transmission of discrete information from a channel with interference, which states that the probability of erroneous decoding of received signals can be provided as arbitrarily small by selecting the appropriate method of encoding signals.

Noise-tolerant codes are codes that allow you to detect or detect and correct errors that occur as a result of interference.

Noise immunity of coding is provided by introduction of redundancy in code combinations, ie due to the fact that not all symbols in code combinations are used for information transfer.

All noise-tolerant codes can be divided into two main classes: block and continuous (recurrent or chain).

In block codes, each message (or message element) corresponds to a code combination (block) of a certain number of signals. The blocks are encoded and decoded separately. Block codes can be uniform when the length of code combinations ***n*** is constant, or non-uniform when ***n*** is variable.

Uneven noise-tolerant codes have not received practical application due to the complexity of their technical implementation.

Both block and continuous codes, depending on the methods of redundancy, are divided into separate and inseparable. The role of individual characters is clearly delineated in separate codes. Some symbols are informative, others are verifiable and are used to detect and correct errors. Separate block codes are usually called n -codes, where n is the length of the code combinations, k is the number of information symbols in the combinations. Inseparable codes do not have a clear division of the code combination into informational and valid characters. This class of codes is still small. Separate block codes are divided, in turn, into non-systematic and systematic.

Most known separate codes are systematic codes. In these codes, the test symbols are determined as a result of linear operations on certain information symbols. For the case of binary codes, each test character is selected so that its sum modulo two with certain information symbols becomes zero. Decoding is reduced to checking for parity of certain groups of characters. As a result of such checks, information is given on the presence of errors, and if necessary - on the position of the characters where there are errors.

Consider the binary code that has found the most widespread use in practice. Recall that a binary code is a code based on $m=2$. The number of bits n in the code combination is called the length or significance of the code. The symbols of each digit can take the values 0 and 1. The number of units in the code combination is called the weight of the code combination and denote w.

The degree of difference between any two code combinations of this code is characterized by the so-called distance between codes d. It is expressed by the number of positions or symbols in which the combinations differ from each other, and is defined as the weight of the sum modulo two of these code combinations.

Errors, due to the influence of interference, are manifested in the fact that in one or more bits of the code combination, zeros turn into ones and, conversely, ones turn into zeros. The result is a new - erroneous code combination.

If errors occur in only one digit of the code combination, they are called one-time. In the presence of errors in two, three, etc. digits, the errors are called double, triple, etc.

Experimental studies of communication channels have shown that the errors of the symbols when transmitted over the communication channel, as a rule, are grouped into packets of different durations. A bundle of errors means a section of a sequence

that begins and ends with erroneously accepted characters. Inside the pack can be properly accepted elements.

An error vector e is used to indicate the places in the code combination where there is a character distortion. The error vector of an n-bit code is an ***n-*** bit combination, the units of which indicate the positions of the distorted characters of the code combination.

The weight of the error vector w_e characterizes the multiplicity of the error. The sum modulo two for the distorted code combination and the error vectors give the original undistorted combination.

Noise immunity of coding is provided by introduction of redundancy in code combinations. This means that of the n characters of the code combination, $k < n$ characters are used to transmit information . Therefore, of the total number $N_o=2^n$ possible code combinations, only $N=2^k$ combinations are used to transmit information . Accordingly, the whole set $N_o=2^n$ possible code combinations is divided into two groups. The first group includes the set $N= 2^k$ allowed combinations, the second group contains the set $(N_o-N)=2^n-2^k$ forbidden combinations.

If on the receiving side it is established that the received combination belongs to the group of allowed, it is considered that the signal came without distortion. Otherwise, it is concluded that the accepted combination is distorted. However, this is true only for such obstacles, when the possibility of transition from one permitted combination to another is eliminated.

In the General case, each of the N allowed combinations can be transformed into any of the N_o possible combinations, ie there are $N<N_o$ possible transmission options, of which:

- N options for error-free transmission;
- $N(N-1)$ options for transition to other permitted combinations;
- $N(N_o-N)$ options for transition to prohibited combinations.

Thus, not all distortions can be detected. The share of erroneous combinations that are detected is $\frac{N(N_0-N)}{NN_0}=1-\frac{N}{N_0}$.

To use this code as a correction set, the forbidden code combinations are divided into N non-intersecting subsets. Each of the subsets corresponds to one of the allowed combinations. The error is corrected in (N_o-N) cases equal to the number of

forbidden combinations. The share of erroneous combinations that are corrected from the total number of erroneous combinations that are detected is $\frac{N_0 - N}{N(N_0 - N)} = \frac{1}{N}$.

The method of division into subsets depends on what errors should be corrected by this code.

Suppose you need to build a code that detects all errors with a multiplicity of ***t*** and below.

To construct such a code means that from the set N_o is possible to choose ***N*** allowed combinations so that any of them in the sum modulo two with any vector of errors with weight $W_l \le t$ would not give as a result any other allowed combination. This requires that the smallest code distance satisfy the condition $d_{\min} \ge t + 1$.

In the general case, to eliminate multiplicity errors, the code distance must satisfy the condition $d_{\min} \ge 2\sigma + 1$.

To correct all multiplicity errors not more than and simultaneously detect all multiplicity errors not more than *t* and (at) the code distance must satisfy the condition $d_{\min} \ge t + \sigma + 1$.

Relationship of corrective power with code distance

Increasing the corrective capacity of the code is achieved by storing ***n*** by reducing the set ***N of*** allowed combinations (or reducing the number ***k of*** information symbols). Of course, in practice, the codes are built in reverse order: first, the number of information symbols ***k*** is selected based on the volume of the source alphabet, and then provides the necessary corrective capacity of the code by adding redundant characters.

Let the volume of the alphabet of the source ***N*** be known . The required number of information symbols $k = \log_2 N$. Let also know the total number of errors ***E*** that need to be corrected.

The task is to determine the significance of the code ***n*** , given the ***N*** and ***E*** , which has the necessary corrective capabilities.

The total number of erroneous combinations to be corrected is $E \ge 2k = E \ge N$. Because the number of erroneous combinations is $N_o - N$, the code provides correction of no more than $N_o - N$ combinations. Therefore, the necessary condition for the ability to correct errors can be written as: $NE \le N_0 - N$, and get $N_0 \ge (1 + E)N$, or $N \le \frac{2^n}{1 + E}$. This formula expresses the condition for selecting the value of the code ***n*** . Consider some cases. If there are errors of different

multiplicity, it is first necessary to ensure the elimination of single errors, the probability of which is greatest. The number of vectors of single errors is possible $E = C_n^1 = n$. In this case, the dependence becomes: $N = 2 \le \frac{2^n}{1+n}$.

When building the code, it is advisable to use the table 7.1. It should be borne in mind that the code must also satisfy the condition $d_{min} > 3$.

Table 7.1

n	2	3	4	5	6	7	8	9
2n/(1 + n)	1,88	2	3,2	5,33	9,2	16	28,4	51,2

If it is necessary to ensure the elimination of all errors of multiplicity from 1 to l , it is necessary to take into account that:

- the number of possible single errors $E_1 = C^l_n$;
- the number of possible double errors $E_2 = C^2_n$;
- the number of possible l - fold errors $E_l = C^l_n$.

Total number of errors $E = \sum_{i=1}^{l} C_n^i$. This dependency becomes: $N = \le \frac{2^n}{1 + \sum_{i=1}^{l} C_n^i}$.

This condition is a lower bound for the length correction code, i.e. it determines the required minimum code length n , which provides error correction at a certain given the multiplicity of the permitted combinations of N or number of information symbols $k = \log_2 N$.

The same condition is the upper bound for N or k , ie determines the maximum possible number of allowed combinations or information symbols for the code of length n , which provides correction of errors of a given multiplicity.

The main indicator of the quality of the correction code is its ability to ensure the correct acceptance of code combinations in the presence of distortions under the influence of interference, ie noise immunity of the code.

The corrective capability of the code is provided by redundancy, ie the lengthening of code combinations. At lengthening of code combinations the equipment becomes more complicated, time of transfer and processing of the information increases. Therefore, redundancy is also an important characteristic of the code. To assess the redundancy of the code use the concept of redundancy factor $K_{red} = \frac{\rho}{n} = \frac{n-k}{n}$, where ρ – the number of redundant characters in the code combination.

Example . Determine the corrective capacity of the code having the following allowed combinations: 00000; 01110; 10101; 11011.

The corrective capacity of the code is determined by the minimum code distance. Let's make a matrix of distances between code combinations (table 7.2).

Table 7.2

	00000	01110	10101	11011
00000	0	3	3	4
01110	3	0	4	3
10101	3	3	0	3
11011	4	3	4	0

As can be seen from the matrix, the minimum code distance $d_{min} = 3$. Therefore, this code is able to: detect double errors; eliminate one-time errors; eliminate and detect one-time errors.

Heming's code

Heming's code refers to systematic codes in which of the ***n*** symbols that form a combination, n_0 symbols are informational, and the last $k=n-n_0$ are redundant (control), intended for verification (control symbols in all combinations occupy the same positions) . Heming's codes allow to correct all single errors (at code distance ***d=3***) and to define all double errors (at ***d=4***), but not to correct them.

The relationship between the number of information and control symbols in the Heming code is found on the basis of such considerations. When transmitting a combination on a noise channel, any of the ***n*** characters of the code may be distorted , or the combination may be transmitted without distortion. Thus, there may be ***n+1*** variants of distortion (including transmission without distortion). Using control symbols, it is necessary to check all ***n+1*** variants. You can use the control symbols ***k*** to describe ***2k*** events. For this condition to be used, $2^k \geq n+1 = n_0 + k + 1.$

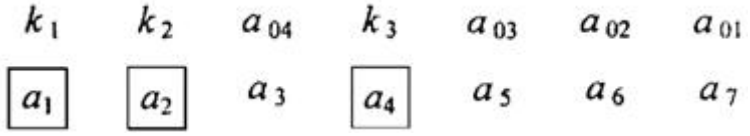
Table 7.3 shows the relationship between ***k*** and n_0, which is obtained from this inaccuracy, where ***k*** - number of control symbols in the Heming code, n_0 - number of information symbols.

Table 7.3. Placement of control symbols in Heming code combinations

n_0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
k	2	3	3	3	4	4	4	4	4	4	4	5	5	5	5

In the Heming code, the control symbols are placed in places multiples of the power of 2, ie in positions 1, 2, 4, 8, etc. Information symbols are placed in the

remaining places. For example, for a seven-element coded combination, you can write



The symbols of the Heming code, which are surrounded by rectangles, are *control* , the last - *information* , where a_3 - the highest (fourth) bit of the source code combination of binary code to be encoded, a_7 - the lowest (first) digit. After placing the code combination of control and information symbols in the appropriate places in the Heming code, special test equations are compiled, which are used to determine the presence of distortions and their corrections. 3 check equations i receive control symbols when encoding the source code combination of binary code.

To determine the control symbols it is necessary to use the following algorithm:

- All characters of the Heming code with bit numbers are arranged in ascending order of numbers and below them write the bit numbers in binary code

a_1	a_2	a_3	a_4	a_5	a_6	a_7
0001	0010	0011	0100	0101	0110	0111

- The first equation is at checkout amount for vcix mod 2 bits, the rooms are in LSB 2^0 is a unit: $S_1 = a_1 \oplus a_2 \oplus a_5 \oplus a_7$.

- The second equation is at checkout amount for the mod 2 vcix discharges in the rooms which is a unit of the second place corresponding binary equivalent (2^1): $S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7$.

- The third equation checkout is at 2 mod amount for all discharges room unit which is in third place (2^2): $S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7$.

Similarly, other checksums are formed (with a larger number of information and control symbols, respectively).

As can be seen from the above equations, each checksum includes only one indeterminate control symbol k_i (a_1, a_2, a_4, respectively), and all other information symbols are known.

Bci test equations under the Heming condition must be equal to 0 when summed by mod 2. From this condition i find the control symbols.

Example . It is necessary to transmit an information code combination:

a_3	a_5	a_6	a_7
1	1	0	0

with the number of digits n_0=4.

From the formula $2^k \geq n+1 = n_0 + k + 1$ we determine that ***k=3*** . Write checkout amounts: $S_1 = a_1 \oplus a_3 \oplus a_5 \oplus a_7$; $S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7$; $S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7$.

Substitute into the equation the values of known information symbols. From the condition of zero in all checksums, we determine the control symbols, respectively. From the first equation $a_1=0$, from the second $a_2=1$, from the third $a_4=1$.

Accordingly, the following combination will be transmitted:

a_1	a_2	a_3	a_4	a_5	a_6	a_7
0	1	1	1	1	0	0,

which is a combination of Heming's code.

Cyclic codes

Cyclic codes have been widely used due to their effectiveness in detecting and correcting errors. The schemes of encoding and decoding devices for these codes are extremely simple and are based on conventional shift registers.

The name of the codes came from their property, which is that each code combination can be obtained by cyclically rearranging the symbols of the combination belonging to the same code. This means that if, for example, the combination $a^0a^1a^2...a^{n-1}$ is a permissible combination of cyclic code, then the combination $a^{n-1}a^0a^1a^2...a^{n-2}$ also belongs to this code.

Cyclic codes convenient to consider, giving a combination of binary not a sequence of zeros and ones, as well as fictitious polynomial of variable x: $G(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0$, where a_i - the numbers of the number system (binary 0 and 1).

The greatest power of x with a nonzero coefficient a is called the power of the polynomial.

Representation of code combinations in the form of a polynomial allows to reduce actions on combinations to action on polynomials. The addition of binary polynomials is reduced to the addition modulo two coefficients with equal powers of the variable x; multiplication and division is carried out according to the usual rules

of multiplication and division of logical functions, the obtained coefficients for equal powers of the variable x are added modulo two.

According to the definition of the cyclic code for the construction of the generating matrix $P_{n,k}$ enough to choose only one original ***n-*** bit combination $V_i(x)$. Cyclic shift can be obtained (***n*-1**) of different combinations, of which any ***k*** combinations can be taken as a starting point. Summarizing the rows of the generating matrix in all possible combinations, you can get other code combinations. It can be shown that the code combinations obtained from some combination $V_i(x)$ by cyclic shift satisfy the conditions proposed for the set of initial combinations.

The cyclic shift of the combination with one in the highest ***n*** -th digit is identical to the multiplication of the corresponding polynomial by ***x*** with simultaneous subtraction from the result of the polynomial (x^n -1) or (x^n + 1), because the operations are modulo two. Therefore, if we take some polynomial ***P (x)*** as a source , then the process of obtaining basic polynomials can be represented as follows:

$U_1(x) = P(x)$;

$U_2(x) = P(x)x - C_2(x^n + 1)$;

$U_3(x) = P(x)x^2 - C_3(x^n + 1)$;

……

$U_n(x) = P(x)x^{n-1} - C_n(x^n + 1)$,

where C_2, C_3,...,C_n - coefficients that take the value 1 at $P(x)x^i>(x^n-1)$ and the value 0 at $P(x)x^i<(x^n-1)$.

In this way of constructing basic polynomials, the polynomial ***P(x) is*** called generating.

If we accept the condition that the polynomial $P(x)$ is a divisor of the binomial (x^n+1), then the basic combinations, and with them all allowed combinations of code, acquire the properties of divisibility by $P(x)$. It follows that the affiliation of the code combination to the group of allowed can be easily verified by dividing its polynomial into a polynomial-forming $P(x)$. If the remainder of the distribution is zero, then the combination is allowed.

This property of the cyclic code is used to detect or correct errors. Indeed, if under the influence of interference the allowed code combination is transformed into

the forbidden one, the error can be detected by the presence of a residue when dividing the combination into a polynomial-forming $P(x)$.

Thus, the polynomial $P(x)$ must satisfy the requirement - it must be a divisor of the binomial (x^n+1). The choice of $P(x)$ uniquely determines the cyclic code and its corrective properties.

The cyclic (n, A) -code can be obtained by multiplying a simple A-valued code, expressed as a polynomial of degree (k - 1), by some polynomial $P(x)$ *of* degree (n - k).

Another procedure for obtaining a cyclic code is possible. To do this, the code combination of a simple k -valued code $G(x)$ *is* increased by the monomial x^{n-k}, and then divided by the polynomial $P(x)$ *of* degree (n-k). As a result of multiplying $G(x)$ by $x^{n-k,}$ degree of each monomial included in $G(x)$ will increase by (n - k). When dividing the product $x^{n-k}G(x)$ by the polynomial $P(x)$ forming a particle $Q(x)$ of the same degree as $G(x)$.

The result of multiplication and division can be presented as follows:

$$\frac{x^{n-k}G(x)}{P(x)} = Q(x) + \frac{R(x)}{P(x)},$$

where $R(x)$ - remainder of dividing $x^{n-k}G(x)$ by $P(x)$.

Since the fraction $Q(x)$ has the same degree as the code combination $G(x)$, $Q(x)$ *is* also a combination of a simple k -valued code.

Thus, the code combination of cyclic (n, k) -code can be obtained in two ways:

– by multiplying a simple code combination of degree (k - 1) by the monomial x^{n-k} and adding to this product the remainder obtained from the division of the product by the polynomial $P(x)$ *of* degree (n - k),

– by multiplying a simple code combination of degree (k -1) by the polynomial $P(x)$ *of* degree (n - k).

According to the first method of encoding the first k symbols of the obtained code combination coincide with the corresponding symbols of the original simple code combination.

According to the second method, in the obtained code combination, the information symbols do not always coincide with the symbols of the original simple combination. This method is easy to implement, but due to the fact that the resulting code combinations do not contain explicit information symbols, complicates the decoding process.

In practice, the first method of obtaining a cyclic code is usually used.

When constructing a cyclic code, the number of information bits k is first determined by a given volume of code. Then there is the smallest length of code combinations n, which provides detection or correction of errors of a given multiplicity. This problem is reduced to finding the desired generating polynomial $P(x)$. The degree of the forming polynomial must be equal to the number of test digits .

Since in the cyclic code the error recognizers are the residues from the division of the polynomial of the received combination into the forming correction polynomial, the capacity of the code will be higher, the more residues can be formed as a result of this division.

It is known that a binomial of type $(x^n + 1) = (x^{2z-1} + 1)$, the expansion of which must include the forming polynomial as a factor, has the property that it is a common multiple for all without exception irreducible polynomials of degree n and decomposes by factors of all irreducible polynomials of degree 2, which are divisible without remainder by the number z .

The simplest cyclic code is code that provides one-time error detection. The vector of a single error corresponds to the monomial x^i, the degree of which i can take values from 1 to n . In order for an error to be detected, the monomial x^i must not be divisible by the polynomial $P(x)$. Among the irreducible polynomials included in the decomposition of the binomial $x^n + 1$, there is a polynomial of the smallest degree $x + 1$. Thus the forming polynomial of this code is the binomial $P(x) = x + 1$. The remainder of dividing any polynomial by $x + 1$ can take only two values: 0 and 1. Therefore, for any number of information bits, only one check digit is required. The value of the symbol of this digit ensures the parity of the number of units in the code combination.

This cyclic code with parity check provides detection of not only one-time errors, but also all errors of odd multiplicity.

To build a cyclic code that corrects single or detects double errors, it is necessary that each single error corresponds to its identifier, ie the remainder of the division of the polynomial of the received combination into a forming polynomial. Since the number of possible errors is unitary n and irreducible polynomial of degree ρ can give $2^\rho - 1$ nonzero residues is a necessary condition for correcting any single error is the inequality: $2^\rho - 1 \geq n$.

Hence find the degree of the forming polynomial: $\rho = n - k \geq \log_2(n+1)$ and the total length *n of the* code combination.

Since the forming polynomial $P(x)$ must be included as a factor in the expansion of the binomial $(x^n + 1) = (x^{2z-1} + 1)$, using the previously noted properties of this binomial, as well as condition (5.6), we can choose the forming polynomial.

However, not every polynomial of degree ρ included in the decomposition of a binomial $x^n + 1$ can be used as a polynomial. It is necessary that for each of the n single errors provided its own, different from the others, the remainder of the distribution of the received code combination on the forming polynomial. This will be the case if the chosen irreducible polynomial of degree ρ, being the divisor of the binomial $x^n + 1$, is not included in the expansion of any other binomial $x^i + 1$, the degree of which is < n .

Code-forming polynomials that are able to correct errors of any multiplicity can be determined using the following Hamming rule:

– The specified number of information bits k determines the number of check bits required ρ to correct single errors, and is the forming polynomial.

– Considering the obtained (n, k) -code and non-correcting n-bit code determine additional bits to ensure the correction of one error in this code and find the corresponding generating polynomial.

– This procedure is repeated until a code is received that corrects independent errors up to and including this multiplicity.

Example . Encode a simple information group $G(x)$ = 1011 with a cyclic code that detects double or single errors.

For a given number of information symbols $k = 4$ determine the significance of the code. Using the ratio $\rho = n - k \geq \log_2(n+1)$ and table 7.1, we obtain n=3,2.

To construct a cyclic code, it is necessary to choose a polynomial ***P(x) of*** degree $\rho = n - k = 3$, which must be included as a factor, and to decompose a binomial $x^n + 1 = x^{2z-1} - 1$. In our case, this binomial has the form $x^7 + 1$. Its constituent coefficients must be irreducible polynomials whose powers are divisors of the number z =3. Therefore, the coefficients of the binomial $x^7 + 1$ must be irreducible polynomials of the first and third powers. Using tables of irreducible polynomials obtain: $x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1)$.

Choose creating a polynomial multiplier: $P(x) = x^3 + x^2 + 1$.

The coding is carried out in the first way. For this initial code combination $G(x)$ multiplied by $x^{n-k} = x^3$: $x^{n-k}G(x) = x^3(x^3 + x + 1) = x^6 + x^4 + x^3$.

Determine the remainder $R(x)$ from the distribution $x^{n-k}G(x)$ on the forming polynomial $P(x)$:

$$\begin{array}{r|l} x^6+x^4+x^3 & x^3+x^2+1 \\ \underline{+\,x^6+x^5+x^3} & \overline{x^3+x^2} \\ x^5+x^4 & \\ \underline{+\,x^5+x^4+x^2} & \\ x^2 & \end{array}.$$

The residue $R(x) = x^2$. Thus, the polynomial $F(x)$ cyclic code will look like: $F(x) = x^{n-k}G(x) + R(x) = x^6 + x^4 + x^3 + x^2 = 1011100$.

Example . The received message with the cyclic code $P^*(x) = x^6 + x^4 + x^3 + x^2$. Check by decoding the presence of errors in the accepted combination, if the forming polynomial $P(x) = x^3 + x^2 + 1$.

Decoding is performed by dividing the polynomial of the resulting combination into a polynomial

$$\begin{array}{r|l} x^6+x^4+x^3+x^2 & x^3+x^2+1 \\ \underline{+\,x^6+x^5+x^3} & \overline{x^3+x^2} \\ x^5+x^4+x^2 & \\ \underline{+\,x^5+x^4+x^2} & \\ 0 & \end{array}.$$

The remainder of the distribution $R(x) = 0$. Therefore, the combination is accepted without distortion.

Example . The resulting combination $P^*(x) = x^6 + x^4 + x^2 + 1$, encoded by a cyclic code. The polynomial $P(x) = x3 + x2 + 1$. Check for code combination errors.

Divide the polynomial of the obtained combination by the polynomial forming

$$\begin{array}{r|l} x^6+x^4+x^2+1 & x^3+x^2+1 \\ \underline{+\,x^6+x^5+x^3} & \overline{x^3+x^2+1} \\ x^5+x^4+x^3+x^2+1 & \\ \underline{+\,x^5+x^4+x^2} & \\ x^3+1 & \\ \underline{+\,x^3+x^2+1} & \\ x^2 & \end{array}.$$

The remainder is $R(x) = x^2 > 0$. Therefore, the combination is accepted with errors.

Bowes-Chowdhury-Hawkingham codes

Bose-Chowdhury-Hawkingham (BCH) codes are a large class of codes that can correct several errors and occupy a prominent place in the theory and practice of coding. Interest in BCH codes is determined by at least the following four circumstances:

– among BCH codes at small lengths there are good (but, as a rule, not the best of known) codes;

– relatively simple and constructive methods of their coding and decoding are known (though if the only criterion is simplicity, other codes should be preferred);

– Reed-Solomon codes, which is a well-known subclass of non-binary codes, have certain optimal properties and a transparent weight structure;

– a full understanding of BCH codes appears to be the best starting point for studying many other classes of codes.

One of the classes of cyclic codes that can correct multiple errors are BCH codes.

The primitive BCH code that corrects t_u errors is the code of length $n=qm-1$ over $GF(q)$, for which the elements $\alpha^1,\alpha^2,...,\alpha^{2t_u}$ are the roots of the generating polynomial. Here is a primitive element $GF(qm)$. The generating polynomial is determined from the expression $g(x)=LCM\left\lfloor f_1(x),f_2(x),...,f_{2t_u}(x)\right\rfloor$ where $f_1(x), f_2(x)$ ... – minimal polynomials root of $g(x)$. The number of test elements of the BCH code satisfies the ratio $r=n-k\le m\cdot t_u$.

Example. Determine the value of the generating polynomial to construct a primitive code over $GF(2)$ of length 31, which corrects double errors (t_u = 2). Based on the definition of the BCH code, the roots of the polynomial $g(x)$ are: $\alpha^1,\alpha^2,\alpha^3,\alpha^4$, where the primitive element $GF(qm)=GF(25)$. The generating polynomial is determined from the expression $g(x)=LCM\left\lfloor f_1(x),f_2(x),f_3(x),f_4(x)\right\rfloor$, where $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$ are the minimum polynomials of the roots $\alpha^1,\alpha^2,\alpha^3,\alpha^4$, respectively.

Note. The minimum polynomial of the element β for $GF(qm)$ is determined from the expression, $f(x)=\left(x-\beta^{q^0}\right)\cdot\left(x-\beta^{q^1}\right)\cdot\left(x-\beta^{q^2}\right)\cdots\left(x-\beta^{q^{l-1}}\right)$, where l is the smallest integer at which $\beta^{q^l}=\beta$. The values of the minimum polynomials will be as follows:

$$f_1(x)=(x-\alpha^1)(x-\alpha^2)(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16})=x^5+x^2+1;$$

$$f_2(x)=(x-\alpha^2)(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16})(x-\alpha^1)=x^5+x^2+1;$$

$$f_3(x)=(x-\alpha^3)(x-\alpha^6)(x-\alpha^{12})(x-\alpha^{24})(x-\alpha^{17})=x^5+x^4+x^3+x^2+1;$$

$$f_4(x)=(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16})(x-\alpha^1)(x-\alpha^2)=x^5+x^2+1.$$

Due to $f_1(x) = f_2(x) = f_4(x)$, then:

$$g(x) = f_1(x) \cdot f_3(x) = (x^5 + x^2 + 1) \cdot (x^5 + x^4 + x^3 + x^2 + 1) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

.

In practice, when determining the values of the generating polynomial use a special table of minimal polynomials, and the expression for the generating polynomial $g(x) = f_1(x) \cdot f_3(x) \cdots f_j(x)$. The work is carried out in the following sequence.

For a given length of code *n* and the multiplicity of errors t_u correcting, determine:

- from the expression *n=2m-1* the value of the parameter *m*, which is the maximum power of the factors $g(x)$;
- from the expression $j=2t_u-1$ the maximum order of the minimum polynomial included in the number of factors $g(x)$;
- using the table of minimal polynomials, the expression for $g(x)$ is determined depending on *m* and *j*. To do this, from the column corresponding to the parameter m, polynomials with orders from 1 to *j* are selected, which as a result of multiplication give the value of $g(x)$.

Note. The expression for $g(x)$ kept minimal polynomials only odd degrees of α, as is usually the corresponding minimal polynomials of even degrees of α with similar expressions. For example, the minimum polynomials of the elements $\alpha^2, \alpha^4, \alpha^8$ correspond to the minimum polynomial of the element α^1, the minimum polynomials of the elements $\alpha^6, \alpha^{12}, \alpha^{24}$ correspond to the minimum polynomial, etc.

Example. Determine the value of the generating polynomial to construct a primitive code over *GF(2)* of length 31, which provides $t_u=3$. Determine the values of *m* and *j*.

$m = \log_2(n+1) = \log_2 32 = 5$;

$j = 2t_u - 1 = 2 \cdot 3 - 1 = 5$.

From the table of minimal polynomials, respectively, we obtain *m*=5 and *j*=5.

$$g(x) = 45 \cdot 75 \cdot 67 = (x^5 + x^2 + 1)(x^5 + x^4 + x^3 + x^2 + 1)(x^5 + x^4 + x^2 + x + 1) =$$
$$= x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1.$$

Given the initial data: n and t_u or k and t_u to build a cyclic code often lead to the choice of an inflated value of m and, as a consequence, to an unjustified increase in the length of the code. This provision reduces the efficiency of the received code, because part of the information bits are not used at all.

Example. Construct a generating polynomial for the BCH codes in the field *GF(16)*, constructed as an extension of the field *GF(2)*. Codes should correct errors of multiplicity 2-7.

In the table. 7.2 given representation of the field *GF(16)*, as an extension of the field *GF(2)*, built on a primitive polynomial $p(z) = z^4 + z + 1$. It also includes the minimum polynomials *GF(2)* for all elements from the field GF(16), where $\alpha = z$ – the primitive element *GF(16)*. It is noticeable that the minimum polynomials for any even degree always already exist in one of the previous rows of the table. Generating a polynomial for the BCH code of length 15, which corrects two errors:

$$g(x) = LCM\left\lfloor f_1(x), f_2(x), f_3(x), f_4(x)\right\rfloor =$$
$$= LCM\left\lfloor x^4 + x + 1, x^4 + x + 1, x^4 + x^3 + x^2 + x + 1, x^4 + x + 1\right\rfloor =$$
$$= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) = x^8 + x^7 + x^6 + x^4 + 1$$

Since the degree $g(x)$ is equal to 8, $n - k = 8$. Hence k = 7 and we obtained the generating polynomial (15,7) – code BCH, which corrects 2 errors. Note that the BCH codes are built on given n and t. The value of k is not known until $g(x)$ is found.

In the same way we can construct a generating polynomial for another primitive BCH code of length 15. Let $t = 3$:

$$g(x) = LCM\left\lfloor f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x)\right\rfloor =$$
$$= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) = \quad .$$
$$= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

A generating polynomial for the (15,5) - code BCG was obtained, which corrects three errors. Let $t = 4$:

$$g(x) = LCM\left\lfloor f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x), f_7(x), f_8(x)\right\rfloor =$$
$$= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1)(x^4 + x^3 + 1) = \quad .$$
$$= x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

Table 7.2. field representation GF(24)

in tne form of a degree	In the form of a polynominal	In binary	In decimal	Minimal polynomials
0	0	0000	0	
α^0	1	0001	1	$x+1$
α^1	z	0010	2	x^4+x+1
α^2	z^2	0100	4	x^4+x+1
α^3	z^3	1000	8	$x^4+x^3+x^2+x+1$
α^4	$z+1$	0011	3	x^4+x+1
α^5	z^2+z	0110	6	x^2+x+1
α^6	z^3+z^2	1400	12	$x^4+x^3+x^2+x+1$
α^7	z^3+z+1	1011	11	x^4+x^3+1
α^8	z^2+1	0101	5	x^4+x+1
α^9	z^3+z	1010	10	$x^4+x^3+x^2+x+1$
α^{10}	z^2+z+1	0111	7	x^2+x+1
α^{11}	z^3+z^2+z	1110	14	x^4+x^3+1
α^{12}	z^3+z^2+z+1	1111	15	$x^4+x^3+x^2+x+1$
α^{13}	z^3+z^2+1	1101	13	x^4+x^3+1
α^{14}	z^3+1	1001	9	x^4+x^3+1

A generating polynomial for the (15,1) -code BCH was obtained. This is a simple iterative code that fixes seven errors. Let $t = 5, 6, 7$. Each of these cases leads to the same generating polynomial as for $t = 4$. At $t > 7$ the BCG code is not defined, because the nonzero elements of the field *GF(16)* are only 15.

In the table. 7.3 shows a representation of the field GF(16) as an extension of the field GF(4), built on a primitive polynomial $p(z) = z^2 + z + 2$. This table also contains the minimum polynomials over GF(4) for all elements from the field *GF(16)*, where $\alpha = z$ – primitive element.

Example. We find generating polynomials for BCH codes that correct from 1 to 6 errors in the code combination. The code must be constructed in the field *GF(16)* obtained as an extension of the field *GF(4)*. Generating a polynomial for the BCH code over *GF(4)* of length 15, which corrects single errors:

$$g(x) = LCM\left\lfloor f_1(x), f_2(x)\right\rfloor = (x^2+x+2)(x^2+x+3) = x^4+x+1.$$

Table 7.3. field representation GF(42)

GF(4)			
+	0 1 2 3	*	0 1 2 3
0	0 1 2 3	0	0 0 0 0
1	1 0 3 2	1	0 1 2 3
2	2 3 0 1	2	0 2 3 1
3	3 2 1 0	3	0 3 1 2

This code encodes a sequence of 11 four-character characters (equivalent to 22 bits) to encode a sequence of 15 four-character characters. Such a code is not a Heming code.

In the same way we can find the generating polynomials for other codes over *GF(4)* of length 15. Let $t = 2$:

$$g(x) = LCM\lfloor f_1(x), f_2(x), f_3(x), f_4(x) \rfloor = (x^2 + x + 2)(x^2 + x + 3)(x^2 + 3x + 1) = .$$
$$= x^6 + 3x^5 + x^4 + x^3 + 2x^2 + 2x + 1$$

A generating polynomial for the (15, 9) - code BCH over GF(4) was obtained, which corrects two errors.

Let $t = 3$: $g(x) = x^9 + 3x^8 + 3x^7 + 2x^6 + x^5 + 2x^4 + x + 2$. This gives a (15, 6) - BCH code over GF(4), which corrects three errors.

Let $t = 4$: $g(x) = x^{11} + x^{10} + 2x^8 + 3x^7 + 3x^6 + x^5 + 3x^4 + x^3 + x + 3$. This gives a (15, 4) -BCH code over GF(4), which corrects four errors.

Let $t = 5$: $g(x) = x^{12} + 2x^{11} + 3x^{10} + 2x^9 + 2x^8 + x^7 + 3x^6 + 3x^5 + 3x^4 + 3x^3 + x^2 + 2$. This gives a (15, 3) - BCH code over GF(4), which corrects five errors.

Let $t = 6$: $g(x) = x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$. It turns out (15, 1) - BCH code over GF(4), which corrects six errors. This is simple code with repetition, which actually fixes seven errors.

Since the BCH codes are cyclic codes, the general rules of encoding information with cyclic codes are used in coding.

Lecture №8. Digital presentation of information. Loss of information during encoding. Positional number systems. Direct, complementary, reverse codes

Signal sampling means the transformation of functions of continuous variables into functions of discrete variables, by which the original continuous functions can be restored with a given accuracy. The role of discrete samples is performed, as a rule, by quantized values of functions in a discrete coordinate scale.

A discrete time system is an algorithm with an input sequence $s(k)$ and an output sequence $y(k)$, which can be linear or nonlinear, invariant or time-varying. A discrete time system is linear and invariant in time if it conforms to the principle of superposition (response to several inputs is equal to the sum of responses to each input separately), and the delay of the input signal causes the same delay of the output signal.

The essence of sampling analog signals is that the continuity in time of the analog function $s(t)$ is replaced by a sequence of short pulses, the amplitude values of which are determined by weight functions, or directly by sampling (counting) instantaneous signal values $s(t)$ at time t_n. The representation of the signal $s(t)$ on the interval T by a set of discrete values c_n is written in the form:, $(c_1, c_2, \ldots, c_n) = A\left[s(t)\right]$ where A – the sampling operator.

Recording signal recovery operation $s(t)$: $s'(t) = B\left[\left(c_1, c_2, \ldots, c_n\right)\right]$.

The choice of operators A and B is determined by the required accuracy of signal recovery. The simplest are linear operators. In the general case: $c_n = q_n(t)s(t)dt$, where $q_n(t)$ – system of weight functions.

The samples in the latter expression are associated with the integration operation, which provides high noise immunity of sampling. However, due to the complexity of the technical implementation of "weighted" integration, the latter is rarely used, with high levels of interference. More widespread are methods in which the signal $s(t)$ is replaced by a set of its instantaneous values $s(t_n)$ at time intervls t_n. The role of weight functions in this case is performed by lattice functions. The period of time t between adjacent samples is called the sampling step. Sampling is said to be uniform with frequency $F = \frac{1}{T}$ if the value t is constant over the entire range of signal conversion. In case of uneven sampling, the value t between samples can change according to a certain program or depending on the change of any signal parameters.

Reproduction of a continuous signal by samples can be performed on the basis of both orthogonal and non-orthogonal basis functions. The reproducing function is $s'(t)$ respectively represented by the approximating polynomial: $s'(t) = c_n y_n(t)$, $y_n(t)$ where is the system of basis functions.

Optimal are sampling methods that provide a minimum numerical range for a given signal reproduction error. If nonorthogonal basis functions are used mainly static algebraic polynomial form: $s'(t) = c_n t_n$.

If the values of the approximating polynomial coincide with the values of the samples at the time of their reference, then such a polynomial is called interpolation. Lagrange polynomials are commonly used as interpolation polynomials. To implement interpolation polynomials, it is necessary to delay the signal by the sampling interval, which in real-time systems requires certain technical solutions. Taylor polynomials are usually used as extrapolation polynomials.

During *quantization* realize continuous transformation for the value of the magnitude of the discrete scale of values from a finite set of allowed, called quantization levels. If the quantization levels are numbered, the result of the transformation is a number that can be expressed in any number system. Rounding with a certain bit of instantaneous values of a continuous analog quantity with a uniform step by argument is the simplest case of sampling and quantization of signals when they are converted into digital signals.

Sampling of analog signals with digital conversion is associated with signal quantization. The essence of quantization is to replace the innumerable possible values of the function, in general random, a finite set of digital samples, and is performed by rounding the instantaneous values of the input function $s(t_i)$ at times t_i to the nearest values $s_i(t_i) = n_i$, where i – the quantization step of the digital scale. Quantization with a constant step is called uniform.

When quantizing signals in a large dynamic range of values, the quantization step can be non-uniform, for example, logarithmic, ie proportional to the logarithm of the values of the input signal. The set range of the quantization scale from $s_{\min}$ to $s_{\max}$ and the quantization step determine the number of scale distributions $N = \frac{s_{\max} - s_{\min}}{i}$ and, accordingly, the digital bit quantization. As a result of sampling and quantization, the continuous function $s(t)$ is replaced by a numerical sequence $\{s(kt)\}$. The rounding error $\eta = s(t_i) - s_i(kt)$ is called quantization

noise. The required quantization accuracy is estimated by the effect of the resulting quantization noise on the subsequent signal processing.

With a sufficiently small quantization step, any value within it can be considered equally probable, while the values are distributed according to a uniform law.

The input signal usually contains an additive mixture of useful signal $s(t)$ and interference $q(t)$.

In practice, the quantization step is usually chosen so that there is no noticeable change in the signal-to-noise ratio.

The impetus for the presentation of continuous (analog) signals in digital form was the need to classify speech signals during World War II. An even greater stimulus to the digital conversion of analog signals was the creation of a computer that could be used as the end device of a digital information transmission system, entrusting it with logical operations for receiving and processing signals.

Nowadays, any type of information can be transmitted in digital form, providing the necessary reliability at a significant transmission speed. A strong impetus in the development of digital information transmission systems was the creation of information and computer networks and the availability of technological base for the production of high-speed switches, broadband modems, multiplexers, amplifiers and others.

Encoding is the representation of certain rules of discrete messages in some combinations, consisting of a certain number of character elements. These elements are called code elements, and the number of different elements that make up the combinations - the basis of the code. Code elements form code combinations. For example, if we make combinations of different combinations (0 and 1), it is a code with a base two or a binary code. If all combinations have the same number of characters, the code is called uniform. The well-known Morse code is a non-uniform code. The coding rule is usually expressed by a code table, in which each character of the message is associated with a certain code combination.

Therefore, one of the main advantages of transmitting information in digital form is the ability to use encoded signals and the optimal in given conditions, the method of their reception. It is important that in digital transmission, all types of signals, such as speech, music, television, data, can be combined into one common stream of information, the transmission of which is formalized. In addition, the seal when using a computer allows you to more efficiently use the spectrum and time,

protect the channel from unauthorized access, combine into a single process the transmission of digital information and digital switching of channels and messages.

Of course, some advantages of digital types of transmission need to be paid for, in particular, a wider spectrum of the emitted signal, the presence of intercharacter interference, the need to synchronize transmission systems and more. The pace of implementation of digital technology suggests that all these difficulties can be overcome.

Recovery of a continuous signal by sampling can be performed on the basis of both orthogonal and non-orthogonal basis functions. The reduction function is $s'(t)$ respectively represented by the approximating polynomial: $s'(t) = \sum_n c_n y_n(t)$, where $y_n(t)$ is the system of basis functions.

Orthogonal basis functions ensure the convergence of the series when $n \to \infty$. Optimal are sampling methods that provide a minimum numerical range for a given signal reproduction error. If nonorthogonal basis functions are mainly used algebraic polynomial of degree following: $s'(t) = \sum_{n=0}^{N} c_n t^n$.

Positional number systems

Under *the number system is* understood a set of techniques for the name and designation (record) of numbers. Symbols used to denote numbers are called numbers. In addition to numbers, delimiters are also used (+, -,,,. And others). Like ordinary language, the language of numbers has its own alphabet. The decimal number system has become a common language of numbers. Any number in it is represented by a set of ten digits: 0,1,2,3,4,5,6,7,8,9. The value of each digit in the number record depends on the position it occupies in the number. For example, in the entry 555.5, the number 5 occurs four times, but in each position it has a different meaning: the leftmost digit 5 means the number of hundreds and has a value of 500, the next digit 5 means the number of tens, 5, which is preceded by a comma, means the number of units and, finally, the number 5 after the comma - the number of tenths of a unit. The decimal number system is positional.

Positional number system is a system in which when writing numbers, the same digit has different meanings depending on the position it occupies in the number. Any positional number system uses a certain number of different digits (symbols) to denote numbers. Therefore, they differ in their basis and basis.

The basis of the number system - a set of different digits used to write numbers in this system. The number of digits in the basis is called the *basis* of the number system.

Positional number systems are called according to their base: decimal - base 10, octal - base 8, binary - base 2. There may be number systems that use more than 10 digits. For example, the hexadecimal number system uses sixteen digits.

Widespread decimal number system associated with the physiological structure of human hands (feet) (10 fingers (toes)). If we had a dozen (12) fingers on our hands, we would most likely use a twelve-year-old system.

Since the positional number system can be based on any positive integer greater than one, many such systems can be created.

To assess the need for a number system as a basis for designing a computer is important, in addition to the simplicity of arithmetic operations in it, also what is usually called the *economy of the system* . This means a set of numbers that can be written in this system with a certain number of characters. To write 1000 numbers (from 1 to 999) in the decimal number system, you need 30 characters (10 digits for each digit). And in the binary system with the help of 30 characters you can write 2^{15} different numbers (because for each binary digit you need only characters 0 and 1, then with the help of 30 characters you can write numbers containing up to 15 binary digits). But $2^{15} > 1000$, so having 15 different digits, you can write more different numbers than with three decimal places. In the general case, if we take n characters and take a number x as the basis of the number system, then the number of digits will be defined as ($\boldsymbol{n / x}$) digits, and the number of numbers that can be written will be equal to $\boldsymbol{x^{n/x}}$. Consider this expression as a function of the variable x. The maximum of this function is achieved at x = 2,718281828459045. This number is the basis of natural logarithms, but it is not an integer. The closest integers to this number will be 3 and 2. Due to the simplicity of the technical implementation of the simulation of digits of the binary system (0, 1) in the computer most often used binary number system, also known implementations of computers with ternary number system.

To determine in which number system the number is written, its base is written at the bottom after the number. For example, 708_{10}, 36_8, 101_2.

In any positional number system, its base is written as 10, is one in the senior digit and 0 in the junior.

All positional number systems are built on one general principle : a positive integer ***P>1*** is chosen - the basis of the number system; the record of any

number M is given in the form of a polynomial, ie a combination of degrees of the basis of the number system P with coefficients a_0, a_1, a_2,…, a_k, taking values from 0 to P-1, ie from the basis of the number system: $M_{(p)} = a_k p^k + a_{k-1} p^{k-1} + \ldots + a_0 p^0 + a_{-1} p^{-1} + \ldots + a_{-m} p^{-m}$.

If, when writing a number, we discard the degrees of the basis of the number system, then the number can be written in the following compact form: $M_{(p)} = a_k a_{k-1} \ldots a_1, a_0 a_{-1} \ldots a_{-m}$.

Compact representation of numbers, as well as convenient algorithms for adding, multiplying, led to the widespread use of positional number systems.

One of the first creators of electronic computers, Professor A. Atanasov, proposed to use a binary number system in computers. The set of symbols used to represent and process information in a computer is the smallest. It includes only two characters - 0 and 1, with a combination of which (sequences of ones and zeros) you can write any number, and may require a different number of bits (binary digits), as shown in table 1. Use in modern Computer binary representation of information, as noted earlier, is due to the convenience of technical implementation of devices for storing and processing information.

Table 8.1. Images of numbers in positional number systems used in computers

Decimal number system P = 10	Binary number system P = 2	Octal number system P = 8	Hexal number system P = 16
1	2	3	4
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Coding of information

Errors always occur when transmitting via communication channels. Their reasons may be different, but the result is the same - the data is distorted and can not be used on the receiving side for further processing. As a rule, the possibility of bit twisting in the transmitted data stream at the level of the physical channel is in the range of 10^{2} ... 10^{-6}. At the same time, users and many application processes often demand the possibility of errors in the received data not worse than 10^{-6} ...10^{-12}. Errors are dealt with at various levels of the seven-level OSI model (mostly in the first four). There are many different ways to deal with errors.

In one method, on the transmitting side, the transmitted data is encoded by one of the known error correction codes. On the receiving side, respectively, is decoding the received information and correcting the detected errors. The ability of the applied error-correcting code depends on the number of redundant bits generated by the encoder. If the redundancy is small, ie there is a risk that the received data will contain undetected errors, this can lead to errors in the application process. If you use code with high corrective power, it leads to low data rates. Thus, knowledge of the theory of noise-tolerant coding allows to determine the optimal parameters of noise-tolerant code depending on the task.

Lecture №9. ADC serial arithmetic, bitwise equilibrium, parallel type. Integrating ADCs, multichannel ADCs. Structural methods to improve the characteristics of the ADC. Digital-to-analog converters (DAC). Digital meters of electrical and non-electrical quantities

Analog-to-digital converters (ADC)

Analog-to-digital converters (ADCs) are devices that receive input analog signals and generate corresponding digital signals that are suitable for processing by microprocessors and other digital devices.

In principle, the possibility of direct conversion of various physical quantities into digital form is not excluded, but this problem can be solved only rarely due to the complexity of such converters. Therefore, now the most rational way is to convert quantities of different physical nature first into functionally related electrical, and then with the help of voltage-to-code converters - in digital. It is these converters that are meant when talking about ADCs.

The procedure of analog-to-digital conversion of continuous signals, which is implemented using an ADC, is the conversion of a continuous function of time $U(t)$, which describes the input signal, in a sequence of numbers $\{U'(t_j)\}$, j = 0,1,2,…, which are related to some fixed points in time. This procedure can be divided into two separate operations: sampling and quantization.

The most common form of sampling, as noted, is uniform sampling, which is based on the sample theorem. According to this theorem a_j, the instantaneous values of the signal at discrete moments of time $U(t_j)$ should be used as coefficients $t_j = j\omega t$, and the sampling period should be chosen from the condition: $t = 1/F_m$, where F_m – maximum frequency of the spectrum of the converted signal.

Then we obtain the well-known expression of the reference theorem $U(t) = \sum_{j=-\infty}^{\infty} U(j\omega t)\frac{\sin\left[2\pi F_m(t - j\Delta t)\right]}{2\pi F_m(t - j\Delta t)}$.

For signals with a strictly limited spectrum, this expression is an identity. However, the spectra of real signals go to zero only asymptotically. The

application of uniform sampling to such signals causes the emergence of specific high-frequency distortions in information processing systems, which are due to the sample. To reduce these distortions, it is necessary to either increase the sampling rate or use an additional low-pass filter in front of the ADC, which will limit the spectrum of the input signal before its analog-to-digital conversion.

In the general case, the choice of sampling rate will also depend on the type of function $f_j(t)$ used in the first section formula and the allowable level of errors that occur when restoring the initial signal according to its samples. All this must be taken into account when choosing the sampling frequency, which determines the required ADC speed. This parameter is often set to the ADC developer.

Consider in more detail the place of the ADC when performing a sampling operation.

For sufficiently narrowband signals, the sampling operation can be performed using the ADCs themselves and thus combined with the quantization operation. The main regularity of such sampling is that due to the finite time of one transformation and the uncertainty of its completion, which, in general, depends on the parameters of the input signal, it is not possible to obtain a clear correspondence between sample values and time to which they should be attributed. As a result, when working with signals that change over time, there are specific errors, dynamic in nature, to assess which introduce the concept of aperture uncertainty, which is mainly characterized by aperture time.

Aperture time t_a is the time during which the uncertainty between the value of the sample and the time to which it refers remains. The effect of aperture uncertainty is manifested either as an error of the instantaneous value of the signal at a given measurement time, or as an error of the time at which the measurement is performed at a given instantaneous value of the signal. With uniform sampling, the consequence of aperture uncertainty is the occurrence of amplitude errors, which are called aperture and numerically equal incremental signals during aperture time.

If we use a different interpretation of the effect of aperture uncertainty, then its presence causes "shaking" of the true moments of time in which the signal is taken, relative to the moments that are equidistant on the time axis. As a result, instead of

uniform sampling with a strictly constant period, sampling with a fluctuating repetition period is performed. This causes a violation of the conditions of the sample theorem and the appearance of the already considered aperture errors in digital information processing systems.

This value of the aperture error can be determined by decomposing the expression for the input signal into a Taylor series near the reference points, which for the j point has the form: $U(t) = U(t_j) + t_a U'(t_j) + \frac{t_a^2}{2} U''(t_a) + \ldots$ and in the first approximation gives the aperture error: $\Delta U_a(t_j) \approx t_a U'(t_j)$ where t_a – aperture time, which for this case in the first approximation is time ADC conversion.

A sinusoidal test signal is usually used to estimate aperture errors $U(t) = U_m \sin \omega t$.

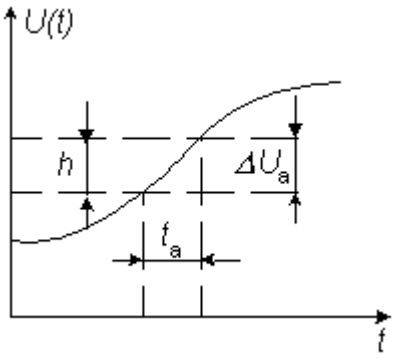


Fig. 9.1

Assuming that for a N-bit ADC with a resolution 2^{-N}, the aperture error should not exceed the quantization step (Fig. 9.1), the relationship between signal frequency ω, aperture time t_a and relative aperture error will be as follows: $1/2^N = \omega t_a$.

To ensure the sampling of the sinusoidal signal, the frequency of which is 100 kHz with an error of 1%, the ADC conversion time must be equal to 25 ns. At the same time, with the help of such a high-speed ADC, it is possible in principle to sample signals that have a spectrum width of about 20 MHz. Thus, sampling using the ADC itself causes a significant difference in requirements between the speed of the ADC and the sampling period. This difference reaches 2 ... 3 orders of magnitude

and greatly complicates and increases the cost of the sampling process, because even for relatively narrowband signals requires a fairly fast ADC. For a fairly wide class of signals that change rapidly, this problem is solved with the help of sampling and storage devices that have a small aperture time.

ADC serial arithmetic, bitwise equilibrium, parallel type

There are now a large number of methods for converting voltage - code. These methods differ significantly from each other in potential accuracy, conversion speed and complexity of hardware implementation. Fig. 9.2 shows the classification of ADC by conversion methods.

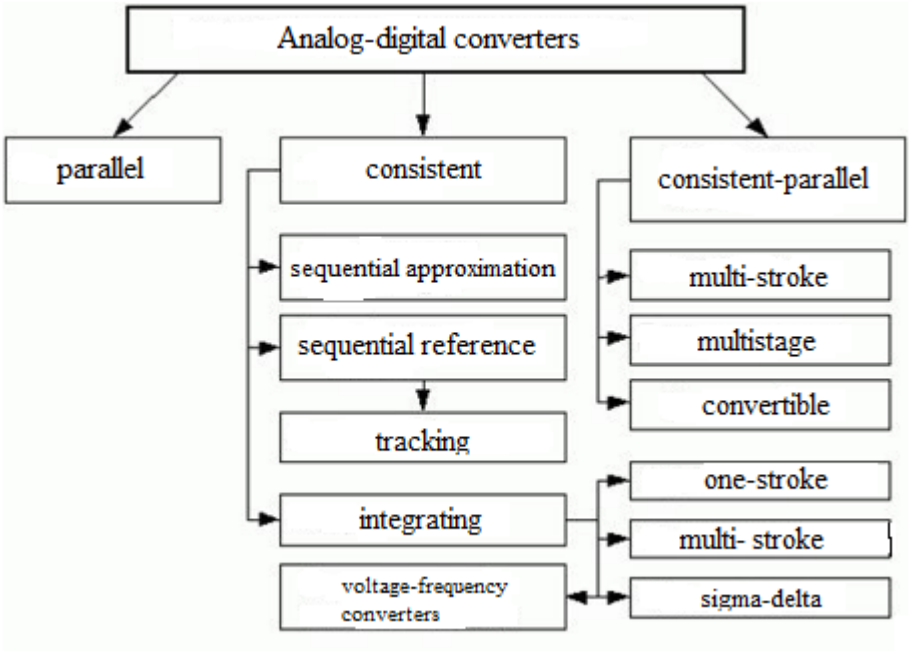


Fig. 9.2

The classification of ADCs is based on a feature that indicates how the process of converting an analog quantity into a digital one unfolds over time. The conversion of sample signal values into digital equivalents is based on quantization and coding operations. They can be performed using either sequential, or parallel, or sequential-parallel procedures of approximation of the digital equivalent to the converted value.

The operation of analog-to-digital conversion by the method of **sequential calculation** can be illustrated using the block diagram in Fig. 9.3.

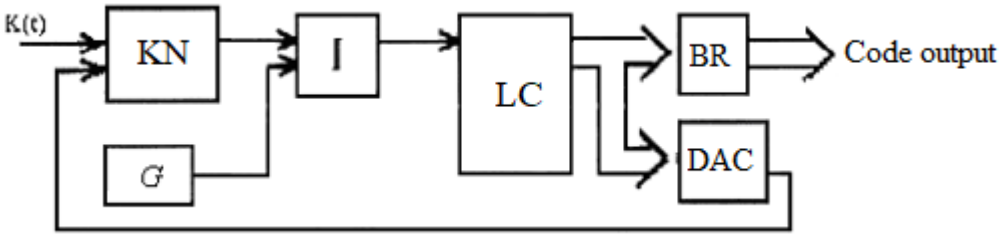


Fig. 9.3

The circuit includes: clock signal generator (G), voltage comparator (KN), circuit I, counter (LC), buffer register (BR), digital-to-analog converter (DAC). The scheme works as follows. An analog signal is fed to the input of the converter $x(t)$, which is connected to one of the inputs of the voltage comparator KN. The second input of the comparator is supplied with a reference voltage (U_{et}), which is formed at the output of the DAC under the control of a circular word at the output of the LF. The comparator generates at its output a signal of either a logical unit or a logical zero, depending on which value is greater. If $U_{et} < x(t)$, then at the output of the comparator a unit is formed, which allows the passage of pulses from the clock generator through the circuit I to the counting input of the LC counter. At the output of the counter is the process of counting these pulses in binary code from 2^0 to 2^{n-1}. The binary code from the LF is fed to the input of the DAC, at the output of which a stepped signal is formed U_{et}. Each step of this signal corresponds to the level of the sampling interval q. The signal U_{et} is compared with the signal $x(t)$ and at the moment when $x(t)$ it becomes less than U_{et}, a signal of logical zero is formed at the output of the comparator. Scheme I is closed, the counter stops counting and the typed binary code is overwritten in the output buffer register BR for issuance to the user.

The method of **direct reading is** implemented using the so-called ADC of parallel action. This converter has a line of 2^{n-1} voltage comparators, the first inputs of which are parallel and a signal is applied to them $x(t)$. The outputs of the reference voltage divider are connected to other inputs. The outputs of the comparators are connected to a single code to binary converter. The conversion process is carried out in one cycle, and at the output of the line of comparators to the

comparator, which will record $x(t) < U$ will be a wave of ones, and then a wave of zeros of a single code. The structural and functional scheme of transformation is shown in fig. 9.4, and the time diagram is similar to fig. 9.6.

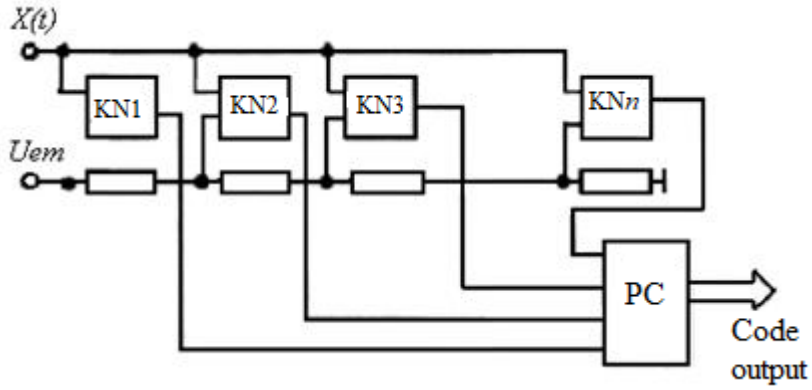


Fig. 9.4

The most common method is bitwise balancing, which provides a conversion time from 1 μs to 1 ms. The structural and functional scheme of transformation is shown in fig. 9.5, and the time diagram - in Fig. 9.6.

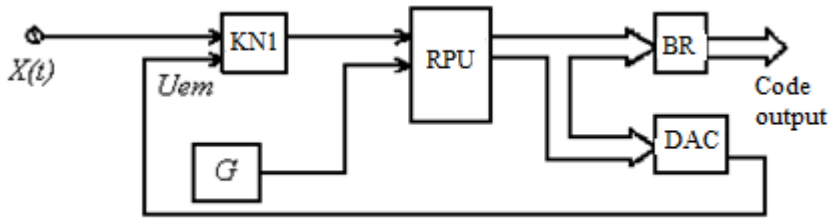


Fig. 9.5

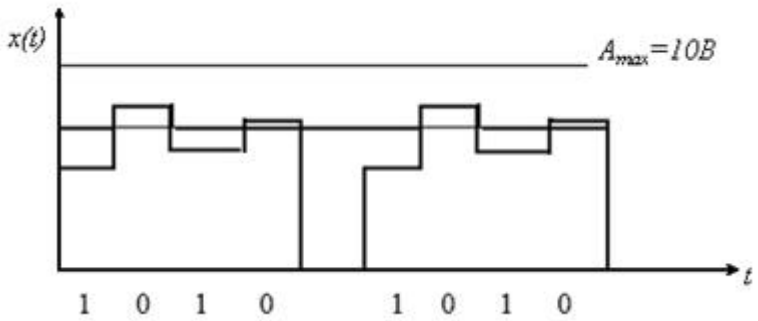


Fig. 9.6

The scheme works like this. The input signal is fed to the ADC input $x(t)$, which is compared with the reference signal U_{et} generated at the output of the DAC. The DAC consists of a $3n$ set of reference signal sources, which are controlled by a special register of bitwise balancing (RPU). The conversion takes place at n time intervals. Moreover, on the first cycle, the ACU forcibly turns on the first digit of the DAC. The value of the first digit of the reference values at the output of the DAC is equal to half the range of signal conversion. Then at the end of the first clock interval the comparator makes a comparison $x(t)$ with U_{et}. If $x(t) < U_{et}$, then the forcibly enabled senior bit of the DAC remains on until the end of the conversion process. This is provided under the control of a certain signal at the output of the comparator (1 or 0). If so $x(t) > U_{et}$, the first digit is turned off at the beginning of the second bar. At the beginning of the second cycle, the second digit of the DAC is forcibly switched on and the comparison $x(t)$ with U_{et}. The procedure is repeated until all bits of the DAC take part in the balancing process. As a result, the ADC output generates a code corresponding to the input signal.

Integrating ADC

It is known that the disadvantage of serial ADCs is the low noise immunity of the conversion results. Indeed, the sampling of the instantaneous value of the input voltage preferably includes the term in the form of the instantaneous value of the interference. Eventually, digital processing of a sample sequence can suppress this component, but it takes time and computational resources. Preferably, in the ADC, the input signal is integrated either continuously or in a certain time range, the duration of which is usually selected as a multiple of the periodic noise. This allows in many cases to dampen the interference at the conversion stage. The price for this is the reduced speed of integrating ADCs.

A simplified diagram of the ADC, which works in two main strokes (ADC two-stroke integration), is shown in Fig. 9.7.

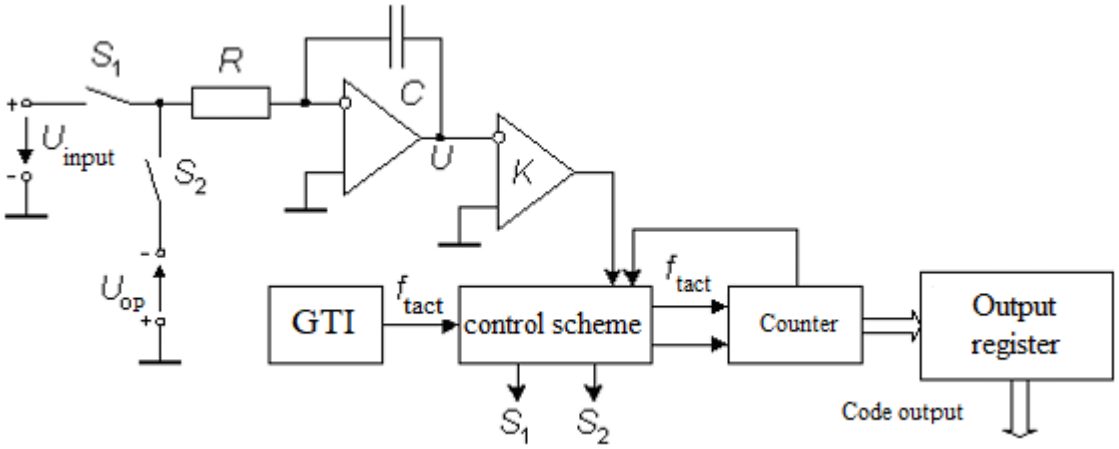


Fig. 9.7

The transformation takes place in two stages: the integration stage and the calculation stage. At the beginning of the first stage, the key S_1 is closed and the key S_2 is open. The integrator I integrates the input voltage U_{output}. The input voltage integration time t_1 is constant; as the timer is used a counter with a counting factor K_c, so that $t_1 = K_c / f_{tact}$.

Until the end of the integration, the output voltage of the integrator is $U_{output}(t_1) = -\frac{1}{RC}\int_0^{t_1} U_{input}(t)dt = -\frac{U_{input\,\text{mid}} \cdot K_c}{f_{tact} \cdot RC}$ where $U_{input\,\text{mid}}$ – the average on time t_1 value of the input voltage.

At the end of the integration stage, the key S_1 opens, and the key S_2 closes and the reference voltage U_{ref} is fed to the input of the integrator. The reference voltage opposite to the sign of the input voltage is selected. At the stage of calculation, the output voltage of the integrator decreases linearly in absolute value, as shown in Fig. 9.8.

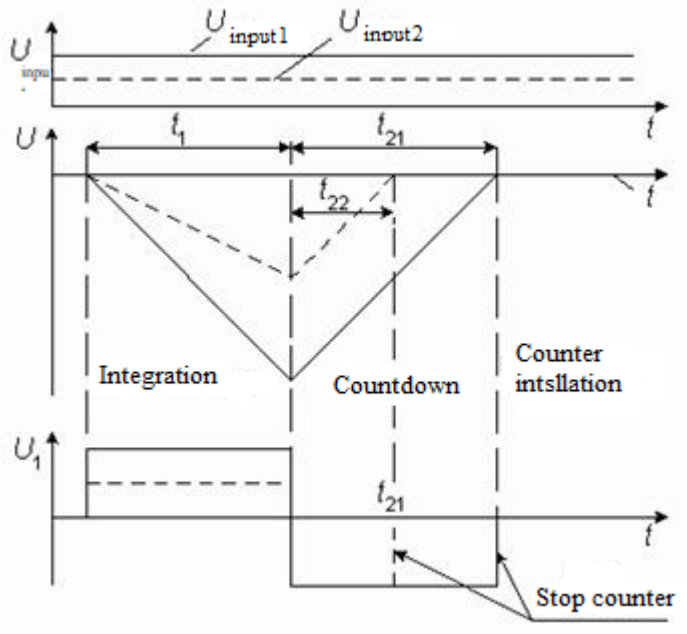


Fig. 9.8

The counting stage ends when the output voltage of the integrator goes through zero. The comparator K is switched and the calculation is stopped. The time range in which the calculation stage takes place is determined by the equation $U_{output}(t_1)+\frac{1}{RC}\int_0^{t_1+t_2} U_{ref}dt=0$.

Next, performing simple mathematical operations and taking into account that: $t_2=\frac{n_2}{f_{tact}}$ where n_2 – the contents of the counter after the end of the counting stage, we obtain the result $n_2=\frac{U_{input\ mid}\cdot K_c}{U_{ref}}$.

From this formula it follows that the distinguishing feature of the method of multi-stroke integration is that neither the clock frequency nor the constant integration RC does not affect the result. It is only necessary that the time frequency t_1+t_2 remains constant over time. This can be ensured by using a simple clock generator, because significant time or temperature frequency drifts occur for a time that is longer than the conversion time.

In deriving the previous expressions, we saw that the final result does not include the instantaneous values of the converted voltage, but only the values

averaged over time t_1. Therefore, the AC voltage is weakened the higher its frequency.

Let us determine the noise attenuation coefficient K_a for the two-stroke integration ADC. Let the input of the integrator receive a harmonic signal of unit amplitude with a frequency f with an arbitrary initial phase φ. The average value of this signal during integration t_1 is equal to

$$U_{av} = \frac{1}{t_1}\int_0^{t_1} \sin\left(2\pi ft + \varphi\right) dt = \frac{\sin\left(\pi ft_1 + \varphi\right)\sin \pi ft_1}{\pi ft_1}.$$

When this value reaches a modulo maximum, then $K_a = \left|\frac{\sin^2 \pi ft_1}{\pi ft_1}\right|$.

The frequency characteristic of the attenuation coefficient of ADC two-stroke integration is shown in Fig. 9.9.

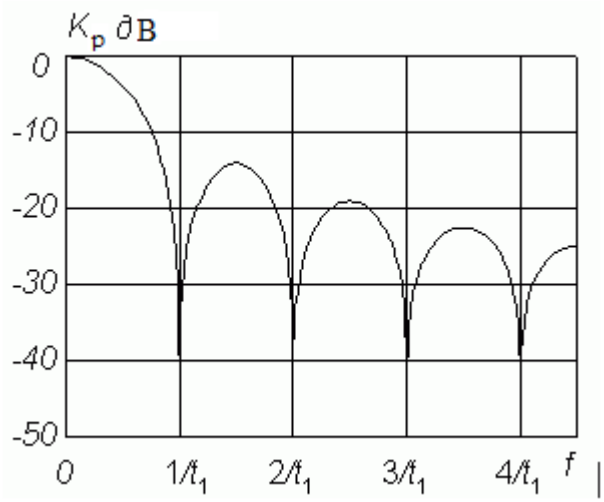


Fig. 9.9

As we see, the alternating voltage, the period of which is an integer times smaller t_1, is completely muted. Therefore, it is advisable to choose a clock frequency such that the t_1 product is equal $K_c \cdot f_{tact}$ to or multiple of the voltage period of the industrial network.

Multichannel ADC

Multi-channel ADCs are quite common today, especially where you want to combine information from several sources, ie, for example, from different sensors. Such ADCs can be used, for example, to monitor input voltages, control

extreme values, record readings, control outputs (loads), and so on. The scheme of multichannel ADC UM-ADC1 based on the microcontroller RIC16F876A is shown in Fig. 9.10.

The commercial version of this device has 40 inputs and outputs, but their number may be different.

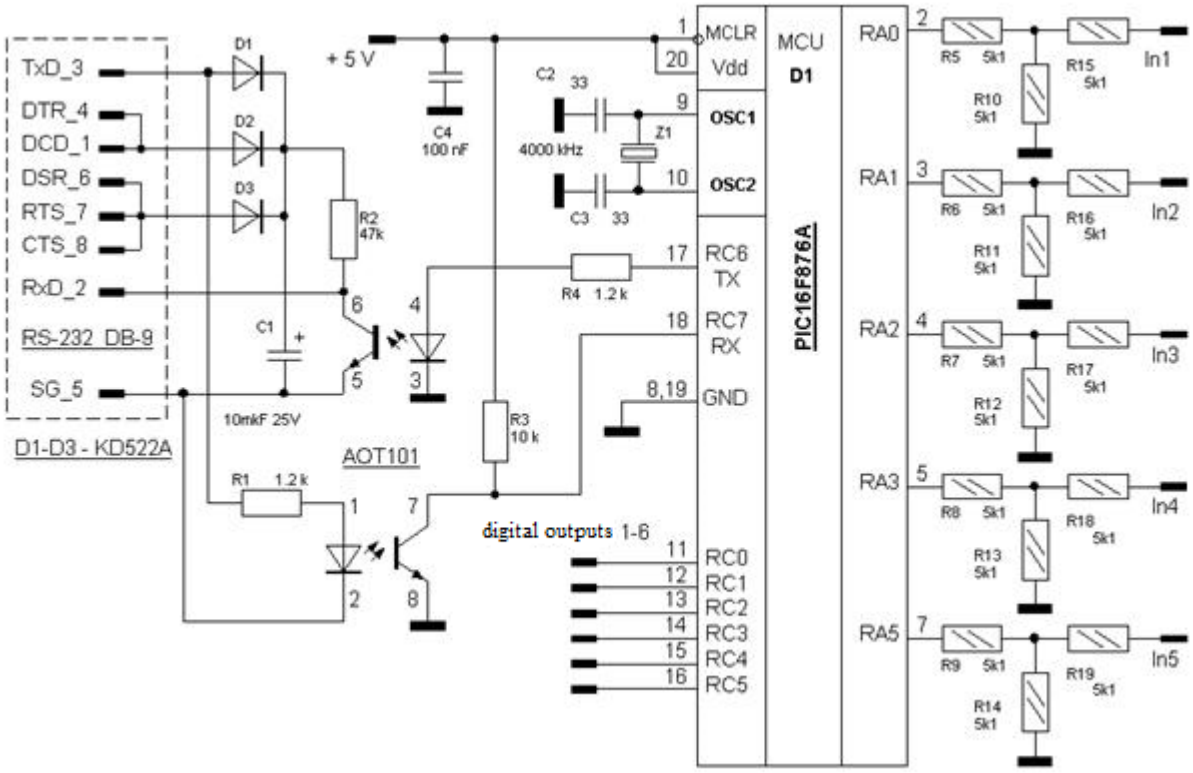


Fig. 9.10

Structural methods to improve the characteristics of the ADC

It should be noted that high accuracy is achieved through the improvement of the element base, the manufacturing process, and the materials used. Certain deviations of the weights of the discharges from the required values are usually corrected by laser adjustment in the manufacture of ADC resistors. This requires an increase in the area of the intracrystalline components and the crystal as a whole, and there is a problem of extracting crystal materials during fitting. These processes disrupt the structure of the materials of the components, reduce the time and temperature stability of the circuit.

A more promising approach is one that avoids physical impact on circuit elements. For example, in the case of using a DAC based on a binary number system,

the reduction of static errors is achieved by correcting the original value by introducing a correction in analog form, which is formed by an additional corrective DAC. In this case, the converted code K_{input} is fed as shown in Fig. 9.11, at the input of the main DAC and the digital computing device (DCD).

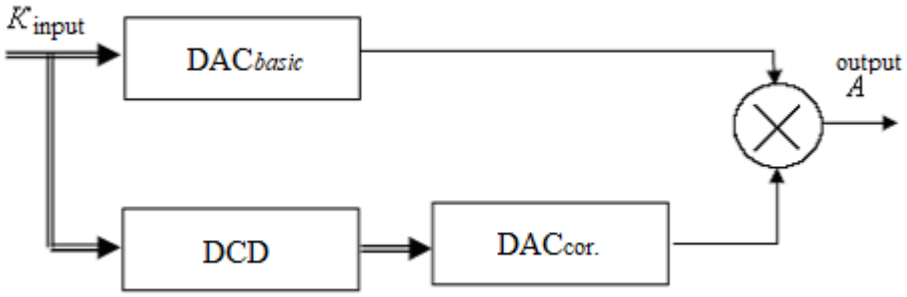


Fig. 9.11

The DCD calculates the correction code received at the input of the corrective DAC. The result of the conversion A_{output} is formed using the adder $\otimes$ of analog values. But the application of this principle in analog-to-digital conversion gives some reduction in the speed of devices.

They also use the introduction of devices in the development of information redundancy in the form of redundant positional number systems (NPSC), which comprehensively improves several characteristics of analog-to-digital conversion. Increasing the bit grid of the device (and consequently increasing the number of cycles in bitwise balancing) increases the accuracy of medium and high speed ADCs implemented on coarse analog nodes, and on the other hand increases the speed of high precision ADCs based on medium speed.

Increasing the speed of multi-bit ADCs is achieved in two ways. The first focuses on the use of a more advanced element base, which is not easy enough. The second way is associated with the introduction of redundancy, mostly structural, in the design.

Digital-to-analog converters (DAC)

The need to perform the operation of restoring the output signal from discrete samples, as well as the need to perform operations of generating reference signals for analog-to-digital conversion raises the problem of digital-to-analog conversion. The

essence of the operation of digital-to-analog conversion is to generate analog signals that correspond to the code words of the discrete signal. Technically, this formation is carried out by a digital-to-analog converter (DAC).

The analog signal at the output of the DAC can be generated by multiplying the reference voltage $E_{ref} = q$ by the weight bit coefficients of the codeword $a_i = 2^i$, so that $U_{output} = q(a_0 2^0 + a_1 2^1 + a_2 2^2 + ... + a_{n-1} 2^{n-1})$.

Technically, the simplest DAC is implemented on the principle of summation of discharge currents $U_{output} = \sum I_1 R_{ref} = R_{ref}(a_0 I_1 + a_1 I_2 + a_2 I_3 + ... + a_n I_n)$ (Fig. 9.12).

The DAC implementation circuit for current summation contains a stable voltage source E_0, a matrix of binary-weighted resistors $\left(R \cdot 2^i\right)$, a set of switches $K \cdot E_i$ that implement bit coefficients and a current-to-voltage converter on the operational amplifier OP.

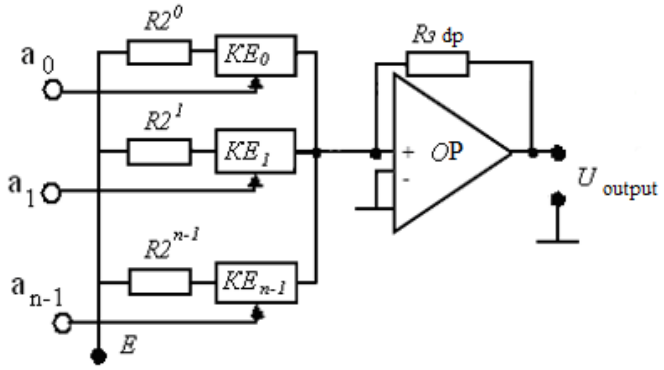


Fig. 9.12

The time diagram of the classical process of digital-to-analog conversion has the form (Fig. 9.13).

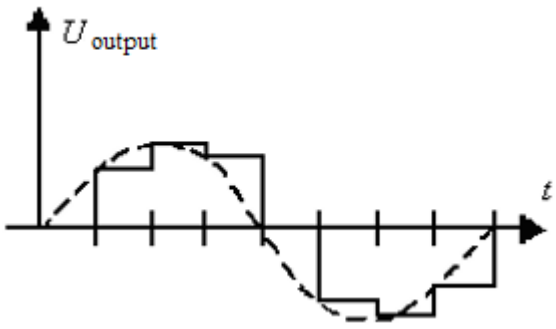


Fig. 9.13

With a small number of discrete samples of instantaneous signal values, this signal bears little resemblance to the original, but can be approximated by analog filtering or interpolation.

The main types of electronic DACs

1) Pulse width modulator is the simplest type of DAC. A stable current or voltage source is periodically turned on for a time proportional to the converted digital code, then the resulting pulse sequence is filtered by an analog low-pass filter. This method is often used to control the speed of electric motors, and is also becoming popular in Hi-Fi audio.

2) DAC resampling, such as delta-sigma DACs, are based on variable pulse density. Resampling allows you to use a DAC with a lower bit rate to achieve a higher bit rate of the final conversion; often delta-sigma DAC is based on the simplest single-bit DAC, which is almost linear. The low-bit DAC receives a pulse signal with modulated pulse density (with a constant pulse duration, but with variable duty cycle), created using negative feedback. Negative feedback acts as a high-pass filter for quantization noise. Most larger bit DACs (more than 16 bits) are built on this principle due to its high linearity and low cost. The speed of the delta-sigma DAC reaches hundreds of thousands of readings per second, bit rate - up to 24 bits. To generate a signal with modulated pulse density, you can use a simple delta-sigma modulator of the first order or higher order as MASH (English. Multi stage noise SHaping). As the oversampling frequency increases, the requirements for the low-pass output filter decrease and the quantization noise attenuation improves.

3) A weighing DAC in which each bit of the converted binary code corresponds to a resistor or current source connected to a common addition point. The current of the source (the conductivity of the resistor) is proportional to the weight of the bit to which it corresponds. Thus, all non-zero bits of code are added with weight. The weighing method is one of the fastest, but it is characterized by low accuracy due to the need to have a set of many different precision sources or resistors. For this reason, DAC weights have a bit rate of no more than eight bits.

4) The R-2R circuit is a variation of the weighing DAC. In R-2R DAC weighted values are created in a special circuit consisting of resistors with resistances R and 2R. This allows you to significantly increase the accuracy compared to conventional DAC weighing, because it is relatively easy to make a set of precision elements with the same parameters. The disadvantage of this method is the lower speed due to the parasitic capacity.

5) The segment DAC contains one current source or resistor for each possible value of the output signal. For example, an eight-bit DAC of this type contains 255 segments, and a 16-bit - 65535. Theoretically, segment DACs have the highest speed, because to convert it is enough to close one key that corresponds to the input code.

6) Hybrid DACs use a combination of the above methods. Most DAC chips belong to this type, the choice of a specific set of methods is a compromise between speed, accuracy and cost of the DAC.

Digital meters of non-electric quantities

The perception of information about objects or processes is carried out by means of devices called primary transducers or sensors. In most cases, the sensors display the input information in the form of an equivalent electrical parameter. That is, the sensor is an element that takes a controlled parameter and converts it into a form suitable for further processing (measurement, transmission, control).

According to the scheme of switching on of sensors it is possible to define two groups of matching and normalizing devices. The first group includes devices in which the sensors are elements of voltage dividers (Fig. 9.14, a), the second - devices in which the sensors are elements of oscillating systems of RF generators (Fig. 9.14, b).

In the devices of the first group, the sensors are most often turned on according to the differential or bridge circuit (Fig. 9.14, c, d, respectively).

Differential circuits are highly stable, as destabilizing factors simultaneously act on both elements of the differential sensor, which compensates for this effect.

The sensors in the bridge circuit are part of the bridge, which is balanced at some (usually zero or initial) value of the controlled parameter.

When measuring some non-electrical quantities, it is not always possible to convert them directly into an electrical quantity. In these cases, a double conversion of the primary measured quantity into an intermediate non-electric quantity is performed, which is then converted into an initial electric quantity.

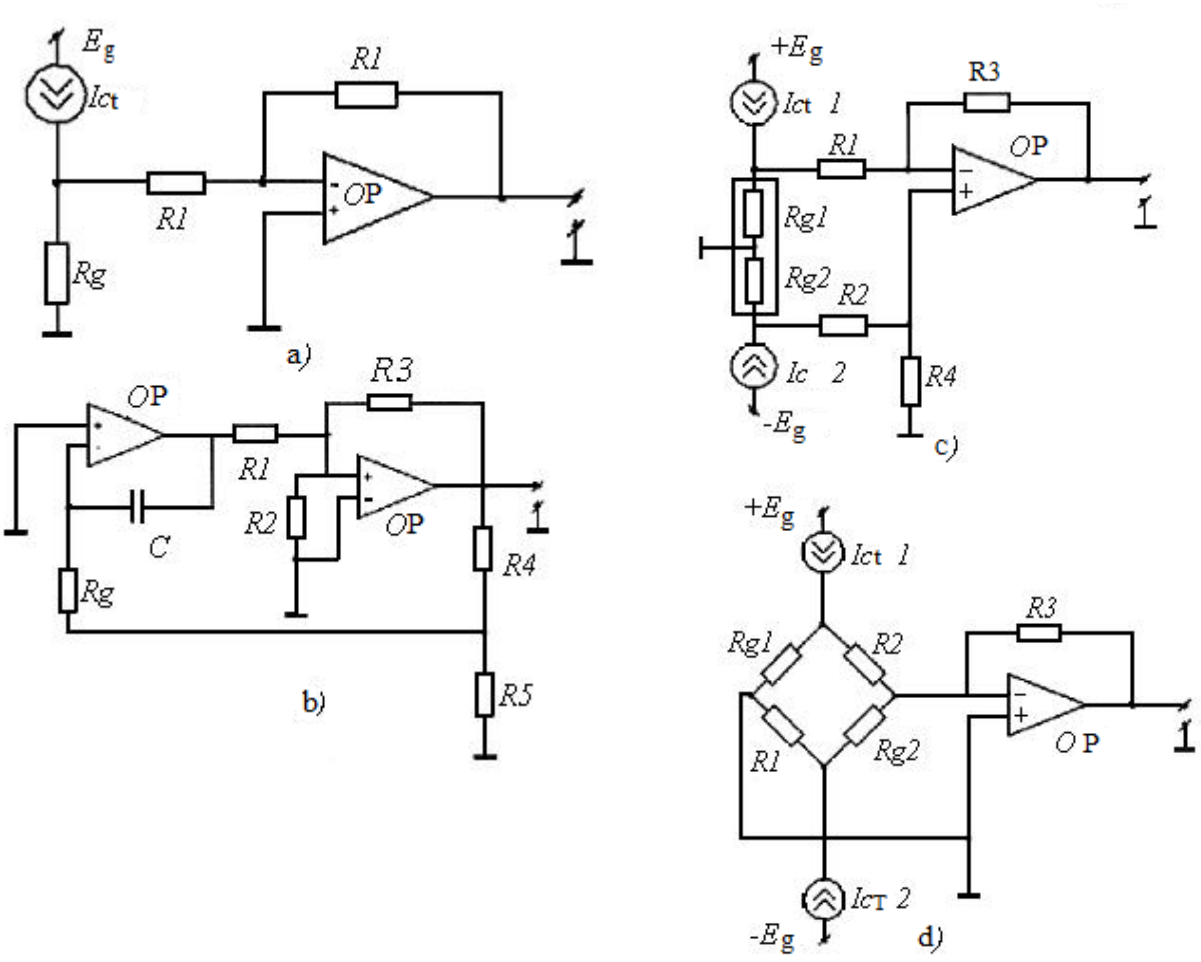


Fig. 9.14

a) with voltage dividers; b) with elements of RF generators; c) according to the differential switching scheme; d) according to the bridge switching scheme

The set of two corresponding measuring transducers form a combined sensor (Fig. 9.15).

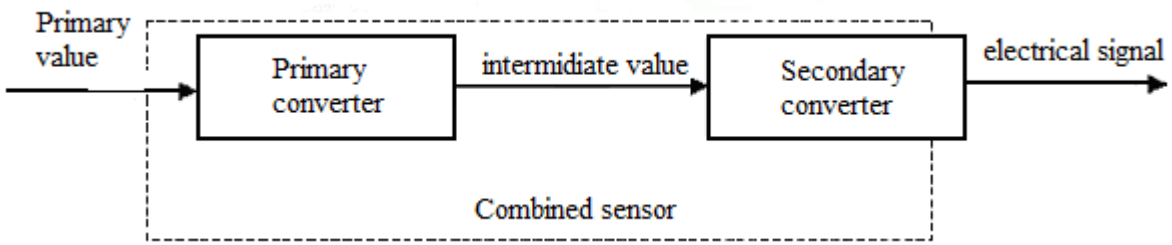


Fig. 9.15

Such transducers are convenient for measuring non-electrical (mechanical) quantities that cause deformation or displacement of the source element in the primary transducer, to which the secondary transducer is sensitive.

Lecture №10. Data multiplexing systems, simultaneous sampling. Sampling-storage schemes. Synchronization of multichannel systems

Multiplexing provides the ability to transmit several streams of information on a single line, which is carried out by fixing for each of them a fixed part of the resource of the line (bandwidth or busy time). The fixed resource allocation of the line remains unchanged even in periods of lack of information, ie the concentration function does not work. The inverse function is demultiplexing. Implementation in communication devices (multiplexers) of the multiplexing function is always combined with demultiplexing.

Time channel switching mode, which is based on the principle of synchronous time multiplexing when transmitting information from one switching node to another, is known as synchronous transfer mode STM (Synchronous Transfer Mode). Multiplexing in digital networks means the combination of n low-speed digital streams into one high-speed stream. Multiplexing is used in order to more efficiently use the bandwidth of the communication line, which led to the use in the terminology of communicators also the concepts of "compaction", "distribution" of the communication line. Output digital streams, which are formed as a result of different network applications (from different services), can differ significantly in nature. This includes the transmission of a constant bitstream, the transmission of 207 data files, speech and video signals in digital form. Thus, multiplexing also provides the adaptation of the transmission medium to a large number of disparate network applications. The digital stream of each application is a signal that meets the criteria and metrics of a particular information message to be transmitted. Time synchronous multiplexing is that the entire band of the signal propagation medium in the communication line for a short period of time, durationt, is alternately provided to the signals n applications. This period of time is called the "time slot", the interval Tc = nr, which corresponds to n time slots, is called the transmission cycle. A characteristic feature of synchronous time multiplexing is that in the multiplex signal, each initial signal corresponds to a time slot with a clearly fixed sequence number within the transmission cycle TC. It should be noted that due to multiplexing for the

signal of multimedia application (voice + video + data) provide several time slots. A device that receives multiple streams from different applications (voice, video, data) and transmits them to a line as a multiplex signal is called a multiplexer (MUX), and a device that performs the reverse function at the other end of the line is called a demultiplexer (DEMUX). MUX and DEMUX must work synchronously and in phase, as the time slots relative to TC at the input and output of the communication line must match. For this purpose, use devices with a high standard of frequency, called timers. Typically, in two-way (duplex) communication systems, the multiplexing and demultiplexing functions are combined in one device, which is also called a multiplexer. Modern time distribution multiplexers are channel-forming equipment. Their main difference from traditional sealing systems with pulse code modulation is that: • multiplexers allow you to transmit digital streams of different speeds (from different sources), so they are also called flexible multiplexers; • multiplexers, which have the property of "attachment / separation" (drop & insert), allow you to separate from the total stream part of the signals or add signals to the common linear stream. This allows you to build networks of complex topology.

Frequency Division Multiplexing (FDM) . The spectrum of each input signal is shifted to a certain frequency range. Frequency division multiplexing (FDM) is essentially an analog technology. FDM achieves a combination of several signals in one environment, transmitting signals in several different frequency bands in one environment (Fig. 10.1).

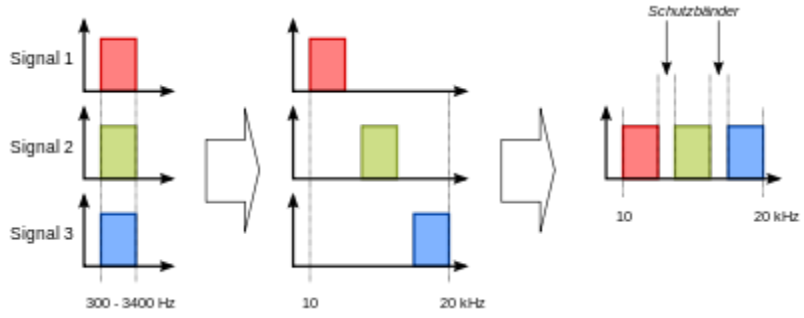


Fig. 10.1

In FDM, signals are electrical signals. One of the most common programs for FDM is traditional radio and television broadcasting from terrestrial, mobile or satellite stations or cable television. Only one cable reaches the customer's residential

area, but the service provider can send several TV channels or signals simultaneously on this cable to all subscribers without interference. The receivers must be tuned to the appropriate frequency (channel) to access the desired signal.

A variant of the technology called wavelength division multiplexing (WDM) is used in optical communication.

Time division multiplexing (TDM) is a digital (or, in rare cases, analog) technology that uses time, not space or frequency, to separate different data streams (Figure 10.2).

Fig. 10.2

TDM includes a sequence of groups of several bits or bytes from each individual input stream, one after another, and in such a way that they can be associated with a corresponding receiver. If this is done quickly enough, the receiving devices will not detect that some time of the circuit has been used to service another logical communication path.

Consider a program that requires four terminals at the airport to access a central computer. Each terminal communicated at 2,400 baud, so instead of purchasing four separate circuits to transmit such a low-speed transmission, the airline installed a pair of multiplexers. There are also a pair of 9600 baud modems and one dedicated analog communication scheme from the airport ticket office to the airline's data center. Some web proxy servers (eg polyps) use TDM in pipelining HTTP multiples of HTTP transactions on identical TCP / IP connections.

Multiplayer media sensor access and multi -point communication methods are similar to time division multiplexing because multiple data streams are time-separated on the same media, but because signals have separate sources rather than

combined into a single signal, it is best considered as access methods. to the channel, not the form of multiplexing.

TD is an outdated multiplexing technology that still provides the basis for most national fixed telephony networks in Europe, providing 2 m / bit voice and signal ports on narrowband call centers such as the DMS100. Each E1 or 2 m / bit TDM port provides a 30 or 31 time slot for CCITT7 signaling systems and 30 voice channels for client-connected Q931, DASS2, DPNSS, V5 and CASS alarm systems.

Multiplexing code division (CDM) , Code division multiple access (CDMA) or spread spectrum is a class of methods where multiple channels simultaneously using the same frequency spectrum, and the spectral bandwidth is much higher than the bit rate or symbol rate. One form is a frequency jump, the other is a direct sequence distribution spectrum. In the latter case, each channel transmits its bits as a coded channel-specific sequence of pulses, called chips. The number of chips per bit, or chips per symbol, is equal to the spread factor. This encoded transmission is typically performed by transmitting a unique time-dependent series of short pulses that are placed within the chip time over a larger bit time. All channels, each with a different code, can be transmitted on a single fiber or radio channel or other media and asynchronously demultiplexed. The advantages over conventional methods are that variable bandwidth is possible (as in statistical multiplexing), that wide bandwidth allows a poor signal-to-noise ratio according to the Shannon-Hartley theorem, and that multi-beam propagation in wireless communication you can fight with a rake-receiver. A significant application of CDMA is the Global Positioning System (GPS).

Sampling devices - storage

When collecting information and its subsequent conversion, it is often necessary to record the value of the analog signal at a certain point in time. Some types of analog-to-digital converters, such as serial approximation, can give completely unpredictable errors if their input signal is not detected during conversion. When changing the input code of digital-to-analog converters due to the non-simultaneous setting of bits, output voltage emissions are observed. To eliminate this phenomenon at the time of installation should also record the output signal of the

DAC. Sampling - storage (SSD) devices (tracking – storage) that perform this function must repeat the input analog signal at the output time interval (observation), and when switching the storage mode to store the last value of the output voltage until the sample signal (Fig. 10.3).

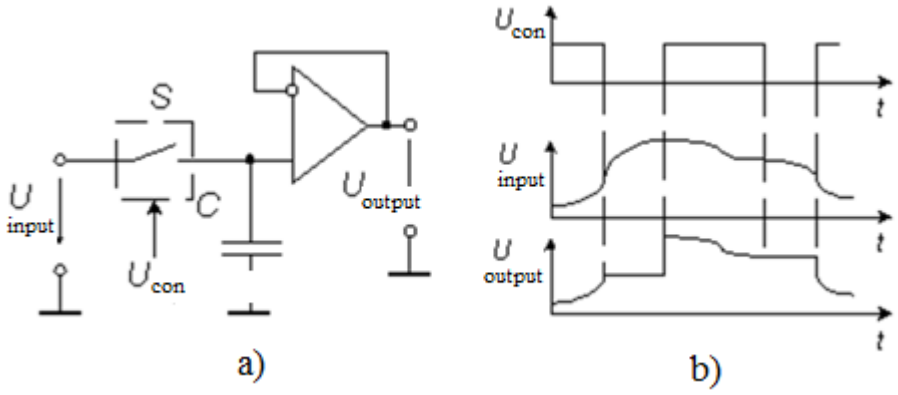


Fig. 10.3

The simplest circuit SSD has a number of disadvantages: when the key is closed, the input signal source has a significant capacitive load. If the source is the OP, it usually leads to its self-excitation. Memory with field-effect transistors at the input, which are used as buffer repeaters, have a significant zero offset.

The main characteristics of SSD

1. The zero shear stress $U_{cm,}$ is determined by the displacement of zero $OU_{1.}$

2. Drift of fixed voltage at a given capacitance $CU\delta_{output}\, t = I\, \delta / {}_p / C$, where I_p is the discharge current of the capacitor. It consists of leakage currents of the capacitor and switch, as well as the input current of the amplifier $OU_{2.}$ At a given leakage current, the amount of drift can be reduced by increasing the capacitance of the capacitor $C_{xp.}$ However, this degrades the dynamic characteristics of the circuit.

Dynamic characteristics

1. The sampling time tb determines how long under the most unfavorable conditions the process of charging the storage capacitor to the value of the input voltage with a given level of tolerance. This time is proportional to the capacity of C. The transfer of SSD to storage mode before the end of the sampling interval is fraught with significant errors.

2. Aperture delay t_a. This is the period between the moment of removal of the control voltage and the actual short circuit of the serial switch.

SSD sampling and storage devices are needed to reduce the dynamic errors that occur when sampling continuous signals that change over time. Their work is based on the principle of fixing the instantaneous value of the signal for the time required for the next conversion in analog-to-digital converters (ADC).

Existing SSD are divided into two classes: analog and digital. In analog devices, the transition from a continuous function $U(t)$ to a continuous sequence $\{U(t_n)\}$ (n = 1, 2, ...), quantization of sample values occurs in the ADC. In digital devices, the input signal is quantized first, and only then its sampling and storage of sample values in digital form.

The principle of construction

From a mathematical point of view, the operation of SSD is based on the operation of gating, which can ideally be performed using the filtering properties of δ-functions:

$$U(t_n) = \sum_{n=-\infty}^{\infty} U(t) \cdot \delta(t - t_n) dt .$$

Actual gating is performed using a sequence of deterministic gating functions g (t), which have a finite duration compared to the length of the input signals, and can be described by the expression where F is the symbol of the functional signal conversion within the gating pulse (Fig. 10.4).

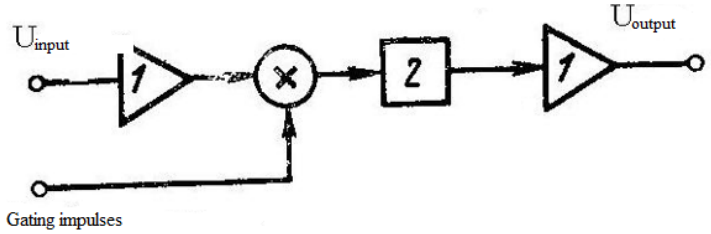


Fig. 10.4

However, the fundamental difference in the work of SSD is the presence of the mechanism of formation of samples. In particular, the case of sampling using the ADC itself can be considered as a kind of method.

Synchronization of multichannel systems

An N-channel communication system is a set of technical means that provide simultaneous and independent transmission of messages from N sources to N recipients on one communication line. Number of lines may be greater by one, but it is always less than N.

When building multi-channel systems (MCS) of information transmission, one of the most important is the task of separation (selection) of channel signals. You can use:

1) $2N$-poles. In this case, the separation does not require conversion of the shape of the primary signals, but BCS can be built only on a small number of channels;

2) separation, which requires the conversion of the shape of the primary signals and their compaction during transmission to the communication line.

The generalized structural scheme of MCS is shown in Fig. 10.5.

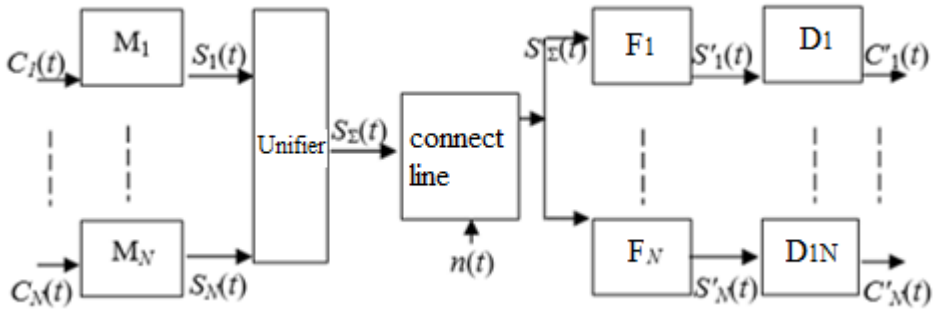


Fig. 10.5

Fig. 10.5 has the following denotations: $C_1(t)$, ..., $C_N(t)$ – primary signals (from information sources); M_1 , ..., M_N – individual channel converters; $S_1(t)$, ..., $S_N(t)$ – channel signals (converted primary); $S_\Sigma(t)$ – group linear signal; $n(t)$ – interference in the communication channel ; F_1, ..., F_N – channel dividers (filters); D_1 , ..., D_N – demodulators; dashes indicate the signals in the receiving part, which correspond to the signals of the transmitting part, but differ from them due to interference.

According to the method of forming a group signal MCS with the transformation of the shape of the primary signals can be divided into:

a) additive (separate), in which the group signal $S_\Sigma(t)$ is the sum of the channel; analytically it is written in the form $S_\Sigma(t) = S_1(t) + S_2(t) + ... + S_N(t) = \sum_i S_i(t)$;

b) combinations (formation $S_\Sigma(t)$ is carried out other operations, such as in the case of dual-channel system with binary messages - the transfer of communication line number combination of the output signal of the primary sources); This can be illustrated by the following table:

Combination number	1	2	3	4
Output 1 source	0	0	1	1
Output 2 sources	0	1	0	1
Frequency, which can encode this combination	f_1	f_2	f_3	f_4
Phase, which can encode this combination	φ_1	φ_2	φ_3	φ_4

c) mixed (combine both methods).

According to the method of separation of channel signals additive MCS are divided as follows:

1) with frequency separation of channels – channel signals differ in frequency, come to the transmission line at the same time, and the separation is carried out by frequency filters;

2) with time division of channels – channel signals differ in time of receipt in a line; signal separation is provided by switching devices, which in turn connect to the source line and information receivers;

3) with phase separation of channels – channel signals have the same frequency, but differ in phase; separated by phase detectors;

4) with the division of signals by form – channel signals modulate carriers that have a unique shape for each channel (code sequences); channel signals simultaneously arrive at the communication line and have overlapping spectra; a sign of signal separation is their shape;

5) with the division of channels by level – a sign of separation is the signal level (amplitude); channel signals are separated by nonlinear filters (for example, amplitude limiters);

6) spatial separation of channels – signals can have exactly the same electrical parameters and differ only in spatial position.

Lecture №11. Application of precision Σ-Δ ADCs in high-precision data measurement systems. Digital processing of audio signals

ADC multi-stroke integration has a number of disadvantages. First, the nonlinearity of the transient static characteristic of the operational amplifier on which the integrator is performed, significantly affects the integral nonlinearity of the conversion characteristic of the high-resolution ADC. To reduce the impact of this factor, ADCs are made multi-stroke. Another disadvantage of these ADCs is the fact that the integration of the input signal in the conversion cycle takes only about a third. Two thirds of the cycle, the converter does not receive the input signal. This degrades the noise-tolerant properties of the integrating ADC. Third, the ADC of multi-stroke integration must have a large enough number of external resistors and capacitors with a high-quality dielectric, which significantly increases the space that the converter occupies on the board and, consequently, increases the impact of interference

These shortcomings are largely eliminated in the design of the sigma-delta ADC. These converters owe their name to the presence of two blocks: an adder and an integrator. One of the principles inherent in this type of transducer, which reduces the error introduced by noise, and thus increase the resolution – is the averaging of measurement results over a long period of time

The main components of the ADC are a sigma-delta modulator and a digital filter. The operation of this circuit is based on the subtraction from the input signal $U_{in}(t)$ the magnitude of the signal at the output of the DAC, obtained in the previous cycle of the circuit (Fig. 11.1).

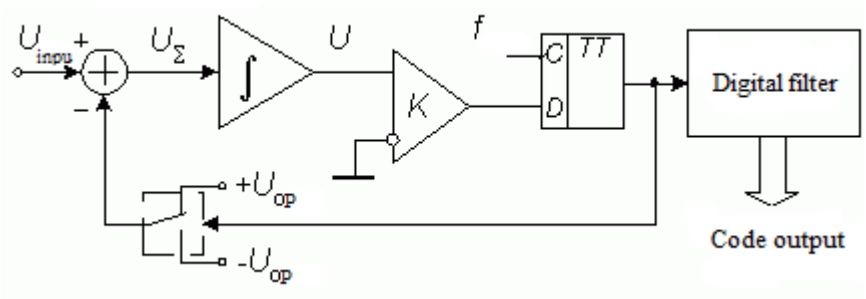


Fig. 11.1

Digital filters with amplitude-frequency response (frequency response) of the type $\left(\sin x/x\right)^3$ are usually used in sigma-delta ADC . The transfer function of such a filter in the *z*-domain is determined by the expression:

$$W(z)=\left[\frac{1-z^{-M}}{M(1-z^{-1})}\right]^3,$$

where *M* is an integer that is set programmatically and is equal to the ratio of the clock frequency of the modulator to the sampling frequency of the filter.

Comparison of the sigma-delta ADC with the ADC of multi-stroke integration shows significant advantages of the former. First of all, the linearity of the sigma-delta ADC transformation characteristic is higher than in the ADC of multi-stroke integration of equal cost. This is due to the fact that the sigma-delta ADC integrator operates in a much narrower dynamic range, and the nonlinearity of the transient characteristic of the amplifier on which the integrator is built, affects much less. The capacitance of the capacitor of the integrator in the sigma-delta ADC is much smaller (tens of picofarads), so that this capacitor can be made directly on the IC chip. As a result, the sigma-delta ADC has virtually no external elements, which significantly reduces the area it occupies on the board and reduces noise levels.

Sigma-delta high-resolution ADCs have a developed digital part that includes a microcontroller. This allows you to implement the modes of automatic zero setting and full-scale self-calibration, save the calibrated coefficients and transmit them at the request of the external processor.

The tracking ADC consists of a comparator, a reversible counter controlled by a comparator, and a DAC. The comparator compares the input voltage with the output voltage of the DAC and controls the direction of the number of the reversible meter (+/-). If the input voltage is greater than the voltage of the DAC, the meter under the action of clock pulses coming to its input counts for addition (+), if less than the voltage of the DAC, the meter counts for subtraction (-).

The main disadvantage of tracking ADCs is the low conversion rate. The conversion can take up to 256 cycles of 8-bit output, 1024 cycles for 10-bit values and so on. In addition, the conversion rate varies depending on the VBX.

Digital processing of audio signals

A digital audio signal is binary data where information is encoded using samples (numerical waveform values at each point in time). Each sample is a set of bits (values 0 or 1). 16-bit or 24-bit signals are commonly used. The number of samples per second in a digital signal is determined by the sampling rate, which is measured in hertz. The higher the sampling frequency, the higher the frequency of the audio signal.

Digital sound is fundamentally different from analog. In analog sound systems (for example, when recording on gramophone records, cassette tapes, etc.), acoustic air vibrations are converted to similar in shape by electric microphone and stored, and sound is reproduced in reverse - through an amplifier and converted to acoustic vibrations through a loudspeaker. Digital sound, on the other hand, converts acoustic oscillations into binary data (1/0).

Converting an analog signal to digital is done using a special device "analog-to-digital converter" (eg, computer sound card). In addition, the digital signal is often generated using digital synthesizers. To play a digital signal, it is converted back to analog and transmitted to the speaker.

The analog-to-digital converter measures the amplitude of the audio signal with a certain sampling rate (sample rate) and with a certain resolution (bit resolution). Accordingly, the main quality characteristics of digital audio are as follows:

Sampling frequency, which determines the frequency with which the signal amplitude is measured and measured in Hertz or kilohertz (kHz). According to Kotelnikov's theorem, the sampling frequency must be at least twice the highest frequency of the useful signal. Since a person perceives sounds with a frequency of up to 20 kHz, for high-quality audio sampling frequency must be at least twice as high as this frequency.

Amplitude resolution, which determines the accuracy with which the signal amplitude is measured. Amplitude resolution is measured in the number of bits allocated to record the value of the amplitude (sample). Since 1 bit = one bit in the binary system, this value is called the bit size, and the number of possible values of

the amplitude x and the number of bits following y is described by the relation $x = 2^y$. That is, for example, a 16-bit bit provides a record of 216 = 65,536 amplitude levels.

Increasing both parameters allows better digitization of sound, but also increases the amount of data. Therefore, in practice, different standards of sampling and bit rate are used. For example, a standard Audio CD has a sampling rate of 44.1 kHz (44,100 samples per second) and a 16-bit bit rate for each channel (stereo). Instead, DVD-Audio can use a sampling rate of up to 192 kHz and a bit rate of up to 24 bits.

If the recorded signal includes frequencies higher than the maximum allowable cut-off frequency (English Nyquist frequency), when it is digitized, there is an effect of overlapping frequency spectra (English aliasing). To prevent this effect, a spectrum overlay protection filter is required, which limits the signal spectrum to a cutoff frequency.

Another side effect of digitizing sound is quantization noise, which occurs due to the rounding of amplitude values. Quantization noise is perceived as a rather unpleasant distortion at a frequency of 3-5 kHz. To reduce this effect, dithering is used, the effect of adding a pseudo-random signal to the signal. Although the overall noise level during dithering increases, the subjectively perceived unpleasant effect decreases.

The task of both analog and digital systems is to reproduce sound with maximum quality. However, there are a number of obstacles to achieving the desired result.

Analog intrinsic noise level, which depends on the capacitance and inductance that limit the bandwidth of the system, as well as the resistance that limits the amplitude.

Digital quantization noise that depends on the sampling frequency that limits the bandwidth as well as the bit rate that limits the dynamic range.

High-quality components are required to achieve higher recording quality, which increases the overall cost of the equipment.

After digitization, the digitized audio signal may be digitally processed, which may include the use of filters or sound effects.

The digital audio can then be stored or transmitted. Digital audio is stored on CDs, iPods, hard drives, or any other digital media. Audio data compression in formats such as MP3, Ogg Vorbis, or AAC is commonly used to reduce file size.

The last step in working with digital audio is the inverse conversion to analog format using a digital-to-analog converter (DAC). Like the ADC, the DAC operates at a given sampling rate and bit resolution, and the modulation frequency may differ from that used in the ADC. In this case, the processes of resampling, increasing or decreasing the modulation frequency (Fig. 11.2).

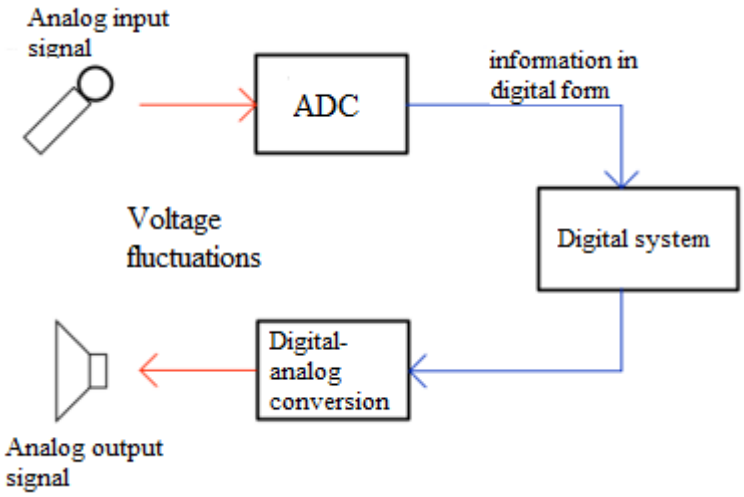


Fig. 11.2

Lecture №12. Types of data transmission channels. Consistency of signals and communication channels

In MCS with phase separation, the carrier oscillations of all channels have the same frequency, but differ in phase. The bandwidth of the group signal does not depend on the number of channels. Such BCS are the most narrowband. However, it is found that the qualitative separation of channel signals is possible if the BCS with phase compaction is built on only 2 channels and the phase shift $\Delta\varphi$ between the carrier oscillations is $\pi/2$.

In practice, phase separation is used in combined BCS, in which the frequency compression on the carrier oscillation of each channel by means of phase compression transmit independently two messages.

The scheme of two-channel BCS with phase separation and AM has the following form (Fig. 12.1).

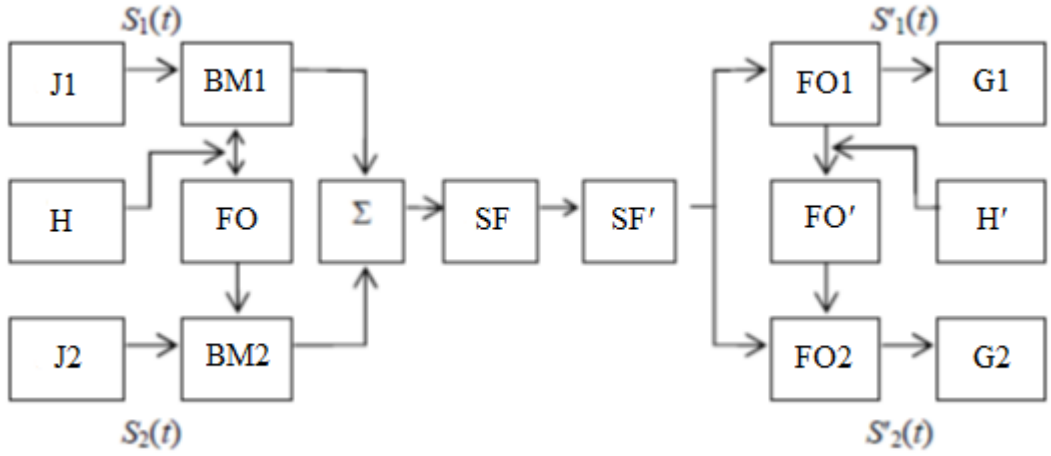


Fig. 12.1

Symbols in the figure: J – source of the primary signal; BM – balance modulator; G_n – carrier frequency generator; FO – phase shifter; Σ – adder; SF – bandpass filter; G1, G2 is the recipient of the primary signal.

The indices in the figure correspond to the blocks of the first and second channels, and the strokes – to the blocks of the receiving part, which are similar to the components of the transmitter.

Advantages of such transmission systems : possibility of construction of narrowband multichannel systems; as part of the combined MCS the ability to increase the number of channels without expanding the frequency band.

To deficiencies should include the need to provide synchronous and phase of generator bearing vibrations.

Multi-channel systems with time division of channels are widely used for transmission of analog and discrete information. The block diagram of MCS, which explains the principle of time compaction, is presented in Fig. 12.2. Information from analog sources $J_1 ... J_N$ is sampled and transmitted to the line by a sequence of pulses. To do this, the transmitting and receiving devices MCS contain special switches, which periodicall for a short time connect the line simultaneously to the sources and recipients of information $O_1 \ldots O_N$ of each channel.

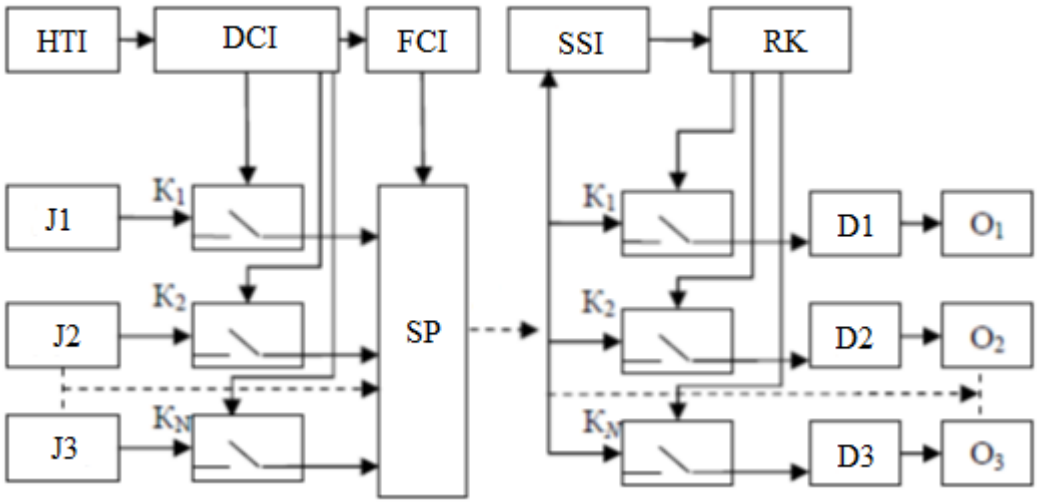


Fig. 12.2

Switches are electronic switches $K_1 ... K_N$, the state of which is controlled by signals from the switches of the transmitter and receiver channels. Switch channels transmitter consists of a generator of clock pulses HTI, which generates pulses with a repetition rate $F_N = N\ F$, where N – number of system channels and F – pulse repetition frequency of one channel generator and distributor channel impulse DCI. The frequency of repetition is determined according to the theorem Nyquist with consideration of the actual performance of equipment. Thus , to ensure the required accuracy of signal recovery

take $F_d = (2,5...5)F_B$, where F_B is the upper limit frequency of the spectrum of the transmitted signal . The sequences of pulses of HTI and DCI turns coming to the keys corresponding channels and offer them for a time equal to the duration of the control pulses. Simultaneously, the circuit corresponding channel key receiver K'_1 ... K'_N. When closed key K $_1$ transmitter key K'_1 receiver, then a zaknene range transmission of information on the first channel. At simultaneous closing of keys K_2 and K'_2 the circle of the second channel, etc. is formed. After closing the keys K_N, K'_N ends the first cycle of transmission duration T_n, then the process is repeated.

In the receiver from the outputs of the keys K'_1 ... K'$_N$ signals are fed to the demodulators D_1 ...D_N, which allocate the transmitted messages. These messages are transmitted more recipients information O_1 ...O_N.

It follows from the principle of transmission that for the normal operation of the BCS with time separation it is necessary to ensure the operation of the keys in the same sequence and with the same time ratios as on the transmitting side, i.e. to ensure their synchronous operation. To do this, the transmitter has a device for generating clock pulses FSI, which are transmitted at the end of each transmission cycle . These pulses in the receiver are allocated by the selector of synchronizing pulses SSI and control work of the divider of channels of RK.

BCCs with time compaction are characterized by a wider spectrum than BCSs with frequency compaction. For example, a 2048 kHz band is required to transmit 32 PM channels in a time-compacted system, while 128 kHz would be sufficient for a frequency-compressed system. However, this provides a higher reliability of reproduction of transmitted data and a greater range of services.

When constructing BCS with time division, the possible number of channels is limited by two factors: on the one hand, taking into account the above requirement for the sampling period of primary signals, the signal of each channel must reach the transmission line no later than through $T_d = \frac{1}{2F_B}$. On the other hand, it is not always possible to increase the number of simultaneously working subscribers by reducing the duration of pulses from each channel due to the limitation of the spectrum of the

group communication channel. It is known that for a sequence of rectangular pulses the ratio is valid $\Delta f \approx \frac{1}{\tau_{imp}}$. Therefore, the shortening of the channel signals, possible in terms of the accuracy of the reproduced information, is undesirable due to the increase in the amount of information transmitted and the need to expand the bandwidth of the communication channel.

Lecture №13. Shannon's theorem. Effective coding. Shannon-Fano and Huffman codes. Prefix codes. Reducing information redundancy. Block coding. The carrying capacity of the channel

Errors always occur when transmitting via communication channels. Their reasons may be different, but the result is the same - the data is distorted and can not be used on the receiving side for further processing. As a rule, the possibility of bit twisting in the transmitted data stream at the level of the physical channel is in the range of 10^{2} ...10^{-6} . At the same time, users and many application processes often demand the possibility of errors in the received data not worse than 10^{-6}...10^{-12} . Errors are dealt with at various levels of the seven-level OSI model (mostly in the first four). There are many different ways to deal with errors.

In one method, on the transmitting side, the transmitted data is encoded by one of the known error correction codes. On the receiving side, respectively, is decoding the received information and correcting the detected errors. The ability of the applied error-correcting code depends on the number of redundant bits generated by the encoder. If the redundancy is small, ie there is a risk that the received data will contain undetected errors, this can lead to errors in the application process. If you use code with high corrective power, it leads to low data rates. Thus, knowledge of the theory of noise-tolerant coding allows to determine the optimal parameters of noise-tolerant code depending on the task.

Noise-tolerant codes are one of the most effective means of ensuring high fidelity both when storing and transmitting discrete information. A special theory of coding noise immunity has been created, which has been developing rapidly recently.

K. Shannon formulated a theorem for the case of transmission of discrete information from a channel with interference, which states that the probability of erroneous decoding of received signals can be provided as arbitrarily small by choosing the appropriate method of encoding signals.

Noise-tolerant codes are codes that allow you to detect or detect and correct errors that occur as a result of interference.

Coding noise immunity is ensured by introducing redundancy in the code combinations, ie due to the fact that not all symbols in the code combinations are used to transmit information.

All noise-tolerant codes can be divided into two main classes: block and continuous (recurrent or chain).

In block codes, each message (or message element) corresponds to a code combination (block) of a certain number of signals. The blocks are encoded and decoded separately. Block codes can be uniform when the length of code combinations *n is* constant, or non-uniform when n is variable.

Uneven noise-tolerant codes have not received practical application due to the complexity of their technical implementation.

Both block and continuous codes, depending on the methods of redundancy, are divided into separate and inseparable. The role of individual characters is clearly delineated in separate codes. Some symbols are informative, others are verifiable and are used to detect and correct errors. Separate block codes are usually called n-codes, where n is the length of the code combinations, *k* is the number of information symbols in the combinations. Inseparable codes do not have a clear division of the code combination into informational and valid characters. This class of codes is still small. Separate block codes are divided, in turn, into non-systematic and systematic.

Most known separate codes are systematic codes. In these codes, the test symbols are determined as a result of linear operations on certain information symbols. For the case of binary codes, each test character is selected so that its sum modulo two with certain information symbols becomes zero. Decoding is reduced to checking for parity of certain groups of characters. As a result of such checks, information is given on the presence of errors, and if necessary - on the position of the characters where there are errors.

Basic principles of noise-tolerant coding

To clarify the idea of noise-tolerant coding, consider the binary code that has found the widest application in practice. Recall that a binary code is a code based on $m = 2$. The number of bits n in the code combination is called the length or significance of the code. The symbols of each digit can take the values 0 and 1. The number of units in the code combination is called the weight of the code combination and denote *w*.

The degree of difference between any two code combinations of this code is characterized by the so-called distance between codes *d*. It is expressed by the number of positions or symbols in which the combinations differ from each other, and is defined as the weight of the sum modulo two of these code combinations.

Errors, due to the influence of interference, are manifested in the fact that in one or more bits of the code combination, zeros turn into ones and, conversely, ones turn into zeros. The result is a new – erroneous code combination.

If errors occur in only one digit of the code combination, they are called one-time. If there are errors in two, three and so on. Errors called double digits, three and so on.

Experimental studies of communication channels have shown that the errors of the symbols when transmitted over the communication channel, as a rule, are grouped into packets of different durations. A bundle of errors means a section of a sequence that begins and ends with erroneously accepted characters. Inside the pack can be properly accepted elements.

An error vector e is used to indicate the places in the code combination where the characters are distorted. The n-bit code error vector is an n-bit combination in which the units indicate the positions of the distorted characters of the code combination.

The weight of the error vector *we* characterizes the multiplicity of the error. The sum modulo two for the distorted code combination and the error vectors give the original undistorted combination.

As already noted, the noise immunity of the coding is ensured by introducing redundancy in the code combinations. This means that of the n characters of the code combination, $k < n$ characters are used to transmit information . Therefore, of the total number $N_o = 2^n$ possible code combinations, only $N = 2^k$ combinations are used to transmit information . Accordingly, the whole set $N_o = 2^n$ possible code combinations is divided into two groups. The first group includes the set $N = 2^k$ allowed combinations, the second group contains the set $(N_o - N) = 2^n - 2^k$ forbidden combinations.

If on the receiving side it is established that the received combination belongs to the group of allowed, it is considered that the signal came without distortion. Otherwise, it is concluded that the accepted combination is distorted. However, this is true only for such obstacles, when the possibility of transition from one permitted combination to another is eliminated.

In the General case, each of the N allowed combinations can be transformed into any of the N_o possible combinations, ie there are $N > N_o$ possible transmission options, of which:

N options for error-free transmission;

$N(N-1)$ options for transition to other permitted combinations;

$N(No-N)$ options for transition to prohibited combinations.

Thus, not all distortions can be detected. The share of erroneous combinations that are detected is $\frac{N(N_0 - N)}{N \cdot N_0} = 1 - \frac{N}{N_0}$.

To use this code as a correction set, the forbidden code combinations are divided into *N* non-intersecting subsets. Each of the subsets corresponds to one of the allowed combinations. The error is corrected in *(N_o – N)* cases equal to the number of prohibited combinations. The share of erroneous combinations corrected from the total number of erroneous combinations detected is $\frac{N_0 - N}{N(N_0 - N)} = \frac{1}{N}$.

The method of division into subsets depends on what errors should be corrected by this code.

Suppose you need to build a code that detects all errors with a multiplicity of *t* and below.

To construct such a code means that from the set N_o it is possible to choose *N* allowed combinations so that any of them in the sum modulo two with any error vector with weight $W_l \leq t$ would not give as a result any other allowed combination. This requires that the smallest code distance satisfy the condition $d_{\min} \geq t + 1$.

In the general case, to eliminate errors of multiplicity σ the code distance must satisfy the condition $d_{\min} \geq 2\sigma + 1$.

Similarly, we can establish that to correct all errors of multiplicity not more than σ and simultaneously detect all errors of multiplicity not more than *t* and (at $t \geq \sigma$) the code distance must satisfy the condition $d_{\min} \geq t + \sigma + 1$.

Codes with the specified corrective ability

Increasing the corrective capacity of the code is achieved by storing n by reducing the set *N of* allowed combinations (or reducing the number *k* of information symbols). Of course, in practice, the codes are built in reverse order: first, the number of information symbols *k* is selected based on the volume of the source alphabet, and then provides the necessary corrective capacity of the code by adding redundant characters.

Let the volume of the alphabet of the source *N* be known. The required number of information symbols $k = \log_2 N$.

Let also know the total number of errors *E* that need to be corrected.

The task is to determine the significance of the code *n*, given the *N* and *E*, which has the necessary corrective capabilities.

The total number of false combinations to be corrected, equals $E>2^k = E>N$. Because the number of erroneous combinations is $N_o - N$, the code provides correction of no more than $N_o - N$ combinations. Therefore, the necessary condition for the possibility of correcting errors can be written in the form $NE \le N_0 - N$, and get $N_0 \ge (1+E)N$ or $N \le \frac{2^n}{1+E}$ (13.1).

Formula (13.1) expresses the condition for selecting the value of the code n. Consider some cases. If there are errors of different multiplicity, it is first necessary to ensure the elimination of single errors, the probability of which is greatest. The number of vectors of single errors is possible $E = C_n^1 = n$.

In this case, the dependence (13.1) will take the form $N \le \frac{2^n}{1+n}$ (13.2).

When building the code, it is advisable to use the table. 13.1. It should be borne in mind that the code must also satisfy the condition $d_{min} = 3$.

Table 13.1 - Identifying the significance of the correction code

n	2	3	4	5	6	7	8	9
$2^n/(1+n)$	1.88	2	3.2	5.33	9.2	16	28.4	51.2

If it is necessary to ensure the elimination of all errors of multiplicity from 1 to l, it is necessary to take into account that

the number of possible single errors $E_1 = C_n$,

the number of possible double errors $E_2 = C^2_n$,

the number of possible l- fold errors $E_l = C^l_n$.

Total number of errors $E = \sum_{i=1}^{l} C_n^i$.

In this case, the dependence (13.1) will take the form $N \le \frac{2^n}{1+\sum_{i=1}^{l} C_n^i}$ (13.33).

Condition (13.3) is a lower bound for the length koryhuyuchooho code, meaning it determines the required minimum code length n, which provides error correction at a certain given the multiplicity of the permitted combinations of N or including information symbols $k = \log_2 N$.

The same condition is the upper bound for N or k, ie determines the maximum possible number of allowed combinations or information symbols for the code of length n, which provides correction of errors of a given multiplicity.

Any correction code is characterized by a number of indicators: length n, number of information symbols k or redundant symbols $\rho = n - k$, the total number of all possible code combinations $N_o = m^n$ (for binary codes $N = 2^n$), the number of

allowed code combinations (code power) N, the weight of the code combination w, the weight of the error vector w_1, the code distance d, and others.

However, the main indicator of the quality of the correction code is its ability to ensure the correct acceptance of code combinations in the presence of distortions under the influence of interference, ie noise immunity of the code.

The corrective capability of the code is provided by redundancy, ie the lengthening of code combinations. At lengthening of code combinations the equipment becomes more complicated, time of transfer and processing of the information increases. Therefore, redundancy is also an important characteristic of the code.

To assess the redundancy of the code use the concept of redundancy factor $K_{red} = \frac{\rho}{n} = \frac{n-k}{n}$, where ρ – number of redundant characters in the code combination.

Example. Determine the corrective capacity of the code having the following allowed combinations: 00000; 01110; 10101; 11011.

Solution . The corrective capacity of the code is determined by the minimum code distance. Let's make a matrix of distances between code combinations (tab. 5.2).

Table 13.2 - Distance matrix

	00000	01110	10101	11011
00000	0	3	3	4
01110	3	0	4	3
10101	3	3	0	3
11011	4	3	4	0

As can be seen from the matrix, the minimum code distance $d_{min} = 3$. Therefore, this code is able to:

a) detect double errors;

b) eliminate one-time errors;

c) eliminate and detect one-time errors.

Systematic codes

Nowadays, the widest class of correction codes are systematic codes. These codes belong to the group of separate block codes. For a systematic code, the sum modulo for the two allowed combinations also gives the allowed combination.

Matrix code representation is widely used in coding theory. All allowed code combinations of the systematic (n, k) - code can be obtained by having k original allowed code combinations. The source code combinations must satisfy the following conditions:

1. The number of initial combinations should not include zero.

2. The code distance between any pairs of source combinations must be at least d_{min}.

3. Each initial combination, as well as any non-zero allowed combination, must contain the number of units not less than d_{min}.

4. All initial combinations must be linearly independent, ie, none of them can be obtained by summing the others.

The original combinations can be obtained from a matrix consisting of k rows and n columns:

$$P_{n,k} = \begin{vmatrix} a_{11}\ a_{12} \dots a_{1k}\ b_{11}\ b_{12} \dots b_{1\rho} \\ a_{21}\ a_{22} \dots a_{2k}\ b_{21}\ b_{22} \dots b_{2\rho} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{k1}\ a_{k2} \dots a_{kk}\ b_{k1}\ b_{k2} \dots b_{k\rho} \end{vmatrix}.$$

Here the characters of the first k columns are informative and the last ρ columns are checkable. The matrix $P_{n,k}$ is called generating. The generating matrix $P_{n,k}$ can be represented by two submatrices – information U_k and test H_ρ.

$$P_{n,k} = \left\| \begin{matrix} a_{11}\ a_{12} \dots a_{1k} \\ a_{21}\ a_{22} \dots a_{2k} \\ \dots\dots\dots\dots\dots \\ a_{k1}\ a_{k2} \dots a_{kk} \end{matrix} \right\| \left\| \begin{matrix} b_{11}\ b_{12} \dots b_{1\rho} \\ b_{21}\ b_{22} \dots b_{2\rho} \\ \dots\dots\dots\dots\dots \\ b_{k1}\ b_{k2} \dots b_{k\rho} \end{matrix} \right\|,$$

where

$$U_k = \begin{vmatrix} a_{11}\ a_{12} \dots a_{1k} \\ a_{21}\ a_{22} \dots a_{2k} \\ \dots\dots\dots\dots\dots \\ a_{k1}\ a_{k2} \dots a_{kk} \end{vmatrix},\ H_\rho = \begin{vmatrix} b_{11}\ b_{12} \dots b_{1\rho} \\ b_{21}\ b_{22} \dots b_{2\rho} \\ \dots\dots\dots\dots\dots \\ b_{k1}\ b_{k2} \dots b_{k\rho} \end{vmatrix}.$$

To construct a generating matrix, it is convenient to take the information matrix in the form of a square unit matrix

$$U_k = \begin{vmatrix} 1\,0\,0 \dots 0 \\ 0\,1\,0 \dots 0 \\ 0\,0\,1 \dots 0 \\ \dots\dots\dots\dots \\ 0\,0\,0 \dots 1 \end{vmatrix}.$$

The test submatrix H_ρ must be constructed in compliance with the following conditions:

a) the number of units in a line must be at least d_{min-1} ;

b) the sum modulo two of any two lines must contain at least d_{min-2} units.

Test symbols are formed by linear operations on information symbols. For each code combination, ρ independent sums modulo two must be added . The choice of information symbols involved in the formation of a test symbol depends on the method of decoding the code. It is very convenient to make checksums by means of the checking matrix H constructed in such way. First, a submatrix H^l is constructed , which is transposed with respect to the submatrix H_ρ:

$$H^1 = \begin{vmatrix} b_{11}\, b_{21} \ldots b_{k1} \\ b_{12}\, b_{22} \ldots b_{k2} \\ \ldots\ldots\ldots\ldots\ldots \\ b_{1\rho}\, b_{2\rho} \ldots b_{k\rho} \end{vmatrix}.$$

Then a single matrix isassigned to it on the right

$$H = \begin{vmatrix} b_{11}\, b_{21} \ldots b_{k1}\, 1\, 0 \ldots 0 \\ b_{12}\, b_{22} \ldots b_{k2}\, 0\, 1 \ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ b_{1\rho}\, b_{2\rho} \ldots b_{k\rho}\, 0\, 0 \ldots 1 \end{vmatrix} \quad (13.4).$$

The algorithm for determining the test symbols on the information using the matrix (13.4) is as follows. The positions occupied by the units in the first line of the submatrix H^l determine the information bits that should participate in the formation of the first check digit of the code combination. The positions of the units in the second line submatrices H^1 determined information bits involved in the formation of the second category and so checkout etc.

The test matrix *H is* very convenient for determining the location of the error in the code combination, and, consequently, error correction. The verification of code combinations is performed by summing modulo two test symbols of code combinations and test symbols calculated according to the received information. The result will be a set of control equations, each of which represents the sum modulo two of the control symbols and a certain amount of information.

Warehouse control equations easily determined checkout matrix *H*. The structure of the first reference equality should include characters whose positions are occupied units in the first row of the matrix *H* . The structure of the second control equality should include characters whose positions are occupied units in the second row of the matrix *H*, etc.

As a result of ρ such checks, a ρ-bit binary number (syndrome) will be obtained , which will be equal to zero in the absence of errors and non-zero in the case of errors.

If to correct single errors in code combinations to obtain syndromes is quite simple, then to correct double, triple, etc. errors, as well as to correct bundles of errors, the construction of syndromes is quite difficult and in these cases usually seek help from a computer.

Example. Construct a generating matrix of systematic code capable of correcting a single error ($\sigma = 1$) when transmitting 16 messages.

Solution . Because the number of allowed code combinations $N = 16$, the number of information bits in the code combinations
$k = \log_2 16 = 4$.

Therefore, the number of rows of the generating matrix $P_{n,\,k}$ should be equal to four.

Using condition (13.2) and table. 13.1, we find the code length $n = 7$. Therefore, the number of columns of the matrix $P_{n,k}$ is equal to seven.

Since the number of test bits $p = n - k = 3$, the number of columns of the test submatrix H_ρ is equal to three. The number of units in each row of the submatrix H_ρ must be at least $d_{min\,-\,1}$. For $\sigma = 1$ $d_{min} = 2\sigma + 1 = 3$. Choose following combinations for the submatrix H_ρ : 111, 110, 101, 011. Therefore, the test submatrix can look like:

$$H_\rho = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} \text{ or } H_\rho = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} \text{ or } H_\rho = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}.$$

The information submatrix U_k – taken as a single matrix with the number of rows and columns equal to $k = 4$:

$$U_k = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

The final type of the generating matrix:

$$P_{7,4}=\begin{vmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&1&0\\0&0&1&0&1&0&1\\0&0&0&1&0&1&1\end{vmatrix} \quad \text{or} \quad P_{7,4}=\begin{vmatrix}1&0&0&0&1&1&1\\0&1&0&0&0&1&1\\0&0&1&0&1&1&0\\0&0&0&1&1&0&1\end{vmatrix} \quad \text{or}$$

$$P_{7,4}=\begin{vmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{vmatrix}.$$

Example. Construct a test matrix H *of* systematic code (7,4), the forming matrix of which has the form

$$P_{7,4}=\begin{vmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&1&0\\0&0&1&0&1&0&1\\0&0&0&1&0&1&1\end{vmatrix}.$$

Write an equation to determine the test symbols.

Solution . Find the submatrix H^l which is transposed with respect to the submatrix H_ρ:

$$H^1=\begin{vmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\end{vmatrix}.$$

Assigning to it the unit matrix U_k on the right, we obtain a test matrix

$$H=\begin{vmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\1&0&1&1&0&0&1\end{vmatrix}.$$

Test symbols are determined by summing modulo two specific information symbols. The algorithm for determining the test symbols is determined by the submatrix $H^{\prime}$. The positions occupied by the units in the first line $H^{\prime}$ determine the information digits that should participate in the formation of the first check digit. The positions of the units in the second line determine the information digits involved in the formation of the second check digit. Finally, the positions of the units in the third line determine the information digits involved in the formation of the third check digit. Therefore, the test symbols are defined by equalities $b_1=a_1+a_2+a_3$, $b_2=a_1+a_2+a_4$, $b_3=a_1+a_3+a_4$.

Example. Construct a combination of systematic code (7, 4) for the case when a simple excessive combination has the form 0110.

Solution . In this case, the values of the information symbols are as follows:

$a_1 = 0$; $a_2 = 1$; $a_3 = 1$; $a_4 = 0$.

Then, using the equations obtained in the previous example for the test symbols, determine:

$b_1 = 0+1+1 = 0$, $b_2 = 0+1+0 = 1$, $b_3 = 0+1+0 = 1$.

Therefore, the message 0110, encoded by the systematic code (7, 4), will take the form: 0110011.

Example. The code matrix (7, 4) has the form

$$P_{11,7} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{vmatrix}.$$

Construct a combination of systematic code (11, 7), which corresponds to the excessive combination 1101001.

Solution . Since to obtain a combination of the form 1101001 it is necessary to add the first, second, fourth and seventh rows of the unit matrix U_k, to construct test bits should make the same rows of the submatrix H_p:

$$\begin{array}{r} 1111 \\ +\,1110 \\ 1011 \\ \hline 0110 \\ \hline 1100 \end{array}.$$

Thus, the code combination (11, 7) has the form 11010011100.

The most common error detection codes include parity code, direct and inverse iteration codes, and correlation code. Consider them in more detail.

The code contains only one redundant character. The redundant character is selected so that the total number of units in the code combination is even. Checking the code combination is done by summing the module two of all its characters. The redundancy of the code is equal to $K_{red} = \frac{\rho}{n} = \frac{1}{n}$, where n is the length of the code combination; $\rho = 1$ – the number of valid characters.

The code allows you to detect single errors and all errors of odd multiplicity, because only in these cases, the number of units in the combination will be odd. No errors of pair multiplicity are detected.

Parity code encoder

The parity code encoder is an electronic device that automatically adds one control symbol "0" or "1" to the information symbols of the binary code information inputs, the value of which depends on the number of units in the binary code combination. If the number of units in the binary code is even, then the control symbol is "0", if odd - "1".

Thus, in all code combinations of code with parity check, the number of units must always be even. It is possible to detect a distorted combination due to interference on the basis of parity analysis of the number of units in the combination. If the parity is violated, the code combination is not accepted.

The parity scheme of the parity code encoder depends on how the normal binary code is entered - in parallel or in series. Fig. 13.1 shows a diagram of the encoder in parallel input of the binary code in the register.

When a signal with a logic level "1" is applied to the input L of the register, the binary combination, which is fed to the information inputs D of the register DD1, is written to the outputs Q1-Q4. The element that produces the fifth control symbol is the adder modulo 2 m2 - DD2.

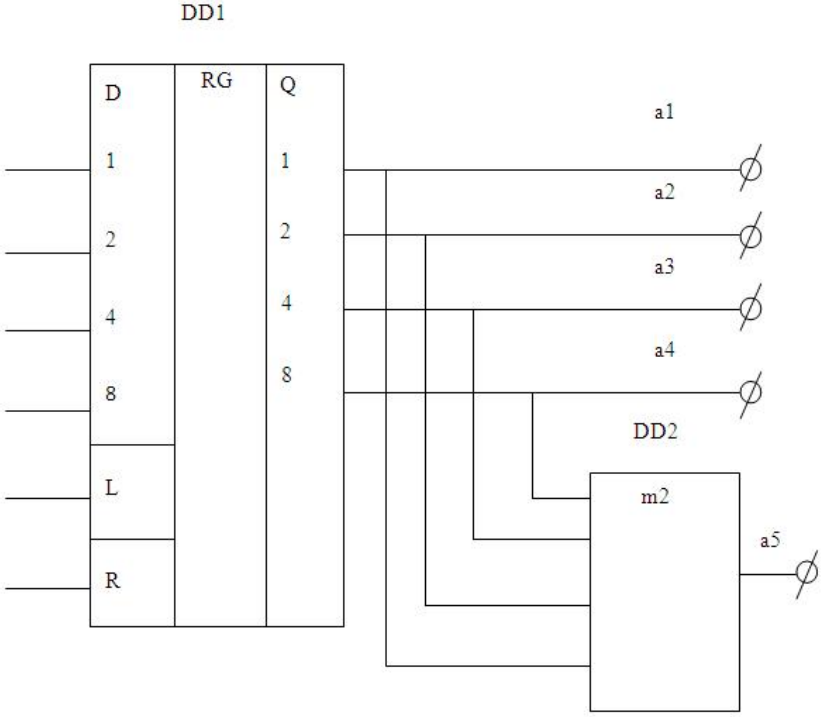


Fig. 13.1

If the number of units at its input is even, then zero will appear at its output, if odd - one. Therefore, in some literature sources, the adder modulo two is called *the parity generator* .

Thus, the input four-bit combination at the input of the encoder automatically turned into a five-bit. The fifth symbol - the control, which in the receiving device allows you to determine whether there is an error in the received combination.

If the information combination of the usual binary code is entered into the encoder sequentially, the scheme becomes complicated (Fig. 13.2).

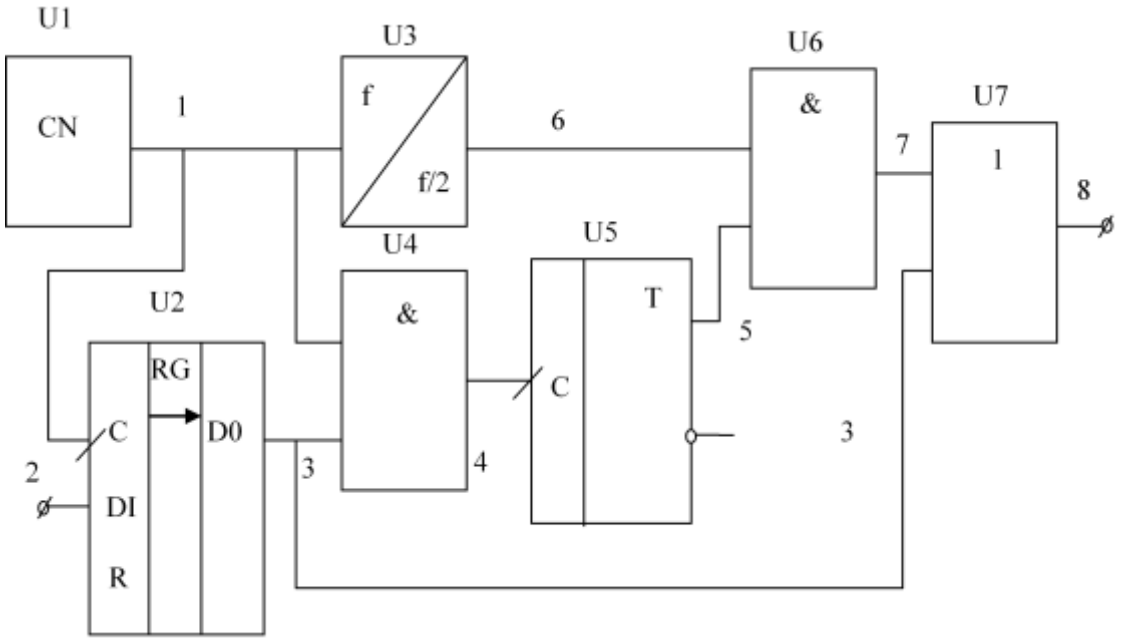
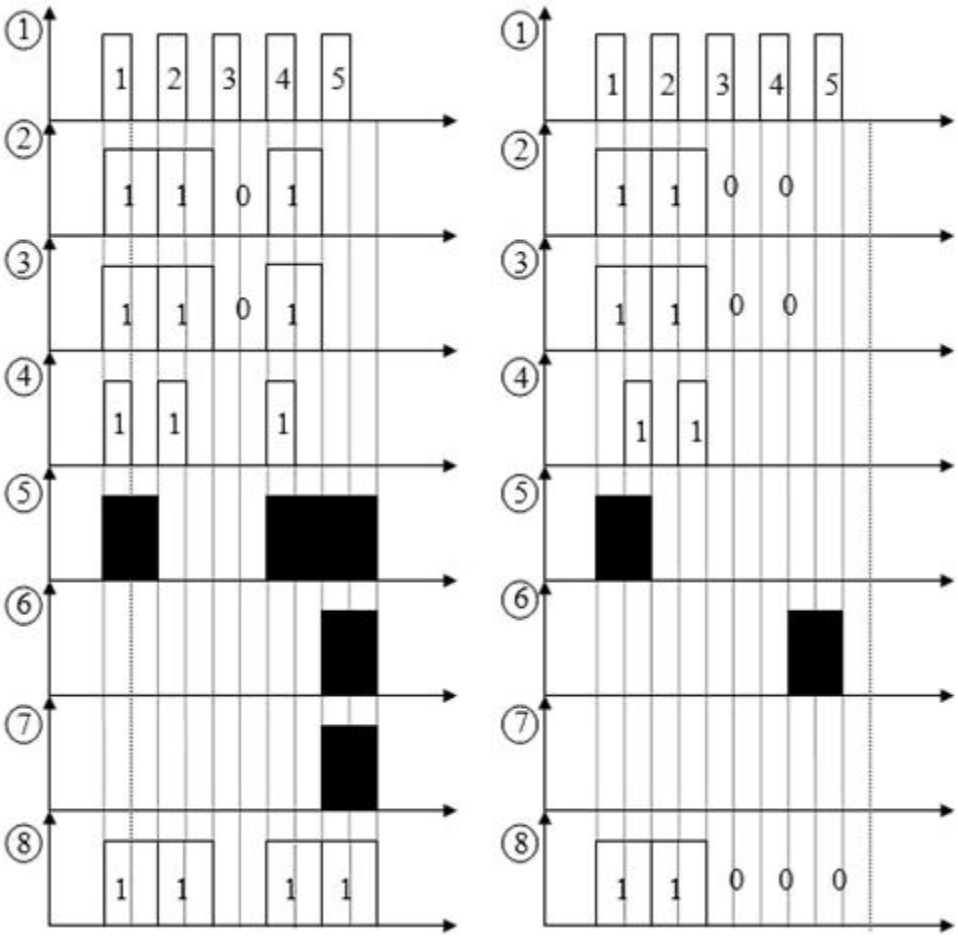


Fig. 13.2

To explain the operation of the circuit, we present time diagrams of its operation on the example of four-bit binary codes 1101 and 1001 (Fig. 13.3).

The input information binary combination 1101 is fed to the serial input of the serial register U2 and is output in serial code through the element U7 to external devices in accordance with the clock pulses of the generator U1 (Fig. 13.3, a). The element that determines the parity of the units of the input combination is the counting trigger U5. With an even number of pulses coming to its counting input, the level of logic zero will be set at the direct output of the trigger, with an odd number - the level of logic unit. If the input combination 1101 is in effect, it will pass to external devices during the first four cycles. The fifth pulse from the generator is extracted from the sequence of pulses by the circuit U3, which is a frequency divider by 5. This pulse removes information about the state of the direct input of the trigger U5, is transmitted through the element U7 and added to the main code combination.



a) b)

Fig. 13.3

In essence, the counting trigger performs the same function as the adder modulo two in the scheme of Figs. 13.2.

Element U4 separates the information units of the input code by a pause if they follow each other.

Parity code decoder

The parity code decoder is an electronic device that analyzes the received parity combination and, if it is violated, prohibits its further conversion. If the received combination is written in a series-parallel register, the decoder circuit has the following form (Fig. 13.4).

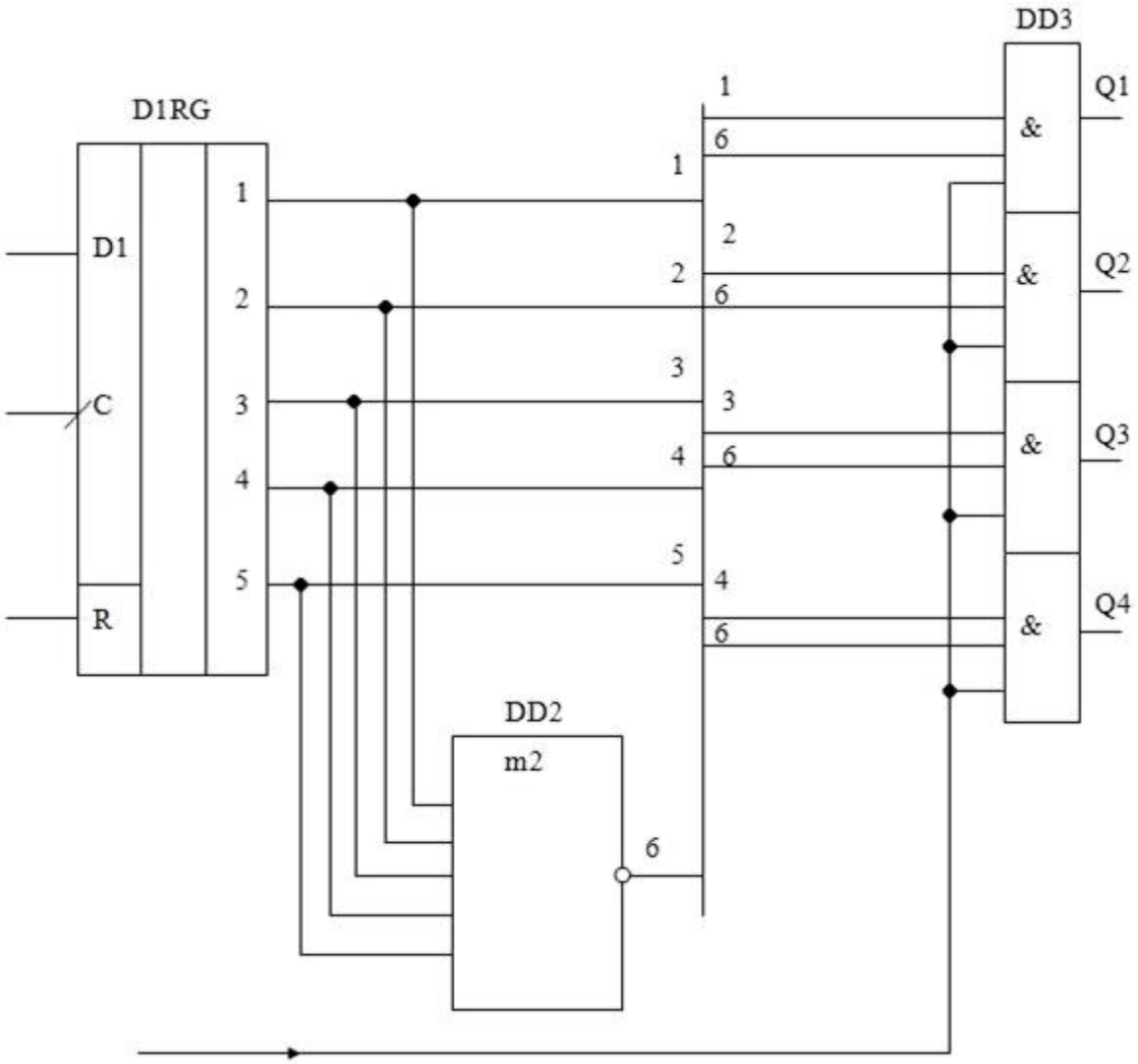


Fig. 13.4

Accepted in the serial-parallel register combination of code with parity check, coming to the adder modulo 2 DD2, is analyzed, and if the number of units is even, the inverse output of the adder will be the level of the logical unit, which goes to all elements of the circuit DD3 .

In the presence of a separate pulse, which is applied simultaneously to all elements of the circuit DD3, the received combination appears at the outputs Q1 - Q4 and passes on. If the parity is violated, the output of the inverse adder modulo two m2 will appear a level of logical zero, which will close all the elements DD3 and the accepted combination will not pass.

There are a number of codes in which to increase the noise immunity to the information code combination, which consists of n_0 information symbols of the binary redundant code, k control symbols are added, and $k = n_0$. Thus, the total

number of elements in the code combination is equal to $2n_0$. Therefore, such codes are called *codes with doubling the number of elements* .

Depending on the method of formation of control symbols, the following codes are classified into:

a) codes with direct doubling;

b) codes with inverse doubling;

c) correlation codes.

Consider in more detail the schemes that implement the above codes.

Code encoder with direct repetition

A direct repetition code encoder is a device that automatically adds the same (repeats) binary code to an information code combination, doubling the total number of binary characters.

In fig. 13.5 shows a diagram of such an encoder. Assume that the number of information characters of the binary code $n_0 = 4$.

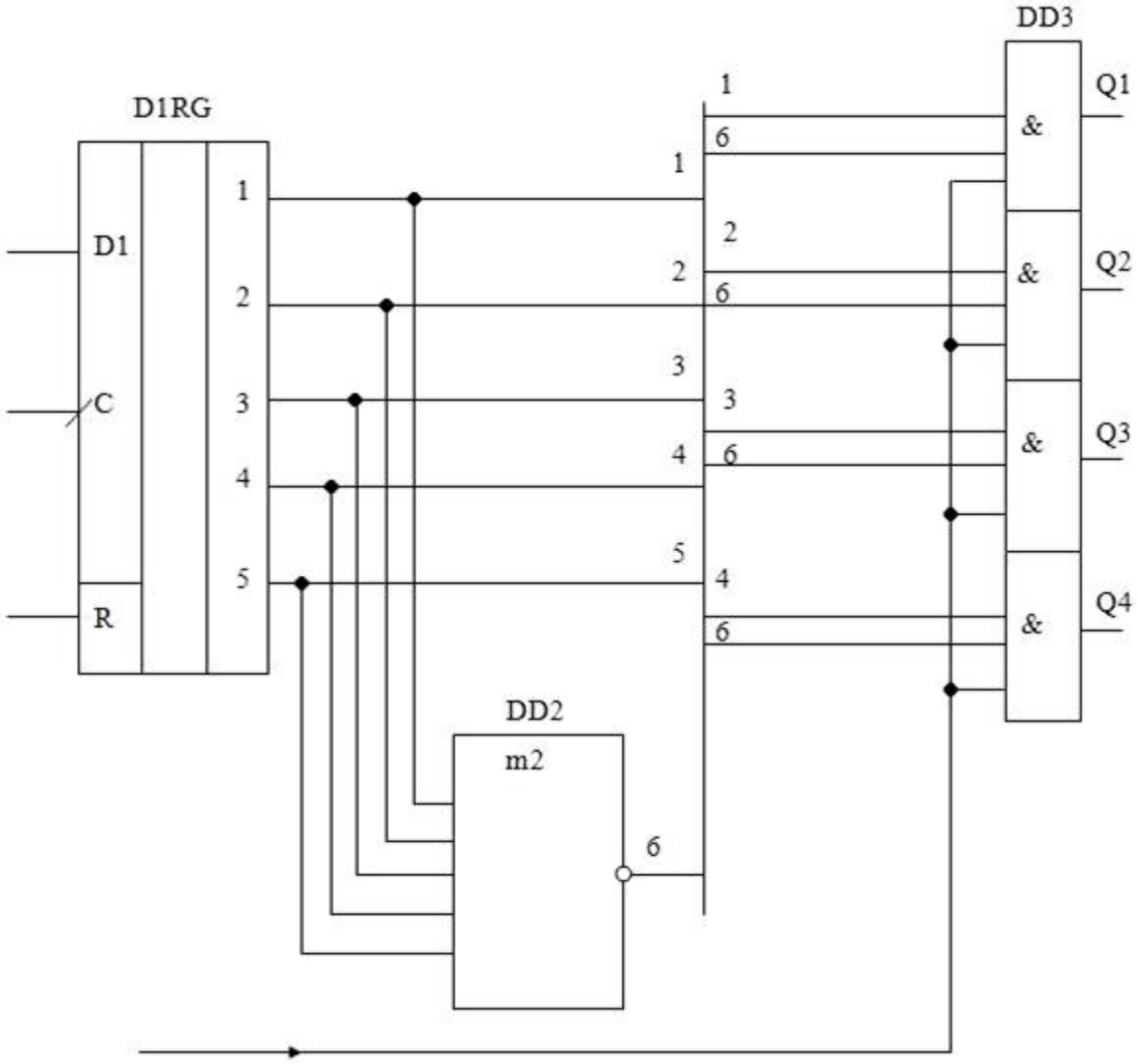


Fig. 13.5

The input code is transmitted to the inputs D1-D4 of the multiplexer U3. The fifth - eighth inputs of the same multiplexer are connected in parallel with the first - fourth of its inputs. When the pulses from the generator U1 to the input of the binary counter U2 at its outputs is formed binary (address code), which controls the conversion of parallel code at the inputs of the multiplexer in series at its output. Thus, if the inputs of the multiplexer is given the code 1101, then 8 cycles of the generator for further conversion by external devices will receive the code 11011101.

Code decoder with direct repetition

The code decoder with direct repetition (Fig. 13.6) compares the information and control symbols of the combination received by the receiver (they must repeat each other) and gives a command for its further conversion, if the effect of interference is not detected.

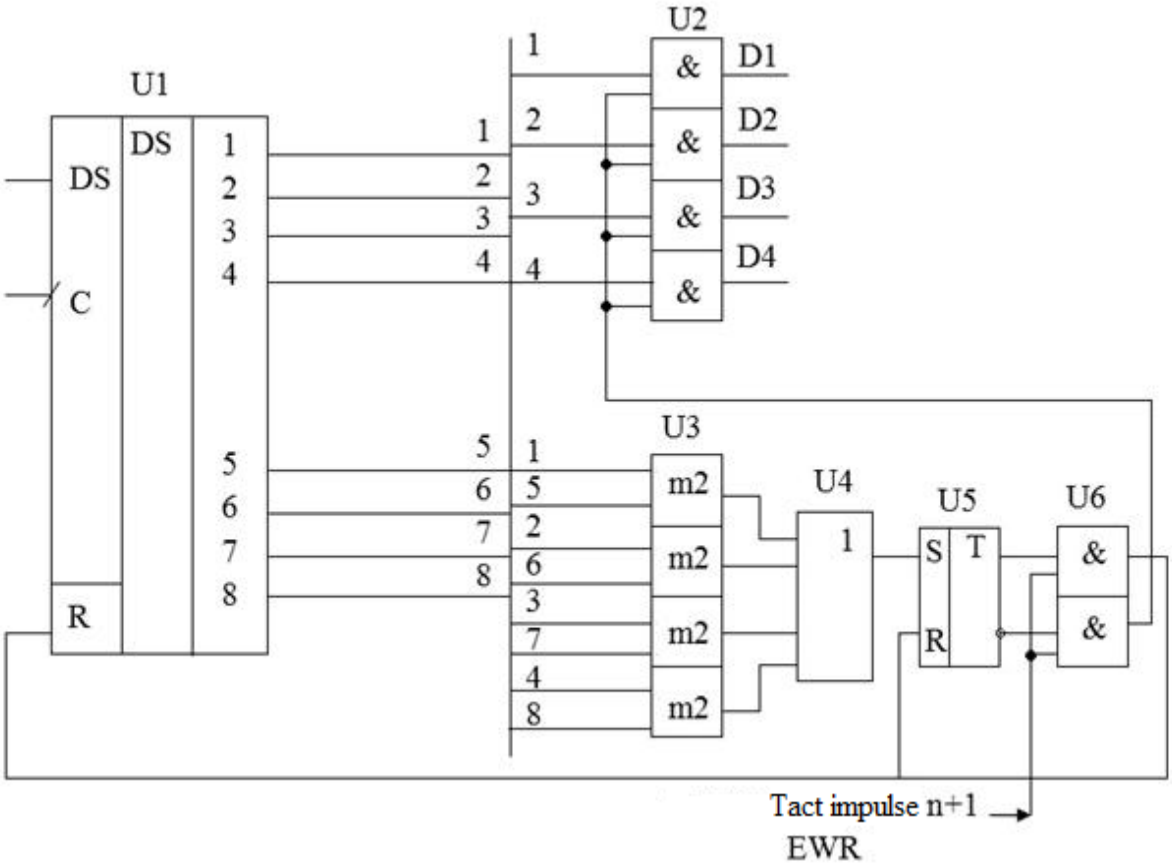


Fig. 13.6

After writing to the serial-parallel register U1, the code combination with direct repetition will appear at outputs 1-8 of the register. Combinations at outputs 1-4 and 5-8 repeat each other. The corresponding symbols of the combinations n_0 (at outputs 1-4 of the register) are compared with the symbols $k = n_0$ (at outputs 5-8) using the block of adders modulo 2 - U3.

If the combination of the code with direct repetition is accepted correctly on outputs vcix of adders on the module 2 there will be a level of logical zero. At the output of the circuit "OR" U4 will also be the level of logic "0" and the trigger U5 is in the initial state (the direct output of the trigger has the potential of logic "0". input that allows reading information EWR. The momentum of this entrance there vciyeyi after writing down the case. If the code is a combination of *n* bits, the divided pulse appears on the *n + 1* cycle (in our example the ninth cycle).

At the moment of receipt of this pulse, the logic elements of the block U6 are opened, at the input of the lower element of this block, the level of the logic unit appears, which opens all the elements "I" of the block U2. The output combination D1-D4 from the outputs of register 1-4 through the elements "I" of the block U2 is fed for further conversion.

If the combination is distorted due to interference, the level of the logical unit will appear at one (or several) outputs of the adders of the block U3, which will bring the trigger U5 to a single state. The 9th clock pulse at the EWR input opens the upper logic element of the block U6, at its output the level of the logic unit will appear, which will enter the inputs R of the register Ul and the trigger U5 and reset them to the initial state (combination erased). The elements of the U2 unit will be closed and the distorted combination will not appear at its outputs.

Note that the output unit of the decoder U2 contains only four information bits. Control discharges performed their role in the analysis of the correctness of the accepted combination and are not used in the future.

Codes with inverse repetition of the information binary code combination are formed according to the following rule: if the number of units in the information combination is even, then the same combination is automatically added (direct repetition) and if the number of units in the information combination is odd, the inverted information combination is added. For example, if the information code combination has the form 1100, then it is added to the same combination 1100 and in general for further conversion will receive a combination 11001100.

If the information combination has the form 1110, then the combination 0001 will be added to it and the combination of the inverse code will have the form 11100001.

Code encoder with inverse repetition

An inverse iteration code encoder is an electronic device that automatically adds a control combination of $k = n_0$ characters to the information n_0-bit combination of binary code according to the following algorithm: if the number of information code units is even, the same combination is added if odd, an inverted information combination is added to it. The scheme of such an encoder is shown in Fig. 13.97.

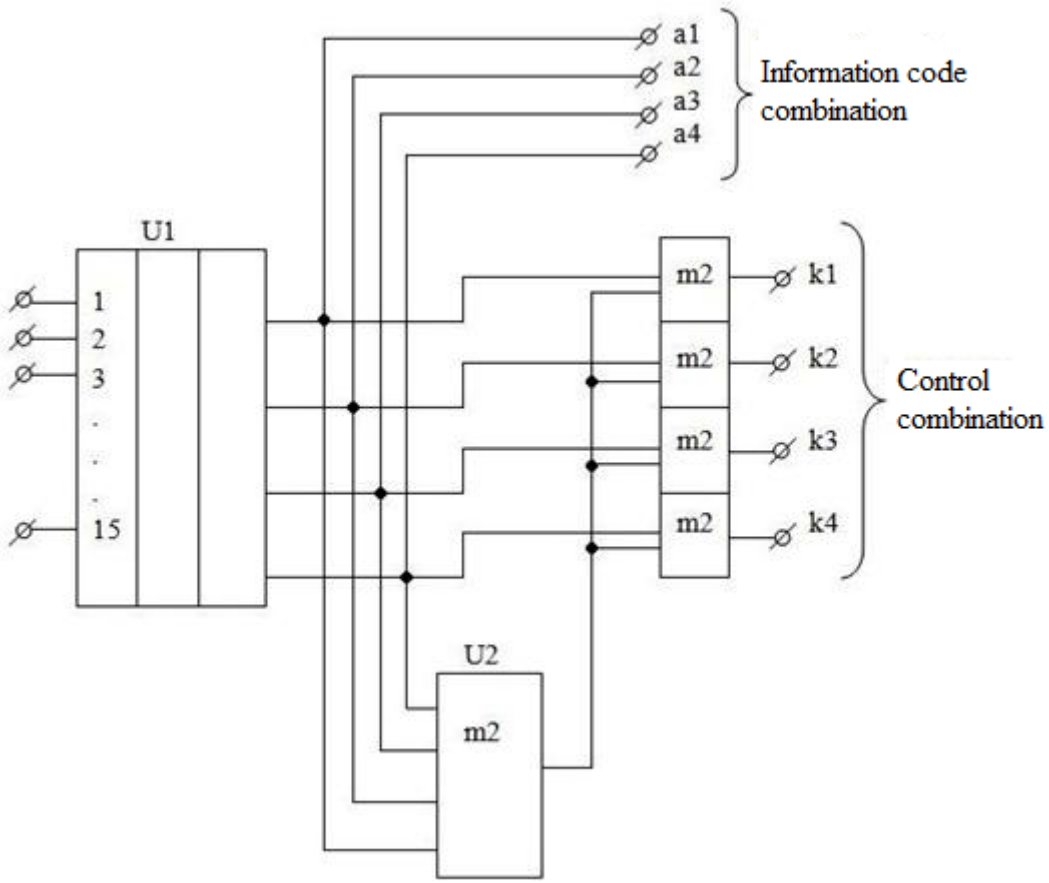


Fig. 13.7

The information code combination appears at the outputs of the encoder U1 when applied to one of its 16 inputs level logical "1". The adder modulo 2 U2 analyzes the parity of the number of units in the information code combination.

If this number is even, the level of logical zero will appear at its output and all the symbols set at the outputs of encoder U1 will pass to the outputs of the adder unit according to module 2 U3. If the number of units of the information combination is odd, the level of the logical unit will appear at the output of the element U2 and the outputs of the block U3 will pass the inverted symbols of the information code combination set at the outputs of encoder U1, ie the circuit works according to the established algorithm.

Code decoder with inverse repetition

An inverse iteration code decoder is a device that, according to a known generation algorithm, compares the information and control symbols received by the receiving device and gives a command for its further conversion if no interference is detected.

The scheme of the decoder of the inverse code is given in Fig.13.8.

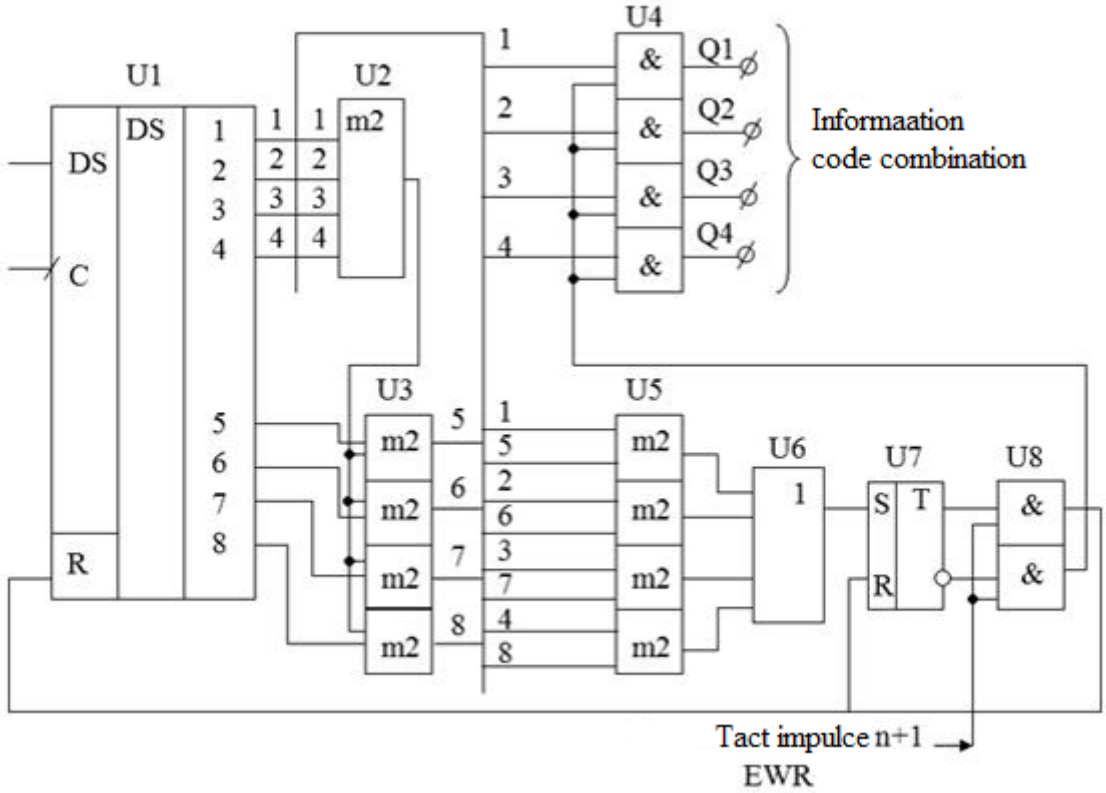


Fig. 13.8

The decoder works this way. The received combination is fed to the input DS of the series-parallel register U1 and after entering the input "C" *n* clocks are fixed at the outputs 1-8 of the register. (At outputs 1-4 - information part of the code combination, at outputs 5-8 - control combination).

The adder by module 2 U2 analyzes the number of units in the information part of the code combination and gives a command to the inverting unit U3, depending on the parity of the number of units in the information combination, the symbols of the control combination at outputs 5-8.

As a result, when the inverse code combination is received correctly, the information combination from the outputs of register 1-4 and the combination from the outputs of the block U3 must be the same. The adder unit modulo 2 U5 compares the corresponding symbols of these combinations and when received correctly at the outputs must be equal to the logical "0". In the presence of an error in the received

combination on one (or several) outputs of the block U5 logic "1" will appear, the trigger U7 will pass to unit state and after receipt on input EWR *n+1 of a* clock pulse through the top logic element of the block U8 resets (erases) data that was recorded in the register Ul; i returns the trigger U7 to its original state. When properly received after receiving the input EWR *n+1* pulse (in our example - the 9th), the signal from the lower element "I" of the block U8 allows the passage of the information part of the code combination of the inverse code through the elements "I" of the block U4 for further conversion .

The encoder of the correlation code is an electronic device that when it receives n_0 information symbols of binary code automatically adds to each information symbol one control "0" or "1" and the value of which is formed by the rule: information zero is converted to 01, information unit turns into 10. Thus the total number of characters of the code combination of the correlation code $n = n_0 + k = 2n_0$. Therefore, this code is attributed to the code with doubling the number of elements. The scheme of the encoder of the correlation code is shown in Fig. 13.9, and time diagrams of his work in Fig. 13.10.

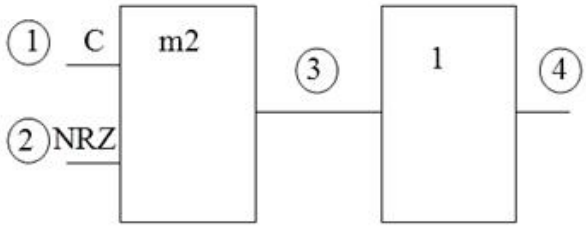


Fig. 13.9

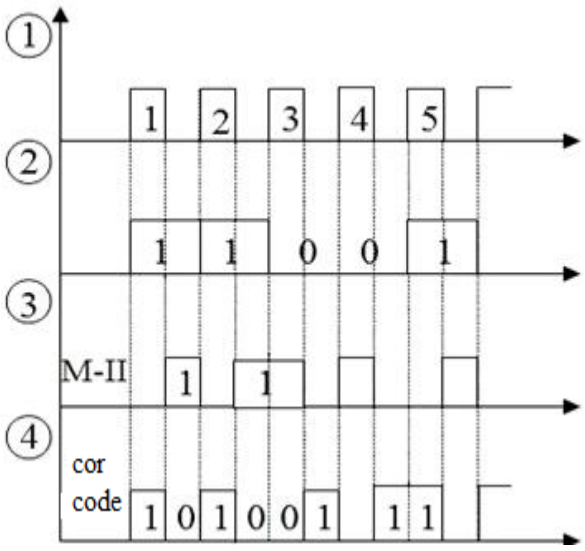


Fig. 13.10

Correlation code decoder

The correlation code decoder is a device that analyzes the received code combination. At the reception, the error is determined in the case when the paired elements contain the same characters, ie 11 or 00 (instead of 10 and 01). When received correctly, the paired elements are discarded and the initial combination remains.

The scheme of the correlation code decoder is shown in Fig. 13.11.

To detect the error, the received code combination is divided into groups of 2 characters in each and the parity of each group is analyzed.

If received correctly, each group must have different characters (01 or 10). If the same characters are accepted in any group (00 or 11) - the combination is considered distorted by noise and is erased, because in this case the error is not detected.

The specified error detection algorithm is implemented as follows.

During N, a combination of the received correlation code is written to the serial-parallel register U1. Each pair of symbols (1-2, 3-4, ..., [(N-1) - N]) is compared using adders modulo 2 (block U2) and in the case of correct reception at all outputs of adders will be logical " 1 ", which are fed to the inputs of the circuit" I "U3. 3 receipt (N +1) of the clock pulse from the control circuit at the output of the element U3 will be a pulse that will allow further conversion of the information part of the combination (symbols Ul, U3, U5, ... U_{N-1}).

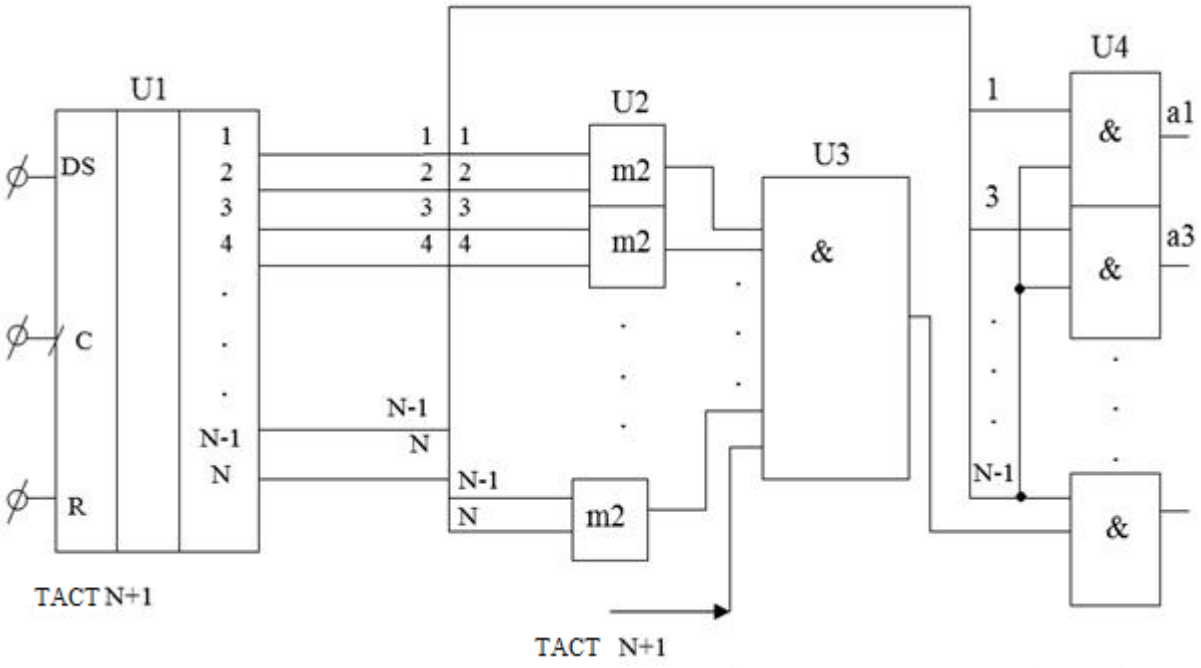


Fig. 13.11

In the case of an error in any pair of symbols at the output of the element U3, the value of the logical "0" does not change with the arrival of the (N + l) -th clock and the combination will not pass through the block of elements U4.

3 arrival (*N+2*) clock recorded in the register U1 combination of the correlation code is reset.

In fig. 13.12 shows another scheme of the decoder of the correlation code.

In this scheme, each pair of correlation code symbols is compared with one adder modulo 2 directly during the sequential receipt of code symbols at the input of the serial-parallel register U1, which has only two outputs. Element U3-adder modulo 2 with inversion compares each pair of characters every 2 cycles.

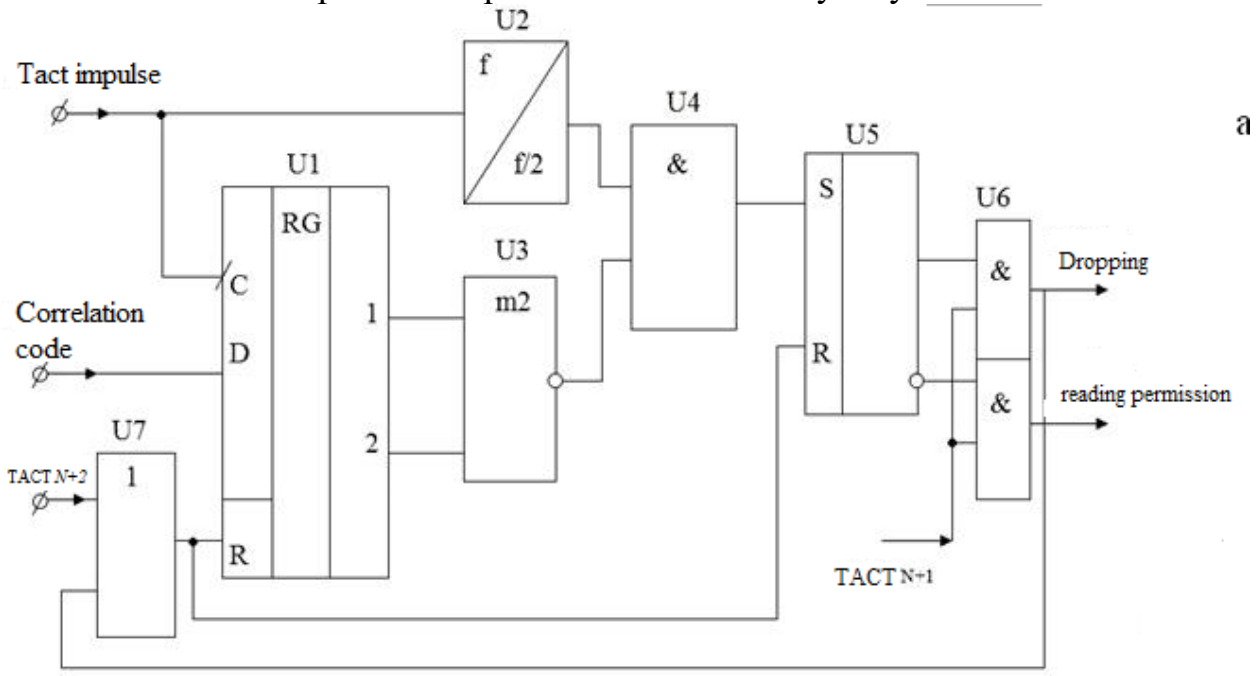


Fig. 13.12

In order for the result of comparison of each pair of symbols to be carried out only on even clocks of input pulses, permission to transmit the result of comparison is given through the frequency divider of clock pulses U2 with a counting factor of 2. In the case element U3 will appear level logical "1", which will be written to the RS-trigger U5. After receiving the code combination of the correlation code on (N+1) clock, the circuit elements U6 give permission to read the information in the case of correct reception or to reset the information in the presence of an error. 3 receipts

The most common codes with error detection and correction include the Heming code, cyclic and recurrent codes. Consider them in more detail.

Heming's code refers to systematic codes in which of the n symbols that form a combination, n_0 symbols are informational, and the last $k = n - n_0$ with redundant (control), intended for verification (control symbols in all combinations occupy the same positions) . Heming's codes allow to correct all single errors (at code distance d = 3) and to define all double errors (at $d = 4$), but not to correct them.

The relationship between the number of information and control symbols in the Heming code is found on the basis of such considerations. When transmitting a combination on a noise channel, any of the n characters of the code may be distorted, or the combination may be transmitted without distortion. Thus, there may be $n+1$ variants of distortion (including transmission without distortion). Using control symbols, it is necessary to check all $n+1$ variants. You can use the control symbols k to describe $2k$ events. To do this, the condition must be used:

$$2^k \geq n+1 = n_0 + k + 1.$$

Table 5.3 shows the relationship between k and n_0, which is obtained from this inaccuracy, where k is the number of control symbols in the Heming code, n_0 - information symbols.

Table 5.3 - Placement of control symbols in combinations of the Heming code

n_0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
k	2	3	3	3	4	4	4	4	4	4	4	5	5	5	5

In the Heming code, the control symbols are placed in places multiples of the power of 2, ie in positions 1, 2, 4, 8, etc. Information symbols are placed in the remaining places. For example, for a seven-element coded combination, you can write

k_1	k_2	a_{04}	k_3	a_{03}	a_{02}	a_{01}
$\boxed{a_1}$	$\boxed{a_2}$	a_3	$\boxed{a_4}$	a_5	a_6	a_7

The symbols of the Heming code, which are surrounded by rectangles, are *control* , the last - *information* , where a_3 - the highest (fourth) bit of the source code combination of binary code to be encoded, a_7 - the lowest (first) bit. After placing the code combination of control and information symbols in the appropriate places in the Heming code, special test equations are compiled, which are used to determine the presence of distortions and their corrections. From the verification equations i get control symbols when encoding the original code combination of

binary code. The following algorithm must be used to determine the control symbols.

1. All characters of the Heming code with bit numbers are arranged in ascending order of numbers and below them write the bit numbers in binary code

$a_1\, a_2\, a_3\, a_4\, a_5\, a_6\, a_7$

0001 0010 0011 0100 0101 0110 0111

2. The first equation is at checkout amount for vcix mod 2 bits, the rooms are in LSB 2^0 is a unit: $S_1 = a_1 \oplus a_2 \oplus a_5 \oplus a_7$.

The second equation is at checkout amount for the mod 2 vcix discharges in the rooms which is a unit of the second place corresponding binary equivalent (2^1): $S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7$.

The third equation checkout is at 2 mod amount for all discharges room unit which is in third place (2^2): $S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7$.

Similarly, other checksums are formed (with a larger number of information and control symbols, respectively).

As can be seen from the above equations, each checksum includes only one indeterminate control symbol k_i (a_1, a_2, a_4, respectively), and all other information symbols are known.

All the test equations under the Heming condition must be equal to 0 when summed by mod 2. 3 of this condition i find the control symbols.

Heming code decoding algorithm

On the receiving device, the combination of the Heming code is decoded - determines the presence of distortion of the received code combination and, if one symbol of the combination is distorted, it is automatically first determined and then corrected. Determination and correction of the distorted symbol is carried out on the basis of verification equations. When taken correctly, all amounts must be zero. In the case of distortion, the binary number that is the result of these sums (error syndrome) is converted to a decimal number, which indicates the number of the distorted character, which is then corrected by inversion.

For example, when receiving the encoded above undistorted Heming code combination:

a_1	a_2	a_3	a_4	a_5	a_6	a_7
0	1	1	1	1	0	0,

the values of vcix symbols of the combination are substituted in the appropriate places in the checksums:

$S_1 = a_1 \oplus a_2 \oplus a_5 \oplus a_7 = 0;$

$S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7 = 0;$

$S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7 = 0.$

A sign of a correctly accepted combination is the equality of zero in all sums.

Assume that the sixth character of the code combination is distorted, ie instead of the combination 0111100 will be taken:

$a_1\ a_2\ a_3\ a_4\ a_6\ a_7\ a_8$

0 1 1 1 1 1 0

Substitute the values of the accepted symbols into verifiable equations:

$S_1 = a_1 \oplus a_2 \oplus a_5 \oplus a_7 = 0;$

$S_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7 = 0;$

$S_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7 = 0.$

Convert the binary number (syndrome) to decimal. We obtain the decimal number 6, which i indicates the number of the distorted character.

Heming code encoder

Based on the above rules, the Heming code encoder is built (Fig. 13.13). Bin automatically determines the value of the control symbols of the code for known information, which consist of redundant combinations of ordinary binary code.

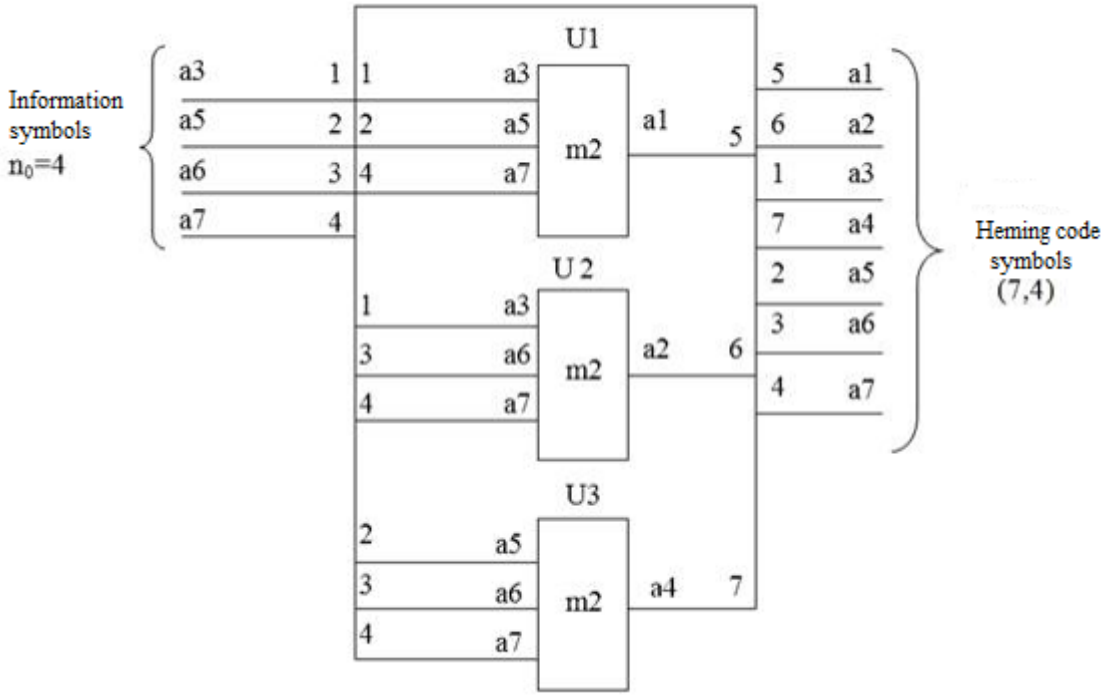


Fig. 13.13

Information symbols according to checksums are fed to the inputs of cymators by mod 2 Ul, U2, U3. At the input of the adders, the control symbols a_1, a_2, a_4 are

obtained . The information and control symbols are arranged in the required order and output via the bus for further conversion.

Heming code decoder

The Heming code decoder (Fig. 13.14) analyzes the received code combination and in case of distortion of any one character (information or control) automatically corrects the distorted character.

The input of the Heming code combination is fed to the adders by mod 2 U1-U3 according to the checksums. The result of checksums S1-S3 is formed at the outputs of adders. When taken correctly, the sums SI, S2, S3 must be zero.

In the case of distortion of one character at the outputs of the elements U1-U3 will appear binary code combination (error syndrome), which is decoded by the decoder U4. At one of the outputs of the decoder appears the level of the logical unit, which corresponds to the number of the distorted character.

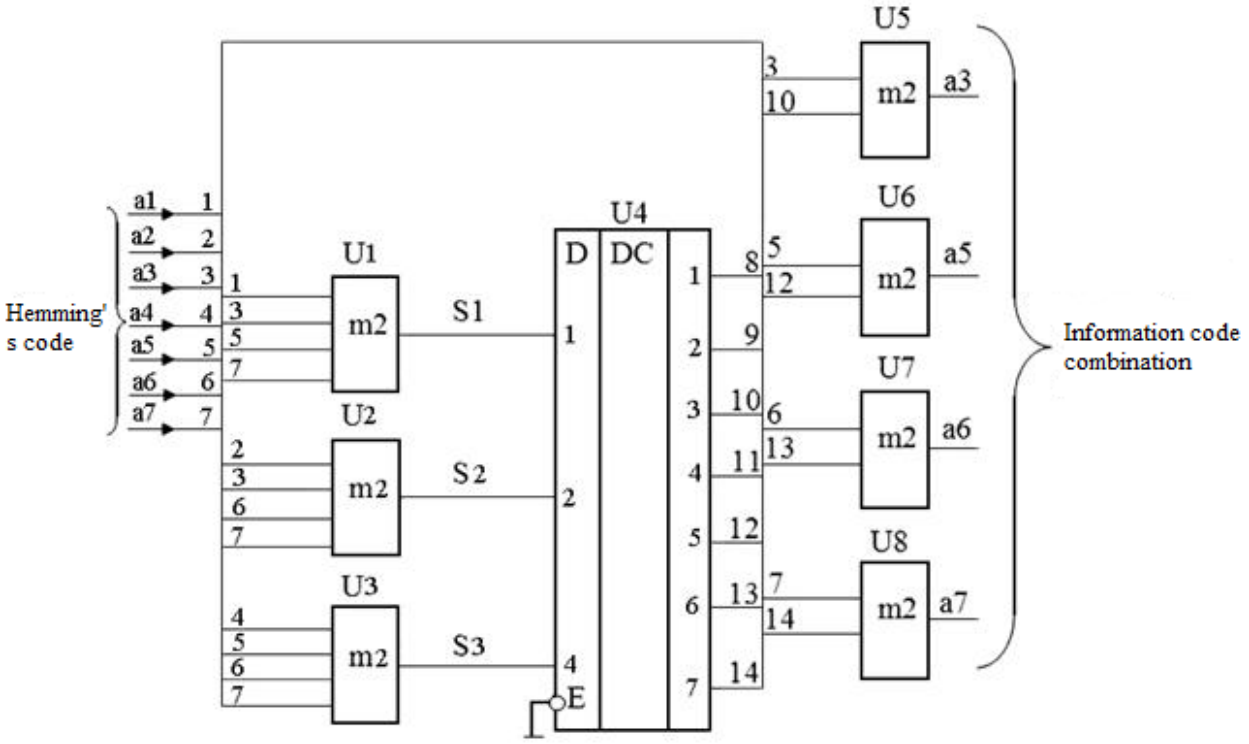


Fig. 13.14

The adders for mod 2 U5-U8 receive the corresponding signals from inputs a_3, a_5, a_6, a_7 (information symbols) and the corresponding output of the decoder.

When received correctly at the outputs 1-7 of the decoder - logic zeros and at the outputs of the adders U5-U8 will appear information symbols a3, a5, a6, a7 without changes.

In case of distortion of one of the symbols, for example, the third, the level of the logical unit will appear at the third output of the decoder and at the output of the adder for mod2 U5 the distorted symbol is automatically inverted, becoming correct.

Recurrent codes

Cyclic codes allow you to detect and correct both single and double errors, and bundles of errors. However, the practical application of these codes to correct bundles of errors is not complicated by the fact that with not too much redundancy, the length of the code combinations is much longer than the length of the bundles. In this regard, recurrent codes are more convenient.

Recurrent (continuous) codes differ in that in them encoding and decoding operations are performed continuously over a sequence of characters without dividing them into blocks.

These codes are mainly intended to fix bundles of errors. The formation of test symbols is carried out by adding two or more information symbols offset from each other by a certain distance t, called the addition step.

The length corrected by the recurrence code of the error pack l depends on the step of adding t and is determined from the condition

$l = 2t$.

The minimum required distance between bundles of errors, which ensures the correction of all errors in the bundle of length l, is equal to

$a = 6t + 1$.

Of all the recurrent codes, the most common is the so-called chain code, which is characterized by extremely simple methods of encoding and decoding.

In the chain code, each check character is formed by modulo two two information symbols spaced apart by the step of adding *t*.

Denoting the sequence of information symbols by $a_0a_1a_2...a_{2t}a_{2t+1}...$, we obtain the following sequence of valid symbols for the chain code: $b_0=a_0+a_t$; $b_1=a_1+a_{t+1}$; .. $b_t=a_t+a_{2t}$; $b_{t+1}=a_{t+1}+a_{2t+1}$... In the general flow of characters of a chain code between each two information symbols there is one check

$a_0b_0a_1b_1a_2b_2...a_tb_ta_{t+1}b_{t+1}...a_{2t}b_{2t}a_{2t+1}b_{2t+1}...$

Because the number of test symbols generated over time is equal to the number of information symbols received during the same time, the redundancy of the chain code is equal to 0.5.

At the reception, information and test symbols are separated and registered independently of each other. From the received sequence of information symbols control symbols are formed in the same way as test symbols were formed during transmission. After a delay of the value (3t + 1), each generated control symbol is compared with the corresponding received test symbol. If one of the test symbols is distorted, there will be a discrepancy between the corresponding control and test symbols. If one of the information symbols is distorted, then there will be a discrepancy between the two control and the corresponding test symbols, in the formation of which this information symbol participates.

From the considered principle of error correction in the code it follows that the correct error correction is possible only if two of the three characters covered by the check are accepted correctly. This is done provided that in the step of adding t, the length of the error packet is not more than 2t and the test characters are transmitted to the channel with a delay of (3t + 1).

Thus, the code detects and corrects bundles of errors relatively simply, but at the cost of great redundancy.

As you know, using data compression allows you to use disk space more efficiently. No less useful is the use of compression in the transmission of information in any communication system. In the latter case, it is possible to transmit a much smaller (usually several times) amount of data and, therefore, much less channel bandwidth is required to transmit the same information. The gain can be expressed in reduction of time of employment of the channel and, accordingly, in considerable economy of rent.

The scientific premise of the possibility of data compression is the well-known from the theory of information coding theorem for the channel without interference, published in the late 40's in Claude Shannon's article "Mathematical Theory of Communication". The theorem confirms that in a communication channel it is

possible to convert a sequence of source symbols into a sequence of code symbols in such a way that the average length of code symbols can be arbitrarily close to the entropy of the message source H (X), determined as:

$$H(X) = -\sum_{i=1}^{N} p(x_i) \cdot \log_2 p(x_i),$$

where $p(x_i)$ – the possibility of a specific message xi of N possible characters of the source alphabet.

The number *N* is called the volume of the source alphabet.

When transmitting messages encoded with binary uniform code, do not take into account the statistical structure of messages, which, regardless of the probability of their occurrence, are code combinations of the same length, ie the number of binary characters per constant message. Such codes are redundant.

The most effective way to reduce message redundancy is to build optimal codes that have a minimum length of codewords..

Given the statistical properties of the message source, you can minimize the average number of characters required to express a single character of the message, which in the absence of noise can reduce the transmission time or the amount of storage.

Effective coding of messages to transmit them on a discrete channel without interference is based on Shannon's theorem, which can be formulated as follows: a source message with entropy $H(Z)$ entropy can always be encoded by sequences of characters with the volume of the alphabet so that the average number of characters per message L_{av} is as close as possible $\frac{H(Z)}{\log m}$, but less than it.

To obtain the optimal code having the minimum length of the code combination, it is necessary to achieve the least redundancy of each of the word codes, which in turn must be built from equally probabilistic and interdependent characters. In this case, each code element should take the value 0 or 1 if possible with equal probabilities, and the choice of the next element should be independent of the previous one. The algorithm for constructing such a code was first proposed by K.

Shannon in 1948 and later modified by R. Fano, in connection with which it was named Shannon-Fano.

According to the algorithm at the beginning of the coding procedure, all characters of the source alphabet are entered into the table in descending order of probability. In the first stage of coding, the symbols are divided into two groups so that the sums of the probabilities of the symbols in each of them were as identical as possible. All symbols of the upper group are assigned an element of the code combination 0, and all lower - 1. In the second stage of coding, each of the groups is again divided into two equal probability subgroups. The second element of the code combinations for the upper subgroup is assigned a value of 0, and the lower - 1. The encoding process continues until there is one character left in each subgroup. Similarly, an alternative variant of the optimal Shannon-Fano prefix code can be constructed, in which in the coding process the upper subgroups of symbols are assigned a code element 1, and the lower – 0.

This code differs from the previous one in that its code combinations for the corresponding characters will be inverse.

Example. Consider the Shannon-Fano algorithm on the example of source coding, the alphabet of which consists of 8 characters a_i, $i = 1,2,...,8$, and the probabilities of occurrence of characters in the message are equal to negative powers of two, i.e. $P(a) = (1/2)^i$. For the source message ensemble to present a complete group of events, it is accepted in the example $P(a_8) = P(a_7) = (1/2)^7$. The procedure for dividing characters into groups and subgroups and the formation of code words are shown in table 13.4

The average number of bits per symbol when encoded in this case by the Shannon-Fano code is equal to

$$L_{av} = \sum_{i=1}^{8} P(a_i) l(a_i) = \frac{1}{2} + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{16} \cdot 4 + \frac{1}{32} \cdot 5 + \frac{1}{64} \cdot 6 + \frac{1}{128} \cdot 7 + \frac{1}{128} \cdot 8 = 1\frac{63}{64},$$

and the entropy of the source

$$H(A)=\sum_{i=1}^{8}P(a_i)\log(a_i)=\frac{1}{2}\cdot\log\left(\frac{1}{2}\right)+\frac{1}{4}\cdot\log\left(\frac{1}{4}\right)+\frac{1}{8}\cdot\log\left(\frac{1}{8}\right)+\frac{1}{16}\cdot\log\left(\frac{1}{16}\right)+$$
$$+\frac{1}{32}\cdot\log\left(\frac{1}{32}\right)+\frac{1}{64}\cdot\log\left(\frac{1}{64}\right)+\frac{1}{128}\cdot\log\left(\frac{1}{128}\right)+\frac{1}{128}\cdot\log\left(\frac{1}{128}\right)=1\frac{63}{64}$$

Table 13.4 - Dividing symbols into groups and subgroups

Characters a_i	Probability $P(a_i)$	Coding stages							Code combinations
		I	II	III	IV	V	VI	VII	
a_1	$\frac{1}{2}$	0							0
a_2	$\frac{1}{4}$	1	0						10
a_3	$\frac{1}{8}$	1	1	0					110
a_4	$\frac{1}{16}$	1	1	1	0				1110
a_5	$\frac{1}{32}$	1	1	1	1	0			11110
a_6	$\frac{1}{64}$	1	1	1	1	1	0		111110
a_7	$\frac{1}{128}$	1	1	1	1	1	1	0	1111110
a_8	$\frac{1}{128}$	1	1	1	1	1	1	1	1111111

Thus, the Shannon-Fano code for a given symbol probability distribution is optimal, because the average number of bits per symbol is exactly equal to the entropy of the source.

An example of Shannon-Fano coding for an ensemble of symbols with an arbitrary probability distribution is given in Table 13.5.

Table 13.5 - Coding according to the Shannon-Fano algorithm

Characters a_i	Probability $P(a_i)$	Coding stages				Code combinations
		I	II	III	IV	
a_1	0,22	0	0			00
a_2	0,20	0	1	0		010
a_3	0,16	0	1	1		011
a_4	0,16	1	0	0		100
a_5	0,10	1	0	1		101
a_6	0,10	1	1	0		110
a_7	0,04	1	1	1	0	1110
a_8	0,02	1	1	1	1	1111

The Shannon-Fano algorithm does not always lead to unambiguous code construction. This is due to the fact that when you divide the m symbols of the source into subgroups can be made larger in probability, both the upper and lower subgroups. With different subgroups, the average number of bits per symbol may be different. More efficient is the algorithm proposed by Huffman in 1952, which allows you to build the optimal code with the lowest for a given probability distribution, the average number of bits per symbol.

For binary code, Huffman's algorithm is as follows. The symbols of the message are arranged in descending order of probability and are arranged in the main column of the table in such a way that $P(a_i) < P(a_j)$ for all $i<j$ (table 13.6).

Table 13.6. Huffman's algorithm

Characters a_i	Probabilities $P(a_i)$	Additional columns						
a_1	0,22	0,22	0.22	0,26	0,32	0,42	0,58	1,0
a_2	0,20	0,20	0,20	0,22	0,26	0,32	0,42	
a_3	0,16	0,16	0,16	0,20	0,22	0,26		
a_4	0,16	0,16	0,16	0,16	0,20			
a_5	0,10	0,10	0,16	0,16				
a_6	0,10	0,10	0,10					
a_7	0,04	0,06						
a_8	0,02							

The last two symbols are combined into one auxiliary, the probability of which is equal to the total probability of its constituent symbols. All remaining characters, together with the formed auxiliary character, are again arranged in descending order of probability in the additional column. The last two elements of the column are combined into a second auxiliary symbol, and the next additional column is formed, in which all the elements are arranged in descending order of probability. The procedure continues until the only auxiliary character with a probability equal to one. To form code combinations that correspond to the characters of this message, you must trace the path of the characters in the rows and columns of the table. Code trees are most often used when constructing Huffman codes. On the one hand: this allows you to more clearly display the encoding and decoding procedures, and, on the other hand - to facilitate the software implementation of these procedures.

The average number of bits per character in this code construction is

$$L_{av} = \sum_{i=1}^{8} P(a_i) l(a_i) = 0,22 \cdot 2 + 0,2 \cdot 2 + 0,16 \cdot 3 + 0,1 \cdot 3 + 0,04 \cdot 4 + 0,02 \cdot 5 = 2,8 bit .$$

The entropy of the message source is equal to

$$H(A) = \sum_{i=1}^{8} P(a_i) \log(a_i) = -(0,22 \cdot \log 0,22 + 0,2 \cdot \log 0,2 + 2 \cdot 0,16 \cdot \log 0,16 +$$
$$+2 \cdot 0,1 \cdot \log 0,1 + 0,04 \cdot \log 0,04 + 0,02 \cdot \log 0,02) = 2,754 bit$$

As can be seen from this example, the average length of the code combination and the entropy of the source are almost the same, ie the resulting code is optimal.

The algorithm is two-pass, because in its implementation you need to view the encoded message twice. At the first pass the probabilities (frequencies) of occurrence of characters in the message are calculated and the Huffman tree is constructed.

The second pass encodes the characters received from the source. In this case, the value of the branches of the tree is determined when moving from the leaf corresponding to the encoded symbol to the root. Obviously, to speed up the decoding procedure, the bits of the code combination at the output of the encoder must be issued starting from the highest bit, ie from the branch of the tree coming from the root. In practice, instead of symbol probabilities, absolute values of the number (weight) of symbols in the transmitted message are used, because the number

of symbols is proportional to the probability of their occurrence. For unambiguous decoding, the symbol probability table is reported to the decoder.

Dynamic coding by the Huffman method

The classical method of Huffman coding determines from the beginning of the transformation the probability of occurrence of symbols at the output of the information source, and the symbols are ordered in descending order of the probabilities of their occurrence. A code table is compiled for an ordered list, in which the length of the original combination is determined by the probability of the original character. Naturally, tables (code trees) for each compressed message must be known on the transmitting and receiving sides. This method has two significant disadvantages. The first is that its implementation requires two passes of the coded array.

At the first view, the probabilities of each character in the message are calculated and a Huffman code table is compiled. The next step is to encode based on the static structure of the Huffman tree and transfer characters in compressed form. Thus, the gain obtained by data compression can be significantly reduced, especially when transmitting a relatively short message, due to the need to transmit to the decoder additional information about the code tree.

The second disadvantage is the presence of a delay from the moment of receipt of data from the source to the issuance of appropriate code combinations, which limits the use of uneven coding in synchronous networks, as well as in systems that operate in real time.

In the early 1970s, one-pass information compression methods were developed based on the classical Huffman coding procedure. All these methods are slightly different from each other and their essence is that the transmitter builds a Huffman tree at the rate of data from the source, ie "on the fly". The coding process "learns" the encoder based on the static characteristics of the message source, during which the estimate of the initial probabilities of the message is calculated and the corresponding modification of the code tree is made, this process is called dynamic Huffman coding.

Obviously, to properly recover compressed data, the decoder must also constantly "learn" along with the encoder, making a synchronous change of the code table on the receiving side. To ensure the synchronization of the encoding and decoding processes, the encoder outputs the character in uncompressed form, if it first appeared at the output of the source, and determines it on the code tree. With the reappearance of the symbol at the input of the encoder, it is transmitted by an uneven code combination due to the position of the symbol on the current code tree. The encoder corrects the Huffman tree by increasing the frequency of characters already entered in the tree, or increases the tree by adding new nodes.

The most important condition that must be observed when modifying the code tree is the preservation of the properties of the Huffman tree. To formulate these properties, we turn again to the algorithm for constructing the optimal Huffman code. In static encoding, the characters are placed in a list in descending order of weights (probabilities). The two lightest knots W_i, W_j are then combined and replaced with an internal knot with a weight equal to the sum of the original weights $W_i + W_j$. The newly formed node is placed in the list so as not to violate the order of the nodes by weight. This process is repeated until one, the so-called *root* node remains in the list.

Consider examples of constructing trees of the optimal Huffman code for the letters of the alphabet from A to H. The weights of these symbols, respectively, have: A = 18; B = 10; C = 2; D = 2; E = 1; F = 1; G = 1; H = 1. Let's analyze their properties. As can be seen from the figures, the nodes of the tree are arranged in ascending order of their weight when traversing the tree from the lowermost node to the root on the left, right and bottom up.

Because nodes are paired with weights W_i, W_j there cannot be less than two nodes on the same level, and pairs of nodes are children in a common parent node whose weight is equal to the sum of the weights of the child nodes. It is easy to see that when constructing a tree, it is assumed that a child node with a large weight is connected by a zero branch, and with a lower weight - a single one, then the Huffman tree remains ordered by increasing weight when moving from the lower node to the root in levels from right to left.

When constructing code trees and their modification, the nodes are numbered from the first to the $(2m-1)$-th in the order of increasing their weights. The first number is assigned to the node with the minimum weight. Here m – is the number of characters in the source alphabet. In fig. 13.15 shows a Huffman tree constructed for the above example 18A, 10B, 3C, 2D, 1E, 1F, 1G, 1H, respectively. Let's make numbering of knots of a tree, beginning with a leaf with the minimum weight (the bottom extreme left knot). As can be seen from the figure, the properties of the Huffman tree constructed for static code are retained, although it has acquired a different configuration.

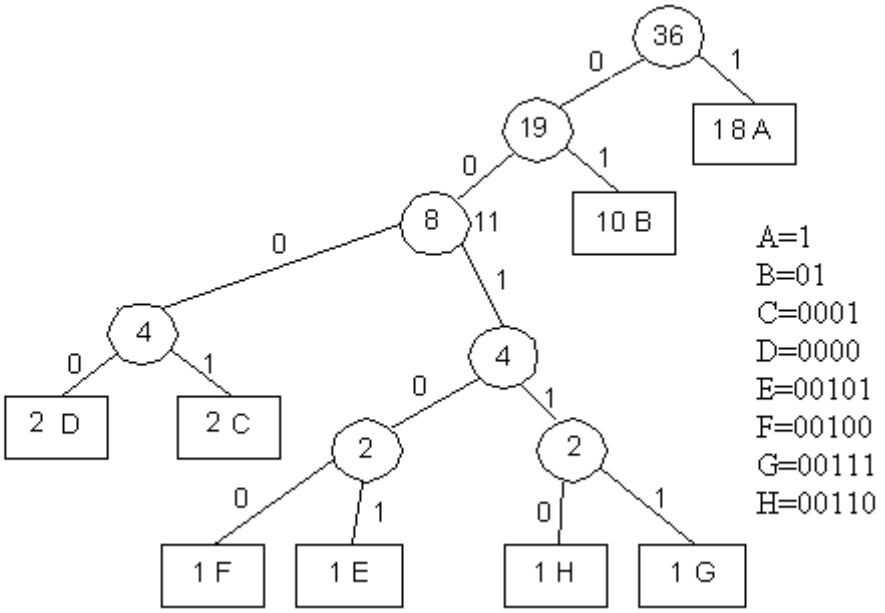


Fig. 13.15

When dynamically encoding Huffman after receiving the next symbol from the source (for example, "C") should increase by 1 weight of the corresponding sheet (Fig. 13.16). In this case, the weight of all nodes on the way from the leaf with the symbol C to the root will increase by 1. But this will lead to a violation of the order of the nodes by weight, inherent in the optimal code tree. Therefore, when receiving the next character from the source, it is necessary to modify the code tree so that it remains optimal, the so-called Huffman tree.

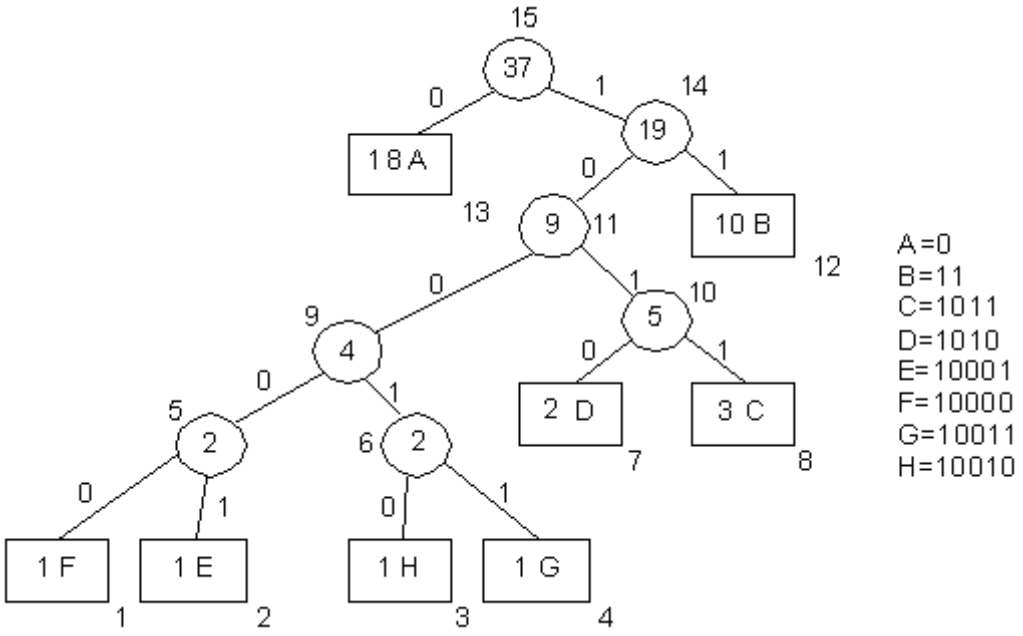


Fig. 13.16

Thus, based on the analysis of the properties of optimal code trees, we can conclude that a code tree that has external nodes is Huffmanian, if it has the following properties:

a) the external nodes (leaves) of the Huffman code tree have weight $W > 0$; each internal (parent) node has subordinate (child) nodes, and its weight is equal to the sum of subordinate (child) nodes;

b) at each level of the tree (except for the root) there must be at least one pair of nodes that have a common parent node;

c) all nodes are numbered in ascending order so that nodes with numbers (2*j*-1) and 2*j* are nodes of one level for l <*j*<*m*-l, and their common parent node has a higher level;

d) the numbering of nodes corresponds to the order in which the nodes are combined according to the static Huffman algorithm.

Dynamic coding by FGK method

Huffman's dynamic code synthesis algorithm was first proposed by N. Feller in 1973, and then modified by R. Gallagher and D. Batog. This is why it is called the "FGK algorithm".

Lets introduce some notations that will be used in the analysis and synthesis of a dynamic Huffman code tree by the FGK algorithm:

m – the alphabet size of the message source;

z_j – j-th alphabet symbol;

$M(k) = z(1), z(2), ..., z(k)$ – the first k characters in the message;

k – the number of characters in the message processed up to the current time;

$z(k)$ – k-th character in the message;

K – the number of different characters processed at a given time;

W_j – the number (weight) of characters z_j, received at the time of message processing;

l_j – the distance from the root of the tree to the z_j-th leaf.

The essence of the algorithm for the synthesis of a dynamic code tree Huffman is the procedure for calculating the leaves and constructing a binary tree with a minimum path weight $\sum_j W_j l_j$. The procedure can be divided into two stages, although when implementing the algorithm, they can be easily combined into one. In the first step, the Huffman tree constructed after processing the message $M(k)$, will be transformed into another, equivalent to the original, which can then be converted into a Huffman tree for simple weight gain $M(k+l)$.

The first stage begins after receiving from the source of the symbol $z(k+l)$ with the assignment of the status of the current node of the leaf $z(k+l)$. Then the current node (including the subtree formed by it) is exchanged with the node with the largest sequence number and weight. After that, the parent node of the last current node is initiated as the new current node. The exchange of nodes, if necessary, is repeated many times until the root of the tree is reached. It is easy to see that the maximum number of permutations that may be required when modifying the code tree is equal to the height of the tree $l_{\max}$.

In the second stage, a tree leaf corresponding to the processed symbol and the following intermediate nodes located on the path from the leaf to the root of the tree are formed..

The original method was proposed by D. Knut. Its essence is that all the characters of the alphabet used, which have not yet appeared at the output of the

source, are marked on the tree with a node zero. Therefore, when the character to be compressed is generated for the first time, the encoder "marks" it, producing a prefix combination. It is followed by the code of the uncompressed character. The code corresponding to the zero weight sheet is used as a prefix combination. For the encoding of characters received from the source for the first time, not the 8-bit combination of ASCII-code is used, but the minimum prefix code, built on the following considerations: if there is a message consisting of m characters $a_i,...,a_m$, the number m can be given as an integer of two and an integer $m = 2^e + r$, where $0 < m < 2^e$.

Then the a_k-th character is encoded by the $(e+1)$-th bit combination of the number $k-1$, if $1 \le k \le 2r$.

Otherwise – e-th bit combination of numbers $k-r-1$. For example, if $m=6$, then $e=2$ and $r=2$, and the source characters will be displayed in combinations: $a_1 \Rightarrow 000$, $a_2 \Rightarrow 001$, $a_3 \Rightarrow 010$, $a_4 \Rightarrow 001$, $a_5 \Rightarrow 10$, $a_6 \Rightarrow 11$.

It is easy to notice that the received code has property of a prefix. This code is optimal if the characters have the same probability.

Dynamic coding by the Witter method

Further improvement of the algorithm for dynamic data coding by non-uniform FGK codes was proposed by D. Witter. This algorithm is called algorithm V. When looking for ways to optimize the procedure of encoding data with Huffman code, the author proceeded from the fact that in the new algorithm the number of node exchanges in the code tree modification process should be limited to a small number (at best one), and a dynamic Huffman tree should be constructed in such a way as to minimize not only the total length of the external path $\sum_j W_j l_j$, but also the magnitude $\sum_j l_j$, $\max\{l_j\}$.Minimizing the height of the tree $h = \max\{l_j\}$ will prevent the formation of long code combinations when encoding the next character in the message. Witter largely managed to solve the problem. The algorithm developed by him has the following two advantages in comparison with the FGK algorithm.

1. The number of node exchanges in which the current node moves up the code tree during its modification is limited to one. In the FGK algorithm, the upper limit of the number of exchanges is $\frac{l_j}{2}$, where l_j – the length of the codeword for the $z_j(k+l)$-th character before the modification procedure.

2. Algorithm V minimizes the length of the outer path of the tree l_j and guarantees a tree of minimum height $h = \max\{l_j\}$ while minimizing the total length of the outer path of the tree $\sum_j W_j l_j$.

The essence of the improvement of algorithm V is the introduction of a new numbering system of nodes of the code tree, called *implicit numbering (Implicit numbering*).

With implicit numbering, the nodes of the Huffman tree are numbered in ascending order from left to right and bottom up, that is, lower-level nodes have numbers smaller than nodes of the next level. The most important feature of implicit numbering is compliance with the necessary condition for building a tree.

For each weight *W*, all external nodes (leaves) of the tree with weight *W* must transmit to all internal nodes of weight *W*.

It is easy to see that this condition is one of the distinguishing features of implicit numbering compared to other algorithms.

Therefore, if the transmission continues from the symbols d, c, g, e, f or from not yet used symbols, the generated codewords in the algorithm V will be shorter than in FGK, ie the strategy of minimizing the external path and height of the tree is optimal assuming that any symbol that appears is equally likely.

The second distinguishing feature of algorithm V is the introduction of the concept of a block of equivalent nodes. In this case, the nodes v and b are equivalent only if they have the same weight and both are either internal or external nodes. The node of the block that has (with implicit numbering) the highest number is called the *leader of the block*. The blocks are ordered by increasing weight, and the block of leaves of weight W must be equal to the block of internal nodes of the same weight.

The third distinguishing feature of algorithm V is the method of modifying the tree after obtaining the next symbol. The main operation of the algorithm to support the condition of implicit numbering (*) is sliding and magnification (*Slide and Increment*). The essence of this operation is that the node declared current exchanges with the leader of its block and then slides in the direction of the tree root on the neighboring block, which is directly adjacent to the block of the current node. Sliding continues until the current node passes the entire block and is installed at the beginning of this block. Then the weight of the current node is increased and the father of the old current node is appointed as the new current node. The operation of sliding with magnification continues until the root of the tree is reached. The choice of the parent node depends on whether the current node was a leaf or an internal node. If the current node was a leaf, the new current node is the parent with whom the current node was associated after the slide. And if the current node was an internal node, then the new current node is assigned to its parent node, which was associated with the current before the slide.

Lecture №14. Multi-channel systems with frequency, time and phase signal distribution. Multi-channel digital information transmission systems. Systems with δ - modulation, phase manipulation, square manipulation. Protection against errors in data transmission systems. Digital communication

The practice of construction of modern telecommunication systems and networks shows that the most expensive links of transmission paths are communication lines (cable, fiber-optic, radio relay and satellite communication lines, etc.). OAs it is not economically feasible to use an expensive communication line to transmit information to a single pair of subscribers, the task arises of building *multi-channel transmission systems* that provide a large number of messages from different sources of information from a common communication line.

Multi-channel systems as well as single-channel can be analog and digital. As a standard channel in analog systems the tone channel (PM channel) is taken, which provides transmission of messages in the frequency band 300 ... 3400 Hz, corresponding to the main spectrum of the telephone channel. In digital systems, the main channel is 64 kbit / s. *Multi-channel analog systems* are formed by combining PM channels into groups, usually multiples of 12 channels. *Digital transmission systems (DTS)* are also formed in accordance with the accepted hierarchical structures.The European hierarchy is based on the primary *DTS* type PCM-30 with a group flow rate of 2048 kbit / s; the basis of the North American system is PCM-24 with a group signal speed of 1544 kbit / s. *DTS*s developed in our country correspond to the European hierarchy.

The principle of construction of multichannel communication systems is explained by the structural scheme on fig. 14.1.

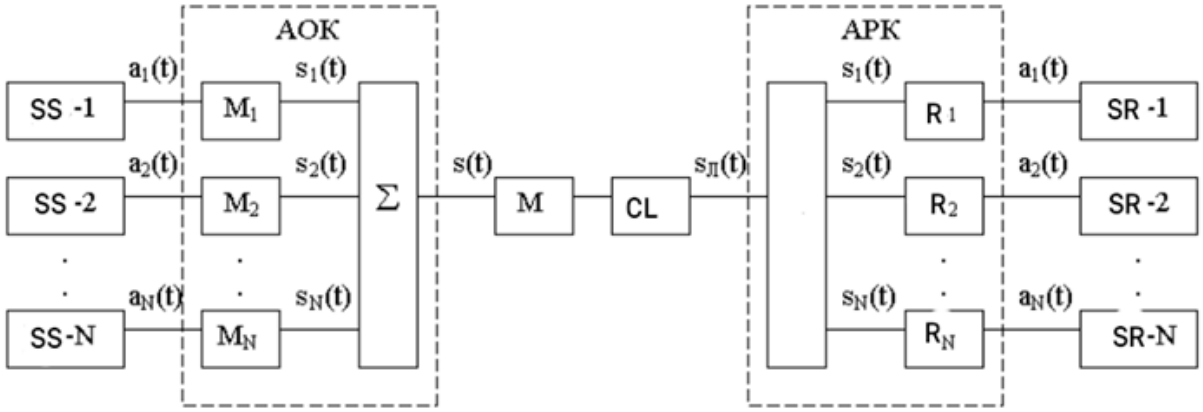


Fig. 14.1

Suppose it is necessary to organize the simultaneous and independent work of N individual channels on a common group path. We believe that the group path can provide signal transmission of any i-th channel $S_i(t)$. Assume that the signal of the i-th channel $S_i(t) = C_i\psi_i(t)$, where $\psi_i(t)$ is carrier function; C_i – some coefficient that reflects the transmitted message. The sum of the channel signals forms a group signal $S_g(t) = \sum_{i=1}^{N} S_i(t) = \sum_{i=1}^{N} C_i\psi_i(t)$. After converting a group signal to a linear one, the latter enters the transmission path (communication line). On the receiving side $S_л(t)$ will again be converted into a group signal, i.e. to a form convenient for performing the signal separation operation. To separate the N channel signals on the receiving side requires a corresponding number of N separating devices, the action of each of them is described by the division operator π_k. Ideally *k-e* separating device must respond only to the signal of *k-th* channel $S_k(t)$ and give zero feedback on signals from other channels: $\pi_k = \{S_g(t)\} = \pi_k = \left\{\sum_{i=1}^{N} C_i\psi_i(t)\right\} = \begin{cases} C_k & i = k \\ 0 & i \neq k \end{cases}$.

A necessary and sufficient condition for signal separation is the condition of *linear independence of channel signals*. It is that identity $C_1\psi_1(t) + C_2\psi_2(t) + \ldots + C_i\psi_i(t) + \ldots + C_N\psi_N(t) \equiv 0$ can be performed in the only case when all the coefficients $C_1, C_2, \ldots C_N$ simultaneously turn to zero. Physically, this means that none of the channel signals can be formed by the sum of other channel signals.

The criterion of linear independence is the inequality to zero of the Gram determinant $G(\psi_1,\psi_2,...,\psi_N)=\begin{vmatrix} (\psi_1\psi_1) & (\psi_1\psi_2) & ... & (\psi_1\psi_N) \\ (\psi_2\psi_1) & (\psi_2\psi_2) & ... & (\psi_2\psi_N) \\ \vdots & & & \vdots \\ (\psi_N\psi_1) & (\psi_N\psi_2) & ... & (\psi_N\psi_N) \end{vmatrix}$, where $(\psi_i\psi_k)$ is a scalar product. The determinant is zero if the functions $\Psi_1, \Psi_2,...\ \Psi_N$ linearly dependent and is positive for linearly independent functions; it is equal to the product of the squares of the norms of the functions if Ψ_i і Ψ_k are orthogonal. In particular, for orthogonal signals, the action of the division operator is reduced to the multiplication of the received group signal by the reference function of the carrier $\Psi_k(t)$

$$\pi_k\left\{\sum_{i=1}^{N} C_k\psi_k(t)\right\}=\int_0^T \psi_k(t)\sum_{i=1}^{N} C_i\psi_i(t)dt = C_k\int_0^T \psi_k(t)\psi_k(t)dt = C_k .$$

The property of orthogonality of functions $\Psi_i(t)$ і $\Psi_k(t)$ is used*:*

$$\int_0^T \psi_i(t)\psi_k(t)dt=\begin{cases}1 & i=k\\ 0 & i\neq k\end{cases}.$$

From the standpoint of geometric representations, the latter condition means that the carriers $\Psi_i(t)$ and $\Psi_k(t)$, and, consequently, the signals $S_i(t)$ and $S_k(t)$ should occupy non-overlapping areas in the signal space.

Let's trace the main stages of formation of a multichannel signal at frequency division of channels (FDC). We will assume that each of the messages to be transmitted $a_i(t)$ зоccupies the frequency band of a standard PM channel $\Delta\omega_a$. In the process of forming a group signal to each channel signal $S_i(t)$ the frequency band $\Delta\omega_i$ which does not overlap with spectrum of other signals is allocated . Then the total frequency band of the *N-channel group will be equal to* $\Delta\omega_g=\sum_{i=1}^{N}\Delta\omega_i$. Assuming that single-band modulation is used, and each channel signal occupies a frequency band $\Delta\omega_i=\Delta\omega_a=\Omega_m$, for the spectrum of the group signal we obtain $\Delta\omega_g=N\Delta\omega_a=N\Omega_m$.

Group signal $S_g(t)$ converted into a linear signal $S_l(t)$, is transmitted over a communication line (transmission path). On the receiving side after conversion of a

linear signal in group, last by means of band channel filters Φ_k with bandwidth $\Delta\omega_k$ and demodulators will be converted into channel messages $a_k(t)$, which are sent to the recipient. *In short, in multi-channel systems with CRC, each channel is assigned a certain part of the total frequency band of the group signal.* At the input of the receiving device of the *i-th* channel signals S_i *of all N channels* operate simultaneously . Using frequency filters Φ_i only those frequencies are allocated $\omega \in \Delta\omega_i$, belonging to this *i-th channel*. Due to the imperfect characteristics of the bandpass filters, there are mutual transient interference between the channels. To reduce these interferences, it is necessary to introduce protective frequency intervals between channels $\Delta\omega_z$. This means that only about 80% of the transmission bandwidth is effectively used in FCD systems. In addition, it is necessary to ensure a very high degree of linearity of the entire group path.

In the temporary channel division method (TCD), the group path is alternately provided by the synchronous switches of the transmitter (K_{tr}) and the receiver (K_{re}) to transmit signals of each channel of the multi-channel system. First, the signal of the 1st channel is transmitted, then the next, etc. to the last channel at number N, after which the 1st channel is connected again, and the process is repeated with a sampling frequency f_s. Non-overlapping sequences of modulated pulses are used as channel signals in TCD systems. $S_i(t)$; set of channel pulses - group signal $S_г(t)$ is transmitted over a communication line. The action of the switch on the receiving side K_{re} can be identified with the key that connects the line to the receiver of the i-th channel only for the time of passage of the pulses of the i-th channel ("temporary filter" $\Phi_{i)}$. After demodulating the messages $a_i(t)$ come to the *i*-th recipient.

Transient interference between channels is also possible during temporary division. The first reason is the imperfection of the frequency response and PFC of the transmission path, the second is the imperfection of the synchronization of switches on the transmitting and receiving side. At TCD temporary protective intervals are also entered.

At TCD the signal of each channel occupies the general frequency band of multichannel system $\Delta\omega_p$. If the message sampling interval is $T_s = \pi/\Omega_m$, the duration

of the channel pulses is equal to $\Delta t_k = T_d / T = \pi / T\Omega_m = \pi / \Delta\omega_g$, that is $\Delta\omega_g = N\Delta\Omega_m$. Although theoretically TCD and FCD are equivalent in terms of efficiency of frequency spectrum use, in real conditions TCD systems are noticeably inferior to FCD in this indicator due to the difficulty of reducing the level of mutual interference in signal separation. However, the undeniable advantage of TCD is the reduction of the level of interference of nonlinear origin due to the different timing of the pulses of different channels, in TCD systems below the peak factor. It is also important that the TCD equipment is much simpler than the FCD equipment. TCD is most widely used in digital PCM systems.

A special case of temporary division is the division of channels by phase (PCD). In this case, harmonic oscillations are used as signals: $S_1(t) = a\sin\omega t$, $S\ (t) = a\sin(\omega t + \Delta\varphi)$. At $\Delta\varphi = \pi/2$ signals $S_1\ (t)$ and $S_2\ (t)$ are orthogonal and easily separated by synchronous or correlation detectors. When $N > 2$ the Gram determinant for signals that differ in phase, vanishes. This means that only two-channel transmission can be provided when dividing by phase.

For signal separation can be used not only signals with non-overlapping spectra (FCD) and signals with non-overlapping time intervals of channels (TCD). In the general case, *the signals occupying the common frequency band and transmitted simultaneously can be separated if they satisfy the condition of linear independence.* These requirements are met by signals that differ in shape (RKF). As the simplest example, you can choose carriers in the form of pulses that reflect the static series $1, t, t^2, \ldots$ $(0 \le t \le T)$. Based on the orthogonalization of the functions of the static series, the orthogonal polynomials of Lagerr, Hermit, and Chebyshev, used in analog multichannel telemetry systems, have been developed.

Orthogonal sequences in the form of Walsh functions (analogs of "rectangular" sines and cosines) are used in digital multichannel systems with form division. Each channel is allocated its "address" in the form of a pulse sequence. The operation of dividing the group signal on the receiving side is carried out using a set of correlation receivers, the reference oscillations for which are known on the receiving side.

Generalization of the division by form, are *asynchronous-address communication systems (AACS).* In AACS, the rigid orthogonality requirement is replaced by a more flexible requirement of "almost orthogonal" signals. These include primarily noise-like signals for which the correlation function is similar to the correlation function of white noise with a limited spectrum. Noise-like signals are formed on the basis of pseudo-random sequences. The sequence form is used as the "address" of the channel. An important advantage of AACS is that there is no need for a central switching station: just type the "address" of the called party, ie change the "form" of the pulse address sequence. Another advantage is that due to the free access to the communication line, any N_a active subscribers from the total number of N subscribers of the system can transfer, and $N >> N_a$. Such systems easily implement bandwidth reserves arising from "low-active" subscribers, for example, you can organize a 1000-channel communication system, which simultaneously transmit any 50-100 subscribers per thousand.

In the combinational method of separation, the group signal is a display of certain combinations of discrete channel messages using numbers corresponding to the combination number. For example, for binary channels $m = 2$ (symbols 0 and 1) when $N = 2$ four combinations are possible: 00, 01, 10, 11, exposed numbers: 0, 1, 2, 3. To each of numbers the transmitter brings in unambiguous correspondence a signal, for example, at FS - pulses on frequencies f_0, f_1, f_2, f_3. On the receiving side by means of the decoder on each value of frequency the number of a code combination, and, consequently, value of symbols in channels is reproduced. This method of transmission is called DFT (double frequency telegraphy). Similarly, you can perform DPT (double phase telegraphy), with the signals of the code combinations differ in phase values: $\varphi_0, \varphi_1, \varphi_2, \varphi_3$ (Fig. 14.2). N-channel systems will require $M = 2^N$ appreciable values of the modulated parameter (frequency, phase). In the General case, it is possible to modulate several carrier parameters simultaneously, for example, amplitude and phase, frequency and phase, etc. Recently, great interest has been shown in the signals of *amplitude-phase modulation (APM),* which provides amplitude modulation of in-phase and quadrature components. If you use four-level signals to modulate each component $C(t) = \pm 1, \pm$

3, then a 16-position KAFM is obtained (Fig. 14.3). Along with the considered square network ensembles of signals on the basis of a triangular network, various options of circular arrangements of signal points are developed.

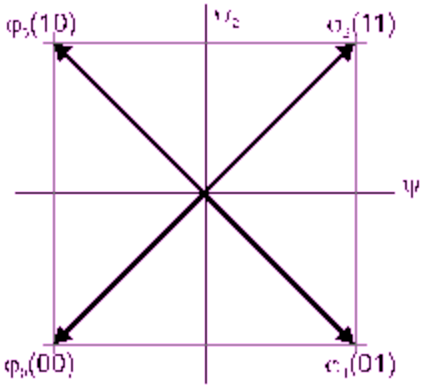


Fig. 14.2

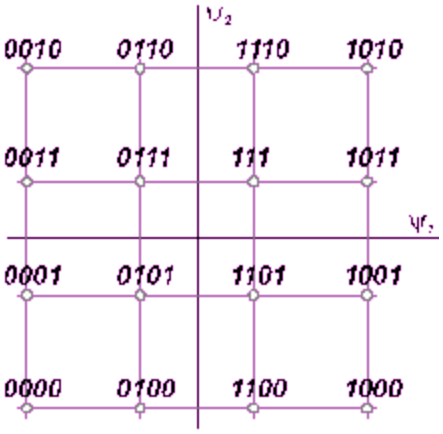


Fig. 14.3

Let us now turn to the principles of information distribution. In order to organize the exchange of information between many sources and recipients of information (subscribers), channels and transmission systems are combined in a communication network. switching nodes. An example of the simplest system of information transmission and distribution is a fully connected network (Fig. 14.4), where endpoints (EPs) are connected to each other on a "one-to-one" basis.

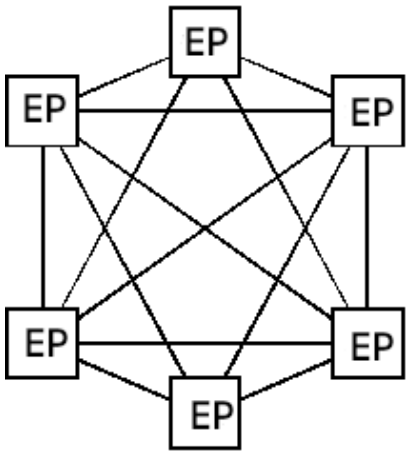


Fig 14.4

Such networks are non-switched networks, where communication between N subscribers (users) is carried out on fixed channels. In each EP it is enough to have only a switch on (N -1) positions, which provides connection of the EP to the desired

channel both for outgoing and incoming communication; there is no loss of connection timeout. However, how easy it is to see from Fig. 14.4 with increasing number of EP sharply increases the number of required connecting channels, for example, if the number of EP is *N*, you will need $N(N-1)/2$ channels .

In order to reduce the number of required channels, switched communication or communication networks are used, where the EPs are connected not directly, but through switching nodes (SN).*When switching channels on the transmitted address of the OP of the recipient, a path is found with currently free channels.* The channel thus formed is provided to the users of EP_S (source) and EP_R (recipient) for the time required to exchange information. In Fig. 14.5 as an example is the structure of the switched telephone network, which contains a set of EP, QC and connecting channels (lines). This method of connection is called channel switching (CS).

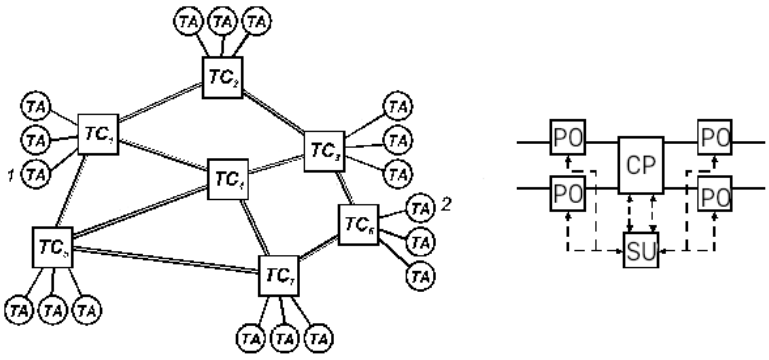


Fig. 14.5

With a heavy network load, some subscribers may be denied a connection. Therefore, measures are designed so that the probability of denial of service was less than a certain allowable value, but it is impossible to completely rule out such a situation.

In many cases, especially in the one-way transmission of information, you can use the method of switching messages (SM). In such systems, in the switching nodes of messages built on a computer, the transmitted messages, accompanied by the address, are received from the EP_S and without failure, processed and stored in the memory of the switching center. The transfer of information to the address of EP_R is made as the necessary channels are released. At the same time, of course, there is a

delay, which makes it difficult to conduct the transmission in real time, for example, during a telephone conversation.

In order to reduce the delay time in modern devices, a variant of the CS method is used, called packet switching (PS), when short blocks of data, called packets, are transmitted from EP_S to EP_R. For example, in the interactive mode, short packets of messages containing 100-200 characters are transmitted between terminals and computers. To date, packet methods of speech signal transmission have been successfully developed and implemented.

The load of the telephone network depends on the number, time of occurrence and duration of telephone conversations. Statistical characteristics of the call flow are studied on the methods of queuing theory, in particular the *theory of teletraffic.*

This theory allows you to set requirements for switching devices and the number of lines that guarantee satisfactory communication quality at a given failure rate or latency.

Under the intensity of the load means the mathematical expectation of the incoming load per unit time (in telephony - 1 hour). *Erlang is accepted as a unit of measurement of intensity of loading (1 hour-lesson).* During the day the load changes, the time of the greatest load is called GLT. Each subscriber on average gives a load in the range of 0.06 ... 0.15 Erl. This value is used to calculate the telephone network and its switching systems.

If the beam, which serves a large number of n load sources that create a Poisson call flow, there are only m channels, each of which deals with the call on average at time T, the probability of message loss (blocking) is according to the Erlang formula $P(B) = \dfrac{A^m / m!}{\sum_{i=0}^{m} A^i / i!}$, where $A = \lambda\ T$ – load intensity in Erlang. In other words, in practice, systems with n> m are provided with many, but not all, possible compounds. This leads to losses (failures), the allowable probability of which is in the range of 0.001 ... 0.01.

To effectively implement the bandwidth of transmission channels and switching nodes, *a set of standard procedures (rules) of interaction and software* is

used to ensure communication, communication interruption if necessary, etc. All of these procedures and rules make up a tiered communication architecture; їIt is based on the concept of a reference model of open systems interaction (OSI), ensuring the introduction of standards at the international level for newly created networks.

To simplify the development and implementation of a network architecture, each system is divided into a number of quasi-independent functional levels. In this case, the interaction of systems in the network is provided in the form of interaction between elements (logical objects) of systems of the same functional level. The reference model of OSI uses seven levels (Fig. 14.6). The lower three levels provide network services, and the upper four provide services to the most end users. The level of the data channel and the physical layer below it provide a channel of error-free transmission between two nodes in the network. The function of the network layer is to set the address and route for transmitting the data packet over the network from the transmission node to the destination node.

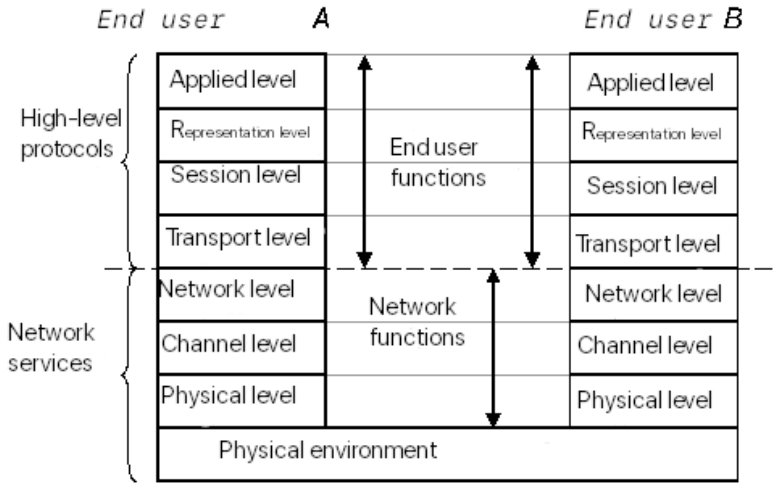


Fig. 14.6

The lower of the upper levels of the OSI (transport layer) provides end-to-end data transmission between network subscribers with a given quality of service, which is a component parameter that determines the characteristics of subscriber interaction. The session level provides the organization of dialogue between network subscribers, ie management of data transfer order, their priority, recovery procedure, etc. The presentation layer controls and converts the syntax of the data block exchanged by end users (codes, data formats, data compression, machine languages,

etc.). Finally, the application layer is used to perform all information and computational processes provided by the user through the transport network: e-mail, teletext, fax, electronic money transfer, packet transmission of voice messages, etc.

The rules of interaction of objects of one level are called protocols and define logical interaction. The reference model of the Air Force adopts the concept according to which the interaction of level objects is provided by providing it with services of the adjacent lower level. The rules of interaction of objects of adjacent levels in one system, as well as inter-network exchange are called interfaces. The most developed are the main standards of the ITTAC (International Telephony and Telegraphy Advisory Committee) and the IOS (International Organization for Standardization), on which large network projects are focused.

Packet-switched data networks in accordance with ITTAC Recommendation X.25 have become widespread. These networks are public data networks that provide services at the three lower levels of the OSI reference model. The ITTAC X.25 Recommendation covers the connection of data terminals, computers and other user systems and devices to the data network. In other words, the X.25 protocol is a communication protocol, it controls the interaction between the end data transmission device (DTE) and the linear data transmission device (LDTD). The procedures of the X.25 protocol and the protocols of other levels of the OSI architecture are described in detail in the special literature.

The basis of the information complex of the Russian Federation is an *interconnected communication network* (ICN), which provides telecommunications services (ES) in the country to the vast majority of subscribers. The most important last part of the OSI is the primary network, which consists of a set of endpoints (EP), transmission lines and switching nodes, forming a network of typical channels and typical group paths that provide transmission of any type of information. The primary network combines trunk, as well as zone and local (urban and rural) primary networks. *Zone primary networks* are networks of regional, regional and national importance that connect local networks with the center of the zone and with each other собою. *Zone networks are connected to the interconnected communication network of the country by means of trunk lines.*

On the basis of typical channels and paths of the primary network secondary networks are formed, which depending on the type of transmitted information are divided into telephone, telegraph, data networks, audio and television broadcasting, facsimile, various departmental networks (transport, navy and air, networks communications for gas and oil pipelines, banking structures, etc.).

The main direction of development of the primary OSI network is the introduction of digital transmission systems (DTS on metal cable, fiber-optic TS, digital RRL, satellite TS). The goal of digitization should be to create a digital primary network designed to provide digital channels to secondary networks and consumers. When building OSI the principle of organizational and technical unity is realized: carrying out of uniform technical policy, application of uniform complex of the most unified technical means, uniform nomenclature of standard channels, and network paths, construction of uniform for primary and secondary networks of operation system.

To interconnect public switched networks with international networks, the ITU (International Telecommunication Union) allocates international codes, interface parameters. Widespread implementation of digital transmission systems, electronic switching, fiber-optic and satellite communication lines allows to provide such information systems that can fully meet the requirements of modern society.

Lecture №15. Optimal reception of continuous and discrete messages Characteristics of interference. Noise immunity for optimal message reception. Coordinated filtering

One of the important tasks of radio engineering is the task of finding the signal against the background of interference. Let the input of the system that solves this problem be oscillated *u(t)*, which in the absence of a signal is noise *n(t)*, that is *u(t) = n(t)*, and in the presence of a signal in the simplest case, it is the sum of the signal and noise: *u(t) = n(t) + s(t)*.

Thus, the signal at the input of the receiver can be recorded as follows *u(t) = n(t) + λs(t)*, where $\lambda = 0$, if there is no signal (denote this case A_0) and $\lambda = 1$, if it is present (case A_1).

λ is unknown in advance and needs to be determined. Errors are possible due to the random nature of the noise or always when solving the problem of finding. If the receiver determines that $\lambda = 0$, let's assume - this is the case B_0, and if $\lambda = 1$, then the case B_1. There are four possible solutions.

1. There is no signal and the receiver determines that it is not - the correct location of the signal (case A_0B_0);

2. There is no signal, and the receiver determines that it is - erroneous finding of the signal, false alarm (case A_0B_1);

3. The signal is there and the receiver determines that it is - the correct location of the signal (case A_1B_1);

4. There is a signal, and the receiver determines that it does not exist - the signal is missed (case A_1B_0).

The simplest solution is to find a completely unknown signal. The task is complicated by the increase of random unknown parameters (or characteristics) of the found signal. In this case, the signal is described not only as a function of time, but also as a function of random parameters *s(t*, α), where $\alpha = \alpha(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a matrix of signal parameters. The task of solving several signals is even more difficult. In this case, a signal u(t) is applied to the input of the receiver , which is the sum of the interference *n(t)* and two signals $s_1(t)$ та $s_2(t)$, which may overlap, ie $u(t) = \lambda_1 s_1(t) +$

$\lambda_2 s_2(t) + n(t)$, where λ_1 i λ_2 – random variables that can take values of 0 or 1. If the signals are not completely known, then the problem of solving signals with random parameters arises: $s_1(t, \alpha_1)$, $s_2(t, \alpha_2)$.

The concept of "solve two signals" can have different meanings. If it is possible to have two signals in the adopted implementation u (t) at the same time, we can mean only a separate finding of signals with given indices, and it is possible not only a separate finding, but also the definition of any parameter α_i. Thus, if the indicators of finding the second signal remain above the allowable in the presence of a random first signal, then we can say that the second signal is solved in essence by finding (estimating) the parameter α_i. If, in addition, the first signal is resolved in the presence of the second, then we can say that the signals are mutually resolved in essence (estimates of the parameter α_i). The mathematical basis for solving all problems of signal reception associated with the risk of errors in deciding on the presence of a signal in its absence, and signal omission, is the theory of probability and mathematical statistics, while providing the ability to take into account not only information about received oscillations , but also a priori data concerning a signal and noise.

Characteristics of interferences

Interference is usually called extraneous electrical perturbations, which are superimposed on the transmitted signal and complicate its reception. At high intensity of interference reception becomes practically impossible.

Interference classification:

a) interference from neighboring radio transmitters (stations);

b) interference from industrial installations;

c) atmospheric disturbances (thunderstorms, precipitation);

d) interference caused by the passage of electromagnetic waves through the atmosphere: troposphere, ionosphere;

e) thermal and fractional noise in the elements of radio circuits due to the thermal motion of electrons.

In the mathematical description, interferences are random functions of time. A random function of a continuous time is usually called a random process, its discrete

analogue is called a random sequence. As a rule, interferences belong to the class of stationary random processes and are characterized both by their distributions and moments of distributions, and by their numerical parameters. The main distribution of most noise signals is normal (Gaussian process). This is explained by the fact that the distribution of the sums of independent random variables, which make up random interferences, converges to normal, regardless of the nature of the distribution of terms (Lyapunov's theorem).

The moment of the first order expresses the average value (constant component) of a random process: $M\{q\} = \overline{q} = \int_{-\infty}^{\infty} q \cdot p(q) dq$, where $p(q)$ – probability density of values q.

The central moment of the second order determines the variance of the process: $D\{q\} = \sigma^2 = \int_{-\infty}^{\infty} (q - \overline{q}) \cdot p(q) dq = \overline{q^2} - (\overline{q})^2$.

The variance expresses the power of the variable component of the process. The square root of the variance value, ie the value σ, is the root mean square value of the scatter of random values q relative to the average value $\overline{q}$.

The mixed moment of the second order is called the autocorrelation function of a random process $q(t)$: $M\{q(t)q(t+\tau)\} = \overline{q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 p(x_1, x_2) dx_1 dx_2 = B(\tau)$.

Value $B(\tau)$ when $\tau \to 0$ is equal to the total power of the random process $q(t)$.

In practice, most random processes have the property of ergodicity. It consists in the fact that the average values for the set of implementations (mathematical expectations calculated from the densities of distributions) coincide with the average values for the time T of one implementation of the process at $T \to \infty$. This allows you to estimate the numerical values of the noise parameters directly at arbitrary intervals [a, b] of setting signals: $\overline{q} = \lim_{T\to\infty} \frac{1}{T}\int_0^T q(t) dt \approx \frac{1}{b-a}\int_a^b q(t) dt$,

$\sigma^2 = \lim_{T\to\infty} \frac{1}{T}\int_0^T (q(t) - \overline{q})^2 dt \approx \frac{1}{b-a}\int_a^b (q(t) - \overline{q})^2 dt$,

$$B(\tau)=\lim_{T\to\infty}\frac{1}{T}\int_0^T q(t)q(t+\tau)dt\approx\frac{1}{b-a}\int_a^b q(t)q(t+\tau)dt.$$

The spectral power density of a random process (frequency distribution of noise and frequency noise) is related to the autocorrelation function of the Fourier transform. In one-sided (physical) representation of the spectrum: $B(f)=4\int_0^\infty B(\tau)\cos 2\pi f\tau d\tau$, $B(\tau)=\int_0^\infty B(f)\cos 2\pi f\tau d\tau$.

Additive interference with a uniform spectrum $B(f)=B_0=const$ is called white noise. The power of white noise in the frequency band 0 - F is proportional to the bandwidth $W_F=\int_0^F B(f)df=B_0F$.

Signal with additive interference is usually characterized not by the absolute power of the interference, and the ratio of the average signal power and interference, which is briefly called the signal-to-noise ratio: $\rho=W_c/W_q$.

The values of random processes are uncorrelated only at an unlimited frequency band. Any restriction of the frequency band makes a certain correlation in the process and only the values of the process can be considered independent of each other, which are spaced at least by a correlation interval τ_0: $\tau_0=\frac{2}{W_F}\int_0^\infty B(t)dt=1/2F$.

Noise immunity for optimal reception of continuous messages

The presence of interference makes it difficult to receive signals, at high intensity signal recognition can become almost impossible. The ability of the system to withstand the negative flow of interference is called interference immunity. The measure of the noise immunity of the optimal reception of continuous messages is the standard deviation or the mean square of the error.

Let $\varepsilon(t)$ be the noise at the output of the receiver. $M\left\{\varepsilon^2(t)\right\}=P_\varepsilon$ is the power of the interference. Primary signal (message) power $b(t)$ is equal to $P_b=M\left[b^2(t)\right]$. Instead of the standard deviation, another indicator is often used - the ratio of signal power to noise power (at the output of the receiver): $\rho_{output}=P_b/P_\varepsilon$. If the signal-to-noise ratio at the receiver input is equal to $\rho_{input}=P_c/P_n$, then the relationship

$g = \frac{\rho_{output}}{\rho_{input}} = \frac{P_b}{P_\varepsilon} \Big/ \frac{P_c}{P_n} = \frac{P_b}{P_c} \Big/ \frac{P_\varepsilon}{P_n}$ characterizes the gain (or loss) in noise immunity due to the use of a particular method of signal processing.

A useful indicator is also a generalized gain equal to: $g' = \frac{\rho'_{output}}{\rho'_{input}} = \frac{g}{\alpha}$, where $\rho'_{output} = \frac{P_b}{P_\varepsilon / F_c}$, F_c – message spectrum width, $\rho'_{input} = \frac{P_c}{P_n / F}$, F – signal spectrum width, $\alpha = F / F_c$.

Potential noise immunity of signals with amplitude, frequency and phase modulation

Signals with discrete amplitude modulation (AM), frequency modulation (FM) and phase modulation (FM) are usually used to transmit binary code elements (0 and 1). In the process of transmission, the elements of the code are distorted by interference, and there are errors of two kinds.

1. When transmitting element 0 can be accepted element 1, the probability of such an event (transition 0 → 1) is denoted by p (1/0).

2. When transmitting element 1, element 0 can be accepted, the probability of such an event (transition 1 → 0) is denoted by p (0/1).

The average probability of error is determined by the formula: $P = P(0)\,p(1/0) + P(1)\,p(0/1)$. In what follows, we assume that the a priori probabilities of transmission of code elements are equal, ie P(0) = P(1) = 0.5, with: $P = 0{,}5\left[\, p(1/0) + p(0/1)\right]$. The interference in the communication channel will be considered fluctuational with the normal law of distribution of instantaneous values: $W(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{\xi^2}{\sigma^2}}$.

Noise immunity of AM, FM, PM signals under the above conditions can be determined by calculating the average probability of error in this way.

Amplitude modulation

The elements of the signal AM are the parcel (code element "1") and the pause (code element "0"): $\left.\begin{array}{l} S_1(t) = a \cdot \sin_0 t \\ S_2(t) = 0 \end{array}\right\} 0 \le t \le T$, where T is the duration of the signal element.

The reception of the AM signal in the case of incoherent reception is carried out by comparing the signal level after the amplitude detector (bypass detector) with some threshold level U_t. Errors occur in the following cases:

1. When transmitting a parcel envelope the amount of signal and interference (E_{cn}) is lower than the threshold level U_n (transition 1→0).

2. When transmitting a pause bypass noise E_n turns out to be bigger than U_n (transition 0→1).

The probability of the first event is equal to $\left.\begin{array}{l} p(0/1) = \int_0^{U_n} \omega(E_{cn})\, dE_{cn} \\ p(1/0) = \int_{U_n}^{\infty} \omega(E_n)\, dE_{cn} \end{array}\right\}$,

probability of the second $W(E_{cn}) = \frac{E_{cn}}{\sigma^2} \exp\left[\frac{-E_{cn}^2 + a^2}{2\sigma^2}\right] I_0\left[\frac{E_{cn} a}{\sigma^2}\right]$, where $W(E_{cn})$ – the distribution density of the envelope sum of the signal and the interference, which is known to be determined by the generalized Rayleigh law. The average probability of error is equal $P = 0{,}5\left[\int_0^{U_n} \omega(E_{cn})\, dE_{cn} + \int_{U_n}^{\infty} \omega(E_n)\, dE_n\right]$. Value P depends on the threshold level U_t. It can be shown that the probability of error is minimal when $U_n = \frac{1}{2}a$ (if $a^2 \gg \sigma^2$). After calculating the integral we obtain $P = 0{,}5\left(0{,}5\left[1 - \Phi\left(\frac{h}{\sqrt{2}}\right)\right] + \exp\left(-\frac{h^2}{4}\right)\right)$, where $h^2 = \frac{a^2}{2\sigma^2}$ – is a signal-to-noise ratio; $\Phi(z) = \frac{2}{\sqrt{2\pi}} \int_0^Z e^{\frac{-X^2}{2}} dx$ – tabulated probability integral. If $h^2 \gg 1$, then $P \approx 0{,}5 \exp\left(-\frac{h^2}{4}\right)$.

The maximum noise immunity when receiving AM signals is observed if the optimal signal filtering is applied in front of the detector. This provides a maximum signal-to-noise ratio equal to h^2.

Frequency modulation

The signal elements at the FM are $\left.\begin{array}{l} S_1(t) = a \cdot \sin \omega_1 t \\ S_2(t) = a \cdot \sin \omega_2 t \end{array}\right\} 0 \le t \le T$. In the receiver, the signals are separated by filters tuned to the frequency ω_1 and ω_2 with subsequent detection.

At incoherent reception of signals of the FM in one of filters the sum of a signal and noise is always present, and in the second only noise. The error in registering the signal, obviously, will be in the case when the bypass noise in the filter without a signal exceeds the bypass amount of the signal and the noise in the filter with a signal.

Given that the signal powers and interference in each of the filters are the same (symmetric channel), the distortion probabilities "1" and "0" will be the same, ie p (0/1) = p (1/0). The probability that the bypass noise in the filter will be greater than the bypass sum of the signal and the noise in the second filter is equal to $p(E_n > E_{cn}) = \int_{E_{cn}}^{\infty} W(E_n) dE_n$.

The envelope sum of the signal and the interference is a random variable that has a generalized Rayleigh distribution law. Therefore, to determine the probability of error, it is necessary to average the probability $p(E_n > E_{cn})$ for all values E_{cn}:

$$p(0/1) = p(1/0) = \int_0^{\infty} W(E_{cn}) p(E_n > E_{cn}) dE_{cn} = \int_0^{\infty} W(E_{cn}) \left[\int_{E_{cn}}^{\infty} W(E_n) dE_n \right] dE_{cn}.$$

Substituting instead of $W(E_{cn})$ and $W(E_n)$ their values and calculating the integral, we obtain $p(0/1) = p(1/0) = \frac{1}{2} e^{-\frac{h^2}{2}}$, where h^2 – signal-to-noise ratio at the output of the filter with the signal.

The average probability of error is: $P = 0{,}5\left[p(0/1) + p(1/0) \right] = \frac{1}{2} e^{-\frac{h^2}{2}}$.

Maximum noise immunity is achieved when receiving FM signals in the case when the optimal signal filtering. With $h^2 = h_0^2$.

Phase modulation

The signal elements in PM are: $\left.\begin{array}{l} S_1(t) = a \cdot \sin \omega_1 t \\ S_2(t) = -a \cdot \sin \omega_1 t \end{array}\right\} 0 \le t \le T$.

Reception of phase modulation signals is possible only with the help of a synchronous (coherent) detector, which distinguishes the phases of the signals.

The probabilities of transitions p (0/1) and p (1/0) in case of fluctuation noise in the communication channel are the same and equal:

$$p(0/1) = p(1/0) = \frac{1}{\sqrt{2\pi}\sigma} \int_a^{\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx = 0{,}5\left[1 - \Phi\left(\sqrt{2}h\right)\right].$$

Average probability of error $P = 0{,}5\left[p(0/1) + p(1/0)\right] = 0{,}5\left[1 - \Phi\left(\sqrt{2}h\right)\right]$.

The maximum noise immunity of the PM signal is achieved with optimal signal filtering when $h^2 = h_0^2$.

Lecture №16. Fundamentals of construction of simple FPGA structures. Programmable logic arrays (PLA). Programmable matrix logic (PML). Programmable macrology chips. Basic matrix crystals (BMC)

Fundamentals of construction of simple FPGA structures

The considered various digital devices belong to standard integrated circuits of low level of integration. They are very widely used even at the current level of development of microprocessor technology, are made on the basis of simple technologies, and therefore have a low cost, which, again, pushes the developers of digital equipment to use them.

But, along with the standard elements, in any digital device there are non-standard modules, without the use of which it is often impossible to do, but their production on standard elements is either impossible or too expensive, as it leads to the use of a large number of cases. , printed circuit boards, increasing time delays, etc. The manufacture of specialized IP to order leads to significant costs and time for their development and manufacture. It is these problems and gave impetus to the creation of integrated circuits with programmable and reprogrammable structures designed to build non-standard modules of the developed electronic system. Despite the fact that such structures can have a significant number of logical elements, their structure and high seriality will be more compelling arguments in this regard.

The use of programmable logic integrated circuits opens up boundless space for the developer of electronic equipment. Based on them, it is possible to create functional equipment of varying complexity. For example, recently the leading FPGA manufacturers have started to advertise their products as a full-fledged replacement for classic digital signal processors.

PLA (programmable logic arrays)

Programmable Logic Arrays (PLAs) have historically been the first of programmable integrated circuits. Examples of PLM can be domestic chips K556RT1, K556RT2, K556RT21.

Their generalized structure is shown in Fig. 16.1.

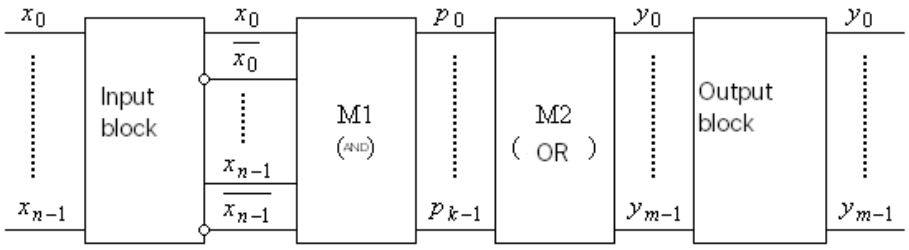
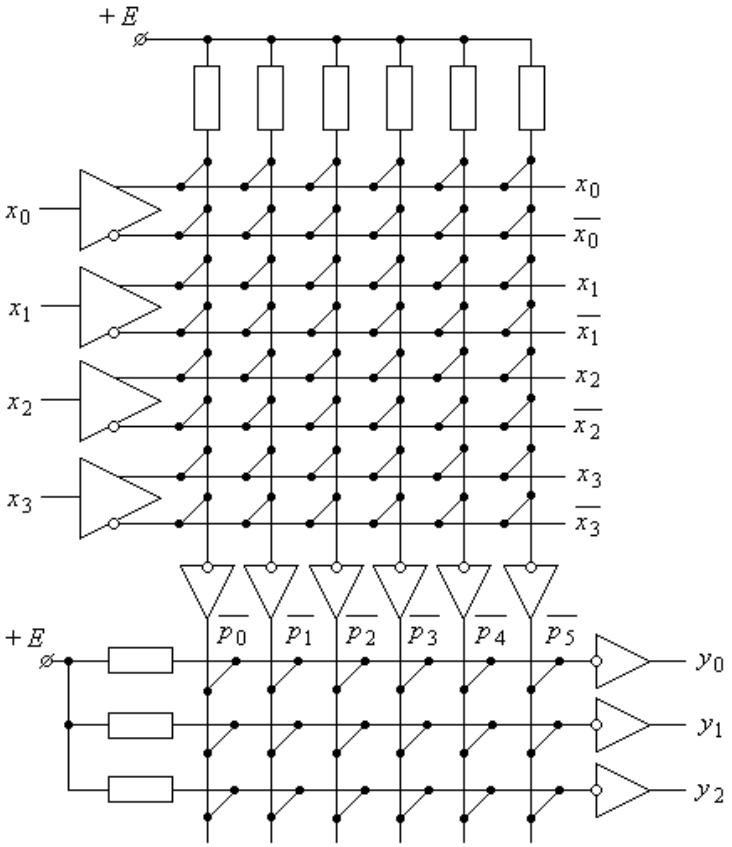


Fig. 16.1

The matrix consists of series-connected input unit designed to form paraphase groups of input signals of the required power; matrix M_1 of high-input elements I; matrix M_2 elements *OR* and the output unit, which provides the necessary characteristics of the output signal directly to the output (the presence of the Z-state, the open collector, etc.).

The main parameters of the PLM are the number of inputs n, the number of minterms (outputs of the matrix M_1) k and the number of outputs m. In reality, inequality always holds: $2^n >> k$, therefore, the number of minterms determines the complexity of the logical functions implemented.



a

b *c*

Fig. 16.2

In fig. 16.2, **a** is an example of a schematic diagram with n = 4 and m = 3 for TTL, and in Fig. 16.2, **b, c** are diagrams of the respective switching nodes, consisting of a diode and a fusible jumper, similar to RPZP.

When using CMON technology, the diodes are replaced by field-effect transistors. The scheme of the matrix (Fig. 16.2, a) remains unchanged, and the schemes of switching nodes provide the implementation of logical operations **AND-NO** and **OR-NO**, which does not affect the final result.

In the practice of designing digital devices on PLA, a simplified image of their circuitry is used. An example of such an image for *n = 4; k = 6; m = 3* is shown in Fig. 16.3. The points set at the intersection of rows and columns of the matrix characterize the corresponding connections.

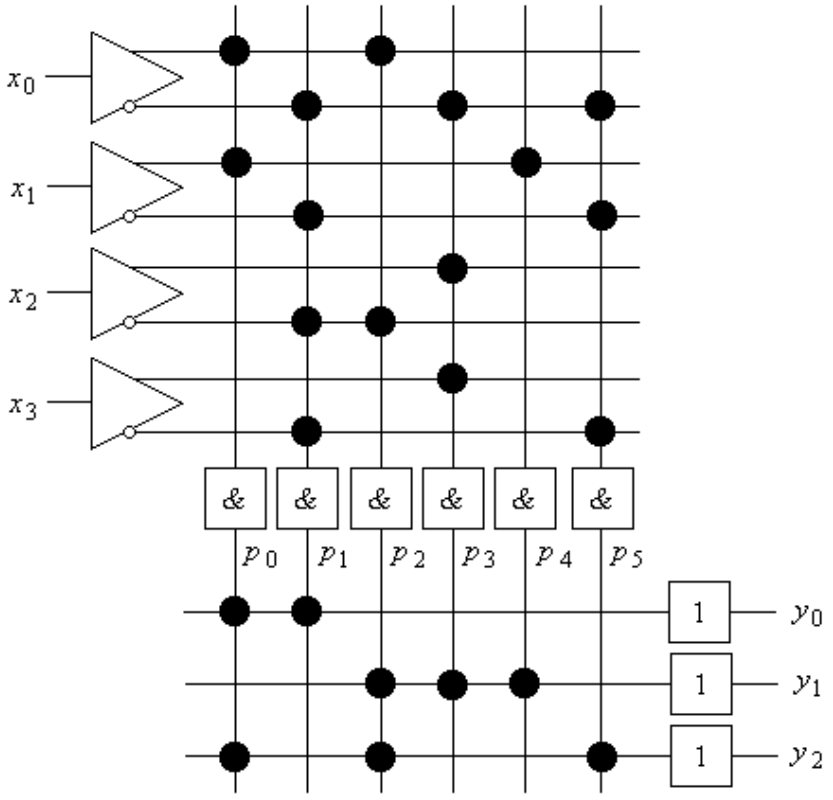


Fig. 16.3

From shown in Fig. 16.3 schemes of the matrix can write down the functions that are implemented using it:

$$y_0 = p_0 + p_1;$$
$$y_1 = p_2 + p_3 + p_4;$$
$$y_2 = p_0 + p_2 + p_5;$$

where

$$p_0 = x_0 x_1;$$
$$p_1 = \overline{x_0 x_1 x_2 x_3};$$
$$p_2 = x_0 \overline{x_2};$$
$$p_3 = \overline{x_0} x_2 x_3;$$
$$p_4 = x_1;$$
$$p_5 = \overline{x_0 x_1 x_3}.$$

One of the features that distinguishes PLA from PZP is the possibility of using a kind of feedback, ie connecting the outputs of PLA with its inputs, which allows you to implement more complex logical functional dependencies. An example of such a scheme is shown in Fig. 16.4, of which we have:

$$y_0 = x_0 \cdot x_1 + \overline{x_0} \cdot \overline{x_1};$$
$$y_1 = x_0 \cdot x_1 + y_0 \cdot x_2 = x_0 \cdot x_1 + x_2 \cdot \left(x_0 \cdot x_1 + \overline{x_0} \cdot \overline{x_1} \right).$$

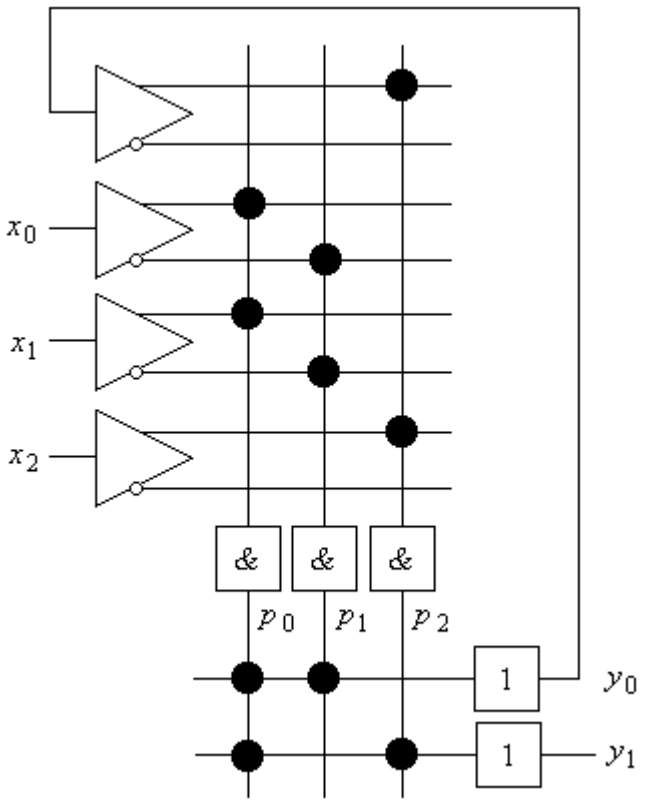


Fig. 16.4

The disadvantage of the PLA architecture was the weak use of resources of the programmable matrix OR, so the further development of chips built on the architecture of programmable matrix logic (PML).

PAL (programmable array logic)

Programmable Array Logic (PAL) is a simplified version of PLA and is characterized by having a programmable matrix of elements **AND** and a fixed matrix **OR**. The emergence of these devices is due to the fact that the vast majority of applications use simple logic functions that can be implemented on a finite number of **OR** elements. PAL eliminates the need to program the **OR** matrix, as a result of which its use is significantly simplified.

An example of the organization of PAL is presented in fig. 16.5.

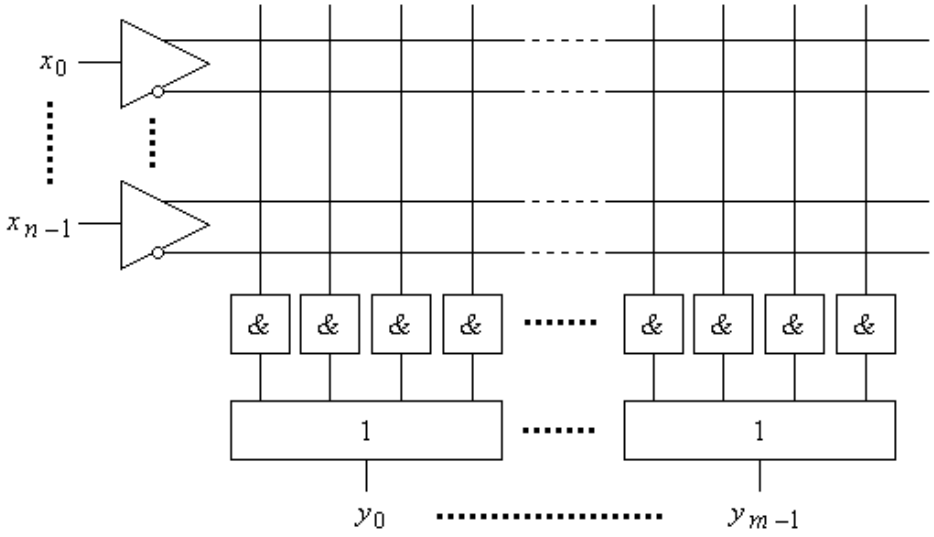


Fig. 16.5

Examples of PAL of this type are chips PAL16L8, PAL22V10 TTL, as well as GAL16V8-5 and PALCE16V8-5 KMON technologies. Among the chips of the CHD countries, the chips of the KM1556 series (HP4, HP6, HP8, CL8) are similar. A variant of this class are FPGAs that have only one programmable matrix AND - for example, the chip 85C508 from Intel.

PAL chips have been widely used in the implementation of simple logic functions and have been intensively improved in various directions. There were PAL with a programmable output buffer, which made it possible to obtain both direct and

inverse values of functions; chips with bidirectional pins that could be used as inputs and outputs; chips with memory elements based on synchronous D-flip-flops or whole registers. All this significantly expands the scope of programmable matrix logic, opened up the possibility of building synchronous digital automata.

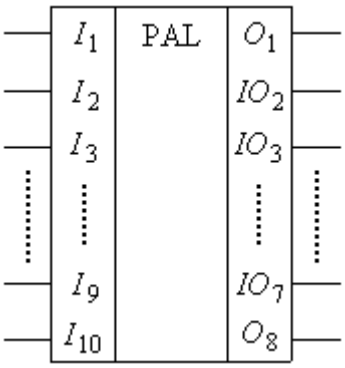


Fig. 16.6

In Fig. 16.6 a conditional image of PAL PAL16L8 is shown - a chip that until recently was widely used in the construction of specialized digital devices. Its programmable matrix I has 64 rows and 32 columns and used for programming 64 × 32 = 2048 burnt jumpers. Each of the 64 elements of the matrix AND has 32 inputs, covering 16 input variables and their negation; 10 of them join the entrances $I_1 \div I_{10}$, and the remaining 6 have the opportunity to join the exits $IO_2 \div IO_7$, which can be programmed as inputs and outputs, and also have a Z-state.

Programmable macrology chips

One of the directions of PAL construction is FPGA, which have only one programmable matrix I or I-NO (OR-NO), but due to the large number of inverse feedbacks on their basis it is possible to build complex logical functions. This class includes chips PLHS501, PLHS502 company SIGNETICS, which are based on the matrix AND-NO, as well as chip XL78C800 company EXEL, built on the basis of the matrix OR-NO.

Since FPGAs at the level of integration are characterized by the number of elements AND-NO, the above chips are characterized as FPGAs of low level of integration with an average number of LE up to 1500… 2000.

In general, they are obsolete not only in terms of their manufacturing technologies, but also in terms of programming.

BMC (basic matrix crystals)

Another area of FPGA development was the basic matrix crystals (BMC), designed to implement non-standard computer devices without the use of integrated circuits of low and medium level of integration, focusing on large integrated circuits (Gate Arrays). The reasons that led to the creation of the BIS for this purpose are as follows. Costs for the development of BIS, as well as extra large IS (EBIS), are very large and are estimated at hundreds of millions of dollars, so it is irrational to develop specialized BIS. The solution to the problem was found in the development of EBIS, the operation of which was adjusted to specific tasks only in the final stages of their manufacture. All this in general has significantly reduced their cost, time for design and manufacture and made it possible to use BIS and EBIS to build unique or specialized devices.

Manufacturers of SND countries produce BMK with the number of equivalent LE up to 50 thousand, with a delay of each of them 20 ns.

Lecture 17. Modern FPGAs

The considered FPGA chips served as a basis for the development of two modern directions of BIS and EBIS with programmable and reprogrammed structures. The first direction continues the development of programmable matrix logic in the series CPLD (Complex Programmable Logic Devices), and the second direction develops the BMC in the series FPGA (Field Programmable Gate Array). An alternative to both directions was the mixed EBIS architecture called FLEX (Flexible Logic Element matriX). The growth of the level of integration made it possible to place on a single chip circuits, the complexity of which corresponds to entire electronic systems.

The current level of EBIS integration is so high that millions of logic elements are placed on the crystal. Real and functional knowledge of EBIS circuitry for digital equipment developers, especially low-skilled professionals, is not essential. Therefore, consider only those features that can provide useful information.

The programmability of EBIS is ensured by the presence in it of a large number of bipolar (keys), which with the help of software can be translated by the user into a state of high or low (zero) conductivity. This sets the required configuration of the circuit formed on the crystal. The number of programmable keys for modern FPGAs reaches several million.

Modern FPGAs use the following types of programmable keys:

- jumpers of antifuse type (thin dielectric punched jumpers);
- LISMON transistors with double gate;
- Key field-effect transistors controlled by configuration memory triggers.

Programming with antifuse jumpers is one-time. The jumpers are small and in the initial state pass very small currents (≈ 10-15 A). The programmable voltage pulse pierces the jumper and creates a conductive channel. The resistance of the channel depends quite precisely on the magnitude of the breakdown current, which makes it possible to create jumpers with a given value of resistance. The parameters of both jumper states can be stored for about 40 years.

Elements of EERROM (FLASH) on LISMON have been considered earlier.

An example of a key circuit controlled by a configuration memory trigger is shown in Fig. 17.1.

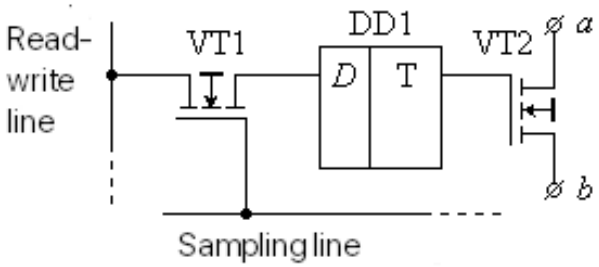


Fig. 17.1

The key VT2 provides switching of the section ab depending on the state of the trigger. The trigger, in turn, is set by a high potential on the sampling line and a corresponding signal (1 or 0) on the read-write line.

The use of such a key opens wide possibilities for the use of FPGAs, as their recording time is determined by tens to hundreds of milliseconds, and the time spent on erasure is not required at all.

BIS, in which the circuit is created using logical elements and programmable keys, are called VIS (VLSI) programmable logic.

User-programmable valve arrays (FPGAs) have a slightly different architecture. They are built on the basis of specialized blocks of identical configuration, which are used as:

- simple logical elements;
- logic modules based on multiplexes;
- logic modules based on ROM;

as well as specialized blocks such as I / O block, etc. The blocks have the ability to combine with each other, creating the necessary configuration of the circuit.

There are quite a number of FPGA architectures that can be of little use to digital device circuitry.

Complex programmable logic circuits (CPLD and FLEX) architecturally include a central switching matrix and a large number of functional blocks (macroelements), peripheral blocks. Their architecture and circuitry are quite complex and not essential

for digital equipment developers. The classic representatives of CPLD are MAX7000 chips from ALTERA.The VLSI market has a large number of CPLD and VLSI PLMs from ALTERA (MAX, FLEX, APEX family), ATMEL (ATF1500 family), VANTIS (MACN family), XILINX (XC9500 family), etc. Within each family there are a number of VLSI different in complexity, cost and other indicators. The software designed for their programming helps to choose the necessary chips.

The next state of development of FPGAs was VLSI, which is defined as "system on crystal" (SoC). The emergence of such VLSI is due to the growing level of integration and speed, resulting in the ability to create on the chip specialized computing modules and entire processors, RAM and RAM, peripherals, and specialized testing devices. Of course, such FPGAs have a large number of specialized units, but, at the same time, on the basis of such FPGAs it is possible to build specialized digital systems of fairly high capacity. Such FPGAs include the VLSI family of the AREX20K / KE type of the ALTERA company, as well as the VLIS family of the VIRTEX type of the XILINX company.

A significant difference between the latest generation of FPGAs is their significant approach to the user in terms of programming and testing. For this purpose, FPGAs are provided with a set of tools and operations that allow to provide testing of VNIS without physical access to each of its conclusions. The set of built-in hardware and software designed to solve test control problems is called the JTAG interface (Joint Test Action Group - the group that developed the test standard). JTAG testing is called Boundary Scan Testing (BST). Peripheral scanning allows you to check the operation of the chips, the mounting connections of the chips to each other on the PCB, read the signals on the pins of the chip during its operation or control these signals.

An important feature of the latest generation of FPGAs is the ability to provide reconfiguration (programming) in the system, which allows you to make changes in the logic of their work. The property of programmability directly in the system is denoted by the abbreviation ISP (In-System Programmable). The need to use this property arises when you need to update the system or when errors are detected.

Lecture 18. Basic parameters of FPGA. Basics of designing digital devices on FPGA in VHDL description language

The main parameters of FPGA

When choosing a FPGA manufacturer, family and type, the digital system developer should use the following criteria:

• logical capacity sufficient for project implementation, with the possibility of future upgrades;

• FPGA speed;

• circuit design and construction parameters;

• availability or cost of development tools, including hardware and software;

• availability of technical and methodological support;

• the cost of chips and the reliability of their acquisition channels.

Based on these criteria, for professionals who do not have sufficient experience with FPGA, we recommend starting work with ALTERA chips (www.altera.com). This company manufactures a wide range of FPGA CPLD families MAX3000, MAX7000, MAX9000, as well as FPGA FPGA families FLEX10K, FLEX6000, FLEX8000. An important advantage is that ALTERA provides users with a free version of the software - the basic package MAX + plus II BASELINE can be obtained on the CD "ALTERA Digital Library" or on the official website of the company.

FPGAs from ALTERA are made with the possibility of programming directly on the board. ByteBlaster and ByteBlasterMV download cable diagrams have been published for programming and downloading the device configuration.

To explain the parameters of FPGA, which the user has to deal with when designing digital devices, consider the features of the functional diagram of the widely used IC family MAX3000 (Fig. 18.1).

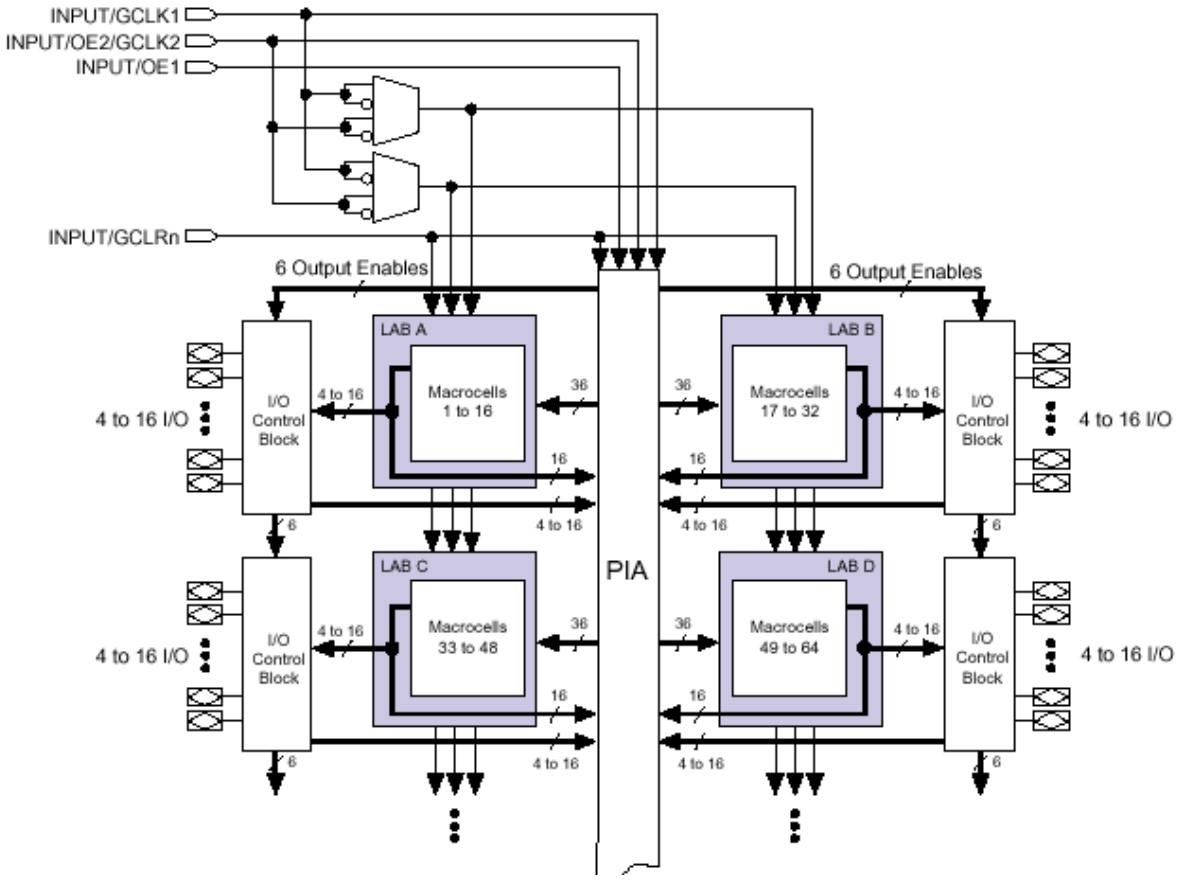


Fig. 18.1

The main elements of the functional diagram are:

• Logic Array Blocks (LAB);

• macroelements ME (Macrocells);

• logical expanders (Expanders): parallel (Parallel) and distributed (Shareble);

• Programmable Interconnected Array (PIA);

• I / O Control Block (I / O).

The great complexity of the matrix determines the allocation of a group of electrical circuits that cover all circuits. Such circuits are called global. Part of the outputs (Dedicated Inputs) is assigned to global circles. These are global circuits of synchronization, zeroing, installation in Z-state of each macroelement. However, these pins can be used as inputs or outputs of the user for "fast" signals processed by FPGA.

As can be seen from Fig. 18.1, the architecture of the FPGA family under consideration is based on logical blocks, which contain 16 macroelements each.

Logical blocks are connected using a programmable connection matrix. Each logic block has 36 inputs from PMZ.

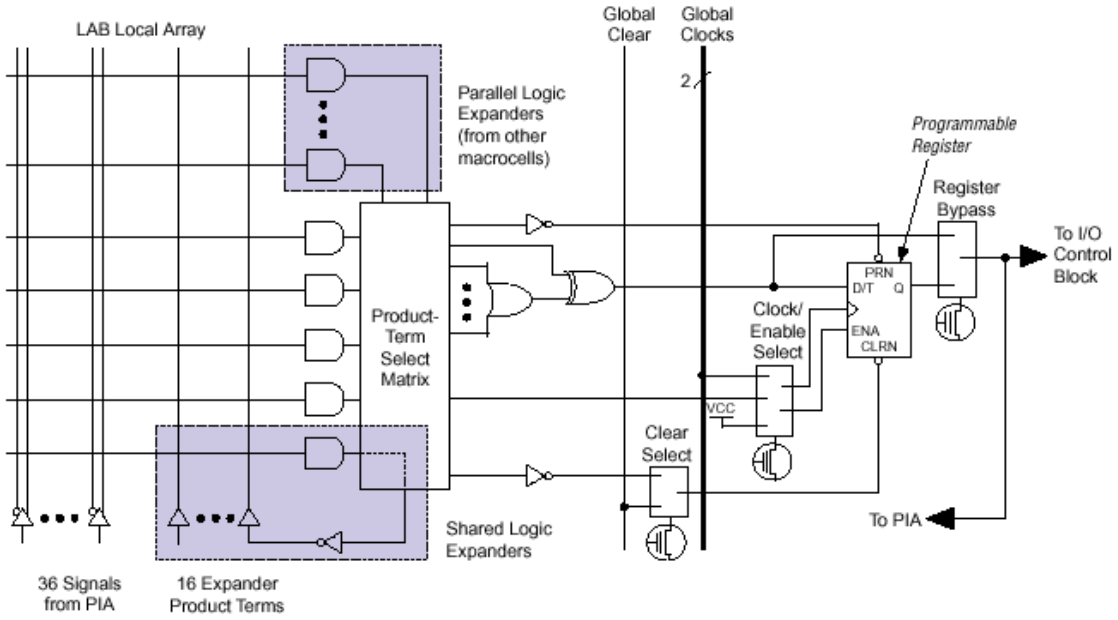


Fig. 18.2

In Fig. 18.2 shows a block diagram of the macroelement FPGA family MAX3000. It consists of three main nodes:

• LAB Local Array (LPM);

• Product-Term Select Matrix (MRI);

• Programmable Register (PR).

Combination functions are implemented on a local programmable matrix and a term distribution matrix, which allows you to combine logical products by **OR** or **EXC. OR (XOR)**. In addition, the matrix of the distribution of terms allows you to switch the control circuit of the trigger macronutrient.

The FPGA of the MAX3000 family has 2 global clock signals, which allows you to design circuits with two-phase synchronization.

In terms of FPGA capacity, they are characterized by the following parameters:

• logical capacity, ie the number of equivalent logical elements of type 2AND-NO;

• the number of macronutrients;

• number of logical blocks;

• the number of user-programmable outputs.

Table. 18.1 provides reference data on FPGAs from ALTERA of different families.

Table. 18.1

Family / Parameter	MAX3000 EPM3256A	FLEX600 EPF6024A	MAX9000 EPM9560	FLEX10K EPF10K250	APEX20K EP20K1000
Logical capacity	5000	24000	5000	250000	2670000
Number of macroelements	256	1960	560	12160	4224
Number of logical blocks	16	196	772	20	264
Number of pins to use	158	45	46	470	780

Logical expanders are used to implement logical functions with a large number of variables (see Fig. 18.3, Fig. 18.4).

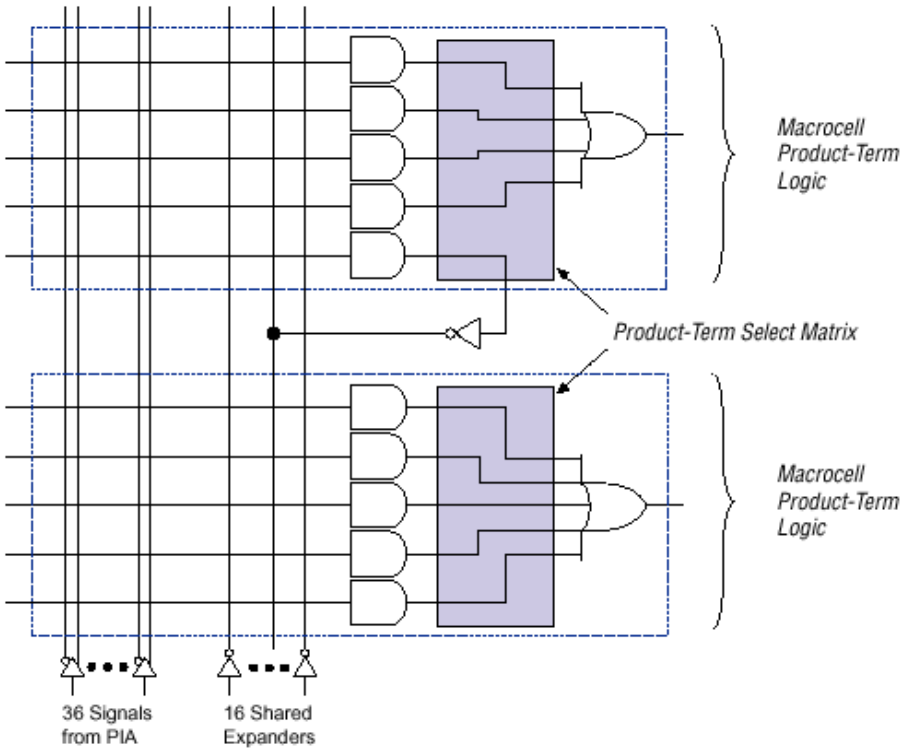


Fig. 18.3

Distributed logic expander (Fig. 18.3) allows you to implement logic functions with a large number of inputs, providing the ability to combine ME, which are part of one LB. In this way, the distributed expander forms a term, the inverse value of

which is transmitted by the term distribution matrix to the local programmable matrix and can be used in any IU of the given LB.

As can be seen from Fig. 18.3, there are 36 PMZ signals, as well as 16 inverse signals from distributed logic expanders, which allows within one LB to implement a logic function up to 52 terms of rank 1.

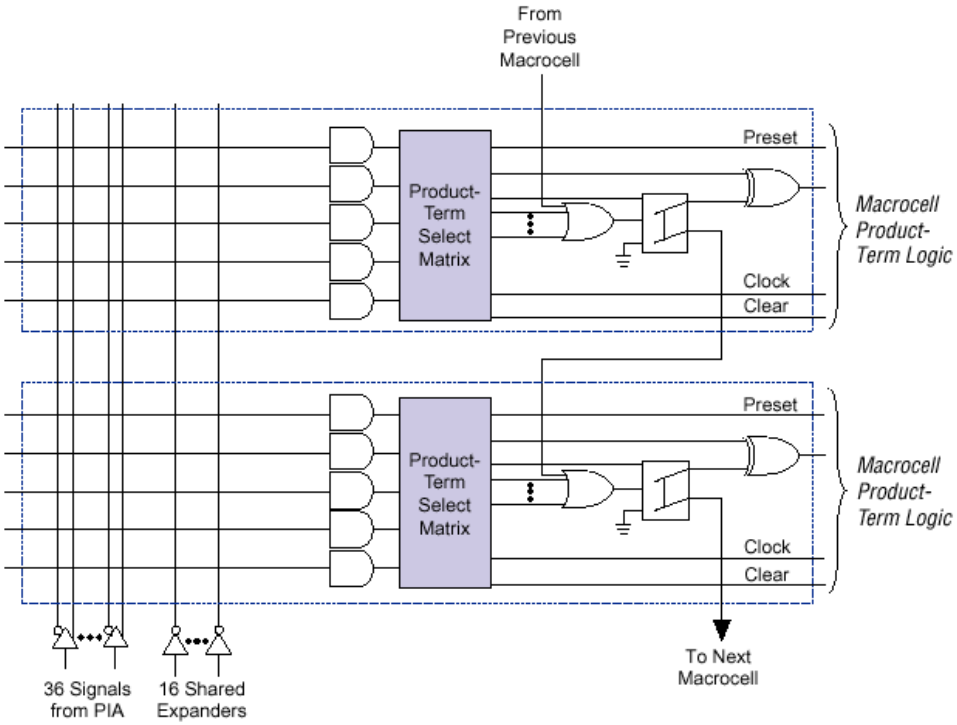


Fig. 18.4

The parallel logical expander (Fig. 18.4) allows the use of local matrices of adjacent ME to implement functions that include more than 5 terms. One link of parallel expanders can contain up to 4 IU, implementing the function of 20 terms.

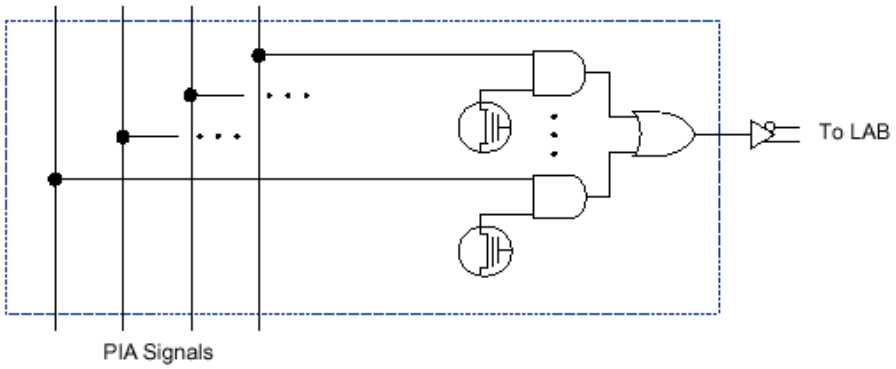


Fig. 18.5

The programmable matrix of connections (Fig. 18.5) is built so that it displays signals from all possible sources - EVB, feedback signals LB, specialized dedicated pins. Only the necessary connections for signal transmission are provided during programming.

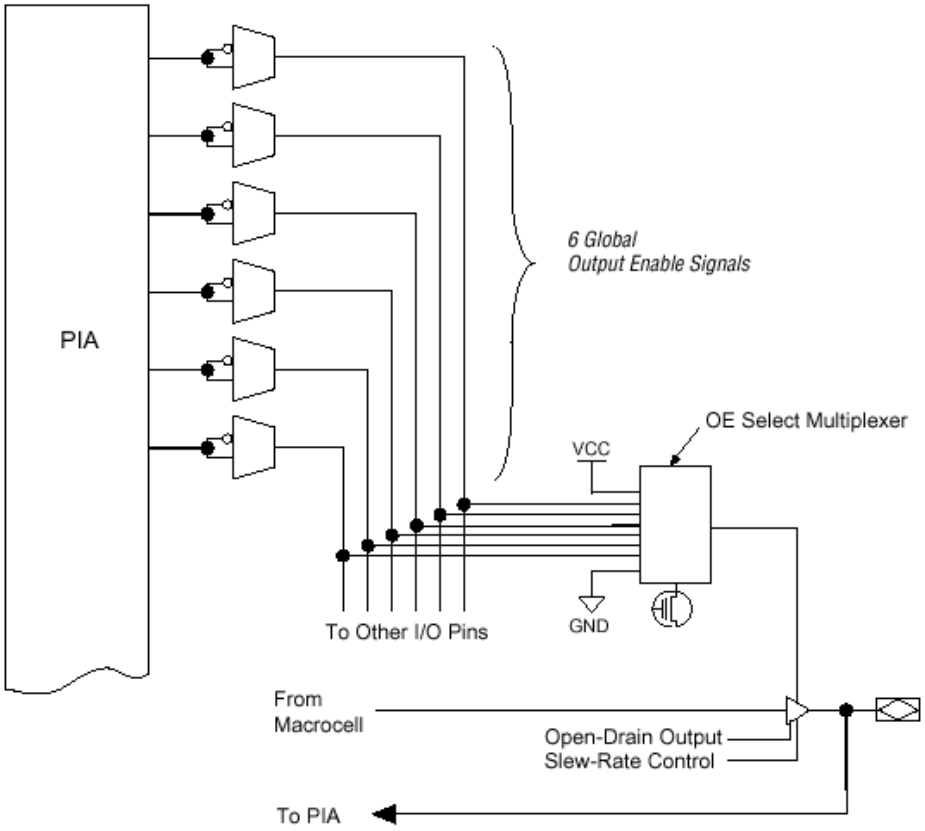


Fig. 18.6

FPGA I / O elements (Fig. 18.6) allow to provide modes of operation for input / output of information, as well as modes with open collector and Z-state.

The MAX3000 family chips fully support the JTAG standard. Programming is provided by the voltage increased in comparison with power supply (3,3 V) which is created by the specialized schemes of the voltage multipliers which are a part of FPGA.

The next group of FPGA parameters are time programming parameters. However, because FPGAs are programmed using CAD, their compilers fully provide the necessary time relationship between the signals and the user's knowledge is optional.

Delays in signal propagation are important for the user. These include:

• input / output signal propagation delay;

• delay of the global clock signal to the output;

• time of establishment of a global clock signal;

• maximum global clock frequency.

For FPGAs of the MAX3000 family, the time delays are in the range of 3… 5 ns, and the maximum clock frequency reaches up to 200 MHz.

Basics of designing digital devices on FPGA in VHDL description language

The development of modern integrated circuits is a complex engineering problem, the solution of which is impossible without the widespread use of computer-aided design systems. The creation of such systems is known to require a means of formally describing the structures and functions of design objects. VHDL (Very high speed integrated circuits Hardware Description Language) was developed in 1980 as a result of a project in the United States to create high-speed integrated circuits. In 1987, the Institute of Electrical and Electronics Engineers (IEEE) recognized this language as a US standard. VHDL is now the world's most widely used language for this purpose. It is used in many computer-aided design systems, the number of users of which is growing rapidly.

VHDL language is used to describe the model of a digital device (device, system). The description in VHDL defines the external connections of the device ("external view" or interface) and one or more "internal views" (see Fig. 18.7). Exterior view sets the interface of the device, a set of signals that are exchanged with the outside world. This type describes the abstract representation of the device "as a whole" and is denoted by the English term entity, which literally means "essence" and most accurately reflects the content of the performance. However, in the literature, the term "essence" is not widespread, to denote the external description of the object, the terms "object interface", "declarative part" and others are used. This guide will use the term "object interface" or simply "interface".

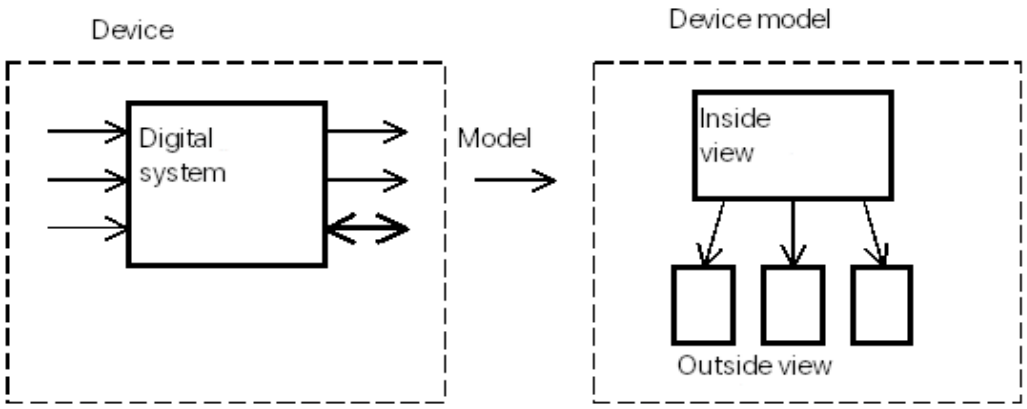


Fig. 18.7

The inside view determines the functionality of the device or its structure. The internal structure of an object is determined by the architecture.

As with programming languages, the VHDL language has its own rules, including rules for describing variable names, objects, data types, and other parameters.

Identifiers in VHDL

Identifiers are a sequence of letters and numbers of arbitrary length. Allowed characters are uppercase (A ... Z), lowercase (a ... z), numbers (0 ... 9), underscore. The first character must be a letter, and the last character cannot be an underscore. Lowercase and uppercase letters are considered identical, for example, Count, COUNT and CouNT are considered as one word. Underscores should not follow each other. Comments start with a double hyphen and follow to the end of the line, for example. The following identifiers are reserved for use as keywords and cannot be used otherwise:

Abs	Access	after	alias	All
And	architecture	array	begin	Block
Body	Buffer	case	component	Configu-ration
Constant	disconnect	downto	else	Elsif
End	Entity	file	for	function

generate	Generic	guarded	if	In
inout	Is	label	library	linkage
loop	Map	mod	nand	New
next	Nor	not	null	Of
on	Open	or	others	Out
package	Port	procedure	process	Range
record	Register	rem	select	severity
signal	Subtyupe	then	to	Transport
type	Units	until	use	vriable
wait	When	while	with	Xor

VHDL language objects

Objects are a storage area of a certain type of data. Objects are created using an object declaration, for example:

variable COUNT: INTEGER;

The result is an object named COUNT that stores an integer value. In addition, COUNT is declared as a variable class.

Data objects can be of three classes:

- Constant - can store a single value of a certain type. This value is determined by the object at the beginning of the simulation and cannot be changed during the simulation.

- Variable - an object of this class can store a separate value of a certain type, however, in the process of modeling it can be assigned different values. Variable assignment statements are used for this purpose.

- Signal - an object of this class has a previous value, has a current value and a set of subsequent values.

Objects of the signal class simulate wire connections in circuits, while variables and constants are used to simulate circuit behavior, they are similar to objects used in the C and Pascal programming languages.

Declaration of constants

The purpose of an object declaration is to name the object, declare its type, class, and even assign a value. Examples of constant declarations are described below:

constant RISE_TIME: TIME: = 10ns;

constant BUS_WIDTH: INTEGER: = 8:

In the first case, the object RISE_TIME is declared, which stores the value of the type TIME, the object at the beginning of the simulation is assigned a value of 10 nanoseconds. In the second case, it is declared that BUS_WIDTH (bus width) type INTEGER (integer) and it is assigned the value 8.

Declaration of variables

Examples of variable object declarations are as follows:

variable CTRL_STATUS: BIT_VECTOR (10 downto 0);

variable SUM: INTEGER range 0 to 100: = 10;

variable FOUND, DONE: BOOLEAN;

In the first case, the variable CTRL_STATUS is declared as an array of 11 elements, and each element of type BIT. In the second case, the variable SUM is declared as an integer in the range from 0 to 100, at the beginning of the simulation the variable is assigned a value of 10. If the variable is not set at the beginning of the simulation, the default value is used. They are the "far left" value in the set of values of this type. For example, a variable of type BOOLEAN has a set of values (FALSE, TRUE) and in the third example, the initial value of the variables FOUND and DONE will be taken by default, ie FALSE.

Declaration of signals

Declarations of objects of the signal class are similar to declarations of variables:

signal CLOCK: BIT;

signal DATA_BUS: BIT_VECTOR (0 to 7);

signal GATE_DELAY: TIME: = 10 ns;

The first example declares a CLOCK object of type BIT, the initial value in the simulation will be taken by default, ie 0 (set of values of type BIT: 0.1 and the leftmost value 0).

Not all objects in the VHDL language are created by declaration, for example, input / output ports are always considered objects of the signal class.

Data types

Each object in VHDL can store values that belong to a specific set of values. This set of values is declared using a type declaration. A type is a name that is associated with a specific set of values and a set of operations. For example, BOOLEAN has a set of values FALSE, TRUE and a set of operators: and, or, nor, nand, not. In the VHDL language it is possible to create new types using declarations and task set tasks.

All possible types in VHDL fall into four major categories:

- Scalar , which in turn are divided into four types:

-Enumeration,

-Integer,

-Physical,

-Floating point.

- Composite - they consist of elements of the same type (arrays) or different types (records),

- Access type - provide access to this type through pointers,

- File types - provides access to objects that contain sequences of values of this type.

Enumeration

At the declaration the user defines a set of certain values, for example:

type MVL is (‘U’, ‘0 ‘, ‘1’, ‘Z);

type MICRO_OP is (LOAD, STORE, ADD, SUB, MUL, DIV);

MVL – enumerated type with an ordered set of values: ‘U’, ‘0 ‘, ‘1’, i ‘Z’. MICRO_OP has a set of values: LOAD, STORE, ADD, SUB, MUL, DIV. The order of writing values in the declaration determines the vocabulary, ie the value on the right is always greater than the value on the left: STORE <DIV is true, SUB> MUL

is false. Values in the listed type have a position number. The position number of the leftmost element is 0. Values in the enumerated type are called enumeration literals (literals of the enumerated type). For example, in the declaration:

type CAR_STATE is (STOP, SLOW, MEDIUM, FAST);

literals are STOP, SLOW, MEDIUM, FAST and only they can be assigned to a variable CAR_STATE.

Integer type

Integer – specifies the type whose set of values is in a given integer range, for example:

type INDEX is range 0 to 15;

type WORD_LENGTH is range 31 downto 0;

subtype DATA_WORD is WORD_LENGTH range 15 downto 0;

type MY_WORD is range 4 to 6;

INDEX – it is a variable of integer type, the set of values of which is in the range from 0 to 15. DATA_WORD – subtype WORLD_LENGTH, the set of values is in the range from 15 to 0. In contrast to the listed type in the integer type, the position number is equal to the value, for example, for the value 31 of the variable WORD_LENGTH position is 31.

Floating point type

The floating point type has a set of values in a given range of real numbers, for example:

type TTL_VOLTAGE is range 1.4 to 5.5

type REAL_DATA is range 0.0 to 31.9;

Letters of the floating point type differ from integers by the presence of a dot (.). As a result, 0 is an integer literal, and 0.0 is a floating-point literal.

Physical type

Physical type stores values that represent the results of measurements of physical quantities: time, length, voltage, current, and so on. Values of this type are expressed as integers multiplied by the base unit, for example:

type CURRENT is range 0 to 1 E9

units

nA; - (base unit) nano-ampere

uA = 1000 nA; micro-ampere

mA = 1000 uA; milli-ampere

Amp = 1000 mA; ampere

end units;

subtype FILTER_CURRENT is CURRENT range 10 mA to 5 mA;

Тут CURRENT is defined as a physical type that has a value in the range of 0 nA to 10^9 nA. The base unit is the nanoampere, and all others are derivatives. The position number of the value is equal to the number of base units represented by this value, for example, 2 mA have a position number of 2000, while 100 nA occupy a position of 100. A physical literal is written as a whole, followed by the name of the unit of measurement (space is required).

Array type

An array object consists of elements of the same type. The following are examples of arrays:

type ADDRESS_WORD is array (0 to 63) of BIT;

type DATA_WORD is array (7 downto 0) of MVL;

type ROM is array (0 to 125) of DATA_WORD;

ADDRESS_WORD – one-dimensional array of 64 elements of type BIT. DATA_WORD – one-dimensional array of 8 elements of the MVL type. ROM – an array of 126 elements of type DATA_WORD, ie in this case we are dealing with an array of arrays.

Array elements are accessed using indexes, for example, ADDRESS_WORD (26) refers to the 27th element of the array ADDRESS_WORD.

VHDL language operations

In the VHDL language there are operations of the following categories:

1. Logical operations
2. Relationship operations
3. Addition / subtraction operations
4. Multiplication / division operations
5. Others

The priority of operations increases from category 1 to category 5. Operations of one category have the same priority and are performed in sequence from left to right. Brackets are used to change the execution sequence.

Logical operations

There are six logical operations: and, or, nand, nor, xor, not. Operations are applied to types BIT, BOOLEAN, to one-dimensional arrays of BIT and BOOLEAN. When performing operations, the bit values "0" and "1" are interpreted as Boolean FALSE and TRUE. The result is the same type as the operands. Operation not is a unary operation, it has a priority of category 5.

Relationship operations

The VHDL language has the following relation operations: =, / =, <, <=,>,> =.

The result of any relation operation is a Boolean expression BOOLEAN. Equality (=) and inequality (/ =) operations apply to all types except the "file" type. The rest of the operations are applied to scalar types (integer, enumerated) or over arrays of discrete type. When the operands are arrays, the comparison is performed from left to right by one element, for example in the following comparison:

BIT_VECTOR ‘(0 "," 1 "," 1 ") <BIT_VECTOR’ (‘1 ‘, ‘0’, ‘1 ‘)

the result is TRUE, because the first element of the vector on the left is smaller than the first element of the vector on the right. Another example, if the type is declared:

type MVL is (‘U’, ‘0 ‘, ‘1’, ‘Z’);

then the result of the comparison:

MVL ‘(‘ U ‘) <MVL’ (‘Z’)

will be TRUE because ‘U’ is to the left of ‘Z’.

Operations of addition, subtraction, concatenation

Operations have designations: +, -, &.

The operands involved in the operations of addition (+) and subtraction (-) must be of the same numeric type, the result is of the same type. Operands in concatenation (&) can be either individual elements or one-dimensional arrays. The result is issued in the form of an array, for example when performing concatenation:

«0» & «1»

an array of characters "01" is formed, or:

‘C’ & ‘A’ & ‘T’

gives "CAT".

Operations of multiplication, divisions

This group of operations includes: *, /, mod, rem. The multiplication (*) and division (/) operands must be either an integer or a floating point type. The result has always the same type as operands. The multiplication operation can have one operand of physical type, and another - either integer or natural type. The result is issued in the form of a physical type. In a division operation, it is permissible to divide a physical object into an integer or natural object. The result is always physical. The division of the physical type into physical gives an integer result.

The operations of finding the remainder of division (rem) and division by modulus (mod) as operands can have an integer type and the result of an integer type.

The result of rem has the sign of the first operand and is defined as follows:

A rem B = A - (A / B) * B

The result of mod has the sign of the second operand and is defined as follows:

A mod B = A - B * N,

where N is some integer.

Other operations

These include: Abs, ** and others. The operation of finding the absolute value (abs) is compatible with any numeric type of operand. The operation of elevation (**) by the operand on the left has an integer or floating point type, and as a right operand (degree) - only an integer type.

Sequential statements

Sequential statements include assignment statements, if-statements and case statements, loop statements, and more.

Assignment operator

These operators are divided into variable assignment operators (denoted by: =) and signal assignment operators (<=). In both cases, the value of the expression to the right of the equals sign is calculated first, and then the resulting value is assigned to the variable or signal to the left of the sign. For example, the operator

abar: = not a;

sets a new value for the variable abar, namely the inverse value of a.

Operator

z <= not (a and b);

sets a new value for the signal z, which is to the right of the sign <=.

The if operator

In the general case, if is a sequence of expressions that determine the conditions. Any expressions whose calculation is a Boolean value (FALSE and TRUE) are used as conditions.

if boolean_expression then

sequential_statement

[elsif boolean_expression then

sequential_statement]

[else

sequential_statement]

end if;

The if expression is computed by looking at the conditions one by one until a true one is found. The set of sequential expressions associated with this condition is then calculated. Conditions of the form elsif in the expression if can be from 0 and more. The condition else can also be used. If expressions can be nested without restrictions.

Consider a simple example

if CTRL = ‘1 ‘then

MUX_OUT <= "10";

else

MUX_OUT <= "01";

end if;

If the control signal CTRL is equal to "1", the output signal MUX_OUT takes the value "10", otherwise MUX_OUT will take the value "01". This completes the if statement.

Consider a more complex example

if CTRLI = ‘1 ‘then

```
if CTRL2 = 0 "then
MUX_OUT <= "0010";
else
MUX_OUT <= "0001";
end if;
else
if CTRL2 = "0" then
MUX_OUT <= "1000";
else
MUX_OUT <= "0100";
end if;
end if;
```

If the control signal CTRL1 is equal to "1", then provided (opens the nested if) CTRL2 = 0 output signal MUX_OUT <= "0010", otherwise (ie for any other values of the signal CTRL2) MUX_OUT <= "0001". This ends the internal (nested) if statement. Otherwise (ie for any other values of the CTRL1 signal), if (opens a new nested if statement) CTRL2 = "0", then MUX_OUT <= "1000", otherwise (ie for any other values of the CTRL2 signal) MUX_OUT <= "1000". This is where the internal (nested) if statement ends, as well as the external if statement.

Case operator

The case operator has the following format:

```
case expression is
when option =>  sequential_statement
when option =>  sequential_statement
- any number of choices.
when others => sequential_statement]
end case;
```

When calculating the expression case, one of the branches is selected in accordance with the value of the expression. The expression can have the value of the listed type or the type of one-dimensional array. Consider an example:

```
type WEEK_DAY is (MON, TUE, WED, THU, FRI, SAT, SUN);
```

type DOLLARS is range 0 to 10;

variable DAY: WEEK_DAY;

variable POCKET_MONEY: DOLLARS;

case DAY is

when TUE => POCKET_MONEY: = 6; - branch 1

when MON I WED => POCKET_MONEY: = 2; - branch 2

when FRI to SUN => POCKET_MONEY: = 7; - branch 3

when others => POCKET_MONEY: = 0; - branch 4

end case;

The variable WEEK_DAY has the value of the listed type (days of the week). The DOLLARS variable has integer values ranging from 0 to 10. Branch 1 is selected when the day of the week is TUE. Branch 2 is selected in the case when the days of the week are MON or WED (vertical bar means OR). Branch 3 covers the values from FRI to SUN, ie FRI, SAT and SUN. Branch 4 covers all the remaining values, ie THU.

Loop operator

The loop operator is used to specify an iteration of a set of sequential expressions.

[Shortcut for loop:] iterative loop scheme

consecutive_expressions

end loop [Shortcut for loop];

There are three iterative schemes. The first has the form:

for id in range

Example of using the scheme:

FACTORIAL: = 1;

for NUMBER in 2 to N loop

FACTORIAL: = FACTORIAL * NUMBER;

end loop;

The body of the loop is executed N-1 times, with the ID NUMBER at the end of each iteration increases by 1. It is assumed that the identifier of the integer type and its values are in the range from 2 to N.

The second iterative scheme is as follows:

while boolean_expression

Example of using the scheme:

J: = 0; SUM: = 10;

WH-LOOP: while J <20 loop - loop має ярлик WH_LOOP

SUM: = SUM * 2;

J: = J +3;

end loop;

The expressions in the loop body are executed one after the other and this sequence is repeated until the condition J <20 is true.

The third is a design in which the iterative scheme is not specified and the exit from the loop is carried out using the expressions: exit, next or return, for example:

SUM: = 1; J: = 0;

L2: loop - loop has a label

J: = J +21;

SUM: = SUM * 10;

exit when SUM> 100;

end loop L2;

In this example, the exit expression causes the L2 loop to exit when the SUM becomes greater than 100. In the absence of an exit loop, it will run indefinitely.

Concurrent statements

Concurrent statements include: process operator, procedure procedure call operator, component specification operator, generate operator, and others. Parallel operators determine the parallel behavior of circuits, the order of their execution does not depend on their appearance inside the block.

In General, the process operator can be written as follows:

[process name:] [postponed] process [(list)]

section of declarations

begin

operators

end process [process name];

The process name and keyword [postponed] are optional and often missing. The list after the keyword process, although optional, is often used in practice to indicate triggers. The keyword may be preceded by various types of type declarations, attribute constants, etc.

Although the process is a parallel operator, it can contain sequential operators. Signals cannot be declared inside the process.

Device interface description

The entity interface declaration is used to describe the device interface in VHDL. It specifies the description name, interface port names, transmission direction, port type. A port is a signal line (bus) through which a device (model) interacts with the environment.

Each interface port can operate in the following modes:

in – the port value is only read for use inside the model;

out – the port value can only be updated by the model, but is not read;

inout – bidirectional port, the value is read and updated by the model;

buffer – buffer port, the value is read and updated by the model, but the signal source can be either a buffer or a single source.

Device architecture description

Architecture (architecture body) simulates the view of the device "from the inside". The device can be considered and described in different ways. It can be represented either as a composition of simpler modules (structural modeling style), or as a set of parallel algorithms (flow style - dataflow), or described as a process of sequential operations (behavioral style - behavioral), or can be described by a combination of these styles.

Multiple architectures can be associated with a single entity-type interface.

Behavioral description

The operation of the device is considered as a process of sequential calculation of expressions included in the process. The impetus for starting the process is to change (event-event) any signal from the list of "perceived" signals (sensitivity list). This list looks very similar to the list of parameters in other high-level programming languages.

Flow description

The dataflow style description uses concurrent expressions in the architecture. The number of concurrently calculated expressions can be any. Since the calculations take place in parallel, the order in which the expressions are written does not matter. The impetus for the start of calculations is to change any of the signals included in the expressions.

Structural description

The structural description interprets the device as a set of components interconnected by signals. Roughly speaking, it is a netlist.

Combined description of architecture

In a single body of architecture, you can combine all the above styles of description.

Features of synthesis of schemes according to descriptions in VHDL language

There are significant differences between the processes of modeling (simulation of circuit behavior) and synthesis of circuits using the VHDL language. Unlike a simulation system, a design system converts design information into a given format. Among them are the widely used EDIF format, specialized formats of various chip manufacturers, as well as Verilog, VHDL and others. The EDIF format is a kind of de facto standard, it is present in most modern design systems and can be used to exchange information. Among the specialized formats is the XNF format from Xilinx, which was widely used in previous versions of CAD from this company. When converting the source module into intermediate text in VHDL, it is usually transformed into structural structures of this language, which use the circuit features of the element base used.

The modeling system usually uses all the constructions of the VHDL language, while the design system does not use all its capabilities. Usually operations on type real are not supported at synthesis (in this case at synthesis the error is issued), the keyword after is ignored, a number of other secondary constructions of language is not supported also. In design systems, the event attribute can only be used to indicate the fronts of clock signals in conditional statements, preferably in if statements.

Modern design systems use the type std_logic to describe circuits, which replaces the bit type. The std_logic type has the following values:

0 - logical zero;

1 - logical unit;

U - value is not initialized;

X is an unknown value;

Z - high output resistance;

W is an unknown value with a weak signal source;

L - logical zero with a weak signal source;

H is a logical unit with a weak signal source;

‘-’ is an indefinite value.

Literature

1. Jens Lienig, Hans Bruemmer. Fundamentals of Electronic Systems Design. Springer; Softcover reprint of the original 1st ed. 2017 edition (July 24, 2018) 2018. 256 pp.

ISBN-10: 3319857622

ISBN-13: 978-3319857626

2. Neal S. Widmer, Gregory L. Moss, Ronald J. Tocci. Digital systems. Principles and applications. Pearson, 2016. – 1004 p.

ISBN-13: 978-0134220130

ISBN-10: 0134220137

3. Maxfield Clive. FPGA design. Young fighter course. - M .: DMK Press, 2015.-408p .: ill. (Series "Programmable Systems").

4. Ryabenky V.M., Zhuikov V.Y., Guliy V.D. Digital circuitry: Textbook. manual. - Lviv: New World - 2000, 2009, - 736 p.

5. Methodical instructions for laboratory work in the discipline "Electronic Systems" for students majoring in 171 Electronics, specialization Electronic components and systems / Compiled by: Osypenko K.S. - K .: KPI, 2017. - 45 c. (https://ela.kpi.ua/handle/123456789/40897)

6. Tochchi R., Widmer N. Digital systems. Theory and Practice, 8th ed .: Per. with English –M.: Ed. Williams House, 2004. - 1024 p.

7. Radio technical information transmission systems. Under the editorship of V.V. Kalmykova - M .: Radio and communication, 1990, -304 p.

8. Vasiliev V.I., Gusev Yu.M., Mironov V.N. Electronic industrial devices., -M .: Higher. school, 1988. -303 p.

9. Baskakov S.I. Radio circuits and signals: Textbook. – M .: Higher school, 2000. -536 p.

10. Snizhko E.M. Modeling and synthesis of discrete systems in VHDL language: Textbook.- D., RVV DSU, 2000.- 92 p.

11. Sergienko A.M., Korneychuk V.I. Microprocessor devices on programmable logic ICs. - Kyiv: "Korniychuk", 2005. - 107 p.

12. Abakumov V.G. Electronic industrial devices. - K.: Higher school, 1978. - 376 c.

13. Kazarinov Yu.M. etc. Radio engineering systems. - Moscow: Radio and Communication, 1990.

14. Molchanov A.A. Modeling and design of complex systems. - K.: Higher school, 1985. -200 c.

15. Zyuko A.G. etc. Theory of signal transmission. –M.: Radio and communication, 1986. -286 p.

16. Sklyar B. Digital communication. Theoretical foundations and practical application, 2nd ed .: Per. with English – M .: Ed. Williams House, 2003. - 1104 p.

17. Amosov V.V. Circuitry and means of designing digital devices. - SPB.: BHV-Petersburg, 2007. - 560 c . - (Textbook).

18. Wakerley JF Design of digital devices: in 2 volumes / JF Wakerley. - M.: Postmarket, 2002.