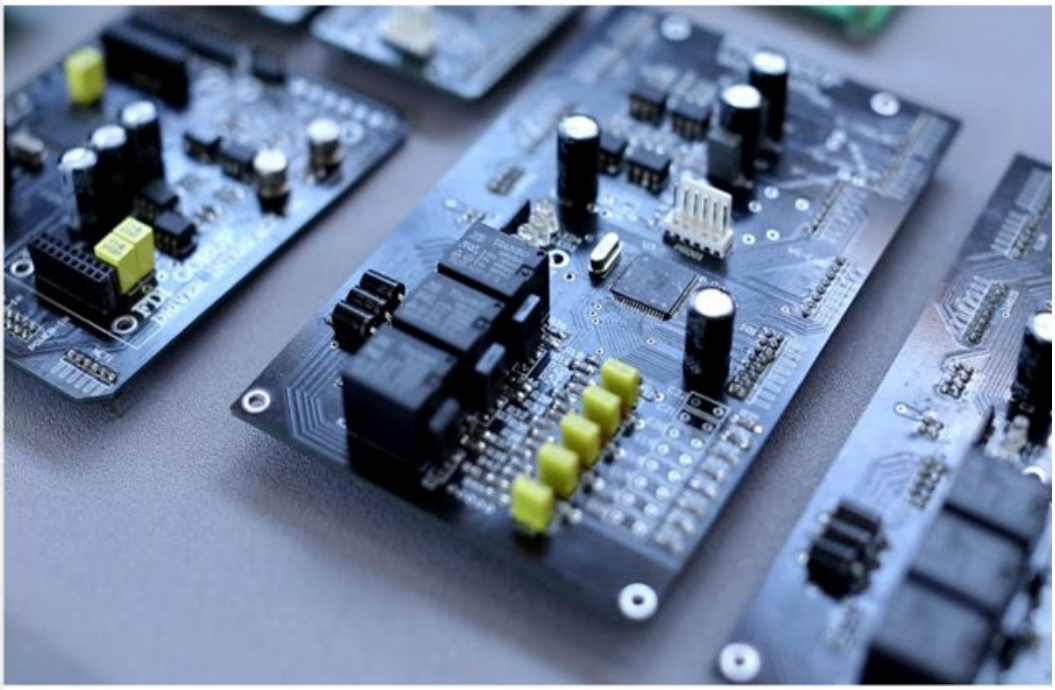
MODERN ELEMENTARY BASE OF DIGITAL SYSTEMS DESIGN

Київ
2023

Kulinich O.M., Lakhno V.A., Kasatkin D.Yu., Kharchuk N.S.



MODERN ELEMENTARY BASE OF DIGITAL SYSTEMS DESIGN

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

NATIONAL UNIVERSITY OF LIFE AND ENVIRONMENTAL SCIENCES OF UKRAINE

O.Kulinich, V.Lakhno, D. Kasatkun, N.Kharchuk

TEXTBOOK

MODERN ELEMENTARY BASE OF DIGITAL SYSTEMS DESIGN

Kyiv - 2023

NATIONAL UNIVERSITY OF LIFE AND ENVIRONMENTAL SCIENCES OF UKRAINE

Faculty of Information Technologies

Department of Computer Systems, Networks and Cybersecurity

O. Kulinich, V. Lakhno, D. Kasatkun, N.Kharchuk

MODERN ELEMENTARY BASE OF DIGITAL SYSTEMS DESIGN

Recommended for publication by the Academic Council of the NULES of Ukraine (protocol No.4 from 25 October 2023)

Kyiv - 2023

UDC 621.3.049.77.001.63
P79

Copying, scanning, recording on electronic medi etc, the book as a whole or any part of it is prohibited

The current elementary base of digital systems design. Textbook / Kulinich O.M., Lakhno V.A., D. Kasatkun, N.Kharchuk – K., 2023 – 168 p.

Recommended by the Academic Council NULES (protocol No. 4 dated 25 October 2023)

Reviewers:

S.V.Tolyupa, Doctor of Technical Sciences, Professor (Kyiv National University named after Taras Shevchenko);

M.M. Konotopets, Candidate of Technical Sciences, Associate Professor (Institute for Special Communication and Information Protection).

I.M. Bolbot -

The current elementry base for storing information - the main types of memory devices and integrated circuits with a programmable structure is presented in the textbook.

The textbook is intended for students who study disciplines related to creation and operation of digital means of information processing in
specialty 123 "Computer Engineering" and 125 "Cybersecurity"

© Kulinich O.M., Lakhno V.A., Kasatkun D.Yu.,
Kharchuk N.S., 2023
© NUBiP Ukraine, 2023

Published for copyright editing.

CONTENTS

Introduction ……..……..

Chapter 1. STORAGE DEVICE ……..

1.1 BASIC INFORMATION. SYSTEM OF PARAMETERS. CLASSIFICATION.……..

1.1.1. The most important parameters of SD.……

1.1.2. Classification of SD……...

1.2. BASIC STRUCTURES OF STORAGE DEVICES……..

1.2.1. 2D structure……

1.2.2. 3D structure……...

1.2.3. 2DM structure……

1.2.4. Sequential access memory……

1.2.5. Video memory…….

1.2.6. FIFO buffer……

1.2.7. Cache……...

1.3. ROM (M), PROM, EPROM, EEPROM MEMORY DEVICES……

1.3.1. Mask SD

1.3.2. Software PROM……..

1.3.3. Memory types EPROM and EEPROM……...

1.3.4. ROM pulse power……...

1.4. FLASH MEMORY……...

1.4.1. Memory type mass erasure……...

1.4.2. Flash memory with asymmetric block structure…….

1.4.3. File flash memory……

1.4.4. Memory type StrataFlash……..

1.5. THE USE OF PROGRAMMED SDs FOR SOLVING THE PROBLEMS OF INFORMATION PROCESSING……...

1.5.1. Implementation of logical (switching) functions…….

1.5.2. Realization of finite automata……

1.5.3. Reproduction of arithmetic and functional dependencies…….

Chapter 2. PROGRAMMABLE LOGIC MATRIX, PROGRAMMABLE MATRIX LOGIC AND BASIC MATRIX CRYSTALS.....

2.1 PROGRAMMABLE LOGIC MATRIX…….

2.1.1. Programmable logic matrices circuits……

2.1.2. Matrix logic programming……...

2.2. FUNCTIONAL VARIETIES OF PLM AND PML……..

2.2.1. Circuits with a programmable output buffer……

2.2.2. Systems with bidirectional outputs…….

2.2.3. Circuits with memory……

2.2.4. PLM with detachable connectors……...

2.2.5. Extension (expansion) of PLM…….

2.3. CLASSIFICATION OF BMC…….....

2.4. MODERN LARGE INTEGRATED SYSTEMS AND EXTREMELY LARGE INTEGRATED SYSTEMS WITH A COMPLEX PROGRAMMING STRUCTURE……...

2.4.1. General information……

2.4.2. Programmable logic and NVIS programming with mixed architecture…

2.4.3. Complex logic programmed by MAX 7000……..

2.4.4. Microcircuits of the FLEX 10K family……..

2.4.5. NVIS logical element……..

2.4.6. Programmable logic circuits of the "chip on a crystal" type.……

2.4.7. Parameters of complex integrated circuits of programmable logic……

2.5. JTAG INTERFACE AND PERIPHERAL SCANNING…….

2.5.1. Programming in the system……

Practical developments…….....

Chapter III. GENERAL CHARACTERISTICS OF MICROCONTROLLERS

3.1. Input/Output. registers……..……

3.1.1. Register of introduction/visitation…….……..

3.1.2. Status Register (SREG)…….

3.1.3. Stack……
3.2. Organization of memory…… ….
3.2.1. Ways of addressing operands……..
3.3. Types (formats) of commands……
3.4. Data types (formats)……..
3.4.1. Commands of logical operations……
3.4.2. Setting and resetting commands…….
3.4.3. Commands of transition, call of routines and return ...
3.4.4. Elements of the interface of the microprocessor system……
3.4.5. Executive devices of the microprocessor system…….
3.5. Universal asynchronous (UART) and synchronous/asynchronous (USART) transceiver modules…..
3.6. SPI module…….
RECOMMENDED BOOKS……..

INTRODUCTION

Digital information processing systems include a processor, memory, peripheral devices and interface circuits. The processor is a standard device. It is not manufactured for a specific system by special order, but solves the required task by sequentially executing specific commands from its own command system.

Memory is also implemented by standard microcircuits chip - the functions remain the same for different systems.

The standard LIC/ULIC leads in terms of integration, as the cost of an optimized high-density LIC/LIC in the hundreds of millions of dollars in this case is acceptable because it is spread over a large number of chips produced.

Along with the standard parts, the system includes several non-standard parts specific to this development. This applies to block management schemes ensuring their compatibility, etc. The implementation of a non-standard part of the system is historically associated with the use of chipsets of low and medium integration. The use of MIS and SIS is accompanied by a sharp increase in the number of IC cases, installation complications and a decrease in the reliability and speed of the system.

In addition, a long period is expected between the completion of the design of the circuit and the presentation of the first experimental samples, and any error or change in the circuit is very expensive.

The resulting contradiction found a solution in the way of developing large integrated circuits and ultra-large integrated circuits (LIC/ULIC) with a programmable and reprogrammable structure. The first representatives of this direction were programmable logic matrices (PLM), programmable logic matrix (PLM) and basic matrix crystals (BMK). Users can organize their own logic circuits, and even in a single copy, using simple and cheap ones compared to hardware developers. And it turns out to be much more convenient.

Programmable logic arrays are based on technologies similar to those used in programmable memory devices, but these technologies are used for somewhat unusual purposes.

PLM is a crystal containing a large number of elements that perform basic logic functions (mostly gates and flip-flops). The user can arrange these functions in one or another logic circuit according to his requirements, and can arbitrarily "break" not only the connections between logical buffer elements "AND/OR", but also determine the purpose of crystal terminals and microcircuits body. It is clear that such technology makes it possible to implement logical schemes that fully correspond to a specific project or development.

The development of LIC / LIC with a programmable and reprogrammable structure turned out to be such a promising direction that it led to the emergence of new effective ways of developing digital circuits, and the problem of creating an entire system on one crystal began to be solved.

Currently, there are many versions of powerful software designed for compatible PCs. Using them the actual programming of the system is carried out according to the description of logic functions and equations developed by the user with the help of computer "logic modeling", during which the correctness of the prepared data is also automatically verified data.

Today, you can write something that looks like a "program" and then use it to create a specialized integrated circuit. This approach provides a number of advantages for designers: increasing the size and simplicity of printed circuit boards, speeding up the process of developing new designs, increasing reliability, reducing the number of required types of standard digital microcircuits and improving one's own qualifications.

Chapter 1

STORAGE DEVICES

1.1 BASIC INFORMATION. SYSTEM OF PARAMETERS. CLASSIFICATION.

Storage devices (SDs) are used for storing and sharing information with other central storage devices. Memory chips account for about 40% of total IC production and play a critical role in many systems of varying importance. Microcircuits and memory systems are constantly being improved both in the field of circuit engineering and in the development of new architectures. At present dozens of different types of SDs have been created and are in use.

The most important parameters of SDs are controversial. For example, high information capacity is not compatible with high speed, and speed, in turn, is not compatible with low cost. Therefore, memory systems are characterized by a multi-level hierarchical structure and its implementation may differ significantly depending on the role of this or that software.

The following levels can be distinguished in the most developed hierarchy of computer memory:

- Register registers are part of the processor or other devices, thus reducing the number of accesses to other memory levels that are performed outside the processor and require more time for information exchange operations.

- Cache memory is used to store copies of information used in current exchange operations. Fast cache memory increases computer performance.

- The main memory (arbitrary, non-volatile) works in the mode of direct exchange with the processor and, if possible, coordination with it in terms of speed. The currently executing program fragment must be in main memory.

- Specialized types of memory typical for some specific architectures (multiport, video memory, etc).

- External memory stores large amounts of information. This memory is implemented on the basis of devices with a movable information carrier (magnetic, optical disks, etc.). This textbook does not cover external storage devices.

1.1.1. The most important parameters of SD.

Information capacity is the maximum possible amount of stored information. It is expressed in bits or words (bytes). A bit is stored by a memory element and a word is stored by a memory cell (CM), i.e., groups of memory cells to which only simultaneous access is possible. Adding the multiplier "K" (kilo) to a unit means multiplying by and adding the multiplier "M" (mega) means multiplying by

The organization of the SD is the number of words to be stored multiplied by their bit size. This gives the information capacity to SD, but the organization of the SD can be different with the same information capacity, so that the organization is an independent important parameter.

The speed of the computer is evaluated by reading, recording and the duration of the reading / recording cycles. The reading time is the time interval between the moments of reading the signal and the word at the output of the control unit. The recording time is the time interval after the appearance of the registration signal, which is sufficient to bring the SD to the state specified by the input word. The minimum allowable interval between successive readings or recordings create a corresponding cycle. The length of the cycles may exceed the time of reading or recording and it may take some time to establish the necessary initial state of the memory after these operations.

Reading, recording and cycle time are traditional parameters. Some modern SDs need to be supplemented with new ones. The reason lies in the more complex nature of accessing the stored data, when accessing the first word of a certain group of words (packet) takes longer time than accessing the subsequent ones. For such modes enter the access time parameter during the first access (*Latency*) and the transmission speed for subsequent packet words (*Bandwidth*). Transfers, in turn,

are evaluated by two values - the limit (in the packet) and the average (*Latency*). As the package falls, the average pace drops, increasingly deviating from the limit.

In addition to certain basic parameters, a whole set of time intervals is also defined for SDs. The above mentioned dynamic parameters work. In addition to them there are a number of mode parameters, the provision of which is necessary for the normal functioning of the SDs, since it has several control signals for which a certain relative synchronization must be ensured. Duration and limits of mutual location in time are set for these signals.

One of the possible sets of SDs signals (Fig. 1.1) includes the following signals:

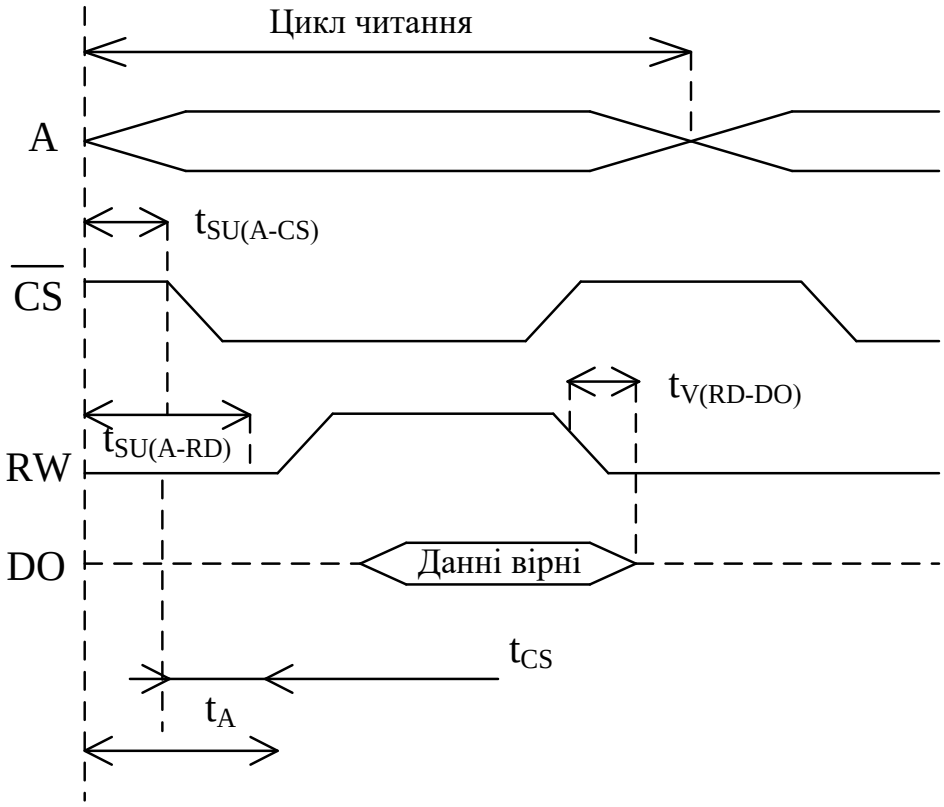
A is an address, the bit size of which *n* is determined by the number of cells of the memory card, that is, the maximum possible number of words that can be stored on the memory card. In the case of SDs the number of cells is expressed as an integer to the power of two. The address is the number of the cell to which the link is sent. It is obvious that the bit rate of the address is related to the number of written words N by ratio $n = \log_2 N$ or example, SDs with an information capacity of 64,000. words has 16-bit addresses expressed in words:

$$A = A_{15} A_{14} A_{13} \ldots A_0$$

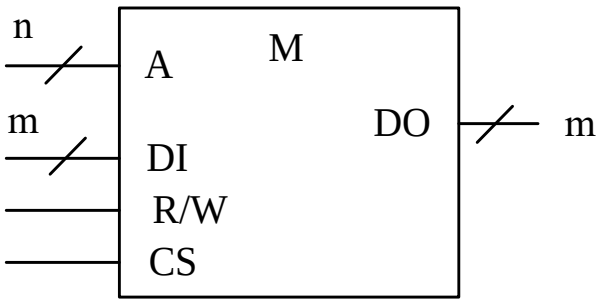
CS - (*Chip Select*) or CE (*Chip Enable*) which enables or disables the operation of this chip.

R / W - (*Read / Write*) specifies the performed operation (with a value of one - reading, with a value of zero - writing);

DI and **DO** (*Data Input*) and (*Data Output*) are input and output data buses which data transfer rate *m* is determined by the SD organization.



a)



b)

Fig. 1.1. Typical SD signals and their time graphs

The requirements for the relative time position of the two signals (A and B) are determined by the time, hold and save settings.

The specified time of signal A in relation to signal B $t_{SU(A-B)}$ is the interval between the beginnings of both signals.

Holding time $t_{H(A-B)}$ is the interval between the start of signal A and the end of signal B.

The holding time $t_{V(A-B)}$ is the interval between the end of signal A and the end of signal B.

The duration of signals is denoted as t_W (index from the word Width - width).

The following sequence of signals is characteristic of SD. First of all, the address is specified so that subsequent operations do not touch any cell other than the selected one. Then the operation of the microcircuit is turned on by the signal CS (CE) and the read / write signal R / W is given. If, for example, a read operation is specified, the control unit makes the data available for reading after applying the specified signals which requires a certain time. The data is read by the falling edge of the R / W signal, the position of which in time should ensure the determination of the correct data at the output of the SD.

A sample time diagram for the analyzed set of signals of the SD and the reading operation is shown in Fig. 1.1 b.

Index A (from the word *Address*) indicates according to the access time standard the time intervals from the appearance of a certain control signal to the appearance of an information signal. The access time of the corresponding address signal is indicated if followed by a stop, as , but sometimes as .The access time of the relative CS signal is indicated in the same way. Time is also called sample time and time is selection time.

In addition to the specified parameters the SD also uses a number of others that do not require special attention, as they are traditional for digital circuits. An exception is the property of energy independence, that is, the ability of the control unit to record data when the voltage is turned off. Energy independence can be natural or artificial, which is achieved by introducing backup power sources that are automatically connected to the information carrier when the main power is turned off.

1.1.2. Classification of SD

The most important feature of the classification of the SD is the method of data access.

With address access, the code when entering the address indicates the cell from which the exchange takes place. All addressable memory cells are equally accessible during access. These SDs are the most advanced, other types of memory are often built on the basis of addressable memory with appropriate modifications.

Addressable memory is divided into RAM and ROM. The operating software stores data, participates in the exchange during the execution of the program and can be changed at any time. RAM elements, as a rule, do not have nonvolatile memory. In ROM, the content either does not change at all or changes, but rarely and in a special mode. According to the operating mode it is "permanent memory".

RAM is divided into static and dynamic memory. In the first version the memory elements are triggers that retain their state when voltage is applied to the circuit and there is no new data. In the second variant data is stored in the form of capacitor charges formed by elements of MOSFET structures. Self-discharge of capacitors leads to data destruction, so they must be periodically regenerated (every few milliseconds). At the same time the packing density of dynamic memory elements is several times higher than the packing density achievable in static RAM.

Regeneration of data in dynamic SD occurs with the help of special drivers. SD with dynamic memory elements are also developed. They have an internal regeneration system, and its external behavior with respect to control signals is similar to the behavior of static APs. Such SDs are called quasi-static.

Static RAM is called SRAM (static RAM), and dynamic RAM is called DRAM (dynamic RAM).

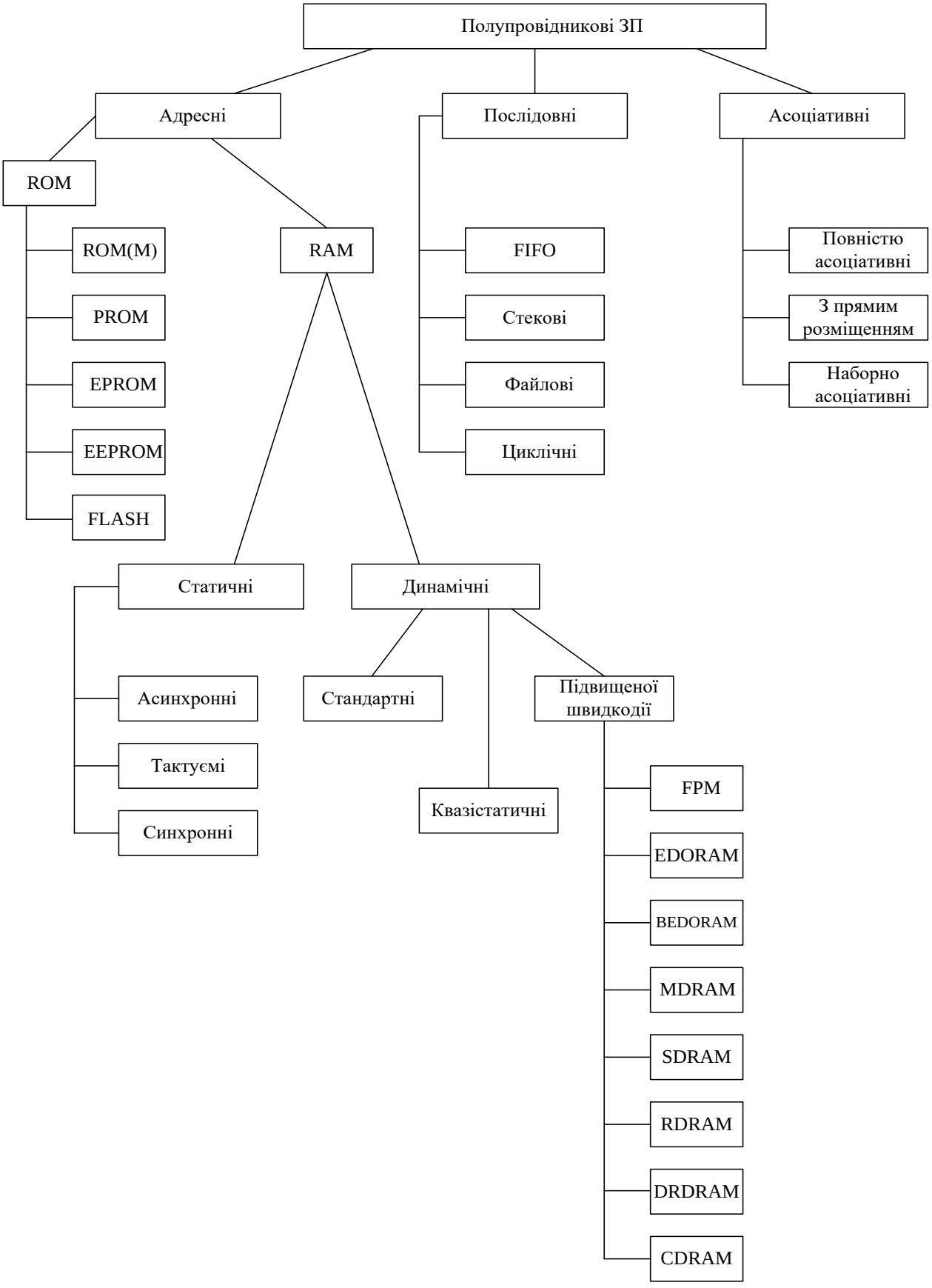


Figure 1.2. Classification of SD semiconductors

Static RAM can be divided into asynchronous, clock and synchronous (pipeline). Signals can be called pulses and levels in asynchronous control. For example, the CS enable signal may remain unchanged and allow memory access for many cycles. In the SD clock generator some signals must be pulsed, for example, the CS enable signal must change from passive to active in each cycle of memory access (the edge of this signal must be formed in each cycle). This type of SD is often called synchronous. The term "in time" is used here to "release" the term "synchronous" for new types of SD. Their transport path for data transfer is organized synchronously with the clock frequency of the processor, which increases the data transfer rate several times. More specifically, the essence of the organization of the SD pipeline is discussed in paragraph 1.8, since it plays an important role in increasing the dynamic speeds of the SD (SDRAM option).

Dynamic RAMs are characterized by the largest information capacity and low cost, so they are used as the main memory of a computer. Many high-performance architectures have been developed as this memory requires high performance, as noted in the classification. These architectures are discussed in detail in section 1.7.

Static SDs are 4...5 times more expensive than dynamic SDs and have approximately the same information capacity. Their advantage is high speed, and a typical field of application is cache chips.

ROM-type non-volatile memory is programmed during production by integrated technology methods using one of the masks used in this case. This is a permanent memory for a consumer in the full sense of the word, as it cannot change its meaning.

The letter P (for Programmable) appears in the designations of the following three ROM variants. This is a user-programmable memory. Its contents is stored once (in PROM) or can be overwritten by erasing old information and storing new information (in EPROM and EEPROM). In EPROM erasing is carried out by irradiating the crystal with ultraviolet rays. EEPROM is erased by electrical signals. The English names mean Electrically Programmable ROM and Electrically Erasable ROM. PROM programming and EPROM and EEPROM reprogramming

is performed under normal laboratory conditions using special programmers or in special modes without special devices (for EEPROM).

Flash memory is similar to EEPROM memory (or otherwise E^2PROM), but has design and technological features that allow it to be separated into a separate type.

Both EPROM and E^2PROM data are written using electrical signals.

In serial access memory the recorded data creates a certain queue. Reading occurs word by word in the order of writing or in reverse order. The model of such a SD is a sequential chain of memory elements in which data is transferred between adjacent elements.

The direct order of reading occurs in FIFO buffers with the discipline "First in - first out", as well as in file and cyclic memory.

The difference between a FIFO memory and a file buffer is that in a FIFO, writing to an empty buffer immediately becomes available for reading, that is, it reaches the end of the chain (buffer model). In a SD file data is entered at the beginning of the chain and appears at the output after a given number of hits equal to the number of elements in the chain. Due to the independence of reading and writing operations the actual location of data in memory during reading is independent of any external feature. As recorded data is combined into blocks, they are linked by special start and end characters (files). Receiving data from the SD file begins after the receiver detects the beginning of the block symbol.

In cyclic memory words are accessed one after the other with a fixed period determined by the amount of memory. This type includes video memory (VRAM) among solid-state storage devices.

Reading in reverse order is characteristic of stack software, for which the discipline "last in, first out" is implemented. Such buffers are called LIFO (Last In - First Out) buffers.

The time of access to a certain unit of information stored in the following MD is a random variable. In the worst case such access may require viewing the entire volume of stored data.

Associative access looks for information by some feature, not by the location in memory (address or position in a queue). In the most complete version all words stored in the memory are simultaneously checked for the correspondence of the sign, for example, for the coincidence of certain word fields (tags - from the English tag word) with the symbol specified by the input word (tag address). The words performing the function are displayed. The discipline of issuing words if the tag encounters several words and the discipline of writing new data may differ. The main area of use of associative memory in modern computers is data caching.

Technical and economic indicators of MDs largely depend on their scheme and technical implementation., It is also possible to classify MDs according to this feature, but it is more convenient to consider this issue in relation to individual types of memory.

1.2. BASIC STRUCTURES OF STORAGE DEVICES

Addressable memory devices are classified as static and dynamic operating devices and ROM. Many variants of these SDs have a lot in common from the point of view of structural schemes, which makes it more rational not to consider each SD completely separately, but to study some generalized structures with a further description of the memorised elements for different SDs.

The similarity of structures is especially characteristic for static RAM and ROM. Dynamic RAM structures are specific and are discussed below. 2D, 3D, and 2DM structures are most typical of static RAM and ROM.

1.2.1. 2D structure

In the 2D structure (Fig. 1.3) the memory elements are organized into a rectangular matrix with dimensions $M = kxm$, where M is the information capacity of the memory in bits; k - the number of recorded words; m is their transmission speed.

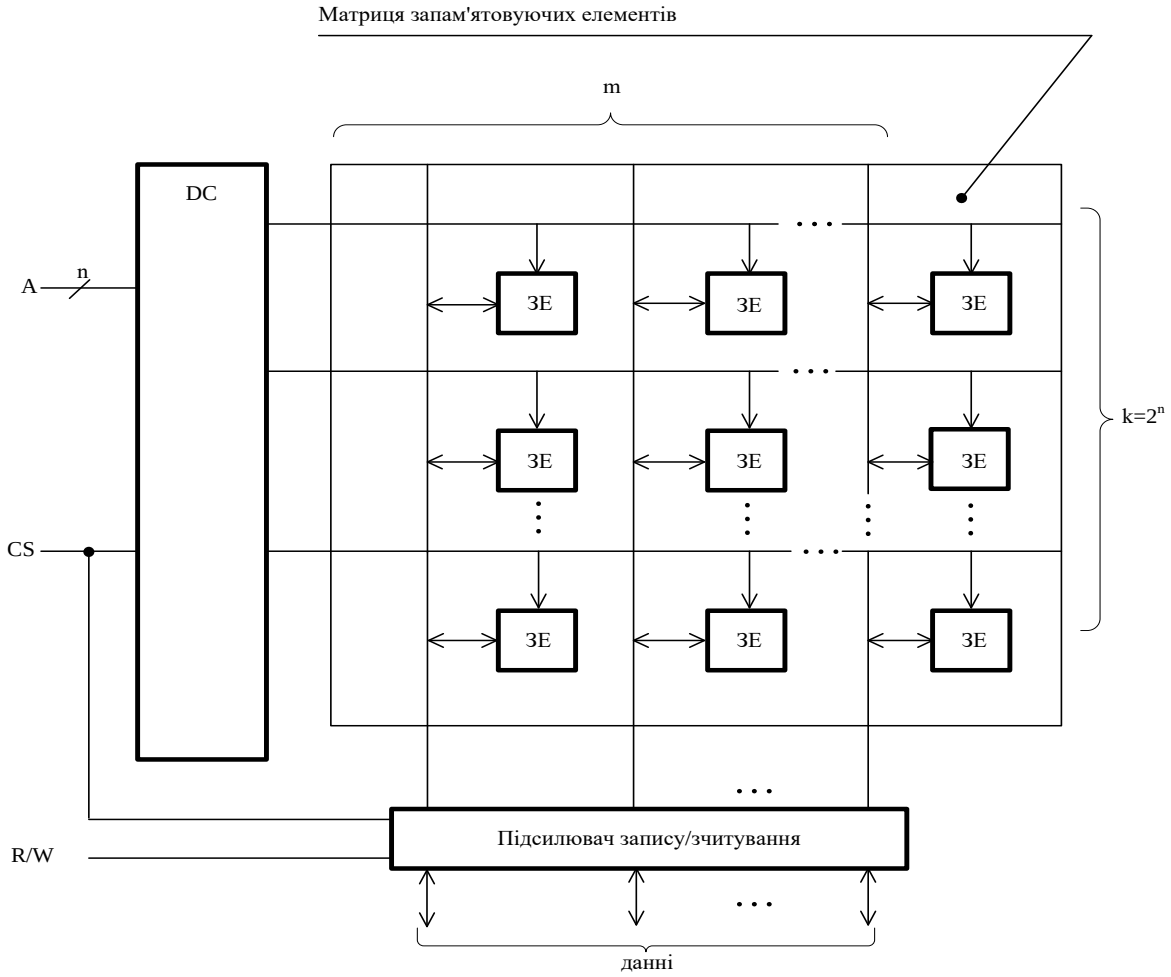


Fig. 1.3 2D structure of the SD type.

The DC address code decoder, when the CS (Chip Select) enable signal is present, activates one of the output lines allowing simultaneous access to all elements of the selected line, that stores the word which address corresponds to the line number. The elements of one column are connected by a vertical line - an internal data line (bit line, read/write line). Column elements store bits with the same names for all

words. The direction of the exchange is determined by the read/write amplifiers under the influence of the R/W signal (read - reading, write - writing).

The two-dimensional structure is used only for low-information storage devices, since several of its disadvantages become apparent with increasing capacity, the most obvious of which is the excessive complexity of the address decoder (the number of decoder outputs is equal to the number of written words).

1.2.2. 3D structure

The 3D structure makes it possible to radically simplify address decoders using two-coordinate discretization of memory elements. The principle of sampling two coordinates (Fig. 1.4, a) is explained on the example of a ROM-type memory card that performs only data reading operations.

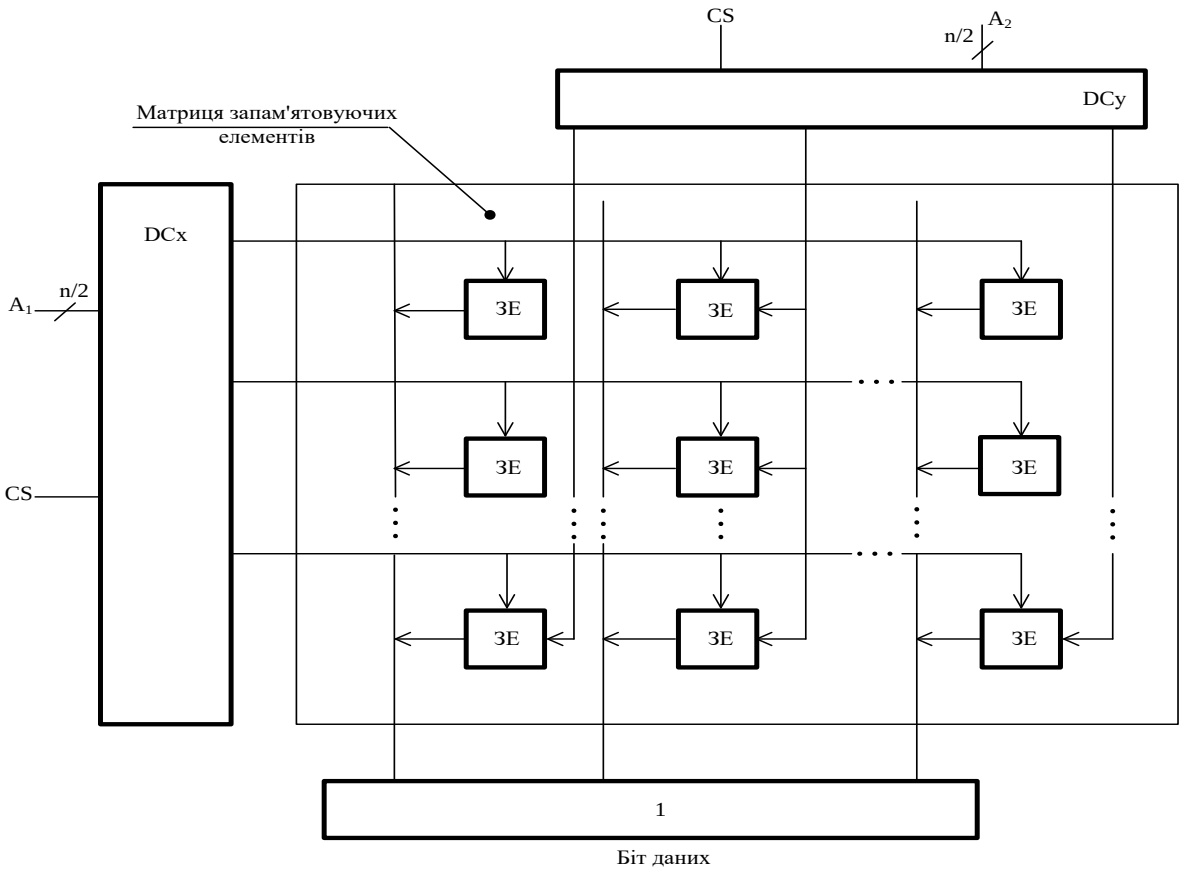


Figure 1.4. 3D structure of SD with one-bit organization (a)

Here, the n-bit address code is split into two halves, each of which is decoded separately. A memory element is selected at the intersection of the active output lines of both decoders. There will be exactly such intersections:

$$2^{n/2} \times 2^{n/2} = 2^n$$

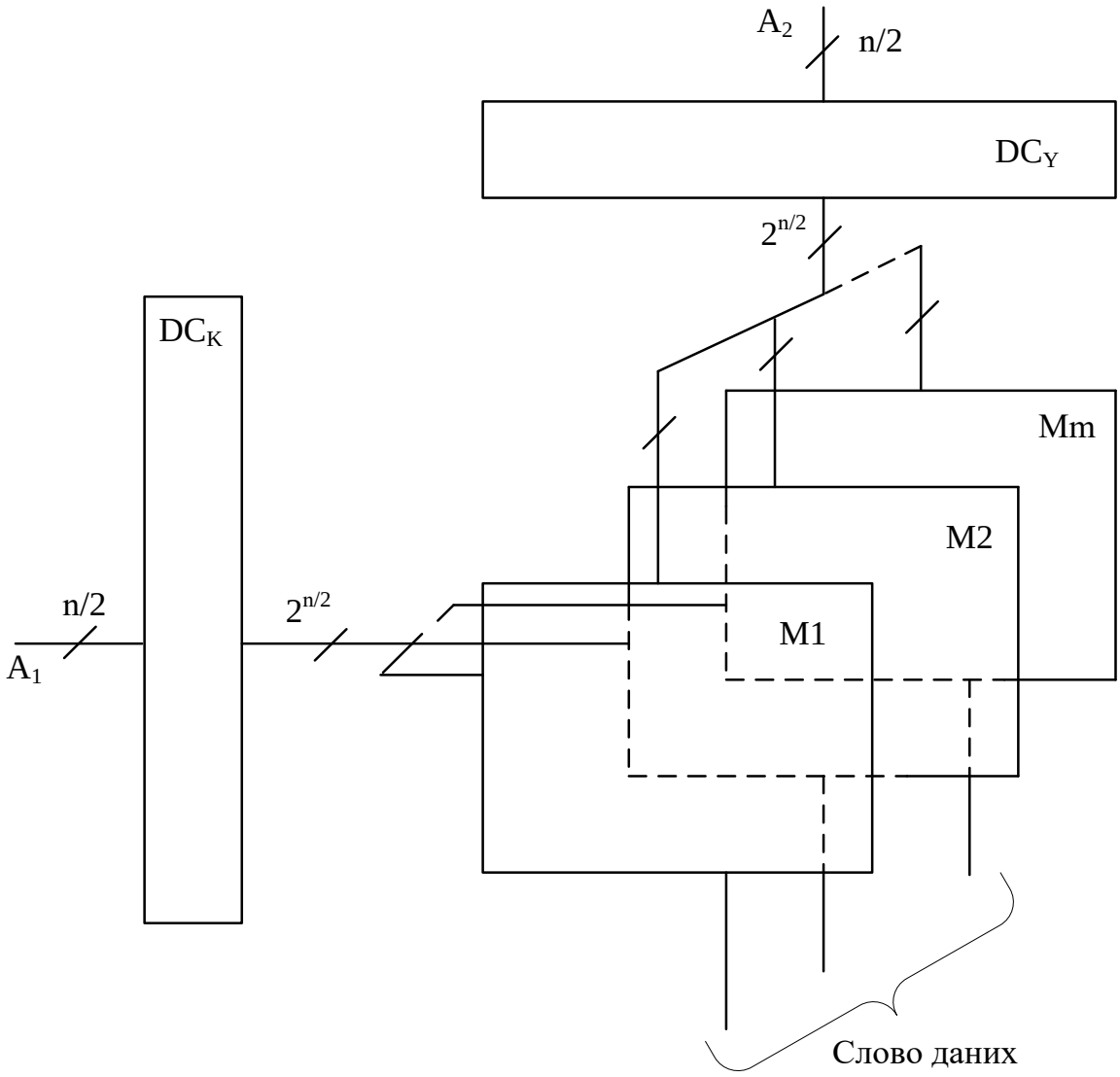


Fig. 1.4. SD 3D structure with multi-bit organization

$2^{n/2} + 2^{n/2} = 2^{n/2+1}$ which is significantly less than with real values of n. This significant difference is noticeable even for a memory card with a small capacity: for a 2D structure, when storing 1 thousand. In short, a decoder with 1024 outputs will be required, while the 3D structure requires two decoders with 32 outputs each. The disadvantage of the 3D structure is primarily the complication of memory elements that have bi-coordinate sampling.

3D structure shown in fig. 1.4, and for SD with one-bit organization it can be used in SD with multi-bit organization (Fig. 1.4, b), thus obtaining a "three-dimensional" character. In this case, several matrices are controlled from two decoders, with

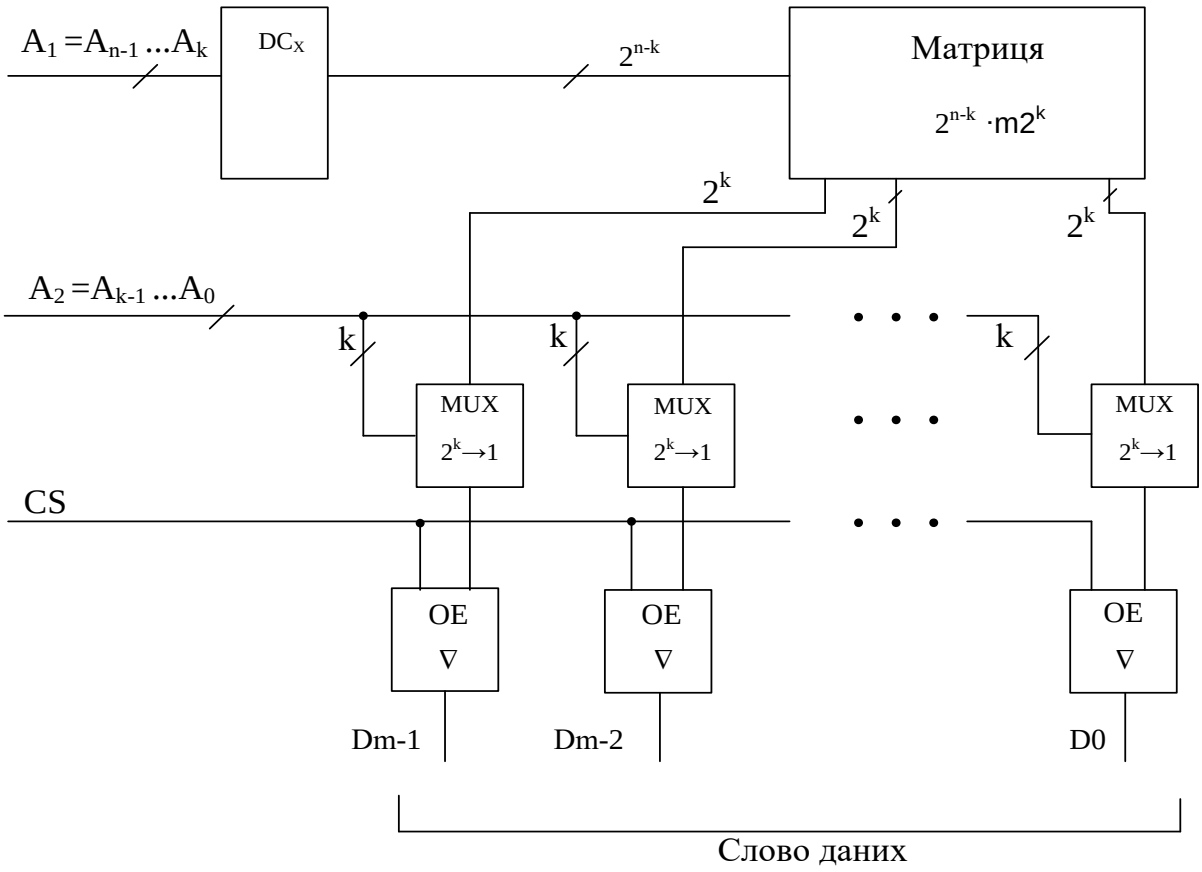
respect to which they are included in parallel. Each matrix outputs one bit of the addressed word, and the number of matrices is equal to the number of bits of the written words.

3D structures are also quite limited in use, since 20M structures (modified 2D) combine the advantages of both considered structures - address decoding is simplified and there is no need to remember two-coordinate sampling elements.

1.2.3. 2DM structure

ROM (Fig. 1.5, a) with a 2DM structure for a matrix of memory elements with addressing from the DCx decoder has the character of a 2D structure: the excited output of the decoder allocates the entire line. However, in contrast to the 2D structure the length of the line is not equal to the number of bits of the stored words, but many times more. At the same time the number of rows of the matrix decreases and, accordingly, the number of decoder outputs decreases. To select one of the lines not all bits of the address code are used, but some of them. $A_{n-1}...A_0$ Remaining bits of the address(A_{k-1} до A_0) are used to select the desired word from the set of words contained in the line. This is done with the help of multiplexers which address inputs are equipped with codes $A_{k-1}...A_0$. The length of the line is equal to m 2^k, where m is the bit size of the stored words. For each "segment" of a line length 2^k the multiplexer selects one bit. The output word is created at the outputs of the multiplexer. With the agreement of the CS signal received at the OE-inputs of the controlled buffers with three states, the output word is transmitted to the external bus. the multiplexer selects one bit. The output word is created at the outputs of the multiplexer. With the agreement of the CS signal received at the OE-inputs of the controlled buffers with three states, the output word is transmitted to the external bus.

Figure 1.5 shows a more general view of the 2DM structure for 3G RAM with read and write operations. The "long" line is read from the matrix M.



a)

Figure 1.5. RAM Structure 2DM for RAM

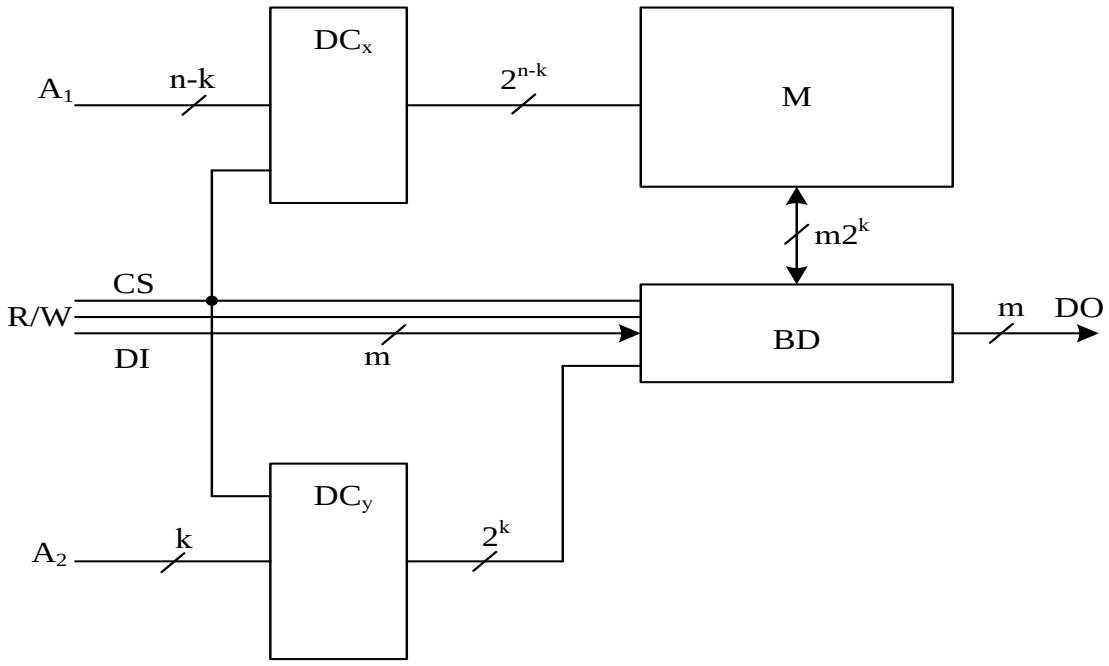


Figure 1.5 Structure of RFP tiau 2DM for RAM (b)

The data on the required segment of this line is written (or read from) by controlled data buffers BD, which receive the output data of the second decoder DCy and perform not only multiplexing functions, but also data transfer functions under the influence of the R / W signal.

1.2.4. Sequential access memory

Sequential access memory is built either by advancing data in a chain of elements (similar to shift registers), or by storing data in an address register with the necessary management of access addresses.

The main representatives of this type of memory are video memory, FIFO buffer and stack.

1.2.5. Video memory

The video memory works cyclically, codes appear sequentially at its output in the order of scanning the monitor screen with a beam that determines the brightness parameters (color, brightness) of the elementary points of the screen - pixels. The current image on the monitor - a frame - is represented by a row of words, the length of which is equal to the number of pixels on the screen. A word corresponding to one pixel can have a bit size of 8 (for black-and-white monitors) to 24 (for full-color monitors). In the case of implementation based on addressable memory, cyclic access to data is provided by an address counter with a module equal to the number of words in the memory. When reading after each access, the address is increased by one, providing sequential access to all cells of the SD. When the counter is full, a frame start signal is generated to control the monitor (to start frame synchronization). Recording is possible in batch or single recording mode. In the first case the counter overflow signal and its transition to the initial address is a signal to start transferring a block of data from the main memory or video buffer. In the second case, the variable address of the cell (the number of pixels) and the data are stored in the buffer, and when this address and the contents of the counter match one cycle of writing a new word is performed. At other times, the SD works in normal mode.

Figure1.6. shows the construction of a cyclic SD with the promotion of information with the presentation of data storage and rewriting elements in the form of static registers.

The lower channel of the MUX is selected during reading, and the recorded data is continuously overwritten from the output to the input of the chain of memory elements. Special synchronization signal codes are entered into the data sequence (frame and line, but only the frame is shown in fig.1.6 to clarify the principle). The appearance of the synchronization signal code at the output is determined by the comparator and synchronizes the start of the monitor scan.

Batch recording can start after receiving a transfer request during framing sync code transmission. At the same time a frame transmission permission signal is generated from the computer's memory to the DI input, and the multiplexer is switched to the upper channel.

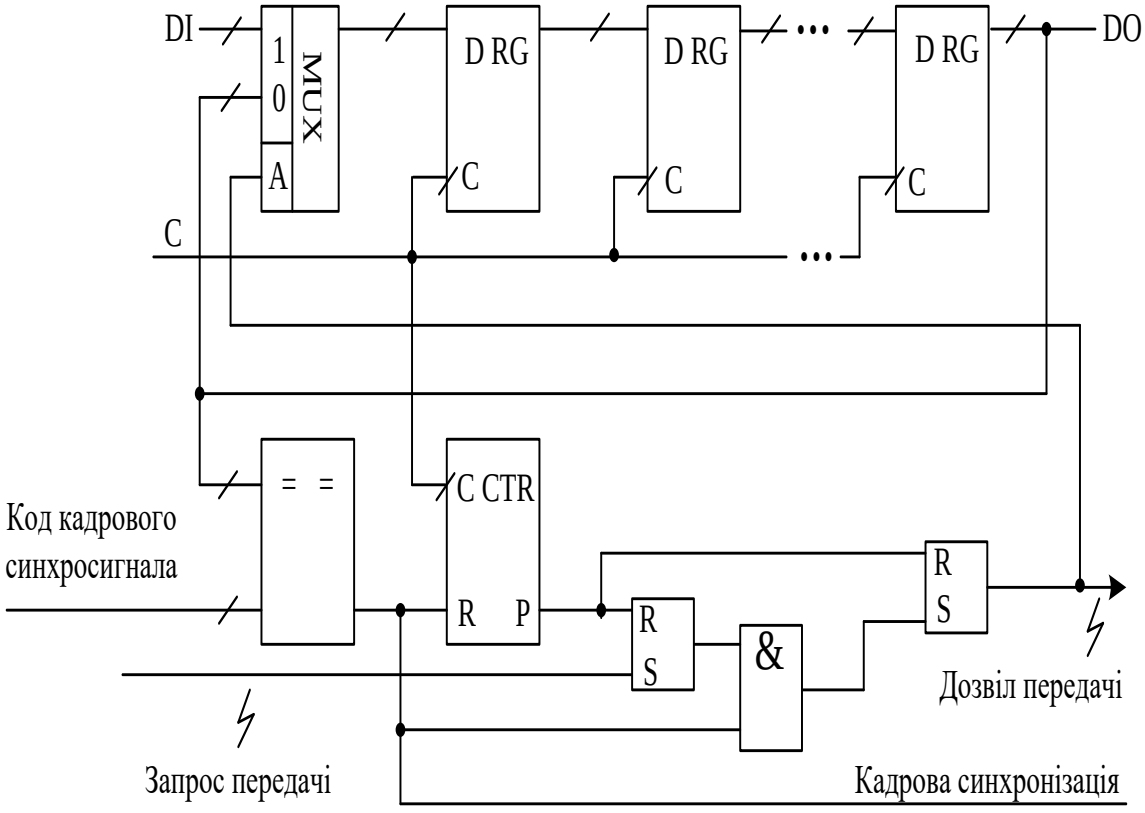


Figure 1.6. Video memory structure.

After receiving the entire frame the CTR counter, the capacity of which is equal to the length of the frame, overflows, and on the overflow signal of SD, the CTR returns to the cyclic overwriting mode.

The device must have an additional scheme for comparing the counter code and the input address code (pixel replacement code number) for single entries. When they overlap the multiplexer switches to the upper channel for one duty cycle ensuring that only one word is replaced.

1.2.6. FIFO buffer

The FIFO buffer, an example of its structure is shown in fig. 1.7, is a SD for storing queues (lists) of data with word selection order, the same as the order of their arrival. The interval between words can be completely different, since the time of writing a word into the buffer and reading a word from the buffer is set independently of each other by external control signals.

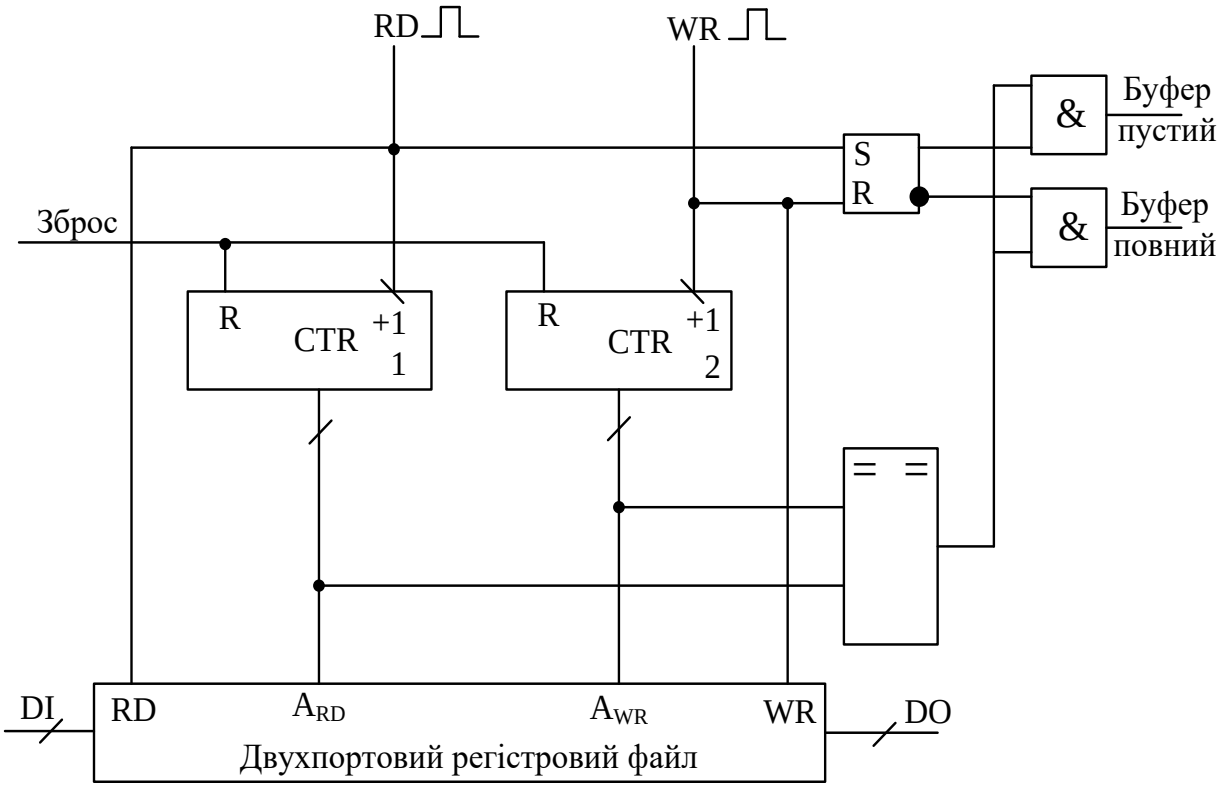


Fig.1.7 FIFO buffer structure

The ability to have different rates of receiving and sending words is necessary, for example, if the receiver can receive data that arrives regularly at a certain frequency, and the source of information sends words at a higher rate and possibly

also irregularly. Such data enters the FIFO buffer at its own pace and is regularly read from it at the frequency required by the recipient of the data. The new word is placed at the end of the queue, and is read from the beginning of the queue.

In the diagram (fig. 1.7), both address counters CTR1 and CTR2 are reset to zero before starting work. During input addresses are incremented by one with each access, that is, incremented starting from zero. The same thing happens when reading words, so that the read address always "chases" the write address. If the addresses match during reading, the buffer is empty. If during writing the addresses are equal, then the buffer is full (the addresses occupy the entire capacity of the counter). These situations are signaled by appropriate signals. If the buffer is full, you must stop receiving data, and if it is empty, you must stop reading. The line is lengthened or shortened depending on the difference between the number of written and counted words. The transition through zero does not cause complications.

The task of building a stack can be solved in essentially the same way. We'll meet this challenge later in the show when we look at building a microprocessor.

1.2.7. Cache

Cache memory stores copies of information transferred between devices (primarily between the CPU and main memory). It has a small capacity compared to the main memory and a higher speed (implemented on trigger memory elements).

When reading data, the cache is first accessed (fig. 1.8). If the cache has a copy of the data of the addressed main storage cell, the cache generates a hit signal and sends the data to the common data bus. Otherwise, the Hit signal is not generated, a reading from the main memory is performed, and the read data is simultaneously moved to the cache.

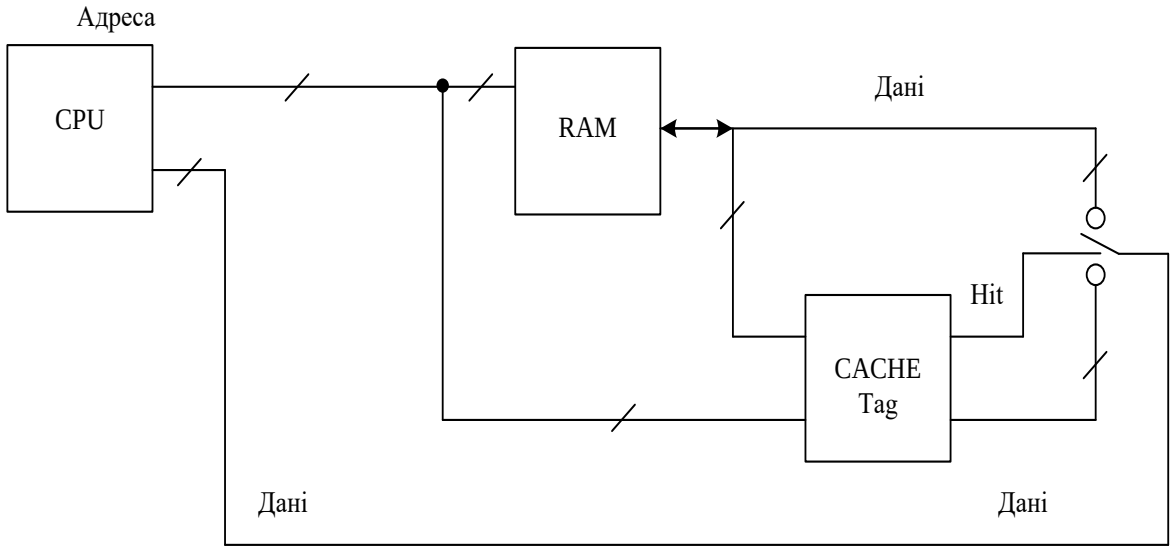


Fig.1.8. Cache structure

Caching efficiency is determined by the fact that most applications are cyclical and use the same data multiple times. Thus, when you first use data from the relatively slow main memory, multiple accesses take less time. In addition, when the processor uses the cache main memory is freed and data can be regenerated in dynamic memory or memory can be used by other devices.

The capacity of the cache is much smaller than the capacity of the main storage, and each information unit placed in the cache memory must be accompanied by additional data (tag) indicating a copy of the contents of which main storage cell this information unit is.

Fully connected cache memory (FACM) the structure of which is shown in fig. 1.9 data is stored in each cell, and the "tag" field contains the full physical address of the information, a copy of which is stored. For each exchange, the physical address of the requested information is compared with the "tag" fields of all cells, and if they match, a hit signal is set in any cell.

When reading and the value of the signal Hit = 1, the data is sent to the data bus, if there are no matches (Hit = 0), then when reading from the main memory, the data together with the address are placed in a free or last unused cache cell.

When writing, the data along with the address is usually first cached (in the detected cell when Hit = 1 and free when Hit = 0). Copying data to the RAM is

carried out under the control of a special controller in the absence of access to the memory.

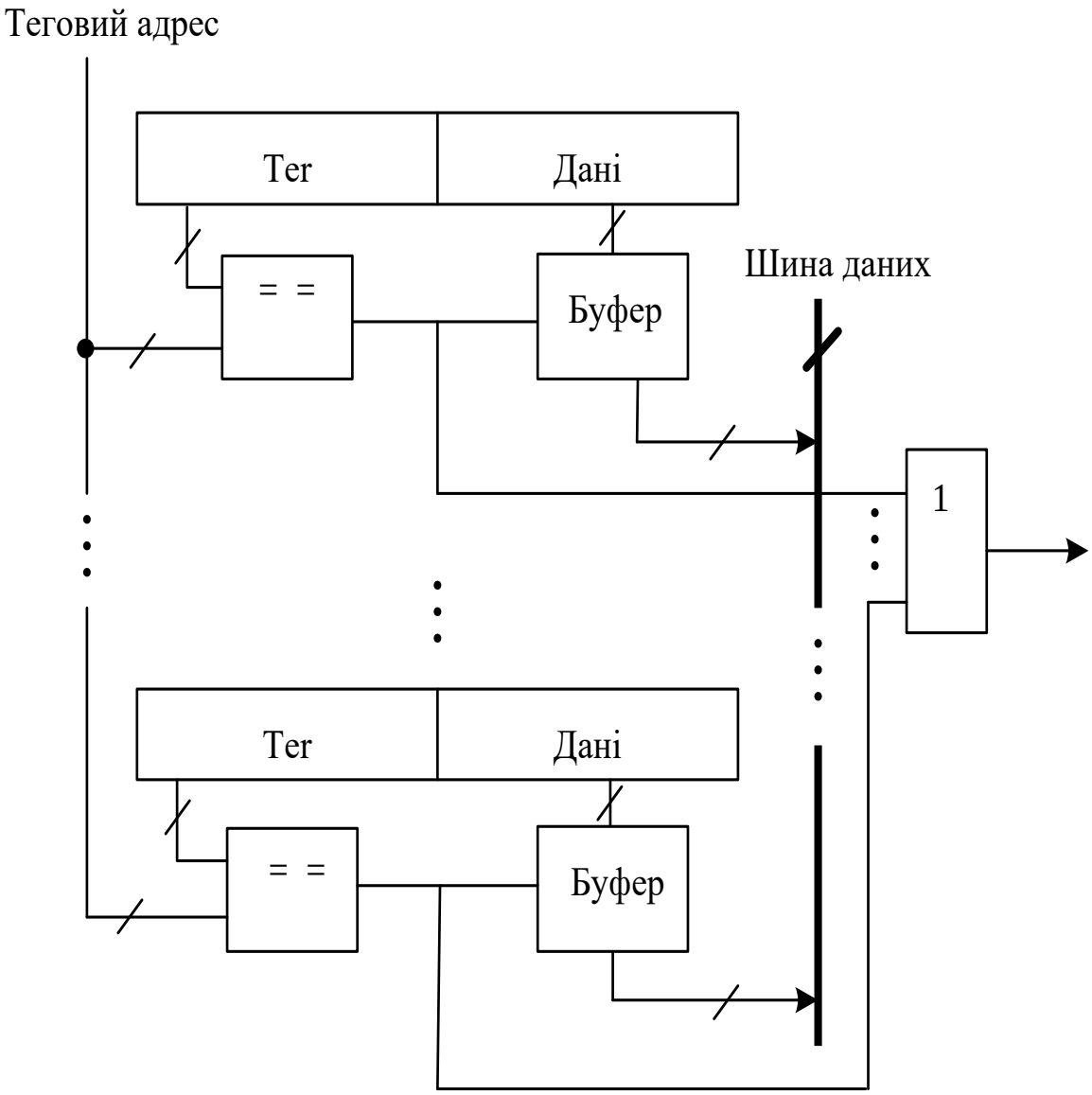


Fig. 1.9. A fully associative cache structure

FACM memory is a rather complex device and is used only for small volumes, mainly in special applications. At the same time this type of cache provides the greatest functional flexibility and conflict-free addresses, since any information unit can be loaded into any cell of the cache.

The complexity of FACM forces us to look for other cache structures that are more economical in terms of hardware costs for their implementation. Such structures include direct allocation cache and set associative architecture cache

(multi-way association). For a special consideration of these structures, we note that the main parameters of the cache are the size of the lines (cache line) and their number (Fig. 1.10). A line is a set of words. We will consider its capacity as a corresponding main memory page.

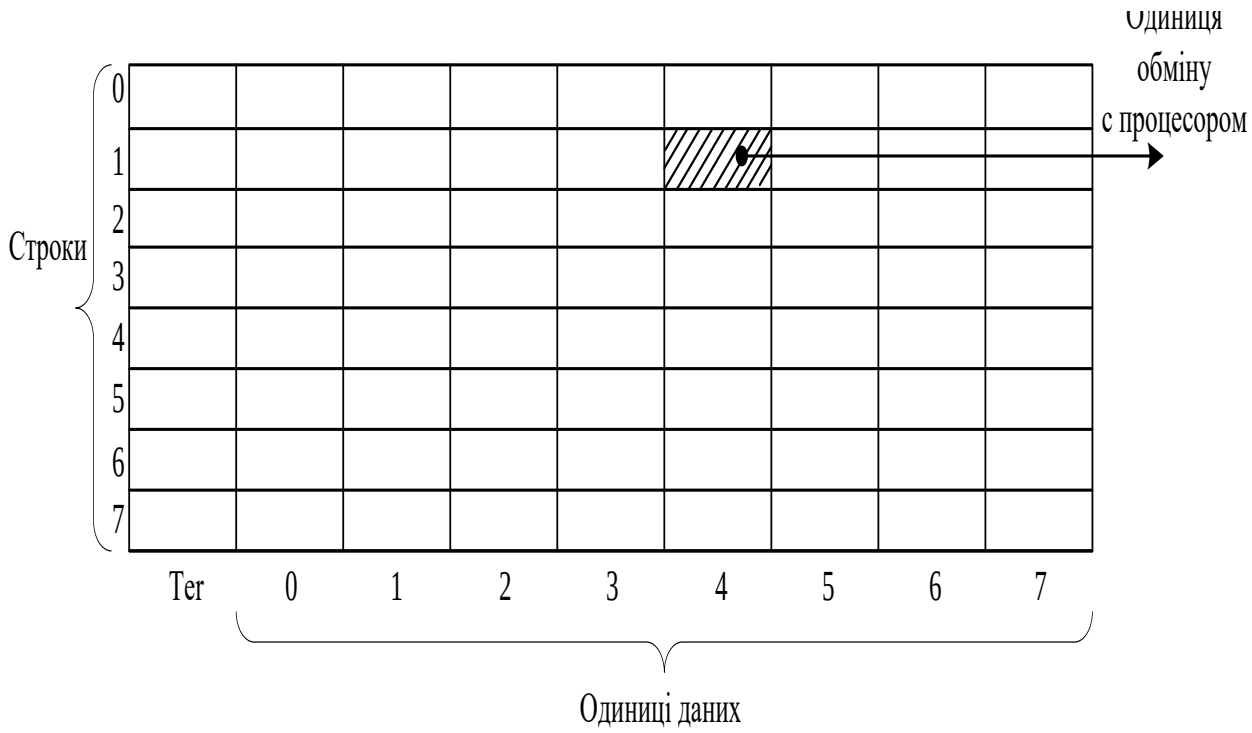


Fig. 1.10. Representation of the cache as a set of lines

In the FACM structure, which is also called the random loading structure, any page can be loaded into any line of the cache buffer (fig. 1.11, a). The full physical address is used as a tag, if we are talking about addressing individual words, or the high bits of this address minus the low bits (hash) if the hash addresses a word in a line.

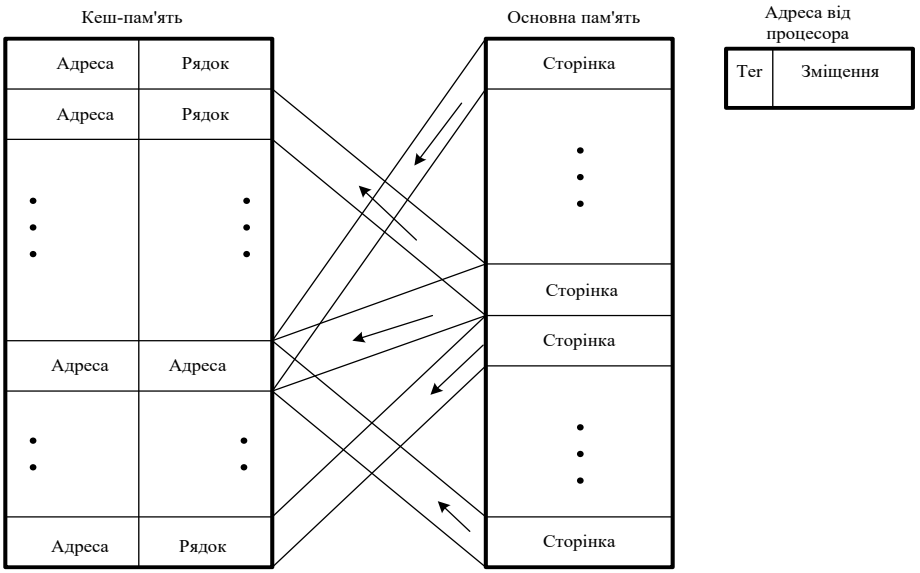
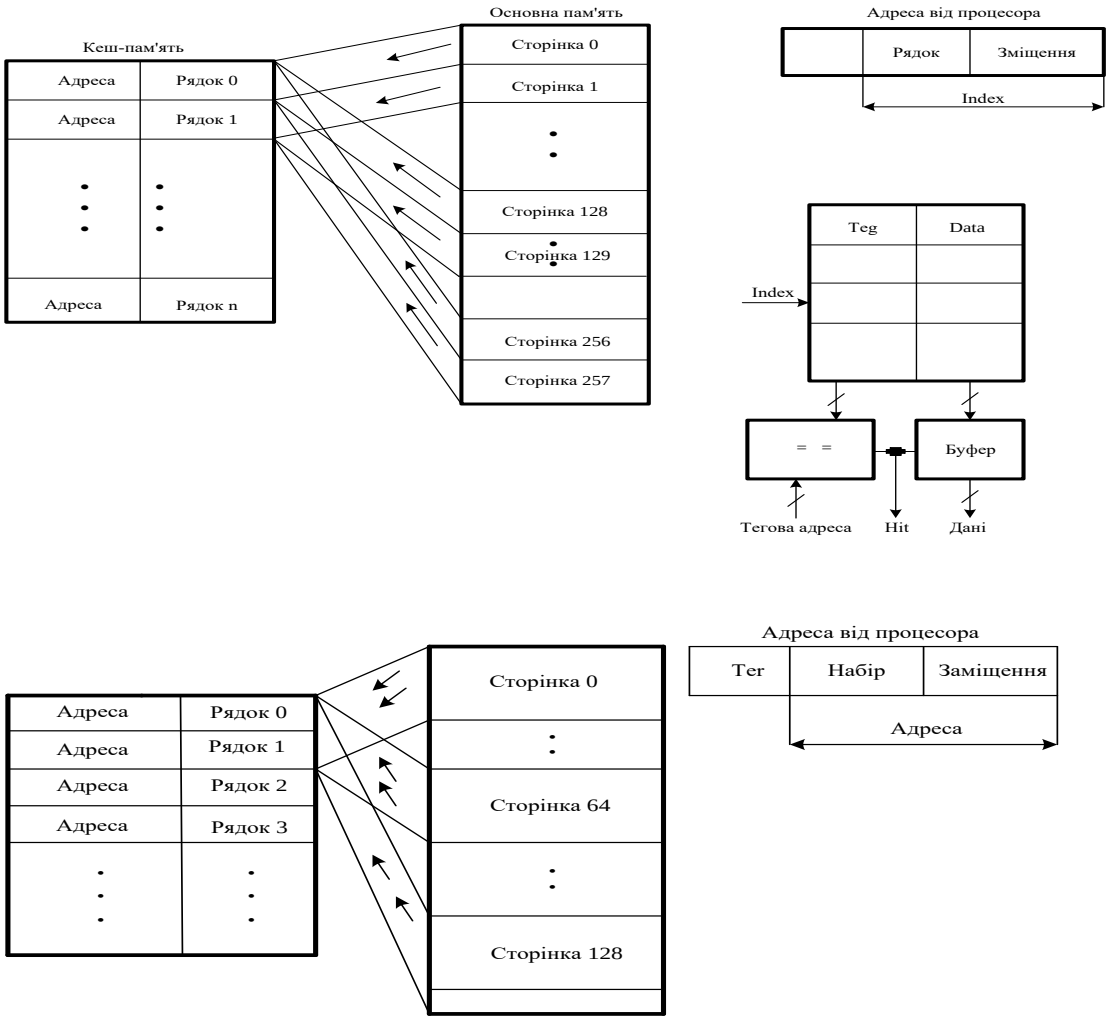
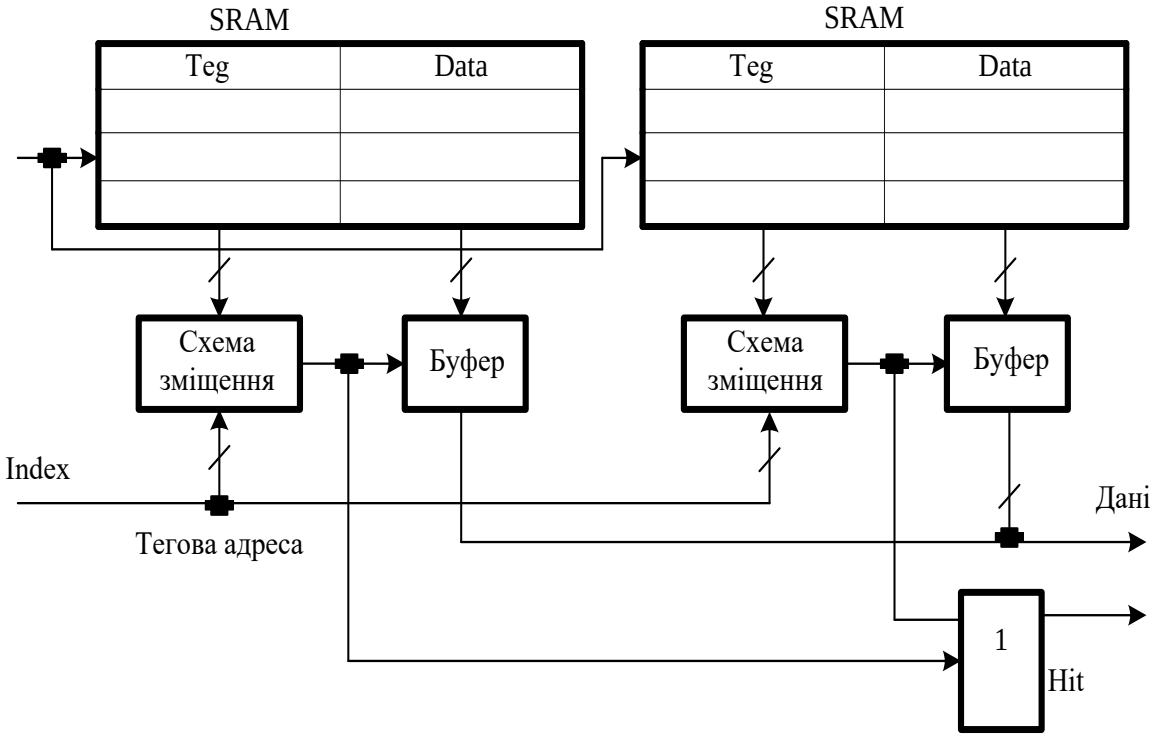


Fig. 1.11. Clarification of cache organization with random loading



b)



c)

Fig. 1.11. Explain the organization of the cache with direct placement (b) and collective-associative (c)

In other words, in this case the high-order bits of the address are treated as a mark, while the low-order bits are used for in-line addressing.

In a cache with direct allocation (with direct mapping), several pages of the main memory strictly correspond to one cache line (fig. 1.11, b). Since only one side can occupy a line at a time, this requires a special marker. The address from the processor is divided into three parts. The lower digits (offset) indicate the position of the word in the line. Middle digits allows you to select one of the cache lines. High digits create a tag. The line address is read. The address field of the numbered line is compared to the address labels and if there is a match, a hit signal is generated to send the information and then the word is selected from the data line by multiplexing. When loading from external memory, the entire line is replaced. It should be noted here that block transfers in modern systems are carried out quite quickly.

The direct embedded cache tag has a significantly reduced bit size. Usually the line number is the address of the page modulo the integer two. In fig. 1.11(c),

it is 128. The value of direct embedded cache i to save hardware cost. The disadvantage is the limitation of the number of pages in the cache, which can prevent the formation of an optimal page set, since moving a page to the cache removes another page that may be needed for the optimal page set.

An intermediate option in terms of complexity and performance between FACM structures and direct placement is a cache with association in several directions (association set). In this variant, several lines of the cache are combined into sets, and the middle bits of the memory address are determined not by one line, but by a set (fig. 1.11, c). The cache is divided into sets with a small number of rows that are multiples of two, that is, 2, 4, 8 ... etc. (in the figure, it is 2). The main page of memory can only be placed in the set which number is equal to the address of the page modulo (modulo 64 in this case). The position of the page in the set can be any. The comparison of the tags with the high-order bits of the address is performed only for the lines included in the set.

Different structures differ in the number of lines in the cache set: with two entries, with four entries, etc.

In this example two separate blocks of memory are used for even and odd lines. Even and odd lines (the words they contain) are highlighted at the same time. Reading starts with the block that coincides between the tag and the address tag. At the same time the addressed word is separated from the line due to the confusion. If there are no matches, the main storage is accessed and the line is replaced in one of the cache blocks.

The block diagram of the associative cache set is shown in figure 1.11 (c). Compared to a direct cache, an associative set cache has a longer flag (only one bit in the example provided). The ability to freely place pages in a set allows you to create a better composition of pages in the cache, since you can choose a page to replace. In modern microprocessor systems the cache memory of the first level, denoted L1 (from the English word Level (intraprocessor)), as a rule, has an associative set structure, and the cache memory of the second level L2 (external) - a structure with direct allocation. Many companies produce associative memory

chips. For example, one of the Cyrix chipsets has 4K lines, a 15-bit tag address, and a 16-bit output. Conventional SRAM is most often used in conjunction with cache controllers to create a cache.

The high-performance IBM Power 3 microprocessor uses an associative cache of 32 KB for instructions and 64 KB for data in 128 directions. A 256-bit bus is used in the Power 3 system to communicate with the second-level cache. L2 cache capacity is from 1 to 16 MB. By the way, MP Power 3 was used in the supercomputer that managed to defeat the world chess champion Garry Kasparov.

1.3. ROM (M), PROM, EPROM, EEPROM MEMORY DEVICES

ROM (non-volatile memory) storage devices store information that either does not change at all (ROM (M) and PROM types) or changes rarely and not in real time (EPROM and EEPROM memory types).

Information is written to the memory of the ROM mask (M) during the manufacture of microcircuits at industrial enterprises according to a template (mask) at the final stage of the technological process.

Flash PROMs are programmed after their production by an electronic industry enterprise in consumer laboratories without using complex technological processes. Simple devices (programmers) are used for this.

Non-volatile memory programming consists in one or another arrangement of communication elements between the horizontal and vertical lines of the matrix of memory elements.

ROM storage devices have a multi-bit organization (usually 8-bit or 4-bit, for some 16-bit chips) and are usually implemented according to the 2DM structure. The simplest SD can have a 2D structure. There are various technologies for the production of constant voltage converters – TTL, CMOS, n-MOS, etc.

1.3.1. Mask SD

The communication element in the SD mask can be diodes, bipolar transistors, MOP transistors, etc.

In the PROM diode matrix (M) (fig. 1.12, a) the horizontal lines are word sampling lines, and the vertical lines are reading lines. The read word is determined by the position of the light diodes in the nodes of the coordinate grid. If a diode is present, the high potential of the selected horizontal line is transmitted to the corresponding vertical line and a logical unit signal appears in this bit of the word. In the absence of a diode, the potential is close to zero, since the vertical line is connected to the ground through a resistor. In the displayed matrix the word 11010001 is read when sampling line Sh1 is violated (this word is stored in cell number one). In case of violation of Sh2, the word 10101011 (stored in cell No. 2) is read. The sample buses are the outputs of the address decoder, each address combination triggers its decoder output, resulting in a word being read from the addressed cell.

In a matrix with diode elements diodes are produced in some nodes of the matrix, but not in others. At the same time in order to reduce the cost of production, SDs try to change only one template during production, so that some elements of the connection are complete and functional while others are incomplete and missing. In MOSFETs the thickness of the sub-gate oxide is often increased in MOSFETs transistors, corresponding to zero memory, which leads to an increase in the cutoff voltage of the transistor. In this case the operating voltages of the SD are not able to open the transistor. The permanently closed state of the transistor is similar to it. he matrix with MOSFET transistors is shown in fig. 1.12 b.

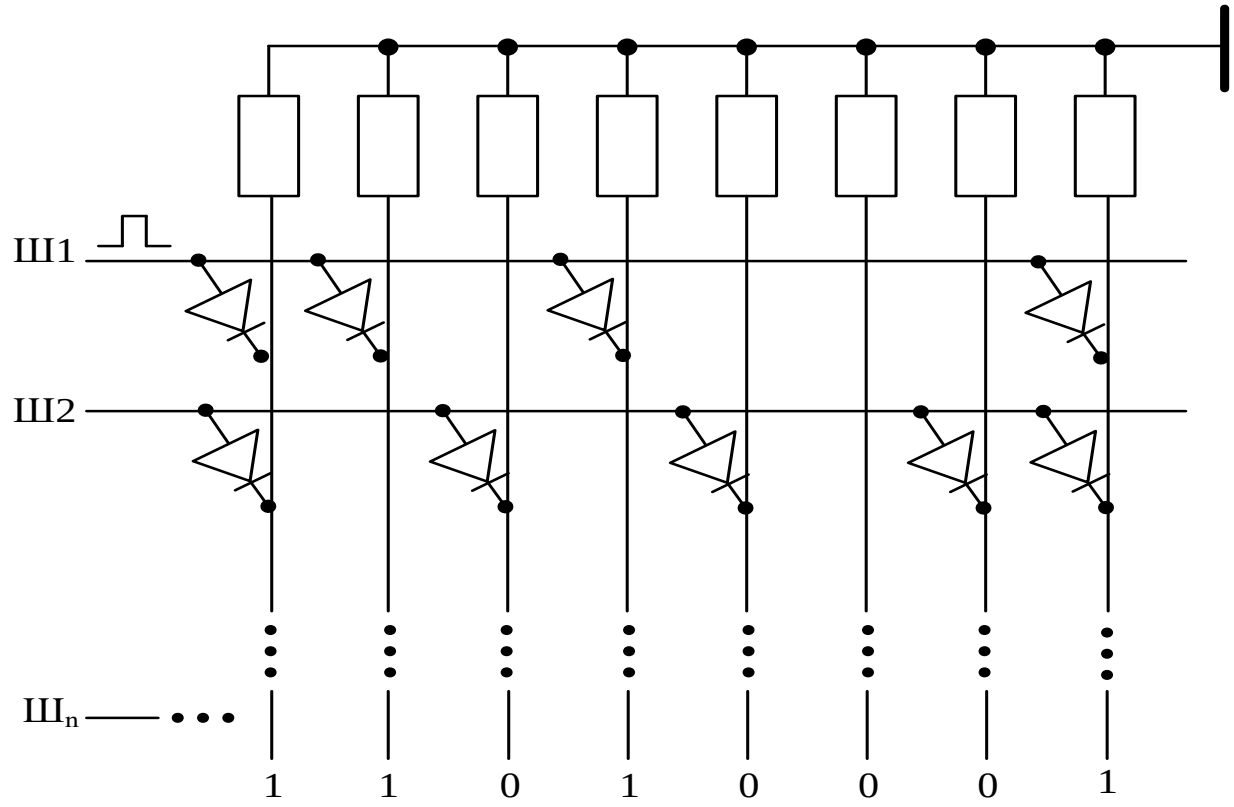


Fig. 1.12. Matrix of diode memory elements of the SD mask (a)

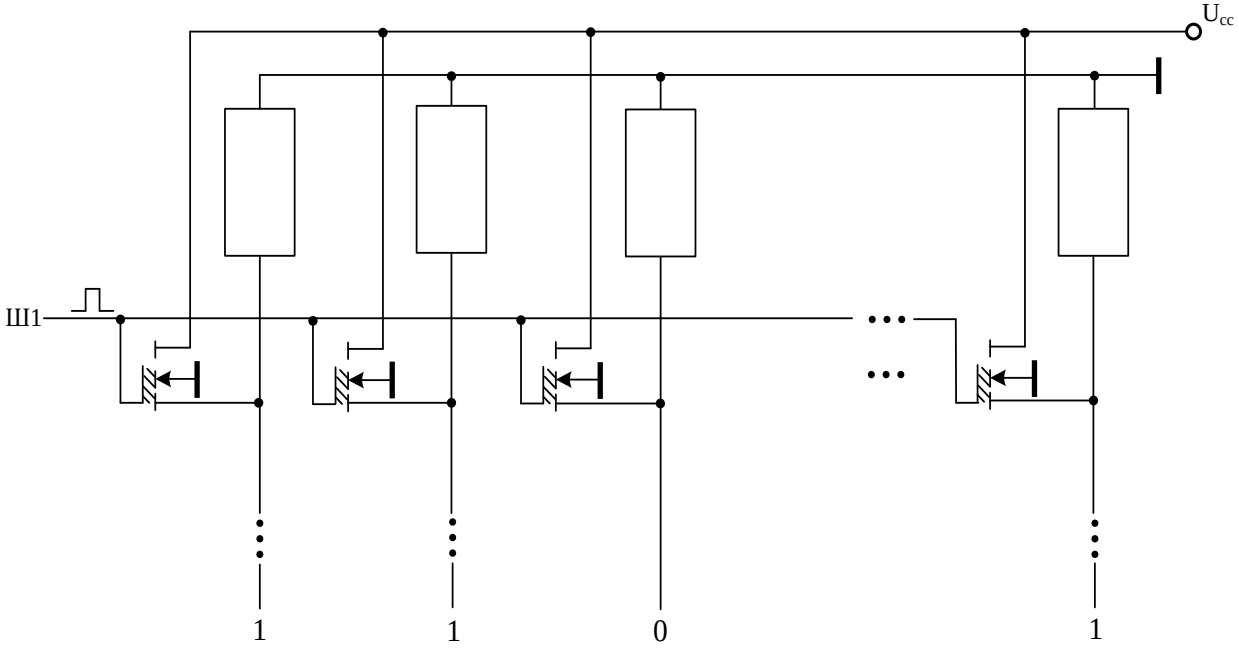


Fig. 1.12. Matrix of MOP elements

SD with mask programming is characterized by the compactness of memory elements, and therefore, a high degree of integration. With large volumes of

production, mask programming prevails, but with insufficient costs of SD, the costs of designing and manufacturing a template for technological programming of SD will be excessively high. Here you can see the scope of application of masked SD - storage of standard information that has a wide audience. In particular, masked SDs have letter codes of the alphabet (Russian and Latin), tables of common functions (sine, square, etc.), standard software, etc. as "firmware".

1.3.2. Software PROM

Microcircuits are programmed by removing or creating special jumpers in PROM. All jumpers are present (or absent) in the starting element. Only the necessary ones remain or appear after programming.

The elimination of some jumpers is characteristic of the SDs with floating jumpers (safety type). At the same time in the initial state the SD has all jumpers, and during programming some of them are eliminated by merging with current pulses of sufficiently large amplitude and duration.

In the SDs with floating jumpers these jumpers are connected to the electrodes of diodes or transistors. Jumpers can be metal (initially made of nichrome, later made of titanium-tungsten and other alloys) or polycrystalline (silicon). In the initial state the memory element stores a logical unit, and a logical zero must be written by melting the jumper.

Forming part of the jumpers corresponds to circuits that initially have non-conductive jumpers as a pair of diodes connected in reverse, or thin dielectric layers that break down during programming, creating a low resistance. The most compact and ideal are schemes with thin stamped dielectric anchors (fuse type). Their use is typical for programmable logic SBIs, which are discussed in section 1.The second type of PPROM memory elements are two diodes connected in opposite directions. In the initial state, the resistance of such a circuit is so large, which practically corresponds to an open circuit, and the memory element stores a logical zero. To write a unit an increased voltage is applied to the diodes, which penetrates the diode

shifted in the opposite direction. The diode breaks, creating a short circuit in it and acts as a conductive jumper.

Memory elements with floating jumpers and pairs of diodes are shown in fig. 1.13, in the original state and after programming.

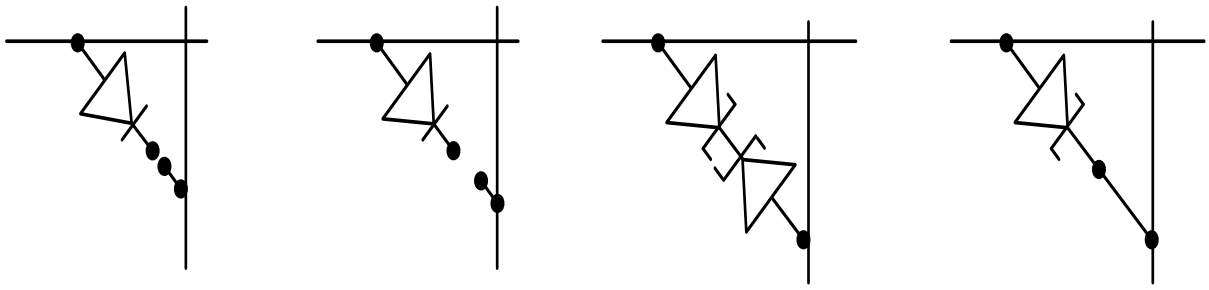


Fig. 1.13. Memory elements with floating jumpers and diode pairs

The matrix of SD memory elements with floating jumpers using TTL technology (K155REZ microcircuits) is shown in figure 1.14. SDP has a 32x8 organization. The matrix contains 32 transistors with 9 emitters (8 working and one technological for determining the ignition mode, the technological emitter is not shown in the figure). A high potential on either sample bus activates the corresponding emitter-follower transistor. Before programming, transistors transmit a high base potential to all output (bit) lines, that is, words consisting of the same units are written under all addresses. Burnout of the jumper in the circuit of any emitter gives a zero in the specified bit of the word, for example, for cell No. 1, it is possible to program a write to this address of the words 10100101. SD connected to external circuits through buffer cascades with outputs OK or TK has a 2D structure.

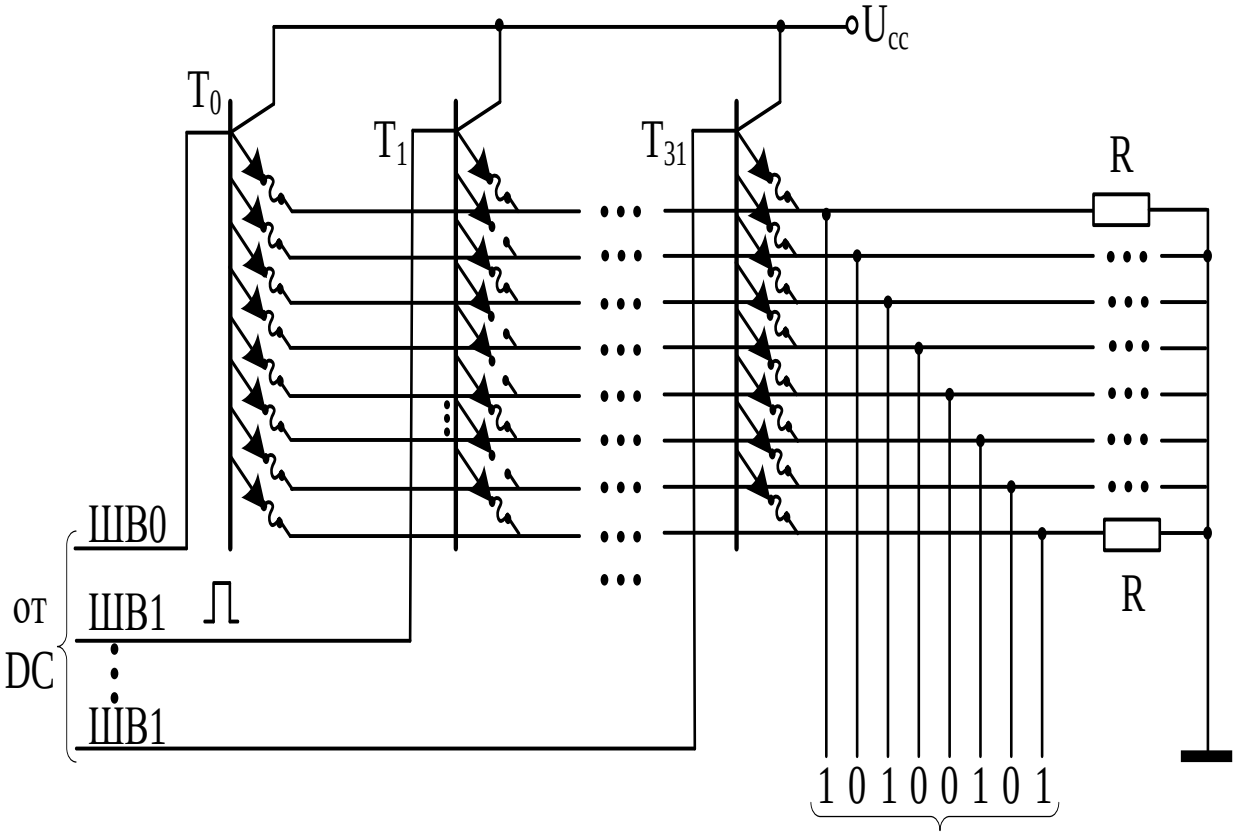


Fig. 1.14. Matrix of memory elements with floating jumpers in TTL technology

The programming of SDs with the floating jumpers is implemented using simple hardware and can be accessed by circuit engineers even without special equipment. Figure 1.15 shows a multi-emitter transistor (BET) with floating jumpers and additional elements that provide programming of the SD. The outputs of this memory element are transferred to external circuits through buffer cascades with three states, the operation of which is turned on by the OE signal. At the same time there is no signal that allows the operation of the OE programming pulsers, and they do not affect the operation of the system. During programming the data buffers are transferred to the third state (OE = 0) and the operation of the F sections is allowed. The word that is written into this cell is specified in the data lines D7 ... D0. Those bits of the word that have ones will have a low voltage level at the outputs of the profiles. The corresponding MET emitters will be at low voltage and the jumper burning current will pass through them.

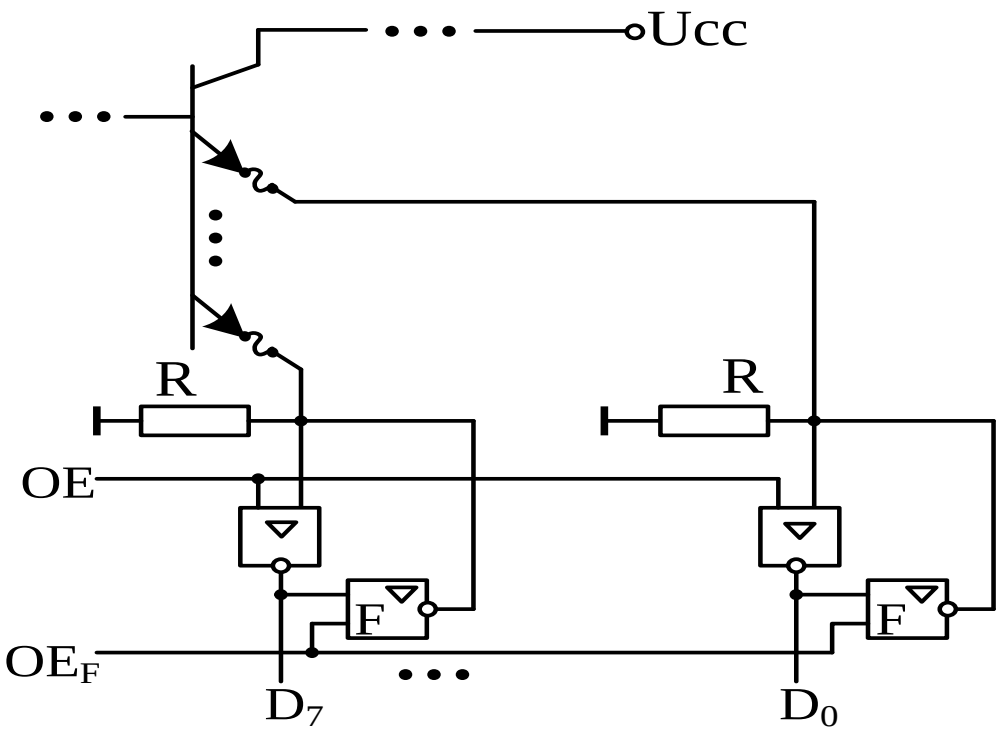


Figure 1.15. Scheme of a memory cell

During reading the absence of the jumper will give a null signal to the input of the data buffer. Since the buffer is inverted, one signal will be removed from its output, that is, the one that was recorded. Addressing of a programmable cell is usually provided by an address decoder that provides a high potential level for the base of the addressed MET.

To burn the jumpers, currents of tens of milliamps are applied to them in the form of a series of pulses (for greater reliability of burning). Not all jumpers can be burned correctly, the programmability coefficient, for example, for the K556 series is 0.5 ... 0.7. It is possible to restore the conductivity of the armature after some time due to migration in electrical materials in the SD with floating armature.

Fusible connectors occupy a relatively large amount of space on the crystal, so the degree of integration of SD with such anchors is much lower than that of SD masks. At the same time the ease of user programming and the low time costs have led to the widespread use of PROM software. The low cost of user-programmable

software is due to the fact that the manufacturer makes the chips without taking into account the specifics of the software, that is, it is exempt from the design according to special orders as well as the associated costs.

The K556 series microcircuits which have an information capacity of 1 ... 64 Kbit and an access time of 70 ... 90 ns, take the leading place among domestic electronic devices.

The external organization of ROM (M) and PROM is simple: the input signals for them are the address code and the chip selection signal CS. In time, the sequence of signals is as follows: first the address code is entered (so that the address is decoded and access to an unpredictable cell is excluded), then the CS chip select signal is received and after a delay caused by the circuit, the speed, at the data outputs are set to the correct values of the read signals.

1.3.3. Memory types EPROM and EEPROM

In reprogrammed EPROM and EEPROM (or E2PROM) memory cards, the old information can be deleted and replaced with new information as a result of a special process during which the memory card is brought out of the operating mode. The operation mode (reading data) is a process that occurs at a relatively high speed. Replacing the memory contents requires significantly longer operations.

According to the method of erasing old information flash drives with ultraviolet erasing and electric erasing (E2PROM) are distinguished.

The memory elements of modern RZPs are MNOP and LIZMOP transistors (the addition of LIZ to the designation MOP comes from the words Avalanche Charge Injection).

The MNOP transistor differs from the conventional MOP transistor in that it has a double-layer sub-gate dielectric. On the surface of the crystal is a thin layer of silicon dioxide SiO_2 then a thicker layer of silicon nitride Si_3N_4 and then the diaphragm (fig. 1.16, a). Centers of charge capture appear at the boundary of the dielectric layers. Due to the tunnel effect, charge carriers can pass through a thin oxide layer with a thickness of no more than 5 nm and accumulate at the boundary

of the layers. This charge contains the information stored in the MNOP. The charge is registered by creating an electric field strength under the diaphragm sufficient for the tunneling of charge carriers through a thin layer. A charge of any sign can be formed at the contacts SiO_2 of the dielectric layers, depending on the direction of the electric field inward. The presence of charge affects the threshold voltage of the transistor.

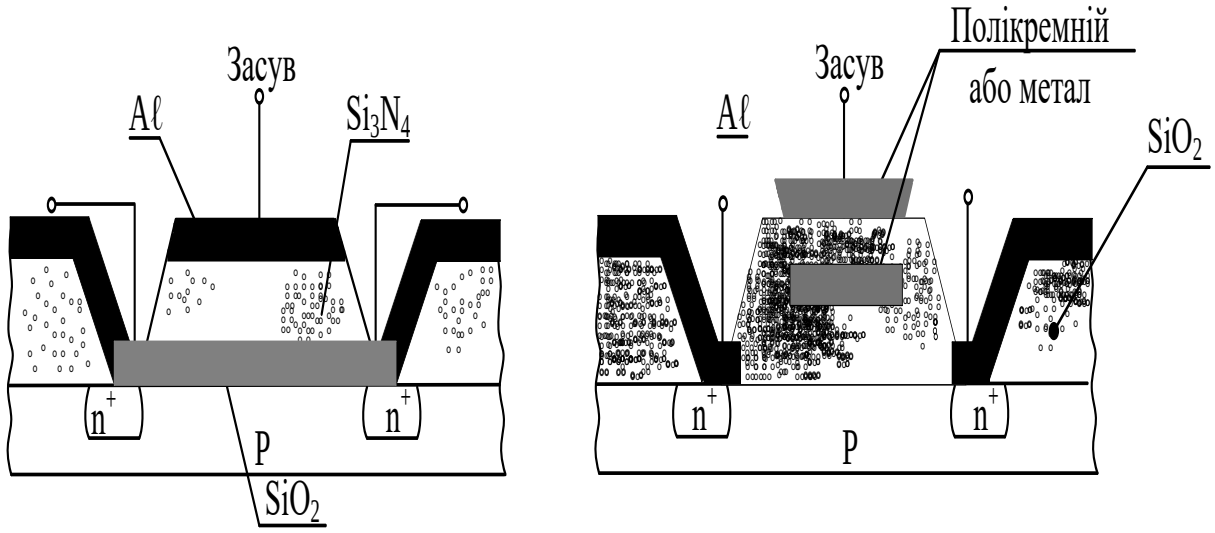


Fig. 1.16. Structures of MNOP (a) and LISMOP (b) transistors with two gates

For an MNOP with a channel, the negative charge at the contact of the layers increases the threshold voltage (it screens the action of the positive voltage on the gate and turns off the transistor). At the same time, the threshold voltage rises so that the operating voltage on the gate of the transistor is unable to open it (to create a conductive channel in it). A transistor that has no charge or has a different sign is easily opened by the value of the operating voltage. Thus, a bit is stored in the MNOP: one of the states is interpreted as representing a logical unit, the other as a zero.

Relatively high voltages, about 20 V are used when programming the SDs.After removing the high voltages, the tunneling of charge carriers through the dielectric stops, and the limit voltage set for the transistor remains unchanged.

After $10^4...10^6$ overwriting, MNOP no longer keeps the charge stable. RPZP on MNOP transistors are non-volatile and can store information for months, years and decades.

Before new storage, old information is deleted by entering zeros in all memory elements. Type of SD - RPZP-ES.

LISMOP transistors always have a so-called floating gate, which can be the only one or additional to the usual (steering) gate. Single-gate floating transistors are used in SD-type RPZP-UV, and dual-gate transistors are suitable for use in both RPZP-UV and RPZP-ES. Let us consider a more modern two-gate LISMOP transistor.

The principle of operation of the LIZMOP with a double gate is similar to the MNOP transistor - there is also an area between the control gate and the channel area, where a charge can be entered during programming,that affects the value of the threshold voltage. Only the charge introduction area is not a boundary between the dielectric layers, but a conductive area surrounded on all sides by a dielectric (most often made of polycrystalline silicon), which, like a trap accumulates a charge that can be stored for a very long time. This zone is called a floating gate.

When a positive voltage pulse with a relatively large amplitude of 20 ... 25 V is applied to the control gate, an avalanche breakdown occurs in reverse-biased pn junctions, the region of which is saturated with electrons. Part of the electrons with sufficient energy to cross the potential barrier of the dielectric region passes through the floating gate. Removing the high programming voltage restores the transistor regions to their normal state and blocks the electrons in the floating gate, where they can remain for a long time (many years in high quality devices).

A floating gate charged with electrons increases the threshold voltage of the transistor to such an extent that a conducting channel is not formed in the transistor in the operating voltage range.

When there is no charge on the floating gate, the transistor works in the usual key mode.

Destruction of information can be carried out in two ways - ultraviolet radiation or electrical signals.

In the first case, the IMS housing has a special transparent window for crystal irradiation. Silicon dioxide and polysilicon are transparent to ultraviolet rays. These rays create photocurrents and thermal currents in transistor regions making regions of the device conductive, and the charge leaves the floating diaphragm. The operation of erasing information in this way takes several tens of minutes, the information is immediately erased all over the crystal. In schemes with UV erasure, the number of reprogramming cycles is significantly reduced, that is, under the influence of ultraviolet rays the properties of materials gradually change. The number of overwriting cycles in domestic integrated circuits is 10 ... 100.

An electrical erasure of information is carried out by applying a low (zero) voltage to the control valves and a high programming voltage to the drains. An electric erasure has advantages: you can erase information not from the entire crystal, but selectively (separately for each address). The duration of the "erase-write" process is much shorter, the limit on the number of reprogramming cycles are significantly relaxed (10^4...10^6 such cycles are allowed). In addition, you can reprogram the software without removing the microcircuit from the device in which it works.In this case, electric erasure circuits take up more space on the crystal, so the degree of their integration is lower, and the cost is higher. Recently, these shortcomings are rapidly being overcome, and UV erasure has been replaced by electric erasure.

The predecessors of the two-gate LISMOP transistors were single-gate transistors that had only a floating gate. These transistors were usually made with a p-channel, so injecting electrons into the floating gate created a conducting channel in the transistor, and removing the charge caused the channel to disappear. When using such transistors, the memory elements consist of two transistors connected in series: a common type key MOSFET for sampling the addressed element and a LISMOP which state is determined by a stored bit. Information is erased by ultraviolet rays.

The connection of two-frame LISMOP transistors with sampling lines and reading lines in SD matrices is shown in fig. 1.17. A logical zero is registered by loading the floating gate with the injection of "hot" electrons in the programming mode. Erasing information, which is understood as removing the charge from the floating gate leads to the preservation of logical units in all memory elements, that is, in this case, the studied transistors open and also transmit voltage to the read line.

Among domestic RPZP-UV (designated with the letters RF) the most famous is the K573 series with a wide range of denominations, and among RPZP-ES (designated with the letters RR) — the KR558 series (based on p-MNOP), K1609, K1624, K1626 for LIZMOP with two gates.

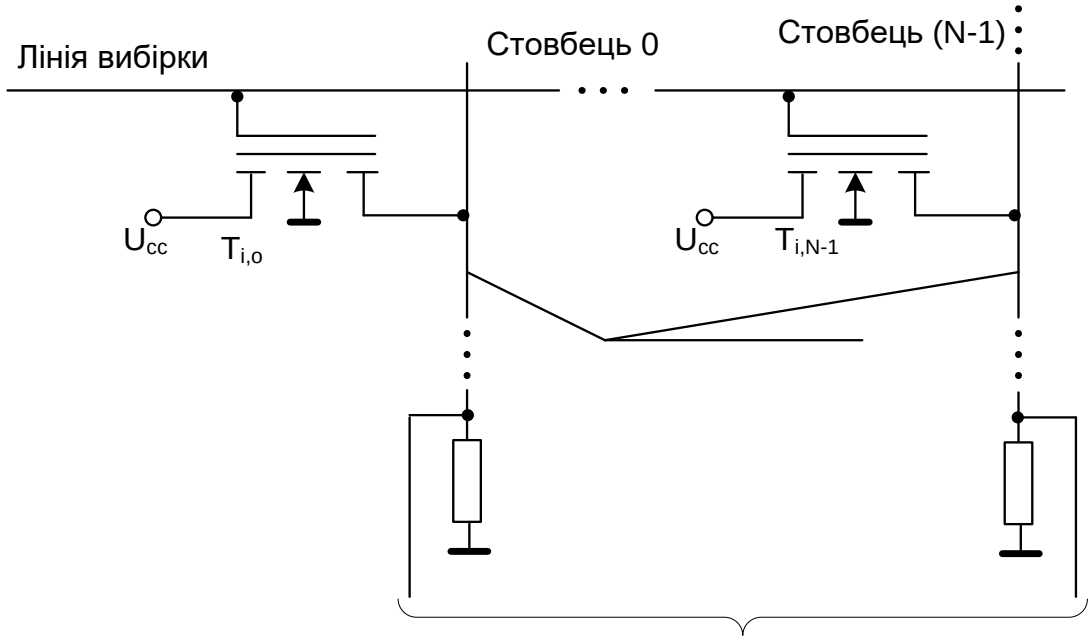


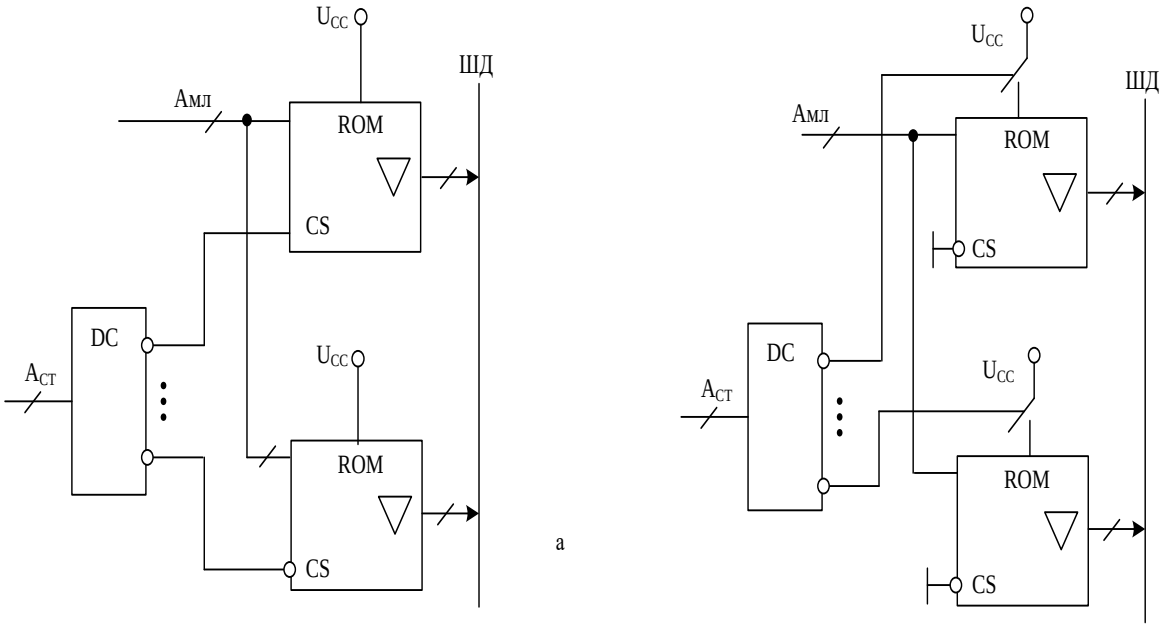
Fig. 1.17. The scheme of connecting LIZMOP transistors with a double gate to the line of sampling and reading in the RPZP

Currently, domestic ROMs are characterized by the following parameters: mask integrated circuits have an information capacity of up to 1 Mbit with an access time of about 200 ns, microcircuits with floating jumpers of 64 Kbit and 80 ns, respectively, RPZP-UV 1 Mbit and 350 ns. ns, RPZP-ES 64 Kbit and as 250 ns.

At the level of world technology, there is a memory card of the RPZP-UV type with an information capacity of up to 8 Mbit at an access time of 45 ns (Atmel) a memory card of the RPZP-ES type with an information capacity of up to 256 Kbit at 90 ns access time and an acceptable number of cycles overwriting 10^5 at a time. More than 10 years of data retention. This SD uses a single 5V power supply and draws 2mA when active and 100μA when no contacts are present. Writing bytes or pages is possible in 3 ms (SGS-Thomson).

1.3.4. ROM pulse power

The energy independence of all memory elements which store information when the power is turned off opens up the possibility of saving energy during their operation, thereby improving their thermal regime, which increases the reliability of circuits. Power can only be applied to the currently addressed IS. Figure1.18 shows a conventional variant of building a memory module consisting of several integrated circuits and a variant with a pulsed power supply. In the usual version the voltage is constantly connected as additional to all ICs, and the selection of addressed ICs is carried out by the CS signal.



a) and b)

Fig. 1.18. Non-volatile memory modules with normal (a) and pulsed (b) power supply

In the variant with pulsed power constant operation of all ICs at the SC inputs is allowed, but power is supplied only to the selected microcircuit by a key controlled from the outputs of the address decoder, which decodes the higher bits of the address.

The power switching mode can repeatedly reduce the power consumption of the module, but at the same time, it increases the access time of the integrated circuit for single random accesses, that is, it takes some time to establish the IC mode after the power is turned on.

There is no time loss when reading data from nearby addresses, while the high-order bits of the address remain unchanged.

1.4. FLASH MEMORY

Flash memory in terms of the type of memory elements and the basic principle of operation is similar to memory of the type E^2PROM however, a number of architectural and design features make it possible to distinguish it in a separate class. The development of flash memory is considered the culmination of a decade of development of electrically erasable memory circuits.

Flash memory schemes do not provide for the deletion of individual words, the deletion of information is performed either for the entire memory at the same time, or for sufficiently large blocks. It is clear that this allows for simplification of the SD schemes, that is, it helps to achieve a high level of integration and speed while reducing costs. Technologically advanced flash memory chips are made of high quality and have very good parameters.

According to one version the term Flash is associated with a characteristic feature of this type of memory - the ability to erase its entire volume at once. According to this version even before the appearance of Flash memory devices were used to store secret data that automatically erased stored information when unauthorized access was attempted and were called Flash devices (flash, moment).

This name was transferred to memory, which had the property of quickly erasing the entire data table with a single signal.

Deleting all order information at once is the easiest way, but has a drawback: even replacing one word in the order requires deleting and re-entering the entire order. This is inconvenient for many applications. Therefore, along with the schemes with simultaneous erasure of all contents, there are schemes with a block structure, in which the entire memory table is divided into blocks that are cleared independently of each other. The size of such blocks is very different: from 256 bytes to 128 Kbytes.

The number of Flash reprogramming cycles, although large, is limited, that is, the cells "wear out" during rewriting. To increase the durability of the memory, special algorithms are used in its work, which help to "equalize" the number of rewrites on all blocks of the microcircuit.

Depending on the field of application, flash memory has architectural and technical variations. Two main areas of effective use of flash memory are storage of data that does not change very often (especially updated programs) and replacement of memory with magnetic disks.

For the first direction, due to the rare update of the content, the parameters of the erase and write cycles are not as important as the information capacity and speed of information reading. Erasures in these charts can be simultaneous for all memory and block erasures. There are schemes with specialized blocks (asymmetric block structures) among devices with erasing blocks. By the name of the so-called bootblocks, the information in which is reliably protected by the hardware from accidental erasure, these flash drives are called bootblock flash memory. Boot blocks store system initialization programs that allow the system to be put into a working state after power-on.

Chips for replacing a magnetic hard disk (Flash-File memory) contain more advanced ways of rewriting information and have identical blocks (symmetric block structures).

One of the elements of the Flash memory structure is a disk (matrix of memory elements). Two trends are developing in memory device designs: one based on OR-IT(HI) cells and the other based on AND-NOR cells (NAND).

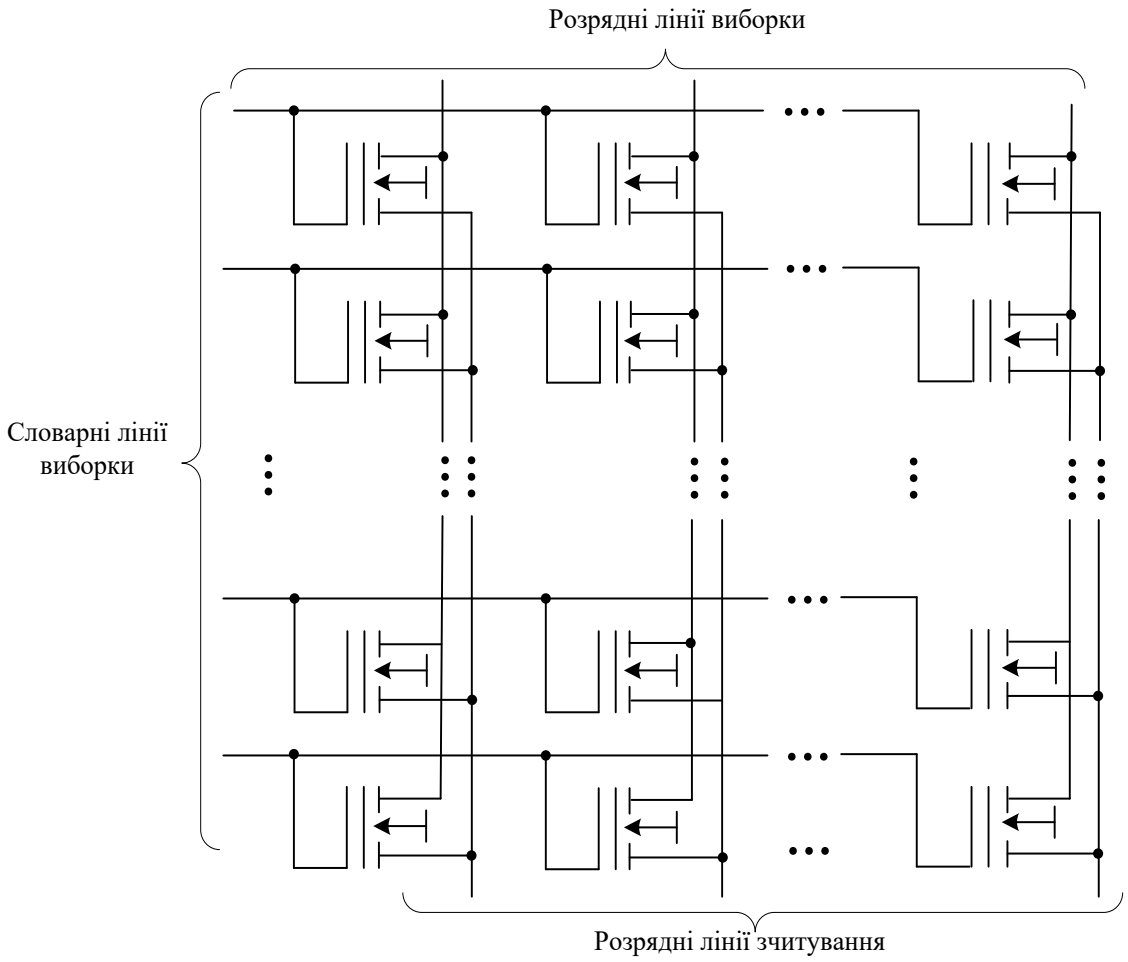


Fig. 1.19. The structure of the flash memory matrix is based on OR-NOT cells

Accumulator batteries based on OR-HI cells (parallel LIZMOP double frame transistors) provide fast random access to words. They are acceptable for a variety of applications, but the most obvious is their use in memory to store data that is rarely updated. At the same time, there is useful continuity with previously used ROMs and EPROMs, the typical control signals are retained, providing random

sample readout. The structure of the memory matrix is shown in fig. 1.19. Each column is a set of transistors connected in parallel. Sampling bit lines have a high potential. All transistors of unselected lines are closed. In the dedicated line the transistors open and transmit a high level of voltage to the line of reading discharges, the floating gates of which do not have an electronic charge and therefore the threshold voltage of the transistor is normal (not increased).

OR-NO memory devices are widely used by Intel. There are some opinions about the competitiveness of these drives in applications related to the replacement of magnetic hard drives with Flash memory.

Structures with AND-NOT cells are more compact, but do not provide a random access mode and are practically used only in schemes for exchanging magnetic disks. In the diagrams on these cells the disk itself is more compact, but the number of logical elements surrounding the disk increases.

Various means and methods are used to improve technical and economic indicators in flash memory circuits:

1. Interruption of writing processes while reading the processor (Erase Suspend). Without it, there would be long periods of CPU downtime, as writing takes a long time. After an interruption, the recording process resumes under the control of the internal flash memory.

2. The internal command queue which controls the operation of Flash memory allows you to organize the pipeline of operations and speed up the reading and writing processes.

3. Programming the length of words stored in SD according to different I/O ports.

4. Introduction of reduced energy consumption modes in the absence of reports on the SD including the deep rest mode, in which the power is reduced to extremely low values (for example, the current consumption is reduced to 2 μA). These features are very important for devices with autonomous (battery) power.

5. Possibility of adaptation to work with different voltages (5 V, 3.3 V, etc.). The circuit itself "senses" the power level and makes the necessary switches to adjust to it.

6. Introduce page buffers into memory structures to quickly collect new data for writing. Two such buffers can operate in a mode called "ping-pong", where one of them receives the words to be written, while the other ensures that its contents are kept in memory. When the first buffer is full, the second one will be freed and they will swap places.

7. Different means of protection against accidental or unauthorized access.

Addressable flash memory focused on storing information that does not change very often, can have all information erased at once (bulk erase architecture) or block erasure (load block flash architecture).

Continuity with SDP types E^2PROM and EPROMs developed earlier. Flash circuits are better E^2PROM in terms of information capacity and cost in applications where erasing of individual words is not required, and compared to EPROM they have the advantage that they do not require special conditions and equipment for erasing data, which is also much faster.

1.4.1. Memory type mass erasure

Intel Bulk Erase memory, the most famous among flash memory manufacturers has a byte write time of about 10 μs, allows erase 10 cycles, programming voltage is 12 V ± 5%, active mode current is about 10 mA, standby mode is about 50 μA. Access time for reading is approximately 100 ns, erasing time and programming time of the whole crystal is 0.6 ... 4 s for crystals with a capacity of 256 Kbit ... 2 Mbit.

In contrast to traditional control of memory circuits using address and control signals Flash memory has additional control over the command words written by the processor in a special register, it works only at a high voltage level at the output of the microcircuit, which is marked U_{pp} (programming voltage). In the absence

of this layer, the system functions as read-only memory under the control of conventional signals that determine read, power-off, third-state verification and ID issuance operations.

Figure 1.20 shows the structure of Bulk Erase Flash (Intel Circuits 28F010, 28F020, etc.).

The A inputs are addressable and during the write cycle the addresses are set in the latch register by the STB signal. Data input/output (DQ lines) is performed through third-state buffers. During a write cycle data is latched into a latch register.

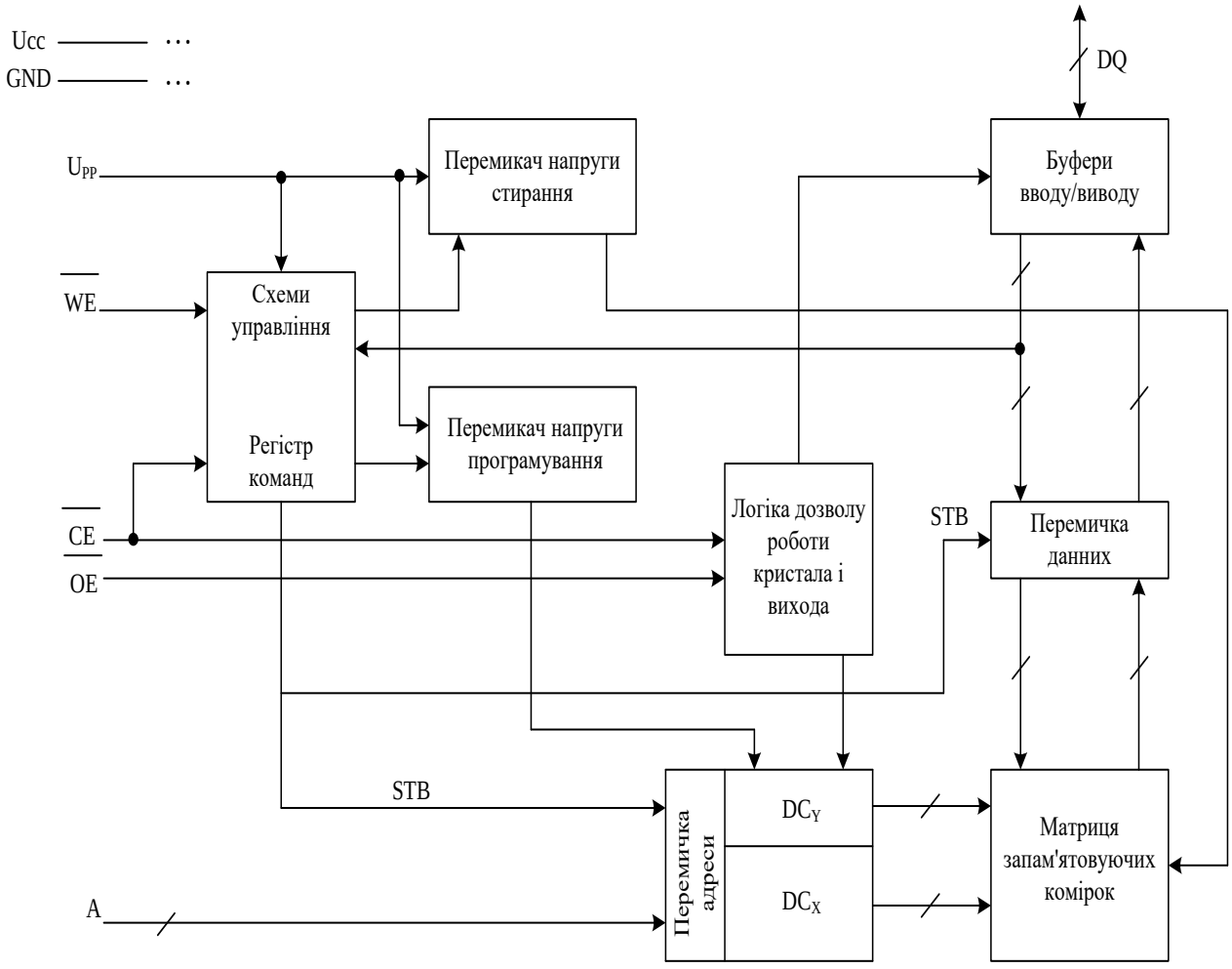


Fig. 1.20. Flash memory structure with simultaneous deletion of data from the entire crystal (Bulk Erase type)

CE, OE and WE signals are L-active. The CE signal activates the control logic, I/O buffers, address decoders DCy, DCx and read amplifiers. When the CE

signal is high (no pattern selected), the buffers go into the third state and power consumption drops to the standby level.

A low level of the OE signal allows the buffers to output data during read cycles (only when the CE signal is low, of course).

The WE signal allows you to write to the command register and the matrix of memory cells and loads fixed registers with its edges (negative - address fixing register, positive - data).

The control circuits and the contents of the command register determine the state of the voltage level switches that U_{pp} are used in different modes of operation (cleaning, programming or reading).

To receive output data during reading the active state of the CE and OE signals is required. At the same time a low U_{pp} level allows reading data only from the memory, and a high level allows reading identification codes and information to verify the erase / programming operations. Operations with the ID output the manufacturer code and the chip. This information allows for the coordination of circuit erasure and programming algorithms, as well as hardware programming that is performed automatically.

Identification codes are located in two memory cells and can be read using a certain combination of signals or the instruction register (reading after entering the 90H instruction register).

During a write operation the address and data codes are set in the internal latch registers. The same operations are performed at a high level, in addition memory erasure and programming are allowed. U_{pp} All actions related to changing the contents of memory are performed using the command register. The command register does not occupy any position in the address space and is loaded by a normal write cycle from the processor at a low level U_{pp} . Its contents act as input to the internal machine to control the memory erase and programming circuits. Seven commands are used, two of which define read operations (data and identification codes), the other two refer to erase operation (prepare for erase / erase and verify

erase), two commands refer to programming operation (prepare for programming / program and verify programming) and one command specifies a chip reset operation.

When the level drops, U_{pp} command register is reset, allowing the chip to perform read-only operations.

The clear command deletes all the bytes of the matrix in parallel, after which they must all be checked. For this the bytes are addressed and activated by applying a special voltage. A read from the OFFH code cell indicates that all bits of the byte are cleared. If another code is read, the erase operation is repeated. The check then resumes from the address of the last byte to be checked. The verification process continues until the last address is reached.

Memory is programmed by byte (sequentially or with random access). The read cycle from the processor outputs byte data, which is compared with the given one. Equality of bytes indicates successful programming. The programming process then proceeds to the next byte.

The reset command is a way to reliably cancel a clear/program command. After each of these commands a reset operation code can be entered in the command register, which will exclude the possibility of any actions related to the specified command. The contents of the memory cannot be changed. To further bring the diagram to the desired state, you need to enter the appropriate command in the command register.

When the SE signal reaches a high level, it goes into reduced power mode. If this occurs during cleaning, programming, or data verification, active current is maintained until certain operations are completed.

1.4.2. Flash memory with asymmetric block structure

Flash memory boot block (BBFP) circuits are characterized by block erasure and asymmetric block architecture. Blocks are specialized and come in different sizes. Among them is the so-called boot block (BB), the contents of which are

hardware protected against accidental deletion. The BIOS stores the Basic Input/Output System (BIOS) software that is required for proper system operation and initialization.

Blocks also include BP (parameter blocks) and GB (main blocks), which do not have hardware write protection. Power supplies store relatively frequently changed system parameters (identification codes, diagnostic programs, etc.). GB blocks store basic control programs, etc.

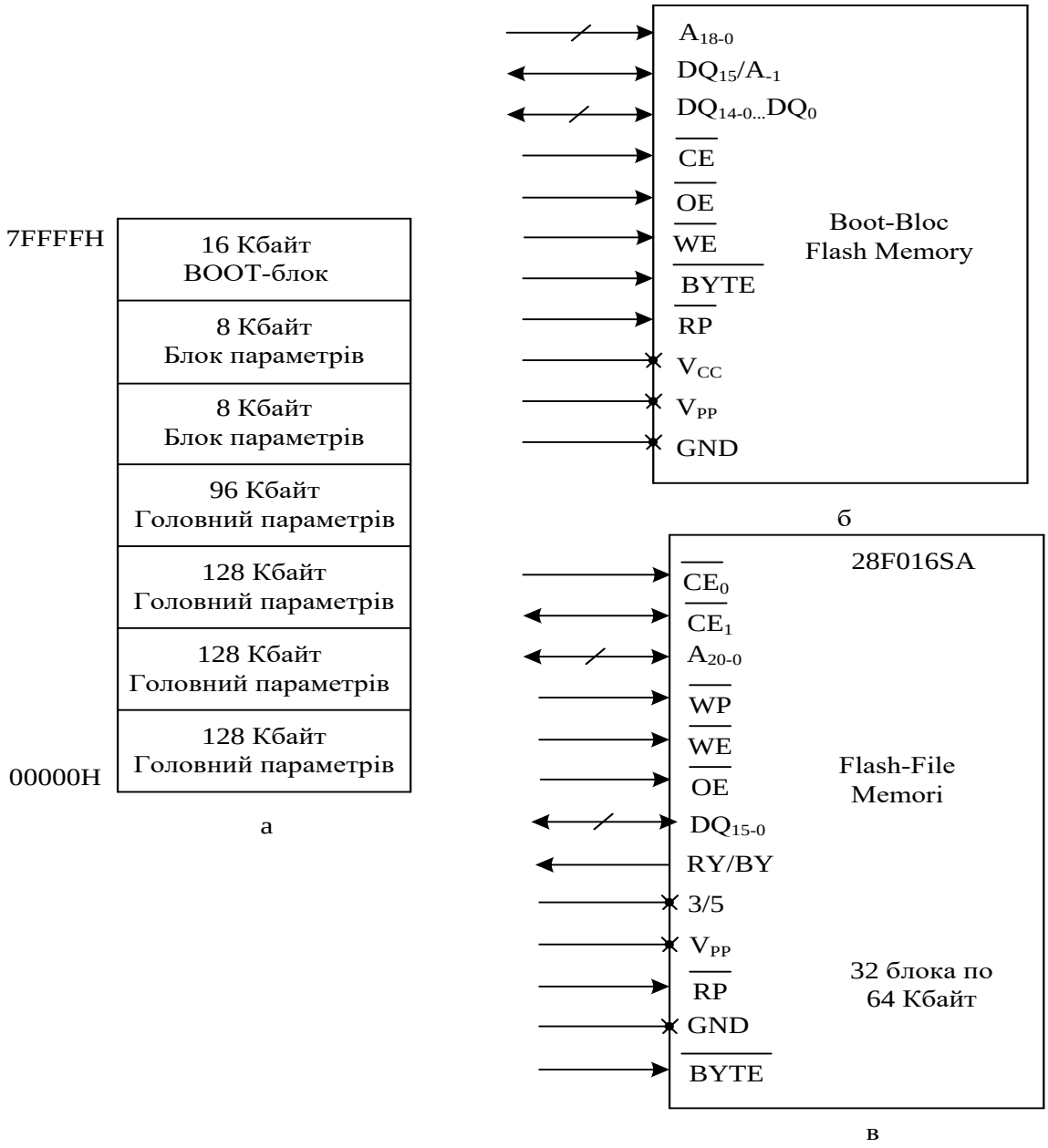


Fig. 1.21. Allocation of address space and external organization of flash memory with asymmetric block structure (a, b) and external organization of file flash memory (c)

Chipsets Boot-block Flash memory are designed to work with different microprocessors and to match them have two variants of the location of BB in the address space: upper and lower, which is reflected in the IC designation by the letters T (Top) or B (bottom). Figure 1.21, for example, shows a memory map (allocation of address space) for a 4 Mbit integrated circuit with a top layout.

BBFPs with a capacity of 1...16 Mbit are currently being produced, and integrated circuits with an information capacity of up to 256 Mbit are expected in the next generations.

In terms of operation, BBFPs are similar to Bulk Erase memory, in both types of erase/program ICs they are controlled by an internal machine which input is commands entered from the processor. In BBFP schemes this role is performed by the so-called Command User Interface (CUI).

The external organization of a typical BBFP is shown in figure 1.21, b on the example of an integrated microcircuit with an information capacity of 4 Mbit.

Addresses are specified using a 19-bit code A_{18-0} which means that up to 512 characters are stored in memory. The BYTE signal specifies the organization of 8-bit or 16-bit memory. In byte organization bytes are transmitted along a line DQ_{7-0} and a line DQ_{15} acts as the least significant bit of the address A_{-1} which determines which byte of a given cell is transmitted (high or low). Conclusions DQ_{15-0} are data input-output lines when organizing the dictionary.

The output voltage at the RP (Reset / Power Down) can have three levels: 12V ± 5%, logic unit level H and low level L. At 12V ± 5%, BB is open, erase and programming operations can be performed in it. The battery closes at a voltage below 6.5 V.

With many power-saving modes BBFP circuits, in particular, implement an automatic power-saving (APS) mode, by which, after the completion of a read cycle the circuit automatically enters a static state with a current consumption of about 1 mA, where it remains until the start of the next read cycle.

With an unselected scheme (with a high signal level at the CE output and the RP output, that is, CE = RP = H), the power consumption drops to the resting level (tens of μA). When RP = L, registration is not only prohibited, but also a mode of deep power reduction is introduced, in which the consumption current is reduced to fractions of μA.

Active mode corresponds to the combination of signals CE = L and RP = H. Signals OE and WE have a normal purpose. Flash memory Starter microcircuits can work with different supply and programming voltages (Smart Voltage technology) and have a read access time of 60 ... 70 ns, currents in active mode 15 ... 25 mA and very small currents in deep mode power reduction. (about 0.2 μA).

1.4.3. File flash memory

Flash File Memory (FFP) occupies an important place in the memory hierarchy. For many years, microcomputer storage of large volumes of data relied on well-designed and relatively inexpensive external hard drives based on magnetic and later optical disks. In many computers the memory system is organized as a combination of a magnetic hard disk with dynamic semiconductor RAM.

Having significant advantages, SDs drives as electromechanical devices also have a number of disadvantages: sensitivity to shocks and vibrations, pollution, limited speed and significant energy consumption. These disadvantages are especially noticeable in portable devices with autonomous (battery) power. It is enough to note that storage devices consume a maximum of about 3 W of power, which in systems with a supply voltage of 3.3 ... 5 V means a current consumption of 0.6 ... 0.9 A, which quickly discharges the batteries.

File flash memory is designed to replace hard disks, which reduces energy consumption hundreds of times, simultaneously increases the mechanical strength and reliability of hard disks, reduces their size and weight, increases the speed of reading data by several orders of magnitude, while maintaining software compatibility with management memory tools. However, disk storage still has advantages in terms of information capacity and cost.

The use of FFP to replace disk storage in portable computers is one of the most important factors contributing to the development of this direction. At the same time the traditional combination of "hard disk - dynamic RAM" can be replaced by the combination of "flash memory - static RAM". In this case, the program commands written in the FFP are read directly by the processor, the results are also written directly in the FFP, and intensive computational operations that require the fastest possible access to memory and writing data with a byte resolution are performed using yati high- speed static memory.

The FFP drive is divided into blocks that are analogous to the sectors of a magnetic disk opened in the MS-DOS operating system. Software tools have been developed that provide exchange between Flash blocks similar to how the MS-DOS operating system provides exchange between disk sectors.

FFP blocks are identical and have the same information capacity (symmetric block architecture). Since write operations in FFP are performed much more often than in other types of flash memory, these operations are given great importance - page buffers are introduced, which allow you to quickly collect a certain amount of data for writing for their further transfer to the drive at a lower speed.

Intel FFP chipsets have an information capacity of 4 ... 32 Mbit with an access time of 70 ... 150 none, supply voltage 5; 3.3 or even 2.7 V. They have a byte or controlled bitrate (8 or 16), their programming voltage is also usually multivariable (3.3; 5; 12 V).

The external organization of the FSD is shown in fig. 1.21, c, on the example of a microcircuit with an information capacity of 16 Mbit (IC type 28P0168A from Intel) The circuit battery with a total information capacity of 16 Mbit is divided into 32 blocks of 64 Kbytes.

Let's explain the meaning of some pins and signals. Address bus: lines A_{20-6} choose one of the blocks, lines A_{15-1} select a word within one block (a 64 KB block contains 32 KB), a line A_0 is a byte select bit that defines the upper and lower byte in the byte memory organization and is turned off when these are dictionary organizations. The processor receives the starting address of the block of data that

is stored in the address queue. The actual address of the memory cell to be replaced consists of an address counter.

The data bus is provided DQ_{15-0} DQ_{7-0} lines to input and output the least significant byte of data, send commands to the CUI command user interface in a write cycle, and output data from the buffer, an identifier or status registers in their respective read modes. Lines DQ_{15-8} are intended to transfer the most significant byte during dictionary memory organization. They output the data of the drive, buffer, or identifier in the appropriate read mode, but these lines are not used to read from the status registers. If the crystal is not defined or the pin is disabled, the data bus lines enter the third state.

Lines CE0 and CE1 are crystal inputs, the crystal is undefined at high level of each, and power consumption drops to a standby level after the current write or erase operation is complete.

The OE signal opens the output buffers at a low level and drives them to the third high level.

The WE signal controls user interface access to CUI commands, page buffers, data queue registers, and address queue latches.

RP (Reset/Power-Down) puts the circuit into a deep power-saving state, turning off all circuits that consume static power. When leaving this state, the recovery time of the circuit is 400 ns. After switching to a low level, the operation of the automatic fixing equipment stops and the schedule is reset.

The RY / BY signal indicates the state of the internal DVR. A low level means busy, a high level (by the way, the signal is produced by an open stack stage, which requires the connection of an external circuit Ucc - R to create a high level) means either ready for new operations, canceled or deep power saving state depending on what is being done.

WP (write protection) has the following meaning. Each block has a write ihibit bit (lock bit). Low WP provides protection, i.e. write or erase in a block can only be performed when Lock-bit = 0. At high WP level write and erase operations can be performed by blocks regardless of the lock state.

The BYTE signal at a low level switches to the byte mode of the circuit, at a high level - to the dictionary mode and turns off the line buffer A_0.

The programming voltage Upp and the output voltage of the supply (can be 3.3 or 5 V - the input is marked with a fraction of 3/5) enter the system through a voltage switch located inside the system.

For example, let`s take **the parameters of the company FFP Intel** / 28F032SA:

organization 2Mx16 or 4Mx8 (recipient's choice), supply voltage 3.3 or 5V (recipient's choice), programming voltage 12V, for loop removal per block 10 , 64 independently closed blocks of 64 KB each or 64 blocks of 32 KB;

case type TSOP 1.2x14x20 mm with 56 outputs;

technology with a topological standard of 0.6 μm;

access time when reading 70 or 150 ns when powered from 5V and 3V, respectively;

word/byte recording time does not exceed 9 ms;

block recording time no more than 2.1 s for byte mode and no more than 1 s for dictionary mode;

the erasure time of the block is no more than 10 s, and the erasure time of the crystal is no more than 25.6 s.

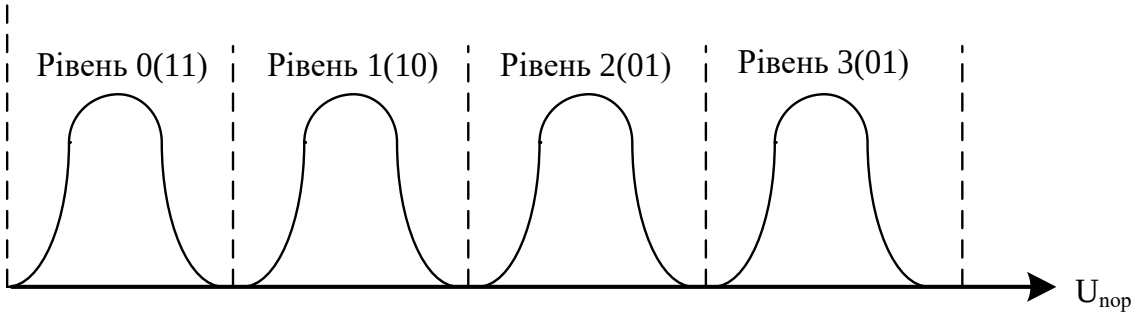
1.4.4. Memory type StrataFlash

In 1997 Intel introduced a new type of Flash called StrataFlash, which for the first time stored two bits in a single memory element instead of one. This is ensured by the fact that the presence or absence of charge is not only registered in the floating gate of the transistor, but also its value is determined, which can have several values. By distinguishing four levels, two bits can be stored in one element.

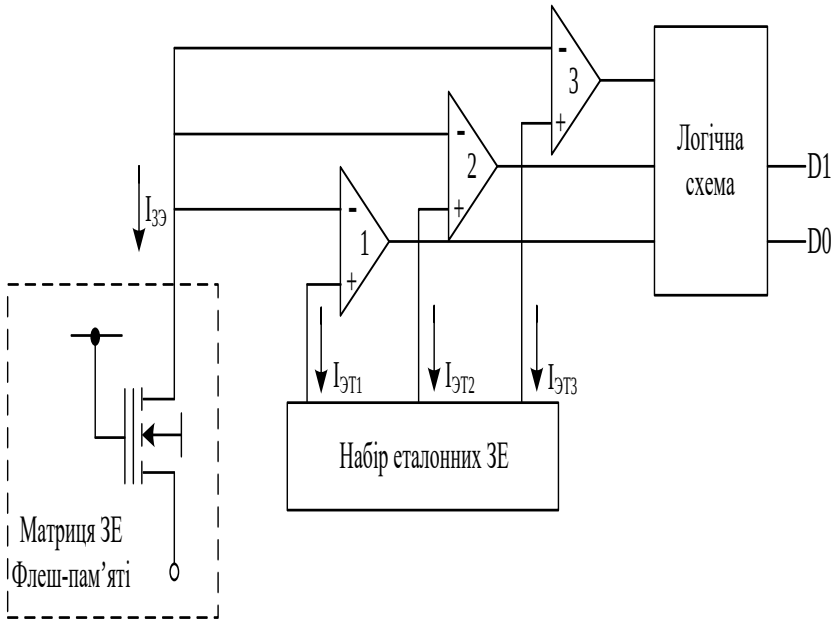
Prior to StrataFlash memory, increasing memory capacity was achieved by reducing the size of circuit elements and other improvements in lithography processes. StrataFlash has offered a different approach to this problem. The storage of two bits was achieved in practically the same memory elements that previously

stored one bit, overcoming the difficulty of increasing the tolerance of charge values entering the moving gate. In the second half of the 90s commercial samples of StrataFlash memory appeared. At the same time, the capacity increased from 32 Mbit to 64 Mbit without noticeable changes in the area of the crystal. Memory elements are programmed by entering one of 4 charge values into the floating gate, each of which corresponds to a pair of binary digits 11, 10, 01, 00. Depending on the load the memory transistor has one of four threshold voltages. When reading information, the read voltage is applied to the gate of the transistor. The current of the memory transistor depends on the threshold voltage. The state of the floating gate can be determined by detecting the current.

Figure 1.22 shows the distribution of the threshold voltages in the four-level reading element (a) and the reading circuit of the state of the reading transistor (b).



a)



Заряд в ПЗ	Виходи підсилювачів читання				Вихідні дані	
	1	2	3			
Q11	1	1	1	1	1	1
Q10	0	1	1	1	1	0
Q01	0	0	1	0	0	1
Q00	0	0	0	0	0	0

b)

Fig. 1.22. Graphs of the distribution of threshold voltages in a four-level memory element (a) and a diagram of reading data from this element (b)

1.5. THE USE OF PROGRAMMED SDs FOR SOLVING THE PROBLEMS OF INFORMATION PROCESSING

In the previous paragraphs memory devices were considered from the point of view of their main task - storing information. However, programming memory is also a universal means of solving a wide variety of information processing tasks. The possibility of using this tool in a specific field is determined by the possibility of presenting the solution to the problem in tabular form. This form of solution is possible in the case of problems of a very different nature.

In order to explain the capabilities of PPSDs in the area of solving information processing problems, it is worth considering the main dependencies associated with the reconstruction of logical and numerical functions.

1.5.1. Implementation of logical (switching) functions

PPSD with the organization $2^m \times 1$ takes an m-bit address and outputs a one-bit result (0 or 1). This mode of operation directly reproduces the switching function of m-variables, since for each input set during the programming of the SD,

the required output variable can be assigned. For example, a PPSD with a 1024x1 organization can be used to render a switch function with 10 arguments.

PPSD with the organization $2^m \times n$ input to an input with an m-bit address forms an n-bit word at the output, which is stored in a cell with a given address. Such a SDP restores the system of switching functions, the number of which is equal to the bit size of the original word. Indeed, any m-argument switching function can be reproduced on each output and set of outputs of n different functions.

In PPSD the functions are implemented in perfect disjunctive normal form, each possible conjunction has its own hardware (the output line of the address decoder) and therefore can be written into the output function. Any minimization of functions when preparing the problem for solution based on the PPSD is not necessary, especially if the functions are already minimized, then for the convenience of data preparation for the programming of the SDPs they will have to be expanded to the most burdensome form (SDNF).This is done either by completing the Carnot map and then writing the function without any sum of units, or by inserting the missing variables into each conjunction x_i by multiplying the conjunction by the unit expression $x_i \backslash / \overline{x_i}$ and opening parentheses (x_i - entered variable).

The implementation of functions in SDNF determines the high cost of memory elements, however, the price of a memory element is much lower than the price of a logical element, therefore, even with a multiple reservation of the number of memory elements (compared to the number of logical elements needed to restore functions by the traditional method), implementation on PPSD may be beneficial.

The specificity of PPSD indicates the expediency of its use to perform, first of all, functions that are not subject to significant minimization.

At the same time, the working time is the time of reading data from the SD.

1.5.2. Realization of finite automata

In the canonical circuit of the PPSD machine, it can replace the combined circuit because it is able to reproduce switching functions. Therefore, the structure of the automaton without loss of generality can also be presented in the form shown in Fig. 1.23.

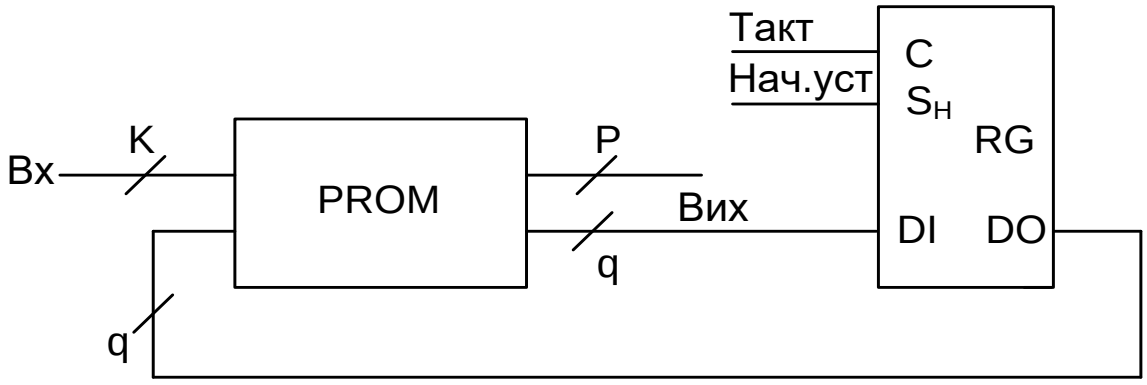


Fig. 1.23. The structure of the machine is implemented on the basis of memory chips

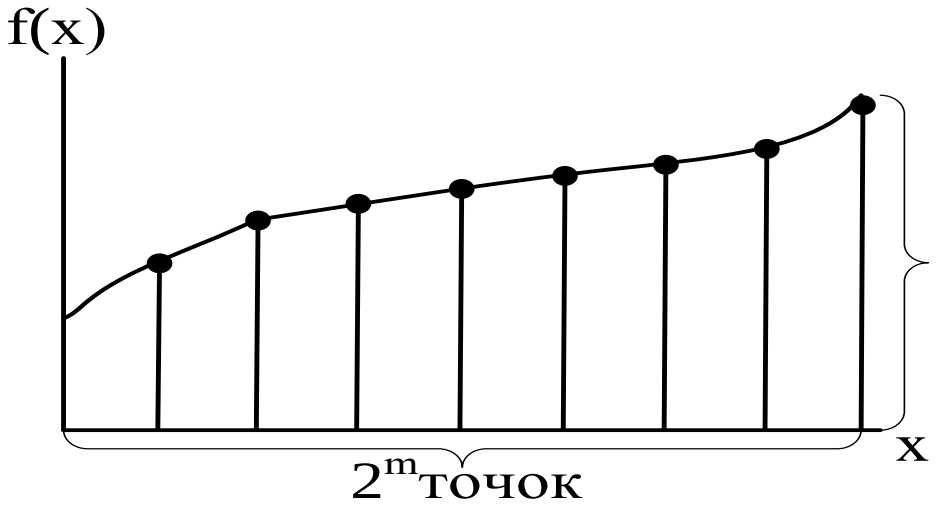
The previous setting register determines the initial state of memory elements (machines). The code for the new state and output function is read from the memory). based on this state and input signals. In the next beat these processes are repeated. In each cycle the automaton enters a new state and outputs output functions according to transition tables and output data.

The effectiveness of PPSD is determined by the volume of tables describing the operation of the machine. After reducing the tables of transitions and outputs to one, we get the total number of inputs m = k + q and the number of outputs n = p + q, therefore, the amount of memory required to implement the automaton $M = 2^{k+q}(p+q)$.

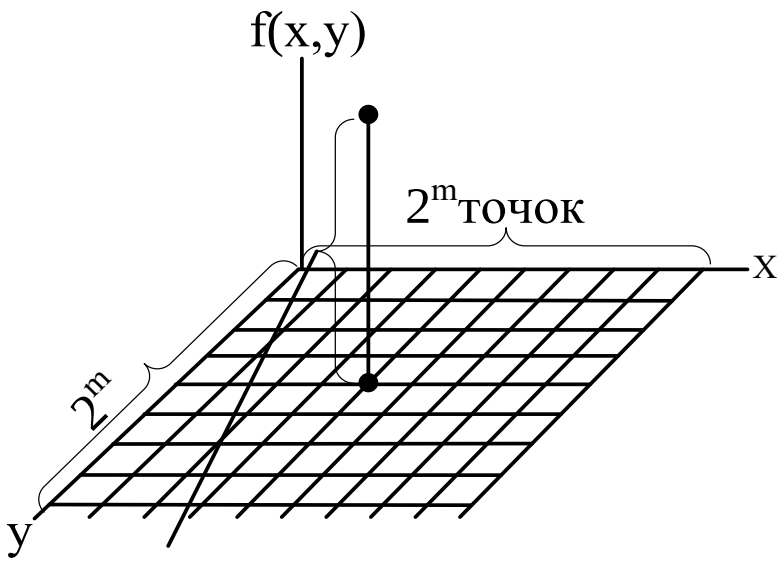
1.5.3. Reproduction of arithmetic and functional dependencies

Arithmetic operations and numerical (illogical) functions are often encountered as tasks solved by digital devices. Functions are defined analytically or tabularly.

For single-argument functions the amount of memory in the tables is easy to calculate by knowing the function's arguments and bits. When assigning an argument with an m-bit code, the number of points at which the function is specified will be 2^m (Fig. 1.24, a) . If the speed of the code representing the function is n, then of course the memory capacity in bits will be n 2^m.



a)



b)

Fig. 1.24. Determine the amount of memory when restoring numerical functions of one (a) and two (b) arguments by the tabular method

As the number of arguments increases, the amount of memory for storing function tables increases dramatically. For a function with two powers of arguments n, the number of points at which the function is given will be determined as the product of the number of points on each of the coordinates and 2^{2m} (Fig. 1.24, b). The amount of table memory in this case will be $M = n\,2^{2m}$.

For functions ℓ of arguments $M = n\ 2^{\ell m}$.

Therefore, with an increase in the number of words and the number of function arguments the amount of memory in arrays increases rapidly, and a purely tabular way of solving the problem becomes unacceptable. In such cases, tabular methods are often very useful, in which the size of the tables can be significantly reduced by introducing a small number of simple data operations.

In the case of arbitrary functions $f(x)$ the simplest table-algorithmic method is the segment-linear approximation, in which only the nodal values of the function are stored, and the function is calculated in the intervals between the nodes, assuming that it varies linearly in the intervals. The number of nodes is assigned based on the accuracy of the linear approximation of the functions at the positions. The following representations of the argument and function correspond to the piecewise linear approximation with a constant step:

$$x = x_1 + \Delta x,\ f(x) = f(x_1) + \Delta f(x_1)\Delta x h^{-1}$$

where x_i is the coordinate of the i-th nodal point; Δx - the difference between the values of x and the coordinates of the nodal point nearest to the left; $\Delta f(x_i)$,) is an increase in the on function section from x_i before x_{i+1} h is the approximation step (for ease of implementation by digital methods, the step is taken as an integer 2).

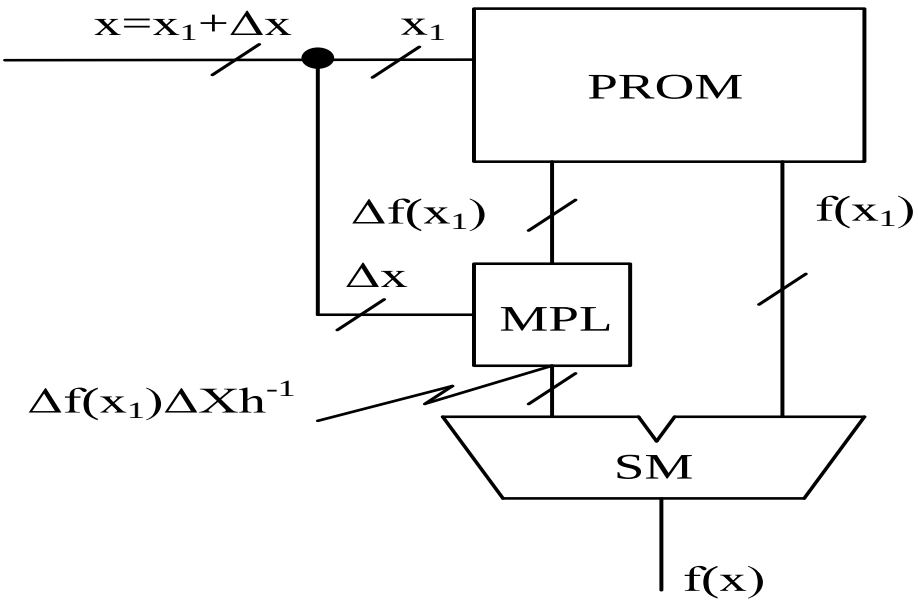


Fig. 1.25. The structure of the converter with the approximation of a segmental linear function

According to the formulas given above, the structure of the functional converter with segmental-linear approximation has the form shown in Fig. 1.25

As a rule, the amount of memory when switching from a tabular to a tabular-algorithmic method is significantly reduced and the speed remains quite high.

Planar-segment approximators can be used for functions of two variables.

SECTION 2.

PROGRAMMABLE LOGIC MATRIX, PROGRAMMABLE MATRIX LOGIC AND BASIC MATRIX CRYSTALS

2.1 PROGRAMMABLE LOGIC MATRIX

The concept of "PLM" is by no means new: it was introduced in the 1970s by the Monolithik company. Since then, this term has been used by many manufacturers, who sometimes changed and very significantly their products. The internal architecture of PLM circuits follows from the well-known proposition that any function of combinatorial logic can be reduced to a sum of products ("sum" is a "logical OR", "product" is a "logical AND"). Classical tools, such as de Morgan's theorem or Carnot tables (maps) allow you to obtain minimized Boolean equations from complete equations or truth tables obtained directly from the study of an implemented logic system.

The PLM scheme consists of the following logical blocks:

- Several input terminals, each of them has a buffer element and that element has both direct and inverse outputs;

- Multi-input programmable “AND” logic array and any input signal or its addition via a programmable switch field can be applied to each input of each “AND” logic through a repeater or inverter;

- A programmable array of "OR" logic elements with a specified number of inputs, each of them is connected to the corresponding number of "AND" element outputs;

- Buffer circuits located between the outputs of the “OR” gates and the external outputs of the PLM: inverters, buffers of the third state or D – triggers can be used here. In the latter case PLM is used to implement not only combinatorial circuits, but also relatively complex sequential circuits, such as counters or registers;

- Several feedback loops that return the output signals to the programmable switch field so that they can be processed as input signals;

Input buffers, unless they perform more complex operations, convert single-phase input signals into two-phase and create signals with the power required to power the "AND" element matrix.

The output buffers provide the necessary load capacity for the outputs, enable or disable the PLM output to the external buses using the OE signal and sometimes perform more complex operations.

Variables $x_1 \ldots x_m$ are fed through the input buffer to the inputs "AND" of the elements (branches), and *1 terms* are formed in the "AND" matrix. This term is understood here as a conjunction connecting input variables presented in direct or inverse form. The number of formed words is equal to the number of conjunctions or the same number of exits of the"AND" matrix.

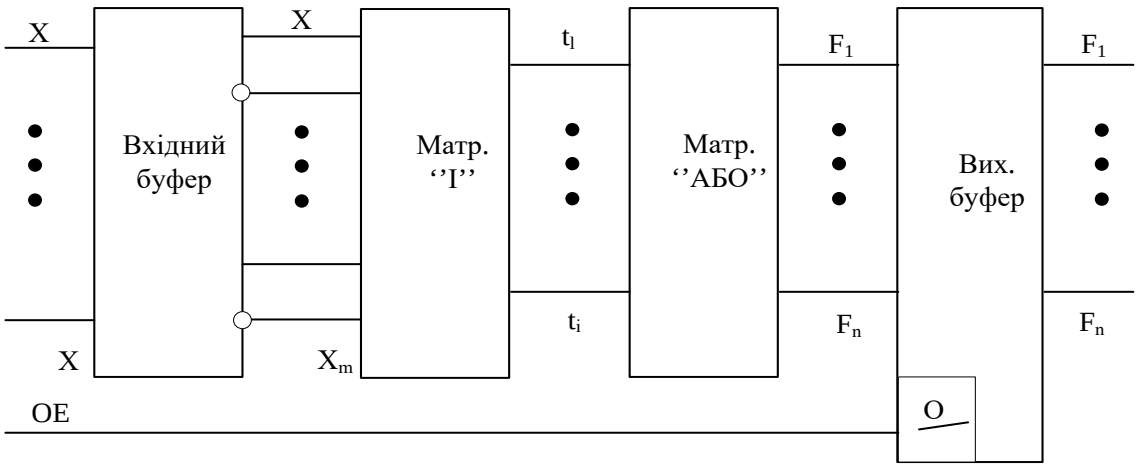


Fig. 2.1. The basic structure of PLM

The terms are then fed into the "OR" matrix, that is, to the inputs of the dissociators which form the output functions. The number of disjunctors is equal to the number of created *n functions.*

Thus, PLM implements the disjoint normal form (DNF) of reconstructed functions (two-level logic). PLM is capable of implementing *an n* system of logical

functions from *m* arguments containing no more than *l* thermal pools. The reproduced functions are combinations of any number of terms formed by the"AND" atrix. Exactly which terms will be generated and which combinations of these terms will generate the output functions is determined by PLM programming.

2.1.1 Programmable logic matrices circuits

PLMs are produced both on the basis of bipolar technology and on MDN - transistors. Matrices have horizontal and vertical link patterns at the intersection of which nodes are created or deleted during programming.

Figure 2.2. shows in a simplified form (without buffer elements) the generalized structure of PLM with programmable jumpers burning. A fragment for reproducing the system of functions with dimensions 4, 7, 3 is given.

$$F_1 = \overline{x_1 x_2} x_3 \vee x_2 \overline{x_3} \vee x_1 \overline{x_4} = t_1 \vee t_2 \vee t_3$$

$$F_2 = \overline{x_1 x_2} x_3 \vee x_1 \overline{x_2 x_3} \vee x_1 x_2 x_4 \vee x_2 \overline{x_3} x_4 = t_1 \vee t_4 \vee t_5 \vee t_6$$

$$F_3 = x_1 \overline{x_4} \vee x_1 \overline{x_2} = t_1 \vee t_7$$

Diodes connecting the horizontal and vertical rails serve as the connecting elements of the "AND" matrix, as shown in fig. 2.2. What is a thermal production chain t_1 Before programming, all jumpers are intact and communication LEDs are placed at all grid nodes. The result will be zero for any combination of arguments because the values of the forward and reverse arguments are simultaneously applied to the circuit input and $\overline{X} \times X = 0$. When programming only necessary communication elements remain on the circuit, and unnecessary ones are removed by burning jumpers. In this case the introduction of the conjugate is postponed x_1, x_2 and x_3 - . A high output voltage level (logic 1) will only appear when all inputs are high, a low voltage on at least one input sets the output voltage low as the LED for that input opens. In this way the operation "AND" s performed and the term $x_1 \overline{x_2 x_3}$ is formed.

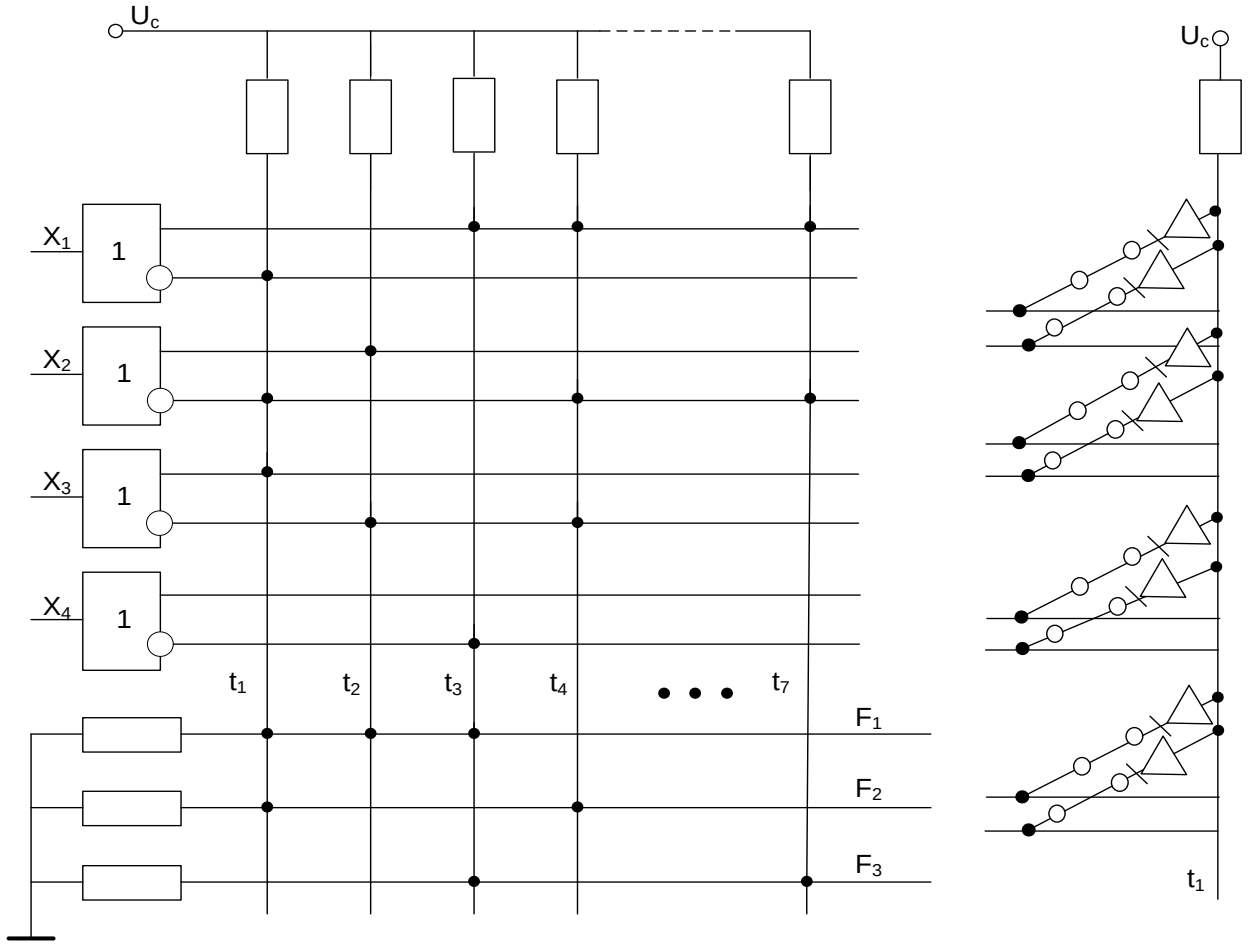


Fig. 2.2. PLM diagrams in bipolar technology

- Communication elements in the "OR" matrix are transistors, as *t1*are contained according to the scheme of the emitter follower relative to the thermal lines and form an "OR" circuit relative to the output (horizontal lines). When displaying programmed matrices, the presence of connecting elements (entire jumpers) is signaled by a dot in the corresponding node.

- "OR - NOT" and "AND - NOT" circuits are used as the main logic element in circuits on MDN - transistors. The operations performed in the first and second PLM matrices also change accordingly. In particular, in circuitry with MDN technology the "OR - NOT" structure is usually the main element. In this case the PLM structure looks like this (Fig. 2.3.). Such a PLM is a sequence of two "OR - NOT" matrices, one of which is used to generate terms, and the other to generate output functions.

- Term t_1 in this case is equal to:

$$t_1 = \overline{x_1 \vee x_2 \vee \overline{x_3}} = \overline{x_1 x_2 x_3}$$

and function

$$F_1 = \overline{t_1 \vee t_2 \vee t_3}$$

On the basis of these expressions it is possible to show the well-known relationship between operations, expressed by de Morgan's rules, which indicates the real convergence of the functional characteristics of bipolar PLM and PLM on MDN transistors: if the inputs of the latter are applied to the inputs of the latter, they are inverted with respect to the bipolar arguments of the PLM, the output will be a result that differs from the bipolar PLM output only by the inversion.

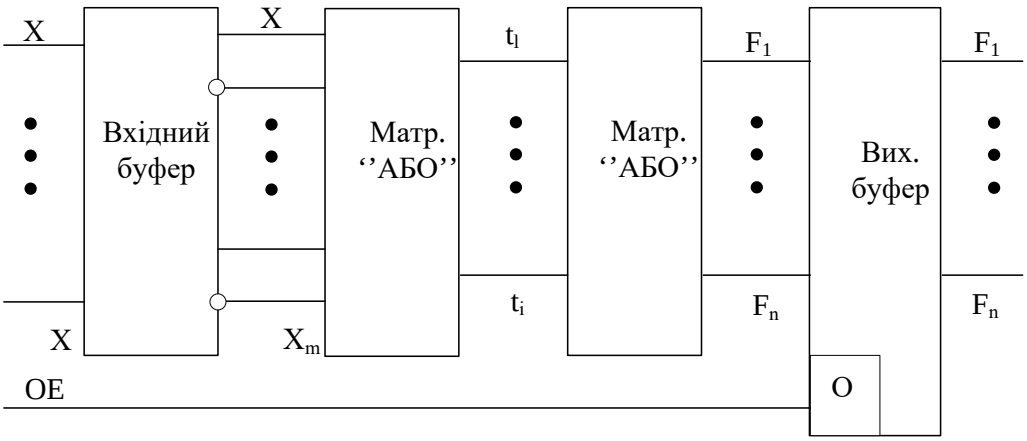


Fig. 2.3. PLM circuits implemented on MND transistors

PLM circuits are quite bulky, so they are presented in a simplified form F. Images are used in which the multi-input elements "AND" and "OR" are conventionally replaced by single-input elements (Fig. 2.4).

With the help of PLM, you can reproduce disjoint - normal forms of functions and even their forms in parentheses. In this case, the expressions in parentheses are obtained first, and then taken as arguments to the final result. Feedback from the output to the input appears in the circuit, the logic depth of the circuit and the delay of the output signal delay increase.

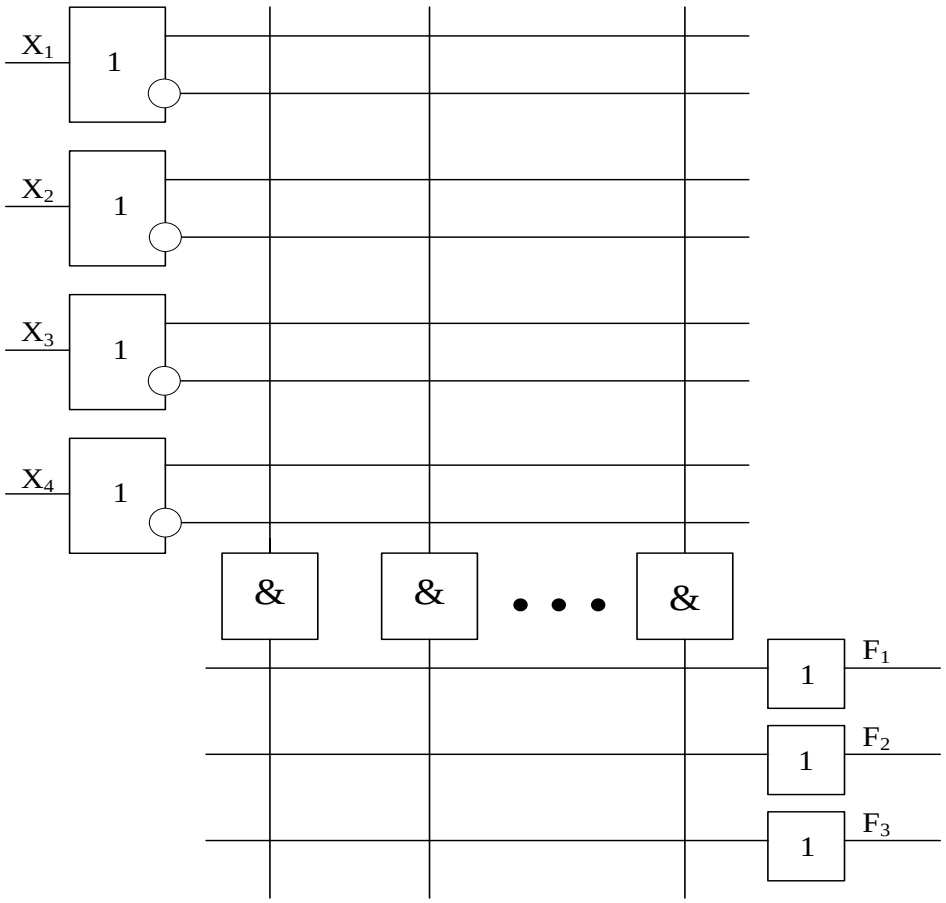


Fig. 2.4. A simplified circuit of a logical element.

For functions:

$$F = x_1 x_2 \vee (x_1 x_2 \vee \overline{x_2 x_1}) x_3$$

it is necessary to use the PLM included in the circuit shown in Fig. 2.5.

2.1.2. Matrix logic programming

One of the important applications of VIS with programmable logic is the replacement of VIS with a low and medium level of integration in the implementation of the so-called arbitrary logic. The power of PLM logic is often not used in these structures. This can be seen, in particular, when reproducing typical practical crosses with each other under the same conditions. In such cases, the possibility of using the outputs of any connectors by any disjunctors (as provided by PLM) becomes an unnecessary complication. Rejecting this capability means refusing OR matrix programming and results in a PML structure.

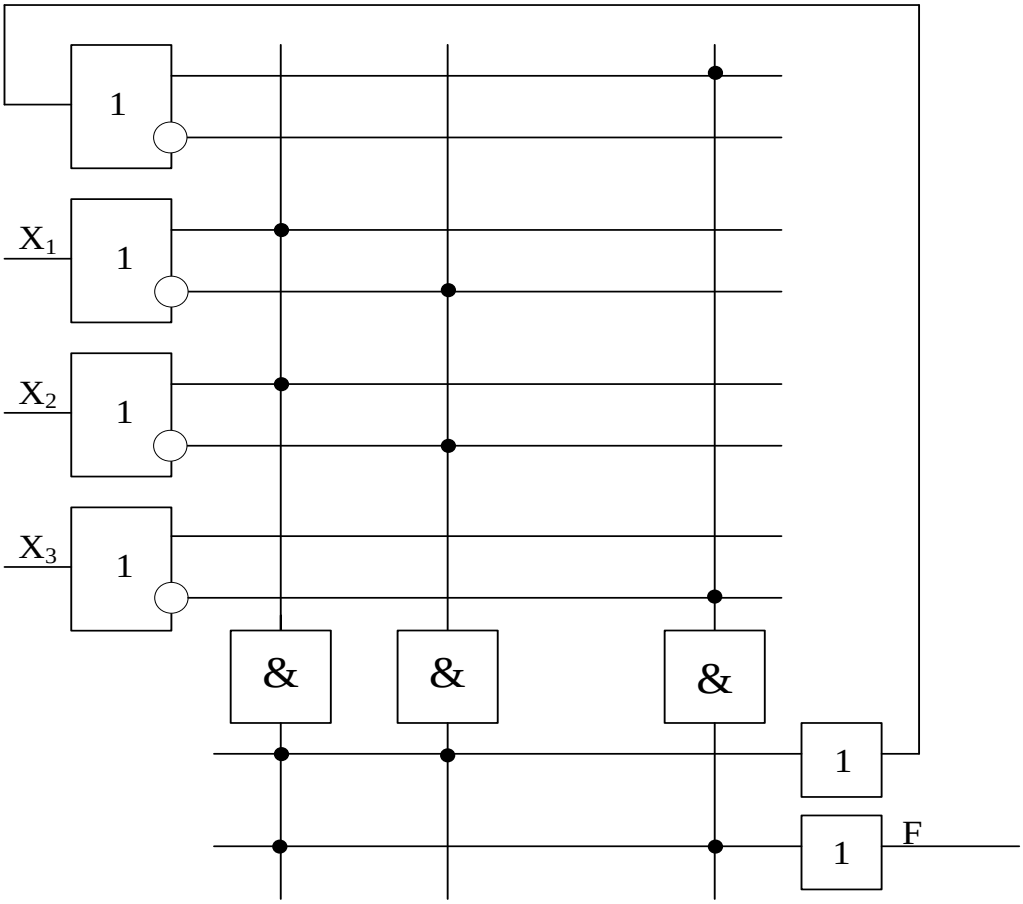


Fig. 2.5. PLM switching circuit when playing blocking forms

The outputs of the "AND" elements are rigidly distributed among the elements (inputs of the "OR" matrix) in PML (fig. 2.6.). In the shown PML, there are *m* inputs, *n* outputs and 4 "AND" elements, because for each "OR" element there are four connectors.

Compared to PLM, PML circuits have less flexibility in terms of functionality because they have a fixed “OR” matrix, but they are easier to manufacture and use. The advantages of PML are especially evident when designing simple devices.

Troubleshooting preparation for PML has a lot in common with the PLM troubleshooting approach, but there are also differences. It's important for PML to reduce the number of " AND" elements for each output, but if for PLM they try to find a representation of the function with the most common terms, then for PML it

is not necessary because the "AND" elements are fixed to their outputs and they cannot be used by other outputs.

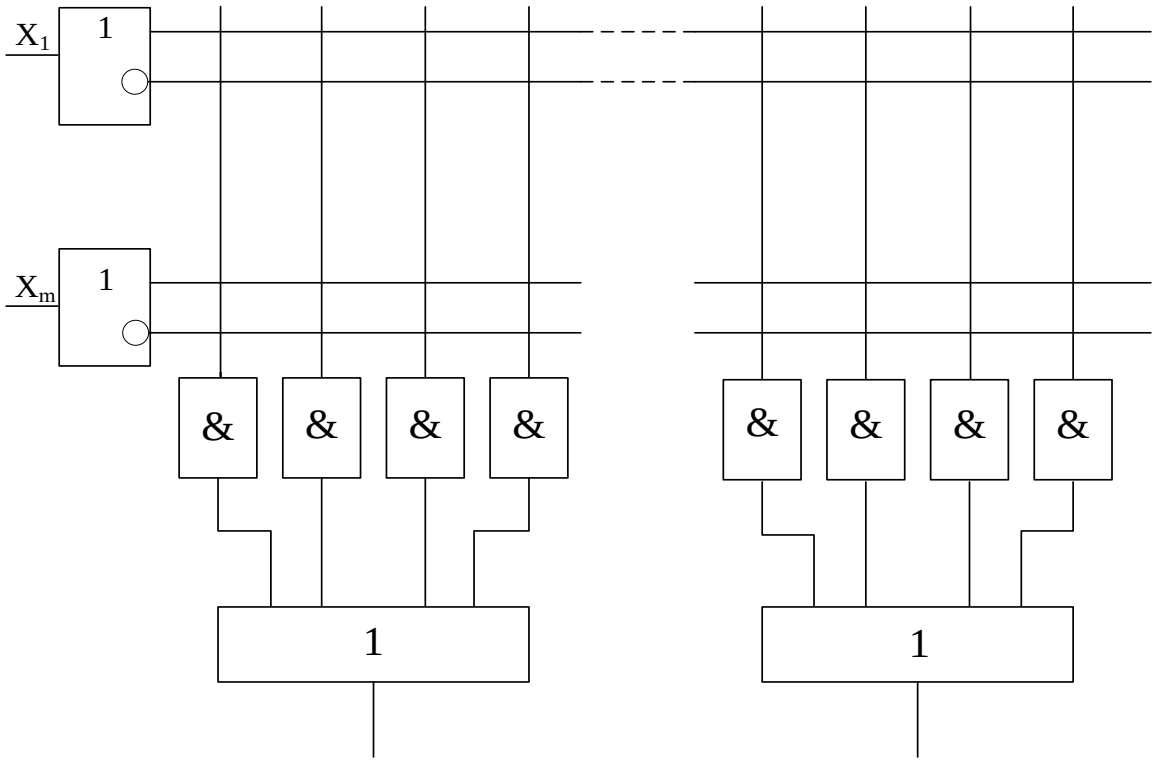


Fig. 2.6. The basic structure of PLM.

2.2 FUNCTIONAL VARIETIES OF PLM AND PML

The considered structures of PLM and PML are the basic ones from which the development of these directions began. In the future the functionality of PLM and PML was enriched by a number of methods, mainly the following.

2.2.1 Circuits with a programmable output buffer

The output functions can be obtained in direct or inverse form in these diagrams. In such a circuit (fig. 2.7.) the functions F1 ... Fn produced by the matrices pass through the output buffer, and the discharge circuits from the "ground" potential through the fusible link (PP). In this case the output functions of the matrix are passed through the buffer without changes. If the lower input jumper of the adder is broken, it will receive a logic unity signal from the power supply through the resistor R. Summing modulo 2 with one the Fj* functions will

be inverted. Therefore, the functions pass through the buffer without changes in lines with full jumpers and are inverted in lines with missing jumpers.

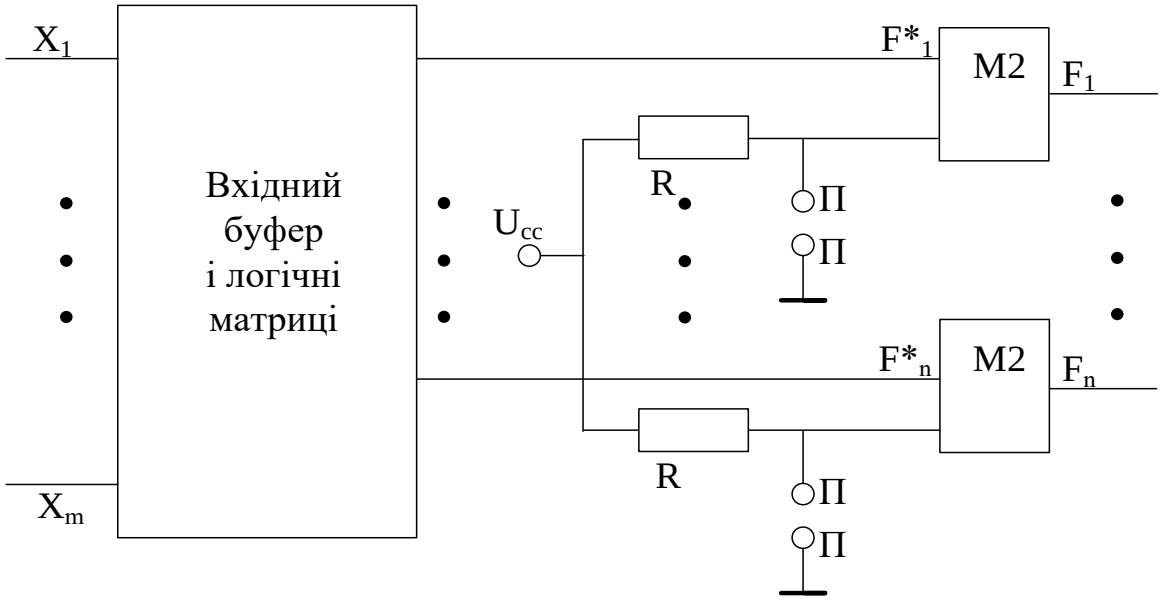


Fig. 2.7. The circuit with a programmable output buffer.

The programmable buffer provides additional opportunities for minimizing the number of deadlines in the implemented system. In the original system functions can be replaced by their reciprocals if this reduces the number of members. Reverting to the original system simply resets the buffer.

2.2.2. Systems with bidirectional outputs

It is possible to build a system in which some outputs can be adapted to be treated as inputs or outputs, depending on the programming of the jumpers using elements with three output states. In such a scheme one of the switches is designed to control an element with three output states (fig. 2.8). The output of the element is simultaneously connected to the "AND" matrix as an input.

Four I/O output modes are possible depending on how the switch inputs are programmed:

1. All jumpers are intact. In this mode the output of the switch is zero, the buffer has a third output state, and the output acts as an input.

2. All the jumpers burnt out, there is a short circuit at the output of the switch, the buffer is active, the output works as an output (its signals are not used in the "AND" matrix).

3. Leave feedback. This mode differs from the previous one only in that the output signals are used in the "AND" matrix.

4. Controlled output. Switch inputs are programmable.

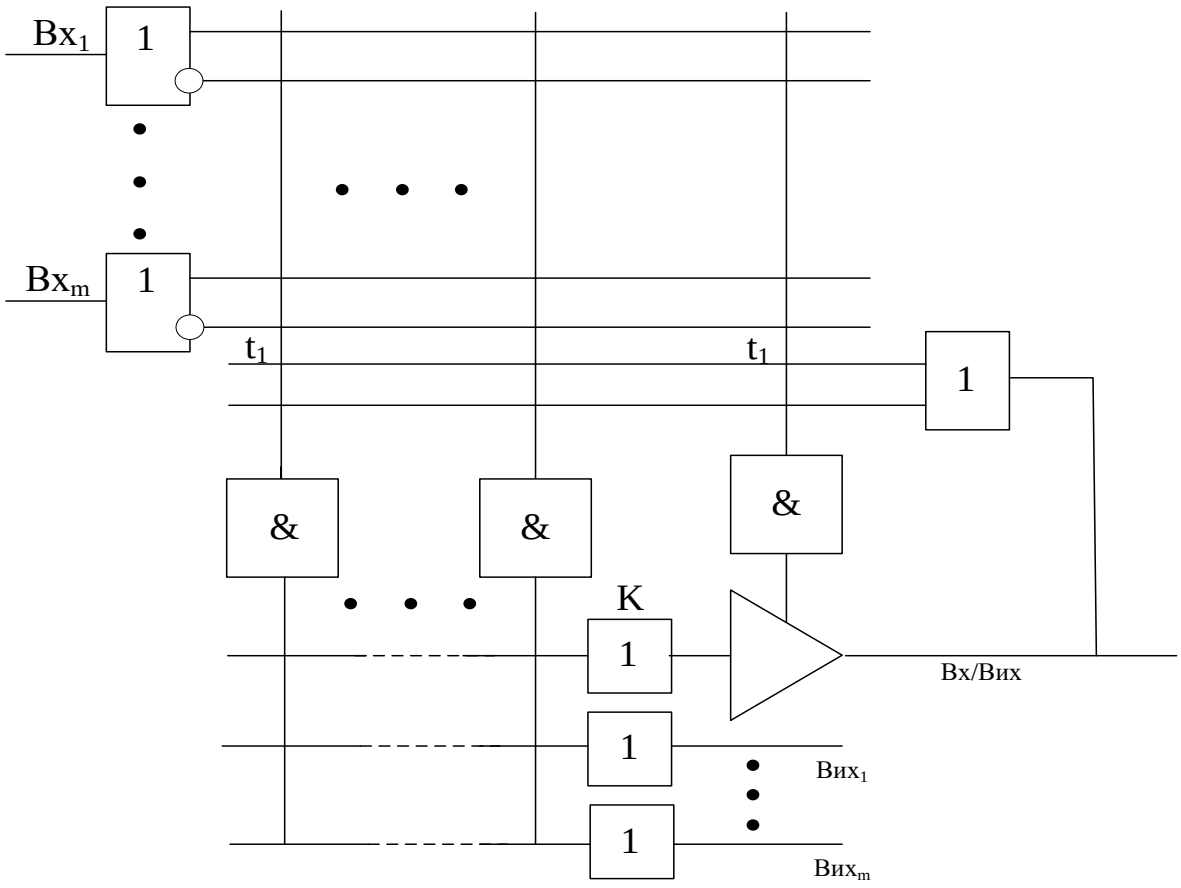


Fig. 2.8. Diagram with a bidirectional buffer

It is possible to change the ratio of the number of inputs – outputs in a diagram with a certain number of bidirectional outputs. If the number of inputs is equal to *m,* the number of outputs and the number of bilateral outputs is equal *to p,* then the number of inputs can be increased from *m* to *m*+ *p*, and the number of outputs from *n to n*+ *p.* provided that the sum of the number of inputs and outputs does not exceed *m*+ *n*+ *p.*

2.2.3. Circuits with memory

These circuits make it possible to build automata in the most convenient way, since, in addition to the combined part they contain D-type triggers (registers) on the crystal (fig. 2.9). PLM with memory has four parameters. In addition to the three usual parameters, it also has a parameter *r* - the number of memory elements (register bits).

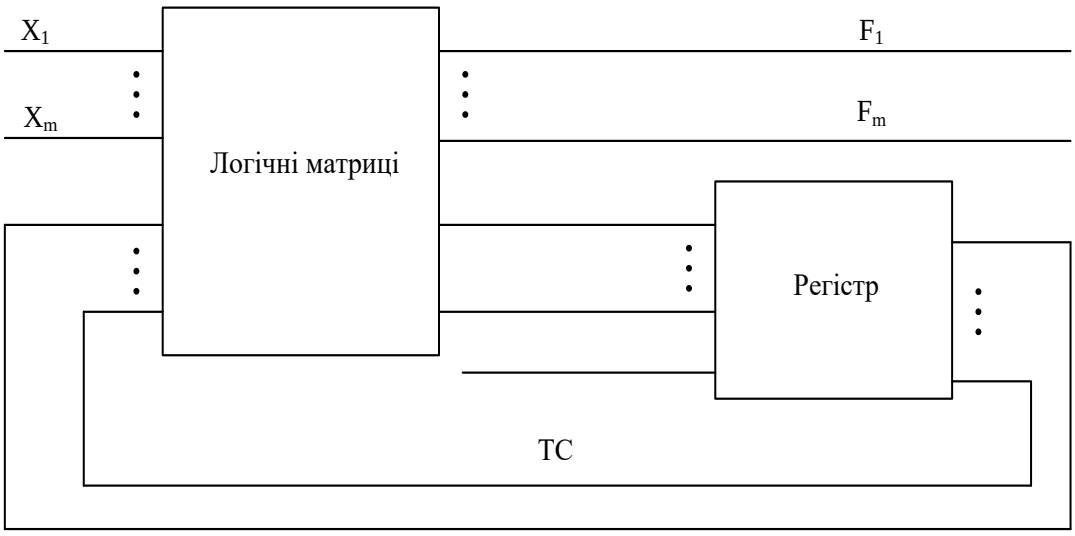


Fig. 2.9. PLM structure with memory

The structure (see fig. 2.9) corresponds to the canonical circuit of the automaton. The result of this stage of information processing depends on the results of the previous steps, which are provided by the register feedback to the PLM input. The maximum number of internal machine states is 2G. The automaton is considered synchronous - the feedback circuit is only enabled in conjunction with clock signals (TS).

2.2.4. PLM with detachable connectors

Along with the circuit modifications discussed above, there are also special modifications related only to PLM. They include the variant with the so-called divided conjunctions. The method of " conjunctions separation is as follows.

Two adjacent “OR” elements are assigned a certain number of connectors that can be freely shared between these adjacent elements. Other “OR” elements

cannot use this combination set. Full programming of the matrix ”OR” is missing here, but this modification is a step towards PLM.

The variant with distributed connectors removes the most obvious limitation of the functionality of simple (full) LMLs - a fixed number of "AND" elements at the inputs of "OR" elements, which may not be enough for reproduction of complex functions. By having a PMJI with distributed connectors and placing a complex function next to a line, you can borrow a part of the entire set of connectors from the simple function to the complex function.

An option for implementing the schematic divisibility of multipliers is shown in (fig. 2.10). The PML has an additional set of "OR" elements and a module 2 addition, with the help of which the signals from both outputs of the main "OR" circuits can be combined to form the residual values of the functions *F1* and *F2*. The main "OR" circuits can be combined according to the operating connectors or add-ons by module 2 and will be divided into the main outputs *F1* and *F2*.

The modulo 2 addition operation provides additional functionality. The nature of the obtained functions depends on which of the three transistors in the two shown groups will be conductive.

The first domestic PLM appeared in the KR1556 series (HL8 microcircuit, then 4 microcircuits, XP6, XP8). The XP8 chipset is a PLM with bidirectional outputs (inputs - outputs). The number of inputs can be from 10 to 16. The total number of inputs and outputs should not exceed 18.

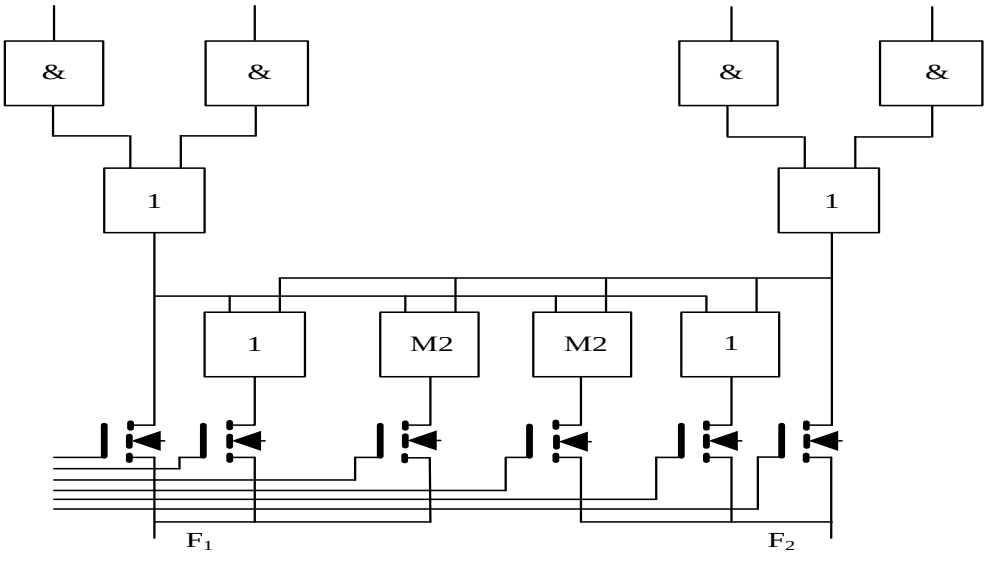


Fig. 2.10. Separation of terms in PLM

The structure and principle of operation of domestic PLM and PML will be considered in more detail in the second chapter.

2.2.5. Extension (expansion) of PLM

If the dimension of the problem exceeds the capabilities of the existing PLM, it should be increased. When the number of functions in the system *N* exceeds the number of PLM outputs, several PLMs are turned on in parallel at the inputs (fig. 2.11).

Some functions are duplicated at the outputs of each PLM. The total number of PLMs is defined as [*N / n*]. Since the number of conditions is considered sufficient (Lsyst < L), all PLMs can be programmed for the same conditions.

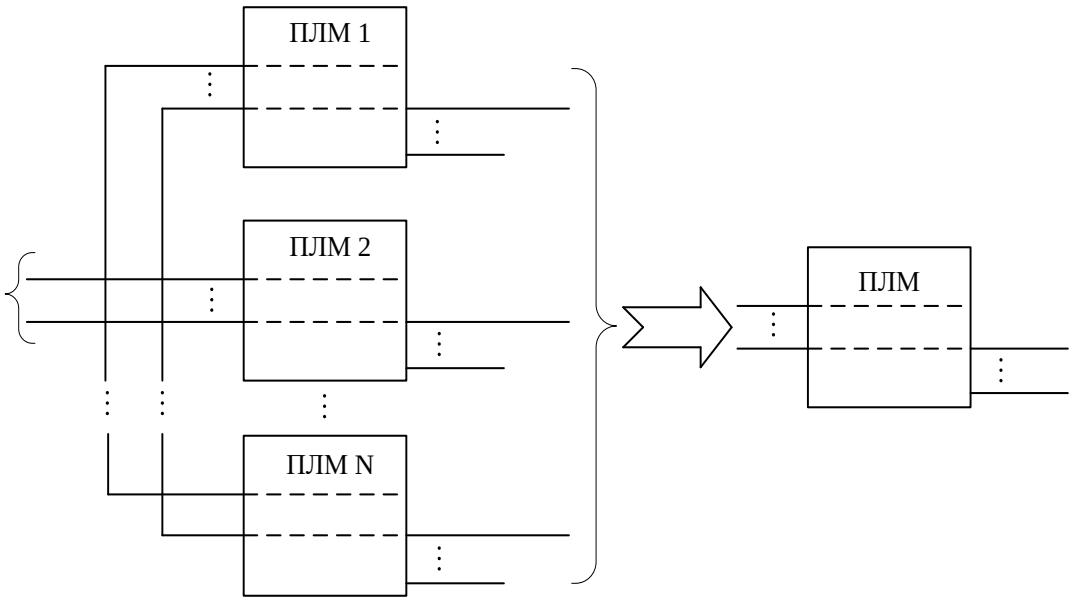


Fig. 2.11. PLM extension circuit with a number of outputs

If the number of Lsyst system elements exceeds the number of PLM blocks (Lsyst> L), then additional blocks with the same number of inputs and outputs are connected to one PLM. The PLM inputs are switched in parallel and the corresponding outputs are combined or simply combined if they are third-state or adder logic outputs. Each PLM is programmed for its conditions, and then the necessary functions are "collected" from the terms at the outputs (Fig. 2. 12).

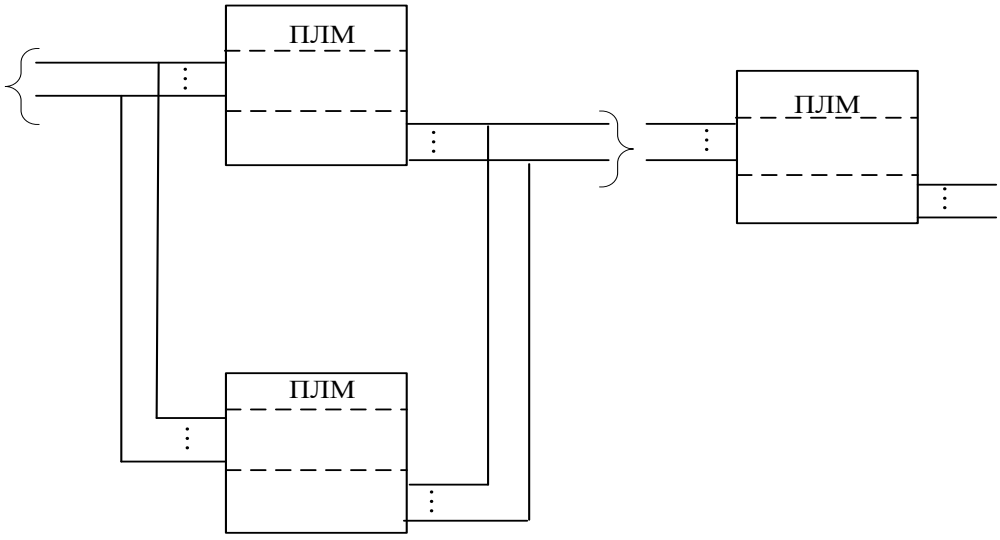


Fig. 2.12. PLM extension circuit by number of members

Expanding the number of inputs is the most difficult task associated with the decomposition of a system of functions. As a special case, if all terms contain no more than *m* variables, the set of terms can be divided into subsets of no more than *m* identical variables. The implementation will require a number of PLMs equal to the number of subsets, and the PLM inputs will be connected in the same way as for term expansion. The input variables of each PLM will be associated only with forming a member of that subset.

A standard technique for augmenting a PLM with inputs is to pass excess arguments to a pre-decoder which outputs allow one of the PLMs to process the remaining arguments. The diagram of the expansion of the number of PLM inputs per block is shown in fiureg 2.13. In times of large expansion of the number of inputs, this method is not very suitable because redundant variables dramatically increase the number of PLMs in the circuit (doubling by adding each redundant input).

The first samples of base matrix crystals (BMC) appeared in 1975 as a means of implementing non-standard high-performance computer systems without the use of microcircuits of low and medium integration. The development of BMC additionally made it possible to manufacture non-standard parts on VIS.

The cost of designing VIS / DVIS is high and amounts to tens or hundreds of millions of dollars. It is clear that the production of VIS/DVIS becomes profitable only when it is sufficiently spent, which cannot be said about the development of non-standard parts of specific systems.

A way out of the current situation was found in the development of VIS/DVIS, the functioning of which can be adapted to solve certain tasks at the final stages of their production. At the same time, semi-finished products are produced in large quantities without focusing on a specific customer. Customizing blanks only at the final stages of VIS/DVIS production is much cheaper and requires much less design time. Such VIS / DVIS are called partially non-standard, not fully non-standard.

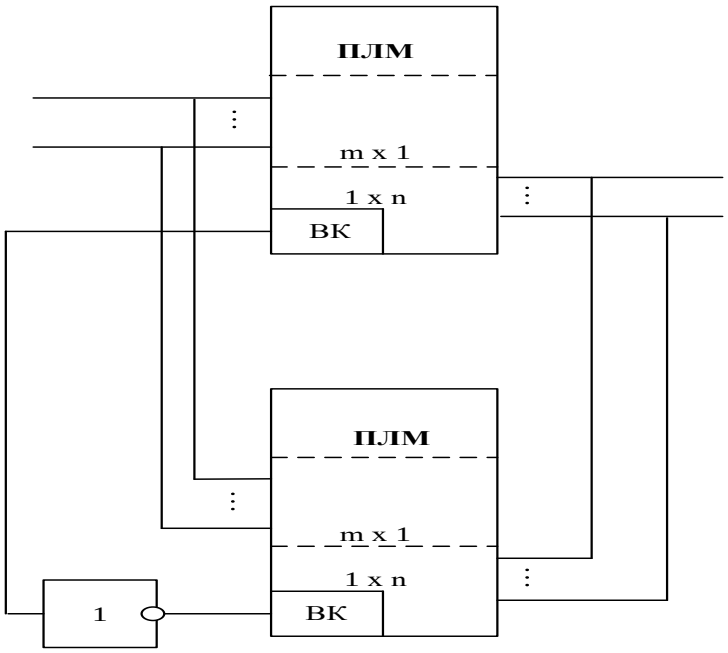


Fig. 2.13. Circuit for expanding the number of PLM inputs per block.

Basic matrix crystals (valve matrices with mask programming)

The development of semi-custom VIS / HVIS has led to the appearance of a number of variations. In the case of BMC, these are channel, channelless and block architectures.

The basis of the BMC of the first generation is a set of base cells (BC) evenly distributed over the crystal, between which there are free zones for making

connections. This architecture is called a channel one. Primary cells occupy the inner region of the BMC, where they are arranged lines and columns containing groups of non-switching elements. In the peripheral circuit of the crystal there are input-output cells, that is, a set of circuit elements which is aimed at implementing BMC connections with external circuits.

Thus, the BMC is a workpiece that is transformed into the necessary circuit by making the necessary connections. The user can implement devices of a certain class on the basis of BMC by specifying one or another option for connecting elements to the crystal.

Optimizing the result was not the price for reducing design time and costs. BMCs lose to completely customized circuits in terms of crystal surface and speed, because some of their elements become redundant (they are not used in this circuit), the relative location of elements and connection paths are not the best. Since the beginning of the 1980s the production of BMC has undergone extensive development. The technologies KMDI-I, TTLSH, ESL, etc are used. At the moment the level of integration of BMC has reached millions of heads on the crystal.

It is necessary to balance the number of base cells, crystal trace resources and the number of contact fields for connecting external terminals as best as possible when designing a BMC. Violation of connections between certain parameters can significantly reduce the full use of crystal resources.

Traceability of BMC is primarily determined by the area intended for inter-element connections in orthogonal directions. Insufficient traceability leads to a reduction in the number of primary cells used. Excessive trace capacity leads to unreasonable use of the crystal surface, which reduces the level of integration of the BMC. The same can be said about the number of external terminals of the BMC, for a modern BMC up to 500 ... 600 external terminals are required.

The designer (user) must have an idea of the existing BMCs, their varieties and design features.

A basic cell is defined as a set of circuit elements that are regularly repeated on a certain area of the crystal. This set can consist of both non-switched and partially switched elements. The basic cells of the inner part of the BMC are called basic matrix cells (BMC), the cells of the peripheral zone are called peripheral basal flanks (PBF). There are two ways of organizing BMC cells:

one logical element can be created from MBC elements, and several cells are used to perform more complex functions;

any functional node can be created from MBC elements, and the composition of cell elements is determined by the circuit of the most complex node.

A functional cell (FC) is a functionally complete diagram implemented by combining elements in one or more BCs.

A functional cell library is a set of FCs used in a design. This library is created during the development of BMC and frees the designer from the work of creating some typical patterns on the crystal, as it presents ready-made solutions for their implementation. The library contains a large number of functional elements, nodes and their parts. Using the library, the designer implements circuits which operation has already been tested and their parameters are known. An equivalent gate (EB) is a group of BMC elements that corresponds to the possibility of realizing the logic function of the gate (a bidirectional AND-NOT or OR-NOT element). The term "equivalent valve" is intended to evaluate the logic complexity of the BMC.

Trace channels are paths in the BMC for possible placement of connections.

2.3. CLASSIFICATION OF BMC

The BMC classification is shown in figure 2.14. The basic and classic structure of the BMC channel. In the inner (central) zone of such a CMC, a matrix of basic cells and tracking channels is placed.

Channels can be vertical and horizontal or only vertical. Channel BMCs can have a greater ability to form connections, but have a low density of arrangement

due to the significant cost of the crystalline surface in the region between the connections.

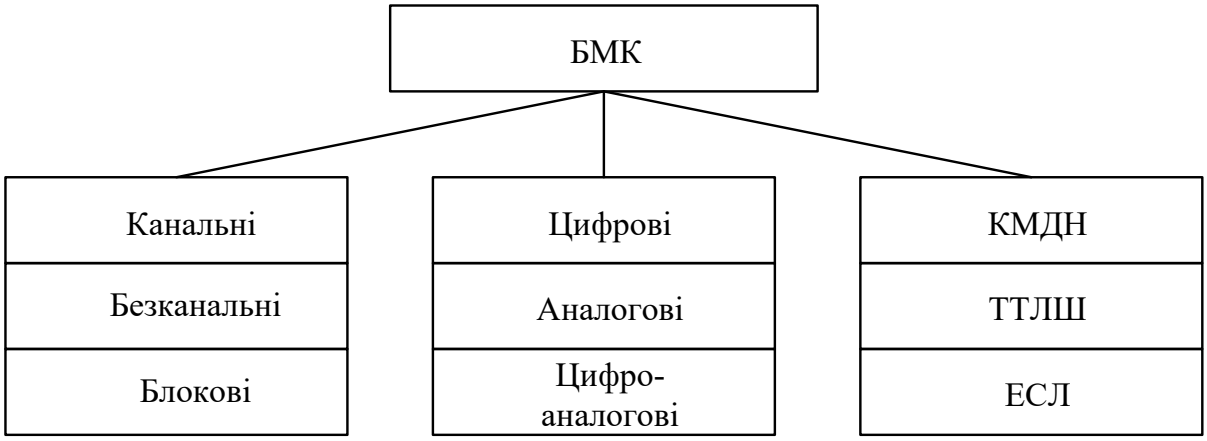


Fig. 2.14. Classification of the main matrix crystals

An increase in the level of BMC integration leads to a rapid increase in the number of necessary connections between basal cells, and therefore the area for them. The search for ways to create a BMC with a high level of integration while minimizing the area of connectivity led to the development of a channelless BMC architecture. The inner region of such a BMC contains densely packed lines of basic cells, and continuous channels for tracking connections are absent. In such a crystal, each BC placement area can be used both to create logic and to create interconnections. As a result of a more rational arrangement of connections in a non-channel BMC, the signal transmission delay is also reduced, since the lengths of the connections and parasitic capacitances are reduced.

KMDN technology is characterized by channelless BMC - circuits in which the compactness of the circuit elements and the low dispersion power of the BC when working at high frequencies allow for dense packing of the main cells.

Since in non-channel BMC the position of trace channels and cells on the working field is not constant, and the crystal area can be redistributed between trace channels and functional cells during design, the loss of crystal area is reduced. For

example, in a BMC with a dense arrangement of lines of transistors on the working field, logic elements are implemented in some lines, and other lines are used to track channels, the transistors in them remain unconnected and unused (traces pass through them). Depending on the load on the channels, they can be assigned a different number of lines of transistors.

The variable cell length architecture is also used in KMDN BMC. At the same time each row is a serial connection of pairs of *n- and p*-channel transistors. If pairs of closed transistors are placed in such a long chain in certain places, then the circle will be divided into basic cells of a given length.

The internal region (IR) of the crystal is limited by the peripheral region (PR located along the edge of the rectangular plate of the BMC. In the peripheral part there are special printed circuit boards, a set of circuit components for signal input/output troubleshooting, and contact pads (PCs). Increasing the level of integration leads to the possibility of implementing increasingly complex devices and systems on a single crystal. This led to the creation of BMC block structures, the architecture of which simplifies the construction of combined devices containing both logical data processing blocks and memory blocks or other specialized blocks. At the same time several sub-matrix blocks are implemented in the BMC, each of which has a smaller BMC structure. Channels-routes are laid between the blocks. On the edge of the blocks there are cascades of internal buffers to generate sufficiently strong signals.

The type of processed signals (digital, analog) affects the quality and composition of the circuit elements of the primary cells. According to this BMC are divided into digital, analog and digital-analog. Analog and digital-analog BMC appeared after digital and are less common, they have a composition of basic cells, which allows you to obtain circuits of operational amplifiers, analog keys, comparators, etc.

The classification by manufacturing technology reproduces only the main variants of BMC. The maximum speed options are implemented on ESL-type circuits or on gallium arsenide. A significant place is occupied by the circuit

engineering of KMDN. A medium-speed BMKCwas created on the basis of the TTLSH technology. An important feature of BMC is the number of connective layers (currently 2 ... 6). Multi-tiering simplifies tracking and allows you to create BMCs with a higher level of integration. With an increase in the number of layers, trace channels between cells can be eliminated, moving to a channelless structure.

One of the variants of the BC BMC of the KMDN type is presented in figure 2.15. Only transistors with *n and p*-channels serve as circuit elements of such BCs.

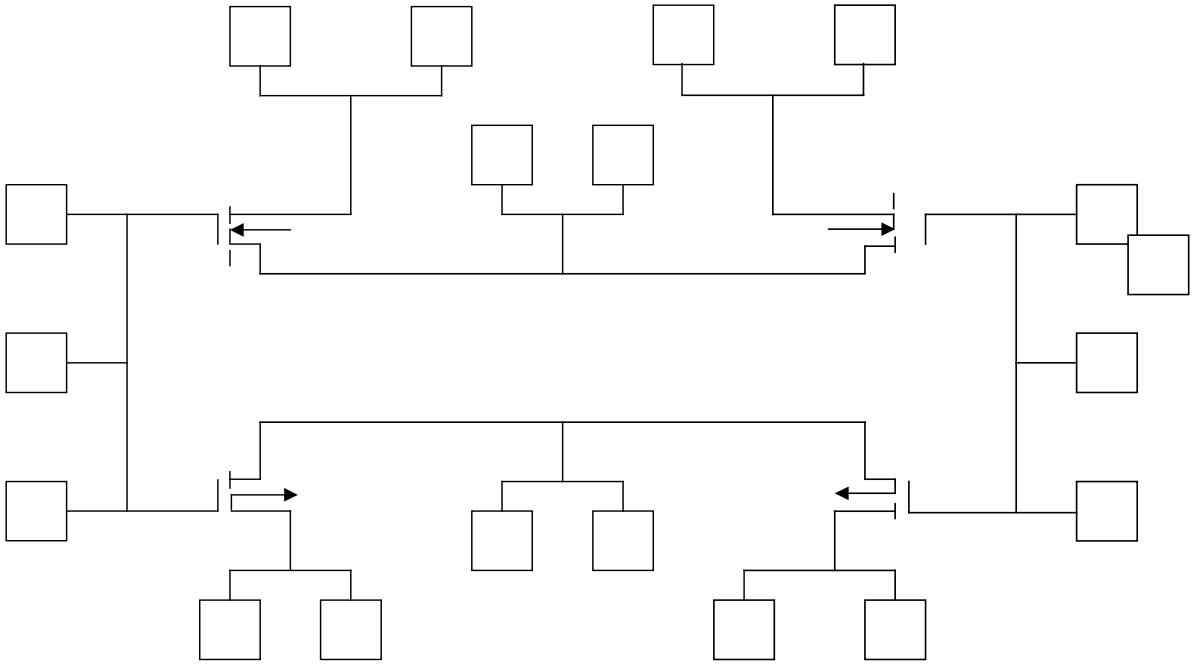


Fig. 2.15. Variant of the BMC base cell of the KMDN type

The number of transistors in the cell is selected based on the results of the analysis of the frequency of use of various logic elements in devices of this class, as well as the load and speed requirements. Crystals with the number of transistors in the cell 4, 8 or 10 give a high transistor utilization factor. The schematic diagram of a cell with 4 transistors is shown in figure2.15. Square elements are contact bars for roller shutters and are attached to cellular elements. Transistors can be connected in series or in parallel, that is, you can get typical AND-NOT and OR-NOT logic elements. In KMDN transistors with opposite conduction channels are always used in pairs, so pairs of transistors can have a common gate.

Cell complexity is achieved by combining simple cells into a group.

BMC parameters can be divided into 4 groups:

1. Functional capabilities (number of equivalent valves, type of BC, number of MBC and PBC, composition of the functional cell library).

2. Electrical parameters (voltage levels of logic signal coding, voltage and power, signal propagation delays, maximum switching frequencies).

3. Design and technological (case type, number of tips, number of metallization levels, crystal surface).

4. Characteristics of work.

The main parameters of some domestic BMCs representing different principle and technical types are given in table 1.5.

Table 1.5.

Parameters of the domestic BMC

Diagram	Type BMC	number of equivalent valves ths	number of contact zones	valve delay, ns	Valve performance MTV/exhaust
K1520XM6	Channel ESL	10	208	0,35	2
Staircase	Channel TTJIIII	50	256	2.5	0,05
From series«Poem »	KMDN	22	132	twenty	-
NAME-30	KMDN	thirty	202	2.0	0,015

The level of modern world technologies allows the production of BMCwith millions of equivalent gates, which have a delay of 0.1 ... 0.2 ns.

2.4 MODERN LARGE INTEGRATED SYSTEMS AND EXTREMELY LARGE INTEGRATED SYSTEMS WITH A COMPLEX PROGRAMMING STRUCTURE

2.4.1. General information

Microcircuits PLM, PML and BMC considered in chapter 1 marked the beginning of two main branches of the further development of software-structured logic circuits. CPLD (Complex Programmable Logic Devices) were the continuation of the PML line, and FPGA (Field Programmable Gate Arrays) of the BMC line (Fig. 2.16). The desire to combine the advantages of both lines led to the creation of VIS / HVIS with a mixed (combined) architecture, for which no generally accepted name has yet been developed (Altera uses the name FLEX (Flexible Logic Element Matri) - flexible). Increasing the level of integration has made it possible to place circuits on the crystal, the complexity of which corresponds to all systems. These diagrams are called SOC (Systems On Chip).

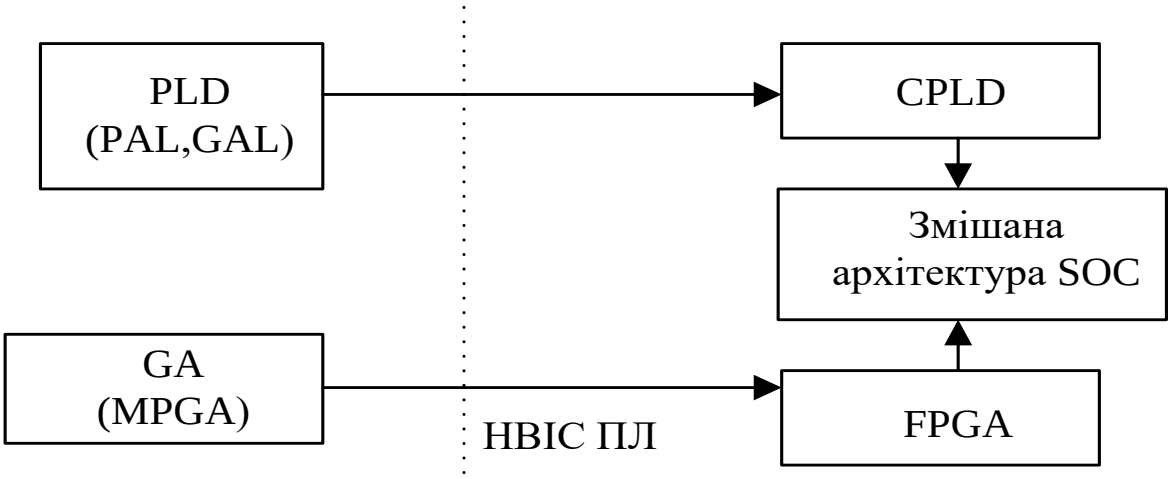


Fig. 2.16. Interdependence of generations of
NVIS programming logic

Several dozen companies have already been involved in PL NVIS development, including Xilinx, Altera, Actel, Atmel, AMD (Vantis), Lattice (all US) and several others. The above-mentioned companies quite fully represent the range of NVS PL products, although they do not deplete it. The scope of application of NVIS PL is extremely wide, not only large system units can be built on them, but also systems

as a whole, including memory and processors. NIVS PL are classified according to several features.

The division of NVIS PL by structural and technological type is presented in figure 2.17. Programmability, that is, the ability to perform a specific design on a standard NVIS ensures that it has a large number of bipolars, the conductivity of which can be set by the user, either very small (corresponding to an open key) or sufficiently large (corresponding to a closed key). The states of the keys determine one or another configuration of the circuit formed on the crystal. The number of programmable bipolar devices in NVIS PL depends on its complexity and can reach several million. Modern PL NVIS is characterized by the following types of programmable keys:

LISMOP transistors with a double gate;

Key transistors controlled by configuration memory triggers ("shadow" SD).

Programming using fusible links is done once. High-quality Actel jumpers are compact and have low currents in the non-conductive state. The programming voltage pulse breaks the jumper and creates a conductive channel with polysilicon between the electrodes. The magnitude of the current created by the programming pulse affects the diameter of the purge channel, which allows you to control the parameters of the conductive jumper.

EPROM-based systems require a long (about an hour) removal of the old configuration under the influence of ultraviolet radiation, with NVI soldering from the device and limiting the amount of programming due to the deterioration of material properties under the influence of ultraviolet radiation. In the case of the EEPROM configuration memory when erasing with electrical signals, it is not necessary to unsolder the microcircuit from the device, and a sufficiently large number of erasing cycles up to 100,000 is not allowed, erasing old and recording new information takes milliseconds. At the same time, the area of the electrical erasure memory element was initially approximately twice as large as the area of the UV erasure element. EEPROM circuits have improved recently and are increasingly replacing EPROM circuits.

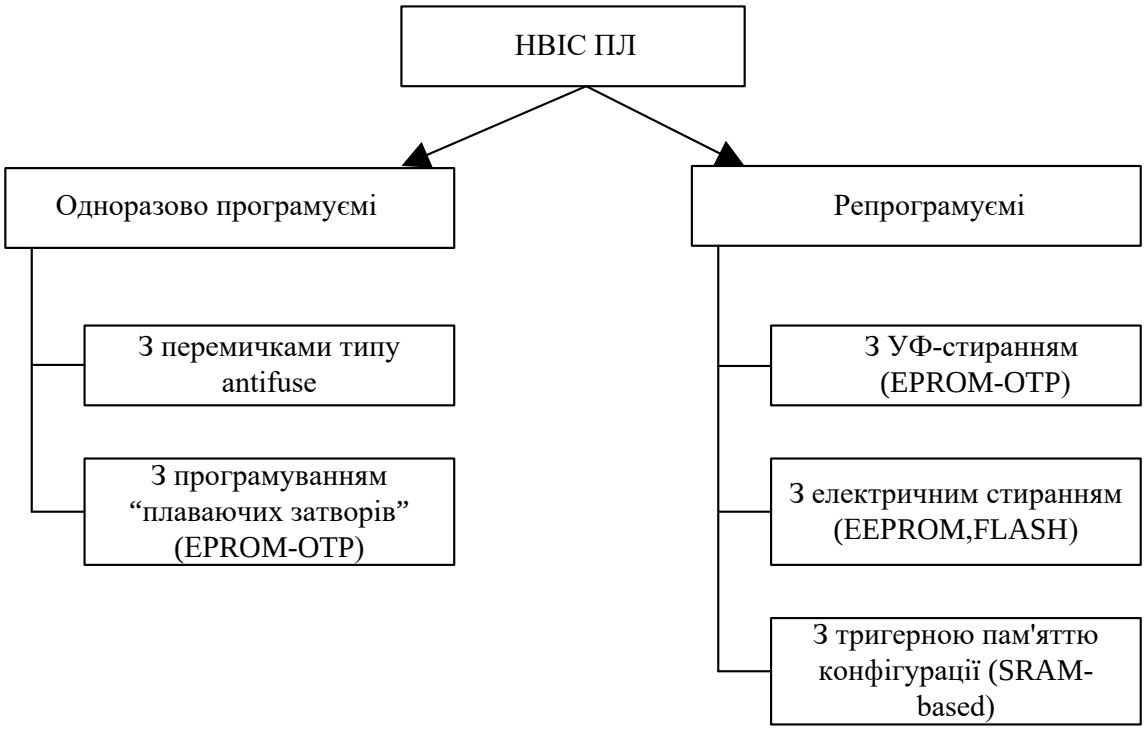


Fig. 2.17. Classification of NVIS PL by type of programmable elements.

A transistor switch controlled by a configuration memory trigger is shown in figure 2.18. Key transistor T2 closes or opens section 1-2 depending on the state of the trigger, the output of which is connected to the gate of transistor T2. During programming a high potential is applied to the sampling line and transistor T1 turns on. A signal sent from the read and write lines sets the trigger to a logic "1" or "0" state. In the operating mode the transistor T1 is closed, the trigger keeps its state unchanged. Since the configuration memory trigger does not require high speed, it is designed to be compact and optimized for maximum stability of stable ethanes. A perturbation of several volts for such a trigger does not affect its state. Configuration trigger memory (SRAM-based) circuits were first developed by Xilinx.

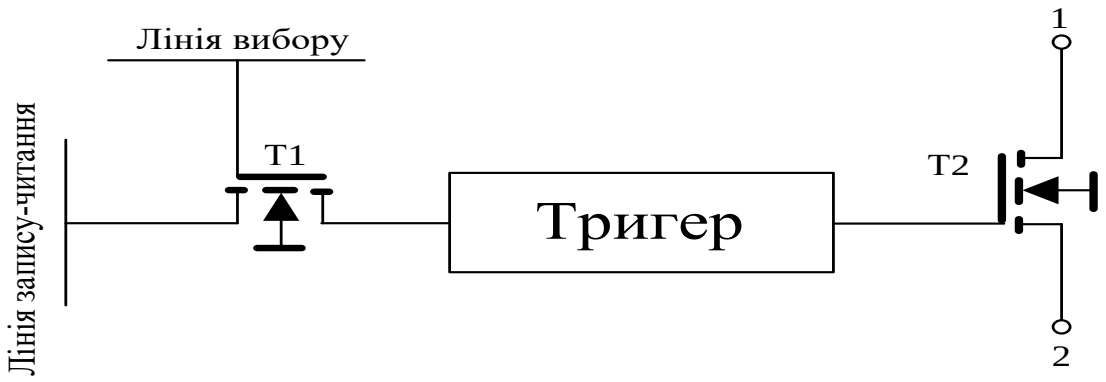


Fig. 2.18. Circuit of a key transistor controlled by a configuration trigger.

The loading of relevant data into the configuration memory is programmed by NVIS PL. The fast online programming process can be run an unlimited number of times. The configuration is destroyed every time the power is turned off in PL NVIS with trigger memory. After turning on the power, a process of programming (initialization) of the system is required - downloading configuration data from some non-volatile memory, which requires tens or even hundreds of milliseconds, if we are not talking about special NVIS PL with so-called dynamic online reprogramming.

Configuration memory triggers are arranged in the NVIS crystal interspersed with the circuit elements they set up. Reprogramming the PL NVIS with the configuration trigger memory occurs in the same mode as the operating mode by writing the code sequence to the trigger chain. Despite the increased complexity of the memory element and energy dependence due to these and other advantages, the most important place in the new FPGA and CPLD variants is occupied by PL NVIS with a trigger configuration memory.

Speaking about the general properties of PL NVIS, it should be noted that due to the regular structure they are implemented with a level of integration close to the maximum, since the method of programming the connection requires an additional crystal surface. Unlike BMC, NVIS PL is manufactured completely ready not only with logical elements, triggers, etc., but also with connections. The NVIS PL user

does not contact his manufacturer for any final operations, as he does the programming himself. This gives the reason to classify NVIS PL as a standard product with well-known advantages - mass production and cost reduction.

As in the development of other highly integrated microcircuits, in the case of PL NVIS great importance is attached to the issue of power reduction. As NVIS becomes more complex, power consumption becomes the most important factor. Since the power is proportional to the square of the supply voltage Ucs, then, if for a long time the typical supply voltage of microcircuits, including AI VIS / HVIS, was 5V, now it can be 3.3V; 2.7 V; even 1.8V1.5V.

Since the transmission of low-level signals via external communication is not acceptable due to the low stability of such signals, two supply voltages are often used in NVIS: higher for the data input/output circuits and lower for the main supply of the logic circuits. Therefore, it is possible to manufacture transistors with different threshold voltages for different areas of the crystal.

For high-speed, low-voltage circuits a deep submicron KMDN technology is being developed, which returns to the old idea of building a silicon-on-dielectric (SOI) circuit with the elimination of many parasitic circuit components. In circuits with especially low-voltage logic a very low power of the element is obtained, and at the same time such a density that the complexity of the circuit is reached up to 40 million gates (approximately 100 milliliters of transistors).

A hierarchy of power reduction modes is often used in PL NVIS circuits. In active modes during programming, the so-called *Turbo bit* selects one of two modes. The value ON of this bit increases the speed of the circuit when the power is limited to an acceptable value. If the value is OFF, this bit gives a reduced power (reduced speed) mode.

Special schemes for detecting changes in input signals are used to implement the turbo-discharge mode. Each input provides a simple circuit containing “OR”, “M2” and a delay element. Each change in the input signal is detected and leads to the supply of normal power to the circuit, which is necessary for fast switching processes in it. Then the power of the circuit is automatically reduced, the currents

reach the level of microamps, and the power consumption drops until new transients appear due to new changes in the input signals.

Standby mode is used when all input variables remain constant. In this case, the scheme remains ready for the trasition to operating mode. You can also use deep power reduction modes with very low power consumption. In these modes the diagram retains its information state, but the transition to normal speed mode requires the same defined time.

2.4.2. Programmable logic and NVIS programming with mixed architecture

Integrated Programmable Logic Circuits (PLICS) are architecturally based on the PLM framework. They consist of a central switching matrix, function blocks (FBs) called macrocells and input/output blocks. The SPLIS architecture is shown in figure 2.19.

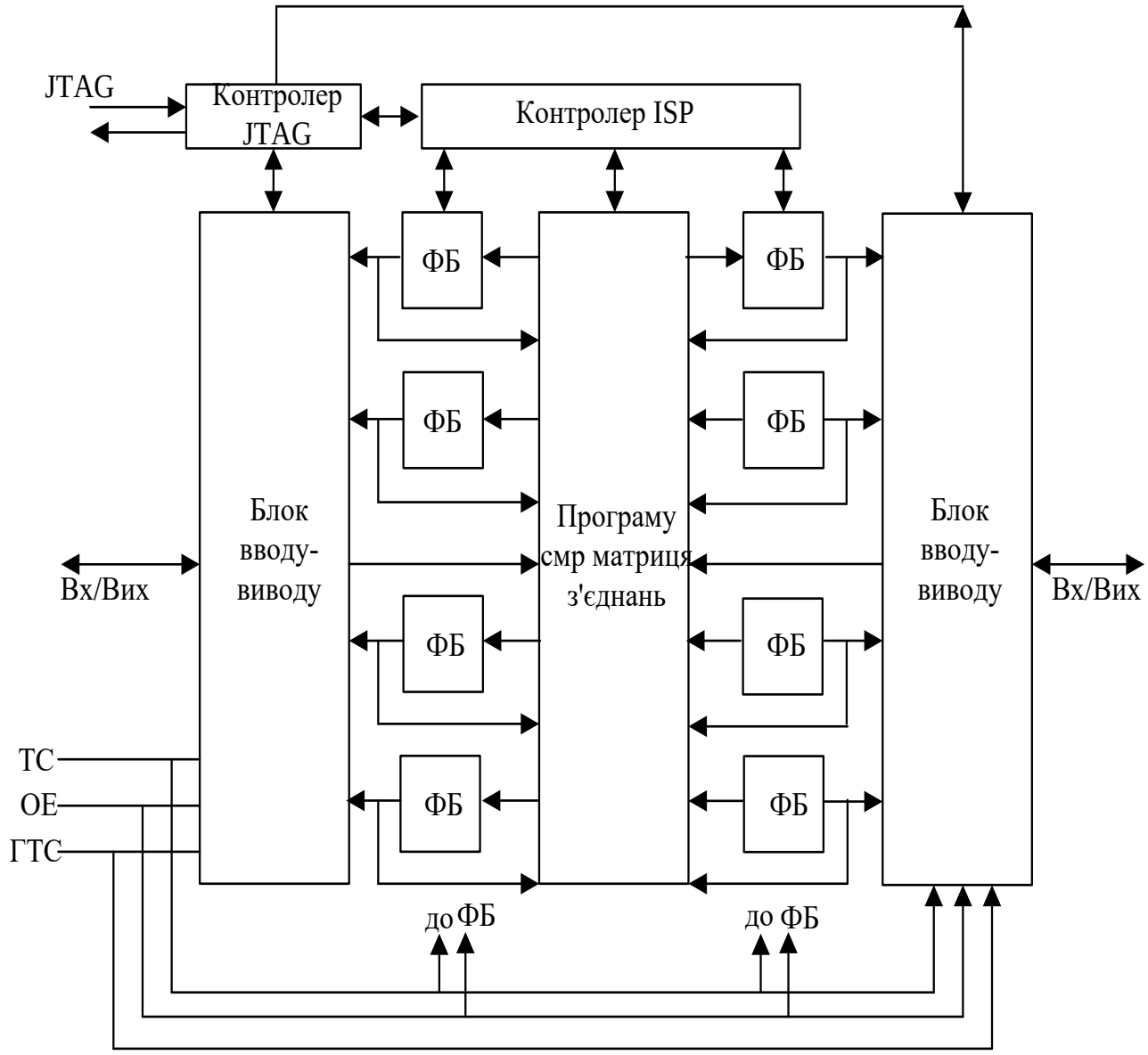


Fig. 2.19. The structure of the programmable logic circuit

The function blocks are similar to PLM and contain a multi-input programmable matrix of "AND" elements that generates consistent conditions from the variables imposed on its inputs, a group of "OR" elements among which the generated terms are distributed, and several other elements to extend the capabilities of the FB function blocks, they implement two-level logic with options for generating output (direct, inverse, combinational or register output).

Unlike typical PLM segment communication systems, SPLIS uses a continuous or one-dimensional continuous communication system with all links identical, which provides good predictability of signal delays on the channels - an important advantage that facilitates the design and manufacture of workable high-speed circuits. In the communication lines themselves, the number of programming keys is small, but many keys will not be used so a single-matrix switching system as a whole requires a fairly large number of keys.

A typical PMZ (fig. 2.20) allows you to connect the output of any FB with the inputs of others. The FB input is connected to horizontal lines crossing all vertical lines. Each input can be connected to any output by programming the connection points between the vertical and horizontal lines. In other words, IIMZ provides complete block switching capability. A local switching system similar to the global switching system can exist inside the FBs themselves.

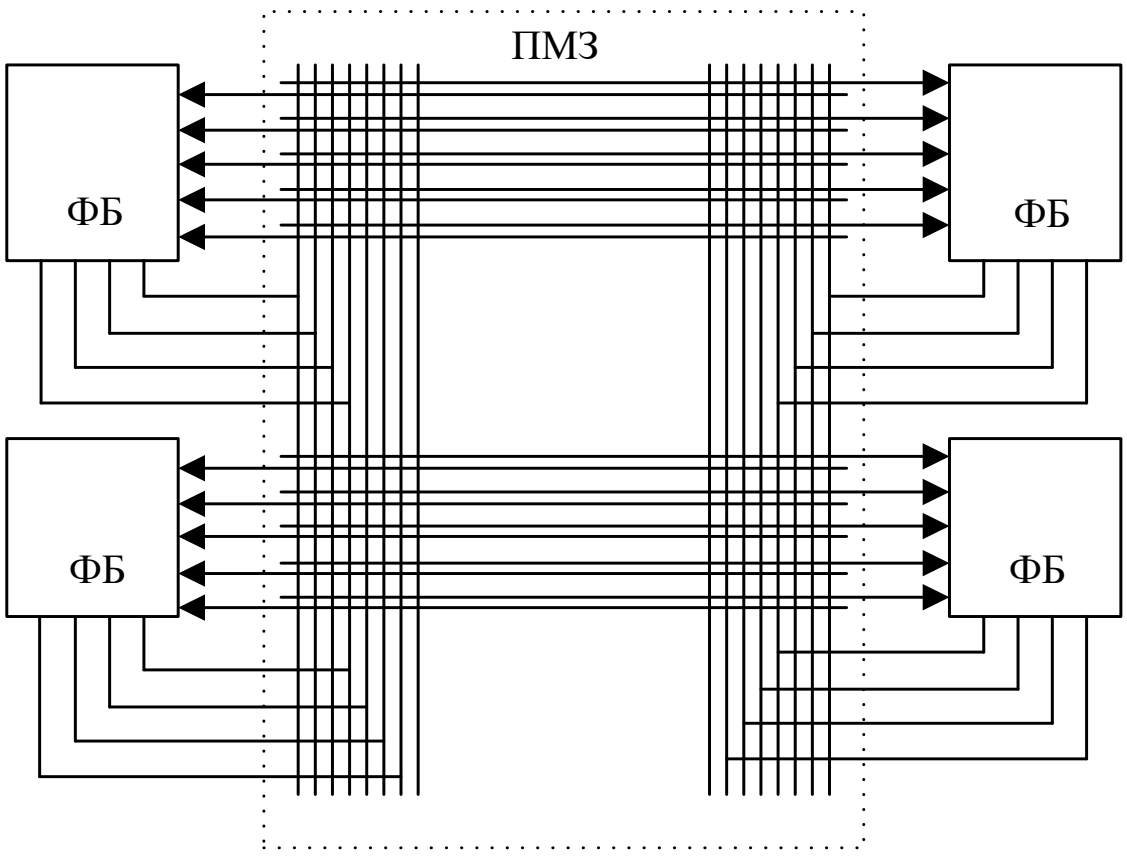


Fig. 2.20. Switching scheme of SPLIS functional blocks using the connection programming matrix

SPLIS with a mixed architecture is produced by many companies. The leading companies are Altera (MAX, FLEX, APEX family, etc.), Atmel (ATF 1500 family, etc.), Vantis (formerly known as AMD family, MASN family), etc. In each of them the consumer has a number of families of representatives, differing in complexity, cost and other parameters, in order to flexibly meet various needs. The total number of SPLIS varieties turns out to be very large. For a more detailed consideration, let's highlight a typical SPLIS among the products of leading companies.

Classic representatives of SPLIS are the MAX 7000 chips from Altera, which have an EEPROM configuration memory. For the "old" SPLIS with low complexity the configuration trigger memory was not characteristic at all. Later, with the development of deep submicron and multilayer metallization technology the situation changed, the configuration trigger memory also appeared in SPLIS, but at the same time their architecture also changed, so they should be classified as a mixed architecture DVIS PL (FLEX, APEX), which we will consider below.

2.4.3. Complex logic programmed by MAX 700 0

The SPLIS structure consists of logic blocks (LBs), each of which contains 16 macrocells (MCs), which receive terms from the local programmable matrix "AND", the matrix of programmable connections (PCM) and I/O blocks. The chips of the MAX 7000 family have the designation ERM 7XXX, where the number of MCs in this chip is mixed in the last three positions (from 32 to 256). Logical operations take place in logical blocks, which are combined into one diagram using PMC, each RO contains 16 MCs, so the junior representative has 2 logical blocks and the senior one has 16.

The PMC makes it possible to feed a signal to any LB input from any source and they are organized in such a way that the circuit of programmable keys and signal delays in the signal path are small. The LB input can receive a signal from any vertical PMC line that runs continuously along the entire length between the LB columns (fig. 2.20). Each logic block is connected to its I/O block, which has 6 to

12 pins. Part of the MC has no external outputs and is used to transmit a feedback signal to the PMC for building serial nodes.

Each logic block provides construction of combinational and sequential circuits. However, if the macro cell performs a given function and its frame terms are insufficient, the logical expander resource can be used. A logical expander (LE) is created by changing the terms of the previous macrocells which are transferred to the next ones, thus the macrocells are connected in a chain. You can create a chain whose length does not exceed eight links (fig. 2.21). It is clear that as a result of such actions, an additional delay for the appearance of the output signal is introduced.

Input/output blocks allow flexible control of the output buffer (fig. 2.21). The PMC generates six global OE outputs. For some representatives of the MAX 7000 family, it is possible to program an open-collector type output, additionally, you can program the rate of change of the output signal, etc.

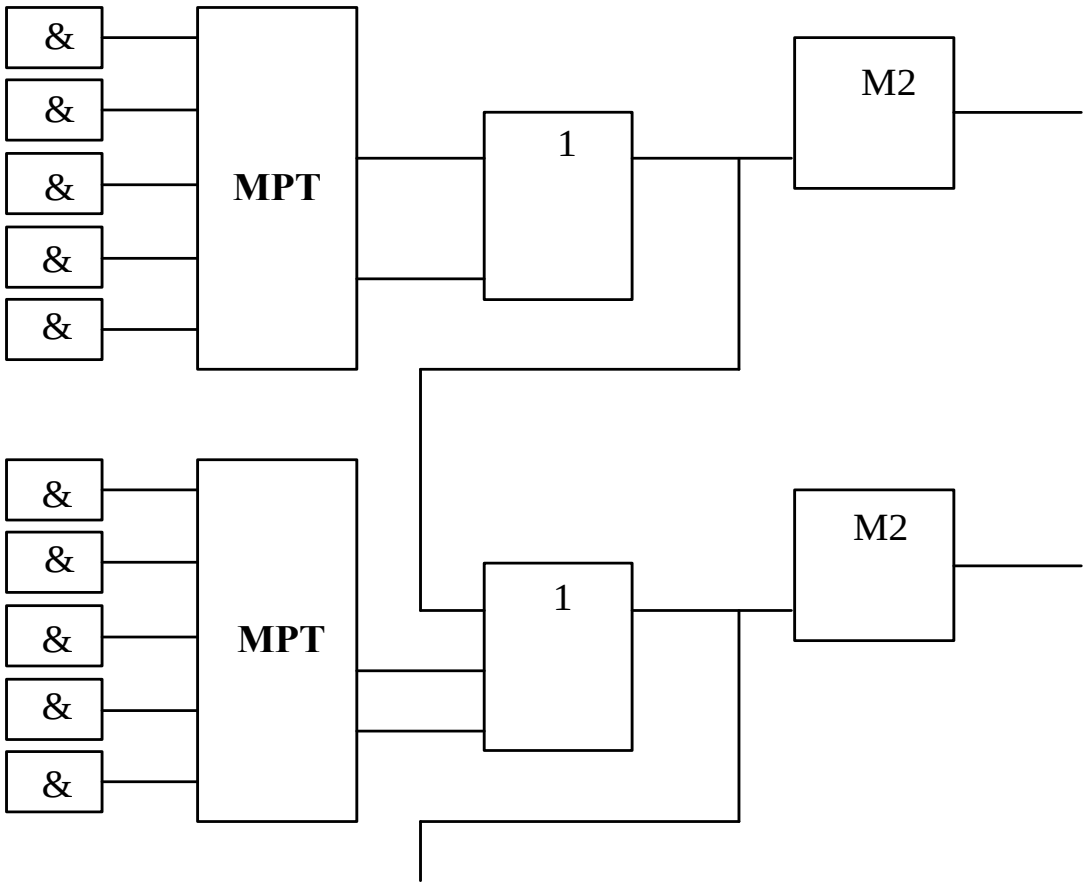


Fig. 2.21. The macrocell parallel expander circuit

Following the MAX 7000 family, which belongs to the second generation of CPLDs, Altera has released the third generation MAX 9000 family. Both families have a lot in common, but the third generation also has new features: JTAG interface, ISP programming without high voltages, etc.

2.4.4. Microcircuits of the FLEX 10K family

By architecture FLEX-type chipsets occupy an intermediate position between: classic versions of SPLIS and gate arrays (VM). Retaining a number of features of previous SPLIS developments, FLEX-type chipsets also have table-type logic elements, their logic blocks are arranged in the form of a matrix, and connecting channels pass horizontally and vertically between LBs, which is characteristic of virtual machines. At the same time, the paths in the channels are not segmented, but continuous, which is characteristic of SPLIS and ensures uniform predictability and low delay values.

A fragment of the structure of the FLEX 10K microcircuit, by which it is possible to judge the microcircuit as a whole (other parts are a repetition of the fragment). The fragment contains logical blocks that have a local PMC and consist of logical elements LE (Logic Element) table type, lines and columns of the global programming matrix from combinations of HPMZ elements and input-output elements attached to the ends of HPMZ rows and columns. In addition, the microcircuit contains modules of variable memory (VRAM).

For the first time such built-in memory appeared in the FLEX 10K family, previously memory resources consisted of elements used in logic converters of a massive type.

FLEX microcircuits implement a two-level division of means of logical data transformation. The smallest structural unit that performs logical operations is a logical element (LE). A compact group of eight LEs will form a Logic Array Block (LB). Logical blocks act as independent structural units of the next level of the hierarchy. They are all arranged in rows and columns that correspond to the rows and columns of the global programmable connection matrix.

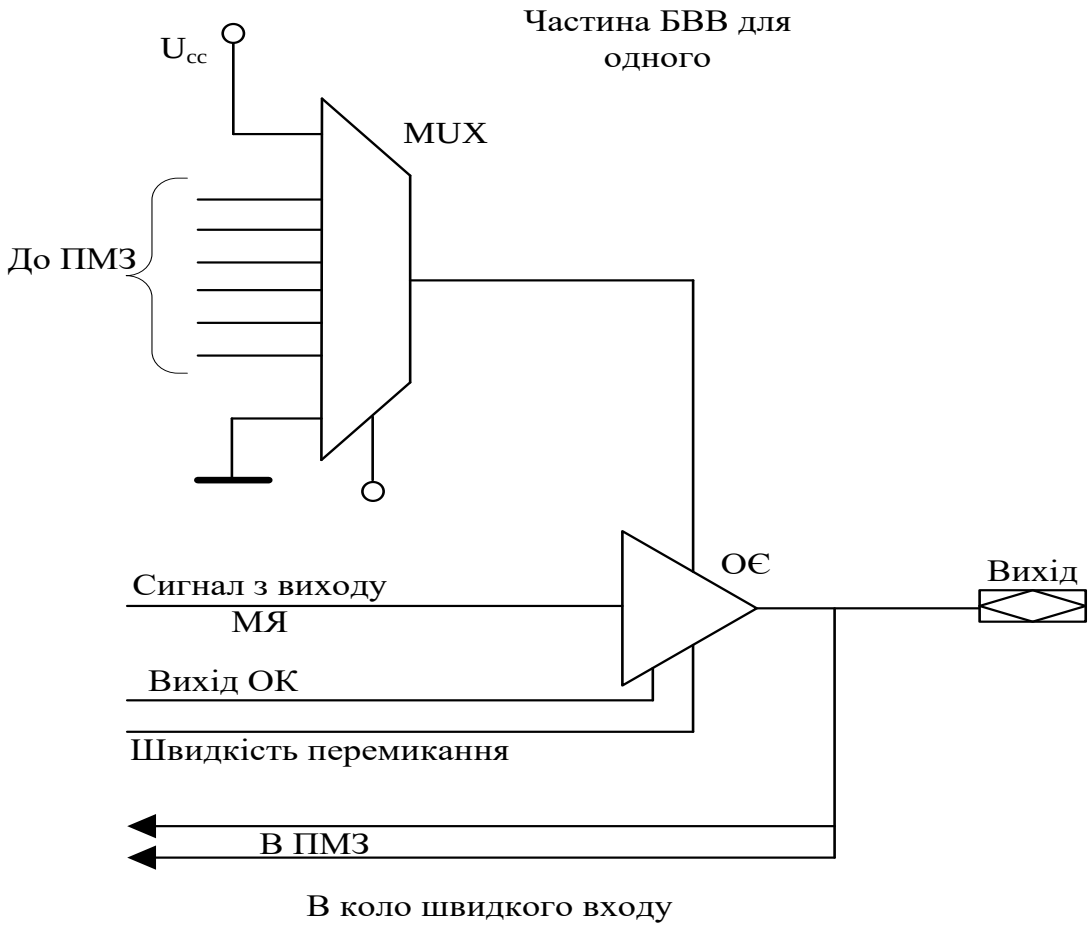
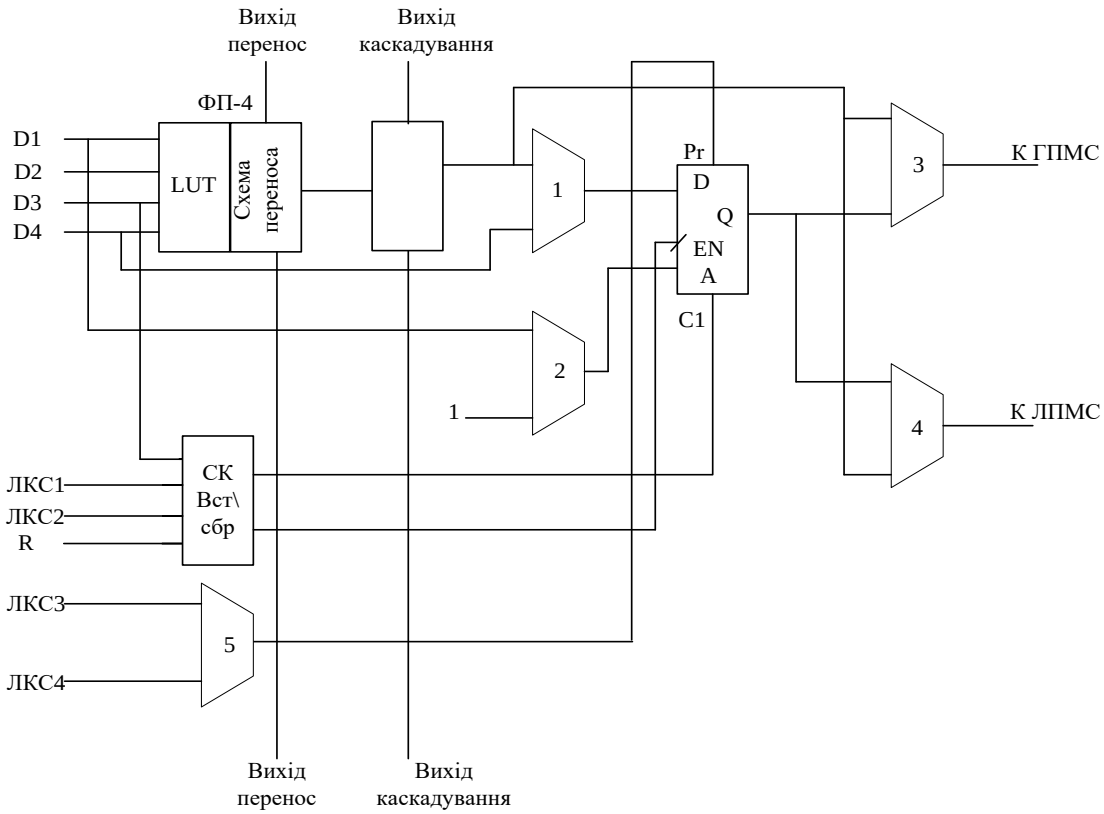


Figure 2.22. I/O block diagram.

2.4.5. NVIS logical element

The logical element has a four-level LUT (16-bit memory), a transfer circuit, a cascade circuit, a programmable trigger, a reset/set control trigger circuit and several programmable multiplexers (fig. 2.23). The functional converter FP-4 can also operate in the mode of reproduction of two functions of three variables (16 bits of memory in this case are divided into two blocks of 8 bits), one of which is a transfer function necessary for systems such as counters or adders with serial transmissions or other circuits with transfer functions. In FLEX microcircuits it was possible to implement high-speed transmission chains (with a delay of 1nc per cascade).



2.23.- The circuit of the logic element FLEX10K NVIS.

A functional converter performs functions with low delays. Synchronous splitter can act as trigger D, T, RS, JK. The input signals of asynchronous reset and factory reset are created by the control circuit, which receives two local control valves LUS1, LUS2, the common reset signal DVIS and the input variable D3. The installation/removal control circuit includes programmable multiplexers that allow you to define one of six trigger control modes. All modes are asynchronous - reset or get operations on different versions.

The trigger can be used both together with the combinational part of the logic element, and independently as a separate element, if the signal from the input D 4 enters its input through multiplexer 1.

LE output signals through multiplexers 3, 4 can be entered into the global and local programmable combinational matrix in the programmable combinational or register version. Trigger charging is possible from either of the two local control signals LUS3 and LUS4.

Functions with more than four arguments can be obtained in two ways as a composition of functions of 4 variables. The first method is to use the cascading templates that exist in each LE. A cascade diagram can be built on the basis of any logical operation except addition modulo 2. With a cascade diagram separate functions of 4 variables are combined into a function with a larger number of arguments. The second method, which is more common for DVIS PLs uses a feedback circuit to create a "function from a function". First, the logic elemens create functions that depend on no more than 4 arguments, and then these functions act as LE arguments, which is the final result. The capabilities of both methods correspond to the capabilities of the decomposition of the reproduced functions.

For the first time the FLEX 10K NVIS family uses ZMP (Embedded Array Blocks) variable memory arrays with a total capacity of approximately 6 Kbit to over 20 Kbit of static memory. Such blocks are in the middle of each row of the logical block matrix. Each block has 2048 programmable bits of memory. The possibilities of the block are not limited to its use as a tent block. The block can also be an array function converter to obtain complex functions (for digital signal processing, arithmetic logic operations, interface functions, etc.). For example, a 4x4 multiplier capable of operating at frequencies up to 50 MHz can be implemented in one block. If you build a similar device without using ZMP, then it would be necessary to occupy 8 logical LAB blocks and the maximum operating frequency would be about 20 MHz.

2.4.6. Programmable logic circuits of the "chip on a crystal" type.

Reducing the topological dimensions of the elements on the IMS crystal will lead to their speed of 500-600 MHz and the possibility of placing the entire system on them. The definition of NVIS as a crystal-based system arises from two factors: first, due to the high degree of integration, which allows placing a very complex circuit on the crystal (different functional blocks are simultaneously implemented by the same hardware); secondly, due to the fact that NVIS contains specially

designated areas - hardware cores. The introduction of specialized hardware cores has a number of positive consequences:

- implementation of complex functions requires a much smaller crystal surface;
- the circle of users is shrinking;
- versatility increases;
- reliability and durability of systems increases.

Thus, the introduction of specialized hardware cores, on the one hand leads to an increase in efficiency and a decrease in the cost of hardware, and on the other hand limits their sales market. Therefore, the most obvious choice is to use blocks of RAM as hardware cores. Each system needs these blocks to one degree or another. As memory cores, it is most profitable to use small blocks with the possibility of changing the memory organization and operating modes. Modern DVIS already uses SRAM - cells, 16-32 bits of RAM for each logical block. This memory can be used not only for LV configuration, but also combined into multi-space memory arrays. However, this option does not provide maximum speed and significantly reduces the amount of logic available to the user (ie, every 16-32 bits of memory the entire logic block is output, i.e. 10-20 logic elements). On average, a custom-designed 256-512-bit RAM block can be implemented in an area approximately 10 times smaller than that of a similar configuration mapped from memory cells distributed over the chip and access time is also reduced by 3-4 times.

2.4.7. Parameters of complex integrated circuits of programmable logic

VIS PL has many parameters, the most important of which are:

- programming frequency determined by the type of programmed elements;
- level of integration;
- speed of action;
- structured organization (PML, PLM, BMK, mixed architecture).

In addition to the parameters listed above, a number of others play an important role: the type of basic logical element, the presence of a JTAG interface,

programming in the system, compliance with standard interfaces, the availability of means of concealment, implemented project.

The level of integration of NVIS PL is estimated by the number of equivalent gates placed on the crystal. Actually estimating the number of goals is not such a simple task, because in one case they are not in their pure form, in the other - the blocks are not broken down into goals. Therefore, to assess the complexity of the crystal a technique is used, which consists of the following elements: the reference node is recorded on the diagram the maximum number of times by which the degree of integration is assessed. A set of simple, triggers and decoders was chosen as a reference circuit.

The first generation NVIS PL gateways had a fairly uniform structure and had two equivalent gateway numbers defined for them - a full number and the one that users could use for programming. Later, specialized areas of memory and other structural elements appeared, the complexity of DVIS in this case was determined without taking them into account.

The speed of NVIS is characterized by the delay of signal propagation along certain paths or the maximum possible frequency of the circuit as a whole. NVS PL parameters change very quickly, as new generations are produced using more advanced technologies.

2.5. JTAG INTERFACE AND PERIPHERAL SCANNING

The term JTAG refers to a set of tools and operations that allow testing VIS / DVIS without physical access to each of its outputs. The abbreviation JTAG comes from the name of the developer - Joint Test Action Group. The term "peripheral scanning" (PS) refers to testing according to the JTAG standard (IEEE St d 1149.1). Such testing is possible only for microcircuits inside which there is a set of special elements - peripheral scanning cells and circuits for controlling their operation.

The PS elements are placed between each external terminal of the chip and the crystal circuits that make up the device under test. More and more modern

microcircuits are equipped with a JTAG interface, that is, peripheral scanning capabilities. The basic concept of peripheral scanning is shown in figure 2.24. Cell scanning can work in different modes. In working mode, they simply pass signals through each other and do not change the operation of the device. At the same time, one cell is required for normal logic outputs, two for third-state outputs (the other for generating the buffer control signal) and three for bidirectional outputs, as shown in figure 2.24. Input signals pass through the PS elements directly to the corresponding main circuits of the crystal.

In the test mode the flow of signals through the cells stops, and the cells themselves, connected in series form a shift register, which also has additional functions. The test code can be entered into such a shift register from the input test data, which will be applied to the input pins of the main circuit of the crystal. The result obtained by the main circuit is loaded into the PS elements at its outputs and is then output sequentially for comparison with the expected correct result outside the device.

The PS scheme (fig. 2.25) includes two multiplexers and two triggers D. Depending on the "Mode" address of the input of the output multiplexer, the cell either freely passes the signal from the input to the output, or enters the state of the second flip. - flop on exit. The "Shift" input multiplexer address signal controls the supply of the first input flip-flop (from the logic inputs of the chip) or the signal from the previous cell. Thus, according to the synchronization signal for the first triggers and when the "IN" signal is transmitted through the input multiplexer, these triggers are loaded into all cells in parallel. When the signal from the previous cell is passed through the input multiplexer, the clock signal is shifted by one bit in the register formed by the serial connection of the cells.

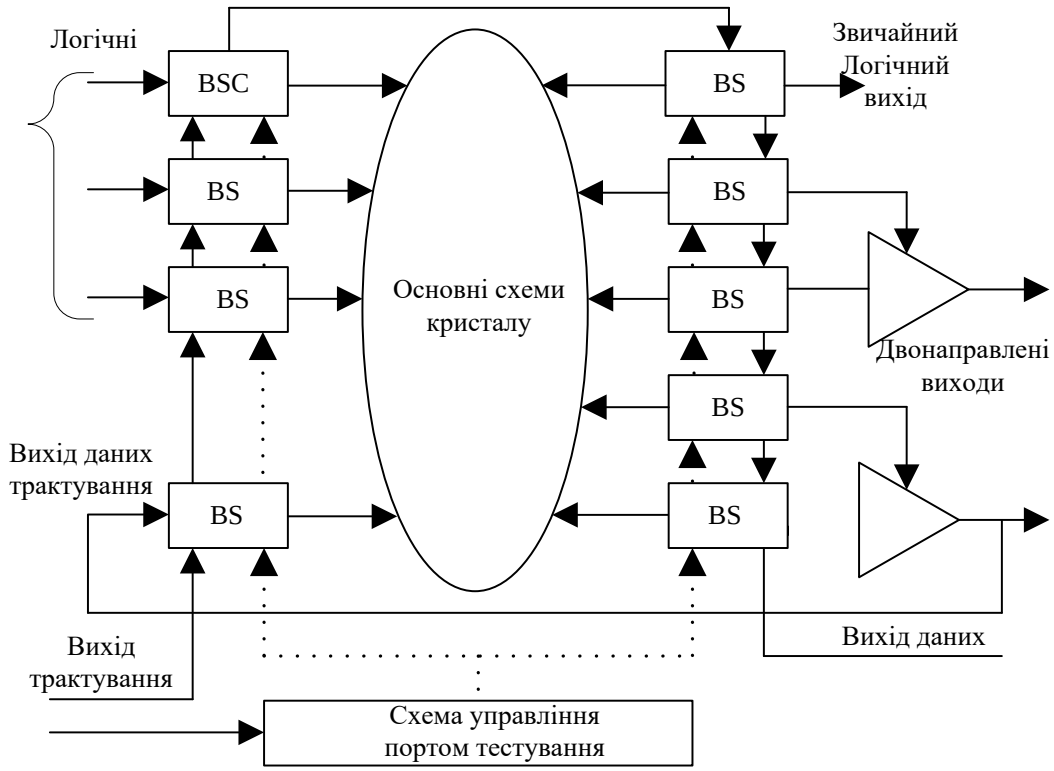


Fig. 2.24. Hardware structure of the JTAG interface.

According to the "Restore" synchronization signal the current contents of the register created from the chain of the first triggers are written to the static register consisting of the other flip-flops. Shifts in the register on triggers 1 will not affect the contents of the register on flip-flops 2.

Peripheral scanning allows you to check the operation of the microcircuits themselves, the mounting connections of the microcircuits to each other on the printed circuit board, reading signals at the output of the microcircuit during its operation or controlling these signals.

Checking the operation of the microcircuits themselves consists in their introduction and observation of the obtained result, checking the correctness of the installation of the microcircuits on the board is carried out, for example, in the case of interaction of two microcircuits with a JTAG interface. In this case, the test information is output to the output cells of one microcircuit, and then transcribed to the input cells of another. If all connections work, the information received is identical to the information entered.

The sequence of actions during a peripheral scan: loading the command code, loading the data, executing the command to read the result. Due to the advanced capabilities of the JTAG interface, it is also possible to reprogram the chips directly in the system without removing the chips from the device.

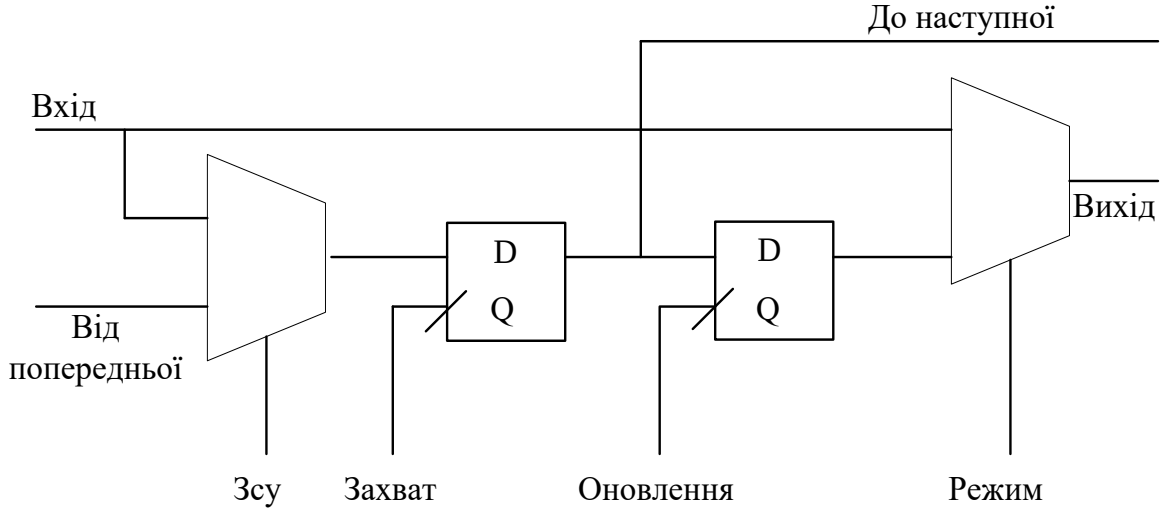


Fig. 2.25. The circuit of the peripheral scanning cell.

2.5.1. Programming in the system

Programming in the system is one of the most important advantages of DVIS PL, which allows you to easily change the logic of their operation. The need for changes arises both to eliminate errors not detected during the initial testing, and to modernize the system. The feature of the possibility of programming directly in the system is denoted by the abbreviation ISP (In System Programmable).

Creating a VIS/NVIS well suited for programming imposes additional requirements on its architecture and software development tools. The task of changing the operation of VIS/NVIS is easier to consider when using fine-grained logic blocks, the availability of thermal separation schemes, inversion control, flexible setting parameters and start-up time.

The programmability of the system is increased if some functions of the PL NVIS remain free during the design to simplify the design in the future. At the same time, the connection is recommended for speed, functionality and resources. If the original design occupies more than 90% of the chip capacity and its speed is close

to the limit, then to facilitate subsequent changes consider using a higher level of integration or a faster operating chip, as it is risky to expect a successful "upgrade" without resource reserves.

Please note that when reprogramming the system, the assignment of external pins must be saved, otherwise the PCB assembly will have to be changed.

PL NVIS configuration methods with trigger memory allow you to load it from external memory in various ways, the form of their presentation can be sequential or parallel, the role of the configured NVIS can be passive or active. It can be used to configure a PLC with a JTAG interface.

The HVIS PL uses several hundred bits of configuration for each logic block and its connection. Each bit causes a necessary state in the circuit of a logic block, programmable multiplexer, programmable communication key, etc. Configuration data is generated using the design automation system. Chipsets have three pins for one of six configuration modes. Three modes are called active in them, PL NVIS itself produces synchronization and control signals that need to be exchanged with the source of configuration data. In active serial mode data comes from serial PROM, in active parallel modes data comes from ROM with 8-bit organization, and then converted into a bit stream in the chip itself.

In passive modes the data source generates synchronization and exchange control signals. The most common mode is passive serial, in which the input data is entered into the chip on the leading edge of the clock pulse. Both modes are called peripheral, in which the NVIS acts as an external device of the microprocessor system accepting 8-bit data from the bus and regulating the speed of signal exchange. Peripheral mode is called synchronous if an internal generator generates clock signals to convert a stream of bytes into bits, and asynchronous if clock signals from an external source are used for this purpose.

It is possible to combine several PL NVIS in a serial chain and use a single bit stream to configure it. Among the PL NVIS there are those in which the trigger and non-volatile configuration memory are implemented at the same time. In this case,

it is possible to configure without additional external devices by automatic charging after switching on.

Microcircuits with software and analog-digital structures.

Many control and computer systems use information presented not only in digital, but also in analog form. Analog signals are produced by many sensors of physical quantities, the information about them should enter the digital part of the system. Some operations (normalization, filtering, analog-to-digital conversion, digital-to-analog conversion, etc.) are often necessary before and during the output of these values to digital devices. Therefore, posing the problem of integrating the entire system on one chip often leads to the problem of implementing programming tools for analog signal processing on this chip.

Analog and analog-digital fragments have long been embedded in the structures of microprocessor systems in the form of separate microcircuits of low and medium integration, which use discrete (mounted) control elements. BMC technology has also found application in the field of analog-digital technology, but until recently, the programming of analog and analog-digital circuits on crystals with a high degree of integration was practically not mastered by the user himself. The difficulties of mastering analog ICs are explained by the reduced accuracy compared to those based on discrete elements. ADCs and DACs blocks have been built into VIS (Analog Devices, Intel) for a relatively long time, but with very inconsistent results. The emergence of "crystal systems" made the problem of integration of reliable programmable analog and analog-digital circuits and the digital part of the system particularly relevant. On the basis of microcircuits with software structures analog and analog-digital signal processing subsystems can be quickly designed, debugged, manufactured, and entered the market.

Several large companies producing microcircuits with programmable structures (Motorol Lattice Semiconductor, Cypress) have already responded to the requirements of the time and paid attention to the development of programmable analog and analog-digital structures, made both on a separate crystal and with a digital part. systems.

Digital signals take only two values, one of which corresponds to a logical value and the second to a logical zero. There is no problem with the exact assignment of these signals - you just need to reliably distinguish one of these signals from the other. The situation is completely different in analog technology, where the signal must transmit the exact value of the quantity with an error of tenths or hundredths of a percent. Traditionally, the ratio of resistances of precision resistors played the role of dosing parameters in the circuits of amplifiers, adders, etc. RC time constants were determined jointly by the values of the resistance of the precision resistor and the capacity of the working capacitor. For example, in the circuit of the amplifier, that is, a device for duplicating a signal issued by a constant voltage two precision resistors R1 and R2, the functional characteristic of the circuit depends on the resistance coefficient, in an idealized form it has the form:

$$U_2 = -\frac{R_2}{R_1}U_1$$

where U1 and U2 are the input and output voltages, respectively. The translator has an idealized functional characteristic of the form:

$$U_2 = -\frac{1}{RC}\int_0^{\infty} U_1(t)dt$$

the role of the scale factor is played by the product of the resistance of the precision resistor of the input circuit and the capacitance of the capacitor of the feedback circuit.

In circuit engineering with discrete components the problem of implementing precision resistors has a satisfactory solution. For IC technology the problem is somewhat more complicated, but there is an alternative circuit solution, with the help of which resistors are simulated by circuits containing capacitors (fig. 2.26).

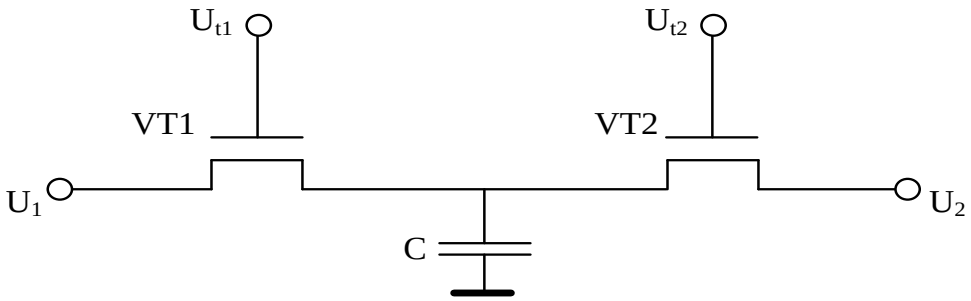


Fig. 2.26. Resistor modeling system in microcircuit engineering.

Such circuits include capacitor C and key transistors VT1 and VT2 controlled by clock voltages *Ut1* and *Ut2*. Transistors VT1 and VT2 are alternately closed under the action of clock voltages, and capacitor C is alternately connected through a closed key transistor to voltage *U1 or U2*. At the moment when the key transistor is closed, the charge on the capacitor changes by the value $q=q1=q2=C$ (*U1- U2*). The charge is changed by short pulses of current flowing through the capacitor when the corresponding key transistor is closed.

The average value of the current in the circle between points 1 and 2 is the value

$$i = q/t = (U_1 - U_2)C/T$$

where *T* is the period of clock pulses.

From obtaining the expression, it can be seen that at the average values of the signals, the circuit conducts as a resistor with resistance *R=T/C,* the resistance is proportional to the period of the clock pulses, that is, inversely proportional to their frequency. The difference is that the resistor regulates the flow of the load transmitted through it smoothly, and the circuit with switched capacitors - pulsed.

On the basis of circuits with switched capacitors, it is possible to build various working circuits similar to those known from traditional analog circuits, by replacing resistors with their equivalents. The fictitious resistances of the circuit are controlled by clock frequency *f=1/T*, that can be changed remotely. This puts analog devices on the same level with digital ones in terms of remote system programming. In circuits with switching capacitors circuits are built with the

dependence of the functional characteristic only on the capacitance coefficient, which can be set with high accuracy. It is circuits based on switched capacitors that make it possible to implement ADC and DAC converters with a high level of parameters. The capacity parameters are not critical to temperature changes and aging. The area of switched circuits and capacitors is significantly reduced (in hundreds of times) compared to circuits containing precision resistors. At the same time, the use of circuits with switched capacitors has its drawbacks. In circuits with continuous signals there are no problems with separating useful information (the average value of the pulsating value) from accompanying parasitic high-frequency components, which positively affects the dynamic characteristics of the devices. In addition, circuits with continuous signals have better noise characteristics.

Practical solutions

Motorola first announced the creation of a VIS with a number of programmable analog components (1997). These VIS were presented under the name MPAA020 (Motorola Field Programmable Analog Arrays), but were not presented in an industrial version. In 1999, Lattice Semiconductor released the ispPAC10 and ispPAC20 family of In - System Programmable analog circuits, followed by the ispPACSO. Cypress Semiconductor has released SOPC class microcircuits with implementation of digital and analog programming logic in a single chip.

Motorola chipsets use switched-capacitor circuits to create work units, Lattice Semiconductor chips use traditional scaling resistors, so continuous signals, Cypress Semiconductor's SOFCs use both types of circuit engineering, i.e. precision resistor circuits and switched-capacitor circuits.

Lattice Semiconductor ispPAC chipsets have a simple architecture (with few configurable resources and input/output pins). The configuration memory is implemented using EEPROM technology and can be loaded through specially designated contacts of the JTAG interface. The permissible number of reprogramming cycles is at least 10,000. The configuration can be closed from

unauthorized access with a secret bit. Configurable resources include not only analog devices, but also digital-to-analog devices (the ispPAC20 chip contains a built-in 8-bit digital-to-analog converter).

The ispPAC10 and ispPAC20 chipsets consist of two input (instrument) amplifiers and an output amplifier. IspPAC10 has four such macroelements and ispPAC20 has two.

The input and output circuits of the amplifiers are made in a differential circuit, protected from the influence of other signals, which makes the quality of signal transmission independent of the position of the communication line on the crystal and improves other properties of macroelements. The input resistances of amplifiers implemented in KMDN circuits are so large that they practically eliminate the problem of loading signal sources, including output sensors of physical quantities. Block connections are programmed from several macrocells and it is possible to build active filter circuits that can be rebuilt. The ispPAC20 8-bit DAC can accept data in either serial or parallel code. Comparators can be programmed to work with or without hysteresis. The connection or disconnection of the feedback resistors is programmed in the circuits of the output amplifiers. The capacitance of the feedback capacitors can be programmed from 1 to 63 pF (a total of 128 possible values).

Analog units should have consolidated functional and accuracy characteristics, taking into account static and dynamic errors. The microcircuits operate from a 5V power supply and have a 2.5V reference voltage source, which allows the amplifiers to be output from the center of the supply voltage for receiving and generating variable signals. The dynamic range is approximately 100 dB in the frequency range up to 100 kHz. A typical value of the harmonic ratio at 10 kHz is 88 dB, and at 100 kHz it is 67 dB. The frequency range for the created filters is 10-100 kHz, the frequency values are set with errors of no more than 3.5-5%. The operating temperature range is from -40 to +85 ˚C. The time to set the output voltage of the digital-to-analog converter with an error of 0.1% does not exceed 6 μs. Differential nonlinearity is transformed by no more units of lower

order. A typical switching time of a comparator with a 10 mV drop amplitude is 750 ns. The maximum current consumption of microcircuits is approximately 20 mA.

The relatively modest precision capabilities of ispPAC microcircuits are acceptable for building many functional units on them. Lattice Semiconductor software called RAS-Designer is intended for use on a PC with the Windows operating system. CAD allows you to enter information about the project, simulate the operation of diagrams, compile projects and download the results of the assembly to the VIS configuration memory.

Lattice Semiconductor has released the CY 8 C 25/26 family, which combines digital and analog programming devices.

SOPCCY8C25 / 26xxx have an eight-bit M8C processor core with a multiplier / storage. You can choose the clock frequency. The amount of program and data memory depending on the type of a chip varies from (4-16 KB of flash memory for programs from 128 to 256 bytes and SRAM memory for data).

In the CY8C25 / 26 microcircuits digital and analog programming blocks (SOC blocks) are located around the core of a high-speed processor, the operation of which will be determined during the system configuration (fig. 2.27).

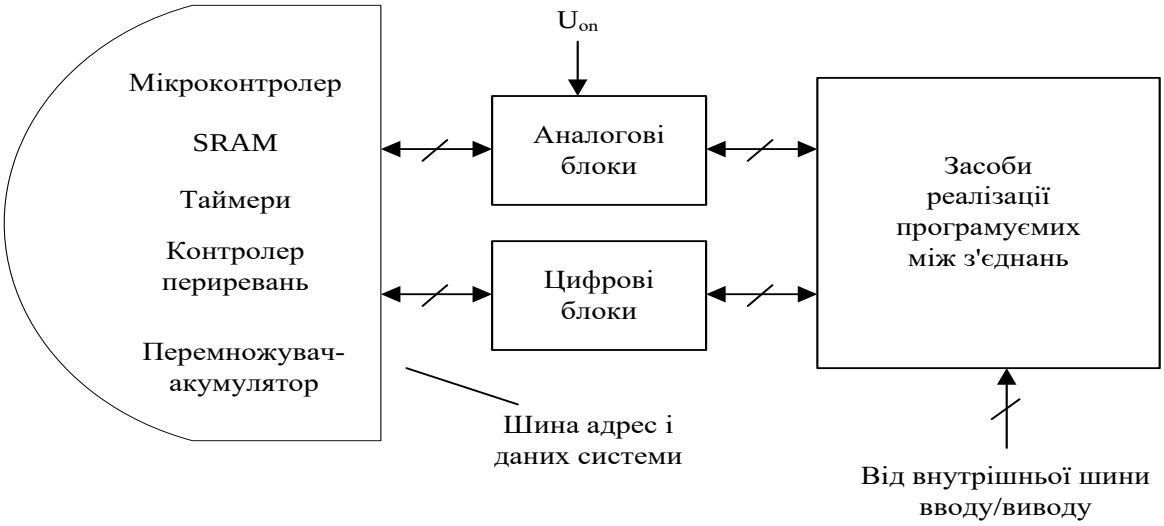


Fig. 2.27. The structure of the microcircuit of the CY8C25 / 26 family

Allocation of digital blocks into a separate part is a variant of the organization of the interface between the processor and the FPGA. SOC - blocks can be connected to each other in two main ways: in parallel and in series. With parallel connections the accuracy of the block is increased (increasing the bit rate of analog-digital converters, etc.). In a series connection the paths consist of several operational connections, such as a sequence of a preamplifier, a filter and an analog-to-digital converter.

The first released CY8C25 / 26 SOC unit contains 12 analog and 8 digital units. From the 12 analog blocks 4 are implemented in circuits with scaling resistors (blocks with continuous signals (CT blocks, time extension blocks)) and 8 are built on switched capacitors (SC, Switched-Capacitor and). The continuous signal blocks are optimized for building amplifiers with programmable gain coefficients and multiplexed input and output channels, differential amplifiers and fast comparators. Blocks using the technique of switching capacitors are intended for creating software filters, registers of successive approximations of analog-to-digital, digital-to-analog converters, etc. modulators.

Using analog and digital switching devices, including multiplexers and switches SOCs are combined into so-called user modules, for which a special library has been prepared. The library contains several types of analog-to-digital converters, digital-to-analog converters, comparators, devices for sampling and storing analog data, various types of filters (low-frequency, high-frequency, bandpass), voltage generators and detectors, special forms, modulators and demodulators (about 40 analog user modules in total).

The digital part of the library (about 30 user modules) includes timers, pulse-width modulators, transceivers and ready-made UART structures, an SPI interface, CRC generators, pseudo random sequence generators, and motor control circuits.

Chapter III GENERAL CHARACTERISTICS OF MICROCONTROLLERS

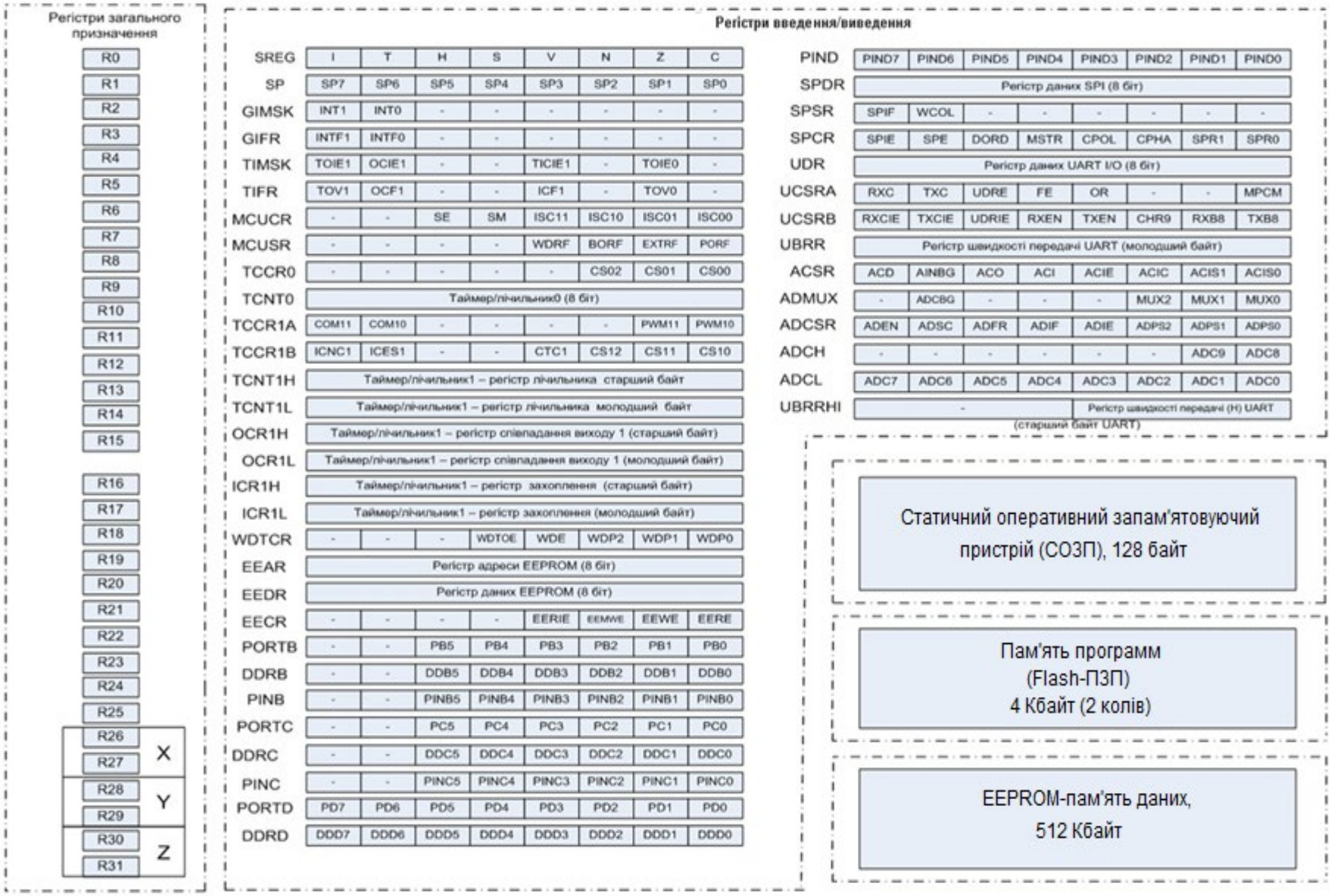
One of the main characteristics of the architecture of any microcontroller is the software model (programmer's model), which includes the part of the structure of the microcontroller accessible to the programmer using the command system. Let`s consider the software model of the Classic family of microcontrollers, which is the basis for other Tiny, Mega and XMega controllers. The program model of the microcontroller has several groups of registers with different functional purposes: general purpose registers (GPR), input/output registers (I/O), static random access memory (SRAM), which is called SRAM in technical literature in English, with a capacity of 128 bytes, EEPROM data memory with a capacity of 512 Kbytes and a non-volatile memory device (PSD) of the FLASH type with a capacity of 2K words (4 Kbytes).

3.1. Types of microcontroller registers

All general-purpose registers are combined into a file. Each file register has its own address ($00..$1F) in the data memory space. Therefore, they can be addressed as memory, despite the fact that physically these registers are not RAM cells.

AVR microcontrollers have 32 GPR which are directly accessible to ALP, unlike microcontrollers of other companies, which have only one such register, for example, register A (accumulator). Due to this any GPR can be used in all commands, both as a source operand and as a receiver operand. Exceptions are only five arithmetic and logical commands that perform actions between a constant and a register (SBCI, SUBI, CPI, ANDI, ORI), as well as a command to load a constant into a register (LDI). These commands can only access the second half of the registers (R16...R31).

Three 16-bit registers (index registers X, Y and Z), each of which is formed by combining two GPRs, are used as pointers for indirect addressing of data memory.


Регістри загального призначення
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
X
R28
R29
Y
R30
R31
Z
Регістри введення/виведення
SREG I T H S V N Z C
SP SP7 SP6 SP5 SP4 SP3 SP2 SP1 SP0
GIMSK INT1 INT0 - - - - - -
GIFR INTF1 INTF0 - - - - - -
TIMSK TOIE1 OCIE1 - - TICIE1 - TOIE0 -
TIFR TOV1 OCF1 - - ICF1 - TOV0 -
MCUCR - - SE SM ISC11 ISC10 ISC01 ISC00
MCUSR - - - - WDRF BORF EXTRF PORF
TCCR0 - - - - - CS02 CS01 CS00
TCNT0 Таймер/лічильник0 (8 біт)
TCCR1A COM11 COM10 - - - - PWM11 PWM10
TCCR1B ICNC1 ICES1 - - CTC1 CS12 CS11 CS10
TCNT1H Таймер/лічильник1 – регістр лічильника старший байт
TCNT1L Таймер/лічильник1 – регістр лічильника молодший байт
OCR1H Таймер/лічильник1 – регістр співпадання виходу 1 (старший байт)
OCR1L Таймер/лічильник1 – регістр співпадання виходу 1 (молодший байт)
ICR1H Таймер/лічильник1 – регістр захоплення (старший байт)
ICR1L Таймер/лічильник1 – регістр захоплення (молодший байт)
WDTCR - - - WDTOE WDE WDP2 WDP1 WDP0
EEAR Регістр адреси EEPROM (8 біт)
EEDR Регістр даних EEPROM (8 біт)
EECR - - - - EERIE EEMWE EEWE EERE
PORTB - - PB5 PB4 PB3 PB2 PB1 PB0
DDRB - - DDB5 DDB4 DDB3 DDB2 DDB1 DDB0
PINB - - PINB5 PINB4 PINB3 PINB2 PINB1 PINB0
PORTC - - PC5 PC4 PC3 PC2 PC1 PC0
DDRC - - DDC5 DDC4 DDC3 DDC2 DDC1 DDC0
PINC - - PINC5 PINC4 PINC3 PINC2 PINC1 PINC0
PORTD PD7 PD6 PD5 PD4 PD3 PD2 PD1 PD0
DDRD DDD7 DDD6 DDD5 DDD4 DDD3 DDD2 DDD1 DDD0
PIND PIND7 PIND6 PIND5 PIND4 PIND3 PIND2 PIND1 PIND0
SPDR Регістр даних SPI (8 біт)
SPSR SPIF WCOL - - - - - -
SPCR SPIE SPE DORD MSTR CPOL CPHA SPR1 SPR0
UDR Регістр даних UART I/O (8 біт)
UCSRA RXC TXC UDRE FE OR - - MPCM
UCSRB RXCIE TXCIE UDRIE RXEN TXEN CHR9 RXB8 TXB8
UBRR Регістр швидкості передачі UART (молодший байт)
ACSR ACD AINBG ACO ACI ACIE ACIC ACIS1 ACIS0
ADMUX - ADCBG - - - MUX2 MUX1 MUX0
ADCSR ADEN ADSC ADFR ADIF ADIE ADPS2 ADPS1 ADPS0
ADCH - - - - - - ADC9 ADC8
ADCL ADC7 ADC6 ADC5 ADC4 ADC3 ADC2 ADC1 ADC0
UBRRHI - Регістр швидкості передачі (H) UART
(старший байт UART)
Статичний оперативний запам'ятовуючий пристрій (СОЗП), 128 байт
Пам'ять программ
(Flash-ПЗП)
4 Кбайт (2 колів)
EEPROM-пам'ять даних,
512 Кбайт

3.1.1 Input/Output. registers

The input/output registers are located in the so-called input/output space of 64 bytes. All I/O registers can be divided into two groups: service registers of the microcontroller and registers related to peripheral devices (including input/output ports). The size of each register is 8 bits.

The allocation of I/O space addresses for the ATmega8515 microcontroller is shown in table 3.1.

Any I/O register can be accessed using the IN and OUT commands, which transfer data between one of the 32 registers and the I/O space. In addition, there are 4 bit-by-bit access commands whose operands are I/O registers: single-bit set/reset commands (SBI and CBI) and single-bit status check commands (SBIS and SBIC). Note that these commands can only access the first half of the I/O registers (addresses $00...$1F).

Just like RAM, I/O registers can be accessed in two ways: as registers themselves (using IN and OUT commands) and as RAM cells (except AT90S1200). In the first case, I/O registers addresses belonging to spatial input/output ($00...$3F) are used. In the second case, the I/O address must be increased by $20.

Table 3.1. I/O registers

Name	Function	Adress
		РВВ(О3П)
ACSR	Analog comparator control and status register	$08 ($28)*
ADCH	ADC data register (highest byte)	$05 ($25)
ADCL	ADC data register (low byte)	$04 ($24)
ADCSR	ADC control and status register	$06 ($26)
ADMUX	ADC multiplexer control register	$07 ($27)
DDRB	Port B data direction register	$17 ($37)
DDRC	Port C data direction register	$14 ($34)
DDRD	Port D data direction register	$11 ($31)
EEAR	EEPROM address register	$1E($3E)
EECR	EEPROM control register	$1C($3C)

EEDR	EEPROM data register	$10 ($30)
GIFR	General interrupt flag register	$3A ($5A)
GIMSK	General interrupt mask register	$3B ($5B)
ICR1H	Timer/Counter Capture Register 1 (High Byte)	$27 ($47)
ICR1L	Timer/Counter Capture Register 1 (Low Byte)	$26 ($46)
MCUCR	General microcontroller control register	$35 ($55)
OCR1H	Timer/Counter Coincidence Register 1 (highest byte)	$2B ($4B)
OCR1L	Timer/Counter Coincidence Register 1 (low byte)	$2A ($4A)
PINB	Outputs of port B	$16 ($36)
PINC	Outputs of port VC	$13 ($33)
FIND	Outputs of port D	$10 ($30)
PORTB	Port B data register	$18 ($38)
PORTC	Port C data register	$15 ($35)
PORTD	Port D data register	$12 ($32)
SP	Stack pointer	$30 ($50)
SPCR	SPI control register	$00 ($20)
SPDR	SPI data register	$0F ($2F)
SPSR	SPI status register	$0E ($2E)
SREG	Status register	$3F ($5F)
TCCR0	Timer/Counter Control Register 0	$33 ($53)
TCCR1A	Timer/Counter Control Register A 1	$2F ($4F)
TCCR1B	Timer/Counter Control Register B1	$2E($4E)
TCNT0	Timer/Counter Count Register 0 (8-bit)	$32($52)
TCNT1H	Timer/counter counter register 1 (highest byte)	$2D ($4D)
TCNT1L	Timer/counter counter register 1 (low byte)	$2C ($4C)
TIFR	Interrupt flag register from timers/counters	$38 ($58)
TIMSK	Interrupt mask register from timers/counters	$39($59)
UBRR	UART baud rate register (low byte))	$09 ($29)
UBRRHI	UART baud rate register (highest byte)	$03 ($23)
UCR	UART control register	$0A ($2A)
UDR	UART data register	$0C ($2C)
USR	UART status register	$0B ($2B)
WDTCR	Watchdog control register	$21 ($41)

3.1.2. Status Register (SREG)

The status register is located at address $3F ($5F). This register is a set of flags that show the current state of the microcontroller. These flags are automatically set to "1" or reset to "0" after certain events (according to the result of the commands). All bits of this register are available for both reading and writing at any time; after resetting the microcontroller all bits of the register are reset to "0". The description of this register is given in table 3.2.

Table 3.2. Status register SREG

Bits	Name	Description
7	I	General interrupt enable. To enable interrupts, this flag must be set to "1". You can enable/disable individual interrupts by setting or resetting the corresponding bits of the interrupt mask registers. If the flag is cleared (0), the interrupt is disabled regardless of the state of these bits. The flag is reset by hardware after entering interrupts and is updated by the RETI command, allowing subsequent interrupts to be processed.
6	T	Saving the copied bit. This register bit is used as a source or receiver by the BLD (Bit Load) and BST (Bit STore) bit copy commands. The specified bit of any RGP can be copied to this bit by the BST command or set according to the contents of the given bit by the BLD command
5	H	Half carry flag. This flag is set to "1" when performing some arithmetic operations, if there was a byte transfer from the junior tetrad (from the 3rd digit to the 4th) or a loan from the senior tetrad
4	S	Sign flag. This flag is equal to the result of an “exclusive OR” (XOR) operation between the N (negative result) and V (number overflow in complementary code) flags. Accordingly, this flag is set to "1" if the result of the arithmetic operation is less than zero. The flag allows you to increase the bit rate of the result of arithmetic operations from seven significant bits to eight (in additional code) with the S flag as the ninth significant bit
3	V	Additional code overflow check box. This flag is set to "1" when the bit grid of the signed result overflow.s It is used when working with signed numbers (given in the additional code)

2	N	A flag with a negative value. This flag is set to "1" if the most significant (7th) digit of the result of the operation is "1". Otherwise, the flag is equal to "0"
1	Z	Zero flag. This flag is set to "1" if the result of the operation is zero
0	C	Carry flag. This flag is set to "1" if the operation resulted in a byte overflow

3.1.3. Stack

As you know, there are two types of stack - hardware stack and software stack., The stack is implemented in software in the ATmega8515 microcontroller. In this case, the stack is placed in the RAM, and its depth is determined only by the size of the free RAM area. A pair of SPH:SPL registers located at addresses $ZE ($5E) and $3D ($5D) is used as the stack pointer, respectively.

The stack pointer registers are normal I/O registers and are therefore programmatically accessible. In addition, the instruction set of microcontrollers includes (PUSH) to stack and read from stack (POP) commands, which allows the program to use the stack for its needs. Since the stack pointer is equal to zero after applying power or after reset, it must be initialized at the beginning of the program by writing the value of the upper free address of the stack data memory (usually the maximum address of the SPD) to it. The stack is filled in the decreasing direction of the value of the SP register. The initial value of SP addresses the top of the stack - the first free cell of the stack memory.

When calling subroutines, the address of the instruction located after the RCALL or ICALL instruction is stored on the stack. The value of the stack pointer is reduced by 2, because 2 bytes are required to store the instruction counter. When returning from the subroutine, this address is removed from the stack and loaded into the command counter. The value of the stack pointer is accordingly increased by 2. The same happens during an interrupt.

3.2. Organization of memory

The memory organization of AVR microcontrollers of the classic family is based on the Harvard architecture, in which not only the address spaces of the program memory and data memory are divided but also the access buses to them.

Moreover, data memory consists of three areas: register memory, static RAM, and EEPROM-based memory. Due to the fact that register memory resides in the address space of RAM, these two areas of memory are usually referred to as one, which can be called static data memory (SDM). Each of the areas (SDM and EEPROM) lies in its own address space.

The generalized memory map of the AVR microcontrollers of the classic family is shown in figure 3.1.

It should be noted that:

• since AVR microcontrollers have a 16-bit command system, the program memory capacity in the figure is not in bytes, but in 16-bit words;

• the symbol "$" in front of a number means that this number is given in the hexadecimal number system.

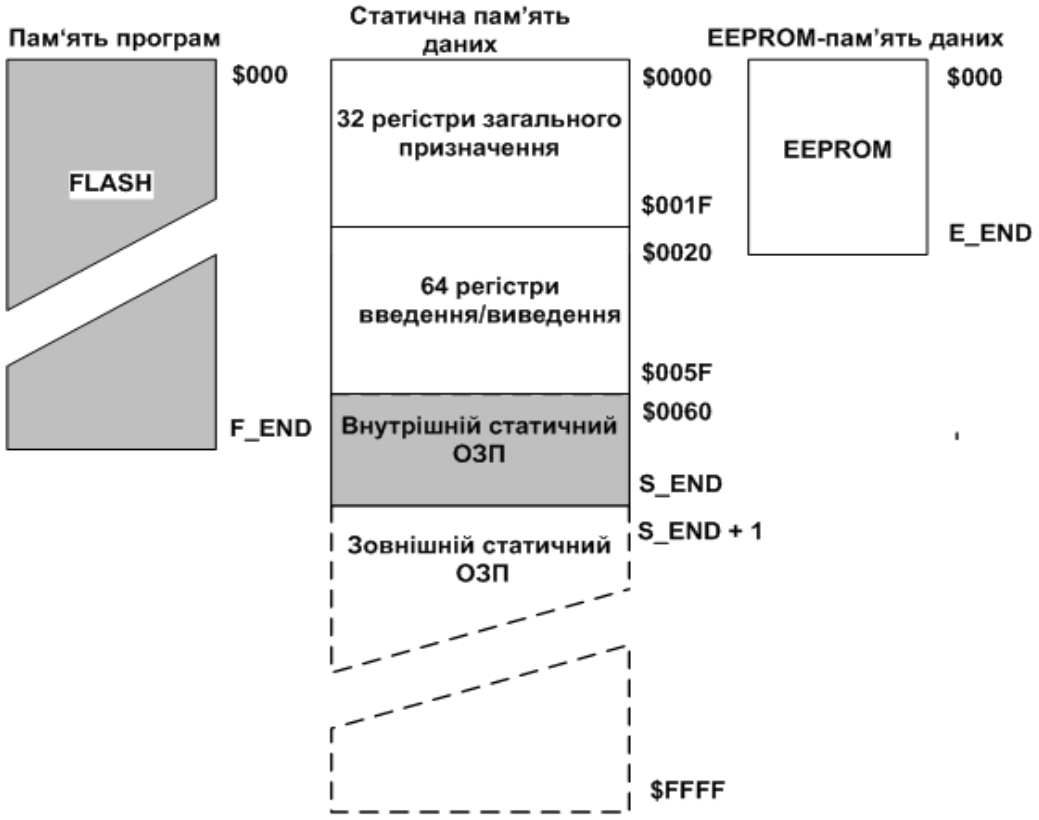


Figure 3.1. Memory card of AVR microcontrollers of the classic family

3.2.1. Ways of addressing operands

The type of addressing to the operands (data involved in the operation) is called the addressing method.

There are only four available addressing methods: implied, immediate, direct and indirect addressing. However, some addressing methods have several variations depending on which memory area is accessed (for direct addressing) or what additional operations are performed on the index register (for indirect addressing).

With implied addressing there is no explicit address of the operand in the command. The controller receives information about the required address of the operand from the operation code of the command, that is, it is said that the operand is addressed implicitly. So, for example, individual status register flags can be addressed in commands such as BRTS k, BRVS k, BRPL k, etc.

With immediate addressing the operand is included in the command itself and is read from the program memory a sa part of the machine code of the command. These are, for example, commands ADIW Rd, K6 (K=6 bits), SBCI Rd*, (K=8 bits), etc.

With direct addressing, the addresses of the operands are contained directly in the command word. In accordance with the structure of the static data memory, there are the following types of direct addressing: direct addressing of one RGP, direct addressing of two RGPs, direct addressing of I/OR, direct addressing of the entire SPD (RGP, I/OR and SOZP).

With indirect addressing, the address of the memory cell is contained in one of the index registers X, Y and Z. Depending on the additional manipulations performed on the contents of the index register, the following types of indirect addressing are distinguished: simple indirect addressing, relative indirect addressing, indirect addressing with pre-decrement and indirect addressing with post-increment.

3.3. Types (formats) of commands

Most microcontrollers of the AVR family have 14 types (formats) of commands shown in figure 3.2.

The first type includes commands which operation code occupies the entire length of the command (16 bits). For example, the LPM command belongs to this group - loading the general-purpose register R0 from the program memory at the address contained in the register pair Z=R31:R30. In addition, commands such as IJMP, ICALL, RET, RETI, NOP, SLEEP, and WDT have this format.

The second type includes commands that, in addition to the operation code, contain a five-bit address of one of the general-purpose registers. For example, these are DEC Rd commands; LD Rd, -X; ST Y, Rr, etc.

The third type has commands in which two operands-registers of general purpose are addressed: Rd - receiver, Rr - source. These are, for example, commands ADD Rd, Rr; CPSE Rd, Rr; AND Rd, Rr, etc.

The fourth type includes two-operand commands, in which one operand: K6=6 bits (K=0...63) is included in the command itself (direct addressing), and the second is a pair of registers: R24, R25; R26, R27; R28, R29; R30, R31, which are encoded by two command bits (dd): 00-R24, R25; 01-R26, R27; 10-R28, R29; 11-R30, R31. For example, the commands ADIW Rdl, K6; SBIW Rdl, K6 where Rdl = R24/R26/R28/R30.

The fifth type has two-operand commands, for example, SBCI Rd*, K; ORI Rd*, K, in which Rd* is one of the sixteen RGPs (R16...R31), and the second K is an eight-bit constant (immediate operand): $0 \leq K \leq 255$.

The sixth type includes commands in which one of the two operands can be a RGP (Rd is the receiver, Rr is the source), and the second is contained in the SPD and is addressed using indirect relative addressing, when the contents of the index register Y or Z are added to the offset q ($0 \leq q \leq 63$). These are, for example, LDD commands Rd, z+q; STD Y+q, Rr, etc.

The seventh type has commands in which one of the eight bits of the RGP is addressed. For example, commands BLD Rd, b; BST Rr, b; SBRC Rr, b, etc., where

Rd is the receiver, Rr is the source (one of the 32 RDPs), and b=0...7 is the bit number of the RGP.

The eighth type includes commands in which one operand is RGP (Rd - receiver / Rr - source), and the second is contained in one of the 64 I/ORs. For example, commands IN Rd, P; OUT P, Rr, where P is the I/ORs address (P=0...63).

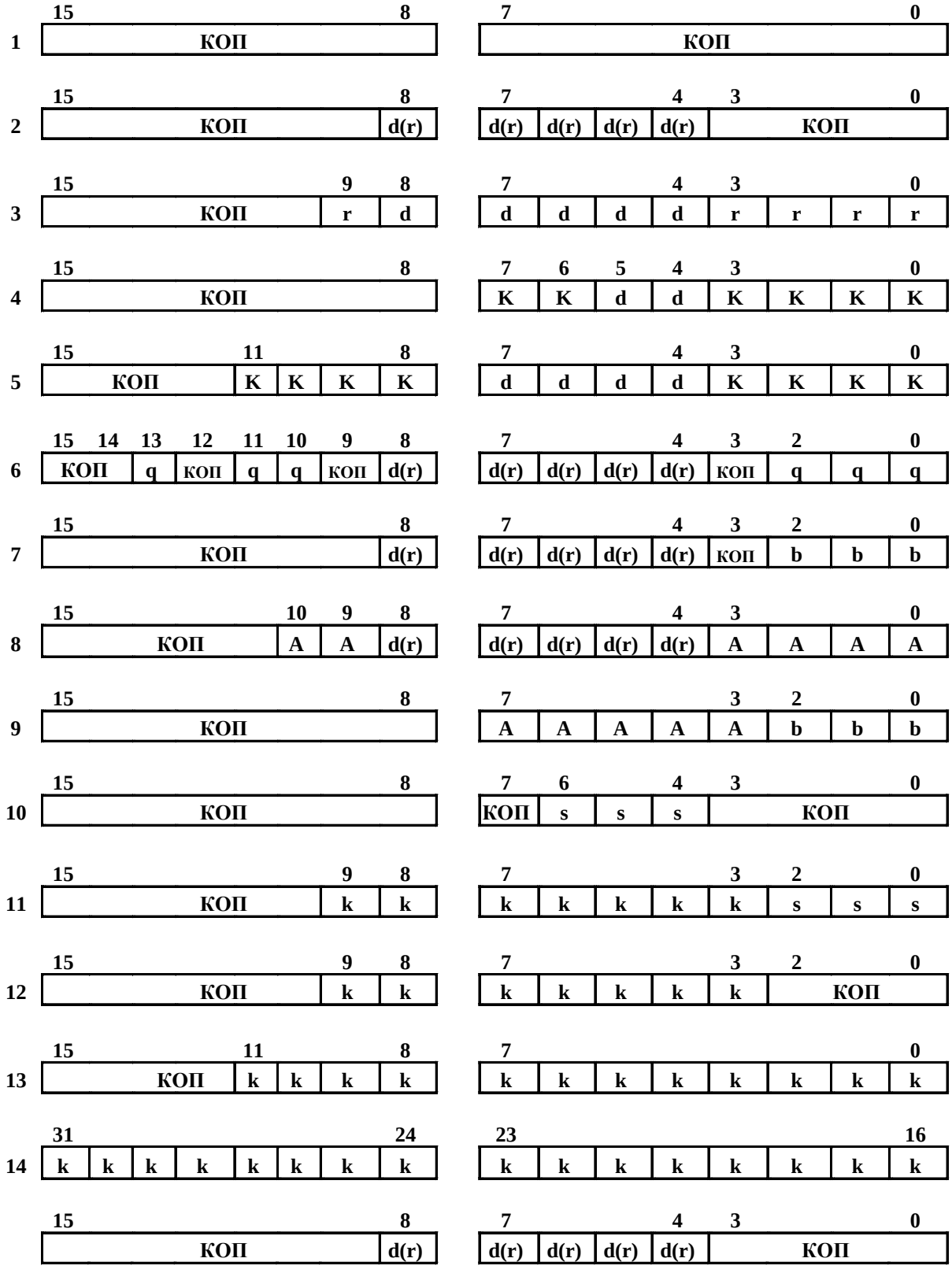


Figure 3.2. Command Types AT90S4433

The ninth type has commands in which individual bits of the lower half of the I/ORs are addressed (P*=0...31). For example, SBI P*, b commands, where b=0...7 is the I/ORs bit number.

The tenth type includes commands in which one of the 8 bits of the SREG flag register is addressed. For example, BSETs commands; BCLR s, where s=0...7 is the bit number in the status register.

The eleventh type has conditional transition commands depending on the value of the indicated bits of the status register. For example, BRBS commands s, k; BRBC s, k, where s=0...7 is the bit number of the status register SREG, and k=7 bits ($-64 \leq k \leq 63$) are added to the address of the next command when transitioning.

The twelfth type has conditional relative transition commands depending on the values of individual status register flags (the flags are addressed implicitly). For example, commands BREQ k; BRCC k, etc., where k=7 bits ($-64 \leq k \leq 63$) are added to the address of the next command on transition.

The thirteenth type includes commands of relative unconditional transition: RJMP k and relative unconditional subroutines call RCALL k, where k=12 bits ($-2048 \leq k \leq 2047$).

The fourteenth type has the following commands: LDS Rd, k – direct loading and STS k, Rr – direct saving, in which one operand is RGP (Rd – receiver/Rr – source), and the second is contained in SPD (RGP, I/ORs and SOZP); k=16 bits ($0 \leq k \leq 65535$) and addresses individual SPD cells.

CO- operation code; K is a data constant; k – address constant; b – specifies the bit number in the input/output register or register file (3 bits); s – defines the bit number in the status register (3 bits); A is the register address in the input/output space; q – sets the offset for indirect addressing (6 bits); d(r) is the receiver register (source) from the register file area (RGP).

3.4. Data types (formats).

It is necessary to know not only the formats of commands (figure 3.3), which control the operation of the microcontroller, but also the formats of data (operands) involved in the execution of one or another command (operation) (figure 3.3) when working with a microcontroller.

Figure 3.3. AT90S4433 microcontroller data types (formats).

The commands of this group are intended for forwarding the contents of cells located in the address space of static data memory (RGP, I/OR and SOZP). The division of the address space into three parts (RGP, I/OR, SOZP) determines the diversity of the commands of this group. Data forwarding performed by group commands can be performed in the following directions:

• RGP <=> RGP;

• RGP <=> I/OR;

• RGP <=> static data memory (RGP, I/OR and SOZP), 4 types of addressing.

This group also includes the PUSH and POP stack commands (absent in AT90S1200), which allow you to store the contents of the RGP in the stack and update it from the stack.

The execution of the commands of this group requires from one to three machine cycles depending on the command.

Table 3.3. Data transfer commands

№	Mnemonic notation	Operands	Description	Operation	*Type*	Number of cycles
1	MOV	Rd, Rr	Register copying	Rd □□Rr	3	1

2	LDI	Rd*, K	Loading a constant	Rd* □□K	5	1
3	LD	Rd, X	Indirect loading	Rd □□(X)	2	2
4	LD	Rd, X+	Indirect loading with post-increment	Rd □□(X), X □□X + 1	2	2
5	LD	Rd, - X	Indirect loading with pre-decrement	X □□X - 1, Rd □□(X)	2	2
6	LD	Rd, Y	Indirect loading	Rd □□(Y)	2	2
7	LD	Rd, Y+	Indirect loading with post-increment	Rd □□(Y), Y □□Y + 1	2	2
8	LD	Rd, - Y	Indirect loading with pre-decrement	Y □□Y - 1, Rd □□(Y)	2	2
9	LDD	Rd,Y+q	Indirect bias loading	Rd □□(Y + q)	6	2
10	LD	Rd, Z	Indirect loading	Rd □□(Z)	2	2
11	LD	Rd, Z+	Indirect loading with post-increment	Rd □□(Z), Z □□Z+1	2	2
12	LD	Rd, -Z	Indirect loading with pre-decrement	Z □□Z - 1, Rd □□(Z)	2	2
13	LDD	Rd, Z+q	Indirect bias loading	Rd □□(Z + q)	6	2
14	LDS	Rd, k	Direct loading	Rd □□(k)	14	2
15	ST	X, Rr	Indirect storage	(X) □□Rr	2	2
16	ST	X+, Rr	Indirect storage with post-increment	(X) □□Rr, X □□X + 1	2	2
17	ST	- X, Rr	Indirect storage with pre-decrement	X □□X - 1, (X) □□Rr	2	2
18	ST	Y, Rr	Indirect storage	(Y) □□Rr	2	2
19	ST	Y+, Rr	Indirect storage with post-increment	(Y) □□Rr, Y □□Y + 1	2	2
20	ST	- Y, Rr	Indirect storage with pre-decrement	Y □□Y - 1, (Y) □□Rr	2	2
21	STD	Y+q,Rr	Indirect storage with displacement	(Y + q) □□Rr	6	2
22	ST	Z, Rr	Indirect storage	(Z) □□Rr	2	2

23	ST	Z+, Rr	Indirect storage with post-increment	(Z) □□Rr, Z □□Z + 1	2	2
24	ST	-Z, Rr	Indirect storage with pre-decrement	Z □□Z - 1, (Z) □□Rr	2	2
25	STD	Z+q,Rr	Indirect storage with displacement	(Z + q) □□Rr	6	2
26	STS	k, Rr	Direct storage	(k) □□Rr	14	2
27	LPM*		Loading from program memory	R0 □□(Z)	1	3
28	IN	Rd, P	Reading the port	Rd □□P	8	1
29	OUT	P, Rr	Write to the port	P □□Rr	8	1
30	PUSH	Rr	Adding a register to the stack	STACK □□Rr; SP□□SP - 1	2	2
31	POP	Rd	Popping a register from the stack	SP□□SP + 1, Rd □□STACK	2	2

An example of a program using data transfer commands

```
ldiR17, $32 ;       Load constant                   R17 = $32
ldi R23, $45;       Load constant                   R23 = $45
mov R17, R23        ; Copy register                 R17 = R23
ldi R26, $60;       Load constant                   R26 = $60
ldi R27, $00        Load constant                   R27 = $00
ld R3, X      ; Indirect download                   R3 = X
ld R4, X+     ; Indirect loading with post-increment,   R4 = X, X = X + 1
ld R2, -X     ; Indirect loading with predecrement,  X = X - 1, R2 = X
ldi R28, $61;       Load constant                   R28 = $61
ldi R29, $00; Load constant                         R29 = $00
ld R5, Y      ; Indirect download                   R5 = Y
ld R6, Y+     ; Indirect loading with post-increment,   R6 = Y, Y = Y + 1
ld R7, -Y     Indirect loading with predecrement,   Y = Y - 1, R7 = Y
```

```
ldd R8, Y+2        ; Indirect load with offset, load the contents of SRAM into R8
                   by; address                    (Y + 2)=$63
                   ;
ldi R30, $62; Load constant                       R30 = $62
ldi R31, $00; Load constant                       R31 = $00
ld R11, Z    ; Indirect download                      R11 = Z
ld R10, Z+   ; Indirect loading with post-increment,  R10 = Z, Z = Z + 1
ld R12, -Z   ; Indirect loading with predecrement,    Z = Z - 1, R12 = Z
ldd R18, Z+5       ; Indirect bias loading,           R18 <= (Z + 5)
lds R9, $001       ; Immediate download                    R9 = $001
lds R10, $022      ; Immediate download                    R10 = $022
lds R11, $080      ; Immediate download                    R11 = $080
st X, R2     ;     Indirect storage                 X = R2
st X+, R10   ; Indirect storage with postincrement  X = R10, X = X + 1
st -X, R11   ; Indirect storage with predecrement   X = X - 1, X = R11
st Y, R7     ; Indirect storage                     Y = R2
st Y+, R6    ; Indirect storage with postincrement  Y = R6, Y = Y + 1
st -Y, R4    ; Indirect storage with predecrement   Y = Y - 1, Y = R16
std Y+2,R16        ; Indirect storage with displacement          (Y + 2) <= R16
st Z, R6     ; Indirect storage                     Z = R6
st Z+, R7    ; Indirect storage with postincrement  Z = R7, Z = Z + 1
st -Z, R17   ; Indirect storage with predecrement   Z = Z - 1, Z = R17
std Z+4,R30; Indirect bias loading            (Z + 4) <= R30
sts $82, R7  ; Immediate storage                     $82 <= R7
ldi R30, $01; Load constant                       R30 = $01
lpm          ; Downloading programs from memory                R0 = Z
in R2, UDR ; Port reading                         R2 <= UDR
out $16, R7  ; Write to the port                   $16(PINB) <= R7
ldi R25, $DE       Load constant                  R25 = $DE
out $3D, R25       ; Write to the port                  $3D(SP) <= R25
```

push R4 ; Adding a register to the stack STACK <= R4;
SP = SP-1
pop R9 ; Popping a register from the stack SP = SP+1,
R9 <= STACK

3.4.1. Commands of logical operations

These commands allow you to perform standard logical operations on bytes, such as "logical multiplication" (AND), "logical addition" (OR), “the exclusive O” operation, and the calculation of the inverse and additional number codes. This group also includes the clear/set register commands and the tetrad rearrangement command. All operations are performed on general-purpose registers and the result is stored in one of the RGP. All logical operations are performed in one machine cycle.

Arithmetic operation commands and shift commands.

This group includes commands that perform such basic operations as addition, subtraction, shift (right and left), increment and decrement. All operations are performed only on general-purpose registers. At the same time, AVR microcontrolles make it easy to operate both signed and unsigned numbers, as well as work with numbers provided in additional code.

All commands of the considered group are executed in one machine cycle, with the exception of commands operating on two-byte values, which are executed in two cycles.

Table 3.4. Arithmetic and logical commands

№	Mnemonic notation	Operands	Description	Operation	Flags	*Type*	Number of cycles
1	ADD	Rd, Rr	Addition without carry	Rd □□Rd + Rr	Z, C,N, V,H,S	3	1
2	ADC	Rd, Rr	Addition with carry	Rd □□Rd + Rr + C	Z, ,N,	3	1

					V,H,S		
3	ADIW	Rdl, K6	Adding two bytes with a constant	Rdh:Rdl ← Rdh:Rdl+K6	Z,C,N,V,S	4	2
4	SUB	Rd, Rr	Subtraction without carry	Rd □□Rd – Rr	Z,C,N,V,H,S	3	1
5	SUBI	Rd*, K	Subtraction of a constant	Rd* □□Rd* – K	Z,C,N,V,H,S	5	1
6	SBC	Rd, Rr	Subtraction with carry	Rd □□Rd – Rr – C	Z,C,N,V,H,S	3	1
7	SBCI	Rd*, K	Subtraction of a constant with carry	Rd* □□Rd* – K – C	Z,C,N,V,H,S	5	1
8	SBIW	Rdl, K6	Subtraction from a two-byte constant	Rdh:Rdl □□Rdh:Rdl – K6	Z,C,N,V,S	4	2
9	AND	Rd, Rr	Logical AND	Rd □□Rd □□Rr	Z,N,V,S	3	1
10	ANDI	Rd*, K	Logical AND with a constant	Rd* □□Rd* □□K	Z,N,V,S	5	1
11	OR	Rd, Rr	Logical OR	Rd □□Rd v Rr	Z,N,V,S	3	1
12	ORI	Rd*, K	Logical OR with a constant	Rd* □□Rd* v K	Z,N,V,S	5	1
13	EOR	Rd, Rr	Boolean exclusive OR	Rd □□Rd □□Rr	Z,N,V,S	3	1
14	COM	Rd	Bitwise inversion	Rd □□$FF □□Rd	Z,C,N,V,S	2	1
15	NEG	Rd	Change of sign (additional code)	Rd □□$00 □□Rd	Z,C,N,V,H,S	2	1
16	INC	Rd	The increment of the register value	Rd □□Rd + 1	Z,N,V,S	2	1
17	DEC	Rd	Decrement of the register value	Rd □□Rd □□1	Z,N,V,S	2	1
18	ASR	Rd	Arithmetic shift to the right	Rd(n) □□Rd(n+1), n=0..6, C□Rd(0), Rd(7)□Rd(7)	Z,C,N,V	2	1
19	LSL	Rd	Logical left shift	Rd(n+1) □□Rd(n), Rd(0) □□0, C□Rd(7)	Z,C,N,V	2	1
20	LSR	Rd	Logical shift to the right	Rd(n) □□Rd(n+1), Rd(7) □□0, C□Rd(0)	Z,C,N,V,S	2	1
21	ROL	Rd	Cyclic shift to the left through C	Rd(0)□C,Rd(n+1)□□Rd(n), C□Rd(7)	Z,C,N,V,H,S	2	1
22	ROR	Rd	Cyclic shift to the right through C	Rd(7)□C,Rd(n)□□Rd(n+1), C□Rd(0)	Z,C,N,V,S	2	1
23	TST	Rd	Test for a null or negative value	Rd □□Rd □□Rd	Z,N,V,S	2	1
24	CLR	Rd	Registry clearing	Rd □□Rd □□Rd	Z,N,V,S	2	1
25	SWAP	Rd	Exchange of tetrads	Rd(3..0)□Rd(7..4),Rd(7..4)□Rd(3..0)	-	2	1
26	SER	Rd	Register setting	Rd □□$FF	-	2	1

3.4.2. Setting and resetting commands

This group includes commands that set or reset a given RFP or RI/O bit. Moreover, there are also additional commands for changing the bits of the status register SREG, because checking the status of the bits of this particular register is performed most often. Conditionally, this group can also include two control transfer commands of the "check/skip" type, which skip the next command depending on the state of the RGP or RI/O bit.

It is worth remembering that in the CBR and SBR commands, the operand is a bit mask, not a bit number. To obtain a bit mask from a bit number you should use the assembler operator "shift to the left" (<<), as shown in the following example:

```
sbr r16, (1<<SE) + (1<<SM)   ;  Set the SE and SM flags
out MCUCR, r16   ;  register MCUCR.
```

It is worth paying attention that the priority of the "+" operator is higher than the "<<" operator, so in order to get the correct result, the terms are enclosed in parentheses.

All commands in this group are executed in one machine cycle, except when the command is skipped as a result of the check. In this case, the command is executed in two or three machine cycles depending on the length of the skipped command.

The group of microcontroller control commands includes only 3 commands:

• NOP – empty operation;

• SLEEP – putting the microcontroller into reduced power consumption mode;

• WDR – reset watchdog timer.

The NOP and WDR commands are executed in one machine cycle, and the SLEEP command is executed three machine cycles.

Table 3.5. Bit commands

№	Mnemonic notation	Operands	Description	Operation	Flags	*Type*	Number of cycles
1	SBR	Rd*, K	Set the bit(s) in the register	Rd* □□Rd* v K	Z,N,V,S	5	1
2	CBR	Rd*, K	Reset the bit(s) in the register	Rd* □□Rd* □□($FF - K)	Z,N,V,S	5	1
3	SBI	P*,b	Set the bit in the RI/O	I/O(P*,b) □□1	-	9	2
4	CBI	P*,b	Reset the bit in the RI/O	I/O(P*,b) □□0	-	9	2
5	BSET	s	Set the indicated bit of the SREG register	SREG(s) □□1	SREG(s)	10	1
6	BCLR	s	Reset the specified bit of the SREG register	SREG(s) □□0	SREG(s)	10	1
7	BLD	Rd, b	Load the bit from T into the register	Rd(b) □□T	-	7	1
8	BST	Rr, b	Save the bit from the register to T	T □□Rr(b)	T	7	1
9	SEC		Check the transfer flag	C □□1	C	1	1
10	CLC		Reset the transfer flag	C □□0	C	1	1
11	SEN		Check the negative number check box	N □□1	N	1	1
12	CLN		Uncheck the negative number	N □□0	N	1	1
13	SEZ		Set the check box to zero	Z □□1	Z	1	1
14	CLZ		Reset the flag to zero	Z□□0	Z	1	1
15	SEI		Set the interrupts flag	I□1	I	1	1
16	CLI		Reset the interrupt flag	I □□0	I	1	1
17	SES		Check the sign	S□□1	S	1	1
18	CLS		Uncheck the sign	S □□0	S	1	1
19	SEV		Check the overflow box	V□□1	V	1	1
20	CLV		Reset the overflow flag	V □□0	V	1	1
21	SET		Check the T box	T □□1	T	1	1
22	CLT		Uncheck T	T □□0	T	1	1
23	SEH		Check the half transfer flag	H □□1	H	1	1
24	CLH		Clear the half transfer flag	H□□0	H	1	1
25	NOP		Empty operation	-	-	1	1

26	SLEEP		Switching to sleep mode	-	-	1	3
27	WDR		Reset watchdog timer	-	-	1	1

All involved RI/O bits have their own symbolic names. The bit names of the input/output registers are given in table 3.6.

Table 3.6. Symbolic names of I/O registers

Адреса	Назва	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$3F($5F)	SREG	I	T	H	S	V	N	Z	C
$3E($5E)	Reserved	—	—	—	—	—	—	—	—
$3D ($5D)	SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
$3C (SBC)	Reserved								
$3B($5B)	GIMSK	INT1	INTO	—	—	—	—	—	—
$3A ($5A)	GIFR	INTF1	INTFO						
$39 ($59)	TJMSK	TOIE1	OCIE1	—	—	TICIE1	—	TOIE0	—
$38 ($58)	TIFR	TOV1	OCF1	—	—	ICF1	—	TOV0	—
$37 ($57)	Reserved								
$36 ($56)	Reserved								
$35 *($55)*	MCUCR	—		SE	SM	ISC11	ISC10	ISC01	ISC00
$34 ($54)	MCUSR	—	—	—	—	WDRF	BORF	EXTRF	PORF
$33 ($53)	TCCRO	—	—	—	—	—	CS02	CS01	CS00
$32 ($52)	TCNTO	Timer/Counler0 (8 Bits)							
$31 ($51)	Reserved								
$30 ($50)	Reserved								
$2F ($4F)	TCCR1A	COM 11	COM 10	—	—	—	—	PWM11	PWM10
$2E ($4E)	TCCR1B	ICNC1	ICES1	—	—	CTC1	CS12	CS11	CS10
$2D ($4D)	TCNT1H	Timer/Counter -Counter Register High Byte							
$2C ($4C)	TCNT1L	Timer/Counter -Counter Register Low Byte							
$2B ($4B)	0CR1H	Timer/Counter - Output Compare Register High Byte							
$2A ($4A)	OCR1L	Timer/Counter - Output Compare Register Low Byte							
$29($49)	Reserved								
$28 ($48)	Reserved								
$27 ($47)	ICR1H	Timer/Counter - Input Capture Register High Byte							
$26 ($46)	ICR1L	Timer/Counter - Input Capture Register Low Byte							
$25 ($45)	Reserved								
$24 ($44)	Reserved								

Table 3.6. Symbolic names of I/O registers

$23 ($43)	Reserved								
$22 ($42)	Reserved								
S21 ($41)	WDTCR	—	—	—	WDTOE	WDE	WDP2	WDP1	WDP0
$20 ($40)	Reserved								
$1F(S3F)	Reserved								
$1E($3E)	EEAR	EEPROM Address Register							
$1D($3D)	EEDR	EEPROM Data Register							
$1C ($3C)	EECR	—	—	—	—	EERIE	EEMWE	EEWE	EERE
$1B($3B)	Reserved								
$1A($3A)	Reserved								
$19($39)	Reserved								
$18($38)	PORTB	—	—	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
$17 ($37)	DDRB	—	—	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
$16($36)	PINB	—	—	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
$15 ($35)	PORTC	—	—	PORTC 5	PORTC4	PORTC 3	PORTC2	PORTC 1	PORTC0
$14 ($34)	DDRC	—	—	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
$13 ($33)	PINC	—	—	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD 1	PORTD0
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
$10($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
$0F (S2F)	SPDR	SPI Data Register							
$0E($2E)	SPSR	SPIF	WCOL	—	—	—	—	—	—
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
$0C ($2C)	UDR	UART I/O Data Register							
$0B($2B)	UCSRA	RXC	TXC	UDRE	FE	OR	—	—	—
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8
$09 ($29)	UBRR	UART Baud Rate Register							
$08($28)	ACSR	ACD	AINBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
$07 ($27)	ADMUX	—	ADCBG	—	—	—	MUX2	MUX1	MUX0
$06 ($26)	ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
$05 ($25)	ADCH	—	—	—	—	—	—	ADC9	ADC8
$04 ($24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
$03 ($23)	UBRRHI					UART Baud Rate Register High			
$02 ($22)	Reserved								
$01 ($21)	Reserved								

Table 3.6. Symbolic names of I/O registers

$00 ($20)	Reserved	

3.4.3. Commands of transition, call of routines and return

This group includes commands of transition, call routines and return from them and commands of the "check/skip" type, which skip the next command when some condition is met. This group also includes the comparison commands that form the flags of the SREG register, which are mainly intended to work together with the conditional transition commands.

The instruction system of microcontrollers of the classic family includes commands for both unconditional and conditional transitions. The indirect (IJMP) and relative (RJMP) unconditional transition commands are the simplest in this group. Their function is only to record a new address in the command counter. Conditional transition commands also change the contents of the command counter, but this change occurs only when some condition is met, or more precisely, when the various flags of the SREG register are set.

All conditional transition commands can be divided into two subgroups. The first subgroup is general-purpose conditional transition commands. This subgroup includes the two commands BRBS s, k and BRBC s, k, which explicitly specify the flag number of the SREG register being checked. Accordingly, the transition is carried out when SREG.s = 0 (BRBC) or SREG.s = 1 (BRBS). Another subgroup consists of 18 specialized commands, each of which performs a transition under some specific condition ("equal to", "greater than or equal to", "there was a transfer", etc.). Moreover, some commands are used after comparing unsigned numbers, others - after comparing signed numbers. Possible conditions that are checked, as well as their corresponding conditional transition commands are listed in table 3.7.

The commands listed in table 3.7 are only equivalent mnemonic designations of the commands BRBS s, k and BRBC s, k with the specified values of the operand s. The BREQ k command has, for example, the same operation code as the BRBS

1, k command and the BRGE k command has the same operation code as the BRBC 4, k command.

Subroutine call commands (ICALL, RCALL) work almost the same as unconditional jump commands. The difference is that before performing the transition, the value of the command counter is stored on the stack. In addition, the subroutine must end with the return command RET.

In the above example the RET command replaces the address contained in the command counter with the address of the command following the RCALL command.

Table 3.7. Summary table of conditional transition commands

Check	Logic condition	Command	Reverse check	Logic condition	Command	Data type
Rd>Rr	(N⊕V)=0	BRLT*	Rd≤Rr	(N⊕V)=1	BRGE*	3i
Rd≥Rr	(N⊕V)=0	BRGE	Rd<Rr	(N⊕V)=1	BRLT	3i
Rd=Rr	Z=1	BREQ	Rd≠Rr	Z=0	BRNE	3i
Rd≤Rr	(N⊕V)=1	BRGE*	Rd>Rr	(N⊕V)=0	BRLT*	3i
Rd<Rr	(N⊕V)=1	BRLT	Rd≥Rr	(N⊕V)=0	BRGE	3i
Rd>Rr	C=0	BRLO*	Rd≤Rr	C=1	BRSH*	Беззнака
Rd≥Rr	C=0	BRSH/BRCC	Rd<Rr	C=1	BRLO/BRCS	Беззнака
Rd=Rr	Z=1	BREQ	Rd≠Rr	Z=0	BRNE	Беззнака
Rd≤Rr	C=1	BRSH*	Rd>Rr	C=0	BRLO*	Беззнака
Rd<Rr	C=1	BRLO/BRCS	Rd≥Rr	C=0	BRSH/BRCC	Беззнака
«Carry»	C=1	BRCS	«No trasfer»	C=0	BRCC	—
«Less than zero”	N=1	BRMI	«More than 0)	N=0	BRPL	—
«Overflow»	V=1	BRVS	«No overflow»	V=0	BRVC	—
«Zero»	Z=1	BREQ	«Not zero»	Z=0	BRNE	—
«Half “	H=1	BRHS	“Not half”	H=0	BRHC	—

Since the microcontroller does not have commands for transitions based on the conditions: greater than and less than, or equal to, then in order to transition under these conditions the operands of the previous comparison command must be written in the reverse order, i.e. instead of CP Rd, Rr -> CP Rr, Rd.

It is worth noting that the control transfer commands disrupt the normal (linear) execution of the main program. Whenever a command from this group is executed (except for comparison commands) the normal functioning of the pipeline is disrupted. Before loading a new address into the pipeline, it is stopped and the sequence of executed commands is cleared. Accordingly, reinitialization of the

pipeline leads to the fact that such commands are executed during several machine cycles

Table 3.8. Control transfer commands

№	Mnemonic notation	Operand	Description	Operation	Flags	*Type*	Number of cycles
1	RJMP	k	Relative unconditional transition	PC □□PC + k + 1	-	13	2
2	IJMP		Indirect unconditional transition	PC □□Z	-	1	2
3	RCALL	k	A relative unconditional subroutine call	STACK □PC+1, PC □□PC + k + 1, SP□SP-2	-	13	3
4	ICALL		Indirect unconditional subroutine call	STACK □PC+1, PC □□Z, SP□SP-2	-	1	3
5	RET		Return from subroutine	PC □□STACK,SP□SP+2	-	1	4
6	RETI		Return from an interrupt subroutine	PC □□STACK, SP□SP+2, I□1	I	1	4
7	CPSE	Rd,Rr	Compare, skip if equal	if (Rd = Rr)PC □□PC + 2 or 3	-	3	1 / 2 / 3
8	CP	Rd, Rr	Compare	Rd - Rr	Z,N,V,C,H,S	3	1
9	CPC	Rd, Rr	Compare with transfer	Rd - Rr - C	Z,N,V,C,H,S	3	1
10	CPI	Rd*, K	Compare with a constant	Rd* - K	Z,N,V,C,H,S	5	1
11	SBRC	Rr, b	Skip if the bit in the register is reset	if (Rr(b)=0)PC □□PC + 2 or 3	-	7	1 / 2 / 3
12	SBRS	Rr, b	Skip if the bit in the register is set	if (Rr(b)=1)PC □□PC + 2 or 3	-	7	1 / 2 / 3
13	SBIC	P*, b	Skip if bit on port is reset	if (P*(b)=0) PC □□PC + 2 or 3	-	9	1 / 2 / 3
14	SBIS	P*, b	Skip if port bit is set	if (P*(b)=1) PC □□PC + 2 or 3	-	9	1 / 2 / 3
15	BRBS	s, k	Jump if the flag in SREG is checked	if (SREG(s) = 1) then C□PC+k+1	-	11	1 / 2
16	BRBC	s, k	Jump if flag in SREG is cleared	if(SREG(s) = 0) then PC□PC+k+1	-	11	1 / 2
17	BREQ	k	Jump if equal	if (Z = 1) then PC □□PC + k + 1	-	12	1 / 2
18	BRCS	k	Jump if the carry flag is checked	if (C = 1) then PC □□PC + k + 1	-	12	1 / 2
19	BRNE	k	Jump if not equal	if (Z = 0) then PC □□PC + k + 1	-	12	1 / 2
20	BRCC	k	Jump if the carry flag is cleared	if (C = 0) then PC □□PC + k + 1	-	12	1 / 2

21	BRSH	k	Jump if equal or greater	if (C = 0) then PC □□PC + k + 1	-	12	1 / 2
22	BRLO	k	Jump if less	if (C = 1) then PC □□PC + k + 1	-	12	1 / 2
23	BRMI	k	Jump if minus	if (N = 1) then PC □□PC + k + 1	-	12	1 / 2
24	BRPL	k	Jump if plus	if (N = 0) then PC □□PC + k + 1	-	12	1 / 2
25	BRGE	k	Jump if greater than or equal to (signed)	if (N □□V= 0) then PC □□□PC +k+1	-	12	1 / 2
26	BRLT	k	Jump if less (with sign)	if (N □□V= 1) then PC□□PC+k+1	-	12	1 / 2
27	BRHS	k	Jump if the half carry flag is checked	if (H = 1) then PC □□PC + k + 1	-	12	1 / 2
28	BRHC	k	Jump if the half carry flag is cleared	if (H = 0) then PC □□PC + k + 1	-	12	1 / 2
29	BRTS	k	Jump if the T flag is checked	if (T = 1) then PC □□PC + k + 1	-	12	1 / 2
30	BRTC	k	Jump if the T flag is cleared	if (T = 0) then PC □□PC + k + 1	-	12	1 / 2
31	BRVS	k	Jump if the overflow flag is checked	if (V = 1) then PC □□PC + k + 1	-	12	1 / 2
32	BRVC	k	Jump f the overflow flag is cleared	if (V = 0) then PC □□PC + k + 1	-	12	1 / 2
33	BRIE	k	Jump if interrupts are enabled	if (I = 1) then PC □□PC + k + 1	-	12	1 / 2
34	BRID	k	Jump if interrupts are disabled	if (I = 0) then PC □□PC + k + 1	-	12	1 / 2

3.4.4. Elements of the microprocessor system interface

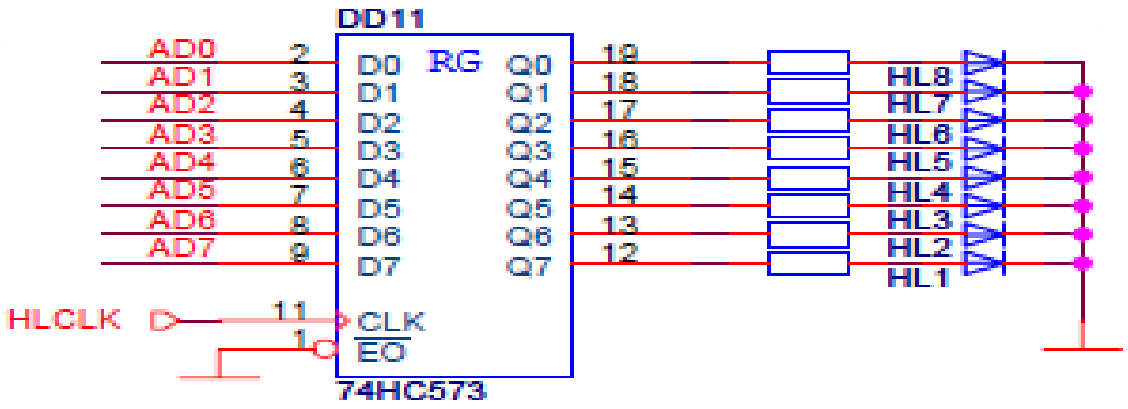


Figure 3.4. diagram of 8-bit LED indicator

The LEDs are connected via a connecting FPGA and occupy only one address in the programming model - 0xA006. The byte located at this address, when read will output the state of each individual LED (on/off) by the corresponding bit in the read byte, and when written to this address the LEDs corresponding to one will turn on (all others will turn off).

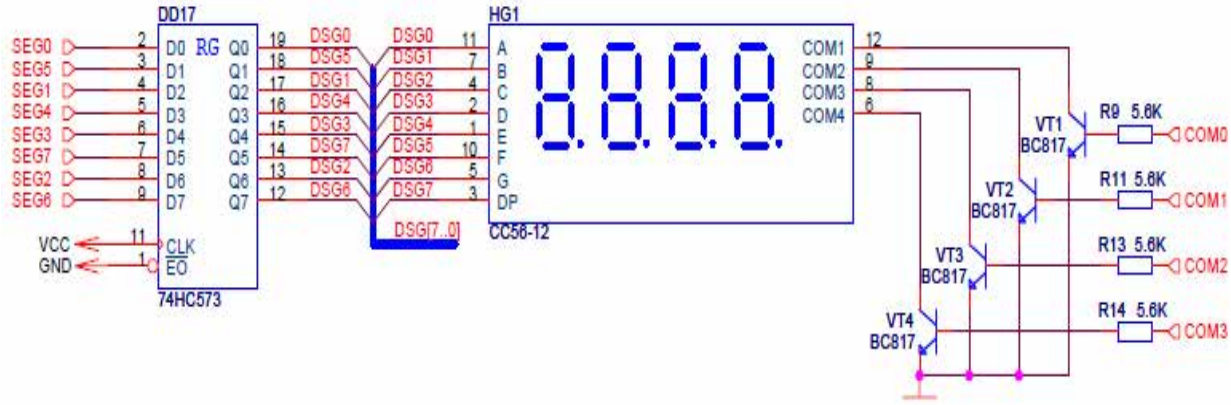


Figure 3.5. Circuit of the seven-segment indicator (static indication)

The static indicator is convenient to use as a simple output of a number due to its "static character" - the information on the scoreboard does not need to be maintained. In fact, it is static only in the programming model, in practice this static character is achieved due to the transparent wrapping of the dynamic indication due to the resources of the connecting FPGA.

What does static indication mean? It means that outputting a number to the indicator is a simple operation, like outputting to a port or writing to RAM. A single-bit (single-digit) indicator is always static because each of its 7 (8 if you count the decimal point) LEDs can be turned on and held in the desired state by simple commands. However, such an implementation for multi-digit indicators becomes too complicated - each digit requires 8 input-output pins in the board itself. So, a 4-digit display must have 32 pins to provide a static indication.

What is a dynamic indication? This is a way to reduce the number of I/O pins for multi-digit 7-SEG indicators. It consists in the alternate output of each digit (that is, the first digit is output, then the second, third, etc.) with a fairly high

frequency. Due to the inertia of the eye, these switching data become imperceptible. And by reducing the amount of data displayed on the scoreboard at the same time the problem of a large number of pins is solved.

The static indicator is divided into two pairs for a programmer. Each pair can represent one byte in hexadecimal notation. The value of a pair of digits is changed by writing the corresponding byte to the addresses 0xA000 or 0xA001. The state of each digit (on\off) is controlled by the corresponding low-order 4 bits of the DC_REG byte (0-on, 1-off) and high-order bits control activation of the decimal points (1-on, 0-off).

Table 3.9. Address of seven-segment indicators

Axx0	Record	[Indicator register 0] (left)	DISPLAY[0]
Axx1	Record	[Indicator register 1] (right)	DISPLAY[1]
Axx4	Record	Powering numbers \ decimal points	DC_REG

All the tasks of multiplexing commands and data are taken over by the FPGA, the control of the display is quite simple. After each writing of commands or data, the necessary delay (~100 ns) must be maintained.

Since by default the LCD does not display anything when it is turned on, it is necessary to initialize it. For this, for example, you can transfer the command code to the command register, which sets the 2-line display mode with a 5x8 dot matrix and an 8-bit data bus (0x38).

Figure 3.10. Character codes for this LCD display.

For example, to display the symbol "!", it is necessary to transfer the code of this symbol, i.e. 0x21 to the RKI data register.

Table 3.11. Control bit combinations for command register

D7	D6	D5	D4	D3	D2	D1	D0	Assignment
0	0	0	0	0	0	0	1	Clearing the screen, AC=0 (AC on DDRAM
0	0	0	0	0	0	1	-	AC=0, reset offsets, start of line from start of DDRAM (AC on DDRAM)
0	0	0	0	0	1	I/D	S	The direction of the cursor or screen shift is selected
0	0	0	0	1	D	C	B	The picture mode is selected
0	0	0	1	S/C	R/L	-	-	Cursor/screen shift command
0	0	1	DL	N	F	-	-	Sweep Options and Data Bus
0	1	AG	AG	AG	AG	AG	AG	Assigning an address to the AC counter in the CGRAM area
1	AD	AD	AD	AD	AD	AD	AD	Assigning an address to the AC counter in the DDRAM area

Table 3.12. Flags that control the work of RKI

I/D Displacement mode of the AC address counter, 0-decrease, 1-increase.

S Screen contents shift mode flag, 0-screen shift does not occur, 1-after writing to DDRAM the next code, the screen shifts in the direction determined by the I/D flag: 0-right, 1-left. When shifting, the contens of DDRAM does not change. Only the internal pointers to find the visible beginning of the line in DDRAM are changed.

S/C Flag-command, which together with the R/L flag performs the operation of shifting the screen (without changes in DDRAM) or the cursor. At 0- the cursor moves, at 1- the screen moves.

R/L Flag command that performs a screen or cursor shift operation along with S/C. 0 – left, 1 – right.

D/L Flag that defines the width of the data bus: 0 - 4 bits, 1 - 8 bits.

N LCD image scanning mode: 0 - 1 line, 1 - 2 lines.

F Character matrix size: 0 – 5x8 points, 1 – 5x10 points.

D Image presence: 0 - off, 1 - on.

C Cursor in strike through mode: 0 – disabled, 1 – enabled.

B Cursor in the form of a flashing place sign: 0 - disabled, 1 - enabled.

Table 3.13. The value of the control flags after deletion

I/D 1 Mode to increase the counter by 1

S 0 without image shift

D/L 1 8-bit bus

N 0 Single line sweep mode

F 0 Matrix of 5x8 points

D 0 Picture is off

C 0 The underline cursor is disabled

B 0 The place holder cursor is disabled

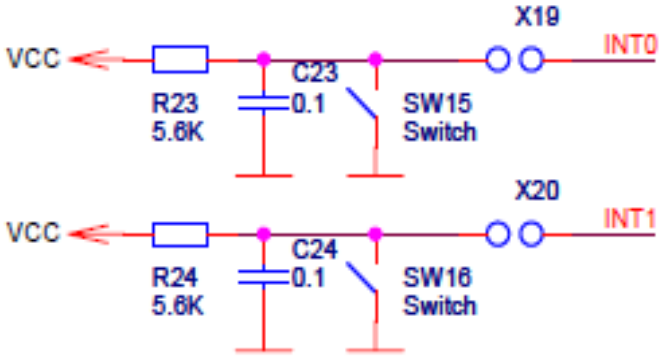


Figure 3.6. Switching circuit of discrete input buttons

A user interrupt (INT0 or INT1) is generated when the button is pressed (its state changes). The circuit shows that the supply voltage, i.e. unit is applied to the INTx input when the key is open. When the key is closed, a low level (ground) is set which provokes an interruption. The capacitor act as a high-pass filter and compensates for contact vibrations. It is also possible to analyze the state of individual bits in port D and thus infer the state of the discrete input buttons.

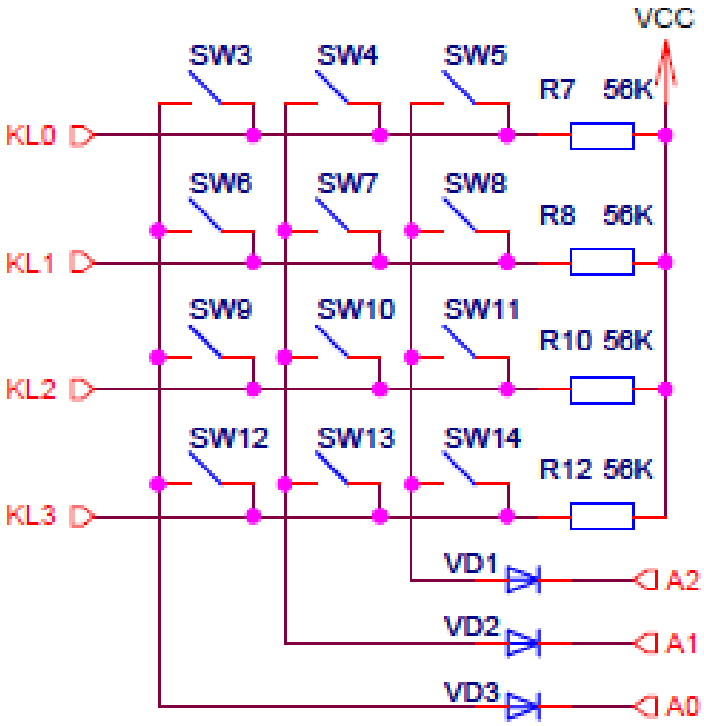


Figure 3.7. Scheme of the digital keyboard

EV8031/AU expansion board

The EV8031/AU expansion board is designed to explore control and regulation tasks that are often encountered in industrial automation systems. For

this, the board has a speaker, a DC motor and an analog temperature sensor. An ADC is used to convert analog signals. To enter information you can use discrete buttons and an encoder (this device is similar to the tuning knob in old radios).

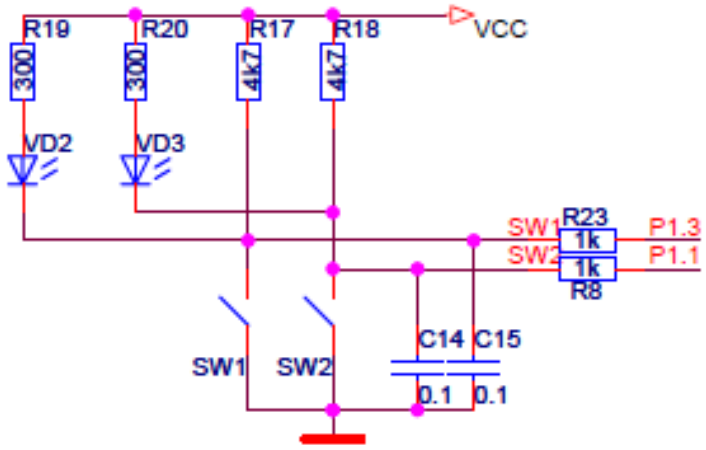


Figure 3.8. Connection diagram of discrete buttons

Ordinary buttons are designed for discrete input to the microcontroller. Connected directly to the pins of the microcontroller they generate a low level when pressed. The capacitors filter contact vibrations.

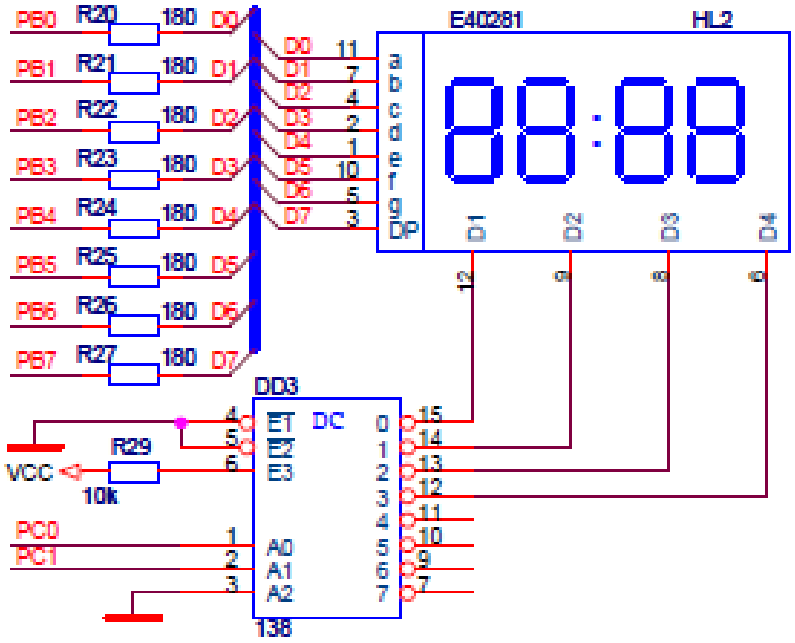


Figure 3.9. Seven-segment indicator connection diagram (dynamic indication)

Control of the dynamic indicator is carried out through memory at address 0xB000. The bits of this byte turn on the corresponding LEDs on the indicator digit.

The concrete digit is selected using the lines PC0, PC1 of port C. The indices of the four digits of the indicator are coded using two lines.

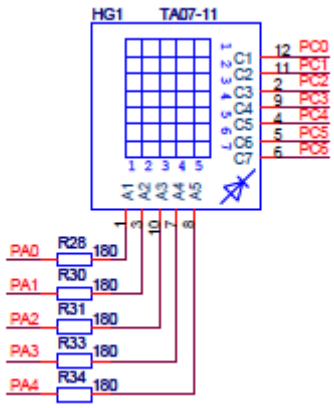


Figure 3.10. Scheme of operation of the sign-synthesizing indicator 5x7

The output to the sign-synthesizing indicator (matrix of LEDs) is carried out using ports A and C. A logical zero on any of the lines selects the specified row or column.

3.4.5. Executive devices of the microprocessor system

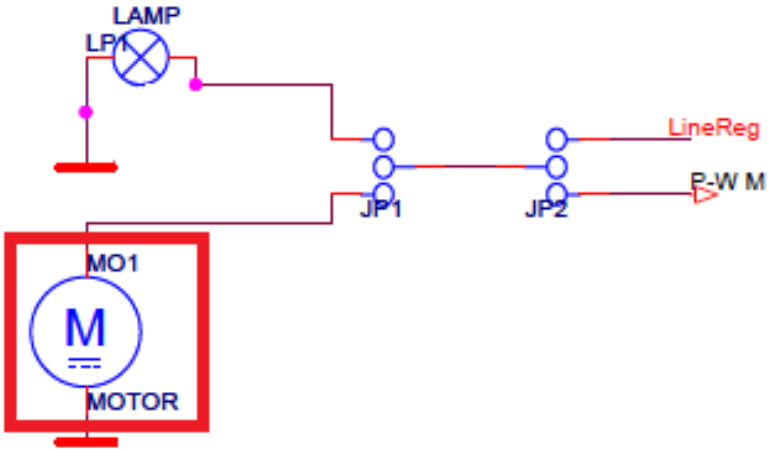


Figure 3.11. fan (motor) connection diagram

The motor presented on the board can be controlled by voltage in two modes.

• LineReg – the voltage across the winding is the DAC output;

• P-WM – pulse width modulation, control from the microcontroller pin P1.0.

The mode is selected by setting jumper JP2 to the appropriate position.

Let's consider the second method: "pulse width modulation"

The T0 eight-bit timer/counter is present in all Mega family microcontroller models, and the T2 timer/counter is present in all, except the ATmega8515x. The block diagram of an 8-bit timer/counter for the ATmega8515x microcontroller is shown in Figure 3.12. The timer/counter consists of 3 input/output registers: the counter register TCNT0, the control register TCCR0 and the comparison register OCR0.

The T0 timer/counter can generate an interrupt when the counter overflows, as well as when the counter register and the comparison register are equal. The flags of both interrupts are in the TIFR register, and enabling/disabling these interrupts is done by setting/resetting the corresponding flags of the TIMSK register.

The TCCR0 register is used to control the timer/counter module. The format of the TCCR0 register is shown in figure 3.13, and the description of its bits is in table 3.14.

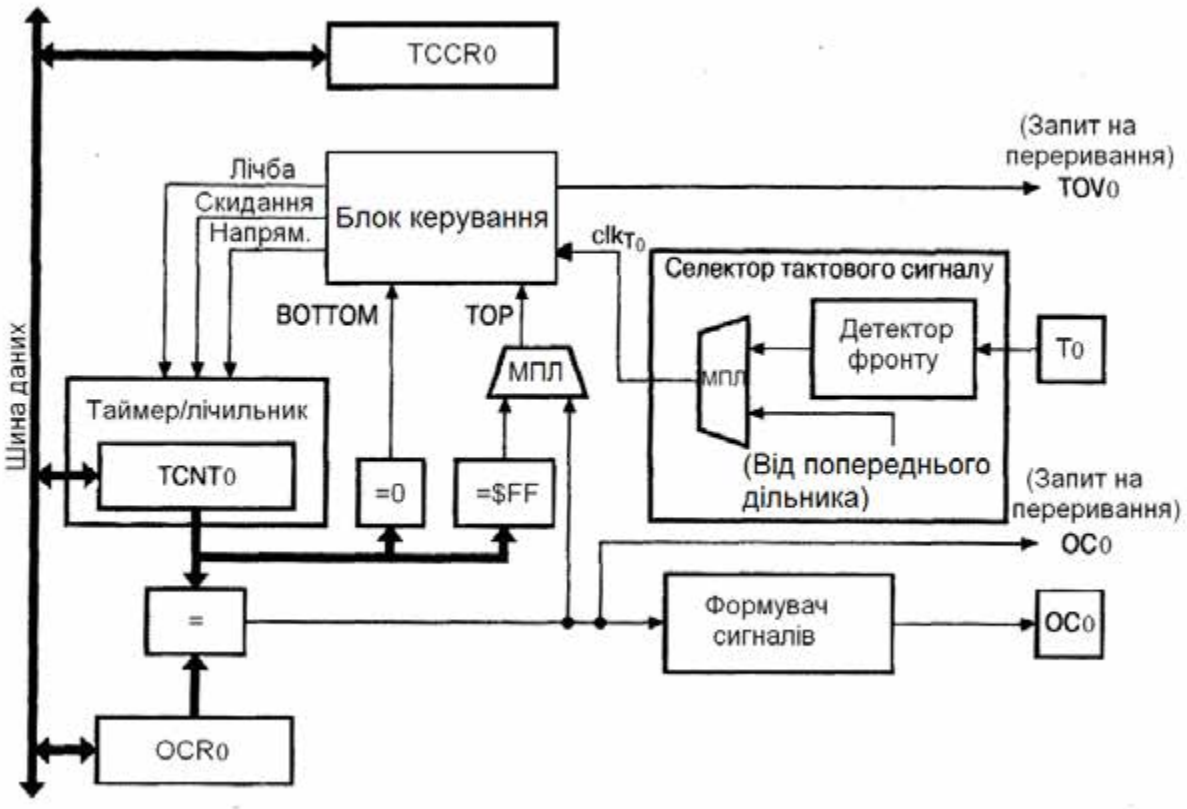


Figure 3.12. Structural diagram of an 8-bit timer/counter

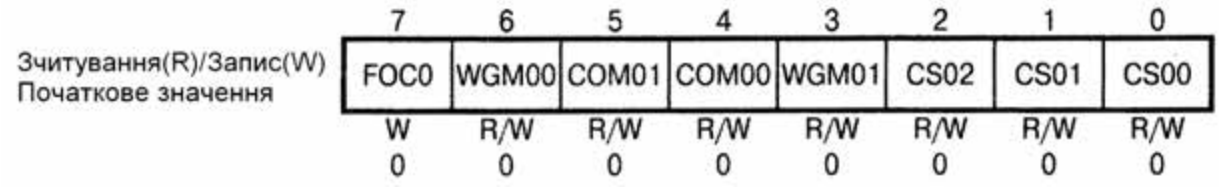


Figure 3.13. TCCR0 register format in ATmega8515 microcontroller

Table 3.14. Bits of register TCCR0

Bit	Name	Description			
7	FOC0	Forced change of the OS0 output state (Normal and STS modes).			
6,3	WGM01: WGM00	Timer/counter mode. These bits define the timer/counter mode as follows:			
		Mode number	WGM01	WGM00	Timer/counter mode T0
		0	0	0	Normal
		1	0	1	Phase correct PWM
		2	1	0	CTC (скидання за збігом)
		3	1	1	Fast PWM
5,4	COM01: COM00	Mode of operation of the comparison unit. These bits determine the behavior of the OS0 output when the "Match" event occurs. The effect of the contents of these bits on the output state depends on the mode of operation of the timer/counter			
2...0	CS02, CS01, CS00	Clock signal control. These bits define the microcontroller's clock source.			

The selection of the clock signal source of the timer/counter T0 can be made using table 3.15.

Table 3.15. Selection of the clock signal source of the timer/counter T0

CS02	CS01	CS00	Clock source.
0	0	0	Timer/counter is stopped
0	0	1	$f_{clkI/0}$
0	1	0	$f_{clkI/0}/8$
0	1	1	$f_{clkI/0}/64$
1	0	0	$f_{clkI/0}/256$

1	0	1	$f_{clkI/0}$/1024
1	1	0	Output T0, the digit is carried out according to the falling edge of the pulses
1	1	1	Output T0, the digit is carried out by the rising edge of the pulses

The "Phase Correct PWM" mode, like the Fast PWM mode is intended for generating signals with pulse-width modulation. However, the counter register functions as a reverse counter in this mode, the state of which changes with each pulse of the clock signal fclkT0 (fclkT2). The state of the counter first changes from $00 to $FF and then back to $00. When the counter reaches the maximum (TOP) value, the counting direction changes, but the counter remains in this state during one period of the fclkT0 (fclkT2) signal. When the counter reaches the minimum value ($00), the counting direction also changes and the TOV0 interrupt flag of the TIFR register is simultaneously set. If the contents of the counter register and the comparison register OCR0 are equal, the OCF0 flag and the TIFR register are set and the state of the OS0 output changes.

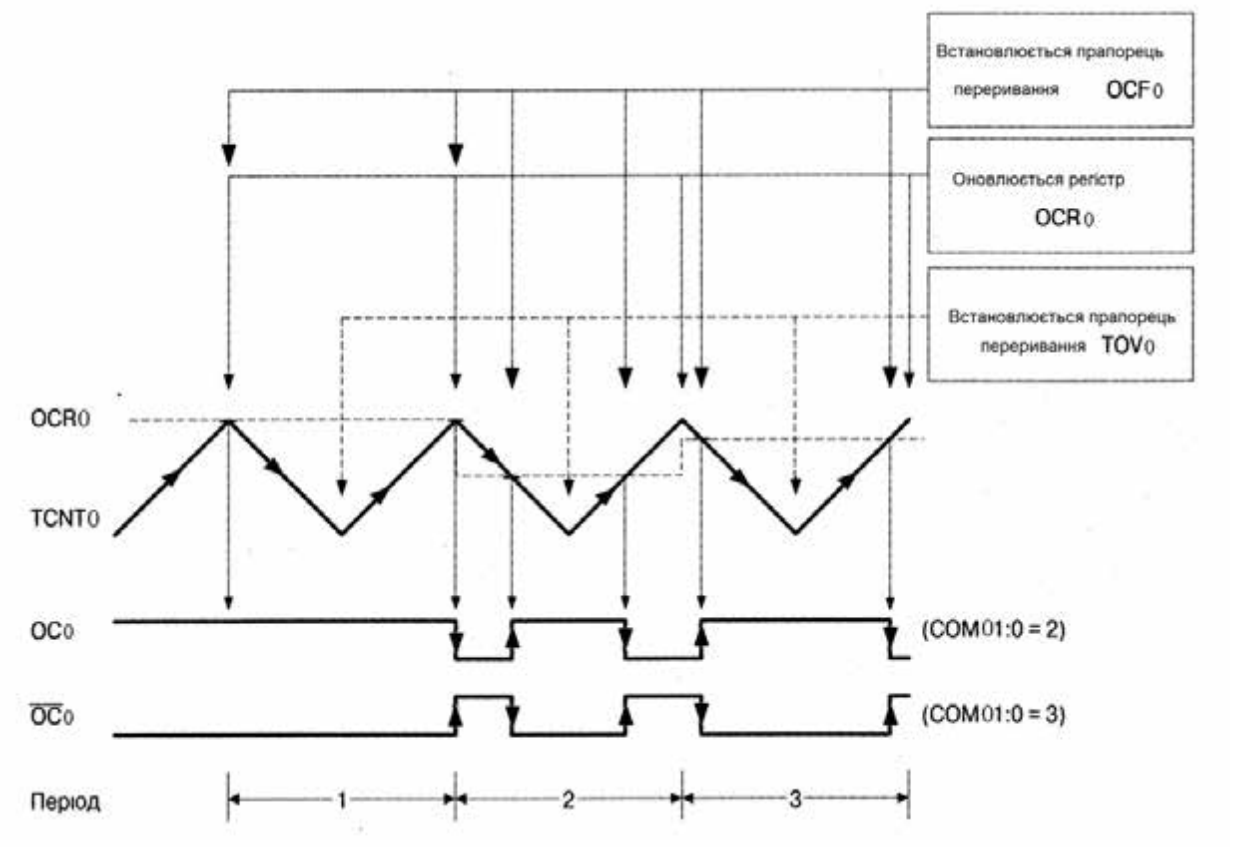


Figure 3.12. Forming a PWM signal in Phase Correct PWM mode

Table 3.16. The behavior of the OS0 output, when"Match" event occurs in the Phase Correct PWM mode

COM01	COM00	Behavior of output OS0
0	0	Timer/Counter T0 disconnected from OS0 pin
0	1	WGM02 = 0 – Timer/Counter T0 dsconnected from OS0 pin; WGM02 = 1 – the output state is reversed if TCNT0 = OCR0
1	0	Resets to "0" on forward count and set to "1" on inverse count (non-inverted PWM signal)
1	1	Set to "1" on forward count and reset to "0" on inverse count (inverted PWM signal)

An example of calculations

The frequency of the generated signal, the pulse duration and the period of the generated signal are defined by expressions

$f_{OC} = f_0/512*N$,

$t_{імп} = T_0 *2*N*(1 + OCR0)$,

$T_{OC} = T_0*512*N$,

where *N* is the division coefficient of the previous divider, OCR0 is the content of the corresponding comparison register, *f0, T0* are, respectively, the frequency and period of the system clock pulse generator, TOP=255.

3.5. Universal asynchronous (UART) and synchronous/asynchronous (USART) transceiver modules

A significant part of microcontrollers of the AVR family have either universal asynchronous (UART) or universal synchronous/asynchronous

(USART) transceiver modules (USART). Some models contain several of these modules.

ATmega8515x microcontroller ATmega8515x has one UART module.

USART/UART modules implemented in microcontrollers of the family can detect the following atypical situations: overflow; error framing; incorrect start bit.

A simplified structural diagram of one USART/UART module is shown in figure 3.13. Elements of the scheme highlighted in gray in the figure are only part of USART modules.

As shown in the figure, the module consists of three parts: a clock unit, a transmitter unit and a receiver unit. The clock unit of the USART modules includes a synchronization circuit that is used when working in synchronous mode and a transmission speed controller. The clock unit consists only of the baud rate controller in UART modules.

The transmitter block contains a single-level UDR buffer, a shift register, parity bit generation circuit (USART only) and control circuit. The receiver unit, in turn, includes circuits for restoring the clock signal and data, the circuit parity control (USART only), two-level (USART) or one-level (UART), UDR buffer, shift register, and control circuit.

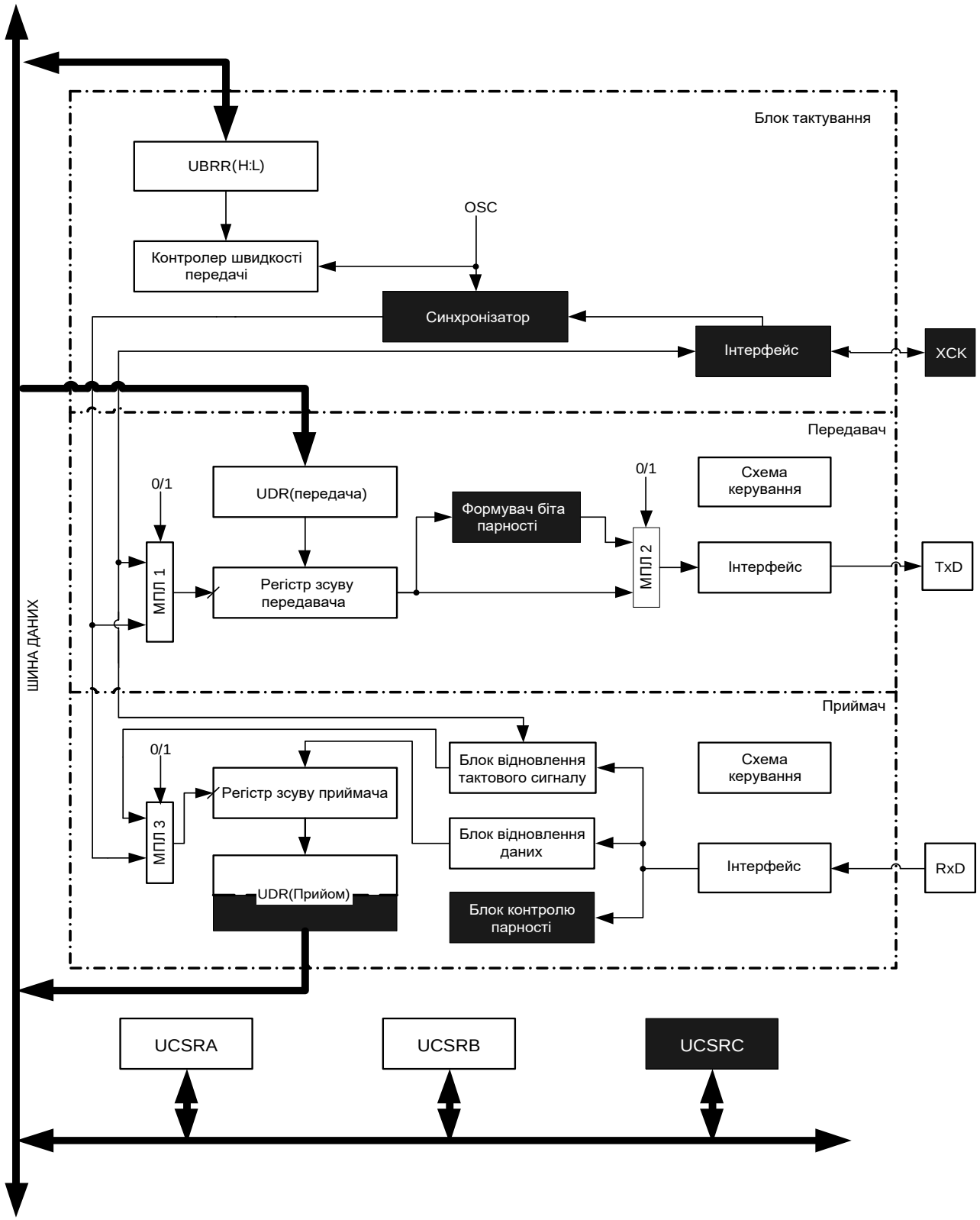


Figure 3.13. Simplified structural diagram of the USART/UART module

.

The receiver and transmitter buffer registers are located at the same address of the input/output register space and are designated as the UDR (Universal Data

Register) data register (UDRn, where n = 0/1). This register stores the lower 8 bits of data being received or transmitted. When reading, the receiver's UDR buffer register is accessed, and when writing, the transmitter's UDR buffer register is accessed.

Two registers are used to control the UART module of the ATmega8515x microcontroller: UCSRA and UCSRB.

	7	6	5	4	3	2	1	0
	RXC*n*	TXC*n*	UDRE*n*	FE*n*	DOR*n*	UPE*n*	U2X*n*	MPCM*n*
Зчитування(R)/Запис(W)	R	R/W	R	R	R	R	R/W	R/W
Початкове значення	0	0	1	0	0	0	0	0

Figure 3.14. UCSRA register format

	7	6	5	4	3	2	1	0
	RXCIE*n*	TXCIE*n*	UDRIEE*n*	RXEN*n*	TXEN*n*	UCSZ*n*2	RXB8*n*	TXB8*n*
Зчитування(R)/Запис(W)	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Початкове значення	0	0	0	0	0	0	1	0

Figure 3.15. UCSRB register format

Table 3.17. Assignment of register bits UCSRA, UCSR0A and UCSR1A of some microcontrollers of the AVR family

Bit	Name	Description
7	RXC	Receive completion check box The flag is set to "1" when a received word is sent from the receiver shift register to the UDR data register and is held to 1 when there is unread data in the receiver buffer (UDR data register). The flag is reset by hardware after clearing the buffer (in the UART after reading the data register). If the RXCIE (RXCIEn) bit of the UCSRB (UCSRnB) register is set, then when the flag is set, a "reception complete" interrupt request is generated

Table 3.17. Assignment of register bits UCSRA, UCSR0A and UCSR1A of some microcontrollers of the AVR family

Bit	Name	Description
6	TXC	Transfer completion flag The flag is set to "1" after all bits of the batch have been transmitted from the transmitter shift register, provided no new value has been loaded into the UDR data register. If the TXCIE (TXCIEn) bit of the UCSRB (UCSRnB) register is set, then a "transfer complete" interrupt is generated when the flag is set. The flag is reset by hardware during the execution of the interrupt processing subroutine or programmatically by writing a log1 to it.
5	UDRE	Clear data register check box This flag is set to "1" when the transmitter buffer is empty (after transferring a byte from the UDR data register to the transmitter shift register). A checked flag means that a new value can be loaded into the data register. If the UCRIE bit of the UCR register (UCSRB) is set, a "data register empty" interrupt request is generated. The flag is reset by hardware when writing to the data register
4	FE	Framing error flag The flag is set to "1" when a framing error is detected, that is, if the first stop bit of the received packet is "0". The flag is reset when a stop bit is received, which is equal to "1"
3	DOR	Overflow flag In UART the flag is set to "1" if new data is placed into the receiver's shift register before the previous word is read from the data register. The flag is reset when the received data is forwarded from the receiver shift register to the buffer
2	UPE	Parity bit check error flag The flag is set to "1" if a parity check error is detected in the data in the receiver buffer. When parity control is disabled, this bit is permanently reset to "0"

Table 3.17. Assignment of register bits UCSRA, UCSR0A and UCSR1A of some microcontrollers of the AVR family

Bit	Name	Description
1	U2X	Double exchange rate If this bit is set to "1", the baud rate controller predivider is reduced from 16 to 8, thereby doubling the speed of the asynchronous exchange over the serial link. In USART the U2X bit (U2Xn) is used only in asynchronous mode of operation. In synchronous mode, it should be reset
0	MPCM	Multiprocessor exchange mode The MRSM bit is used in multiprocessor exchange mode. If it is set to "1", the slave microcontroller waits to receive a frame containing the address. Frames that do not contain a device address are ignored

Table 3.18. UCSRB Register Bits

Bit	Name	Description
7	RXCIE	Permission to interrupt after completion of reception If this bit is set to "1", then when the RXC flag (RXCn) of the UCSRA register (UCSRnA) is set, a "reception complete" interrupt is generated (if the I flag of the SREG register is set to "1")
6	TXCIE	Allow interrupt after transfer complete If this bit is set to "1", then when the TCS flag (TXCn) of the UCSRA register (UCSRnA) is set, a "transfer complete" interrupt is generated (if the I flag of the SREG register is set to "1")
5	UDRIE	Enable interrupt when clearing the UART data register. If this bit is set to "1", then when the UDRE flag is set in the UCSRA register (UCSRnA), a "data register empty" interrupt is generated (if the I flag of the SREG register is set to "1")
4	RXEN	Reception permit If this bit is set to one, the receiver is enabled and the operation of the RXD pin (RXDn) is overridden. When the RXEN bit (RXENn) is reset, the receiver is disabled and its buffer is cleared. The values of the flags RXC (RXCn), DOR/OR, FE (Fen) become invalid

Bit	Name	Description
3	TXEN	Transfer permission Setting this bit to "1" enables the operation of the UART/USART transmitter and overrides the operation of the TXD pin (TXDn). If the bit is reset to "0" during transmission, the transmitter shutdown will occur only after the transmission of the data in the shift register and the transmitter buffer has finished.
2	UCSZ2	Packet format In UART modules, if bit CHR9 (CHR9n) is set to "1", 9-bit data is transmitted and received, if 8-bit is reseted
1	RXB8	8th bit of received data When using 9-bit data words, this bit contains the value of the most significant bit of the received word. In the case of USART, the contents of this bit must be read before reading the UDR data register
0	TXB8	8th bit of transmitted data When using 9-bit data words, the contents of this bit are the most significant bit of the word being transmitted. The required value must be written to this bit before loading the data byte into the UDR register

The operation of the transmitter is enabled by setting the TXEN (TXENn) bit of the UCSRB (UCSRnB) register to "1". When the bit is set, the TXD pin (TXDn) is connected to the USART/UART transmitter and begins to function as an output, regardless of the settings of the port control registers. If a Synchronous operation is used, the programm is needed to override the operation of the XCK (XCKn) output (input or output).

The transfer is initiated by writing the data to be transferred into the buffer register of the transmitter data register UDRn. After that, the data is transferred from the UDRn register to the transmitter shift register. Two options are possible in this case:

- recording in the UDR register is carried out at the moment when the transmitter is in a standby state (previous data has already been transmitted). In

this case, the data is transferred to the shift register immediately after writing to the UDR register;

- writing to the UDR register is carried out during transmission. In this case, the data is transferred to the shift register after the transmission of the last stop bit of the current frame.

If a new data word was written to the UDR register during transfer, then after the last stop bit of the previous word is transferred, the new word is transferred to the shift register. If such a record has not been completed by the end of the frame transfer, then the interrupt flag "Transmission completed" of the TXS (TXSn) of the UCSRA register (UCSRnA) is set. The flag is reset by hardware when entering the corresponding interrupt processing subroutine or programmatically by writing to this bit of the log. 1.

The transmitter is turned off by resetting the TXEN (TXENn) bit of the UCSRB (UCSRnB) register. If a transmission was in progress at the time of execution of this command, the bit will be reset only after the completion of the current and delayed transmissions, that is, after clearing the buffer register and the transmitter shift register. With the transmitter off, the TXD pin (TXDn) can be used as a general purpose I/O line.

Receiving data

Receiver operation is enabled by setting the RXEN (RXENn) bit of the UCSRB (UCSRnB) register. When the bit is set, the RXD (RXDn) pin is connected to the USART/UART receiver and begins to function as an input, regardless of the settings of the port control registers. If the synchronous mode of operation is used, it is necessary to programmatically redefine the operation of the XCK output (XCSn).

Data reception begins immediately after the receiver detects a correct start bit. Each bit of frame content is then read at a rate determined by the baud rate controller settings or the XCK(XCKn) clock. The read data bits are sequentially placed in the receiver's shift register until the first stop bit of the frame is detected. After that, the contents of the shift register is forwarded to the receiver

buffer, from which the received value can be obtained by reading the data register of the module.

The flags DOR (DORn) in the USART and OR (ORn) in the UART represent data loss due to receiver buffer overflowing. In UART the flag is set to "1" if the data of the previous frame has not been read from the data register before the end of frame reception (filling of the receiver shift register). The flag is set to "1" in case of reception of the start bit of a new frame when the buffer and shift register of the receiver are filled in USART.

The receiver is turned off by resetting the RXEN (RXENn) bit of the UCSRB (UCSRnB) register. Unlike the transmitter, the receiver turns off immediately after the bit is reset, that is, the frame received at that moment is lost. In addition, when the receiver is turned off, its buffer is cleared, that is, all unread data is also lost in USART modules. With the receiver off, the RXD pin (RXDn) can be used as a general purpose I/O line.

An example of calculations

Calculation of transmission speed

The speed of data transfer (exchange) is calculated according to the formula:

$$f_{ПД} = \frac{f_{CLK}}{16 \cdot (UBRR + 1)}$$

f_{CLK} – is the frequency of the system clock generator (when connecting an external crystal, it is determined by the frequency at which the crystal is manufactured);

UBRR – is the content of the baud rate controller register that can be changed by software (UBRR = 0...4095).

Let's assume f_{CLK} = 16 MHz, UBRR = 24, then:

$$V_{ПД} = \frac{16000000}{16 \cdot 25} = 40000 \quad посилок / секунду = 40000 \quad біт / секунду = 40 Кбіт / с.$$

Then the duration of one package:

$$t_{ПОС} = \frac{1}{V_{ПД}} = \frac{1000000}{40000} = 25 \quad мксек.$$

Since the transmitter contains the UDR data buffer register, into which the next byte for transmission is programmatically loaded, while the previous byte packets are issued from the shift register, then after the transmission of the stop bit a new byte from the UDR register is transferred to the shift register and a new transmission cycle is initiated.

Thus, the transmission time of one byte is equal to 12*t packets, which with a duration of tPOS = 25 μsec is tB = 25*12 = 300 μsec

.3.6. SPI module

The structural diagram of the SPI module is shown in Figure 3.14. The SPI module uses four microcontroller pins. As with most other peripherals, these pins are general purpose I/O port lines. Pins PB4, PB5, PB6, PB7 are used in the ATmega8515x microcontroller.

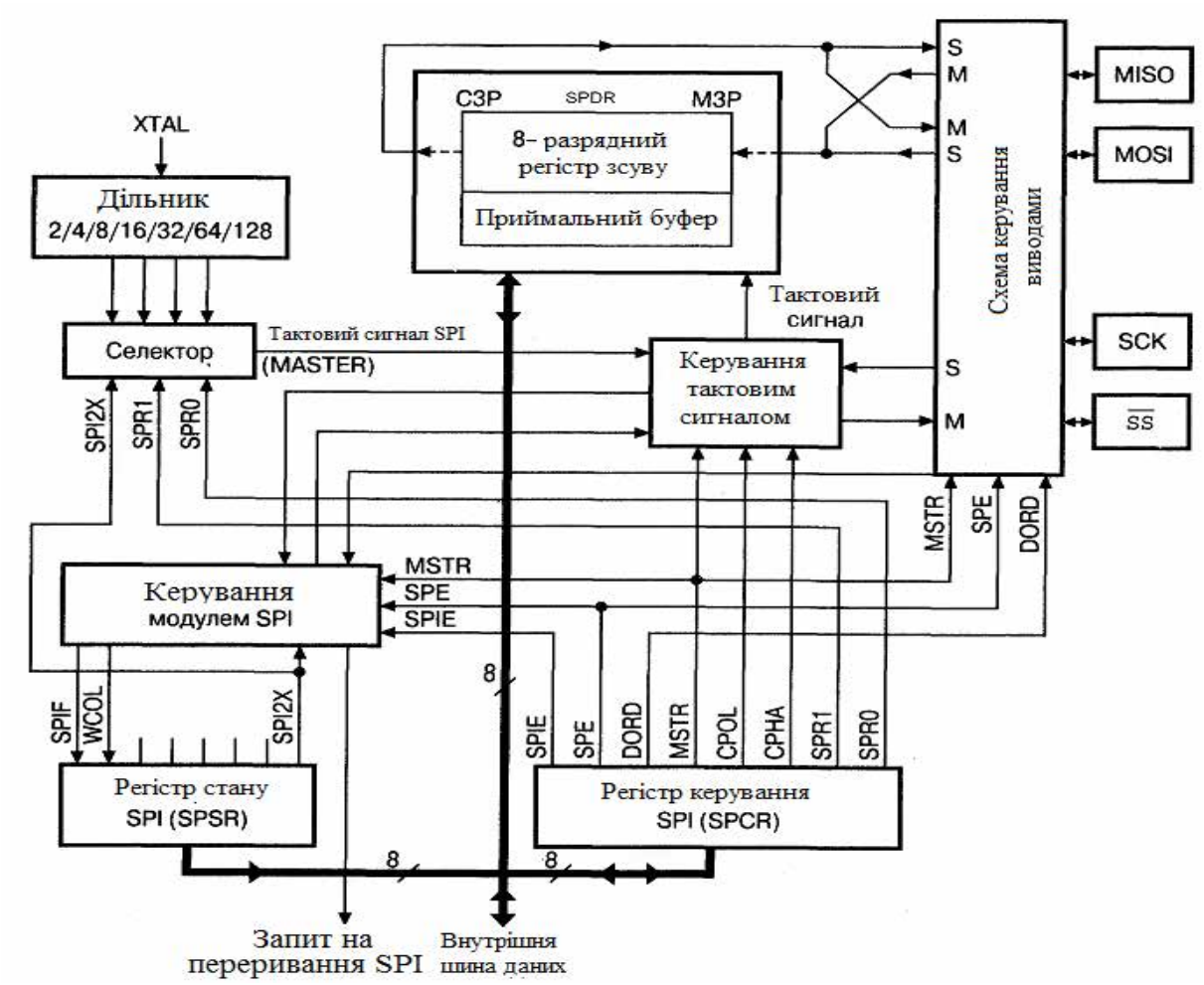


Figure 3.14. Structure diagram of the SPI module

When the SPI module is turned on, the mode of operation of the indicated outputs (direction of data transfer) is redefined according to table 3.19. The direction of data transfer is determined by the state of the corresponding bit of the DDRB register.

Table 3.19. Pins of the ATmega8515x microcontroller used by the SPI module

Pin	Microcontroller	Assigning
SCK	PB5	Output(master) /Input (slave) clock signal
MISO	PB4	Input (master)/ Output (slave) data
MOSI	PB3	Output (master)/ Input (slave) data

$\overline{SS}$	PB2	Choice of device

Table 3.20. Reassigning the operation mode of the SPI module pins

Pin	«Master»mode	«Slave» mode
MOSI	Defined as output by user	Input
MISO	Input	Defined as output by user
SCK	Defined as output by user	Input
$\overline{SS}$	Defined as input or output by user	Input

As can be seen from table 3.20, in certain cases, the user must independently set the operation mode of the output used by the SPI module in accordance with its purpose. At the same time, the possibility of controlling the internal resistors of the terminals working as inputs is preserved regardless of the way of controlling their mode of operation.

The SCK pin of the microcontroller in Master mode works as an output, so the corresponding line of the general purpose I/O port must be programmed as an output.

To control the SPI module, the SPCR control register is assigned, which in the ATmega8515 model is located at address $0D ($2D). The format of this register is shown in figure3.15, and a brief description of the functions of the register bits is shown in table 3.21.

	7	6	5	4	3	2	1	0
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Читання(R)/ Запис ($\overline{W}$)	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$
Початкове значення	0	0	0	0	0	0	0	0

Figure 3.15. SPCR register format

Table 3.21. Assignment of SPCR register bits

Bit	Name	Description
7	SPIE	Enable interrupt from SPI if SPIE = 1
6	SPE	Enabling/disabling the SPI module: SPI = 1 – enabled; SPI = 0 – disabled
5	DORD	Data transfer procedure: DORD = 1 – MZP is transmitted first, DORD = 0 – the first SZR
4	MSTR	Selecting the operating mode ("Master"/"Slave"): MSTR = 1 – Master, MSTR = 0 – Slave
3	CPOL	Determines the polarity of the clock signal, see the table 5
2	CPHA	The phase of the clock signal - determines the moment of reading the signal, see table 5
1, 0	SPR1:SPR0	Transmission speed, see table 6

Control of the state of the module, as well as additional control of the exchange rate is carried out using the SPSR status register located in the ATmega8515 model at address $0E($2E). Bits 7 to 1 of this register are available for reading only, and bit 0 is for both reading and writing. The format of this register is shown in figure 3.16, and the purpose of its digits is described in table 3.22.

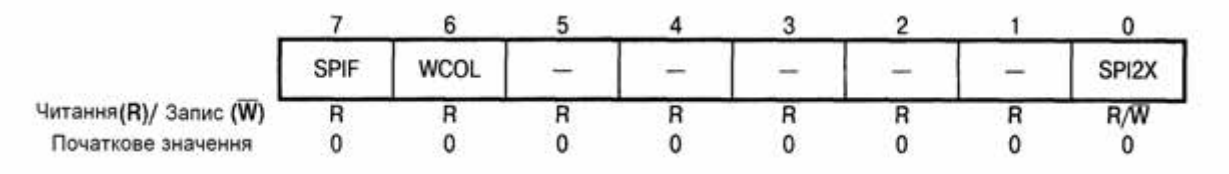


Figure 3.16. SPSR register format

Table 3.22. Assignment of SPSR register bits

Bit	Name	Description
7	SPIF	"End of Transfer" checkbox This flag is set to "1" after the end of transmission/reception of the next byte. If the SPIE flag of the SPCR register is set to "1" and interrupts are enabled, an interrupt from the SPI module is generated at the same time as the flag is set. Also, the SPIF flag is set to "1" when switching the microcontroller from the "Master" mode to the "Slave" mode using the output SS. The flag is reset by hardware, either at the start of the interrupt handling routine, or after reading the SPI status register followed by a call to the SPI data register (SPDR)
6	WCOL	Record conflict flag This flag is set to "1" when attempting to write to the data register (SPDR) during transmission of the next byte. The flag is reset by hardware after reading the SPI status register followed by an access to the SPI data register
5...1	-	Reserved, read as "0"
0	SPI2X	Double exchange rate When this bit is set to "1" and the microcontroller operates in "Master" mode, the frequency of the SCK signal is doubled

The connection of two microcontrollers (master-slave) using the SPI interface is shown in figure 3.17. The SCK output of the master microcontroller is the output of the clock signal, and the output of the slave microcontroller is the input.

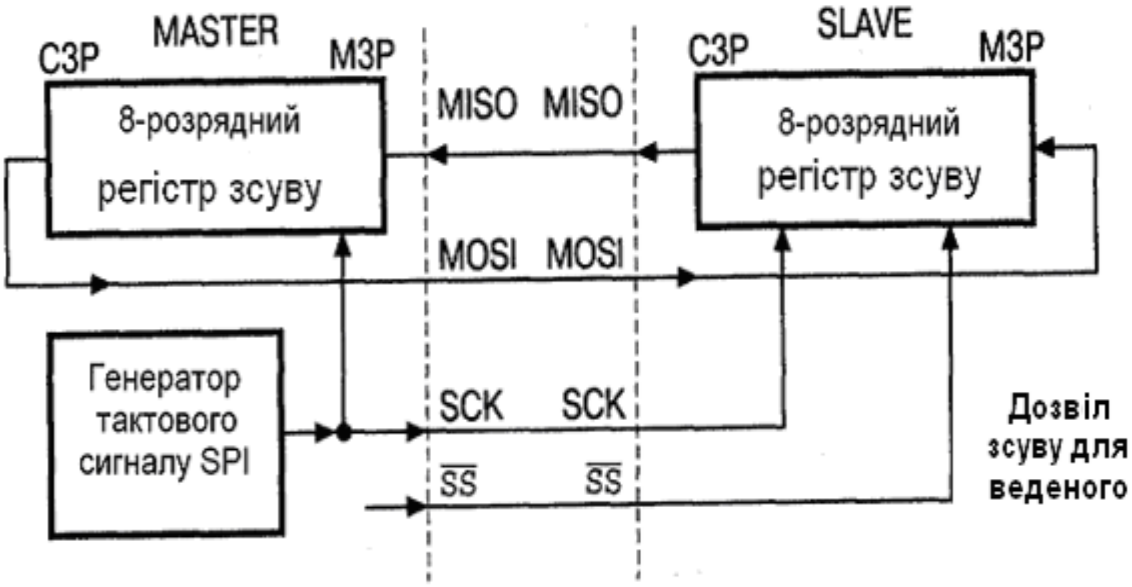


Figure 3.17. Connection of microcontrollers via the SPI interface

It is necessary, first of all, to enable the operation of the SPI module before performing the exchange. To do this, you need to set the SPE bit of the SPCR register to "1". The operating mode is determined by the state of the MSTR bit of this register: if the bit is set to "1", the microcontroller works in "Master" mode, if it is reset to "0" - in "Slave" mode.

Data transfer is carried out in the following way. When writing to the SPI data register of the master microcontroller, the clock signal generator of the SPI module is started, and the data begins to be output bit by bit to the MOSI output and, accordingly, to arrive at the MOSI output of the slave microcontroller. The order of data bit transmission is determined by the state of the DORD bit of the SPCR register. If the bit is set to "1", the lower bit of the byte is transmitted first, if it is reset to "0", the higher bit is transmitted. After outputting the last bit of the current byte the clock signal generator stops with the simultaneous setting of the "End of Transmission" (SPIF) flag to "1". If interrupts from the SPI module are enabled (the SPIE flag of the SPCR register is set to "1"), an interrupt request is generated. After that, the master microcontroller by writing the next byte into the SPDR register can start transferring the next byte or, applying a HIGH level voltage to the input $\overline{SS}$ of the slave microcontroller put the latter into a standby state, and

then, applying a LOW level voltage to the $\overline{SS}$ slave input, start transferring the next byte.

Simultaneously with the transfer of data from the master to the slave, there is a transfer in the reverse direction, provided that there is a LOW level voltage at the input $\overline{SS}$ of the slave. Thus, in each shift cycle data is exchanged between the two devices. At the end of each cycle the SPIF flag is set to "1" in both the master and slave microcontrollers. Received bytes are stored in receive buffers for later use.

Data transmission modes

The SPI interface specification provides for 4 data transfer modes. These modes differ in correspondence between the phase of the SCK clock signal (determines when the signal is read), its polarity and the data being transmitted. There are a total of 4 such combinations determined by the state of the CPHA and CPOL bits of the SPCR register (table 3.236).

Table 3.23. Setting the data transmission mode

Bit	Description
CPOL	Polarity of the clock signal "0" - pulses of positive polarity are generated, in the absence of pulses the output has a LOW level; "1" - pulses of negative polarity are generated, in the absence of pulses, there is a HIGH level at the output
CPHA	The phase of the clock signal "0" - data processing is performed on the leading edge of SCK signal pulses (for CPOL = "0" - on the rising edge, and for CPOL = "1" - on the falling edge) "1" - data processing is performed on the trailing edge of SCK signal pulses (for CPOL = "0" - on the falling edge, and for CPOL = "1" - on the rising edge)

In modes 0, 1 (CPHA = 0) the start of the exchange is determined by setting the signal at the output $\overline{SS}$ kept in the active state - log. 0. At the same time, the master sends the higher data bit to the MISO line if the SZR is transmitted first, or the younger data bit if the MZR is transmitted first (figure 3.24). The host sends a 0.5* T_{SCK} to the MOSI line before the first pulse appears on the SCK line (T_{SCK} is the period of synchro pulses).

In modes 2, 3 (CPHA = 1) the start of the exchange is determined by the first signal change on the SCK line after setting the slave selection signal $\overline{SS}$ to the active log state 0.

The frequency of the SCK clock signal and, accordingly the data transfer rate through the interface are determined by the state of the SPR1:SPR0 bits of the SPCR register and the SPI2X bit of the SPSR register (table 3.24). It is clear that we are talking about a microcontroller that works in the "Master" mode, because it is it, which is the source of the clock signal. The state of these bits is indifferent for a device that is in"Slave" mode.

Table 3.24. Setting the frequency of the SCK clock signal

SPI2X	SPR1	SPR0	SCK signal frequency
0	0	0	f_{CLK}/4
0	0	1	f_{CLK}/16
0	1	0	f_{CLK}/64
0	1	1	f_{CLK}/128
1	0	0	f_{CLK}/2
1	0	1	f_{CLK}/8
1	1	0	f_{CLK}/32
1	1	1	f_{CLK}/64

It should be kept in mind that the operation of the microcontroller in "Slave" mode is guaranteed only at frequencies less than or equal to f_{CLK}/4.

The maximum transfer rate is limited by the signal setup and hold timing diagrams. SPI is used in a wide range of communication speeds from a few Kbit/s to several Mbit/s.

Using output $\overline{SS}$

This pin is intended for selecting the active slave device and is always an input in "Slave" mode (figure 5). When LOW level voltage is applied to it, the SPI module is activated and the MISO pin switches to data output mode (if set by the user, Table 2). The other pins of the SPI module are inputs in this mode. All pins of the SPI module are switched to data input mode when HIGH voltage is applied to output $\overline{SS}$. In this case, the module goes into an inactive state and data receiving does not take place. As a rule, in this state, the program changes the contents of the data register.

It is worth remembering that every time a HIGH level voltage is applied to output $\overline{SS}$ of the slave, the SPI module is reset. Accordingly, if the state change of this pin occurs during a data exchange, both reception and transmission will stop immediately, and the transmitted and received bytes will be lost.

If the microcontroller is in the "Master" mode (the MSTR bit of the SPCR register is set to "1"), the direction of data transmission through the output $\overline{SS}$ defined by the user. If an output is configured as an output, it is used to control the output $\overline{SS}$ microcontroller operating in "Slave" mode.

If, in the "Master" mode, the output is configured as an input, then to ensure the normal operation of the SPI module, it must be supplied with a HIGH level voltage. Applying a LOW-level voltage to this input from any external circuit will be perceived by the SPI module as the choice of this microcontroller as a slave, and accordingly the start of data transfer to it. In order to avoid a conflict on the bus, the SPI module performs the following actions in such cases.

1) The MSTR flag of the SPCR register is reset, and the microcontroller switches to "Slave" mode. As a result, the MOSI and SCK outputs begin to function as inputs (Table 3.24).

2) The SPIF flag of the SPSR register is set, generating an interrupt request from the SPI. If SPI interrupts are enabled and the I flag of the SREG register is set to "1", the interrupt handling routine is started.

Thus, if the master microcontroller uses interrupt-driven data transfer and there is a possibility of applying a LOW voltage to the input $\overline{SS}$, the MSTR flag must be checked in the SPI interrupt handler. When a reset of this flag is detected, it must be set back to "1" by software in order to return the microcontroller to "Master" mode.

Section IV. CONSTRUCTION OF DIGITAL NODES AND DEVICES

4.1 Peculiarities of logical elements schemes

Well-known logical elements (hereinafter -LE) AND, OR, NOT, AND-NOT and others may have some features that need to be taken into account when designing digital equipment.

The AND and OR functions can be defined differently: the AND function equals zero if at least one argument is zero, the OR function is zero if all its arguments equal zero. It follows that for zero signals the AND function acts as an OR function, and the OR function acts as an AND function. The same proposition directly follows from the duality theorem. Logical elements of the same name correspond to basic logical functions (fig. 4.1):

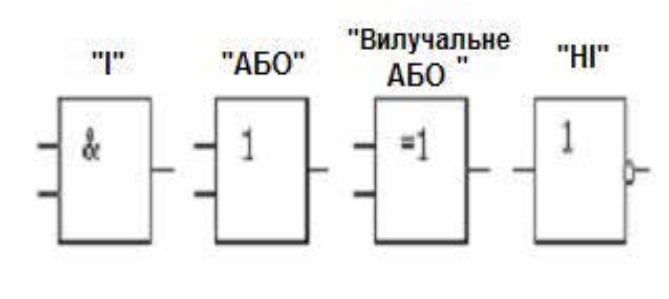


Figure 4.1 – Notations

4.1.1 The basic logical element

Figure 4.2 shows a simplified scheme of LE AND-NO on bipolar *n-p-n* transistors and its notation.

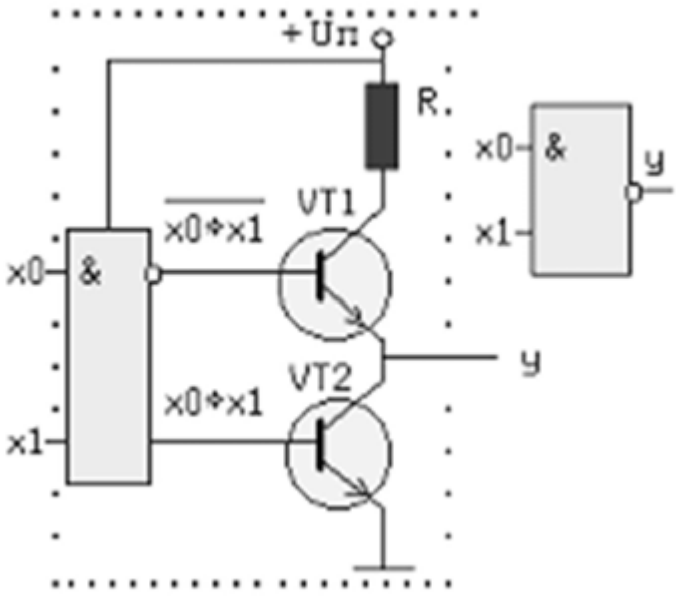


Figure 4.2 – Basic logic element (hereinafter – BLE)

The voltage at the bases of transistors VT1 and VT2 is in antiphase and, if x0* x1 >1, then the lower transistor is open and the upper one is closed, since x0*x1>0. The collector potential VT2 in this case is approximately equal to zero

and, therefore, y >0. At other values of x0 and x1, the lower transistor is closed, and the upper one is open, and at the output of the circuit there is a high level, that is, the circuit works as an AND-NO element. The outputs of several BLEs absolutely cannot be connected together, because *if n* = 1 elements are in the state "1" and the *n-th* is in the state "0", then *n* = 1 transistors VT1 will "sink" currents into a single transistor VT2 of the *n*-th element. The total current may exceed the permissible value and VT2 will fail. Obviously, such possible modes must be taken into account when designing digital devices.

4.1.2 Element with an open collector

The AND-NO logic element with an open collector does not have an internal resistor connected to the power source and is indicated in the field of the element by a rhombus with a line at the bottom (fig. 4.3). It is not difficult to get a logical zero at the output of such an element, for this there must be a high level at the base of the transistor (x0 >x1.1). But a logical unit at the output can be obtained only by connecting an external resistor connected to the power source to the collector of the transistor (the base of the transistor in this case should have a low potential). Such a *resistor* is called a pull-up resistor, because the output voltage of the logical "1" is pulled up to the voltage of the UPS power source $^{U_{П}}$.

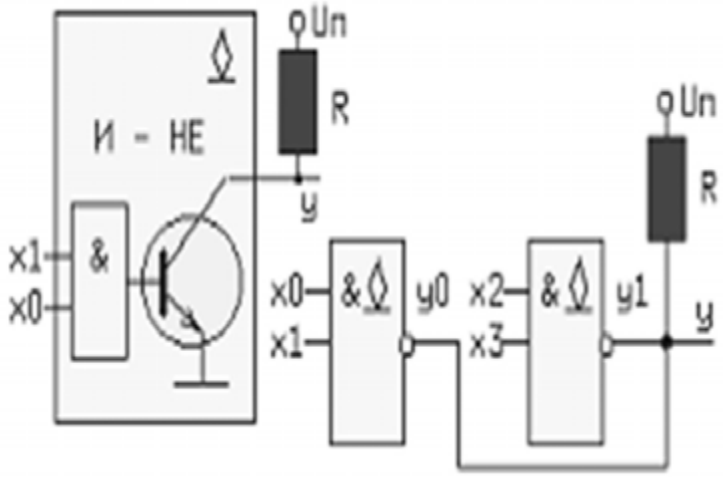


Figure 4.3 – LE with an open collector

In addition to the resistor, relay windings, motors, LEDs, etc. can be connected to the open collector externally. Open manifolds of several elements, in contrast to the basic logic element can be connected together forming an "assembly AND" for direct values of variables because y =y0* y1,1 when y0 >y1,1. Sometimes such a scheme is called "assembly OR", since y=x2* x3 according to the duality ratio is equal

x0*x1=x2*x3.

The logic element AND with an open emitter is indicated by a rhombus, but with a line from above. Open-collector LEs are also used in memory chips and bus formers.

4.1.3 Tri stable elements

Along with the two logic states there is a third technological state, when the output of the logic element is disconnected from the internal circuit. In this case, the resistance between the output and "ground" becomes very large, and the output of the microcircuit does not affect the outputs of other microcircuits connected to it (Fig. 4.4).

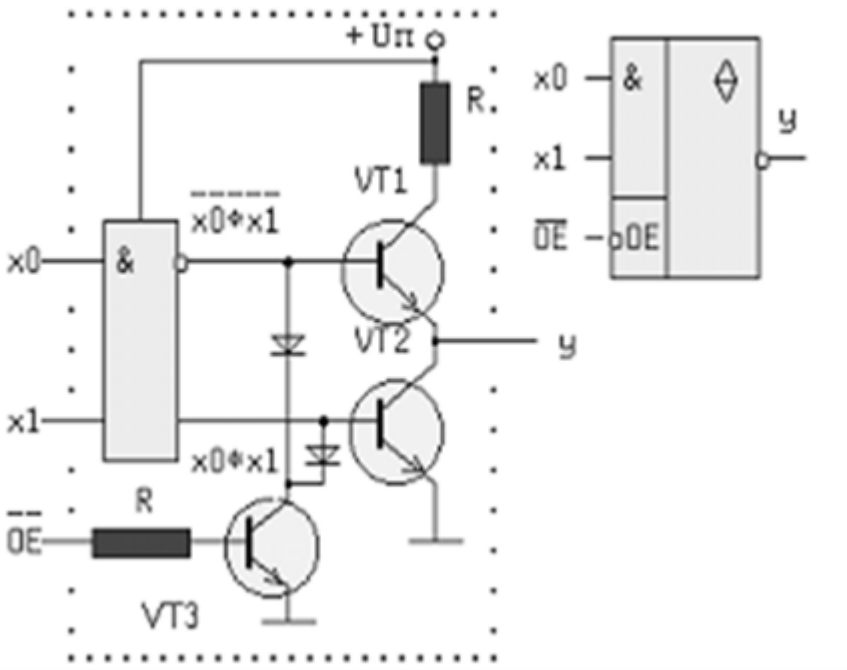


Figure 4.4 – Tristable LE

The outputs of several such elements can also be connected together. Such an inclusion, a type of "mounting AND" is used where several signal sources are connected to the inputs of one or more receivers in turn, without interfering with each other. The third state is also called the high-impedance or Z-state. The AND-NOT scheme with the Z-state is shown in figure 4.4 on the left, and its notation is on the right.

Such elements are used where it is necessary to transmit information along one line from several sources to one or several receivers (for example, to a microprocessor - MP(CPU)). In addition, only one source can be active. Since there is only one data line, in order for the outputs of the passive sources not to distort the information at the output of the active source (or, as it is known from electrical engineering do not shunt the "Yk" output), they must be transferred to the third state, that is, disconnected from the data line.

4.2 Time parameters of logical elements

Consider the response of the inverter to a change in the input signal. The inertial properties of the inverter lead to a signal delay during its passage from the input to the output (Fig. 4.5).

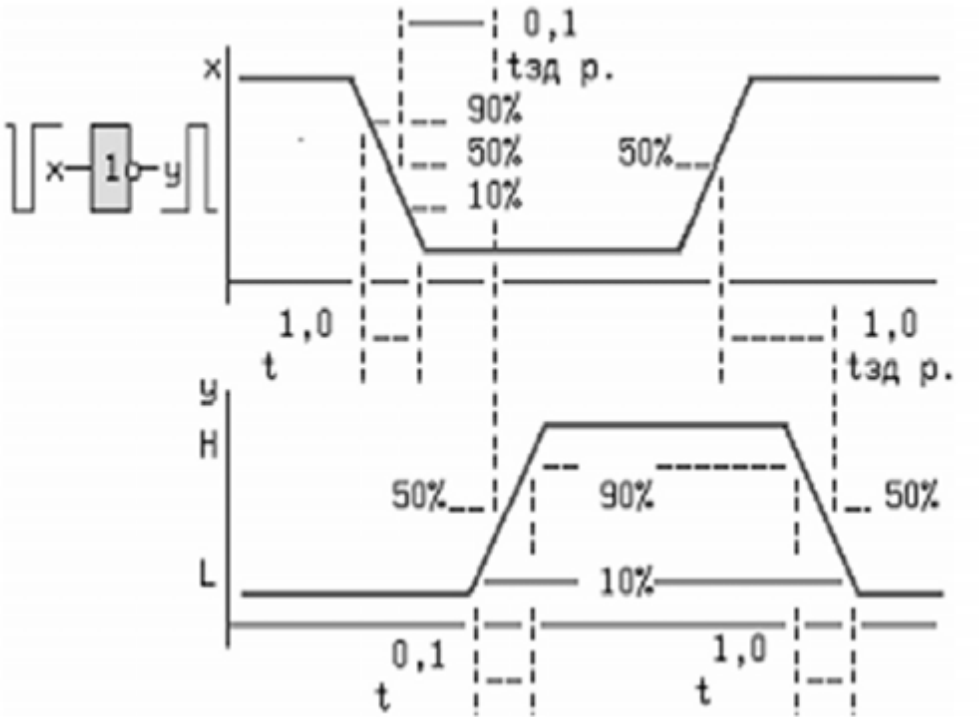


Figure 4.5 – Time parameters

The process of voltage change from a low level L to a high level H is called a signal edge (positive edge), and the reverse process is called a decline (negative drop, negative edge). The duration of fronts in figure 3.5 is marked *t*1,0 – negative and *t0*,1 – positive. The values $t_{зд.р}$ 0,1 i $t_{зд.р}$ 1.0 indicate the delay time of signal propagation from the input to the output during the transition from 0 to 1 and vice versa. The minimum duration of the pulse at the input of the element $t_{i.}$ min proportional to the average value $t_{зд.р.сер.}$ equal to the half sum of *t*zd.r 0.1 and *t*zd.r

1.0. The maximum frequency of input pulse *F*max is backward proportional to *t*zd.p.cep. It follows from the above mentioned that the speed of the element is higher, the smaller *t*zd.p.cep is and is denoted by *t*pd (propagation delay).

Two more parameters are used in digital and microprocessor technology:

- t_{ycm}/t_{SU} – input signal setup time (clock setup time).
- The time interval between the start of a signal at one specified input and an active transition at another specified input (usually clock);
- t_y/t_H – hold time. The time during which the signal is held at a given input after an active transition at another given input (fig. 4.6).

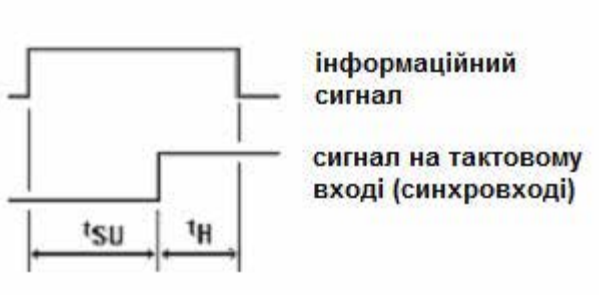


Figure 4.6 - Setup time and hold time

Another parameter is used in sequential schemes: *t*co (clock to output delay) - the delay in the appearance of the output signal from the moment of the appearance of an active edge at the clock input. Definitions of the above values with their domestic and international designations are given in the designation section of some microcircuit parameters.

The speed of the circuits depends not only on the listed parameters, but also on the form of the algebra of the LF representation (Fig. 4.7).

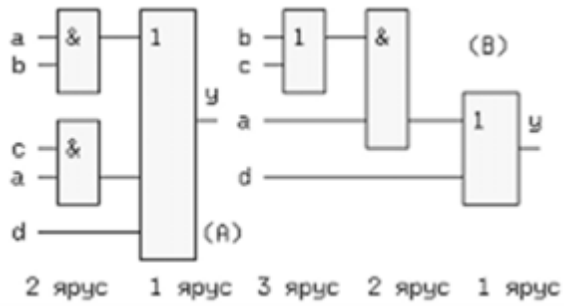


Figure 4.7 – Two ways to optimize logic circuits

Let y=a*B +c*a+d +a*b +c+d. Circuit (A) corresponds to the first form (DNF), and circuit (B) to the second.

If the average delay time of the signal in each element is the same, then $2^{t}_{зд.р.сер} > 3^{t}_{зд.р.сер}$

and two-tier schemes (DDNF) are usually faster. However, the number of elements and/or conductors can be reduced (in scheme (B) one conductor is less) in the entry with brackets. The choice of a circuit that is optimal in terms of speed or in terms of the minimum number of connections/elements remains up to the developer. Most automated design systems (hereinafter referred to as CAD) have the ability to optimize the digital devices being developed according to the specified criteria, for example, "MAX+plus II", "Quartus II", "WebPack ICE2, etc.

4.3 Transient processes in logical circuits

The difference in the delay time $t_{зд.р}$ from zero during the passage of the signal through the logic circuit can lead to interference in the output signal. These interferences take the form of short pulses, and in some cases lead to serious failures in the operation of circuits. Consider the device in figure 4.8. If the elements of the circuit do not delay the signal, and *x0* and *x*1 are in antiphase, that is, *x*0 >*x*1, then *y* >*x*1**x*1>1. If each of the four inverters has a delay $t_{зд.р}$, then *x*0 is

delayed relative to $x0$ by $4^{t}_{зд.р}$ and a short unplanned “negative” pulse (interval 1...2) occurs at the output of the circuit, shifted by $t_{зд.р}$ of the AND-NO element (interval 0...1). The process of passing input signals to the common output is called competition or "racing".

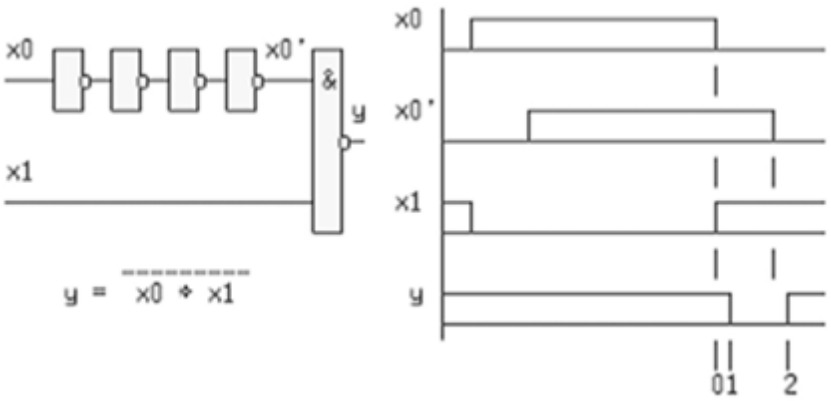


Figure 4.8 – Formation of an interference signal

The harmful effect of "races" can be eliminated in several ways, one of which is to add an additional additive to the LF. Let some LF equal

F =x1*x2+x1*x0,

then, when $x2 > x0 > 1$, interference may appear due to the fact that the $x1$ signal is delayed relative to $x1$ by the amount of the inverter delay (fig. 4.9).

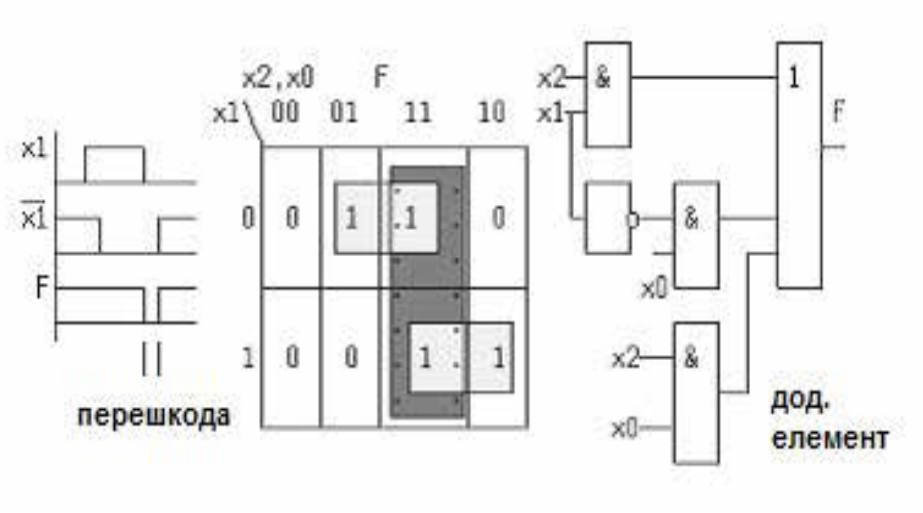


Figure 4.9 – Method of removing interferences

Adding an additional implicant (dotted in the table) eliminates the problem, as in the critical situation when $x2 \square x0 \square 1$, the additional component $x0^* x2 > 1$ and the function $F = x1^*x2 + x1^*x0 + x0^*x2$ is always equal to unity at $x2 > x0 > 1$.

In display devices such short interferences can be ignored, as they will be imperceptible to the eye, but in the shaper of control values, the influence of such interferences cannot be neglected.

The effect caused by competition can also have a positive value, for example in a frequency doubler (another useful application of "EXCLUSIVE OR"). For clarity, the delay of the output signal is not shown in figure 4.10.

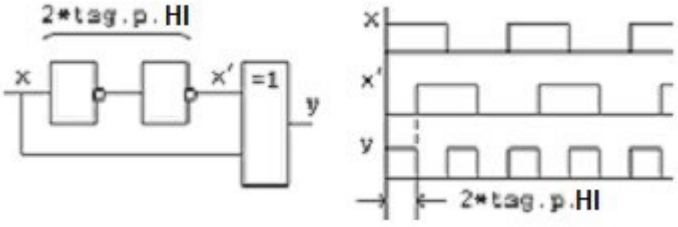


Figure 4.10 – Frequency doubler

4.4 Branching coefficient

The branching factor (*K* times, N) or load capacity is the maximum number of inputs that can be connected to a separate output of the microcircuit (hereinafter referred to as MC). In addition, the sum of the input currents must be less than the output current of a separate output of the MC (Fig. 3.11). If the MC has several outputs, the total current of all outputs must not exceed the passport value for this microcircuit, even if individual outputs of the microcircuit are unloaded.

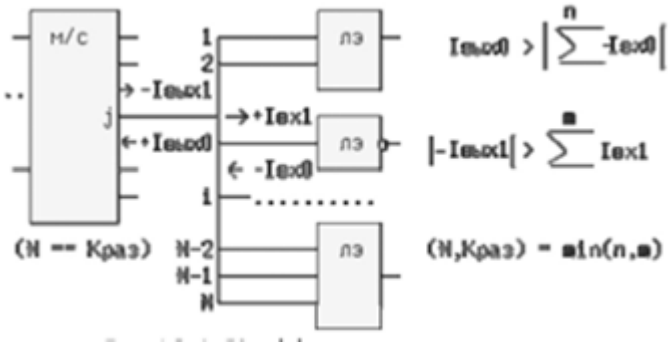


Figure 3.11 - Branching coefficient

To determine *N*, find separately two sums of input currents for logical 0 and 1 at the *j*-output. The minimum of the two values will be

$K_{раз=}Min*M,n^2$,

where *M, n* are integer values.

LIST OF REFERENCES

1. Osnovy skhemotekhniky elektronnykh system:Pidruchnyk / B.I.Boiko, A.M.Hurzhii, V.Ya.Zhuikov та ін. - K.:Vyshcha shkola, 2004. – 527s.: il.

3. Spravochnik po tsyfrovoi skhemotekhniky /V.I. Zubchuk, V.P. Sigorskyi, A.N.Shkuro. —K.: Tekhnika, 1990. - 448 s.

5. Elektronnye tsyfrovye vychislitelnye mashyny. Samofalov K.H., V.I.,Korneichuk, Tarasenko V.P. Izdatelskoe obiedinenie « Vyshcha shkola», 1976., s. 480.

7. Prianishnikov V. A. Elektronika: Polnyi kurs lektsii - 4-e.izd -.SPB: KORONA, 2004. - 416 s., il.

8. Osnovy tsyfrovoi elektroniki. - http://www.bashedu.ru/wsap/ posobie/Content.htm.

Kulinich O.M., Lakhno V.A., Kasatkin D.Yu., Kharchuk N.S.

MODERN ELEMENTARY BASE OF DIGITAL SYSTEMS DESIGN

Educational and methodological manual

Видавець ФОП Ямчинський О.В.

03150, Київ, вул. Предславинська, 28

Свідоцтво про внесення до Державного реєстру

суб'єкта видавничої справи ДК № 6554 від 26.12.2018 р.

Виготовлювач ТОВ «ЦП «КОМПРИНТ»

Формат 60×8 /16. Наклад 50 пр. Ум. друк. арк. 23 ,8. Зам. № 143

03150, Київ, вул. Предславинська, 28

Свідоцтво про внесення до Державного реєстру

суб'єкта видавничої справи ДК № 4131 від 04.08.2011 р.