POSSIBLE OPTION OF ASSOCIATIVE COOPROCESSOR ORGANIZATION AT FUNCTIONAL LEVEL ON PLIC BASIS FOR SPECIALIZED COMPUTER SYSTEMS

[a]MARTYSHKIN A.I.

Penza State Technological University, Russia, Baydukova passage / Gagarina street, 1a/11, Penza, Penza region, 440039, Russian Federation

e-mail: [a]alexey314@yandex.ru

Abstract: The possibility of associative coprocessor practical module realization on the modern element base for specialized computer systems is considered in the article. The purpose of the article is the development and the study of possible ways for an associative coprocessor block development based on PLICs that perform associative functions and data storage functions for computer systems. In this paper, an associative coprocessor is proposed that is connected to the PCI bus of the computer system, providing search and "more - less" comparison operations for 32 words simultaneously preloaded into the associative memory. The final conclusions are drawn in the article. The simulation of the associative coprocessor operation was carried out in CAD Web pack ISE of Xilinx company. The results obtained in this article may be applied in the search engines of various purposes: in database servers, search engines, and for the rapid implementation of search tasks in operating systems.

Keywords: associative memory, addressing, coprocessor, computing system, memory cell, multiple coincidence analyzer, memory addressing, bus interface.

1 Introduction

The main area of computer application today is the work with large amounts of data, in which the most laborious operations are all sorts of data searches and sorting. Existing computing systems (CS) use the address memory architecture, i.e. in order to search for data in memory, it is necessary to read each memory module address and compare it with the search argument. Thus, it takes a lot of computer time to search for information in memory. This circumstance has a very negative effect on CS speed as a whole. It is much faster to access information by the association (content). The essence of addressing principle by content is the following one (Figure 1). There is an array of data with the capacity of N words, it requires you to find all the words beginning with the "A" character and ending with the "H" symbol. The search argument here is the word A *** H, where the * mark denotes the digits that do not affect the search result. The memory array on the hardware level is structured in such a way that the signal of the coincidence appears on the output of memory cells (MC), the contents of which coincide with the value of the arrived search argument. Further, the sampling of MC is performed according to the produced coincidence signals.

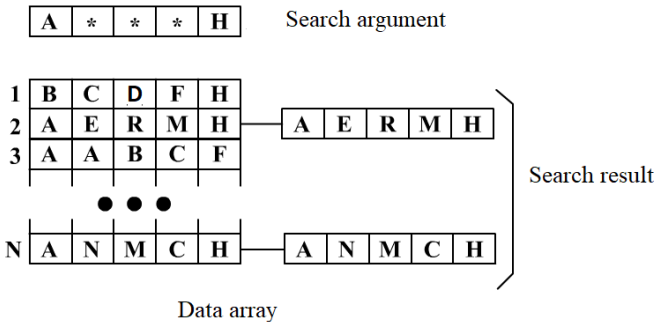

Fig 1. The essence of addressing principle by content

2 Theoretical part

This article as a whole is of a research nature. In the course of the subject area study, the literature (Kohonen, 1982) was analyzed in order to search for unaffected and unresolved problems. A number of issues related to the possibility of hardware implementation of an associative coprocessor for fast data search has not been reflected in publications adequately, but some of the problematic issues were considered in (Tanenbaum & Bos, 2015; Martyshkin, 2011 Villalobos Antúnez et al, 2013).

The purpose of this article is to develop and to study an associative coprocessor module based on PLIC for specialized computing, for example, multiprocessor systems. This issue is topical today due to global informatization and an almost universal operation with colossal volumes of various data. In order to achieve this goal, the article solves device structure determination problems and the principles of its functioning.

The proposed hardware coprocessor has the ability of address and associative access to the data stored in memory. Addressable access is required to work with a specific record and to use test libraries developed for address memory (Martyshkin, 2015; Martyshkin & Yasarevskaya, 2015; Kolesnikova & Kamasheva, 2017).

The device consists of two parts: the main part that realizes the functions of the asocial co-processor, namely: the usual (address) record in the associative storage device (ASD); associative recording in ASD; ordinary reading from ASD; associative reading from ASD; the search for coincidences and a part of the interface with the CS on which the function of signal conversion is assigned. These signals coming from the central processor (CPU) are converted to the signals with which the coprocessor will operate, i.e. this part of the device organizes the interface with the CPU.

Today, associative access to data is realized in two ways: software way based on the distribution of memory, depending on the content of data and implemented by software, and hardware way, based on the use of special hardware designed for data storage and associative search. It can be implemented in the form of parallel ASD, where the search argument enters all the MS in parallel. Thus, the mass comparison operation is performed, and the search is performed in one clock cycle. Another variant of the hardware method implementation of associative data access is the consecutive-bit ASD, where the search occurs bit by bit. In this case, the search time directly depends on the number of data bus bits (Martyshkin, 2016a; Martyshkin, 2016b).

The processes similar to the biological mechanisms of data remembering and processing can be represented using various models of associative memory (AM), which allow to display the relations (associations) of arbitrary complexity between information objects. However, all these relations can be implemented in the form of simple constructions - the triples of components: an ordered pair of information objects O and V and the relation type A: $O \xleftarrow{A} V$ (Ognev & Borisov, 2000). One of the simplest AP models for mapping such relationships is shown in Figure 2, a. The model consists of an associative storage medium associated with two input channels and one data output channel.

During the record stage, the input information is fed to the input K from the first input channel, and the characteristic information C, representing the context in which the input information is written into the memory, is fed via the second channel.

At the associative sampling stage, when the key K appears, the response R is formed on memory output, associated with the K key. Thus, the information written in the memory can be selected using any of its fragments applied as search engines. Specifying different context C, you can specify the information to be sampled more precisely.

In the context of introduced Φ3 definition, there is the issue of structured data accumulation and retrieval organization in such a way that access to them is possible on the basis of an associative sampling. Figure 2b shows the AP model, which allows to answer the following questions: in what you can you organize and record the elements of structured information, and also perform the search process cycling, in which a selected information element becomes the key for new information search.

The sets of values can be entered on the three input channels simultaneously. An output channel is used for information sampling. The address information K(t) is fed by first channel at the time t, and the second one is used to transfer the sign C(t). The response R(t) is also fed through the feedback channel to the input of the associative medium. During the operation of such an AP, the keys K(t) and the signs C(t) are fed through time intervals corresponding to the feedback channel delay.

The memory process will be considered under the assumption that the triple (K(t), C(t), R(t-Δ)) is a single static image, given at time t. At that, its simultaneous record into memory is possible by one operation. Suppose also that at the stage of recording R(t) and K(t) are the same.

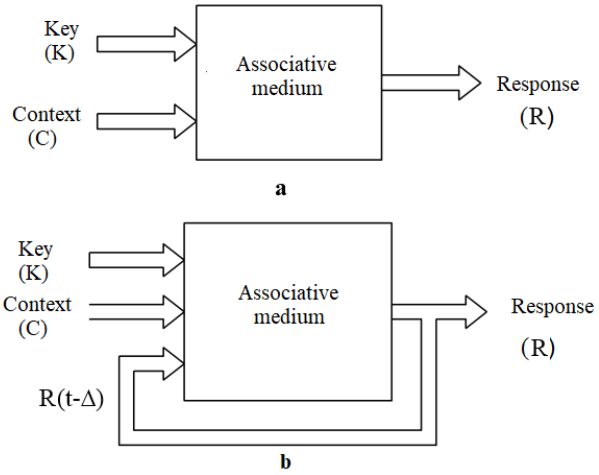

Fig 2. A model of associative memory without feedback (a) and with feedback (b)

During the record stage, K(t) and C(t) are fed to the AP inputs, and R(t) is developed at output, identical to K(t). After this, R(t-Δ) is formed with the delay Δ at the input. Each new triple, appearing at the entrances, is recorded in memory.

At the sampling stage from AP, the K key connected with the context information C, is fed to the input, after which K can be removed from the input. Thus, the copy of K appears at the output as a response. When the delayed signal R(t-Δ) appears at the memory input, the pair (C, R) becomes a new key sign, leading to an associative sample of the next image R(t), etc. Thus, the entire recorded sequence of images is selected together with the context information. The considered model implements the memory suitable for structured knowledge record and retrieval.

The connection of the described device to CS is possible in several ways (Zilker & Orlov 2011): direct connection to the processor bus, at that, the coprocessor is included in the motherboard; the connection to the serial interface (USB), with this method of connection the coprocessor is implemented as a separate housing and is provided with a separate power supply unit; the connection to the computer expansion bus (PCI), in this case the coprocessor is executed in the form of an expansion board (Martyshkin, 2018***).***

According to the mentioned possible ways of a developed device connection to the CS, let's determine the architectural and the structural features of the associative coprocessor, having previously compared them with existing analogues.

As was mentioned above, the device will have to be included in the motherboard, with the direct connection to the processor bus which will lead to the coprocessor cost and its universality increase. The speed of such a system is high. The coprocessor, connected via the USB interface, will look like a separate external module, but it will work in sequence, which leads to performance decrease. But such a block is comparatively cheaper than the previous one. The connection of the coprocessor to the PCI bus will allow to implement it in the form of an expansion board, which is relatively inexpensive to implement, in contrast to the mentioned analogues. At that, the work with the module will be carried out on a parallel interface, which will allow to achieve maximum performance in comparison with analogues.

Having analyzed the advantages and the disadvantages of known and practiced method organization for the main part of the device and the part of interface with the CS, it was decided to perform the main part in the form of parallel AMU, since this method has the maximum capacity. The connection to the special CS is realized via the PCI bus, since it has a sufficiently high throughput. Moreover, the associative coprocessor will be located in the address space of CS input/output.

3 Determination of associative coprocessor module structure

The simplified block diagram of the coprocessor can be represented as two blocks (Figure 3, a): the coprocessor (main part) and the PCI bus interface (the part of the interface with the system).

Having analyzed the chosen method of the main part implementation in the coprocessor structure, it was decided that the main block is the memory module of AMU, which is a MC array and performs the functions of data storage and association search with an argument. A MC consists of a storage element that performs data storage functions and a comparison scheme that performs the search, i.e. generating the signals indicating the equality or inequality of a cell content with an argument (Figure 3, b).

In order to implement the functions of data record and reading from a MC, it is necessary to add a multiplexer and address decoder block in the co-processor in order to generate the "Cell Select" signal for a particular MC. Also, you need to include a memory block for reaction record in the coprocessor module to store the values of the responding cells. In order to calculate the

number of responding cells, a block of coincidence counting has been added to the coprocessor. The block of multiple coincidence analyzer (MCA) is necessary for priority selection. The results of its operation generate the signals for an encoder block necessary to convert the binary sequence into an address, with the associative address reading or writing arriving at the inputs of the address selector. There is an argument register in a module to store a search argument. The command decryption unit controls the coprocessor operation and generates control signals. The register of the mask is excluded from the composition of the coprocessor, since the comparison circuit generates three signals ("EQUAL", "MORE" and "LESS") and there is no need to mask the search argument bits (Figure 3, c).

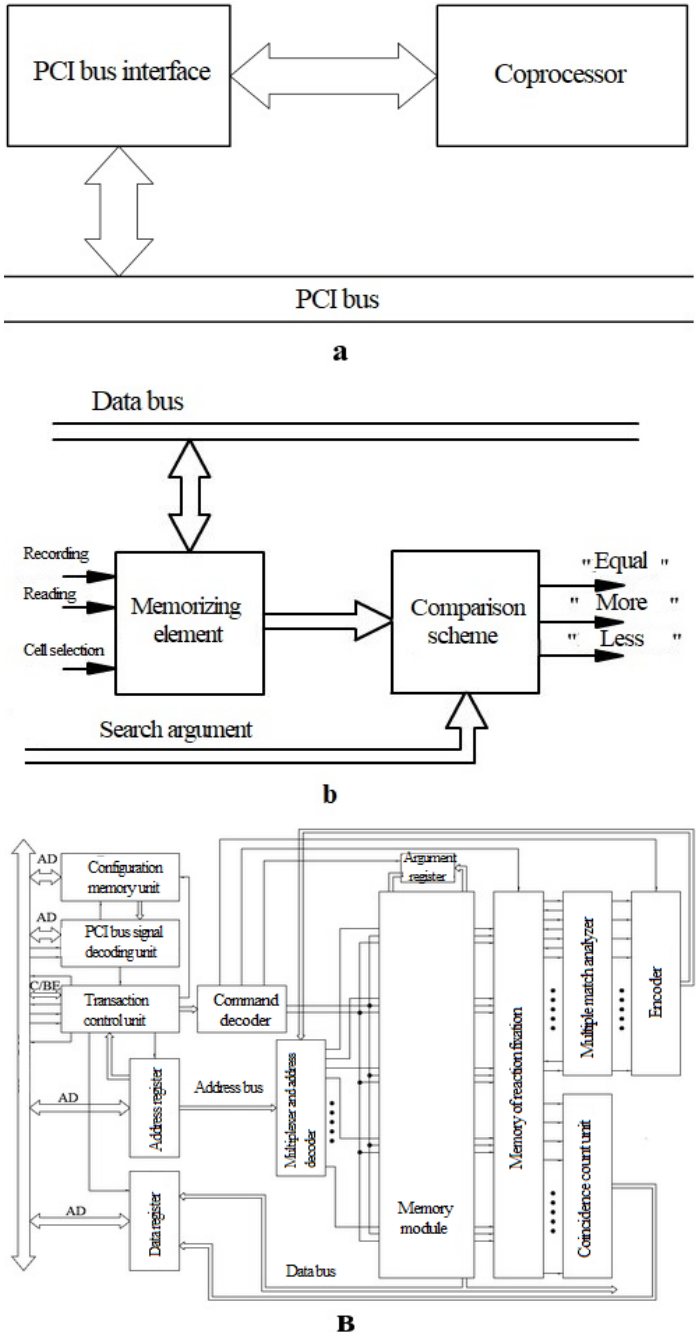

Fig 3. Simplified block diagram of the associative coprocessor module (a); block diagram of memory cell (b); a detailed block diagram of the associative coprocessor module (c)

4 Associative coprocessor module description at the functional level

The basis of the associative coprocessor is the memory module, which is an array of MC. The memory element of a MC is implemented on a parallel register, which is an array of D-triggers, providing the maximum performance and minimum logic necessary to ensure the storage of information. The main signals for the register are the following ones: a 32-bit D signal, which receives data, CE signal and a 32-bit Q signal, from which the data is read stored in the register. The comparison scheme is implemented on the basis of a comparator. The main signals for it are the following ones: 32-bit signals A and B, to which the arguments are fed for comparison, the signals =, <and>, from which the results of comparison are read. A buffer element is included in a memory cell to disconnect the output data bus of

MC from the common output bus. The main signals of the buffer element are the following ones: 32-bit signal D, 32-bit signal Q and signal T. The functional diagram of the MC is shown on Fig. 4, a. The principle of MC operation is the following one. The ArgI input is supplied with the search argument. The data falls into the MC through the DataI input. From the DataO output, the data is read from a MC. The Write and CS inputs are used to control MC operation. MC is selected by CS signal, i.e. the buffer element BUFT passes the signals from the output of the register RG to the output bus DataO. On a single Write signal, the data from the DataI input is written to the RG. Equal, More and Less outputs form the signals "Equal", "More" and "Less" respectively.

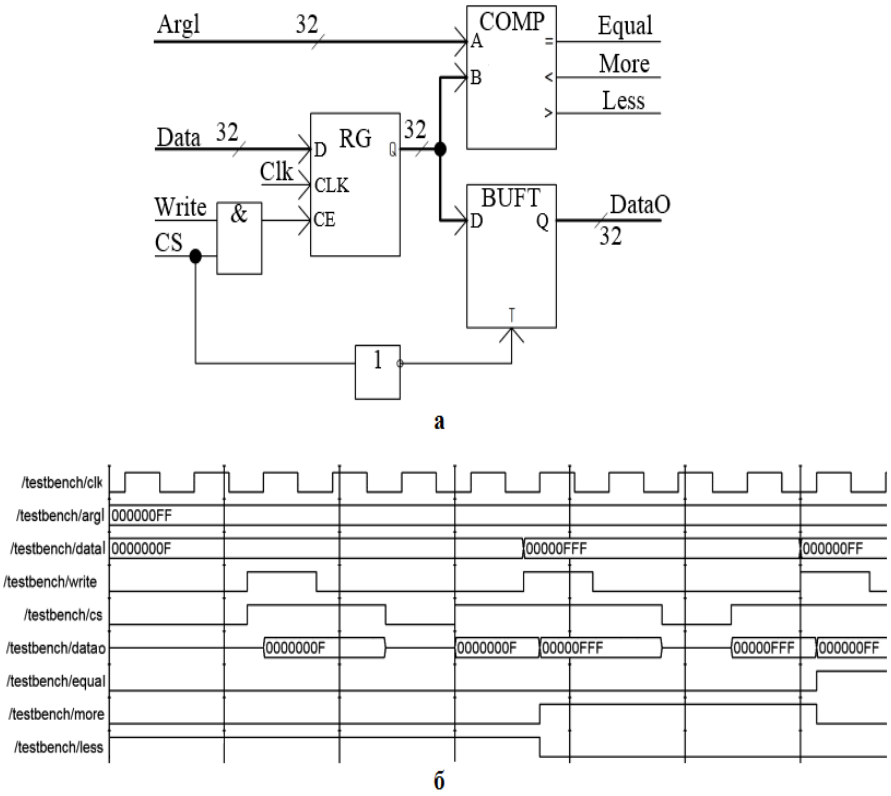

Fig 4. The functional diagram of the memory cell (a); time diagrams of memory cell operation (b)

The time diagrams of MC operation, confirming its efficiency, are shown in Fig. 4, b. Here the record of a number of values is shown: "F", "FFF" and "FF". The search argument is "F". It is seen from the obtained time diagrams, while the signal CS is equal to logical zero, the output bus (DataO) is in the third state, i.e. it is disabled. When a logical unit is fed to the CS input, the values stored in RG can be read from the output data bus (DataO). The record to the cell is performed by sending a logical unit to the Write input. Figure 4, b shows that as soon as a new value is recorded in a MC, the results of the search are set on the output Equal, More and Less.

AMS can be implemented on the basis of the shift register (Figure 5, a) and on the basis of the priority analyzer (Figure 5, b). The main elements of AMS based on the shift register are the looped shift register and the address counter. The result of all MC search is written to the looped shift register (reaction fixation memory). Then a sequence of clock pulses is fed to the register and to the counter. The contents of the register are shifted toward the upper bits until the first one has a logical "1". At this point, the clock signal is blocked automatically. If in the initial state only zeros were written in the counter, then at the end of the calculation, its contents directly indicate the address of the first matched word. This code is put in the address register, after which the word is read. Then the unit in the first digit of the register is reset and the clock pulses are resumed automatically. Again, the contents of the register shift up until the next unit appears in its first digit. After that, the next matched word is read, etc., until the whole queue is serviced. It should be noted that with such a sampling organization, there is no need for an address coder.

The AMS based on the priority analyzer consists of D-triggers that perform the memory functions of reaction fixation and combinational logic. This circuit operates on the front of Clk signal. The search results in a MC are entered into the inputs M0 ... M31, by Fix signal they are fixed on the D triggers. The Work signal permits AMS operation. The priority analyzer is a logical circuit that allows you to select the line with the smallest number among your inputs set to "1". It is built according to the principle of bit consecutive connection. Each single input of this circuit blocks the action of lines with large numbers, thus, only the output corresponding to the first active line is set in the unit. A single signal corresponding to the first active line goes to the output automatically, and the function of the reset signal is the reset of the first of the "responding" triggers.

The memory for reaction and AMS fixation is implemented on the basis of a prioritized analyzer scheme, since this circuit has a faster response time than a scheme based on shift register. The resulting time diagrams of the scheme operation are shown on Figure 5, c.

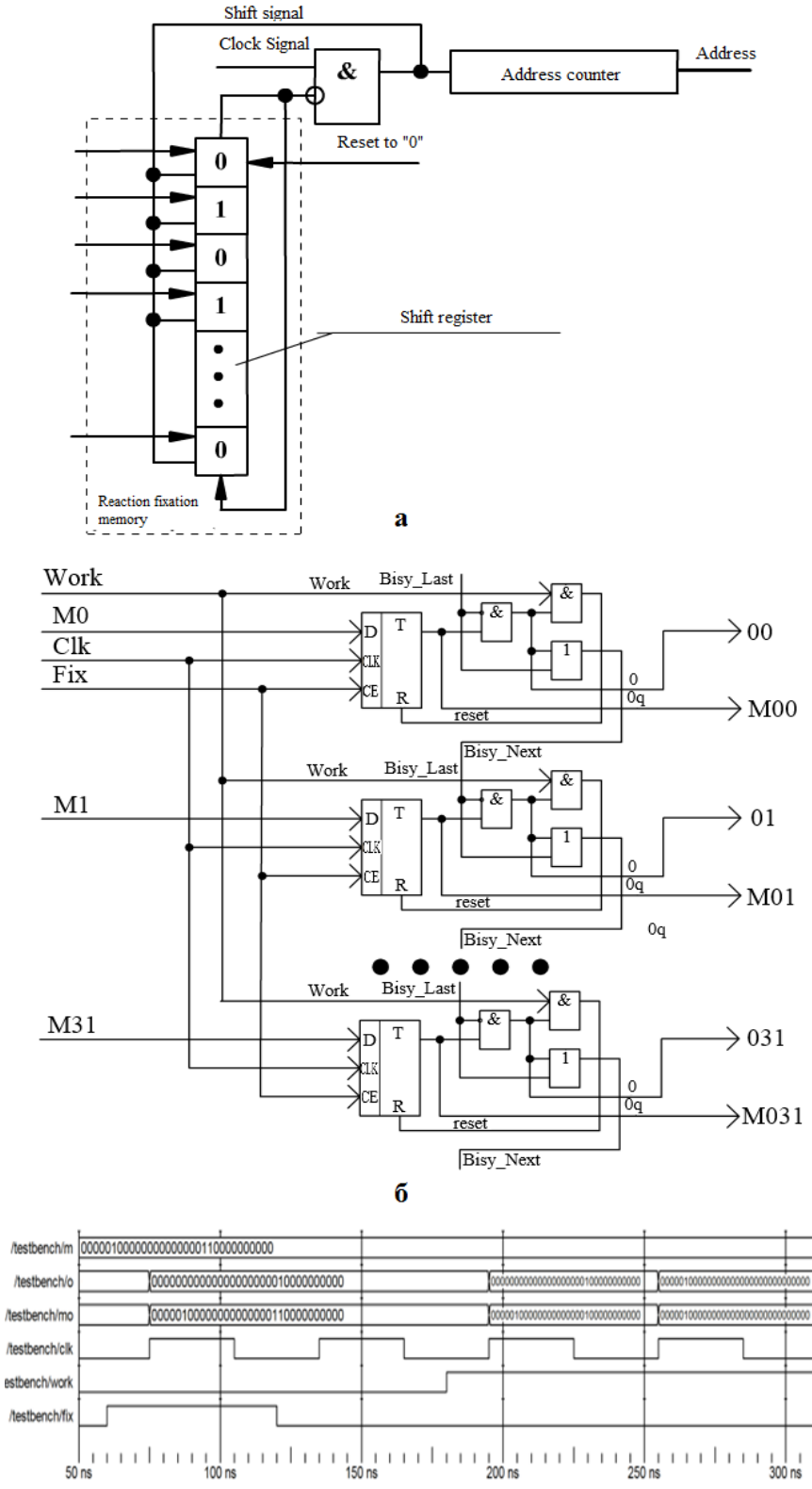

Fig 5. Multiple coincidence analyzer based on the shift register (a) and on the basis of the priority analyzer (b); time diagrams of memory operation for reaction fixation and the analyzer of multiple coincidence (c)

The unit of coincidence calculation can be implemented in several ways: on the basis of the shift register and the counter and on the basis of adder cascades.

The scheme of the coincidence counting unit based on the shift register and the counter (Figure 6, a) consists of a shift register into which the sequence of ones and zeros is loaded, corresponding to the responded and non-responded cells, the counter that counts the quantity, and the element "Logical AND," which controls the counting process. A clock signal is supplied to a circuit. The contents of the shift register are shifted one bit up and depending on the value, the counter value is increased or not

on the shift register output, i.e. all the digits of the register equal to one are calculated. The number of clock pulses required to count all the cells is equal to the number of all memory cells (in our case it makes 32 pulses). In addition, this scheme requires an additional step for a shift register loading.

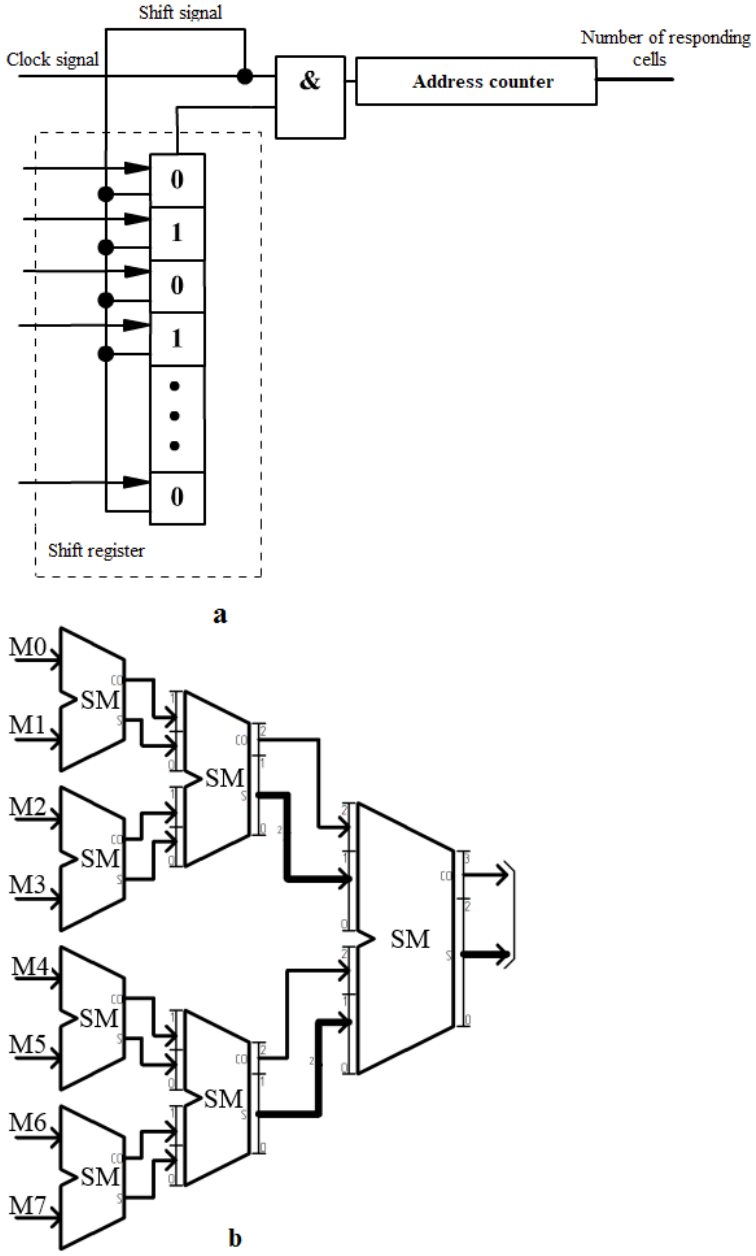

Fig 6. The unit of coincidence calculation on the basis of the shift register and the counter (a) and on the basis of adder cascades

The circuit based on the cascades of the adders (Figure 6, b) is the cascade of adders of different digits, i.e. the input binary sequence is divided into groups of two and these groups are fed to the inputs of single-bit adders. Intermediate results obtained on single-bit adders are also grouped in groups of two and are connected together with the transfer signal to the input of two bit totalizers and so on. In comparison with the previous scheme, this is a more rapid one and in principle it can count the number of cells per clock cycle, so the counting unit is implemented on the basis of adder cascades.

The correct functioning of the associative coprocessor module was verified during the execution of a number of computational experiments. Experience has shown that the use of a hardware coprocessor (when search and comparison operations were performed) can increase the computer system performance by an average of about 25% as compared to the systems that include only traditional processors. In other words, the system with the associative coprocessor shows 1.25 times higher performance than the analogs. Hence, this development should be used in practice. An expansion board with an associative coprocessor module can be used to increase the performance of database servers (by information processing on the stage of reading from memory, and bus load reduction) such as Oracle and SQL Server located on x86 machines, as well as for the processing of graphics, while the use of an expansion board should be economically justified by the criterion of "cost-effectiveness".

The structure of the associative coprocessor module based on PLIC was developed in the article. This structure which differs from the analogs: the device under consideration is executed in the form of a functionally independent block in which algorithms are implemented to perform laborious operations of data search and comparison. In the traditional organization of such

devices, these operations were performed directly by the processor. It is also necessary to note one more advantage of this development, which is connected with the implementation of the module under consideration on the modern element base (PLIC).

The considered hardware module of the associative coprocessor is implemented on PLIC of "Xilinx" company. The VHDL codes of the device were developed consisting of four modules, including the main blocks of the device under consideration: an associative coprocessor, reaction fixation memory and a multiple coincidence analyzer, PCI interface description, the conjugation of an associative coprocessor and a PCI interface.

5 Conclusions

The structure of PLIC-based associative coprocessor module is described. Based on the above description of the device blocks, the VHDL code of the associative coprocessor was developed and debugged at the functional level.

The device mentioned in the work can be physically realized in the form of an CS expansion board, connected via the PCI interface. The associative coprocessor is implemented in hardware, which allows to perform laborious searches and comparisons, thereby unloading the CPU and increasing the overall performance of the CS.

The associative coprocessor described in the article performs the following functions:

1. address record in AMU;
2. address reading from AMU;
3. the search for data equal, larger and smaller than search arguments;
4. associative reading (the reading elements greater, less or equal to an argument);
5. associative record (the record to a memory cell whose contents are larger, less or equal to an argument).

The performance of the device and individual units is verified by testing and debugging of the developed VHDL codes.

Literature:

1. Kohonen T. (1982). Associative memory devices: Trans. from English. Moscow: Mir, p.384.
2. Tanenbaum E., Bos H. (2015). Modern operating systems. St. Petersburg: Peter, p. 1120.
3. Martyshkin A.I. (2011). The study of memory subsystems with transaction buffering on mass service models. XXIst century: the results of the past and problems of the present, No. 3, pp. 124-131.
4. Martyshkin A.I. (2015). The development of a hardware buffer memory of a multiprocessor system. Fundamental research, 12(3), pp. 485-489.
5. Martyshkin A.I., Yasarevskaya O.N. (2015). Mathematical modeling of Task Managers for Multiprocessor systems on the basis of open-loop queuing networks. ARPN Journal of Engineering and Applied Sciences, 10(16), PP. 6744-6749.
6. Salnikov I.I., Babich M.Yu., Butaev M.M., Martyshkin A.I. (2016). Investigation of the memory subsystem of information systems. The International Journal of Applied Engineering Research, 11(19), PP. 9846-9849.
7. Martyshkin A.I. (2016a). Development and research of open-loop models the subsystem processor-memory of Multiprocessor systems architectures UMA, NUMA and SUMA. ARPN Journal of Engineering and Applied Sciences, 11(23), PP. 13526-13535.
8. Martyshkin A.I. (2016b). Mathematical modeling of Tasks Managers with the strategy in space with a homogeneous and heterogeneous input flow and finite queue. ARPN Journal of Engineering and Applied Sciences, 11(19), pp. 11325-11332.
9. Martyshkin A I. (2018). Basic operation principles of associate co-processor module for specialized computer systems based on programmable logical integral schemes. Journal of Fundamental and Applied Sciences, 10(6S), pp.1449-1463.
10. Zilker B.Ya., Orlov S.A. (2011). Organization of computers and systems (2nd ed.), St. Petersburg: Peter, p.688.
11. 11. Ognev I.V., Borisov V.V. (2000). Associative media. M.: Radio and Communication, p. 312.
12. 12. Kolesnikova J., Kamasheva A.V. (2017). The alienation of the rights to life and health: the institutional dimension, Astra Salvensis, Supplement No. 2, pp. 59-65.
13. 13. Villalobos Antúnez J.V., Márceles V., Ayala T. (2013). Epistemología y Ciencia: La Hermenéutica Filosófica como crítica al Método Cienctífico, Revista Electrónica de Humanidades, 16 (9), pp. 105-120