

A quantitative analysis of the performance of functional distributed architectures

S.Zaporojan

Abstract

This paper is concerned with the functional distributed architectures (FDA) performance. In this work a quantitative analysis of FDA performance has been carried out. The carried out analysis makes it possible to determine the limits for the effective usage of FDA structures. The formulae for the speedup of the computations at the FDA structures have been obtained. The formulae for determining the maximum number of executive processors which allows us to achieve the greatest computation's speedup at the given FDA structure have also been obtained. It is shown that use of the executive processors with communication autonomy allows us to introduce the additional levels of the instruction's interpretation, due to which the bus traffic is reduced. A quantitative evaluation of this fact is presented in the paper. **KEYWORDS:** functional distributed architecture, control and executive processor, cyclic interleaved computations, speedup, macrooperation, one/two-level instruction's interpretation.

1 Introduction

The purpose of this paper is to investigate the problem of the performance of FDA. We consider that the analysed FDA consists of two subsystems: control and executive [1,2]. It is supposed that the control subsystem contains a control processor (CP). The executive subsystem consists of one or more executive processors (EP). It is also considered that the communication between processors is performed via a common bus to which a system memory (SM) is connected.

The functional distributed architectures have been investigated in [1]. In this work a formal description of the FDA functioning based on the distribution between the control and executive processor of four phases of the instruction's execution is given. These phases are: fetching of the instructions and input operands from system memory, processing of the input operands on the executive processor as well as writing down the results. In conclusion the principles of FDA organisation are proposed: operational, process, and communication autonomy of the executive processor. Using the proposed principles 16 typical structures have been presented.

It is obvious that to increase the speedup of computations the processors of FDA must operate simultaneously. That is, the computations with interleaving must be used. The overlapped computations are used in a special processor architecture, called the Interleaved Array Processor (IAP) [3]. The last work shows that a greatest speedup is achieved when the cyclic computations with overlapping are organised. The similar computations have been organised at a FDA based on a control processor and a FFT processor [4]. An important problem of how many processors can be used effectively when the computations are organised with overlapping was investigated in [3,5]. In [3] a quantitative analysis of the performance of matrix multiplication by the IAP is carried out. It is supposed that the data are transmitted via the two buses. A general analysis of the performance of the FDA is presented in [5].

The contribution of this paper is to provide a quantitative analysis of the FDA performance that takes into account the characteristics of control and executive subsystems, of common bus and executed algorithms. In section 2 some basic concepts such as basic macrooperation, one-level and two-level instruction's interpretation as well as the notation are introduced. We are also presenting in this section the structure of the analysed system and an algorithm of computations based on a round robin scheduling technique. Sections 3 and 4 present a quantitative analysis of the FDA performance. The boundaries of the efficient usage of the given functional distributed architecture are determined taking into account the characteristics of control and executive processors, of common bus and executed algorithms. The application of the

communication autonomy allows us to introduce the additional levels of the instruction's interpretation, due to which the bus traffic is reduced. Therefore, it is possible to increase the computation's speedup. In section 4 a quantitative evaluation of the above fact is obtained.

2 Basic concepts

Our aim is to provide a quantitative analysis of the FDA performance when the cyclic computations with overlapping are used. Let us consider the FDA shown in Fig.1.

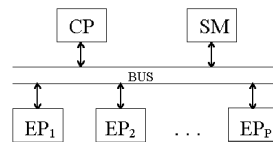


Fig.1. The structure of the analysed system

We shall suppose that the structure of the analysed system consists of one control and P executive processors. Communication between processors is performed via common bus to which a system memory is connected. We assume that there it is enough system memory for any of executed algorithm. We will also consider that the cycle time of system memory is equal to the common bus access time. Besides, we suppose that the executive processor consists of an arithmetical unit and a control one as well as of four registers. Two registers are used as a source of operands and other two are used as a destination. The

executive processor structure can also contain a communication unit and possibly a local memory.

When considering the solution of a fixed problem, we let T_{CP} denote the time required to execute the problem using only the control processor. The time required to solve the task on the FDA is T_{FDA} . Then the speedup of computations on the given FDA is defined as

$$S = \frac{T_{CP}}{T_{FDA}}.$$

There is known [6,7] the inequality, called Amdahl's Law,

$$S \leq \frac{1}{\varphi},$$

where φ is a positive "essentially serial" fraction of a computation. This inequality shows that the speedup is bounded when we increase the number of processors for a fixed problem. In practice we want to execute algorithms with various characteristics.

Let us suppose that an algorithm must be executed by the analysed FDA using $W = K + M$ instructions. It is supposed that K and M represent the number of instructions and macroinstructions that should be executed by the control and executive processors respectively. A macroinstruction we shall call one or more macrooperations with input operand vectors X_1, \dots, X_r of equal length l . If Y_1, \dots, Y_q are result vectors, then the macrooperation (MOp) is defined as follows,

$$(Y_1[j], \dots, Y_q[j]) = MOp(X_1[j], \dots, X_r[j]), \quad (1)$$

for $j = 1$ to l .

We will consider that it is possible to perform both basic macrooperations and the macrooperations of the first level. We will suppose that the executive processor is capable to execute any basic macrooperations, described as follows,

$$y_1 = MOp(x_1); \quad y_1 = MOp(x_1, x_2); \quad (y_1, y_2) = MOp(x_1, x_2). \quad (2)$$

If the basic macrooperations can be only performed on the given FDA structure then one-level interpretation of the macroinstruction is

used. If it is possible to perform the first-level macrooperations via basic macrooperations of the given FDA structure then a two-level interpretation of the macroinstruction is used. The computational power of basic macrooperations can be modified when the executive processor structure is changed. Hence, the computational power of the first-level macrooperations can be also changed.

Each macrooperation can be executed on the executive processor either consequently or using a round robin scheduling technique. Obviously one would expect a greater calculation's speedup in the second case. We assume that it is possible to organise cyclic calculations at the analysed FDA. Then we shall consider the following. First, we suppose that input operands are stored in the system memory. Any final or intermediate result produced by the executive processor must be written back into the memory. Second, we shall consider the following algorithm of computations:

1. For $j = 1$ to l , the control processor fetches the input operands from SM and forms a command packet consisting of the input operands and an MOp code for executive processor.
2. These packets are sent to the executive processors through the common bus using a round robin scheduling technique.
3. When an executive processor receives a command packet it performs the macrooperation, designated by the MOp code, with the input operands that are contained in the packet. When the macrooperation is finished, the executive processor forms an output packet containing the results.
4. The output packet containing results is sent to the system memory through the common bus by the control processor.

Mention that the input operands fetching and the results storing can be performed by the executive processor when it has communication autonomy. In this case the control processor only forms the command packets and sends these packets to the executive processors.

In order to analyse the performance of the FDA, the following notation is used:

t_A is the common bus access time. This is the time required to transfer one item of data between the system memory and an executive processors.

t_{MOV} is the transfer time of one item of data by communication unit via the common bus. It is supposed that $t_{MOV} = t_A$.

t_{CP} is the time required to perform the simple arithmetical operations or the transfer of one item of data by the control processor multiples of t_A : $t_{CP} = at_A$.

T_F is the input operand's packet fetch time in multiples of t_{CP} or t_{MOV} .

f is the number of elements in an input operand packet.

T_{COM} is total time to transfer a command packet to an executive processor.

n is the number of elements in a command packet.

N is the number of command packets.

T_S is the output result packet download time in multiples of t_{CP} or t_{MOV} .

m is the number of elements in an output result packet.

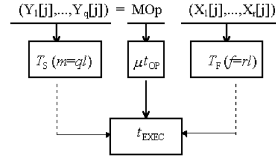
t_{OP} is the computation time of basic macrooperations on the arithmetical unit in multiples of t_A : $t_{OP} = ct_A$.

t_{EXEC} is the command packet processing time on the executive processor.

t_{ILVD} is the interleaved time. This is the time when the work of control and executive processors can be overlapped.

T_{MOP} is the computation time of basic macrooperations on the control processor.

The macrooperation execution on the executive processor may be represented as follows,



where $\mu = 1$ when the basic macrooperations are only performed. When a macroinstruction is performed using a hierarchical interpretation mechanism [2], μ represents the number of basic macrooperations that must be performed when interpreting a macroinstruction.

The number of processors that can be used effectively when the computations are organised with overlapping and using a round robin scheduling technique is determined [5] by,

$$P = \max \left\{ \left[\frac{t_{EXEC}^i}{t_{ILVD}^i} \right] \right\}, \quad (3)$$

where t_{EXEC}^i is the command packet processing time for the i -th macrooperation and t_{ILVD}^i is the overlapped time when the i -th macrooperation is performed. The braces mean rounding of the enclosed value to the smallest integer greater than the initial number.

We will consider the question of whether it is possible to attain a desirable speedup of computations at the given FDA. In particular, the structure that contains the arithmetical unit and the communication unit will be used as an executive processor. Thus, the executive processor has operational and communication autonomy. Therefore, each executive processor is capable of fetching and executing basic and first-level macrooperations from system memory and can also store the

results back into the memory. Below the boundaries of the efficient usage of the given functional distributed architecture are determined taking into account the characteristics of FDA and executed algorithms. We first consider the case when the algorithm is executed using the one-level interpretation of macroinstructions. Then, the case when the algorithm is executed using the two-level interpretation of macroinstructions is investigated.

3 Analysis of the one-level interpretation

This section will investigate the performance of computations at the given FDA when the macroinstructions are interpreted via basic macrooperations described by (2), i.e. $\mu = 1$. Obviously, the arithmetical and communication units of the executive processor structure can operate either consequently or simultaneously. The first structure represents a serial-based executive processor. In the second case we have a parallel-based executive processor. Subsection 3.1 investigates the performance of computations on the FDA with serial-based executive processors and subsection 3.2 describes the performance of computations on the FDA with parallel-based executive processors respectively.

3.1 Computations with the serial-based executive processor

Let the arithmetical and communication units of the executive processor can only operate consequently. In this case the control processor forms the command packet containing the macrooperation code and the address of input operands in the memory. Thus, we may consider that. The formed command packet is then sent via the common bus to the executive processor. When the executive processor receives the command packet, the input operands are fetched from system memory by the communication unit of this processor. After that the arithmetical unit performs the macrooperation, designated by the MOp code that is contained in the command packet. The results produced by the arithmetical unit are then written into the memory by the communica-

tion unit. After the results are stored the executive processor is capable to start the processing of the next command packet. The computations on the analysed FDA are illustrated by timing diagram shown in Fig.2. It is supposed here that the maximum number of executive processors is equal to 2.

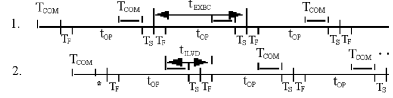


Fig.2. The timing diagram

From this diagram it can be seen that the total overlapped time consists of the time of the command packet transfer to the executive processor as well as of the result storing and input operands fetching times (initially the results do not exist therefore the symbol “*” is used at the diagram). Thus,

$$t_{ILVD} = T_{COM} + T_S + T_F, \quad (4)$$

where,

$$T_{COM} = nt_{CP}, T_S = mt_{MOV}, T_F = ft_{MOV}. \quad (5)$$

Hence,

$$t_{ILVD} = nt_{CP} + mt_{MOV} + ft_{MOV}. \quad (6)$$

From the diagram it is also seen that the command packet processing time on the executive processor is defined by,

$$t_{EXEC} = T_F + t_{OP} + T_S = ft_{MOV} + t_{OP} + mt_{MOV}. \quad (7)$$

Therefore the maximum number of executive processors, defined by (3), is given as,

$$P_{\max}^S = \max \left\{ \left[\frac{t_{EXEC}^i}{t_{ILVD}^i} \right] \right\} = \max \left\{ \left[\frac{T_F^i + t_{OP}^i + T_S^i}{T_{COM}^i + T_S^i + T_F^i} \right] \right\}, \quad (8)$$

where $1 \leq i \leq N$. From equation (8) it follows that $P_{\max}^S = 1$ if

$$t_{OP}^i \leq T_{COM}^i$$

for any i . The command packet processing time can be written in multiples of t_A as

$$t_{EXEC}^i = (f^i + c^i + m^i)t_A, \quad (9)$$

and the overlapped time,

$$t_{ILVD}^i = (n^i a + m^i + f^i)t_A. \quad (10)$$

Then (8) may be written as

$$P_{\max}^S = \max \left\{ \left[\frac{f^i + c^i + m^i}{n^i a + m^i + f^i} \right] \right\}. \quad (11)$$

When one and the same macrooperation is executed, i.e. $i = 1$, or when the macrooperations have identical $T_{COM}^i, T_S^i, T_F^i, t_{OP}^i$ times for all i , then

$$P_{\max}^S = \left[\frac{T_F + t_{OP} + T_S}{T_{COM} + T_S + T_F} \right] = \left[\frac{f + c + m}{na + m + f} \right]. \quad (12)$$

The time required to execute the algorithm by the FDA may be written as

$$T_{FDA} = T_{CP}^K + T_{ESS}, \quad (13)$$

where T_{ESS} is the time required to execute N macrooperations by the executive subsystem. This time is determined as follows:

$$T_{ESS} = T_{COM} + T_F + (N - 1)t_{ILVD} + t_{OP} + T_S = Nt_{ILVD} + t_{OP}. \quad (14)$$

The time required to execute K instructions by the control processor is

$$T_{CP}^K = Kt_{CP}. \quad (15)$$

The time required to execute the algorithm using only the control processor

$$T_{CP} = Kt_{CP} + N(T_F + T_{MOP} + T_S), \quad (16)$$

with

$$T_F = ft_{CP}, \quad T_{MOP} = zt_{CP}, \quad T_S = mt_{CP}. \quad (17)$$

Using the above presented relations the speedup of computations at the FDA with serial-based executive processors can be written as

$$S = \frac{T_{CP}}{T_{FDA}} = \frac{t_{CP}(K + N(f + z + m))}{Kt_{CP} + N(nt_{CP} + mt_{MOV} + ft_{MOV}) + t_{OP}}. \quad (18)$$

Ignoring the time t_{OP} and taking into account that $t_{CP} = at_A$ and $t_{MOV} = t_A$ we have

$$S = \frac{at_A(K + N(f + z + m))}{Kat_A + Nt_A(na + m + f)}. \quad (19)$$

Let us introduce the following notation: $K = \alpha N$, where $\alpha \geq 0$. Then the speedup may be written as

$$S = \frac{a(\alpha + f + z + m)}{a\alpha + na + m + f}. \quad (20)$$

From (20) it follows that the speedup will be achieved when

$$z > n + \frac{(1 - a)(m + f)}{a}. \quad (21)$$

Using (17) the inequality (21) may be written as

$$T_{MOP} > \left[n + \frac{(1 - a)(m + f)}{a} \right] t_{CP}. \quad (22)$$

Thus, the inequality (22) shows when the speedup of computations is available at the FDA with serial-based executive processors. The right part of this inequality may be negative what means that we should have achieved the speedup even if the computation time of basic macrooperation on the control processor had been equal to zero.

Obviously when solving a problem with $N \gg K$, i.e. when $\alpha = 0$, a boundary speedup S_b will be achieved on the given FDA. Then from (20) it follows that

$$S_b = \frac{a(f + z + m)}{na + m + f}. \quad (23)$$

For $a \rightarrow \infty$, i.e. when the control processor is very slow, from (20) it follows that

$$\lim_{a \rightarrow \infty} S = \frac{\alpha + f + z + m}{\alpha + n}. \quad (24)$$

The formula (24) allows us to determine the possible maximum speedup of fixed computation on the FDA with serial-based executive processors.

3.2 Computations with the parallel-based executive processor

This subsection investigates the performance of computations at the FDA with parallel-based executive processors. Suppose that the arithmetical and communication units of the executive processor can operate simultaneously. Then the input operands processing on the arithmetical unit and the fetching of the next input operands by the communication unit (if the next command packet has been received) can be executed simultaneously. After the current processing is finished the arithmetical unit can process the next operands and the communication unit can execute the storing of the last result into the system memory.

Fig.3 shows the timing diagram when executing the algorithm at the FDA with parallel-based executive processors. It can be seen that the total overlapped time is given as

$$t_{ILVD} = T_{COM} + T_F + T_S, \quad (25)$$

i.e. this time is identical with the overlapped time given by the formula (4). On the other hand, the command packet processing time is given as

$$t_{EXEC} = t_{OP}. \quad (26)$$

Hence the maximum number of executive processors is determined by,

$$P_{\max}^P = \max \left\{ \left[\frac{t_{OP}^i}{T_{COM}^i + T_F^i + T_S^i} \right] \right\}, \quad (27)$$

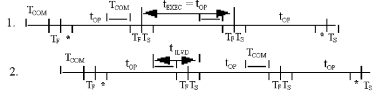


Fig. 3. The timing diagram

or, analogously to (11),

$$P_{\max}^P = \max \left\{ \left[\frac{c^i}{n^i a + f^i + m^i} \right] \right\}. \quad (28)$$

For $i = 1$, or when the macrooperations have identical $T_{COM}^i, T_S^i, T_F^i, t_{OP}^i$ times for all i , we can write

$$P_{\max}^P = \left[\frac{t_{OP}}{T_{COM} + T_F + T_S} \right] = \left[\frac{c}{na + f + m} \right]. \quad (29)$$

We would expect a reduction of the number of parallel-based executive processors to achieve the same speedup comparatively with the case when the computations are organised with the serial-based executive processors. The analysis of the diagrams shown in Fig.2 and Fig.3 as well as the comparison of the formulae (8) and (27) confirm the fact. In particular, dividing (12) and (29) we have the following theoretical estimation

$$\frac{P_{\max}^S}{P_{\max}^P} = 1 + \frac{T_F + T_S}{t_{OP}} = 1 + \frac{f + m}{c}. \quad (30)$$

Practically the number of executive processors may be identical.

From the diagram shown in Fig.3 it follows that the time required to execute N macrooperations by the executive subsystem with parallel-based executive processors is given as

$$T_{ESS} = Nt_{ILVD} + (t_{OP} - T_S) + T_S = Nt_{ILVD} + t_{OP}. \quad (31)$$

It is seen that the obtained formula is identical with (14). Therefore, the formulae of the computation's speedup obtained in previous subsection are true when executing the algorithm on the FDA with parallel-based executive processors.

4 Analysis of the two-level interpretation

We will now investigate the case when the algorithm is executed by the analysed FDA using the two-level interpretation of macroinstructions. Let us suppose that the algorithm is executed by the first-level macrooperations described by (1). Each first-level macrooperation is interpreted via the μ basic macrooperations described by (2). The control processor forms the command packet that must contain the length l of input vector X and the address of input operands in the memory as well as the length r of the first-level macrooperation. Besides, the command packet can contain the address of the result to store it back into the memory. The formed command packet containing as a minimum $n = 3$ elements is then sent to the executive processor. It interprets the first-level macrooperation code executing μ basic macrooperations. When processing the command packet the input operands are fetched from the system memory and the result produced by the executive processor is written back into the memory for each of μ basic macrooperations. The two-level computations at the FDA with serial-based executive processors are illustrated by timing diagram shown in Fig.4.

The use of the two-level interpretation allows to exclude formation and transmission of the command packets when executing each of μ basic macrooperations. Obviously the exclusion of the above mentioned operations permits us to reduce the bus traffic. Below an estimation of the mentioned fact is presented.

From the diagram in Fig.4 it is seen that the interpretation time t_{EXEC}^{IL} of the first-level macrooperation is defined as,

$$t_{EXEC}^{IL} = \mu t_{EXEC}^{OL}, \quad (32)$$

where t_{EXEC}^{OL} is the processing time of the basic macrooperation. The total overlapped time when executing the first-level macrooperation is

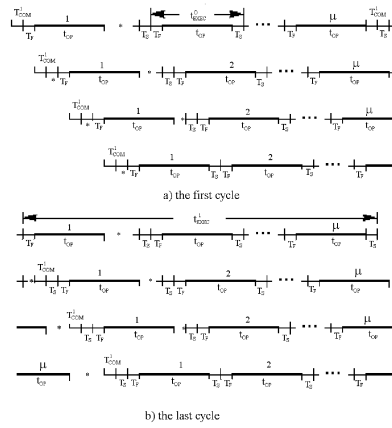


Fig. 4. The timing diagram

defined by

$$t_{ILVD}^{IL} = T_{COM}^{IL} + \mu(T_S + T_F). \quad (33)$$

Hence, in particular, the maximum number of executive processors will be calculated by

$$\begin{aligned} P_{\max} &= \left[\frac{t_{EXEC}^{IL}}{t_{ILVD}^{IL}} \right] = \left[\frac{\mu t_{EXEC}^{OL}}{T_{COM}^{IL} + \mu(T_S + T_F)} \right] \\ &= \left[\frac{t_{EXEC}^{OL}}{(T_{COM}^{IL}/\mu) + T_S + T_F} \right]. \end{aligned} \quad (34)$$

The maximum number of serial-based executive processors is determined as

$$P_{\max}^S = \left[\frac{T_F + t_{OP} + T_S}{(T_{COM}^{IL}/\mu) + T_S + T_F} \right] = \left[\frac{f + c + m}{(n^{IL}a/\mu) + m + f} \right]. \quad (35)$$

When the parallel-based executive processors are used we have $t_{EXEC}^{OL} = t_{OP}$. Therefore,

$$P_{\max}^P = \left[\frac{t_{OP}}{(T_{COM}^{IL}/\mu) + T_S + T_F} \right] = \left[\frac{c}{(n^{IL}a/\mu) + m + f} \right]. \quad (36)$$

It is easy to observe that the formulae (12) and (29) are obtained from the formulae (35) and (36) respectively in the case $\mu = 1$, i.e. when the one-level interpretation is used. It is also easy to see that equation (30) is true when the two-level interpretation is used. A possible difference of the number of the command packet elements is necessary to be taken into account only when the one-level and two-level interpretation are used.

From equation (34) it follows the increase of the maximum number of executive processors when μ is increased. That is, the use of the two-level interpretation allows to increase the implemented concurrency of the FDA. It is important to mention that when the three-level interpretation will be used then the increase of the maximum number of executive processors is very little. This assertion can be easily demonstrated using expressions similar to equations (32) and (33).

From the diagram shown in Fig.4 it follows that the time required to execute N first-level macrooperations by the executive subsystem is given as

$$T_{ESS}^{IL} = T_{COM}^{IL} + T_F + (N - 1)t_{ILD}^{IL} + \mu t_{OP} + T_S. \quad (37)$$

Taking into account equation (33) the last expression may be written as

$$T_{ESS}^{IL} = NT_{COM}^{IL} + (T_S + T_F) [\mu(N - 1) + 1] + \mu t_{OP}. \quad (38)$$

When $\mu = 1$ the formulae (14) and (31) are obtained from equation (38).

The speedup S^I of the executive subsystem based on the two-level interpretation over the executive subsystem based on the one-level interpretation is the ratio of the times required to execute the macrooperations using the one-level interpretation and the two-level interpretation as follows,

$$S^I = \frac{T_{ESS}^{OL}}{T_{ESS}^{IL}}, \quad (39)$$

where the time required to execute μN basic macrooperations using the one-level interpretation is given as

$$\begin{aligned} T_{ESS}^{OL} &= N\mu (T_{COM}^{OL} + T_S + T_F) + t_{OP} \\ &= [N\mu(n^{OL}a + m + f) + c] t_A, \end{aligned} \quad (40)$$

and the time required to execute N first-level macrooperations using the two-level interpretation is

$$\begin{aligned} T_{ESS}^{IL} &= NT_{COM}^{IL} + (T_S + T_F) [\mu(N - 1) + 1] + \mu t_{OP} \\ &= (Nn^{IL}a + (m + f) [\mu(N - 1) + 1] + \mu c) t_A. \end{aligned} \quad (41)$$

Then, we have

$$S^I = \frac{N\mu(n^{OL}a + m + f) + c}{Nn^{IL}a + (m + f) [\mu(N - 1) + 1] + \mu c}. \quad (42)$$

5 Conclusions

This paper has investigated the problem of how must the computations be organised so that to achieve a greatest speedup at the functional distributed architectures. The use of communication autonomy allows us to introduce the additional levels of instruction interpretation, due to which the bus traffic can be reduced. Therefore, it is possible to increase the number of executive processors. It is mentioned that when using the three-level interpretation the reduction of the bus traffic is very little. Therefore, from this point of view, the computations should be organised using the one-level and two-level interpretation. Using equation (42) an estimation of the speedup of the executive subsystem based on the two-level interpretation over the executive subsystem based on the one-level interpretation can be obtained. The executive subsystem uses P_{\max} processors in both cases.

References

- [1] S.Zaporojan. Organizarea si proiectarea unitatilor de calcul cu arhitectura functional distribuita. Autoreferatul tezei de doctorat. Universitatea Tehnica a Moldovei, Chisinau, 1995.
- [2] S.Zaporojan, A.Ursu. A model of functional distributed architectures with local programmability. Proc. of the int. conf. on technical informatics, Timisoara, Nov.16-19, 1994. V.4, pp.163–172.
- [3] G.Wolf and J.R.Jump. Matrix multiplication in an interleaved array processing architecture. The 12th annual int. symposium on comput. architecture, 1985, pp.11–17.
- [4] V.E.Sazanov, S.I.Zaporojan, B.B.Czerniatiev, I.N.Czebykin. Realizatsia protsessora BPF na mikroprotsessorah (A FFT processing unit with microprocessors). V kn.: Proiektirovanie, kontroli i diagnostika mikroprotsessornyh sistem. Izd-vo 1985, pp.86–89 (Russian).

- [5] S.Zaporojan, V.Gasca and A.Ursu. A general analysis of performance of functional distributed architectures. The int. symposium on systems theory SINTES 8. Section computer science and engineering, Craiova, June 6-7, 1996, pp.297–302.
- [6] D.J.Evans. Parallel architectures and algorithms. Preprints of the 4th int. symposium on automatic control and computer science, Iasi, October 29-30, 1993.
- [7] M.Cosnard, D.Trystram. Algorithmes et architectures paralleles. Inter Editions, Paris, 1993.

S.Zaporojan,
Computer Science Department,
Technical University of Moldova,
Stefan cel Mare 168,
2012, Chisinau, Republic of Moldova,
phone: (+373.2) 497016

Received 13 January, 1996