

## Оригинальная статья / Original article

<https://doi.org/10.21869/2223-1560-2019-23-5-161-174>

## Алгоритм и устройство планирования расписания загрузки процессоров в мультипроцессорных системах

Д.Б. Борзов<sup>1</sup> ✉, В.С. Титов<sup>1</sup>, Р.Г. Басов<sup>1</sup>

<sup>1</sup> ФГБОУ ВО «Юго-Западный государственный университет»  
ул. 50 лет Октября, 94, г. Курск, 305040, Российская Федерация

✉ e-mail: borzovdb@kursknet.ru

### Резюме

**Цель исследования** заключается в разработке алгоритма и соответствующего устройства планирования расписания загрузки процессоров, обеспечивающего повышение производительности мультипроцессорных систем и снижение общей величины коммуникационной задержки.

**Методы.** Статья посвящена мультипроцессорным системам. Затрагивается вопрос составления плана загрузки процессоров в них. Предполагается применение так называемых систем реального времени, когда реакция на внештатную ситуацию должна быть «мгновенной». Например, это может быть отказ системы слежения в кабине пилота самолета, либо сбой датчика прицеливания, либо неполадки какого-либо характера на финансовых рынках страны (мира, континента) и т.п. Существующие методы реализации данной задачи используют, как правило, программные подходы, приводящие к длительному времени решения, вследствие чего снижается коэффициент готовности системы и ее производительность. Они не позволяют получить необходимую производительность и быстродействие. Поэтому в статье предложен алгоритм планирования расписания загрузки процессоров в мультипроцессорных системах. Отличительной особенностью работы является направленность на аппаратную реализацию. При этом применяются специально введенные в предложенном алгоритме матрицы времени, порядка и очередности.

В работе предлагается структурная и функциональная схема специализированного устройства планирования загрузки процессоров мультипроцессорной системы. Данное устройство возможно применять как специализированный дополнительный акселератор, который подключается к параллельному порту вычислительной машины специализированной мультипроцессорной системы. В данном случае разработанный акселератор выполняет все функции вычислений разработанного алгоритма. Это позволит снизить нагрузку на всю вычислительную систему и уменьшить этим суммарную нагрузку, повышая производительность. Так же предложена структурная и функциональная схема устройства планирования расписания загрузки. С его помощью возможно построить зависимости изменения соответствующей временной и аппаратной сложности.

**Результаты.** Анализируя зависимости изменения аппаратной сложности устройства от количества процессоров, можно сделать следующие выводы. Время работы устройства имеет экспоненциальную зависимость при росте количества процессоров. Так как имеет место экспоненциальный рост времени числа вариантов загрузки процессоров, то можно сделать вывод о преимуществе аппаратной реализации алгоритма планирования.

**Заключение.** Представленный алгоритм и устройство планирования загрузки процессоров позволяет сделать вывод о потенциальной возможности повышения скорости составления плана загрузки. Это помогает уменьшить суммарное значение коммуникационной задержки и одновременного увеличения производительности мультипроцессорных систем.

**Ключевые слова:** алгоритм; план; загрузка; процессор; мультимикропроцессорная система; расписание.

**Конфликт интересов:** Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

**Для цитирования:** Борзов Д.Б., Титов В.С., Басов Р.Г. Алгоритм и устройство планирования расписания загрузки процессоров в мультимикропроцессорных системах // Известия Юго-Западного государственного университета. 2019; 23(5): 161-174. <https://doi.org/10.21869/2223-1560-2019-23-5-161-174>.

Статья поступила в редакцию 03.06.2019

Статья подписана в печать 17.07.2019

Статья опубликована 25.10.2019

## Algorithm and Planning Device for CPU Loading in Multiprocessor Systems

Dmitriy B. Borzov <sup>1</sup> ✉, Vitaliy S. Titov <sup>1</sup>, Rodion G. Basov <sup>1</sup>

<sup>1</sup> Southwest State University  
50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

✉ e-mail: borzovdb@kursknet.ru

### Резюме

**Purpose of research.** The task of compiling a plan for loading processors in multiprocessor systems of critical purpose (tracking, observation, aiming, atomic and financial systems) is discussed. Modern algorithms implement the problem mainly programmatically and are unacceptable for the considered class of systems because of the long solution time, which leads to a decrease in its availability and a corresponding decrease in performance. Existing approaches, as a rule, short-term planning are often focused on the rapid response of ongoing processes, which also does not allow to achieve the required performance. As a result, the article proposes an algorithm for scheduling the loading of processors in multiprocessor systems, focused on hardware implementation, using the additionally introduced matrices of time, order and order.

**Methods.** The article proposes a block diagram and a functional diagram of a specialized device for planning the loading of processors of a multiprocessor system. A distinctive feature is the use of the developed device as an additional accelerator connected to the parallel port of the computer. The proposed device in this case performs all the calculated functions of the proposed algorithm, thereby reducing the main load on the computing system, thereby reducing the total load. A structural and functional diagram of the device for scheduling a loading schedule is given, which allows the construction of corresponding time diagrams of temporary and hardware complexity.

**Results.** From the analysis of the graphs of the dependence of the increase in the hardware complexity of the device on the number of processors, an exponential increase in the operating time of the device, depending on the number of processors, follows. Due to the exponential growth in the number of processor loading options, the hardware implementation of the scheduling algorithm shows an advantage.

**Conclusion.** With the help of the proposed algorithm and device for planning processor loading, it is possible to increase the speed of load compilation, decrease the total amount of communication delay and increase the total performance of multiprocessor systems.

**Keywords:** algorithm; plan; load; processor; multiprocessing system; schedule.

**Conflict of interest.** The Author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

**For citation:** Borzov D. B., Titov V. S., Basov R.G. Algorithm and Planning Device for CPU Loading in Multiprocessor Systems. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2019, 23(5): 161-174 (In Russ.). [https://doi.org/ 10.21869/2223-1560-2019-23-5-161-174](https://doi.org/10.21869/2223-1560-2019-23-5-161-174).

Received 03.06.2019

Accepted 17.07.2019

Published 25.10.2019

\*\*\*

## Введение

Быстродействие и трудоемкость (программ, подпрограмм, алгоритмов) потенциально могут увеличиваться [1]. Особенно это относится к «сложным» системам, к которым можно отнести системы, за которыми закреплены функции оперативной реакции на исполнительное устройство со стороны ведущей ЭВМ (хост-процессора). Например, это может быть кабина пилота самолета и датчик прицеливания (слежения) за целью. В них требуется постоянный контроль, а также наблюдение за уровнем быстродействия и производительности всей мультимикропроцессорной системы [2-4]. Данный вопрос решается в основном программным путем. Такой подход не позволяет достигать требуемого для критических систем уровня производительности [5-8]. Также требуется постоянный контроль за присутствием задач и их очередей на процессорах мультимикропроцессорных систем.

Важным вопросом для систем реального времени и критических систем является нахождение задач и их очередей на отдельных процессорах мультимикропроцессорных систем. Дополнительным требованием является отсутствие взаи-

модействия назначаемых задач по данным и управлению [8-10].

Эти вопросы могут быть решены на основе методов теорий расписаний, очередей и планирования загрузки процессоров [9-11].

## Материалы и методы

Пусть требуется выполнить множество  $N = \{1, 2, \dots, n\}$  заданий (данных, алгоритмов, подпрограмм)  $i \in N$  и  $j \in N$  ( $i \neq j$ ). Параметру  $j$  поставим в соответствие значения  $p$ ,  $r$  и  $D$ . Значение  $p$  означает время ( $p_j > 0$ ) исполнения операции  $j$  и ее длительность. При этом  $j$  задает время ( $r_j \geq 0$ ) поступления на процессор  $p_j$  и максимальную длительность выполнения ( $D_j > 0$ ). В данном случае ограничением является отсутствие внутреннего или внешнего прерывания.

Тогда, для любого требования  $j \in N$  необходимо определить как время  $S_j$  начала выполнения задания, так и время  $C_j$  окончания его выполнения. Тогда  $C_j = S_j + p_j$ . Предположим,  $S_i$  – это время начала другого задания ( $i \in N$ ,

$i \neq j$ ). Получаем, что если  $S_j < S_i$ , то, следовательно,  $S_j + p_j \leq S_i$ . В задачах планирования, как и в задачах теории расписаний время будем определять в условных единицах, таких, как секунды, минуты и т.д.

Очевидно, что в системах критического характера, как и в реальных мультипроцессорных системах расписание обязано быть оптимальным. Следовательно,  $D_j = +\infty$  при  $j = 1, 2, \dots, n$ . Предположим, что  $t_j$  – это время исполнения некоторого задания  $j$ .

Тогда, требуется построить расписание такое, что:

$$\sum_{j=1}^n t_j \rightarrow \min. \quad (1)$$

Допустим  $c_{ij} = |\tau_i - \tau_j|$ , где  $c_{ij} \geq 0$  является фактической разницей во времени исполнения операции  $n_i$  и  $n_j$ . Назовем эту разницу расстоянием. Тогда матрица расстояний будет выглядеть как  $C = (c_{ij})_{n \times n}$ .

Непересекающиеся подмножества множества  $N$  отвечают условию:

$$\bigcup_{i=1}^k S_i = N,$$

где  $k$  – это требование, предъявляемое к заданию  $N$ .

Разобьем множество  $N$  на непересекающиеся подмножества  $S_1, S_2, \dots, S_k$   $k \geq 2$  так, что

$$S_i \cap S_j = \emptyset, i \neq j, i = j = 1, 2, \dots, k,$$

где  $0 < n_{\min} \leq |S_i| \leq n_{\max}$ ,  $i = 1, 2, \dots, k$ .

Число подмножеств  $k$  ограничено условиями:

$$k \times n_{\min} \leq n \leq k \times n_{\max}.$$

Для всех разбиений  $S = (S_1, S_2, \dots, S_k)$  и для оценки качества разбиения справедливо значение функции  $\varphi(S) = \varphi(S_1, S_2, \dots, S_k)$ . Необходимо найти разбиение  $S^0 = S_1^0, S_2^0, \dots, S_k^0$  так, что

$$\varphi(S^0) = \min_s \varphi(s). \quad (2)$$

Функция  $g_p(S_i) \in S_i$  описывает различие между данными для распределения, а  $\varphi_p(S)$  при фиксированном значении  $p$  будет равна  $\varphi_p(S_1, S_2, \dots, S_k) = \max_{1 \leq j \leq k} g_p(S_j)$ .

Дополнительно введем матрицу времени Time:

$$\text{Time} = \|\text{Time}_{ij}\|, \quad (3)$$

где  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ ,  $m = n$ . В Time, в столбцах располагают процессоры  $P_{ij}$  мультипроцессорной критической системы, а строкам соответствуют номера задач, расположенные по возрастанию соответствующих номеров  $(1, 2, \dots, m)$ . Тогда, на пересечении строк и столбцов представляются предполагаемые времена выполнения каждой задачи  $n$ .

Ситуацию выполнения (невыполнения) заданий на множестве задач  $\{N_{nm}\}$  предлагается сохранять в матрице порядка  $exes$ :

$$\text{Exes} = \|\text{Exes}_{ij}\|, \quad (4)$$

где  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ ,  $m = n$ .

В матрице порядка  $E_{hes}$  вершины столбцов обозначают процессоры  $P_{ij}$  мультипроцессорной системы, а в строках отложены задачи, подлежащие выполнению в порядке возрастания номеров. Значение «Единица» ставится в том случае, если задача выполнена.

Отслеживание очереди выполнения задач предлагается хранить в соответствующей матрице очередности  $Q_i$  ( $i = \overline{1, m}$ ):

$$Q = \|Q_{ij}\|, \quad (5)$$

где  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ ,  $m = n$ . В  $Q$  показана очередь выполнения задач  $N_i$ . В соответствующих столбцах хранятся процессоры  $P_{ij}$  мультипроцессорной системы, а в строках – задачи, назначенные для выполнения. Порядок расположения номеров процессоров предлагается по неубыванию.

В статье представляется метод конвейерного планирования загрузки процессоров в мультипроцессорных системах. Идея метода основана на использовании (1), (2) с дополнительным применением (3)-(5). Задачи (подзадачи, программы, алгоритмы, файлы и т.д.) предполагается планировать для выполнения на процессоры мультипроцессорной системы конвейерно. Предполагается отсутствие учета связей с другими операторами, ожидающими выполнения.

Метод планирования загрузки процессоров в мультипроцессорных системах определяется этапами [12,13]:

1. Постановка задачи в очередь загрузки процессора.

2. Анализ выполнения задач в процессорах мультипроцессорной системы или отсутствие очереди их загрузки.

3. Если  $\|E_{hes}\|=1$ , то план загрузки процессоров составлен, иначе п. 1.

На время и порядок выполнения операций процессорами накладываются следующие ограничения:

– введено антирефлексивное, антисимметричное и транзитивное отношение следования операций  $v \subseteq Z \times Z$ , определяющее допустимый порядок реализации операций;

– потребуем, чтобы выполнялось следующее условие:  $\forall z_a \in Z(p_i)$ ,  $z_b \in Z(p_i)$ ,  $a \neq b: z_a \vee z_b \vee z_b \vee z_a$  (любая пара операций, назначенных на процессор  $p_i$ , выполняется в заданном порядке  $v$ ).

Общая длительность всех операций

$$\text{вычисляется по формуле } t_{ij} = \sum_{j=1}^{C_j} t_{ij}.$$

Первоначально, независимые по данным и управлению операторы, подлежащие выполнению в мультипроцессорной системе, передаются на процессоры для дальнейшего составления плана загрузки.

На следующих шагах проверяется наличие единиц в матрице  $E_{hes}$ , матрице  $Q$ . При этом, анализируется матрица  $E_{hes}$ : если ее элементы равны единице, то процесс составления плана завершается. Если нет, то выполняется возврат к первому шагу для повторного планирования.

## Результаты и их обсуждение

На основе предложенного подхода разработан алгоритм планирования загрузки процессоров в мультипроцессорных системах.

$\text{ProcNum}, \text{ProcNumQ} = 1.$

Если  $\text{Time}(\text{ProcNum}, P_j) = 1$ , то п.3, иначе п. 5.

Если  $\text{Exes}(\text{ProcNum}, P_j) = 0$ , то п. 4, иначе п. 13

$\text{Temp} = 1.$

Если  $Q(\text{ProcNumQ}, \text{Temp}) = 1$ , то п. 6, иначе п. 7.

$\text{Temp} = \text{Temp} + 1$  и п. 5.

$Q(\text{ProcNumQ}, \text{Temp}) = Z_i.$

$\text{Time}(\text{ProcNumQ}, P_j) = Z_i.$

$\text{Temp} = 1.$

$\text{ProcNumQ} = 1.$

$j = j + 1.$

Если  $P_j > nm$ , то конец алгоритма,

иначе п. 2.

$\text{ProcNum} = \text{ProcNum} + 1.$

Если  $\text{ProcNum}_j > n_{nm}$ , то конец алгоритма, иначе п. 3.

Первоначально анализируется наличие в матрице Time заданий. В случае их присутствия, на следующем шаге проверяется матрица Exes и если  $\text{Exes} \neq \emptyset$ , то присутствует задача для постановки в очередь. Для этого в Q выполняется поиск свободной ячейки. Задача ставится в очередь загрузки процессоров. Одновременно выполняется фиксация этого факта в Time. Таким образом, циклически анализируются все строки матрицы задач Time.

На основе предложенного метода и алгоритма планирования загрузки процессоров предложена структурная схема соответствующего устройства (рис. 1) [19,20].

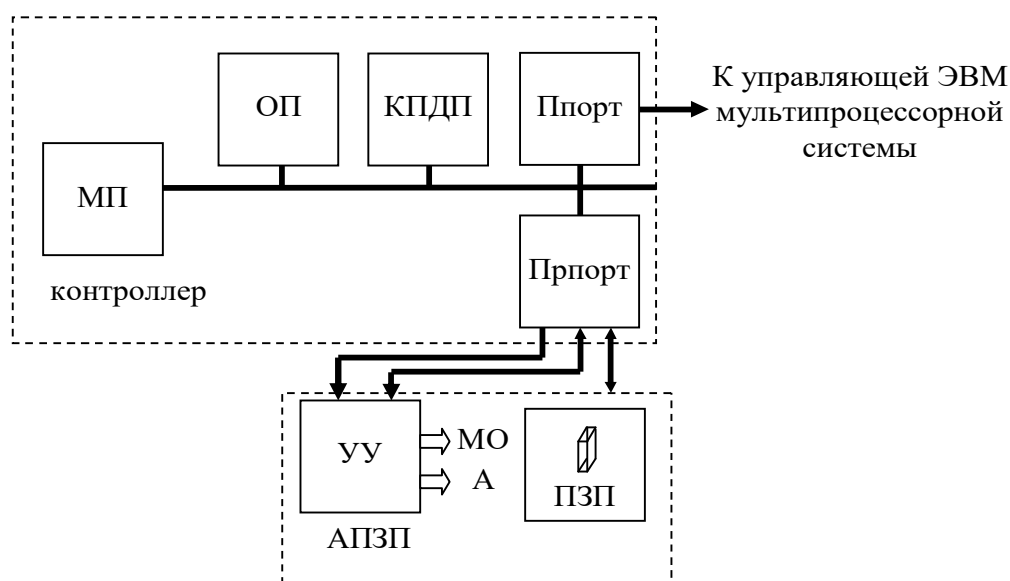


Рис. 1. Структурная схема устройства планирования загрузки процессоров

Fig. 1. Block diagram of CPU planner loading device

Для элементов, представленных на рис. 1, приняты сокращения: блок акселератора планирования загрузки процессоров (АПЗП), блок планирования загрузки процессоров (ПЗП), блок устройства управления (УУ), микропроцессор (МП), основная память (ОП), блок контроллера прямого доступа в память (КПДП), последовательный порт (Ппорт), параллельный порт (Прпорт), микрооперация (МО), адрес ячейки памяти (А).

Согласно предложенному методу и алгоритму планирования загрузки процессоров устройство выполняет составление плана загрузки по предложенному алгоритму. Соответственно блок ПЗП включает в себя устройство, подключенное к параллельному порту ЭВМ.

КПДП процессора извлекает МО предложенного алгоритма из ОП процессора и передает их для исполнения в АПЗП. АПЗП состоит из следующих функциональных блоков: блок плани-

рования загрузки процессов (ПЗП) и устройство управления (УУ). Данные с Прпорт поступают в блок специализированного мультимплексора АПЗП.

После окончания составления плана загрузки через параллельный порт происходит передача результата работы блока АПЗП в контроллер. Далее, по необходимости, результат запоминается в памяти ОП либо через последовательный порт Ппорт передается к управляющей ЭВМ мультипроцессорной системы. После этого, управляющая ЭВМ принимает решение о дальнейших действиях мультипроцессорной системы.

Структурная схема устройства планирования загрузки процессоров в мультипроцессорных системах представлена на рис. 2 [19,20].

На рис. 2 основные блоки устройства означают: УВП – устройство выявления параллелизма; П – процессор; ОП – оперативная память; TIME – матрица времени; EXEC – матрица порядка; Q – матрица очередности.

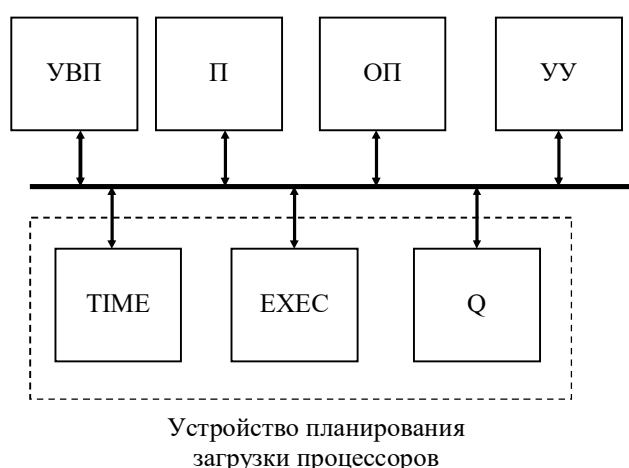


Рис. 2. Структурная схема устройства планирования загрузки процессоров

Fig. 2. Block diagram of CPU planner loading device

Предложенное устройство функционирует следующим образом: по команде процессора П через устройство управления УУ выдается команда устройству планирования выполнить процедуру составления плана загрузки с помощью предложенного в статье алго-

ритма. Для этого используются матрицы TIME, Eхес и Q.

В соответствии со структурной организацией устройства составления плана загрузки процессоров (см. рис. 1,2) разработана его функциональная организация (рис. 3) [19,20].

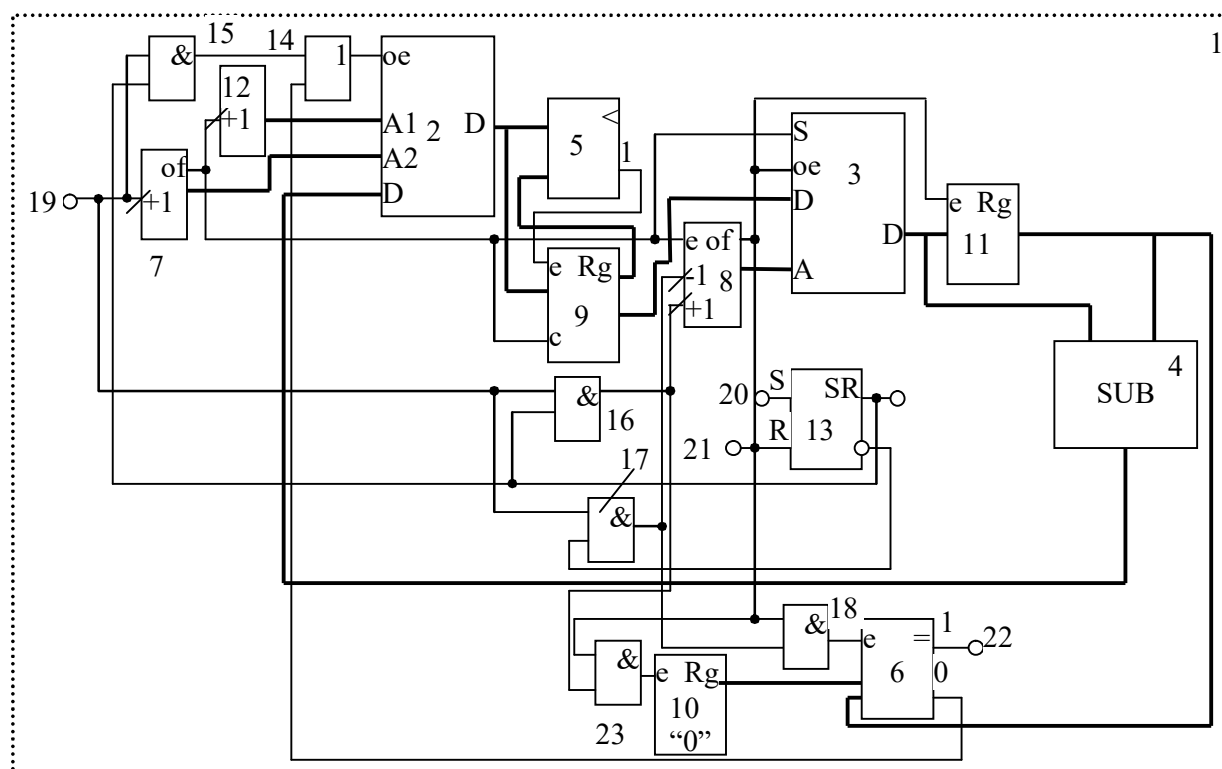


Рис. 3. Устройство планирования загрузки процессоров

Fig. 3. CPU planner loading device

На рис. 3 блок 2 моделирует ОЗУ, необходимое для моделирования очередей, каждая из которых содержит наборы заданий, предназначенных для выполнения на процессорах мультипроцессорной системы. К примеру, ситуация  $\begin{matrix} p_1 & p_2 & p_3 & p_4 \\ A_1 & A_1 & A_1 & A_4 \end{matrix}$ , где  $p_i$  означает процессор, а  $A_i$  — означает, что на процессор  $p_i$  назначено задание  $A_i$ ; промежуточное ОЗУ (блок 3 на рис. 3) моде-

лирует временное содержание очередей заданий; вычитатель 4 и элемент сравнения 5 необходимы для накопления суммарного минимума; элемент 6 сравнения поиска минимального значения загрузки необходим для поиска минимального кода, а соответствующий счетчик 7 адреса — для накопления адресов памяти ОЗУ 2; реверсивный счетчик 8 содержит в себе адреса ячеек ОЗУ 3; первый 9 регистр минимума предна-



значен для хранения текущего минимального значения кода загрузки; регистр 10 хранит в себе код нуля; второй регистр минимума 11 накапливает минимальный код загрузки; счетчик 12 накапливает коды очередей загрузки. Например, распределение заданий

$$\begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \\ A_1 & A_2 & A_3 & A_4 \\ A_5 & A_6 & A_7 & A_8 \end{array}$$

означает, что на первом этапе будут выполняться параллельно задания  $A_1 \ A_2 \ A_3 \ A_4$ , а на втором этапе –  $A_5 \ A_6 \ A_7 \ A_8$ ; триггер 13 выбирает режим работы предложенного устройства.

При подсчете временных характеристик предложенного устройства было учтено быстроедействие его внутренних модулей согласно схмотехнической серии 1533. Проводился анализ времени поступления сигнала с входа на выход каждого схмотехнического элемента схемы. Выполнено суммирование всех полученных временных параметров устройства и сопоставлено с ростом возможного количества процессоров мультипроцессорной системы. В результате, получена зависимость изменения скорости его работы от увеличения внутренних процессорных модулей (рис. 4) [19,20].

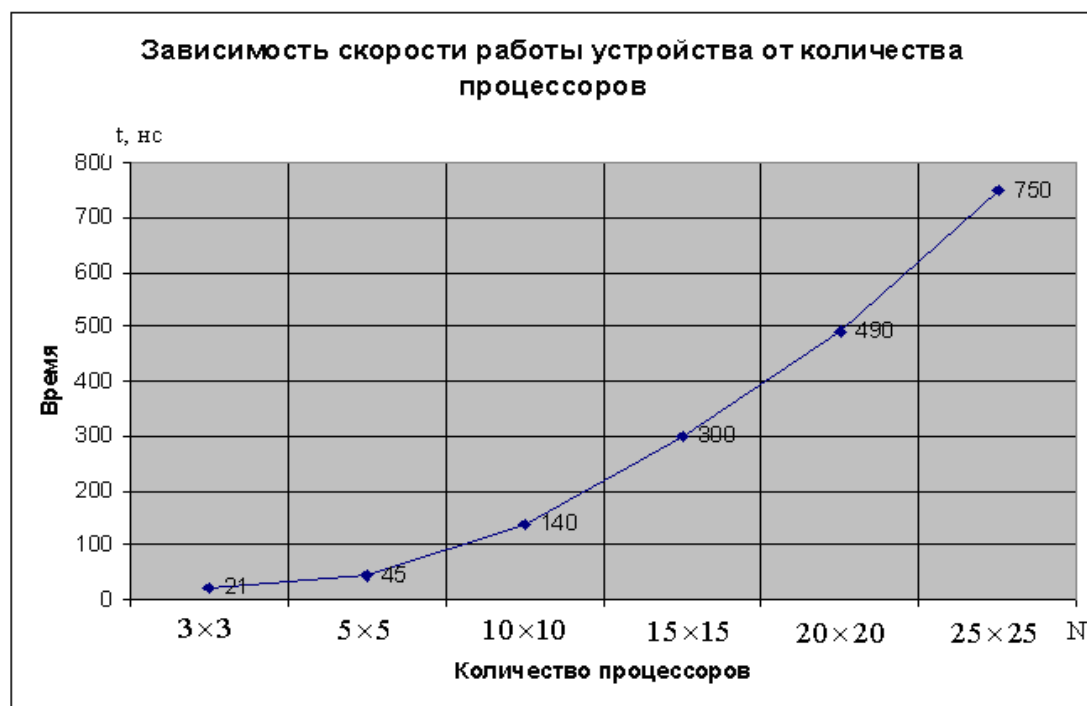


Рис. 4. Зависимость изменения времени работы устройства от количества процессоров

Fig. 4. Device's working time dependence on CPU quantity

Из анализа графика (рис. 4) следует, что скорость работы устройства планирования загрузки процессоров уве-

личивается экспоненциально с ростом количества процессоров мультипроцессорной системы, т.е. с ростом размера очереди загрузки.

При оценке аппаратной сложности устройства планирования загрузки процессоров использовалось представление его элементов (см. рис. 3) в виде эквивалентных вентилях (элемент И) [20].

Зависимость роста аппаратной сложности в зависимости от объема решаемой задачи.

В результате можно построить зависимость роста аппаратной сложности устройства с учетом увеличения внут-

реннего количества процессоров мультипроцессорной системы (рис. 5).

Из анализа зависимостей, представленных на рис. 5, можно сделать вывод, что для работы устройства в случае тридцати процессоров требуется 960 эквивалентных вентилях. В случае роста количества обрабатываемых операторов (размера задачи) прогнозируется приемлемый рост скорости работы устройства и аппаратных затрат.

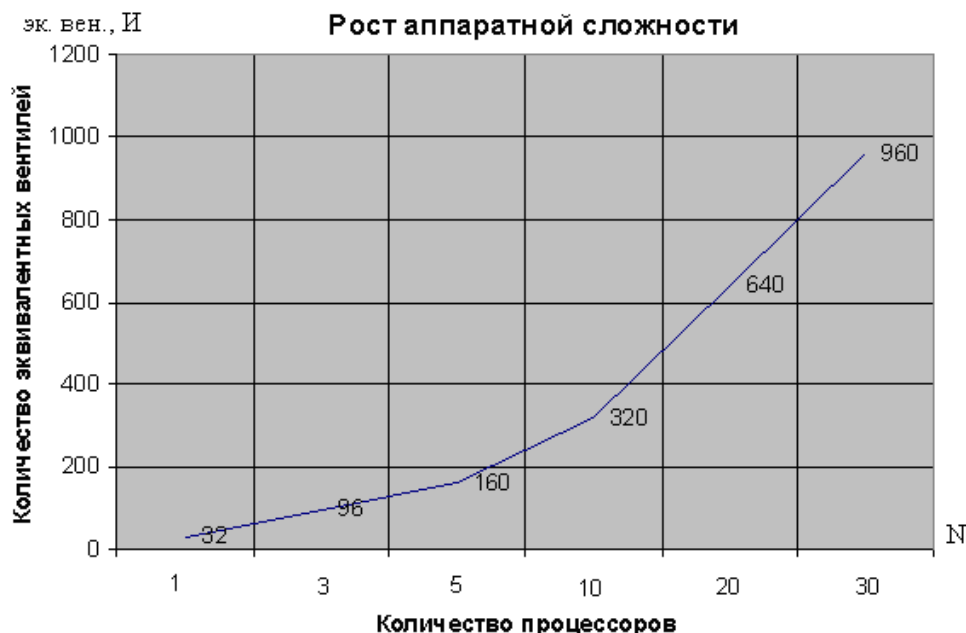


Рис. 5. График зависимости роста аппаратной сложности устройств от количества процессоров

Fig. 5. Graph of hardware complexity growth of devices on CPU quantity

Анализ графиков (см. рис. 4,5) показывает, что при составлении плана топологии для операторов программ малого порядка (менее 500) предложенный метод при сравнении с аналогичной программной реализацией не демонстрирует выигрыша.

При прогнозировании предложенных зависимостей (см. рис. 4, 5) можно

сделать вывод о преимуществе аппаратной реализации в связи с увеличением количества вариантов назначения очередного оператора на процессорные модули мультипроцессорной системы.

## Выводы

В результате сделан вывод о возможности увеличения скорости реше-

ния задачи составления плана загрузки, уменьшения общей величины коммуникационной задержки и повышения суммарной производительности вычислительных систем с помощью предложен-

ного алгоритма и устройства планирования загрузки процессоров в мультипроцессорных системах критического назначения.

### Список литературы

1. Гергель В.П. Теория и практика параллельных вычислений. М.: Бином, 2007.
2. Воеводин В.В. Вычислительная математика и структура алгоритмов. М.: МГУ, 2006.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ– Петербург, 2002. 608 с.
4. Богданов А.В., Станкова Е.Н., Мареев В.В., Корхов В.В. Архитектуры и топологии многопроцессорных вычислительных систем. 2-е изд., М., 2016. 135 с.
5. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники. М.: Бином, 2009. 357 с.
6. Yu-Kwong Kwok and Lap-Sun Cheung. A new fuzzy-decision based load balancing system for distributed object computing // Journal of Parallel and Distributed Computing. 2004. № 64. P. 238-253.
7. Zhang L., Wong T.N. Solving integrated process planning and scheduling problem with constructive meta-heuristics. Inf. Sci. 2016, 340–341, 1–16.
8. Zhang S., Wong T.N. Integrated process planning and scheduling: An enhanced ant colony optimization heuristic with parameter tuning // J. Intell. Manuf. 2014. № 29. С. 1–17.
9. Борзов Д.Б., Ткачев П.Ю. Метод распараллеливания циклов со счетчиком // Известия вузов. Приборостроение. 2015. Т. 58. №2. С. 104–108.
10. Математическая модель выявления независимых параллельных участков последовательных программ / Д.Б. Борзов, С.А. Дюбрюкс, В.С. Титов, С.В. Прилуцкий // Нейрокомпьютеры: разработка, применение. 2009. №12. С. 37-41.
11. Борзов Д.Б., Борисенко Ю.В., Сизов А.С. Метод и аппаратно-ориентированный алгоритм перераспределения подпрограмм в мультикомпьютерах при отказе процессоров и связей между ними // Телекоммуникации. 2013. №11. С. 45-48.
12. Таненбаум Э. Архитектура компьютеров. СПб.: Питер, 2007. 848 с.
13. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем. СПб.: Питер, 2006. 668 с.
14. Гусева А.И. Вычислительные системы, сети и телекоммуникации. М.: Academia, 2016. 640 с.

15. Андон Ф. И., Кукса А. И., Поляченко Б. Е. Об оптимальном планировании процесса обработки на ЭВМ взаимосвязанных задач // Кибернетика. 1980. Я. 3. С. 51–53.
16. Blazewicz J., Ecker K., Pesch E., Schmidt G., Weglarz J. Handbook on Scheduling: From Theory to Applications. Series: International Handbooks on Information Systems, Springer. Berlin, Germany, 2007, 647 p.
17. Борзов Д.Б., Масюков И.И. Планирование загрузки процессоров в мультипроцессорных системах критического назначения // Известия Юго-Западного государственного университета. 2018. Т. 22. №6 (81). С. 168-175. <https://doi.org/10.21869/2223-1560-2018-22-6-168-175>.
18. Борзов Д.Б., Басов Р.Г. Алгоритм планирования загрузки процессоров в мультипроцессорных системах // Машиностроение и техносфера XXI века: сборник трудов XXV Международной научно-технической конференции. Донецк, 2018. Т.1. С. 37-39.
19. Кулачев А.П. Методы и средства анализа данных в среде Windows. STADIA 6.0. М.: Информатика и компьютеры, 2006. 270 с.
20. Лобанов В.И. Азбука разработчика цифровых устройств. М.: Горячая линия - Телеком, 2001. 192 с.

## References

1. Gergel V.P. *Teoriya i praktika parallel'nykh vychislenii* [Theory and practice of parallel computing]. Moscow, 2007 (In Russ.).
2. Voevodin V.V. *Vychislitel'naya matematika i struktura algoritmov* [Computational Mathematics and Algorithm Structure]. Moscow, MGU Publ., 2006 (In Russ.).
3. Voevodin V.V., Voevodin V.I. *Parallel'nye vychisleniya* [Parallel computations]. S.-Petersburg, BHV–Petersburg Publ., 2002, 608 p. (In Russ.).
4. Bogdanov A.V., Stenkova E.N., Mareev V.V., Korkhov V.V. *Arkhitektury i topologii mnogoprotsessornykh vychislitel'nykh sistem* [Architecture and topology of multiprocessor computing systems]. Moscow, 2016, 135 p. (In Russ.).
5. Novikov Yu.V., Skorobogatov P.K. *Osnovy mikroprotsessornoj tekhniki* [Basics of microprocessor technology]. Moscow, 2008, 357 p. (In Russ.).
6. Yu-Kwong Kwok and Lap-Sun Cheung. A new fuzzy-decision-based balancing system for distributed object computing. *Journal of Parallel and Distributed Computing*, 2004, no. 64, pp. 238-253.
7. Zhang L., Wong T.N. Solving integrated process planning and scheduling problem with constructive meta-heuristics. *Inf. Sci.* 2016, 340–341, pp. 1–16.

8. Zhang S., Wong T.N. Integrated process planning and scheduling: An enhanced ant colony optimization heuristic with parameter tuning. *J. Intell. Manuf.* 2014, no. 29, pp. 1–17.
9. Borzov D. B., Tkachev P. Yu. Metod rasparallelivaniya tsiklov so schetchikom [Method of parallelization of cycles with a counter]. *Izvestiya vuzov. Priborostroenie = Journal of Instrument Engineering*, 2015, vol. 58, no. 2, pp. 104-108 (In Russ.).
10. Borzov D. B., Dubrux S. A., Titov V. S., Prilutsky S. V. Matematicheskaya model' vyyavleniya nezavisimyykh parallel'nykh uchastkov po-sledovatel'nykh programm [Mathematical model of identification of independent parallel sections of investigative programs]. *Neirokomp'yutery: razrabotka, primenenie = Neurocomputers: development, application*, 2009, no. 12, pp. 37-41 (In Russ.).
11. Borzov D. B., Borisenko Yu. V., Sizov A. S. Metod i apparatno-orientirovannyi algoritm pererazmeshcheniya podprogramm v mul'tikomp'yuterakh pri otkaze protsessorov i svyazei mezhdu nimi [Method and apparatus-oriented algorithm of subroutines re-placement in multicomputers at failure of processors and connections between them]. *Telekommunikatsii = Telecommunications*, 2013, no. 11, pp. 45-48 (In Russ.).
12. Tanenbaum E. *Arkhitetura komp'yutеров* [Computer architecture] . S.-Petersburg, Peter Publ., 2007, 848 p. (In Russ.).
13. Tsilker B.Ya., Orlov S.A. *Organizatsiya EVM i sistem* [Organization of computers and systems]. S.-Petersburg, Peter Publ., 2006. 668 p. (In Russ.).
14. Guseva A.I. *Vychislitel'nye sistemy, seti i telekommunikatsii* [Computing Systems, Networks and Telecommunications]. Moscow, Academia Publ., 2016, 640 p. (In Russ.).
15. Andon F. I., Kuksa A. I., Polyachenko B.E. Ob optimal'nom planirovanii protsessa obrabotki na EVM vzaimosvyaannykh zadach [On optimal planning of the process of processing on a computer of interrelated tasks]. *Kibernetika = Cybernetics*, 1980, Ya. 3, pp. 51–53 (In Russ.).
16. Blazewicz J., Ecker K., Pesch E., Schmidt G., Weglarz J. *Handbook on Scheduling: From Theory to Applications*. Series: International Handbooks on Information Systems. Springer. Berlin, Germany, 2007, 647 p.
17. Borzov D.B., Masyukov I.I. Planirovanie zagruzki protsessorov v mul'tiprotsessornykh sistemakh kriticheskogo naznacheniya [Scheduling CPU utilization in critical multiprocessor systems]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*, 2018, vol. 22, no. 6 (81), pp. 168-175 (In Russ.) <https://doi.org/10.21869/2223-1560-2018-22-6-168-175>.
18. Borzov D.B., Basov R.G. [Algorithm for scheduling processor loads in multiprocessor systems]. *"Mashinostroenie i tekhnosfera XXI veka". Sbornik trudov XXV*

174 Информатика, вычислительная техника и управление / Computer science, computer engineering and control  
*Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii* ["Engineering and technosphere of the XXI century." Collection of works of the XXV International Scientific and Technical Conference]. Donetsk, 2018, vol. 1, pp. 37-39 (In Russ.).

19. Kulachev A.P. *Metody i sredstva analiza dannykh v srede Windows. STADIA 6.0* [Methods and tools for data analysis in the environment Windows. STATE 6.0]. Moscow, 2006, 270 p. (In Russ.).

20. Lobanov V.I. *Azbuka razrabotchika tsifrovyykh ustroystv* [Azbuka of the developer of digital devices]. Moscow, 2001, 192 p.

---

### Информация об авторах / Information about the Authors

**Борзов Дмитрий Борисович**, доктор технических наук, профессор, ФГБОУ ВО «Юго-Западный государственный университет», г. Курск, Российская Федерация, e-mail: borzovdb@kursknet.ru

**Dmitriy B. Borzov**, Dr. of Sci. (Engineering), Professor, Southwest State University, Kursk, Russian Federation  
e-mail: borzovdb@kursknot.ru

**Титов Виталий Семёнович**, доктор технических наук, профессор, ФГБОУ ВО «Юго-Западный государственный университет», г. Курск, Российская Федерация, e-mail: titov-kstu@rambler.ru

**Vitaliy S. Titov**, Dr. of Sci. (Engineering), Professor, Southwest State University, Kursk, Russian Federation  
e-mail: titov-kstu@rambler.ru

**Басов Родион Григорьевич**, аспирант, ФГБОУ ВО «Юго-Западный государственный университет», г. Курск, Российская Федерация, e-mail: r\_basov@eureca.ru

**Rodion G. Basov**, Post-Graduate Student, Southwest State University, Kursk, Russian Federation  
e-mail: r\_basov@eureca.ru