

АППАРАТНЫЕ СРЕДСТВА СОСТАВЛЕНИЯ ПЛАНА ЗАГРУЗКИ ПРОЦЕССОРОВ В МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ КРИТИЧЕСКОГО НАЗНАЧЕНИЯ

Д. Б. БОРЗОВ, В. С. ТИТОВ, Р. Г. БАСОВ

*Юго-Западный государственный университет, 305040, Курск, Россия
E-mail: borzovdb@kursknet.ru*

Рассмотрена задача составления расписания загрузки процессоров в мультипроцессорных системах критического назначения. Предложен метод планирования загрузки, ориентированный на аппаратную реализацию в связи с невозможностью применения программных средств. Описывается функциональная схема специализированного устройства планирования загрузки, проведен анализ его аппаратной и временной сложности. Путем суммирования времени выполнения операций элементами устройства с учетом увеличения возможного числа процессоров в системе получена диаграмма зависимости, из анализа которой можно сделать вывод, что время работы устройства планирования загрузки процессоров растет экспоненциально с увеличением числа процессоров мультипроцессорной системы, т.е. с ростом очереди загрузки.

Ключевые слова: мультипроцессорная система, критические системы, процессор, расписание, план загрузки, алгоритм, функциональная схема

При использовании критических мультипроцессорных систем (наблюдения, банковских, управляющих и т.п.) часто возникает необходимость моделирования решаемых задач. Применение для этих целей программных средств невозможно из-за повышения времени и аппаратной сложности мультипроцессорной системы [1—5].

Особенно это актуально для задач планирования загрузки процессоров в мультипроцессорных системах критического назначения. Чтобы все множество процессоров постоянно участвовало в работе до момента завершения всей задачи, важно своевременно загружать процессорные модули фрагментами программы. Таким образом, возможно повышение производительности мультипроцессорных систем за счет применения специализированных аппаратных средств.

Предварительно введем ряд допущения и ограничений: Vg — ветви программы параллельного вида; Yar — ярусы программы такого же вида. Подмножество $Vg = \{br_1, br_2, \dots, br_D\}$ операторов, соответствующих участку U_i ($Vg \in U_i, D \in [1, N]$), полученных после выявления независимых фрагментов кода, будем называть ветвями (здесь N — число строк программы).

Внутренние операторы подмножества $Yar = \{Yar_1, Yar_1, \dots, Yar_C\}$ назовем ярусами при отсутствии пересечений внутри участка U_i ($Yar \in U_i, C \in [1, N]$). Операторы должны быть предварительно распараллелены.

Считаем, что пары задач связаны между собой соотношением предшествования: $i \rightarrow j$ означает, что обработка задачи j начинается по окончании обработки задачи i на ветви Vg . Задачи обрабатываются с момента времени $t = 0$ и при этом прерывание на обработку других задач запрещено.

Необходимо определить время начала обработки задач S_i ($i = 1, \dots, n$) так, чтобы $C_{\max} = \max_{i=1, \dots, n} \{C_i\}$ ($C_i = S_i + d_i$) с учетом выполнения ограничений:

1) в произвольное время $t \in [0, C_{\max})$ выполняется условие $\sum_{i=1}^n g_{ik} \varphi_i(t) \leq P_k, k = 1, \dots, p$, где $\varphi_i(t) = 1$, если i -я задача обрабатывается в t -е время, иначе $\varphi_i(t) = 0$. Таким образом, задачи в процессе вычисления должны быть обеспечены процессором P_k ;

2) между задачами не должно быть нарушено отношение предшествования, т.е.

$$S_i + d_i \leq S_j, \text{ если } i \rightarrow j \text{ для } i.$$

Предлагаемый в работе алгоритм планирования загрузки процессоров построен на основе теории расписаний [6—8].

С учетом введенных ограничений алгоритм планирования загрузки процессоров в мультипроцессорных системах состоит из следующих шагов.

1. Получить граф УВ.
2. Преобразовать граф УВ в множество $\{G_{nm}\}$.
3. Разбить $\{G_{nm}\}$ на подмножества $\{S_{so}\}$.
4. Отсортировать операции в подмножестве $\{S_{so}\}$ по уменьшению t_{ij} .
5. Разбить подмножества $\{S_{so}\}$ на $\{S_{so}^k\}$, где $k = 1, \dots, m$, m — ограничение на число процессоров.
6. Установить соответствие операций определенному процессору $\{S_{p_k}^k\}, k = 1, \dots, m$.
7. На каждом временном отрезке множество $\{S_{so}\}$ отсортировать по возрастанию t_{ij} .
8. При $t = 0$ для процессора m загрузить операцию подмножества $\{S_{p_k}^k\}$.
9. Записать в $\{G'_{nm}\}$ новый результат.
10. Повторять п. 5—9, пока выполняется $\{S_{so}\} \neq \emptyset$.
11. Конец.

Множество $\{G_{nm}\}$ — это входная матрица команд, состоящая из Yar и Br . $\{S_{so}\}$ состоит из Yar_i, t_{ij} — плановый срок (момент времени, к которому i -я задача должна быть выполнена), $\{S_{so}^k\}$ — множество $\{S_{so}\}$, ограниченное числом процессоров $\{P\}$. Подмножество $\{S_{p_k}^k\}$ устанавливает расписание команд для определенного P_k , $\{G'_{nm}\}$ содержит план загрузки процессоров в мультипроцессорных критических системах.

Описанный алгоритм может быть реализован специализированным устройством, структурная схема которого приведена на рис. 1 (АПЗП — акселератор планирования загрузки процессоров, ПЗП — блок планирования загрузки процессоров, УУ — устройство управления, МП — микропроцессор, ОП — оперативная память, КПДП — контроллер прямого доступа в память, П_п — последовательный порт, П_{пп} — параллельный порт).

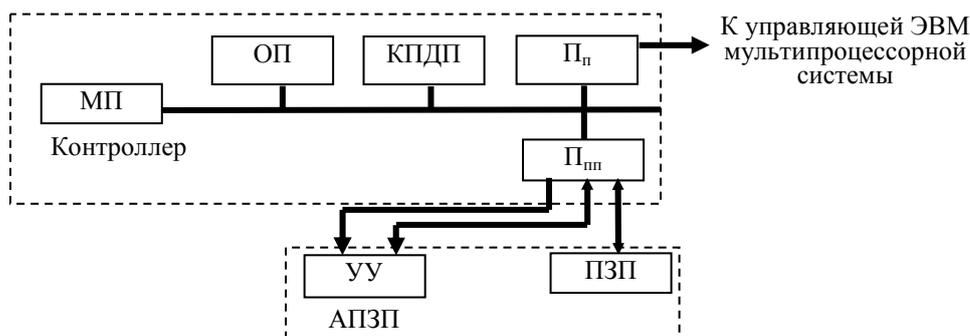


Рис. 1

МП контроллера работает в соответствии с программой, записанной в ОП, и передает исходные данные для реализации алгоритма планирования загрузки в АПЗП. Данные с параллельного порта поступают в мультиплексор АПЗП. В соответствии с режимом работы УУ реализуется алгоритм составления плана загрузки процессоров.

По окончании составления плана загрузки через параллельный порт результат работы блока АПЗП передается в контроллер и запоминается в ОП либо через последовательный порт передается к управляющей ЭВМ мультипроцессорной системы для принятия решения о возможных дальнейших действиях.

Функциональная схема устройства составления плана загрузки процессоров приведена на рис. 2.

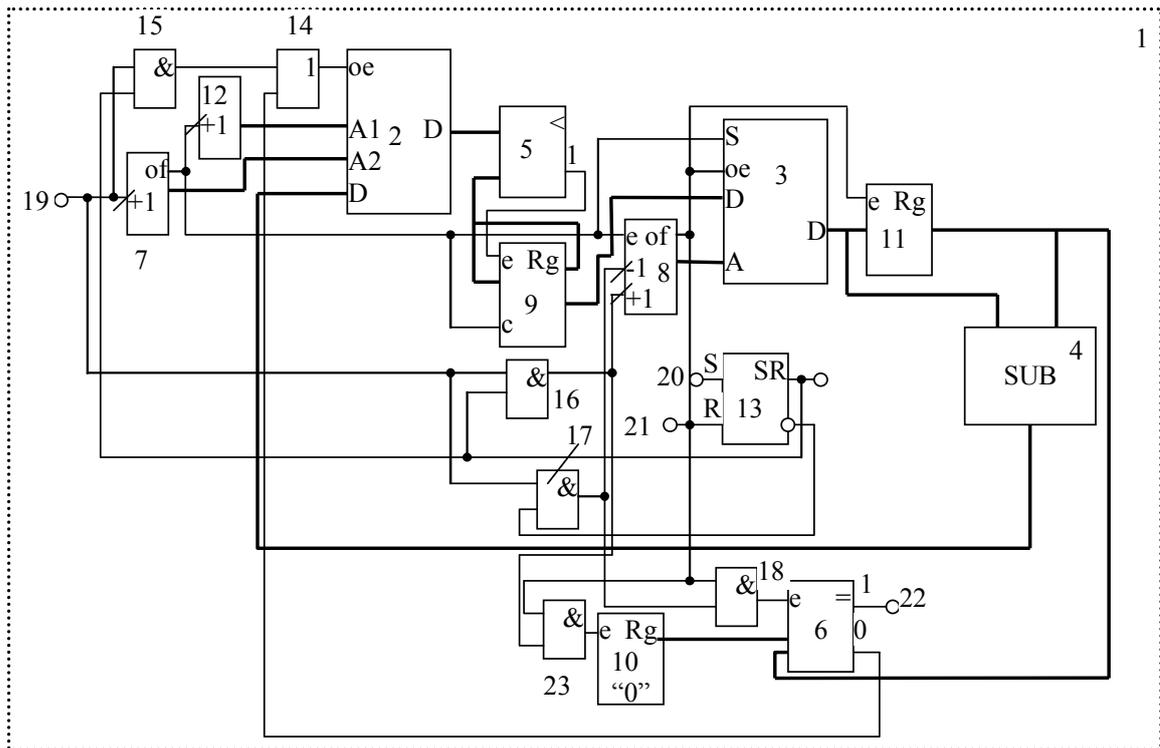


Рис. 2

ОЗУ 2 процессоров предназначено для хранения очередей загрузки заданий в каждом из соответствующих процессоров: например, $P_1 \ P_2 \ P_3 \ P_4$, где на процессор p_1 назначено задание A_1 .

Промежуточное 3 ОЗУ служит для временного хранения очередей загрузки заданий.

Элемент 4 служит для вычисления очередного минимального значения задания.

Элемент сравнения 5 предназначен для вычисления кода минимального значения загрузки.

Элемент 6 необходим для поиска кода минимального значения загрузки.

Счетчик адреса 7 предназначен для хранения адреса элемента 2.

Реверсивный счетчик адреса 8 необходим для хранения адреса ячейки 3.

Первый регистр минимума 9 предназначен для хранения текущего минимального значения кода загрузки процессора.

Регистр кода нуля 10 необходим для хранения кода значения нуля („0...00“).

Второй регистр минимума 11 служит для хранения текущего минимального значения кода загрузки процессора. Код используется при очередном поиске минимального кода значения загрузки, необходимого для поиска очередного минимального значения на новом шаге.

Это означает, что очередной шаг поиска кодов значений загрузки процессоров завершён и необходим очередной поиск, при этом предыдущие значения сохраняются в ОЗУ 2.

Счетчик уровня 12 предназначен для хранения кодов очередей загрузки заданий, на-

$$P_1 \quad P_2 \quad P_3 \quad P_4$$

пример, в случае распределения заданий: $A_1 \quad A_2 \quad A_3 \quad A_4$ на первом этапе будут выполняться

$$A_5 \quad A_6 \quad A_7 \quad A_8$$

параллельно задания $A_1 \quad A_2 \quad A_3 \quad A_4$, а на втором — $A_5 \quad A_6 \quad A_7 \quad A_8$.

Элемент 13 служит для выбора режима работы схемы. Если триггер установлен в состояние „единица“, на прямом выходе присутствует единичный импульс, а на инверсном — единица. Первый 15 элемент И служит для объединения сигналов с выходов первого 14 элемента И и инверсного выхода триггера 13. Вторым 16 и четвертым 17 элементами И необходимы для разделения режима работы схемы: на первый и второй этапы выбора значений загрузки процессоров. Элемент 14 ИЛИ необходим для объединения сигналов с выхода элемента 15 и второго выхода элемента 6. Элемент 18 предназначен для разрешения работы элемента 6. Элемент 23 необходим для разрешения работы регистра 10.

Для анализа производительности предложенного акселератора воспользуемся серией 1533 интегральных микросхем. При подсчете быстродействия и производительности учитывались временные характеристики отдельных внутренних элементов предложенного устройства (табл. 1) согласно серии микросхем ТТЛ 1533 [9—12].

Таблица 1

Временные характеристики	
Тип элемента	t , нс
НЕ	11
И	11
ИЛИ	12
Дешифратор	36
Мультиплексор	20
Сумматор/Умножитель	40
RS-триггер	20
Регистр	21
Счетчик	18
ОЗУ	30

Таким образом, суммируя все время выполнения операций элементами устройства с учетом увеличения возможного количества процессоров в системе, получаем диаграмму зависимости, рис. 3 [13—15].

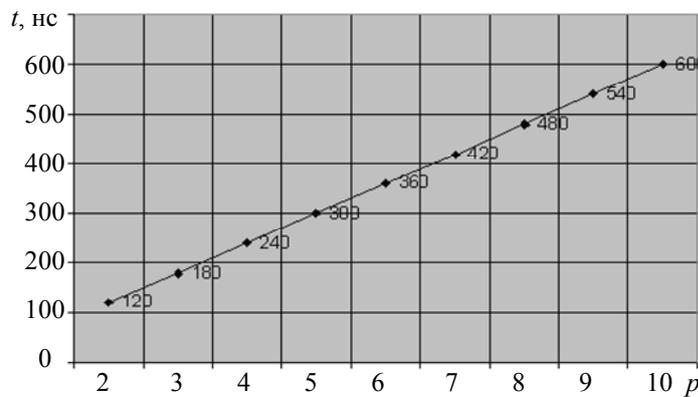


Рис. 3

Для оценки аппаратных затрат воспользуемся представлением модели в виде эквивалентных вентилях (L логического элемента И) [9—12]. Так, RS-триггер эквивалентен двум

логическим элементам И, счетчик — семнадцать элементам И, схема сравнения — шестнадцать И, регистр — тридцати двум И, один разряд сумматора можно представить с помощью тридцати двух элементов И, а 64 разрядный сумматор — с помощью 544 элементов И. На один процессор отводится 1 Б памяти и соответственно 8 триггеров или 16 элементов И для хранения в памяти, например, очереди для пяти процессоров потребуется 160 элементов И, для десяти — 960 и т.д.

В результате можно построить зависимость роста аппаратных затрат от объема решаемой задачи (рис. 4). При этом будет учитывать, что предложенное устройство содержит следующее количество элементов (табл. 2) [13—15].

Таблица 2

Тип элемента	Количество внутренних элементов устройства	
	элементов	Число эквивалентных вентиляей
ИЛИ	1	1
Сумматор/Умножитель/Делитель	1	32
RS-триггер	1	2
Регистр	3	32
Счетчик	3	17
ОЗУ	2	16 на 1 ячейку

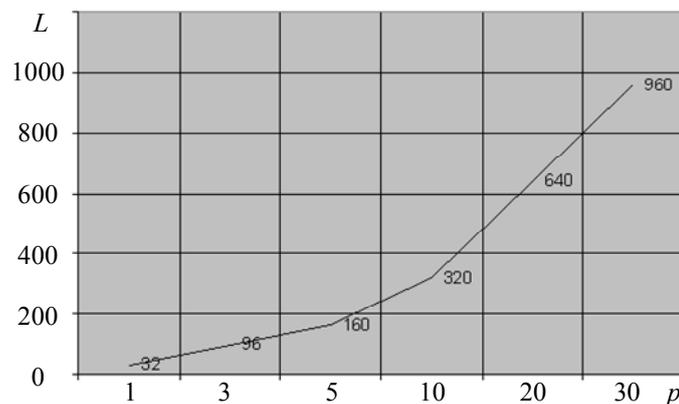


Рис. 4

Из рис. 4 следует, что работа устройства требует относительно небольших аппаратных затрат. Так, для тридцати процессоров требуется 960 эквивалентных вентиляей, что совсем немного при уровне современных технологий, когда на кристалле микросхемы содержится миллиард транзисторов.

СПИСОК ЛИТЕРАТУРЫ

1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб: БХВ-Петербург, 2002. 600 с.
2. Корнеев В. В. Параллельные вычислительные системы. М.: Нолидж, 1999. 340 с.
3. Flynn M. Very high-speed computing system // Proc. IEEE. 1966. N 54. P. 1901—1909.
4. Flynn M. Some Computer Organisations and Their Effectiveness // IEEE Trans. Computers. 1972. Vol. 21, N 9. P. 948—960.
5. Thurber K. J. Large Scale Computer Architecture. New Jersey, Rochelle Park: Hayden Book Company. 1976.
6. Танаев В. С., Ковалев М. Я. Теория расписаний. Групповые технологии. Минск: Ин-т технической кибернетики НАН Беларуси, 1998. 290 с.
7. Лазарев А. А., Гафаров Е. Р. Теория расписаний задачи и алгоритмы. М., 2011. С. 31—32.
8. Конвей Р. В., Максвелл В. Л., Миллер Л. В. Теория расписаний. М.: Наука, 1975. 365 с.
9. Шило В. Л. Популярныe цифровые микросхемы: Справочник. Челябинск: Металлургия, 1988. 352 с.

10. Батушев В. А., Вениаминов В. Н., Ковалев В. Г. Микросхемы и их применение. М.: Энергия, 1978. 248 с.
11. Тарабрин Б. В., Лунин Л. Ф., Смирнов Ю. Н. Интегральные микросхемы: Справочник. М.: Радио и связь, 1984. –528 с.
12. Петровский И. И. и др. Логические ИС КР1533, КР1554: Справочник в 2 ч. М.: Бином, 1993.
13. Макарова Н. В. Статистика в Excel. М.: Финансы и статистика, 2009. 368 с.
14. Закс Л. Статистическое оценивание. М.: Статистика, 2008. 598 с.
15. Кобзарь А. И. Прикладная математическая статистика. М.: Физматлит, 2006. 816 с.

Сведения об авторах

- Дмитрий Борисович Борзов** — д-р техн. наук, профессор; Юго-Западный государственный университет; E-mail: borzovdb@kursknet.ru
- Виталий Семенович Титов** — д-р техн. наук, профессор; Юго-Западный государственный университет; E-mail: titov-kstu@rambler.ru
- Родион Григорьевич Басов** — аспирант; Юго-Западный государственный университет; E-mail: r_basov@eureca.ru

Поступила в редакцию
18.01.19 г.

Ссылка для цитирования: Борзов Д. Б., Титов В. С., Басов Р. Г. Аппаратные средства составления плана загрузки процессоров в мультипроцессорных системах критического назначения // Изв. вузов. Приборостроение. 2019. Т. 62, № 6. С. 517—523.

HARDWARE MEANS FOR PLANNING PROCESSORS LOAD IN MULTIPROCESSOR SYSTEMS OF CRITICAL PURPOSE

D. B. Borzov, V. S. Titov, R. G. Basov

Southwest State University, 305040, Kursk, Russia
E-mail: borzovdb@kursknet.ru

The problem of scheduling processor loading in multiprocessor systems of critical purpose is considered. The proposed method of the load planning is focused on hardware realization in connection with impossibility of application of software. A specialized device for the load planning is developed; functional scheme of the device is described, analysis of its hardware and time complexity is presented. By summing up the execution time of operations by the elements of the device, taking into account the increase in the possible number of processors in the system, a dependency diagram is obtained. Analysis of the diagram allows to conclude that the operating time of the processor load planning device grows exponentially with the number of processors in the multiprocessor system, i.e. with the increase in the size of the load queue.

Keywords: multiprocessor system, critical system, processor, schedule, load plan, algorithm, functional scheme

REFERENCES

1. Voyevodin V. V., Voyevodin V. V. *Parallelnyye vychisleniya* (Parallel Computing), St. Petersburg, 2002, 60S p. (in Russ.)
2. Korneyev V.V. *Parallelnyye vychislitel'nyye sistemy* (Parallel Computing Systems), Moscow, 1999, 340 p. (in Russ.)
3. Flynn M. *Proc. IEEE*, 1966, no. 54, pp. 1901–1909.
4. Flynn M. *IEEE Trans. Computers*, 1972, no. 9(21), pp. 94S–960.
5. Thurber K.J. *Large Scale Computer Architecture*, Hayden Book Company, Rochelle Park, New Jersey, 1976.
6. Tanayev V.S., Kovalev M.Ya. *Teoriya raspisaniy. Gruppyvyye tekhnologii* (Scheduling Theory. Group Technology), Minsk, 1998, 290 p. (in Russ.)
7. Lazarev A.A., Gafarov E.R. *Teoriya raspisaniy. Zadachi i algoritmy* (Scheduling Theory. Tasks and Algorithms), Moscow, 2011, 222 p. (in Russ.)
8. Conway R.W., Maxwell W.L., Miller L.W. *Theory of Scheduling*, Addison-Wesley, 1967, 294 p.
9. Shilo V.L. *Populyarnyye tsifrovyye mikroshemy: Spravochnik* (Popular Digital ICs: Handbook), Chelyabinsk, 1988, 352 p. (in Russ.)
10. Batushev V.A., Veniaminov V.N., Kovalev V.G. *Mikroshemy i ikh primeneniye* (ICs and Their Application), Moscow, 1978, 248 p. (in Russ.)

11. Tarabrin B.V., Lunin L.F., Smirnov Yu.N. *Integral'nyye mikroskhemy. Spravochnik* (Integrated Circuits. A Handbook), Moscow, 1984, 528 p. (in Russ.)
12. Petrovskiy I.I. et al. *Logicheskiye IS KR1533, KR1554. Spravochnik* (Logic IC KR1533, KR1554. A Handbook), Moscow, 1993. (in Russ.)
13. Makarova N.V. *Statistika v Excel* (Excel Statistics), Moscow, 2009, 368 p. (in Russ.)
14. Sachs L. *Statistische Auswertungsmethoden*, Berlin, Springer-Verlag, 1968.
15. Kobzar' A.I. *Prikladnaya matematicheskaya statistika* (Applied Mathematical Statistics), Moscow, 2006, 816 p. (in Russ.)

Data on authors

- Dmitry B. Borzov** — Dr. Sci., Professor; Southwest State University;
E-mail: borzovdb@kursknet.ru
- Vitaly S. Titov** — Dr. Sci., Professor; Southwest State University; E-mail: titov-kstu@rambler.ru
- Rodion G. Basov** — Post-Graduate Student; Southwest State University;
E-mail: r_basov@eureca.ru

For citation: Borzov D. B., Titov V. S., Basov R. G. Hardware means for planning processors load in multiprocessor systems of critical purpose. *Journal of Instrument Engineering*. 2019. Vol. 62, N 6. P. 517—523 (in Russian).

DOI: 10.17586/0021-3454-2019-62-6-517-523