

Интернет-журнал «Науковедение» ISSN 2223-5167 <http://naukovedenie.ru/>

Том 8, №5 (2016) <http://naukovedenie.ru/index.php?p=vol8-5>

URL статьи: <http://naukovedenie.ru/PDF/84TVN516.pdf>

Статья опубликована 22.11.2016.

Ссылка для цитирования этой статьи:

Ткачев П.Ю., Чернецкая И.Е. Устройство распараллеливания циклических участков последовательных программ // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №5 (2016)

<http://naukovedenie.ru/PDF/84TVN516.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

УДК 004.272.26

Ткачев Павел Юрьевич

ФГБОУ ВО «Юго-Западный государственный университет», Россия, Курск
Аспирант кафедры «Вычислительная техника»
E-mail: amdathlon64@yandex.ru

Чернецкая Ирина Евгеньевна

ФГБОУ ВО «Юго-Западный государственный университет», Россия, Курск
Профессор кафедры «Вычислительная техника»
Доктор технических наук, доцент
E-mail: white731@yandex.ru

Устройство распараллеливания циклических участков последовательных программ

Аннотация. Распараллеливание программ является одним из путей повышения быстродействия многопроцессорных вычислительных систем. Большую часть времени выполнения программ в многопроцессорных вычислительных системах занимают циклы, поэтому их распараллеливание приводит к повышению быстродействия вычислительной системы и к более эффективному использованию ресурсов. В данном исследовании представлен разработанный метод распараллеливания циклических участков последовательных программ, основанный на определении возможности параллельного выполнения итераций цикла, поиске и распараллеливании линейных участков внутри циклов с неизвестным количеством итераций. Также представлены структурная и функциональные схемы устройства, основанного на разработанном методе, которое позволяет сократить время выполнения циклических участков последовательных программ в вычислительных системах высокой готовности. Исследование показало, что применение разработанного устройства при распараллеливании циклических участков последовательных программ позволяет сократить временные затраты на распараллеливание исходного линейного участка в 10 раз по сравнению с программной реализацией на хост-процессоре при 50 операторах исходного циклического участка программы. Применение разработанного устройства снижает функциональную нагрузку на хост-процессор, за счет снятия с него функций распараллеливания, что соответствует поставленной задаче исследования.

Ключевые слова: многопроцессорные системы; системы высокой готовности; параллельные ЭВМ; параллельные программы; распараллеливание; оператор; цикл

В настоящее время широкое распространение получили многопроцессорные вычислительные системы. Развитие многопроцессорной техники связано с возрастанием количества процессоров в многопроцессорных системах, что требует загрузки каждого

процессора информационно независимыми задачами и вызывает необходимость распараллеливания последовательных программ для их выполнения на нескольких процессорах одновременно. Поскольку большую часть времени выполнения программ, как правило, занимают циклы, их распараллеливание приведет к увеличению быстродействия системы. Существующие методы и алгоритмы выявления информационно-независимых участков программ в большинстве своем являются программными, и проблема распараллеливания циклических участков последовательных программ рассмотрена в них в недостаточной степени. Задача распараллеливания в многопроцессорных системах при программной реализации выполняется хост-процессором, в число задач которого входит маршрутизация, назначение и перераспределение задач, реконфигурация и функции, необходимые для управления системой, что создает дополнительную нагрузку на хост-процессор. Таким образом, разработка устройства, обеспечивающего распараллеливание циклических участков последовательных программ в режиме реального времени и не загружающих хост-процессор, является актуальной задачей [1-3].

Распараллеливание циклических участков с большим количеством итераций при распараллеливании в системах высокой готовности является продолжением исследований, начатых в [4-7].

Для решения поставленной задачи использован метод распараллеливания циклических участков последовательных программ, состоящий из следующей последовательности этапов распараллеливания:

Этап 1. Определение типа цикла производится по наличию счетчика в цикле: цикл с заранее известным количеством итераций или цикл с неизвестным количеством итераций. В случае цикла с известным количеством итераций далее выполняется этап 2, в случае цикла с неизвестным количеством итераций переходим к этапу 3.

Этап 2. Определение возможности параллельного выполнения итераций по способу, описанному в [4]. Если такая возможность есть, производится распараллеливание, если такая возможность отсутствует, переходим к этапу 3.

Этап 3. Поиск линейных участков внутри цикла по способу, представленному в [5]. Данный способ применим как для циклов с неизвестным количеством итераций, так и к циклам с известным количеством итераций в том случае, если их не удалось распараллелить на предыдущих этапах.

Этап 4. Составление матриц достижимости для найденных на этапе 3 линейных участков.

Этап 5. Составление матриц неполного параллелизма и распараллеливание найденных на этапе 3 линейных участков. Матрицы неполного параллелизма вычисляются по ускоренному способу, описанному в [6]. После составления матриц неполного параллелизма составляется ярусно-параллельная форма готовых параллельных участков, и выдаются управляющие команды хост-процессору.

Хост-процессор вычислительной системы передает в устройство матрицу следования и матрицы входных и выходных переменных исходного циклического участка [7, 8].

Устройство функционирует на базе микропроцессорного контроллера. На аппаратный уровень устройства вынесены операции, требующие больших временных затрат, а именно: определение возможности параллельного выполнения итераций цикла, поиск линейных участков внутри цикла, вычисление матрицы неполного параллелизма, распараллеливание.

Структурная схема разработанного устройства приведена на рисунке 1. Устройство распараллеливания циклических участков последовательных программ включает в себя

следующие блоки и узлы: МП – микропроцессор, ОП – оперативная память, КПДП – контроллер прямого доступа в память, ППОРТ – последовательный порт для связи с хост-процессором, БОВПВИЦ – блок определения возможности параллельного выполнения итераций цикла, БПЛУ – блок поиска линейных участков, БВМНП – блок вычисления матрицы неполного параллелизма, БРЛУ – блок распараллеливания линейных участков.

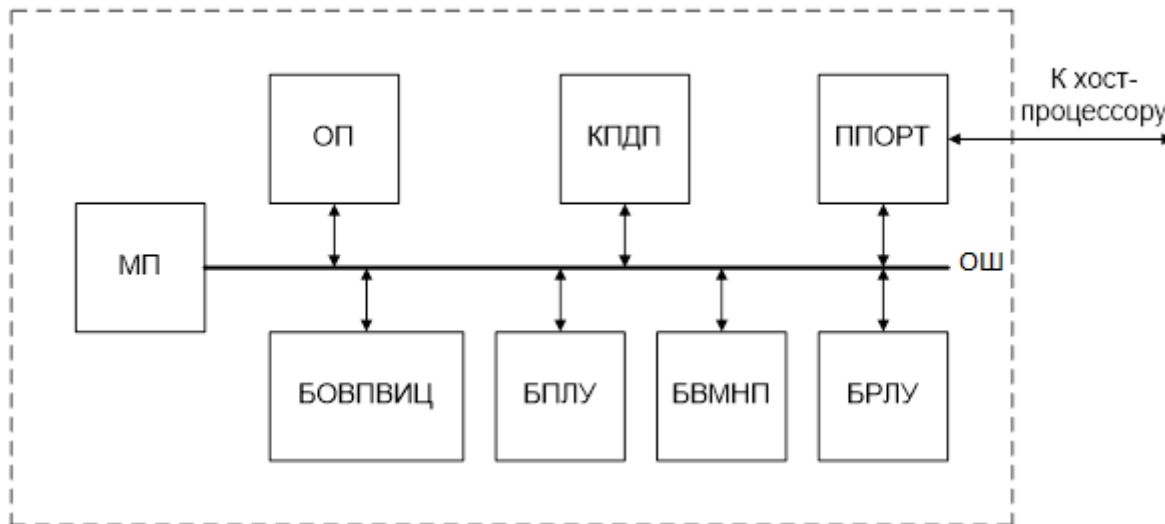


Рисунок 1. Структурная схема устройства распараллеливания циклических участков последовательных программ (составлено автором)

МП устройства работает в соответствии с программой, записанной в ОП и передает в аппаратные блоки устройства соответствующие задачи.

Результатом работы устройства распараллеливания циклических участков последовательных программ является набор команд для хост-процессора, в формате, представленном на рисунке 2.

N_op	N_it	Yarus	Branch
------	------	-------	--------

Рисунок 2. Формат результирующих команд (составлено автором)

Команда содержит номер оператора N_op, номер итерации N_it, поля Yarus и Branch, определяющие номер процессорной группы для выполнения участка, в который входит данный оператор, и порядок выполнения группы.

Функциональные схемы каждого их аппаратных блоков устройства (согласно рисунку 1) представлены на рисунках 3-6.

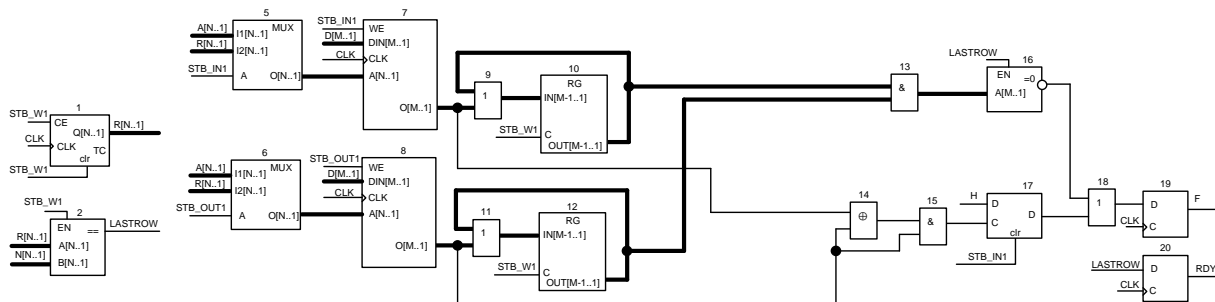


Рисунок 3. Функциональная схема блока определения возможности параллельного выполнения итераций цикла (составлено автором)

Входными данными для блока определения возможности параллельного выполнения итераций цикла являются матрицы входных и выходных переменных. Выходные данные – возможность параллельного выполнения итераций цикла [9].

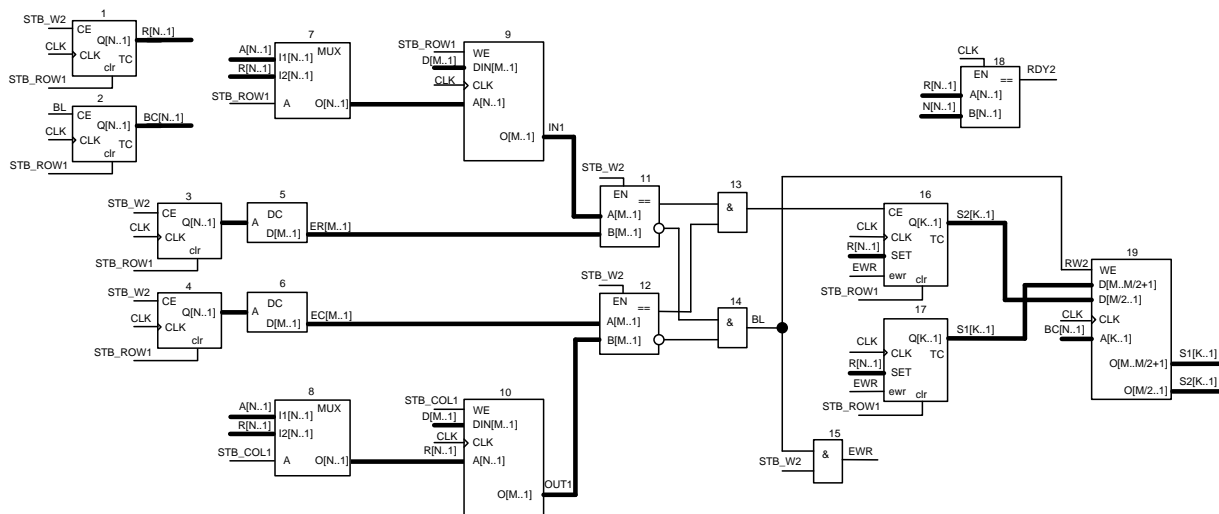


Рисунок 4. Функциональная схема блока поиска линейных участков (составлено автором)

Входными данными для блока поиска линейных участков внутри цикла является матрица следования. Выходные данные – массив найденных линейных участков [10].

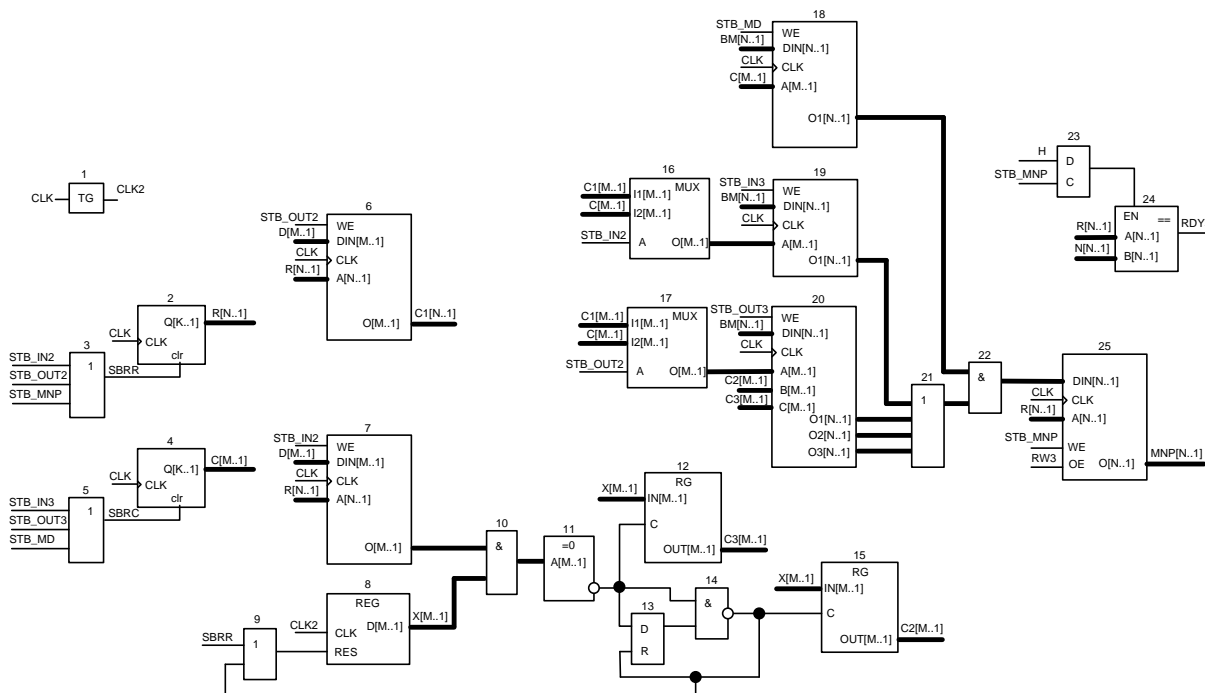


Рисунок 5. Функциональная схема блока вычисления матрицы неполного параллелизма (составлено автором)

Входными данными для блока вычисления матрицы неполного параллелизма является матрица следования, матрицы входных и выходных переменных. Выходные данные – матрица неполного параллелизма.

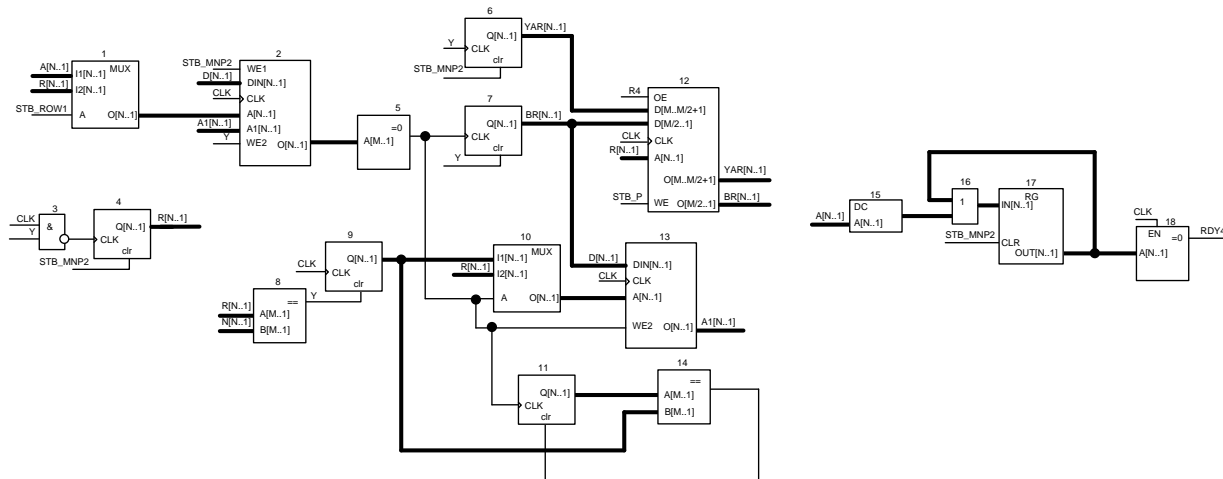


Рисунок 6. Функциональная схема блока распараллеливания (составлено автором)

Входными данными для блока распараллеливания является матрица неполного параллелизма. Выходные данные – массив ярусов и ветвей для распараллеливаемого участка программы.

Анализ производительности устройства распараллеливания циклических участков последовательных программ проведен с учетом временных характеристик отдельных элементов серии микросхем ТТЛ 1533 [11, 12]. Временные задержки элементов серии 1533 приведены в таблице 1.

Таблица 1

Временные задержки элементов серии ТТЛ 1533

Тип элемента	Время задержки, нс
И-НЕ	8
И	11
ИЛИ	12
Сложение по модулю 2	13
Дешифратор	36
Мультиплексор	20
D-триггер	15
Компаратор	31
Регистр	21
Счетчик	18
ОЗУ	30

Таким образом можно оценить быстродействие разработанного устройства. Зависимость времени работы устройства от количества операторов исходной программы представлена на рисунке 7.

В результате анализа графика, представленного на рисунке 7, можно сделать вывод о том, что временные затраты на распараллеливание растут экспоненциально с ростом количества операторов исходного участка программы. При небольшом исходном участке (10 операторов) временные затраты на распараллеливание составят менее 0.1 мс. При исходном участке в 500 операторов временные затраты составят 1,11 мс.

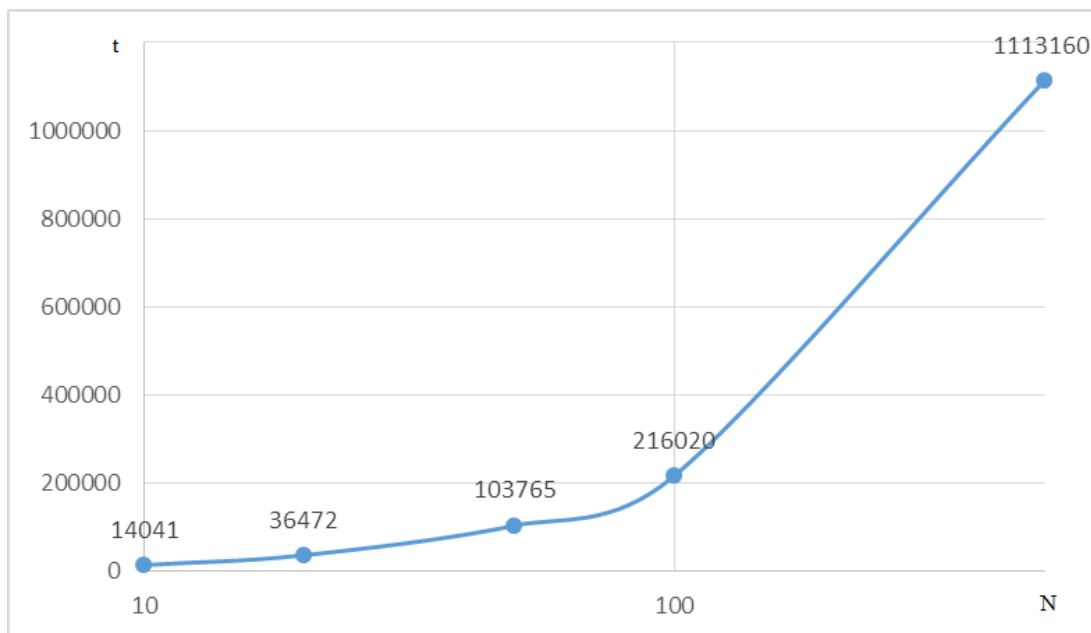


Рисунок 7. Зависимость времени работы устройства от количества операторов исходной программы (составлено автором)

Для оценки аппаратных затрат представим элементы схемы устройства в виде эквивалентных вентиляей. Под эквивалентным вентиляем будем понимать логический элемент «И». В таблице 2 представлено количество внутренних элементов устройства в пересчете на эквивалентные вентиляей.

Таблица 2

Количество внутренних элементов устройства

Тип элемента	Количество	Количество эквивалентных вентиляей
И, ИЛИ, И-НЕ	20	$1 \cdot 20 = 20$
Мультиплексор	8	$8 \cdot 16 = 128$
Регистр	8	$8 \cdot 32 = 256$
Счетчик	10	$10 \cdot 17 = 170$
Дешифратор	2	$2 \cdot 16 = 32$
Компаратор	10	$10 \cdot 16 = 160$
ОЗУ	14	16 на 1 ячейку

Зависимость аппаратной сложности устройства от количества операторов исходной распараллеливаемой программы представлена на рисунке 8.

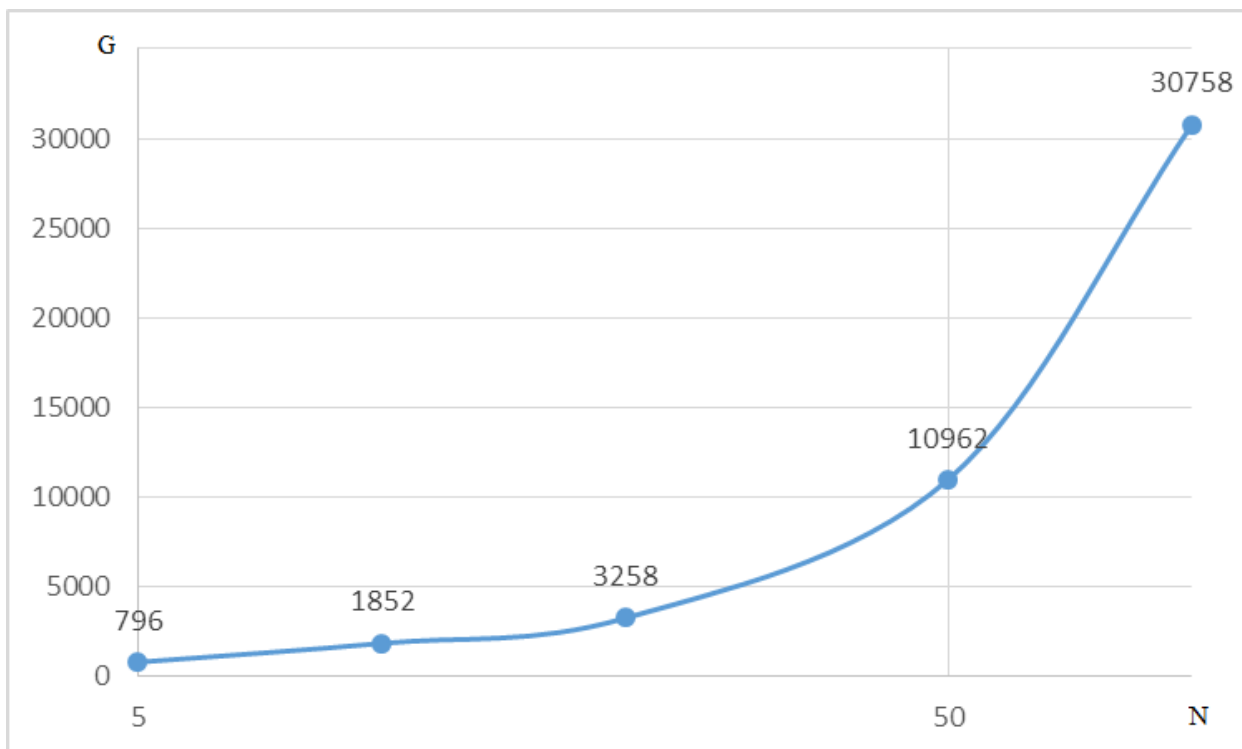


Рисунок 8. Зависимость аппаратной сложности устройства от количества операторов исходной распараллеливаемой программы (составлено автором)

В результате анализа зависимости, представленной на рисунке 8, можно сделать вывод о том, что для работы устройства требуются относительно небольшие аппаратные затраты. Так, например, для распараллеливания участка в 100 операторов необходимо 30758 эквивалентных вентиляей.

Также был проведен сравнительный анализ временных затрат на программную реализацию распараллеливания с аппаратной реализацией в разработанном устройстве. Результаты сравнения времени распараллеливания представлены в таблице 3.

Таблица 3

Результаты сравнения времени распараллеливания

Количество операторов	5	10	50	100
Хост-процессор	98 мкс	178 мкс	1 мс	14 мс
Разработанное устройство	10 мкс	14 мкс	0.1 мс	0.2 мс

Таким образом, исходя из вышеизложенного, в результате решения поставленной актуальной задачи по разработке устройства распараллеливания циклических участков последовательных программ были разработаны структурная и функциональные схемы устройства, которые позволяют сократить время распараллеливания циклических участков последовательных программ в 10 раз по сравнению с программной реализацией при 50 операторах исходного циклического участка программы.

ЛИТЕРАТУРА

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
2. Воеводин Вл.В. Распределенная обработка данных // Вторая сибирская школа-семинар по параллельным вычислениям. Томск: ТГУ, 2004. С. 3-9.
3. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. СПб.: Питер, 2011. 688 с.
4. Ткачев П.Ю., Борзов Д.Б. Метод и алгоритм распараллеливания циклов со счетчиками. // «Известия вузов. Приборостроение», №2 (58), 2015. С. 104-108.
5. Ткачев П.Ю., Борзов Д.Б., Чернецкая И.Е. Метод и алгоритм поиска линейных участков внутри циклов с последующим распараллеливанием. «Известия Юго-Западного государственного университета», №5 (62), 2015. С. 16-20.
6. Ткачев П.Ю. Метод ускоренного вычисления матрицы неполного параллелизма и его программное моделирование. «Известия Юго-Западного государственного университета» Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение, №, 2016. С. 7-12.
7. Ткачев П.Ю., Борзов Д.Б. Поиск и распараллеливание линейных участков внутри циклов на разных уровнях вложенности. Сборник материалов международной научно-технической конференции «Машиностроение и техносфера XXI века». Севастополь, 2015. С. 137-141.
8. Дюбрюкс С.А. Аппаратно-ориентированный подход к выявлению параллелизма в последовательных программах со связями по управлению и циклами. Сборник трудов XVI международной научно-технической конференции «Машиностроение и техносфера XXI века. Т1». Донецк, 2009. С. 123-129.
9. Ткачев П.Ю., Борзов Д.Б. Устройство для определения возможности параллельного выполнения итераций цикла / Патент №158124 Российская Федерация, МПК G06F 7/00, заявл. 27.07.2015, опубл. 20.12.2015, Бюл.№35, 17 с. 2 ил.
10. Ткачев П.Ю., Борзов Д.Б. Устройство для поиска линейных участков внутри цикла / Патент №160027 Российская Федерация, МПК G06F 7/00, заявл. 24.07.2015, опубл. 27.02.2016, Бюл.№6, 18 с. 2 ил.
11. Лаврентьев Б.Ф. Схемотехника электронных средств. Учебное пособие. М.: Академия, 2010. 308 с.
12. Китаев Ю.В. Основы цифровой техники. Учебное пособие. СПб.: СПбГУ ИТМО, 2007. 87 с.

Tkachev Pavel Yourievich

Southwest state university, Russia, Kursk
E-mail: amdathlon64@yandex.ru

Chernetskaya Irina Yevgenyevna

Southwest state university, Russia, Kursk
E-mail: white731@yandex.ru

Device for parallelization of the sequential program looping constructs

Abstract. Parallelization of programs is one of the ways for speeding multiprocessor computer systems. In the multiprocessor computer systems the loops take up most of the time for execution of programs. That's why their parallelization leads to the computer system speeding and a more efficient use of resources. This study presents the developed method for parallelization of sequential program looping constructs, based on ascertaining the capability of the parallel execution of the loop iterations, search and parallelization of linear parts within loops with an unknown number of iterations. The article also presents the structural and functional schemes of the device based on the developed method, which makes it possible to improve the execution time of the sequential program looping constructs in the high availability computing systems. The study showed that the use of the developed device when parallelizing sequential program looping constructs can cut the time spending required to parallelize the initial linear part to 10 as compared with software implementation on the host processor with 50 operators of the initial looping construct of the program. Application of the developed device reduces the functional load on the host processor by the aid of removing the parallelization functions from it, that is corresponding to the task of research in hand.

Keywords: multiprocessor systems; high availability systems; parallel computers; parallel programs parallelization; operator; loop