

**ОПТИМАЛЬНОЕ РАСПАРАЛЛЕЛИВАНИЕ ЗАДАЧ ДИСПЕТЧЕРИЗАЦИИ  
В РАСПРЕДЕЛЕННЫХ ЭНЕРГООГРАНИЧЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ  
НА БАЗЕ MCU/FPGA МОДУЛЕЙ**

© 2014

*М.П. Мусиенко*, доктор технических наук, профессор,  
заведующий кафедрой информационных технологий и программных систем

*В.Ю. Савинов*, аспирант

*Я.М. Крайнык*, аспирант

*Черноморский государственный университет имени Петра Могилы, Николаев (Украина)*

**Ключевые слова:** распределенные компьютерные системы; оптимальное распараллеливание; диспетчеризация; MCU и FPGA модули; энергопитание.

**Аннотация:** В статье рассмотрена задача эффективного распределения энерго- и информационных ресурсов в распределенных энергоограниченных компьютерных системах на основе микроконтроллеров и программируемых вентильных матриц за счет оптимального распараллеливания задач диспетчеризации. Рассмотрен пример распараллеливания вычислительных ресурсов между двумя модулями для решения задач оптимального управления станком с 3D позиционированием.

Миниатюризация электронной техники приводит к широкому использованию вычислительных элементов (микроконтроллеров, вентильных матриц, SoC и пр.) в различных удаленных узлах: датчиках, актуаторах и прочих исполнительных элементах. В этом случае особо остро стоит задача эффективного энергопотребления, так как удаленные узлы, как правило, питаются от автономных источников энергии. Решение данной проблемы возможно за счет эффективной диспетчеризации выполняемых заданий, оптимального распараллеливания потоков данных и команд между всеми вычислительными модулями системы. Таким образом, задача оптимального распараллеливания задач диспетчеризации в распределенных энергоограниченных компьютерных системах является весьма важной и актуальной проблемой.

На сегодняшний день существует множество различных решений в этой области. Например, использование подхода EDF (earliest deadline first) – минимизация возможных запаздываний за счет выбора частоты процессора [1]; методы диспетчеризации задач в распределенных компьютерных системах на основе гибридного метода, который представляет собой сочетание жадных и алгоритмов и усовершенствованного эволюционного метода [2] и многие другие. Однако основным недостатком всех этих методов является то, что при их разработке не учитывался критерий ограниченного энергопотребления системы. Кроме того, в этих методах не рассматривался принцип распараллеливания, который широко используется для решения множества задач.

Целью данной работы является решение задачи эффективного распределения энерго- и информационных ресурсов в распределенных энергоограниченных компьютерных системах на основе микроконтроллеров (MCU – Micro Controller Unit) и программируемых вентильных матриц (FPGA – Field-Programmable Gate Array) за счет оптимального распараллеливания задач диспетчеризации.

Для непрерывной работы удаленного модуля необходимо выполнение условия:

$$E_{зап} + E_{посуп} - E_{ЭК} - E_{проц} > 0,$$

где  $E_{зап}$  – запасенная энергия;  $E_{посуп}$  – энергия поступления от сторонних источников (например, регенерируемых источников энергии);  $E_{ЭК}$  – энергия потребления электронных компонентов (АЦП, датчиков, усилителей и др.);  $E_{проц}$  – энергия потребления микропроцессором.

Для увеличения срока непрерывной работы в статье рассматривается метод оптимальной диспетчеризации, которая позволит уменьшить величину энергии потребления процессора  $E_{проц}$ .

При использовании подхода EDF запаздывание всех задач, поступивших на выполнение, будет минимальным. Режим работы  $\bar{s}_k$  процессора для  $i$ -ой работы с наивысшим приоритетом находится следующим образом:

$$\bar{s}_k = \max_{i=1}^{n_k} \{u_{k,i}\},$$

где  $u_{k,i}$  – текущий коэффициент загрузки процессора:

$$u_{k,i} = \sum_{p=1}^i e_{k,p} / d_{k,i},$$

$e_{k,p}$  – оставшееся время выполнения  $p$ -го задания,

$i=1..n_k$ ,

$k$  – номер набора работ,  $n$  – число работ в наборе  $k$ .

Обозначим множество режимов работ как  $R = \{r_1..r_z\}$ . Частота процессора  $f_i$  при выполнении  $i$ -ой работы будет определяться из выражения:

$$f_i = \min_{z=1}^y \{r_z \mid r_z \geq \bar{s}_k\} \times f_{\max},$$

где  $y$  – количество режимов,  $f_{\max}$  – максимальная частота процессора.

Таким образом, энергопотребление процессора при выполнении  $i$ -ой работы будет равным

$$E_i = CU_i^2 f_i,$$

где  $C$  – динамическая емкость;

$U_i$  – напряжение на процессоре при выполнении  $i$ -ой работы.

Постановку задачи диспетчеризации задач в распределенных энергоограниченных системах осуществим используя известные методы [2]. Пусть в момент времени  $t$  в распределенную систему, состоящую из  $n$  распределенных модулей, для решения поступило некоторое число задач  $m$ , где  $n$  и  $m$  – некоторые натуральные числа. Каждый модуль имеет свое значение мощности  $N_j$ ,  $j = \overline{1, n}$ . Каждое задание имеет свой весовой коэффициент длительности  $a_i$ ,  $i = \overline{1, m}$  со значением мощности для ее выполнения  $N_i^o$ . В задании необходимо распределить все задачи для решения в вычислительной системе таким образом, чтобы вычислительные и энергетические ресурсы были использованы максимально эффективно.

Зададим это значение через  $t_{ij}$ , которое определяется как:

$$t_{ij} = \frac{N_i^o}{N_j} \cdot a_i.$$

Существует некоторая матрица оптимального распределения задач, в которой каждый элемент может принимать следующие значения:

$$\begin{cases} x_{ij} = 1, & i\text{-ое задание назначено } j\text{-му модулю;} \\ x_{ij} = 0, & i\text{-ое задание не назначено } j\text{-му модулю.} \end{cases}$$

Если вычислительные и энергетические ресурсы используются наиболее эффективно, максимальное время работы модуля по обработке задания будет минимальным, т.е.  $\max\left(\sum_{i=1}^m x_{ij} \cdot t_{ij}\right) \rightarrow \min$ . Это означает, что суммарное время решения всех задач в вычислительной системе минимально при времени вычисления на каждом из модулей системы максимально приближенном друг к другу. Для приближения к минимуму максимального среди реально полученных значений вводится ограничения  $\sum_{i=1}^m x_{ij} \cdot z_i \leq v_j$ , где  $z_i$  – объем необходимой оперативной памяти для  $i$ -го задания;  $v_j$  – объем необходимой оперативной памяти  $j$ -го модуля ( $j = \overline{1, n}$ ) [2].

Таким образом, общая математическая модель для решения задания оптимального распараллеливания задач диспетчеризации имеет вид:

$$\begin{cases} \max\left(\sum_{i=1}^m x_{ij} \cdot t_{ij}\right) \rightarrow \min, \text{ при } \sum_{i=1}^m x_{ij} \cdot z_i \leq v_j, j = \overline{1, n}, \\ E_{3АП} + E_{Поступл} - E_{Эк} - E_{Проц} > 0, \\ C \sum_{i=1}^k U_i^2 f_i \rightarrow \min, \text{ при } f_i = \min_{z=1}^y \{r_z \mid r_z \geq \bar{s}_k\} \times f_{\max}. \end{cases}$$

В качестве примера рассмотрим компьютерную систему управления позиционированием головки 3D-системы (принтера, станка с ЧПУ и пр.), построенную с использованием двух вычислительных ресурсов – одного MCU и одного модуля FPGA. Чаще всего такие системы строят на основе использования одного MCU, который, помимо выполнения расчетных операций, осуществляет выполнение и других функций: передачу управляющих импульсов; прием информации с датчиков с последующей ее обработкой; контроль состояния системы с учетом показателей точности и надежности. Несмотря на возможность использования механизма прямого доступа к памяти (DMA – Direct Memory Access), что позволяет снять определенную нагрузку с основного модуля, программная модель MCU, как правило, не предусматривает возможность параллельного выполнения инструкций.

В отличие от MCU, модули FPGA предоставляют возможность добиться параллельного выполнения инструкций, за счет чего скорость обработки данных FPGA во многих случаях превосходит скорость MCU: например, для задачи построения геометрических примитивов в 2D-пространстве FPGA использует всего 4 тактовых импульса на один цикл интерполяции, в то время как показатели MCU для линейной и, особенно, для круговой интерполяции значительно больше – десятки тактовых импульсов [3]. При расчетах для третьей оси координат (3D-модели) преимущество будет еще больше.

Одновременное же использование двух модулей позволяет получить значительный выигрыш в производительности по сравнению с использованием одного MCU для той же задачи. Основным вопросом остается определение, каким образом оптимальнее всего распараллелить вычислительные задачи между модулями.

При организации параллельного взаимодействия MCU и FPGA также возникает задача синхронизации, которую можно решить методом «рандеву», при котором управляющее устройство блокирует дальнейшие действия до получения сигнала от расчетного модуля [4].

Схематическое распределение задач между модулями показано на рис. 1.

Наибольшая эффективность всех системы будет достигаться при минимальных простоях вычислительных модулей. В данном случае это соответствует одновременному окончанию вычислительных расчетов обоих модулей в момент времени  $t_2$ .

Введем показатель коэффициента полезной работы модуля:

$$x = \frac{U_p}{U_a} \cdot 100\%,$$

где  $U_p$  – количество тактов, затрачиваемых на полезную работу;

$U_a$  – общее количество тактов, на измеряемом участке времени.

Для определения эффективности будем использовать показатель квадратичного отклонения  $\sigma$ , который определяет, насколько равномерно распределена нагрузка между модулями.

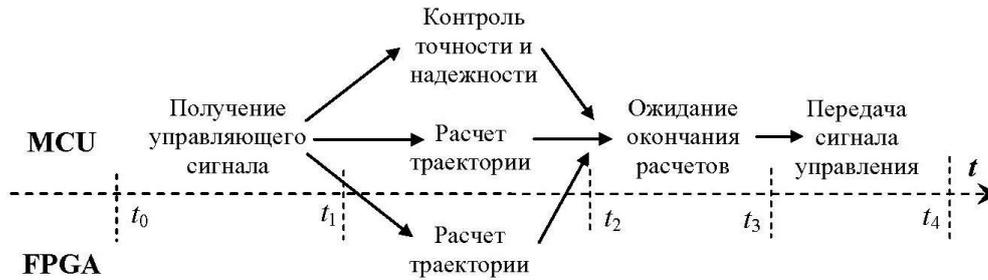


Рис. 1. Распределение задач между модулями

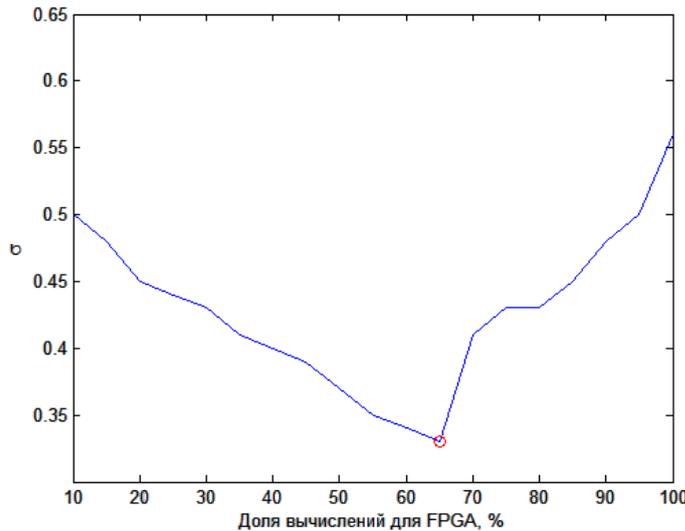


Рис. 2. График зависимости  $\sigma$  от доли вычислительной нагрузки для FPGA

В этом случае оптимальным решением будет выполнение условия:

$$\left\{ \begin{aligned} \sigma &= \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \rightarrow \min; \\ x_i &\rightarrow \max; i = (1, 2); n = 2. \end{aligned} \right.$$

Для экспериментальных исследований в качестве MCU был использован микроконтроллер архитектуры ARM Cortex-M4 на базе отладочного модуля STM32F4 Discovery. В качестве модуля FPGA использована плата Xilinx Spartan 6. Для передачи данных между модулями использовался интерфейс SPI, обеспечивающий скорость передачи данных более 30 Мбит/с.

Полученные опытным путем результаты исследований приведены на рис. 2.

Как видно из рис. 2, оптимальное решение задачи, т.е. выполнение условия (9), выполняется при распределении вычислительных ресурсов между модулями FPGA и MCU равном 65 % и 35 % соответственно.

Таким образом, в работе предложено решение задачи эффективного распределения вычислительных ресурсов в распределенных энергоограниченных компьютерных системах на основе MCU и FPGA модулей, получено математическое выражение для условия оп-

тимального распараллеливания задач диспетчеризации. Рассмотрен пример распараллеливания вычислительных ресурсов между двумя модулями для решения задач оптимального управления станком с 3D позиционированием.

#### СПИСОК ЛИТЕРАТУРЫ

1. Манухин С.В., Сухонос М.И. Алгоритмы оптимизации энергопотребления и повышения эффективности процессоров с масштабированием частоты и напряжения гетерогенного кластера. / Манухин С.В., Сухонос М.И. // – Труды Международной конференции «Параллельные и распределенные вычислительные системы» PDCS 2013 (Украина, Харьков, 13–14 марта 2013 года).
2. Тягунова М.Ю. методы диспетчеризации задач в распределенных компьютерных системах: автореф. дис. на получение научной степени канд. техн. наук: спец. 05.13.05 «Компьютерные системы и компоненты» / М.Ю. Тягунова. – Київ, 2010. – 18 с.
3. Денисов А. Применение FPGA и алгоритмов Брезенхема для повышения быстродействия в системах позиционирования. // Компоненты и технологии. – №10. – 2013. – с. 96–100.
4. Параллельные вычисления. [Электронный ресурс]. URL: [http://ru.wikipedia.org/wiki/Параллельные\\_вычисления](http://ru.wikipedia.org/wiki/Параллельные_вычисления)

**THE OPTIMAL PARALLELIZATION OF DISPATCH PROBLEMS  
IN ENERGY-RESTRICTED DISTRIBUTED COMPUTER SYSTEMS BASED  
ON MCU/FPGA MODULES**

© 2014

*M.P. Musienko*, doctor of technical sciences, professor,  
head of department of information technology and software systems,

*V.Y. Savinov*, a graduate student

*Y.M. Krainyk*, a graduate student

*The Black Sea State University named after Petro Mohyla, Nikolaev (Ukraine)*

*Keywords:* the distributed computing system; the optimal parallelization; scheduling; MCU and FPGA modules; power consumption.

*Annotation:* In this article we consider the problem of energy-efficient distribution of information and resources in a distributed energy-restricted computer systems based on microcontrollers and FPGA through the optimal parallelization of tasks scheduling. An example of computing resources parallelization between two modules for solving optimal control of the machine with 3D positioning is given.