

## МЕТОДЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ

Загуменная Е. В., Радченко С. С., Староверов Р. Н.

*Харьковский Национальный технический университет сельского хозяйства имени Петра Василенка*

*Рассмотрены методы повышения производительности вычислительной техники, а также представлен новый метод, основанный на применении кодов в системе остаточных классов.*

**Постановка проблемы.** В настоящее время существует несколько характеристик вычислительной техники (ВТ), которые разработчики пытаются повысить и благодаря данным показателям усовершенствуется база и характеристика ВТ [3,1].

**Анализ последних исследований и публикаций.** К данным характеристикам можно отнести производительность (быстродействие), отказоустойчивость и надежность ВТ. Одной из наиболее важных характеристик ВТ является производительность. Увеличение требований к данной характеристике постоянно опережает возможности ВТ [2].

**Цель статьи.** Проанализировать методы повышения производительности ВТ [5], предложить новый метод повышения производительности.

**Основные материалы исследования.** Существуют несколько методов повышения производительности такие как: метод, основанный на повышении элементной базы, структурные и программные (математические методы). Первый метод позволяет увеличить тактовую частоту работы ВТ или устройства, интегрировать в одном кристалле процессор, усовершенствовать память и часть устройств ввода вывода, увеличить разрядность шин, памяти и операционных устройств, использовать внутри самого кристалла более совершенствуемые структуры устройства [6]. Второй метод структурный направленный на использовании методов параллельной обработке данных, которая требует такой организации вычислительных средств, когда одновременно выполняется несколько числовых преобразований. Третий метод программный метод или математический связан с созданием новых вычислительных методов решения определенных задач, а также распараллеливание вычислительных процессов. Для реализации максимального повышения производительности этим методом, нужна оперативная и согласованная работа специалистов и разработчиков аппаратного обеспечения, системного программного обеспечения, интегральных микросхем.

Параллельная обработка связанная в первую очередь с применением многопроцессорных архитектур [8]. Параллельная архитектура обладает рядом преимуществ:

- способ построения наиболее высокопроизводительных ВС заключается в применении мультипроцессорных архитектур. Даже производительность самого мощного процессора, можно увеличить, объединив в единую систему несколько процессоров.

- отношение стоимости к производительности при требовании высокой производительности у мультипроцессорных систем, ниже чем у однопроцессорных.

- масштабируемость систем (мультипроцессорная система в связи модульной организацией имеет более широкие возможности для изменения производительности, что обеспечивает широкий диапазон требований в рамках одной архитектуры)

- Мультипроцессорная система позволяет обеспечить большую отказоустойчивость за счет использования структурной избыточности.

Несмотря на данные преимущества при разработке систем параллельной обработке данных возникает ряд недостатков такие как:

- они ограничены в своей области применения;
- отсутствие параллельной математики;
- невозможность распараллеливание некоторых классов задач, на параллельные ветви;
- а также большие затраты на разработку языка и программ.

Эти обстоятельства обуславливают необходимость разработки и применения новых методов повышения производительности ВС, в первую очередь методов основанных на применении кодов в системе остаточных классов (СОК). Первые работы по исследованию СОК были опубликованы в 1955- 1960 году чешскими учеными М. Рамаховым и А. Свободой. Основоболожниками данной теории о построения сверхпроизводительного и высокоустойчивого вычислительного средства стали такие ученые как Акушкин Н.Я., Юдицкий Д.И., Долгов А.И., Евстегнеев В.Г., Торгашов В.А. и др [7].

Основой построения данной системы счисления является теория чисел. Ученные показали, что возможность избыточного кодирования целых чисел в вычислительной технике посредством набора остатков  $\{a_i\}$  от деления этих чисел на взаимопопарно простые натуральные числа  $\{m_i\}, i = \overline{1, n}$  называемые основаниями или модулями СОК.

В результате поиска путей эффективного повышения производительности вычислительных систем интерес возник к "Китайской теореме об остатках" задача восстановления числа  $A_k$  по совокупности его остатков  $\{a_i\}$ .

Коды в классе вычетов являются дальнейшим совершенствованием известных арифметических многоостаточных кодов.

Многоостаточный код имеет вид:

$$A'_k = (A_k, A_k \pmod{m_1}, A_k \pmod{m_2}, \dots, A_k \pmod{m_i}, \dots, A_k \pmod{m_{n-1}}, A_k \pmod{m_n}),$$

то есть  $A'_k = (A_k, a_1, a_2, \dots, a_n)$ ,

$$\text{где } a_i = A_k - \lfloor A_k / m_i \rfloor m_i.$$

При  $\prod_{i=1}^n m_i \geq A_k$ , совокупность остатков  $\{a_i\}$  од-

нозначно определяет операнд  $A_k$  и многоостаточный код принимает вид кода СОК  $A'_k = (a_1, a_2, \dots, a_n)$ , что позволяет реализовать модульные операции по отдельным независим трактам, оперируя только с остатками  $\{a_i\}$  [8-9]. Такое кодирование чисел позволяет построить вычислительное устройство, в котором обработка всех разрядов числа (остатков  $a_i$ ) производится параллельно во времени. В этом случае структурная схема вычислительного устройства в СОК (рис. 1) представляет собой набор отдельных вычислительных устройств, функционирующих независимо друг от друга и параллельно во времени, при чем каждая по своему определенному модулю  $m_i$ .

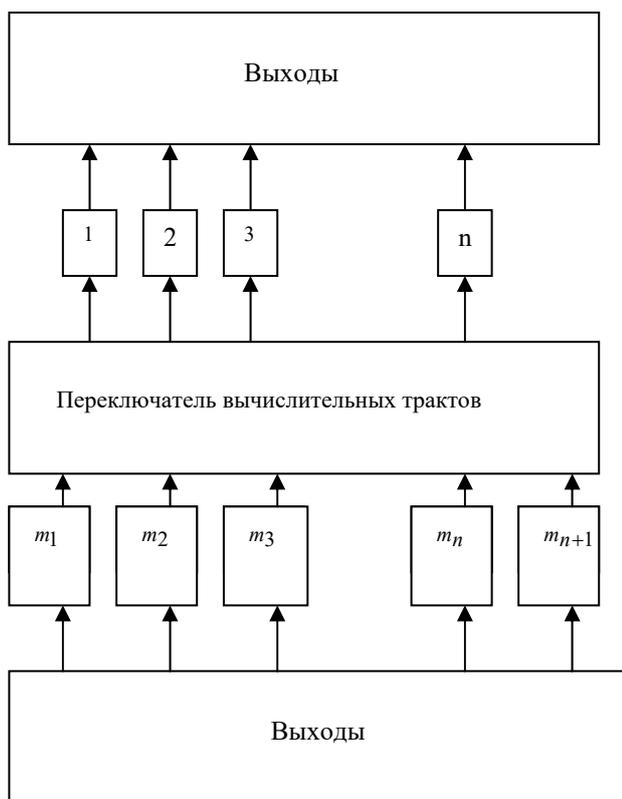


Рисунок 1 - Структурная схема ВУ в СОК

**Выводы.** Таким образом в данной статье было рассмотрено методы повышения производительности, обозначены достоинства параллельной обработки данных, а так же показаны их недостатки. Был предложен новый метод повышения производительности, который связан с использованием специального кодирования чисел так называемой системой остаточных классов.

## Список использованных источников

1. Акушский И. Я. Машинная арифметика в остаточных классах / И. Я. Акушский, Д. И. Юдицкий. – М.: Советское радио, 1968. – 440 с.
2. Жихарев В. Я. Методы и средства обработки информации в непозиционной системе счисления в остаточных классах / В. Я. Жихарев, Я. В. Илюшко, Л. Г. Кравец, В. А. Краснобаев. – Ж.: Волынь, 2005. – 219 с.
3. Яськова Е. В. Методы обработки информации в модулярной арифметике / Е. В. Яськова // Радиоэлектроника и молодежь в XXI веке: тез. докл. XIII международного молодежного форума, Харьков, 30 марта – 1 апреля 2009 г. – С. 187.
4. Малашевич Б. М. Разработка вычислительной техники в Зеленограде. Неизвестные супер-ЭВМ / Б. М. Малашевич // Электроника, наука, технология, бизнес. – 2004. – № 2. – С. 72 – 76.
5. Яськова Е. В. Повышение отказоустойчивости информационно-управляющей системы АСУТП сельскохозяйственного производства на основе использование модулярной арифметики / Е. В. Яськова // Молодежь и сельскохозяйственная техника в XXI веке: тез. докл. III – го международного форума молодежи. – Харьков, 4 - 6 апреля 2007 г. – 143 с.
6. Коляда А. А. Модулярные структуры конвейерной обработки цифровой информации / А. А. Коляда, И. Т. Пак. – Минск: Наука, 1992. – 256 с.
7. Труды Юбилейной Международной научно-технической конференции "50 лет модулярной арифметике", Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ. 550с.
8. Watson R.W. Residue arithmetic and Reliable Computer Design / R. W. Watson, C. W. Hastings. – Washington.: Spartan Books, 1967. – 128 p.
9. Долгов А. И. Диагностика устройств, функционирующих в системе остаточных классов / А. И. Долгов. – М.: Радио и связь, 1982. – 64с.

## Анотація

### МЕТОДИ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ОБЧИСЛЮВАЛЬНИХ ЗАСОБІВ

Загуменна К. В., Радченко С. С., Староверов Р. М.

*Розглянуто методи підвищення продуктивності обчислювальних засобів, а також запропонований метод, котрий заснований на застосуванні кодів у системі залишкових класів.*

## Abstract

### METHODS TO IMPROVE THE PERFORMANCE OF COMPUTATIONAL TOOLS

K. Zagumenna, S. Radchenko, R. Staroverov

*The methods of increasing the computation performance, and presents new method based on the use of codes in the system of residual classes.*