

УДК 681.322

ВАРИАНТ АРХИТЕКТУРЫ ЗАЩИЩЕННОГО МИКРОПРОЦЕССОРА НА ОСНОВЕ СИСТЕМЫ ОСТАТОЧНЫХ КЛАССОВ

Магомедов Шамиль Гасангусейнович, кандидат технических наук, Дагестанский государственный технический университет, 367015, Российская Федерация, Республика Дагестан, г. Махачкала, пр. Имама Шамиля, 70, e-mail: msgg@list.ru

Обосновано, что архитектура традиционных процессоров, ориентированная на минимизацию ресурсных показателей (времени обработки, энергопотребления), не дает возможность эффективно решать вопросы защиты информации. Поэтому автором предлагается специальный процессор (СП), с особым вариантом архитектуры и алгоритмами обработки данных, позволяющими обеспечить их информационную безопасность. Описываемое аппаратно-алгоритмическое решение СП основано на использовании счисления в системах остаточных классов. При этом в оперативной памяти СП хранится и обрабатывается не число «х», а наборы его модулей $(x_1, x_2, \dots, x_n)$. Поэтому хищение злоумышленником набора $(x_1, x_2, \dots, x_n)$ не позволяет ему восстановить число «х». Это обеспечивает значительное повышение уровня информационной безопасности, хотя и ценой затрат дополнительных ресурсов. В статье приведена предлагаемая архитектура СП, детально описаны алгоритмы обработки на нем данных. Указаны особенности алгоритмов в случае использования многоядерных СП.

Ключевые слова: архитектура микропроцессора, система остаточных классов, защита информации, оперативная память, командные циклы, конвейер, суперскалярность, модулярная арифметика

MICROPROCESSOR ARCHITECTURE OPTION BASED ON RESIDUAL CLASSES

Статья поступила в редакцию 04.11.2013, в окончательном варианте 28.11.2013.

Magomedov Shamil G., Ph.D. (Engineering), Dagestan State Technical University, 70 proezd Imama Shamilya, Makhachkala, Republic of Dagestan, 367015, Russian Federation, e-mail: msgg@list.ru

We propose a possible alternative structure is a microprocessor core target setting is to ensure the protection of data in information processing - in contrast to existing microprocessors aimed at minimizing resource indicators (processing time, energy consumption). As a basic framework to accomplish the task you are prompted to use the unit value in the systems of residual classes. The proposed scheme enables processing microprocessor protect them in RAM in the processing of these data. The general operation of the proposed microprocessor technology, listed its distinctive features.

Keywords: architecture of micro processor, the system of residual classes, information security, RAM, command cycles, the pipeline, superscalar.

Ключевым компонентом любых современных систем обработки и передачи данных являются процессорные устройства (ПУ). Сложность современных (ПУ) возросла настолько, что в условиях отсутствия исходных схем ПУ их практически невозможно контролировать даже на уровне национальных институтов развитых стран. Так, число транзисторов в чипе процессора Pentium Nehalem составляет порядка 200 миллионов. В таких огромных массивах объектов можно «скрыть» достаточно эффективные средства хищения, уничтожения или искажения данных, передачи их по желаемому адресу, совершения других несанкционированных действий с данными. При этом за разумное время выявить эти «закладки» фактически невозможно.

В настоящее время происходит обострение конкуренции в сфере информационных технологий и постепенное перемещение противостояния между отдельными государствами и группами государств, крупными транснациональными корпорациями в сферу информационного противоборства. Ярким примером этого являются «утечки» информации о прослушивании со стороны спецслужб США переговоров лидеров других стран. Такие факты, а известности предано, очевидно, далеко не все, дают определенные основания предположить, что несанкционированные средства и каналы съема информации могут появиться в программно-аппаратных продуктах многих ведущих мировых производителей ИТ-средств. Для России эта проблема особенно существенна в связи с отсутствием собственных производств ПУ с достаточно высокими характеристиками. Поэтому использование зарубежных ПУ традиционной архитектуры в российских системах связи и обработки информации потенциально создает угрозы информационной безопасности (ИБ).

Одним из возможных путей решения проблемы ИБ при использовании ПУ является создание национальных вариантов ИТ-продуктов (в частности ПУ), либо разрешение использовать только контролируемые со стороны государства типы ПУ. Целью данной работы является обоснование целесообразности использования специального процессора (СП), в котором имеются механизмы защиты от возможного хищения обрабатываемых данных. В качестве основы для обработки данных в СП предлагается использование систем остаточных классов. В доступной литературе работ по указанной тематике мало [1, 2, 6, 7].

Прежде всего, проведем анализ возможностей обеспечения защиты информации для архитектуры типового ПУ, на основе которой спроектированы процессоры четвертого и пятого поколений. Структурная схема такого ПУ приведена на рис. 1. По функциональному назначению ПУ относится к классу операционных устройств. В его структуре выделяют две части: управляющий блок (устройство управления) и операционный блок (ОБ). Операционный блок служит для обработки данных. Управляющий блок выполняет операции выборки, декодирования и вычисления адресов операндов, а также генерирует последовательности микрокоманд для каждой команды. Подробное описание структуры ПУ и процесса его работы можно найти в [3, 4, 5, 10]. С позиций ИБ использование такого процессора для обработки данных имеет следующие недостатки.

1. Данные, размещаемые в оперативной памяти, во временных файлах, свопфайлах, представлены в открытой, не преобразованной форме. При их хищении или несанкционированном ознакомлении это может приводить к утечке информации. В частности, поскольку после завершения обработки информации на ПУ содержание оперативной памяти и памяти на жестком диске, используемой для свопинга, обычно не затирается. Поэтому несанкционированное ознакомление с содержимым оперативной памяти или жесткого диска после окончания процедур обработки «закрытой» информации может привести к хищению данных.

2. Системные файлы, используемые типовым ПУ и размещаемые в постоянной памяти, потенциально доступны злоумышленнику, который может внести в них несанкционированные изменения. Эти изменения могут стать источником нарушения ИБ, неадекватной обработки данных, аварийных ситуаций.

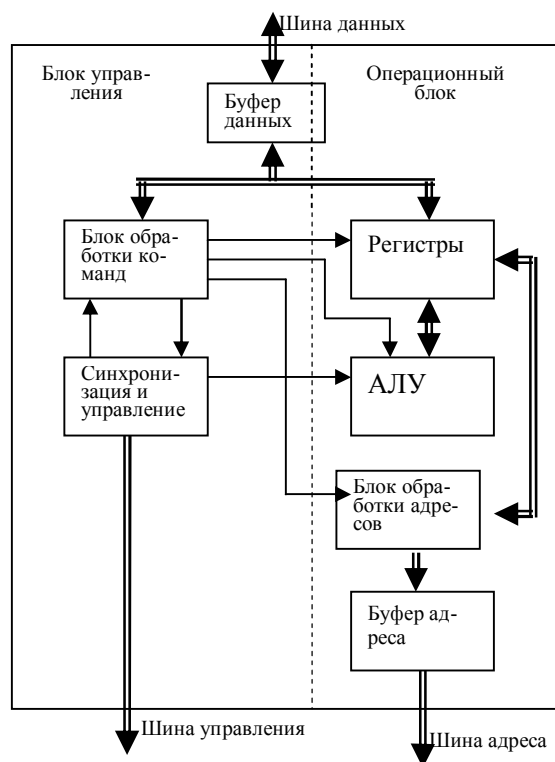


Рис. 1. Структурная организация типового ПУ

Теоретически для решения перечисленных выше проблем возможно использование механизма шифрования данных на уровне процессора и даже встраивание этого механизма непосредственно в процессор. Однако существующие современные методы шифрования достаточно сложны и громоздки, что при включении их в структуру процессора приведет к существенному усложнению его работы.

Кроме того, неясно, как может быть решена в этом случае проблема ключей шифрования. Если ключ шифрования заложить непосредственно в процессор при его проектировании и изготовлении, то этот неизменный ключ при многократном его использовании может быть вскрыт злоумышленником – тогда вся обрабатываемая информация будет ему потенциально доступна. Если же применить механизмы периодического обновления ключей, то технология использования и эксплуатации ПУ существенно усложнится. В типовом ПУ основное внимание уделено обеспечению максимального быстродействия выполнения программ. При этом процессы обеспечения ИБ обработки информации вообще не выделены в качестве самостоятельной задачи, хотя при работе с конфиденциальной информацией высокой ценности это может быть важно.

Следует отметить, что в процессорах Pentium предусмотрен специальный режим обработки данных, имеющий отношение к вопросам обеспечения ИБ, – защищенный режим. Однако он решает только проблемы, связанные с неадекватной обработкой данных, то есть только после того, как проблема уже возникла, но не в процессе текущей обработки информации. Для решения указанных проблем ниже предлагается структурная схема СП, позволяющая обеспечить такой режим обработки данных, при котором данные обрабатываются после их преобразования на основе набора простых чисел, входящих в базис системы остаточных классов (СОК). Поэтому потенциальный злоумышленник без знания этого базиса даже после хищения

данных не может ознакомиться с содержанием данных. Базис СОК формируется в СП для использования в течение любого приемлемого периода.

Приведем базовые понятия, связанные со счислением в СОК. Система остаточных классов, другое название «модулярная арифметика» [8], – непозиционная система счисления, в которой представление числа основано на понятии вычета и китайской теореме об остатках. СОК определяется набором взаимно простых модулей $(P_1, P_2, \dots, P_n)$, называемых базисом; так что каждому целому числу x из отрезка $[0, M-1]$, где $M = P_1 \cdot P_2 \cdot \dots \cdot P_n$, ставится в соответствие набор вычетов $(x_1, x_2, \dots, x_n)$, где $x_i \equiv x \pmod{P_i}$ ($i = \overline{1; n}$). Обычно в качестве оснований P_i , входящих в базис $(P_1, P_2, \dots, P_n)$, берутся простые числа. При этом китайская теорема об остатках [8] гарантирует однозначность представления для всех чисел из отрезка $[0, M-1]$. Более того, само число « x » может быть найдено на основе соотношения

$$x = x_1 * e_1 + \dots + x_n * e_n \pmod{M}, \quad (1)$$

где

$$e_i = M / P_i * (M / P_i)^{-1} \pmod{P_i}, \quad (1 \leq i \leq n). \quad (2)$$

При формировании предлагаемой ниже схемы используются особенности структуры ПУ для одной из удачных архитектур, а именно микропроцессоров Pentium фирмы Intel пятого поколения [4, 9, 10]. Одно из основных отличий ПУ пятого поколения от предыдущих – архитектура суперскалярности, при которой процессор имеет несколько конвейеров. Она присуща не только Pentium, но и ПУ того же поколения других известных производителей: AMD K5, Cyrix M1. В предлагаемой ниже схеме СП также используется архитектура суперскалярности, но с учетом особенностей СОК. Именно наличие скалярности позволяет провести отдельные вычисления по разным модулям на разных конвейерах. В предлагаемой архитектуре СП имеются две внутренние шины, связанные с внутренними регистрами: одна шина предназначена для передачи и размещения адресных данных, а вторая – для вычислительных данных (рис. 2).

Вначале при активизации СП формируется и запоминается база простых чисел в блоке «Формирование оснований СОК». Затем процесс функционирования СП, как и в случае типового ПУ по рис. 1, разбивается на командные циклы, в течение каждого из которых выполняются следующие команды:

- 1) выбирается очередная порция данных и команда из буфера «команд и данных»;
- 2) данные расшифровываются и размещаются во внутренних регистрах СП;
- 3) очередная команда анализируется и декодируется в блоке обработки команд;
- 4) декодированная команда, если она связана с использованием арифметическо-логического устройства (АЛУ), также размещается во внутренних регистрах вместе с данными;
- 5) АЛУ выбирает из внутренних регистров выполняемую команду, данные и организует соответствующие вычисления. Процесс вычислений в АЛУ реализован как обобщение двухконвейерной схемы процессора Pentium. При этом **u**-конвейер основной – он работает с основным потоком инструкций. В нем происходит обработка всего набора основных инструкций вычислительными и логическими командами, командами перехода; сборка всех промежуточных результатов, выполненных на **v**-конвейерах, в частности, вычисления по отдельным модулям базы. Результат вычислений сохраняется во внутренних регистрах СП, а затем после завершения всех вычислений происходит шифрование данных с помощью СОК и их размещение в буфере данных;

- 6) в блоке обработки адресов по результатам работы блока обработки команд и, возможно, АЛУ (для команд условного перехода) с участием блока управления формируется адрес размещения результатов выполнения команды (он размещается в буфере адреса) и адрес следующей выполняемой команды. Этот адрес может отличаться от порядкового при выполнении команд перехода;

7) по шине адреса устанавливается адрес, куда будут переданы результаты через шину данных. Затем по команде управления в определенный момент времени этот адрес передается через шину адресов;

8) блок управления, после формирования всех адресов и данных, организует одновременно передачу адреса размещения результатов через шину адреса и данные через шину данных;

9) по шине управления выбирается следующая выполняемая команда. Для обеспечения тактовой синхронности выполнения всех операций блок управления использует синхронизатор;

10) блок управления также следит за возможными управляющими сигналами от других ПУ-устройств и сигналами, связанными с определенными ситуациями в вычислительной системе и процессоре, такими как прерывание. Если такие сигналы имеются, то блок управления реагирует соответствующим образом.

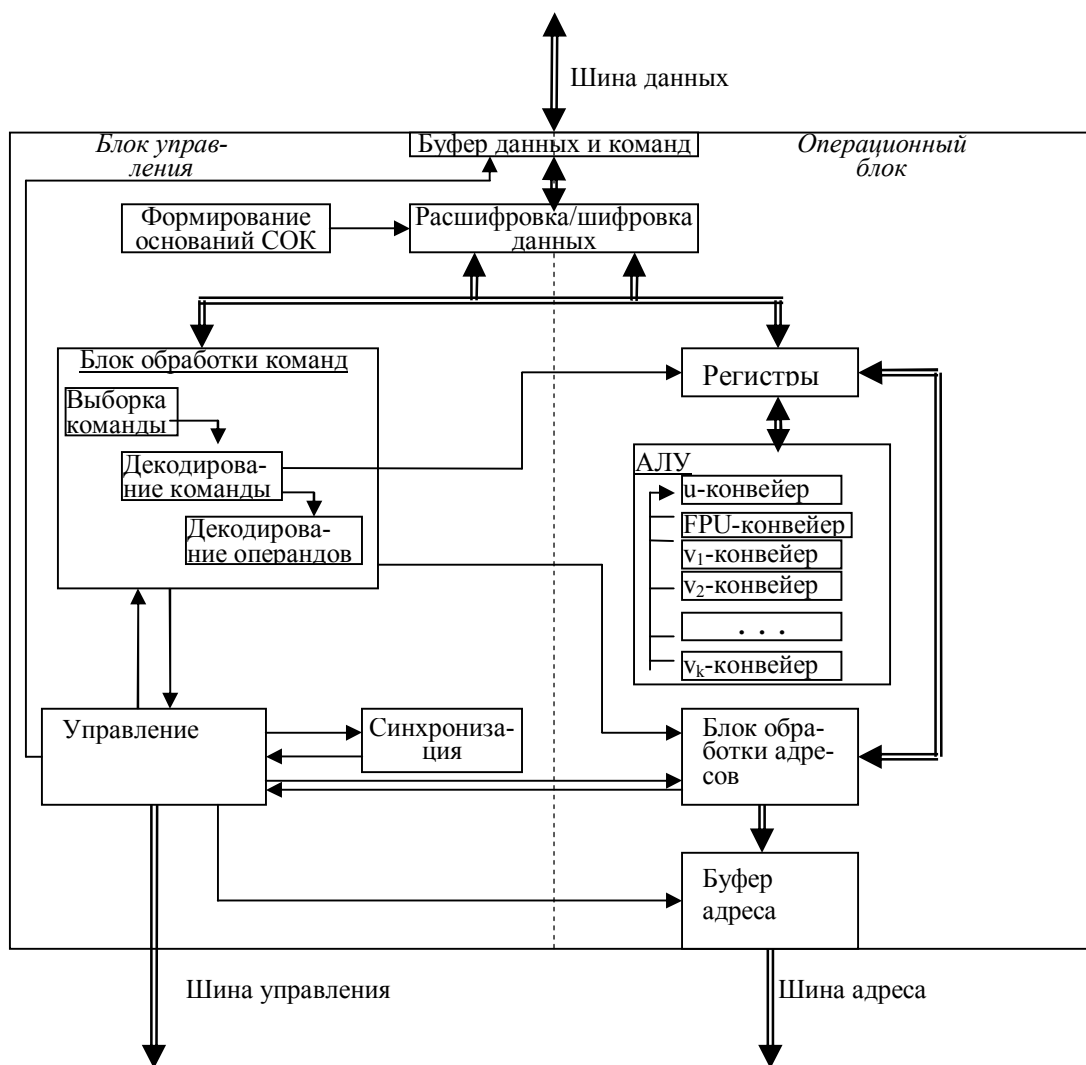


Рис. 2. Структурная схема организации СП на основе СОК

При наличии многоядерных СП процесс обработки команд проводится аналогичным образом. При загрузке в оперативную память очередного программного модуля вначале вырабатывается набор из n модулей, где n – число ядер в процессоре, которые могут быть использованы для проведения вычислений. При этом диапазон изменения простых чисел зависит от вида использования СП. Если многоядерный процессор применяется для проведения параллельных независимых вычислений, то в этом случае простые числа должны быть сравнимы с величинами данных, обрабатываемых каждым ядром. Если же проводятся арифметические вычисления в модульной арифметике с использованием китайской теоремы об остатках, то в этом случае размеры простых чисел должны быть порядка величин остатков.

Перечислим теперь некоторые особенности реализации отдельных функций типового СП применительно к предлагаемой архитектуре, ориентированной на использование СОК.

1. Выполнение любой команды в типовом ПУ можно разбить на два цикла: цикл выборки команды и цикл ее выполнения. В цикле выборки команды устройство управления (УУ) производит чтение кода операции и его декодирование. В цикле выполнения команды в соответствии с типом реализуемой команды осуществляется определение адресов операндов, участвующих в операции; непосредственно выполнение команды; сохранение результатов операции. Для предлагаемой архитектуры вышеприведенная схема претерпевает определенные изменения. Поскольку важное место в работе СП занимают действия в модульной арифметике, то в цикле выборки команды проводится дополнительный анализ. В частности выясняется, является ли операнд операцией в модульной арифметике или нет. В случае модульной операции необходимо ее организовать, что для многоядерного процессора предполагает использование нескольких микроядер – по одному микроядру на каждое основание, входящее в базис модуля. Также добавляются специальные микрокоманды, позволяющие на основе результатов вычислений по отдельным базисам основания сформировать конечный результат. При этом в процессе выполнения всех вычислительных операций происходит «усечение» результата по соответствующему модулю. Укажем также на проблему оптимизации выбора состава и последовательности выполнения микрокоманд ПУ, для решения которой могут быть использованы методы, аналогичные [1].

2. Одним из основных компонентов типового ПУ, связанным непосредственно с обработкой данных, является АЛУ. Наибольшее распространение получили АЛУ, где основным типом данных являются целые числа, характеризующиеся простотой реализации и высокой скоростью обработки информации. Специализированные большие интегральные схемы (БИСы) используются для обработки данных, представленных в виде чисел с плавающей запятой. Предлагаемая нами архитектура СП ориентирована на использование целых чисел в модульной арифметике. Непосредственная форма представления этих чисел (двоичная, двоично-десятичная или иная) допускает использование сопроцессора для работы с числами, имеющими плавающую запятую.

3. Еще одним важным новшеством в ПУ пятого поколения явилось появление блока «предсказания ветви (продолжения)» – branch prediction. Он предназначен для предсказания возможных переходов на основе обработки статистики о последних 256 переходах. В предлагаемой нами архитектуре СП функции предсказателя расширены. Помимо предсказания возможного продолжения, данный блок контролирует процессы работы отдельных конвейеров с точки зрения режима их использования – часть из них могут применяться для выполнения параллельных микрокоманд (как и в процессорах Pentium), а часть – для выполнения вычислений в соответствии со схемами систем счисления в остаточной системе счисления. 4. Многосекционный сумматор, введенный в архитектуру ПУ пятого поколения, в предлагаемой нами архитектуре также имеет особенность использования – сумматор должен дополнительно выполнять функции модульного суммирования результатов вычислений по отдельным основаниям базы модулей.

Описанная выше схема работы СП на основе СОК позволяет хранить в оперативной памяти и обрабатывать не число « x », а наборы его модулей $(x_1, x_2, \dots, x_n)$. Поэтому хищение злоумышленником набора $(x_1, x_2, \dots, x_n)$, вообще говоря, не позволяет ему восстановить число « x » без знания базиса $(P_1, P_2, \dots, P_n)$. Базис же $(P_1, P_2, \dots, P_n)$ вырабатывается непосредственно внутри процессора, хранится внутри него и недоступен злоумышленнику. Поэтому даже наличие у злоумышленника того или иного варианта доступа к оперативной памяти, к временным или своп файлам не позволяет ему воспользоваться полученной информацией. Именно указанное свойство предлагаемого аппаратно-программного решения и обеспечивает защиту оперативных данных в процессе их обработки на СП. Подчеркнем, что задачи обеспечения ИБ данных в процессе их передачи по каналам связи от СП к внешним устройствам выходят за рамки настоящей статьи.

Рассмотренное выше преобразование может быть выполнено и для системных файлов, которые используются только СП, причем исходные коды этих файлов могут храниться в указанном преобразованном виде. Это позволяет защитить их от модификации, подмены и других злоумышленных действий, так как измененный файл СП не сможет адекватно преобразовать в программный код.

Понятно, что предлагаемые аппаратно-программные решения проигрывают по быстродействию традиционным ПУ – это является «платой» за лучшее качество обеспечения ИБ. В связи с этим повторим, что использование в СП нескольких ядер позволяет резко ускорить обработку информации. Сравнение скоростей обработки информации для процессоров с традиционной архитектурой и предлагаемых СП потенциально может быть выполнено с использованием имитационного моделирования на ЭВМ, например на основе пакета MatLab. Для этого необходимо, чтобы продолжительности элементарных операций на традиционных ПУ и СП были известны или могли быть оценены. Итак, сделаем следующие **выводы**.

1. Предлагаемое аппаратно-программное решение для СП позволяет эффективно решать задачи ИБ обрабатываемой информации за счет применения алгоритмов, основанных на СОК.
2. Такие решения целесообразны для использования, прежде всего, в специальных устройствах, для которых вопросы ИБ имеют ключевое значение.
3. Применение в рамках рассматриваемых в статье аппаратно-программных решений многоядерных процессоров позволяет значительно ускорить обработку информации.

Список литературы

1. Исмаилов Ш.-М. А. Разрядно параллельный алгоритм и структура устройства умножения двоичных чисел / Ш.-М. А. Исмаилов, Ш. Г. Магомедов // Информационные и телекоммуникационные системы: информационные технологии в научных и образовательных процессах : мат.-лы V Регион. науч.-техн. конф., 18–20 сент. 2009 г. – Махачкала : ДНЦ РАН, 2009.
2. Краснобаев В. А. Алгоритм реализации операции модульного умножения в системе остаточных классов / В. А. Краснобаев, В. П. Ирхин // Электронное моделирование. – 1993. – № 5. – С. 20–273. Ливенцов С. Н. Основы микропроцессорной техники : учеб. пос. / С. Н. Ливенцов, А. Д. Вильнин, А. Г. Горюнов. – Томск : Изд-во Томского политехнического университета, 2007. – 118 с.
3. Могнонов П. Б. Организация микропроцессорных систем : учеб. пос. – Улан-Удэ : Изд-во ВСГТУ, 2003. – 174 с.
4. Старуев А. В. Метод повышения эффективности использования ресурсов ЭВМ / А. В. Старуев // Прикаспийский журнал: управление и высокие технологии. – 2013. – № 1 (21). – С. 12–15.
5. Червяков Н. И. Методы и алгоритмы округления, масштабирования и деления чисел в модулярной арифметике / Н. И. Червяков и др. // Модулярная арифметика : мат.-лы междунар. науч. конф. – Режим доступа: <http://www.computer-museum.ru/histussr/sokconf0.htm> (дата обращения 25.11.2013), свободный. – Загл. с экрана. – Яз. рус.

7. Червяков Н. И. Принципы построения модулярных сумматоров и умножителей / Н. И. Червяков, И. В. Дьяченко // Модулярная арифметика : мат-лы междунар. науч. конф. – Режим доступа: <http://www.computer-museum.ru/histussr/sokconf0.htm> (дата обращения 25.11.2013), свободный. – Загл. с экрана. – Яз. рус.
8. Червяков Н. И. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н. И. Червяков, С. А. Ряднов, П. А. Сахнюк, А. В. Шапошников. – Москва : ФИЗМАТЛИТ, 2003. – 288 с.
9. Carmean D. Inside the Pentium 4 Processor Microprocessor Microarchitecture / D. Carmean // Intel Development Forum. – 2000.
10. Grisp J. Introduction to Microprocessors and Microcontrollers / J. Grisps. – 2nd ed. – Newnes, 2004. – 278 p.

References

1. Ismailov Sh-M. A., Magomedov Sh. G. Razryadno parallelnyy algoritm i struktura ustroystva umnozheniya dvoichnykh chisel [Bit parallel algorithm and the structure of a binary numbers multiplication]. *Informatsionnye i telekommunikatsionnye sistemy: informatsionnye tekhnologii v nauchnykh i obrazovatelnykh protsessakh: materialy V Regionalnoy nauchno-tekhnicheskoy konferentsii* [Information and telecommunication systems: information technologies in scientific and educational processes: Proceedings of the V Regional Scientific and Technical conference], 18–20 September 2009. Mahachkala, 2009.
2. Krasnobaev V. A., Irkhin V. P. Algoritm realizatsii operatsii modulnogo umnozheniya v sisteme ostatochnykh klassov [Algorithm implementing modular multiplication operation in the system of residual classes]. *Elektronnoye modelirovaniye* [Electronic modeling], 1993, no. 5, pp. 20–27.
3. Liventsov S. N., Vilnin A. D., Goryunov A. G. Osnovy mikroprotsessornoy tekhniki [Fundamentals of microprocessor technology]. Tomsk, Tomsk Polytechnical University Publ., 2007. 118 p.
4. Mogonov P. B. Organizatsiya mikroprotsessornykh sistem [Organization of microprocessor systems]. Ulan-Ude, 2003. 174 p.
5. Staruev A. V. Metod povysheniya effektivnosti ispolzovaniya resursov EVM [A method for increasing the efficiency of computer resource use]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2013, no. 1 (21), pp. 12–15.
6. Chervyakov N. I. et al. Metody i algoritmy okrugleniya, masshtabirovaniya i deleniya chisel v modulyarnoy arifmetike [Methods and algorithms of rounding, scaling and dividing numbers in modular arithmetic]. *Modulyarnaya arifmetika: materialy mezhdunarodnoy nauchnoy konferentsii* [Modular arithmetic: Proceedings of the International Scientific Conference]. Available at: <http://www.computer-museum.ru/histussr/sokconf0.htm> (accessed 25.11.2013).
7. Chervyakov N. I., Dyachenko I. V. Printsipy postroeniya modulyarnykh summatorov i umnozhitel'ey [Principles of construction of modular adders and multipliers]. *Modulyarnaya arifmetika: materialy mezhdunarodnoy nauchnoy konferentsii* [Modular arithmetic: Proceedings of the International Scientific Conference]. Available at: <http://www.computer-museum.ru/histussr/sokconf0.htm> (accessed 25.11.2013).
8. Chervyakov N. I., Ryadnov S. A., Sakhnyuk P. A., Shaposhnikov A. V. Modulyarnye parallelnye vychislitelnye struktury neyroprotsessornykh sistem [Modular parallel computing structures of neuroprocessor systems]. Moscow, 2003. 288 p.
9. Carmean D. Inside the Pentium 4 Processor Microprocessor Microarchitecture. *Intel Development Forum*, 2000.
10. Grisp J. *Introduction to Microprocessors and Microcontrollers*. 2nd ed. Newnes, 2004. 278 p.