УДК 004.4

Исследование пиковой производительности современных микропроцессоров

Д.И. Слинкин, П.С. Зубковский

Научно-исследовательский институт системных исследований РАН, г. Москва,

dima_s@cs.niisi.ras.ru, zubkovsky@cs.niisi.ras.ru

***Аннотация* — Для современных микропроцессоров оценка производительности на основе тактовой частоты теряет смысл в силу наличия нескольких ядер и арифметических сопроцессоров. Для описания быстродействия используется термин «пиковая производительность» – количество операций с плавающей точкой в секунду. В НИИСИ РАН разработано семейство микропроцессоров с архитектурой КОМДИВ. Статья посвящена задаче демонстрации максимальной производительности микропроцессоров во время испытаний. Пиковая производительность является теоретической величиной, скорость работы на реальной задаче всегда окажется ниже. Производительность ядра микропроцессора на целочисленных операциях меньше производительности специализированного сопроцессора на операциях с вещественными или комплексными числами. Поэтому интерес представляют программы, использующие арифметический сопроцессор, и способы их оптимизации. В статье разбирается несколько способов измерения производительности микропроцессоров. В заключении предлагается метод подтверждения соответствия опытного образца требованию технического задания.**

***Ключевые слова* — пиковая производительность, микропроцессор, арифметический сопроцессор, вычислительная эффективность, BLAS, LINPACK.**

I. Введение

После изготовления опытных образцов микро–процессоров проводятся их испытания. Одной из задач испытаний является подтверждение соответствия максимальной производительности опытного образца и пиковой производительности, указанной в техническом задании. Проблема заключается в том, что пиковая производительность – это теоретическая величина, недостижимая на практике. Одного лишь теоретического расчета недостаточно для убедительного подтверждения правильности принятых технических решений. Требуется демонстрация, проводимая при помощи программного теста, в ходе выполнения которой достигается высокий процент от пиковой производительности, а также теоретическое обоснование полученного результата. Рассмотрим несколько подходов, которые могут быть использованы для решения поставленной задачи.

II. Теоретический расчёт пиковой производительности

Конвейерная организация вычислений позволяет разбить выполнение одной инструкции микропроцессора на несколько этапов. Скорость исполнения инструкции обратно пропорциональна времени выполнения самого медленного этапа. Конвейер существенно повышает количество инструкций, исполняемых за такт (IPC – instructions per clock), и, следовательно, производительность микропроцессора. Суперскалярность предполагает параллельное выполнение команд с использованием нескольких функциональных конвейеров [1]. Так, например, после стадии декодирования арифметическая инструкция может быть направлена либо в конвейер целочисленной арифметики, либо в конвейер вещественной и комплексной арифметики. Конвейерная организация и суперскалярность увеличивают эффективность работы процессора, но при написании высокопроизводительных программ должны быть учтены особенности конкретного микропроцессора.

Производительность специализированного сопроцессора выше, чем у ядра микропроцессора, поэтому наибольший интерес представляют задачи, решаемые сопроцессором [2]. Известно о применении графических контроллеров (GPU – graphics processing unit) при тестировании производительности. На GPU переносится часть вычислительной задачи при параллельной работе с микропроцессором.

Для высокопроизводительных алгоритмов должна учитываться зависимость по данным, когда результаты выполнения одной команды являются операндами для другой. Например, длительность команды «умножение с накоплением и вычитанием комплексных чисел» для 64-разрядных чисел составляет 7 циклов процессора. Если в течении этого времени произойдет обращение к задействованному командой регистру, то возникнет аппаратная задержка. Микропроцессор будет ожидать завершения команды, что приведет к снижению производительности.

Упрощенная схема микропроцессора 1890ВМ8Я с архитектурой КОМДИВ приведена на рис. 1.

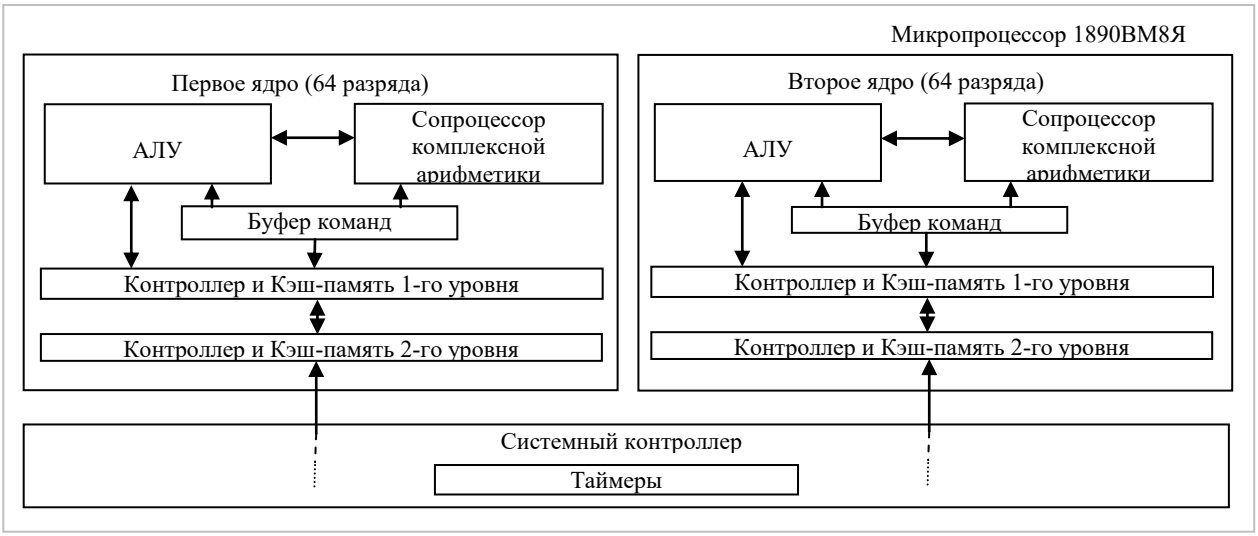

Рис. 1. Схема двухъядерного микропроцессора с архитектурой КОМДИВ

Теоретический расчет пиковой производительности микропроцессора производится по формуле:

$$R_{peak} = N_{clock} * N_{cores} * N_{sections} * f_{cpu}, \quad (1)$$

где R_{peak} – теоретическая оценка пиковой производительности, N_{clock} – число арифметических операций, выполняемых одной секцией сопроцессора за такт, N_{cores} – количество ядер микропроцессора, $N_{sections}$ – число вычислительных секций сопроцессора, f_{cpu} – тактовая частота микропроцессора.

III. Измерение тактовой частоты микропроцессора

Тактовая частота микропроцессора f_{cpu} формируется внутренними генераторами синхросигналов. Если разработчики предусмотрели служебный вывод СБИС, предназначенный для измерения внутренних частот микропроцессора, то f_{cpu} можно продемонстрировать при помощи осциллографа. Так как внутренняя тактовая частота микропроцессора высока, то на служебный вывод подается пониженная частота, которую затем умножают на известный коэффициент.

Программный способ определения f_{cpu} заключается в использовании таймеров и регистра-счетчика в составе микропроцессора. Этот регистр увеличивается на единицу за каждый такт микропроцессора. Программа считывает значение регистра-счетчика в начале и в конце некоторого интервала времени, а затем определяет тактовую частоту. После измерения f_{cpu} пиковая производительность рассчитывается по формуле (1).

Ответственные узлы микропроцессора обязательно проверяются на ранних этапах проектирования, например, во время отладки поведенческой модели [3]. Поэтому, если в процессе испытаний подтверждено, что опытный образец микропроцессора функционирует на частоте f_{cpu}, то реализация соответствующей величины N_{clock} практически гарантирована. Однако не существует методов непосредственной демонстрации f_{cpu}.

IV. Программа определения максимальной производительности

Тестирование микропроцессоров КОМДИВ производится на платах функционального контроля, имеющих в своем составе все устройства и внешние интерфейсы, что и процессорные модули, планируемые к серийному производству. Опытные образцы микропроцессоров устанавливаются в контактирующие устройства и могут легко заменяться. Платы функционируют на реальной (не пониженной) частоте и позволяют запускать различные ОС и программные тесты.

Для подтверждения соответствия величины N_{clock} расчётному значению была разработана «программа определения максимальной производительности». Были выбраны команды сопроцессора векторной арифметики: cmaddsub – «умножение с накоплением и вычитанием комплексных чисел (бабочка Фурье)» и mvmadd – «умножение вещественной матрицы 2x2 на вещественный 2-вектор с накоплением». Оценка производительности ядра процессора на операциях с фиксированной точкой проводилась на паре операций madd – «умножение с накоплением с учетом знака» и sub – «вычитание с учетом знака». Процессоры с архитектурой КОМДИВ запускают эти команды в разных потоках, поэтому они выполняются параллельно. Суммарная вычислительная сложность пары команд madd и sub равна трем операциям. Параметры команд приведены в табл. 1.

Таблица 1

Число арифметических операций за такт

№	Команда	Точность	N_{clock}
1	cmaddsub.s	Одинарная	20
2	cmaddsub.d	Двойная	10
3	mvmadd.s	Одинарная	16
4	mvmadd.d	Двойная	8
5	madd	Одинарная	2
6	sub	Одинарная	1

Например, при тактовой частоте 800 МГц для команды cmaddsub.s теоретическая оценка пиковой производительности одного векторного сопроцессора составит: R_{peak} = 800 МГц * 20 опер./такт = 16 Гфлопс (Giga Floating Point Operations per Second). Для пары команд madd и sub теоретическая оценка пиковой производительности для одного ядра процессора составит: R_{peak} = 800 МГц * 3 опер./такт = 2,4 Гопс (Giga Operations per Second).

Функции, реализующие блоки однотипных команд, были написаны на языке ассемблера, а управляющая и интерфейсная части на языке Си. Время выполнения каждого блока оценивалось с использованием таймеров в составе системного контроллера. В начале своей работы программа определяла тактовую частоту процессора, а после выполнения каждого блока команд производила расчет пиковой производительности по формуле:

$$R_{test} = Sum/T ,$$

где R_{test} – производительность, определенная тестом, Sum – число арифметических операций, T – время выполнения теста.

Для каждого блока арифметических операций рассчитывается процент соответствия между измеренной производительностью и ее теоретической величиной. Удалось достичь результата, близкого к 100% для «бабочки Фурье» с 64-разрядными комплексными числами и 98% для целочисленных и вещественных операций.

Достоинствами данного метода являются демонстрация соответствия теоретического расчета и экспериментального результата тестов и простота написания программы, для которой достаточно минимальной оптимизации ассемблерного кода.

Недостатки заключаются в ограниченности набора проверяемых команд и в том, что программа не имеет смысла с точки зрения решения прикладных задач, она применяется исключительно с целью формальной демонстрации. По мере усложнения и совершенствования процессоров, например, при введении нескольких вычислительных секций сопроцессора, которые должны быть задействованы одновременно, сложность подготовки тестовой программы будет возрастать.

V. Тест на основе библиотечной функции DGEMM

Классическими считаются тесты производительности микропроцессоров, основанные на библиотеках линейной алгебры, таких как BLAS (Basic Linear Algebra Subprograms) [4] или VSIPL (Vector Signal Image Processing Library). BLAS обеспечивает универсальным интерфейсом (API) разработчиков прикладных программ, нацеленных на решение задач линейной алгебры (получение LU-разложения матрицы, вычисление определителя матрицы, решение систем линейных уравнений и др.).

В НИИСИ РАН было проведено исследование выполнения функций BLAS на сопроцессорах и процессорах архитектуры КОМДИВ. В статье [8] приведен анализ кода некоторых BLAS-функций. Тесты на основе этих функций могут быть использованы и для испытаний производительности микропроцессоров.

BLAS-функции разделяют на три уровня в зависимости от порядка числа операций. Первый уровень, $O(n^1)$, соответствует векторным операциям, второй – операциям матрица-вектор, третий – матрично-матричный. Наибольшая производительность достигается для третьего уровня $O(n^3)$. Интерес представляет функция DGEMM – «умножение вещественных матриц с накоплением»:

$$C \leftarrow \alpha AB + \beta C ,$$

где A – матрица $m*k$, B – матрица $k*n$, C – матрица $m*n$, α, β – коэффициенты.

Для функции DGEMM были применены следующие приемы оптимизации:

1) поэтапное исполнение матричного перемножения с расщеплением обрабатываемых матриц на малые блоки, умещающиеся в Кэш–памяти;

2) Использование регистров сопроцессора в качестве «надстройки» над Кэш-памятью и размещение в них часто используемых матричных блоков;

3) Оптимизация размещения данных в регистрах для использования высокопроизводительной команды «mvmadd»;

4) Совмещение во времени вычислительных операций для одной группы строк матричного блока с операциями загрузки/сохранения для другой группы строк.

В результате удалось достичь вычислительной эффективности алгоритма, оцениваемой в 91% от пиковой производительности.

Недостатком рассматриваемого подхода является более низкий результат по сравнению с программой определения максимальной производительности. Подготовка такого теста трудоемка и требует работы высококвалифицированных специалистов, проводящих оптимизацию библиотеки линейной алгебры под конкретный микропроцессор.

Достоинством использования тестов на основе библиотечных функций является приближение тестовой задачи к характерным вычислениям, выполняемым пользователями, что дает возможность создания набора тестов, моделирующих реальные задачи. При оценке производительности задействованы различные команды микропроцессора. Существует возможность проверки правильности полученных результатов путем сравнения с результатами тех же тестов, полученных на другом вычислительном устройстве.

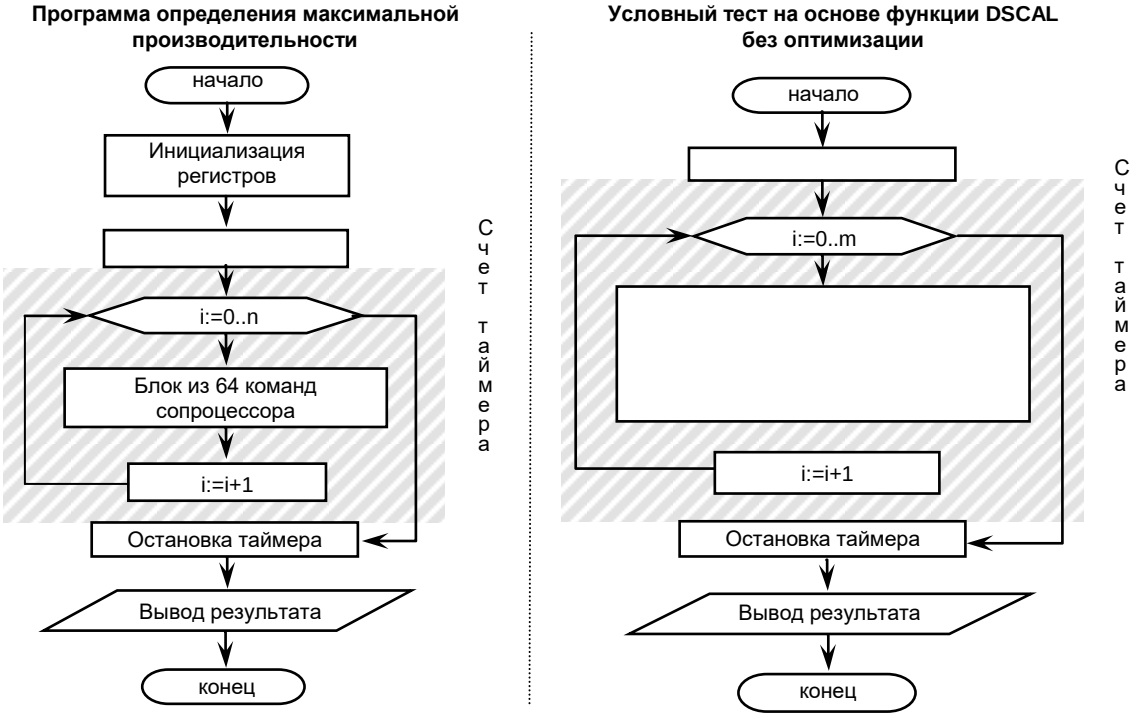

Рис. 2. Блок-схемы программ определения производительности

VI. Сравнение программ определения производительности

Блок-схемы программы определения производительности микропроцессора и теста на основе библиотеки линейной алгебры приведены на рис. 2. С целью упрощения блок-схемы приведен неоптимизированный алгоритм функции первого уровня DSCAL – «умножение вектора на скаляр». Из блок-схемы видно, что программа определения максимальной производительности не учитывает операции загрузки данных в регистры сопроцессора и не требует выгрузки результатов, за счет чего достигается более высокое соотношение команд вычисления и прочих команд. Команды подкачки данных задействованы лишь для сохранения контекста процессора при вызове ассемблерной подпрограммы и при выходе из нее. Количество выполняемых в ходе цикла инструкций превышает 100 миллионов, поэтому команды сохранения и восстановления контекста практически не оказывают влияния на результат. Инструкции организации цикла выполняются в независимом от сопроцессора конвейере целочисленной арифметики и не увеличивают время работы программы. Благодаря этому для некоторых инструкций удалось достичь результата, близкого к 100% от пиковой производительности. Оптимизированные функции библиотеки BLAS, конечно, используют возможности параллельной подкачки и выгрузки данных. Но реальный характер решаемых задач предполагает использование алгоритмов, которые невозможно оптимизировать до уровня полного использования микропроцессора. Поэтому вычислительная эффективность функций BLAS не превышает 91%.

VII. Тест LINPACK

Другим тестом, активно использующим BLAS, является LINPACK benchmark [5], основанный на решении случайной системы линейных уравнений (СЛАУ) методом LU-разложения матрицы [6]. Его современной версией является HPL (High Performance Linpack), предназначенный для оценки производительности параллельных вычислительных систем и используемый при составлении рейтинга самых быстрых компьютеров мира Top500 [7]. HPL решает СЛАУ вида Ax = b, сгенерированную случайным образом. Используется арифметика двойной точности (64 бит) на компьютерах с распределенной памятью. Этот тест является масштабируемым, он может эффективно применяться как для одного единственного процессора, так и для систем, содержащих сотни и более процессоров.

Исходный текст программы находится в открытом доступе. Функция DGEMM является наиболее критичной для HPL. По известной оценке, DGEMM и сводимые к ней функции DTRSM – «решение системы

линейных дифференциальных уравнений» и DGETRF – «LU-разложение матрицы» занимают около 90% от общего времени вычислений.

Гистограмма вычислительной эффективности суперкомпьютеров, основанная на данных рейтинга Top500 на июнь 2015 г., показана на рис. 3. По оси абсцисс обозначены интервалы вычислительной эффективности с шагом 10%, по оси ординат – количество компьютеров, попадающих в заданный интервал. На гистограмме виден существенный разброс вычислительной эффективности. Для 66% компьютеров вычислительная эффективность, измеренная HPL, попадает в диапазон 60-90% от пиковой производительности. Для задачи подтверждения производительности микропроцессора ожидаемый результат будет определяться качеством оптимизации библиотеки BLAS.

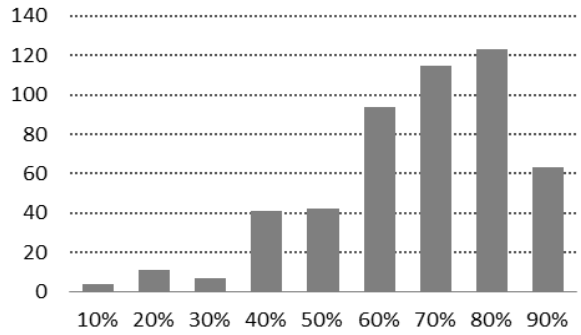

Рис. 3. Гистограмма вычислительной эффективности суперкомпьютеров

Недостатки данного подхода аналогичны тесту на основе DGEMM. Итоговая производительность, видимо, будет ниже, чем у DGEMM за счет накладных расходов [9]. Также добавляется необходимость сборки теста с оптимизированной библиотекой BLAS.

Преимущество состоит в соответствии теста общепринятой методике оценки производительности. Следовательно, появляется возможность сравнения изделий с другими микропроцессорами и ЭВМ. В дальнейшем тест HPL может использоваться для многопроцессорных систем.

VIII. Реальная и пиковая производительности микропроцессоров

Тесты на основе библиотек линейной алгебры не являются бесспорной оценкой быстродействия микропроцессоров. Их результат зависит от различных взаимосвязанных факторов, таких как реализация высокопроизводительного алгоритма, характеристики аппаратных устройств, возможности компилятора и др. Для многопроцессорных систем играет роль эффективность распараллеливания задачи.

В НИИСИ РАН исследование производительности прототипов микропроцессоров производится на ранних этапах проектирования [10] путем запуска разнообразных тестов, таких как Whetstone, Dhrystone, SPEC, EEMBC, UBCS, APEX-MAP и др.

IX. Заключение

Пиковая производительность является теоретической величиной. Если во время испытаний опытного образца микропроцессора необходимо провести демонстрацию его производительности, то убедительного результата можно добиться, выполнив следующую последовательность действий:

1) произвести теоретический расчет пиковой производительности на основе тактовой частоты f_{cpu};

2) измерить реальную тактовую частоту, на которой функционирует опытный образец микропроцессора, что позволяет подтвердить правильность теоретического расчета;

3) выполнить программный тест и подтвердить, что производительность опытного образца соответствует пиковой производительности, указанной в техническом задании.

«Программа определения максимальной производительности» обеспечивает самый высокий результат при наименьшей трудоемкости разработки. Тесты на основе DGEMM или LINPACK могут быть использованы, если проведена оптимизация библиотеки BLAS и вычислительная эффективность находится на уровне 90%. Если реальную тактовую частоту микропроцессора удается поднять выше частоты, для которой была рассчитана пиковая производительность при составлении технического задания, то результат теста окажется близким к величине, которую требуется подтвердить.

Традиционным подходом считается практическая оценка производительности и эффективности процессоров на основе теста LINPACK. Компания-разработчик отдельно заявляет теоретическую пиковую производительность своего микропроцессора. В целях контроля за экспортом компьютерного оборудования в США используется расчетная метрика «Adjusted Peak Performance» (APP). Преимущество предлагаемого метода, в сравнении с аналогами, заключается в демонстрации соответствия теоретической пиковой производительности и практического результата измерений. «Программа определения максимальной производительности» и тест на основе DGEMM позволяют сравнительно легко получить теоретический расчёт функционирования алгоритма с точностью до одного такта микропроцессора.

Рассмотренные тесты могут быть применены и на ранних этапах разработки микропроцессора, например, для логической модели на основе ПЛИС, что позволяет проводить отладку тестов до получения опытных образцов, а также оценивать быстродействие проекта на пониженной частоте.

Литература

[1] Попов А.Ю. Организация суперскалярных процессоров: учеб. пособие по курсу «Организация ЭВМ» / А.Ю.Попов. –М.: Изд-во МГТУ им. Н.Э.Баумана, 2011. -57 с.

[2] Зубковский П.С., Ивасюк Е.В., Аряшев С.И. Сопроцессор комплексных вычислений // Проблемы разработки перспективных микро- и наноэлектронных систем - 2010. Сборник трудов / под общ. ред. академика А.Л.Стемпковского. М.:ИППМ РАН, 2010. С. 356-359.
[3] Николина Н.В., Зубковский П.С., Чибисов П.А. Сопроцессоры вещественной и комплексной арифметики и их тестирование // Проблемы разработки перспективных микро- и наноэлектронных систем - 2010. Сборник трудов / под общ. ред. академика А.Л.Стемпковского. М.:ИППМ РАН, 2010. С. 360-363.
[4] URL:http://www.netlib.org/blas/ (дата обращения 03.02.16).
[5] Jack J. Dongarra Performance of Various Computers Using Standard Linear Equations Software URL:http://www.netlib.org/benchmark/performance.pdf (дата обращения 03.02.16).
[6] Андреев В. Численные методы: учебное пособие. — Издательский отдел факультета ВМиК МГУ имени М.В. Ломоносова (лицензия ИД N 05899 от 24.09.2001г.); МАКС Пресс Москва, 2013. — С. 336.
[7] URL:http://www.top500.org (дата обращения 03.02.16).
[8] Бурцев А.А. О возможности оптимизации некоторых функций библиотеки линейной алгебры с помощью векторного сопроцессора // Труды НИИСИ РАН. — 2014. — Т. 4, № 2. — С.5–15.
[9] A.Heinecke, K.Vaidyanathan, M. Smelyanskiy, A. Kobotov, R. .Dubtsov, G. Henry, A. G Shet, G. Chrysos, P. Dubey Design and Implementation of the Linpack Benchmark for Single and Multi-Node Systems Based on Intel® Xeon PhiTM Coprocessor // IEEE International Parallel & Distributed Processing Symposium, 2013. С. 126-137.
[10] Аряшев С.И., Николина Н.В., Чибисов П.А. Тесты аттестации архитектуры RTL-модели 64-разрядного суперскалярного микропроцессора // Проблемы разработки перспективных микроэлектронных систем - 2005. Сборник научных трудов / под общ. ред. А.Л.Стемпковского. М.:ИППМ РАН, 2005. С. 257-262.

Analysis of modern microprocessors peak performance

D.I. Slinkin, P.S. Zubkovsky

Scientific Research Institute of System Analysis RAS,

dima_s@ cs.niisi.ras.ru, zubkovsky@cs.niisi.ras.ru

***Keywords* — peak performance, CPU, arithmetic coprocessor, computational efficiency, BLAS, LINPACK.**

ABSTRACT

For modern microprocessors, performance evaluation based on clock frequency becomes meaningless owing to the existence of multiple cores and arithmetic coprocessors. The term ‘Peak performance’ – quantity of floating point operations per second – is used for description of performance. SRISA RAS has designed a family of microprocessors with KOMDIV architecture. This article is devoted to the problem of demonstration of the maximum performance of microprocessors during testing. Performance of a microprocessor’s core on integer operations is lower than performance of a specialized coprocessor on real or complex number operations. Therefore, the programs that use arithmetic coprocessor and methods of their optimization are of interest. Some ways of microprocessor performance testing are discussed in the article. The method to confirm compliance of a prototype to the specification requirements is suggested in conclusion.

REFERENCES

[1] Popov A.Yu. Organizaciya superskalyarnyh processorov: ‘Organizaciya EHVM’ course textbook. Izdatelstvo MGTU im. N.Eh.Baumana, 2011. -57p. (in Russian).
[2] Zubkovskij P.S., Ivasyuk E.V., Aryashev S.I. Soprocessor kompleksnyh vychislenij. Problemy razrabotki perspektivnyh mikro- i nanoehlektronnyh sistem - 2010. Collction of scientific papers. under ed. A.L.Stempkovskogo. Moscow, IPPM RAN, 2010. pp. 356-359. (in Russian).
[3] Nikolina N.V., Zubkovskij P.S., CHibisov P.A. Soprocessory veshchestvennoj i kompleksnoj arifmetiki i ih testirovanie. Problemy razrabotki perspektivnyh mikro- i nanoehlektronnyh sistem - 2010. Sbornik trudov. pod obshch. red. akademika A.L.Stempkovskogo. Moscow, IPPM RAN, 2010. pp. 360-363 (in Russian).
[4] Available at:http://www.netlib.org/blas/ (accessed 03.02.16).
[5] Jack J. Dongarra Performance of Various Computers Using Standard Linear Equations Software URL:http://www.netlib.org/benchmark/performance.pdf (accessed 03.02.15).
[6] Andreev V. CHislennye metody: uchebnoe posobie. — Izdatel'skij otdel fakul'teta VMiK MGU imeni M.V. Lomonosova. MAKS Press Moscow, 2013. — 336 p. (in Russian).
[7] Available at:http://www.top500.org (accessed 03.02.16).
[8] Burtsev A.A. On a possibility to optimize some library functions of linear algebra by means of a vector coprocessor. Trudy NIISI RAN, 2014, T4 2 pp. 5-15 (in Russian).
[9] A.Heinecke, K.Vaidyanathan, M. Smelyanskiy, A. Kobotov, R. .Dubtsov, G. Henry, A. G Shet, G. Chrysos, P. Dubey Design and Implementation of the Linpack Benchmark for Single and Multi-Node Systems Based on Intel® Xeon Phi™ Coprocessor // IEEE International Parallel & Distributed Processing Symposium, 2013. pp. 126-137.
[10] Aryashev S.I., Nikolina N.V., CHibisov P.A. Testy attestacii arhitektury RTL-modeli 64-razryadnogo superskalyarnogo mikroprocessora. Problemy razrabotki perspektivnyh mikroehlektronnyh sistem - 2005. Collction of scientific papers. under ed. A.L.Stempkovskogo. Moscow, IPPM RAN, 2005. pp.257-262