

Теоретическая оценка вычислительной способности процессоров Intel

А. А. Ракитский¹

В представленной статье описывается применение инновационного метода теоретической оценки производительности процессоров и вычислительных систем к основным процессорам Intel. Данный метод основан на определении вычислительной способности процессора – характеристики, базирующейся на основных понятиях теории информации, он применяется здесь для определения вычислительной способности большинства основных процессоров Intel (от 80486 до Core 2 Duo). На основе полученных значений характеристики приводится сравнительный анализ рассмотренных процессоров, а также предлагается сравнение вычислительной способности со значениями общепризнанных бенчмарков.

Ключевые слова: производительность, вычислительная способность.

1. Введение

Одной из реалий современного мира давно уже является повсеместное внедрение компьютеров и вычислительных устройств во все сферы жизнедеятельности человека. Мобильные телефон, персональные компьютеры, ноутбуки, суперкомпьютеры и кластеры, планшеты, электронные книги – это далеко не полный список устройств, участвующих в нашей повседневной деятельности. Мало кто может представить свою жизнь без этих устройств. Важнейшей характеристикой всех этих устройств является производительность: то, насколько быстро они смогут решить поставленную перед ними пользователем задачу. В настоящее время методы оценки и сравнения этих устройств заключаются в следующем: исследователь собирает рабочую модель исследуемого устройства, запускает на этом устройстве тестовые наборы программ, решающие те или иные задачи (т.н. бенчмарки), и замеряет время выполнения этих программ, объём требуемой памяти и т.д. На основе полученных данных устройства и сравниваются между собой. Данная методика имеет несколько очевидных минусов: во-первых, она требует обязательное наличие действующей модели исследуемого компьютера (под компьютером будем понимать любое из описанных выше вычислительных устройств), что влечет за собой определённые расходы; во-вторых, данная методика не может быть объективной в силу узконаправленности запускаемых тестовых программ. Другими словами, запуская конкретную программу на компьютере, можно оценить, лишь как будет вести себя компьютер на данной конкретной программе. Чтобы получить более объективную оценку, модель компьютера запускают на множестве различных программ, решающих различные задачи, но даже в этом случае нельзя говорить о полноте и объективности оценки, потому как компьютер в любом случае оценивается лишь на малом подмножестве множества всех допустимых программ.

Б. Я. Рябко в работе [1] предлагает инновационный теоретический метод оценки и сравнения компьютеров, основанный на вычислительной способности компьютера. Под вычис-

¹ Работа выполнена в рамках гранта по гос. контракту №8229 от 6 августа 2012 года.

лительной способностью компьютера подразумевается характеристика, описанная в [1]. Данная характеристика основывается на базовых элементах теории информации Шеннона [2], таких как энтропия, ёмкость канала без шумов и т.д. Если говорить более конкретно, то вычислительная способность характеризует мощность множества различных задач, которые компьютер способен выполнять за заданное время. Для определения вычислительной способности необходимо только описание архитектуры исследуемого устройства: описание набора инструкций процессора, объём и организация памяти, устройство ядра процессора и прочее. Следовательно, чтобы вычислить эту характеристику, нам совершенно не требуется рабочая модель компьютера, а значит, метод сравнения и оценки, основанный на ней, является теоретическим. Это дает множество преимуществ перед бенчмарками: первое – оценка и сравнение устройства может производиться на этапе разработки, когда невозможно ещё получить рабочую модель; второе – данный способ оценки существенно дешевле бенчмарков, опять же по причине того, что не требуется собирать модель.

В данной статье рассматриваются следующие процессоры Intel: 80486, Pentium, Pentium MMX, Pentium Pro, Pentium II, Pentium III, Pentium IV, Pentium M, Intel Core Solo, Intel Core 2 Solo, Intel Core 2 Duo. Эти процессоры отображают примерную эволюцию процессоров. Они имеют различные архитектуры и различное устройство ядер. Для каждого из этих процессоров определяется вычислительная способность и на основе полученных данных проводится их сравнение. Полученные графики сравнения накладываются на подобные графики, построенные с помощью данных различных бенчмарков, и анализируется их соотношение друг с другом.

2. Теоретическое обоснование метода

2.1. Основные идеи и определения метода

Прежде всего, необходимо дать основные определения, которые будут использоваться в нашем методе. Для начала опишем упрощённую модель процессора, которая содержит набор инструкций I и доступную память M . Важно заметить, что под инструкцией процессора мы понимаем не только её имя, но также и все операнды, которые у неё есть. Рассмотрим для примера инструкцию POP. В нашем случае, инструкции POP AX и POP BX будут двумя совершенно различными командами, и обе будут входить в список инструкций процессора.

Процессорной задачей P будем называть некоторую последовательность инструкций $\bar{x}(P) = x_1 x_2 x_3 \dots$, где $x_i \in I$. Отдельно рассмотрим циклы: если некоторая программа P содержит цикл, который выполнится пять раз, то последовательность $\bar{x}$ будет содержать тело этого цикла, повторяющееся пять раз. Вообще, необходимо уточнить, что не все последовательности инструкций допустимы. Точнее, могут быть последовательности инструкций, которые запрещены. Например, возможна ситуация, когда недопустимо следование одной инструкции непосредственно за другой. Иначе говоря, последовательность инструкций должна удовлетворять некоторым ограничениям (или подчиняться некоторым правилам). Представим множество всех допустимых инструкций, как S_c , и будем считать, что две различные последовательности инструкций из множества – это две различные процессорные задачи. Таким образом, любая задача может быть представлена как некоторая последовательность из множества S_c . Также верно и обратное: любая последовательность инструкций из множества S_c будет являться процессорной задачей.

Определим время выполнения инструкции x как $\tau(x)$. Также для простоты будем считать, что время выполнения любой инструкции $\tau(x)$, $x \in I$, является всегда целым числом и наибольший общий делитель всех $\tau(x)$, $x \in I$, равен 1. (Это допущение верно для большинства процессоров, если принять единицу времени за процессорный такт и есть инструкции, которые выполняются за один такт, т.е. $\tau(x) = 1$).

Тогда время выполнения последовательности инструкций $\bar{x} = x_1 x_2 x_3 \dots x_t$, $\tau(\bar{x})$ задаётся формулой

$$\tau(\bar{x}) = \sum_{i=1}^t \tau(x_i).$$

Определение 1. Пусть есть компьютер с набором инструкций I , и пусть $\tau(x)$ – время выполнения инструкции $x \in I$. Тогда вычислительная способность $C(I)$ будет представлена как

$$C(I) = \lim_{T \rightarrow \infty} \frac{\log_2 N(T)}{T}.$$

Утверждение 1. Данный предел существует, когда I – конечное множество, времена выполнения $\tau(x)$, $x \in I$ – целые числа и их наибольший общий делитель равен 1.

Доказательство данного утверждения, как и более подробное описание метода, приводится в работе [1].

2.2. Метод определения вычислительной способности

Проще всего определить вычислительную способность в том случае, если предположить, что все последовательности инструкций являются допустимыми. Давайте представим, что набор инструкций I – это алфавит, и будем считать, что все слова над этим алфавитом (последовательности инструкций) могут быть выполнены. В таком случае можно применить метод вычисления пропускной способности канала без потерь, предложенный Шенноном в [2]. Стоит заметить, что предложенный метод может быть применён для нахождения верхней границы вычислительной способности любого компьютера, так как множество всех допустимых последовательностей инструкций – это подмножество всех слов над алфавитом I .

Пусть имеется компьютер с набором инструкций I , время выполнения которых $\tau(x)$, $x \in I$, и все последовательности инструкций разрешены. В этой ситуации возможно применение описанного выше метода оценки вычислительной способности, предложенного Шенноном. Суть этого метода заключается в том, что $C(I)$ равна логарифму от наибольшего действительного решения X_0 уравнения:

$$X^{-\tau(x_1)} + X^{-\tau(x_2)} + \dots + X^{-\tau(x_S)} = 1,$$

где $I = \{x_1, x_2, \dots, x_S\}$. Иначе говоря, $C(I) = \log_2 X_0$.

3. Применение метода к процессорам Intel

В [3] уже были подробно рассмотрены особенности применения метода на практике, а также представлено описание приложения метода для процессоров Intel 80286, 80486, Pentium и Pentium MMX. Там же рассматривается особенность работы метода с учётом наличия кэш-памяти и конвейера, поэтому в данной статье эти моменты будут опущены, и мы начнём с рассмотрения особенностей применения метода для процессоров семейства Intel P6.

3.1. Особенности определения вычислительной способности процессоров семейства Intel P6

Микроархитектура процессоров этого семейства является базой для большинства более современных архитектур Intel, поэтому её следует рассмотреть особо тщательно. К этому семейству относятся процессоры Pentium Pro, Pentium II и Pentium III. Архитектура этих процессоров радикально отличается от архитектуры семейства P5. Конвейер процессоров P6 состоит из 9 основных частей. Опишем эти блоки:

- 1) Блок предсказания перехода. Этот блок помогает предсказать процессору, по какой ветви условия пойдет программа. Начиная с процессоров P6 этот блок был доведён до довольно высокого уровня точности, так что ситуация с неправильным предсказыванием перехода стала очень редка и в случае верхней оценки вычислительной способ-

ности мы смело можем ею пренебречь. Если обобщить, то данный блок указывает следующему блоку, откуда именно загружать инструкции.

- 2) Блок выборки команд. Этот блок загружает коды инструкций из кэш-памяти кода в двойной 16-байтовый буфер. Более подробно этот блок будет рассмотрен дальше.
- 3) Декодер инструкций. В нём инструкции декодируются и разбиваются на микрооперации.
- 4) Блок переименования регистров.
- 5) Блок пересортировки микроопераций.
- 6) Блок ожидания. В этом блоке накапливаются инструкции, ожидающие исполнения.
- 7) Порты соединения с исполняющими модулями.
- 8) Блок обратной пересортировки микроопераций.
- 9) Блок восстановления регистров. Здесь все временные регистры, которые были использованы при выполнении микроопераций, переписываются в постоянные регистры.

Теперь рассмотрим подробнее эти блоки. Первый блок нас не особо интересует, т.к. ошибка предсказания перехода будет очень редкой и её можно проигнорировать. Поэтому мы сразу перейдём к блоку выборки команд и опишем основные принципы его работы. Блок выборки команд загружает, как уже было сказано ранее, данные из кэша кода в два 16-байтных буфера. Уже из этих буферов блок загружает информацию об инструкциях и формирует с помощью этих данных из полученных инструкций блок инструкций (ifetch-блок).

Прежде всего, инструкции в виде обработанного ifetch-блока поступают в блок декодирования длины инструкции. Очевидно, что обычно определить длину второй инструкции, не определив длину первой в блоке, довольно сложно, но так устроен обычный декодер длины. В процессорах семейства R6 используется декодер длины инструкций, который может за один такт определять длины трёх инструкций, что необходимо для обеспечения параллелизма в вычислениях. Дальше инструкции поступают непосредственно в декодер инструкций. Он состоит из трёх декодеров, которые работают параллельно, однако принципы их работы немного различны. Декодеры назовём D0, D1 и D2. Декодер D0 может декодировать за один такт инструкции, которые генерируют не более 4 микроопераций. Это не означает, что он не может обрабатывать инструкции, которые генерируют больше микроопераций, как раз-таки все подобные инструкции будут поступать именно на него, просто это означает, что он может сгенерировать за один такт не больше четырёх микроопераций. Декодеры D1 и D2 могут обрабатывать только те инструкции, которые генерируют одну микрооперацию. Первая инструкция из ifetch-блока всегда идёт на декодер D0, последующие две идут на декодеры D1 и D2, если это возможно. Если инструкция не может быть обработана декодером D1, то она встаёт в очередь и ожидает, когда освободится декодер D0. Таким образом, можно утверждать однозначно, что минимальное количество генерируемых микроопераций в один такт будет равно 2, а максимальное – 6. На самом деле, так как на всех остальных блоках пропускная способность составляет 3 микрооперации, а мы определяем верхнюю границу оценки вычислительной способности, то можно считать, что этот блок всегда генерирует 3 микрооперации за один такт. Стоит отметить, что данная оценка будет очень близка к правде, так как в абсолютном большинстве случаев этот блок будет генерировать больше, чем 3 микрооперации, и они будут помещаться в очередь ожидания. То есть даже если в какой-то момент и будут сгенерированы всего 2 микрооперации, то это будет скомпенсировано за счёт накопившихся операций в очереди.

Блок переименования регистров управляется таблицей псевдонимов регистров (RAT – register alias table). Операции после декодера через очередь поступают в этот блок, после чего отправляются в блок пересортировки. Блок RAT может обрабатывать одновременно 3 микрооперации за один такт. Это означает, что максимальная пропускная способность процессора не может превышать трёх микроопераций в такт. Никаких дополнительных ограничений

на этот блок нет. Он может переименовывать 3 разных регистра или же трижды переименовать один и тот же регистр (если подобное потребуется) за один процессорный такт.

После переименования регистров идёт стадия переупорядочивания (ROB – Reorder buffer). Все значения регистров помещаются в ROB-запись, если они доступны. В каждой ROB-записи может быть 2 входных регистра и 2 выходных. Вообще, возможны 3 случая для значений входных регистров: значение регистра не изменялось недавно (в таком случае оно загружается из постоянного файла регистров); значение регистра недавно было изменено и процессор ещё не выгрузил его в файл (тогда значение загружается не из файла, а из ROB-записи, в которой хранится результат изменения); значение ещё не готово (это означает, что требуемое значение регистра ещё не вычислено и находится в очереди на выполнение, но сразу же после вычисления значение будет занесено в запись самим исполняющим модулем). Пропускная способность данного блока также составляет 3 микрооперации в секунду, однако возможен случай, когда может произойти замедление, его мы и рассмотрим. Как ни странно, этот случай относится к первому случаю значений регистра, хоть он и является самым простым и понятным. Проблема заключается в том, что постоянный файл регистров имеет всего 2 порта для чтения данных. ROB-блок может получить за один такт 3 микрооперации, и самая худшая ситуация будет, если в каждой операции будет 2 входных регистра, все они будут различны, и все будут относиться к первому варианту значений. Очевидно, что за один такт мы можем считать ровно 2 значения различных регистров, поэтому для записи значений нам понадобится 3 такта. Тем не менее, необходимо заметить, что такая ситуация очень маловероятна и на практике почти неосуществима. Ну а так как мы оцениваем верхнюю границу вычислительной способности, то пренебрежём подобными ситуациями и будем считать пропускную способность по-прежнему равной трём микрооперациям за один такт.

Блок переупорядочивания может хранить в себе 40 микроопераций и 40 временных регистров, а блок ожидания всего 20 микроопераций. Каждая микрооперация ожидает в ROB-блоке до тех пор, пока все её операнды не будут готовы и пока не освободится блок выполнения, подходящий для неё. Это делает возможным выполнение инструкций не в том порядке, в котором они поступили, а сразу же, как только все её операнды станут доступны.

Для соединения с исполнительным модулем есть 5 различных портов (порты нумеруются с 0). Порты 0 и 1 служат для обработки арифметических и подобных операций. Простые операции перемещения, арифметические или логические, могут поступать на любой из этих двух портов в зависимости от того, какой освободится раньше. Порт 0 также включает в себя некоторые операции умножения, деления и сдвигов. Порт 1 поддерживает операции переходов и некоторые MMX- и XMM-операции. Порт 2 служит для всех операций чтения данных, а также для некоторых операций со строками и некоторых XMM-операций. Порт 3 служит для операций расчёта адресов для записи данных, а порт 4 поддерживает операции, которые осуществляют саму запись. Подробное описание того, сколько микроопераций и на какой именно порт генерирует каждая инструкция, можно найти в [7]. В лучшем случае исполнительный модуль может за один такт обработать 5 микроопераций, если они выполняются на разных портах. На самом деле, благодаря возможности выполнять операции не по порядку, пропускная способность в данном модуле будет стремиться к максимальной, но так как пропускная способность предыдущих блоков составляет 3 микрооперации, то мы, опустив случаи замедления выполнения, возьмем пропускную способность исполнительного модуля также в 3 микрооперации.

Блок восстановления – это место, где временные регистры, которые использовались микрооперациями, копируются в постоянные регистры. Когда микрооперация была выполнена, она отмечается в блоке переупорядочивания как готовая к восстановлению. Станция восстановления может обрабатывать одновременно 3 микрооперации за один такт. Все операции находятся в блоке переупорядочивания до тех пор, пока не выполнятся операции, предшествующие им в изначальном порядке. Это тоже может привести в некоторых случаях к снижению пропускной способности, но мы также опустим подобные ситуации, потому как они

маловероятны и снижение скорости будет весьма незначительным и только на короткий промежуток времени.

Таким образом, мы можем считать, что процессоры этого семейства имеют пропускную способность в 3 микрооперации за один такт. А это означает то, что зная количество микроопераций, на которые разбивается каждая инструкция, мы сможем легко оценить и время её выполнения. Очевидно, что инструкция не может считаться выполненной, пока не будут выполнены все её микрооперации. Тогда за время выполнения инструкции мы возьмём количество микроопераций, на которые она будет разбита. Объяснить это достаточно просто. Каждая микрооперация проходит через все блоки, кроме блока выполнения, за один такт. Следовательно, если микрооперации будут выполняться последовательно, то время выполнения инструкции будет совпадать с количеством микроопераций. Предыдущее утверждение является не до конца верным, ведь инструкции задерживаются и могут выполняться дольше в исполнительном модуле. Но благодаря параллельному выполнению микроопераций, их перераспределению и прочим оптимизациям, мы можем считать временем выполнения инструкции именно количество микроопераций. Так как пропускная способность процессора составляет 3 микрооперации, то мы можем считать, что у нас идёт на выполнение 3 независимых последовательности микроопераций. Тогда можно считать, что инструкции выполняются на 3 параллельных процессорах, и, согласно теории, описанной в [1] и [3], вычислительная способность будет рассчитываться по формуле

$$C(I) = 3 \times C(I'),$$

где I' – это список инструкций процессора, последовательно выполняющего микрооперации, из которых состоят инструкции, затрачивая на выполнение каждой микрооперации ровно один такт. Конечно, процессоры Pentium Pro, Pentium II и Pentium III имеют различия в своих списках инструкций. Например, процессор Pentium II, в отличие от Pentium Pro, был дополнен инструкциями MMX, которые также перешли и в Pentium III. Помимо этого, Pentium III был ещё дополнен инструкциями XMM. Для построения уравнений мы вводим наборы инструкций каждого процессора в программу, которая уже преобразовывает введенные данные в уравнение, учитывая при этом объёмы всех видов памяти. Построив таким образом уравнение и решив его, мы получаем данные, представленные в табл. 1.

Таблица 1. Вычислительная способность процессоров Intel P6

Наименование процессора	Кэш-память 1-го уровня, Кб	Кэш-память 2-го уровня, Кб	Оперативная память, Мб	$C(I)$, бит/такт
Pentium Pro	32	512	64	36.62
Pentium II	32	512	128	37.69
Pentium III	32	512	256	42.02

Полученные результаты для семейств процессоров P6 мы теперь можем сравнить с результатами предыдущих семейств процессоров, которые мы вычислили и описали в [3].

Таблица 2. Сравнительные характеристики процессоров Intel P5 и P6

Наименование процессора	$C(I)$, бит/такт	Тактовая частота, МГц	Вычислительная способность, Мбит/с	ICOMP	SPECint95	SPECfp95
Pentium	25.56	150	3834	114	4.13	3.58
Pentium MMX	28.35	200	5670	182	6.37	4.87
Pentium Pro	36.62	200	7323.68	220	8.59	6.34
Pentium II	37.69	400	15076.55	440	15.6	12.9
Pentium III	42.02	500	21011.17	644	21.6	16.2

В табл. 2 видны сравнительные характеристики процессоров P6 и процессоров предыдущего поколения P5. Особо интересно сравнение этих характеристик друг с другом и с бенчмарками потому, что эти семейства принципиально отличаются друг от друга и имеют кардинально различные подходы к составлению уравнения и описанию набора инструкций. Подробнее о составлении уравнения для семейства процессоров Intel P5 можно прочитать в [3]. Но как мы видим в табл. 2, полученные данные очень неплохо коррелируют с данными общепризнанных бенчмарков, что, несомненно, подтверждает состоятельность теории. Для более наглядного отображения данных и сравнения их между собой, мы построим график, отображённый на рис. 1. На этом графике мы за единицу приняли значения первого описанного процессора Intel Pentium, а все остальные процессоры сравнивали с ним.

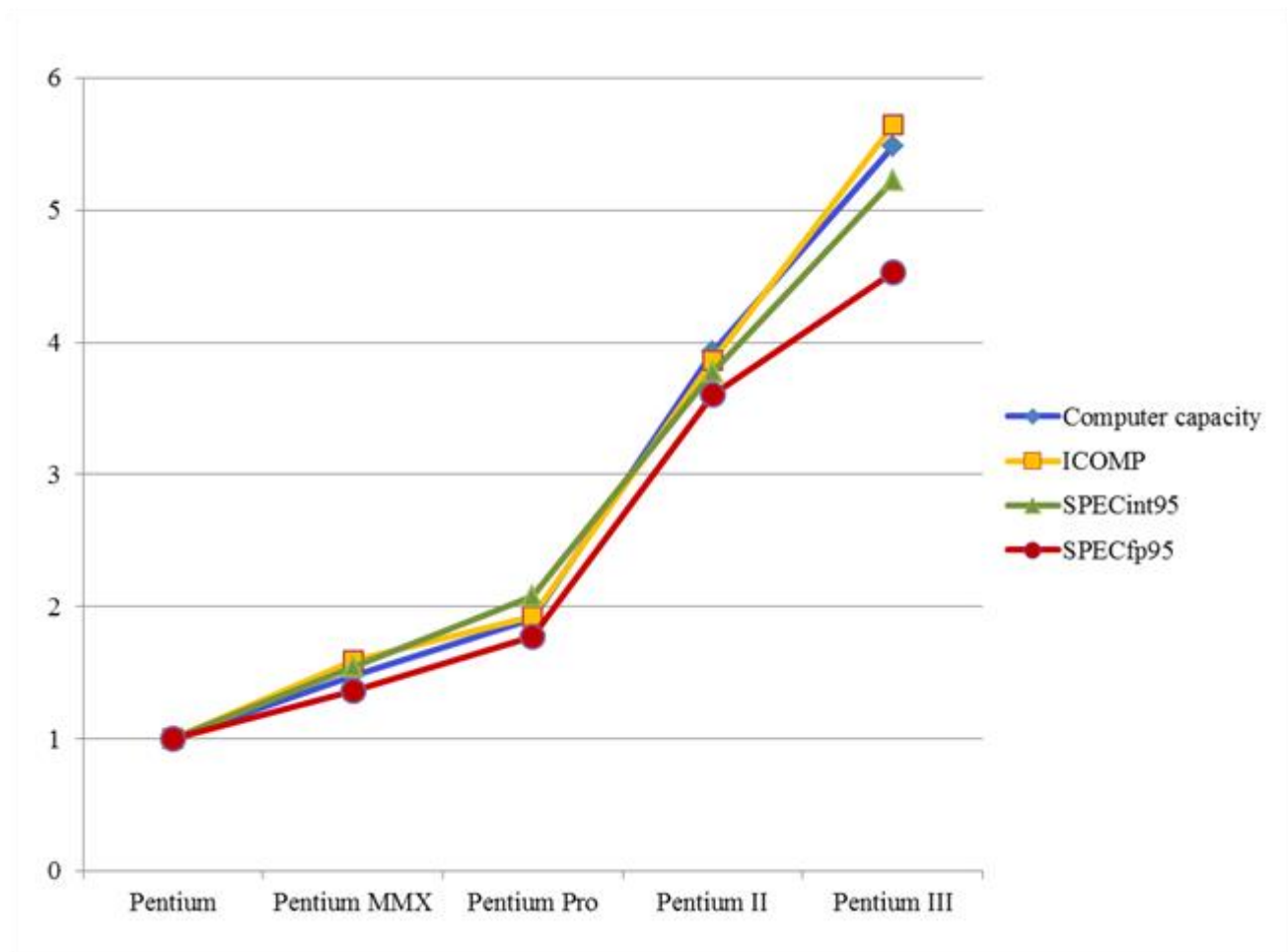


Рис. 1. Сравнительная диаграмма вычислительной способности и общепризнанных бенчмарков для процессоров серий P5 и P6

В качестве основных бенчмарков были выбраны ICOMP – средство для оценки производительности и сравнения процессоров, разработанное самой компанией Intel, и SPEC (Standard Performance Evaluation Corporation) – некоммерческая организация, состоящая из представителей всех производителей и разработчиков как компьютерной аппаратуры, так и программного обеспечения для неё, которая занимается разработкой максимально объективного средства для оценки и сравнения производительности компьютеров и вычислительных систем. Так как процессоры, которые мы сравнивали между собой, охватывают достаточно большой интервал времени, то нужны были бенчмарки, которые включают в себя весь этот интервал, что также было одной из причин выбора ICOMP и SPEC.

Таблица 3. Соотношение результатов оценки производительности процессоров.

Процессор 1 / Процессор 2	Вычислительная способность	ICOMP	SPECint95	SPECfp95
Pentium MMX / Pentium	1.4789	1.5965	1.5424	1.3603
Pentium PRO / Pentium MMX	1.2917	1.2088	1.3485	1.3018
Pentium II / Pentium PRO	2.0586	2.0000	1.8161	2.0347
Pentium III / Pentium II	1.3936	1.4636	1.3846	1.2558

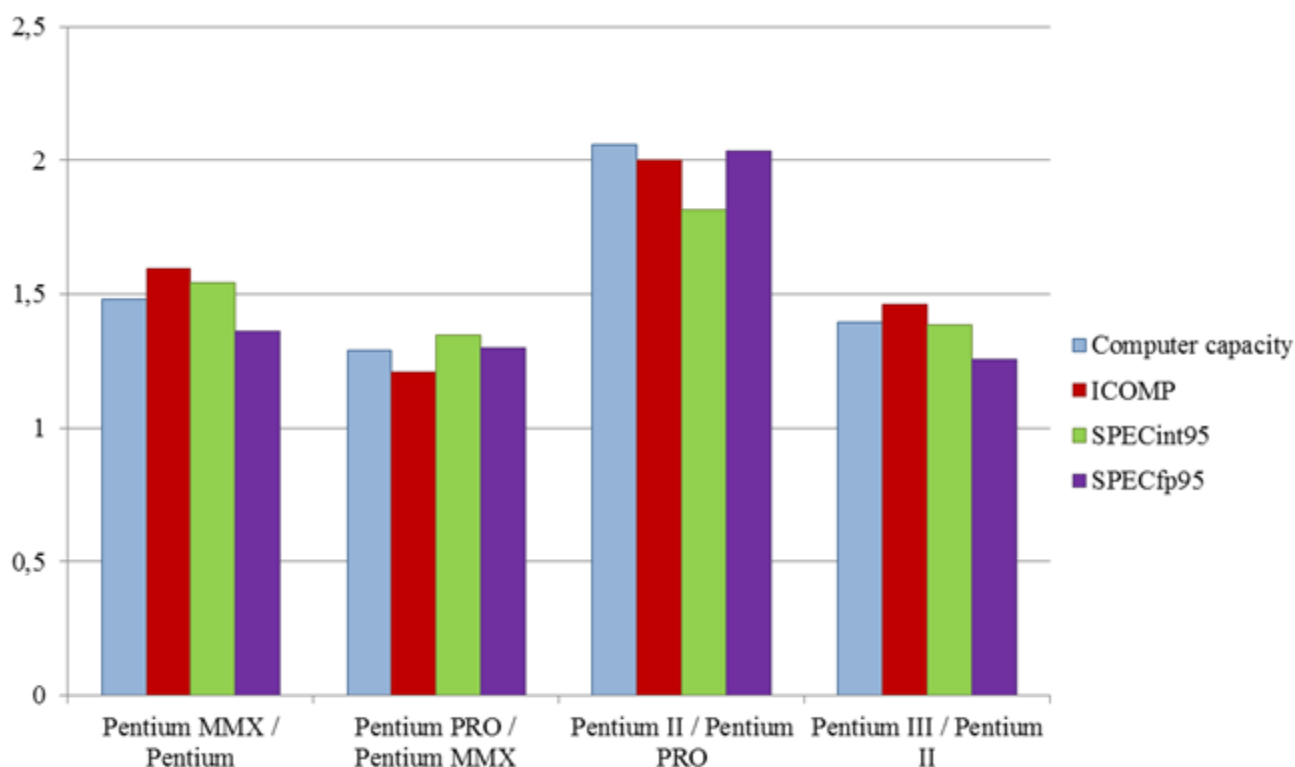


Рис. 2. Диаграмма соотношения оценок производительности процессоров

Помимо описанной выше диаграммы, можно также сравнить характеристики по принципу, как изменилась производительность процессора относительно предшествующей модели. Эта характеристика также является весьма показательной. Такие сравнения приходится делать по той простой причине, что единицы измерения всех этих характеристик являются совершенно различными. Результаты сравнения можно увидеть в табл. 3 и на диаграмме, изображённой на рис. 2.

3.2. Оценка вычислительной способности процессоров микроархитектуры NetBurst (Pentium 4, Pentium D)

Микроархитектура NetBurst сильно отличается от всех остальных процессоров Intel. На практике она показала себя не такой эффективной, как ожидалось, и поэтому больше не используется в новых разработках. Основной задачей этой микроархитектуры было достичь максимально возможно высокой тактовой частоты процессора. Это могло быть сделано

только за счёт увеличения длины конвейера. Если у процессоров серии P6 конвейер состоял из 10 стадий, то Pentium IV, который ещё относят к 7-му поколению, т.е. серии P7, состоит из 20 стадий. Некоторые стадии просто перемещают информацию из одной части чипа в другую.

Кэш-память второго уровня используется в процессоре как для данных, так и для кода. Размер кэш-памяти второго уровня варьируется от 256 Кб до 2 Мб. Взаимодействие кэша с процессором происходит на максимальной скорости через 256-битную шину данных. Кэш-память первого уровня имеет маленький размер (8 либо 16 Кб). Относительно маленький размер этой памяти компенсируется высокой скоростью доступа до кэш-памяти второго уровня. Кэш-код первого уровня заменён на кэш передачи, об этом мы подробнее напишем ниже.

Основное изменение заключается в том, что в данных процессорах кэш данных был заменён на кэш передачи, в котором хранятся не инструкции, а декодированные микрооперации. Основная часть процессора, обработка вне исходного порядка, очень похожа на эту же часть из архитектуры процессоров P6, однако буфер был значительно увеличен и может содержать уже до 126 микроопераций в очереди ожидания. Тем не менее, несмотря на все эти изменения, общая пропускная способность в 3 микрооперации осталась прежней.

Инструкции помещаются в кэш передачи после разбиения на микрооперации. Одной из важнейших причин для этого является то, что на ранних процессорах были так называемые «заторы» на стадии декодирования. Код операции может быть любой длины от 1 до 15 байт. Определить длину кода довольно сложно, и при этом нам необходимо знать длину первой инструкции, чтобы определить, где начинается вторая. А следовательно, очень сложно определять длины кодов параллельно. В 6-м семействе процессоры, как мы уже писали выше, могли декодировать 3 инструкции за один такт. Но это может быть весьма проблематично при большей тактовой частоте процессора. Если все микрооперации будут иметь одинаковый размер, то процессор сможет обрабатывать их параллельно и этот «затор» будет ликвидирован. Это основной принцип процессоров RISC. Информация в кэше передачи представлена в виде строки микроопераций, которые выполняются в последовательности, даже если они не последовательны в оригинальном коде. Дополнительным преимуществом такого подхода является то, что количество тактов, требуемых на переход в кэш, минимизируется.

Есть ещё один момент, который оказывает влияние на расчёт уравнивания. Инструкции, которые не находятся в кэше передачи, отправляются напрямую из декодера инструкций на выполняющий конвейер. Рассмотрим декодер инструкций. В большинстве случаев декодер генерирует от 1 до 4 микроопераций для каждой инструкции. Для сложных инструкций, которые требуют больше 4-х микроопераций, они загружаются из microcode ROM. Именно эти микрооперации нужно дополнительно учитывать при построении уравнения, когда мы будем преобразовывать список инструкций.

Микрооперации из кэша передачи или из декодера поступают в очередь, в которой ожидают своего выполнения. После переименования и переупорядочивания регистров, каждая микрооперация через определённый порт поступает в соответствующий ей исполнительный модуль. Каждый исполнительный модуль имеет один или несколько подмодулей, которые специализируются на определённых операциях (например, умножение или сложение). В [6] представлена более подробная таблица со структурой модулей и подмодулей.

Микрооперация может быть выполнена, только если выполнены следующие условия:

- 1) все операнды микрооперации полностью готовы;
- 2) соответствующий исполнительный порт готов;
- 3) соответствующий исполнительный модуль готов;
- 4) соответствующий исполнительный подмодуль готов.

Два из исполняющих модулей работают с удвоенной скоростью. Это важный момент, потому как он оказывает существенное влияние на построение уравнения. Модули номер 0 и 1, используемые для целочисленных операций, могут удваивать свою скорость, т.е. принимать на обработку новую микрооперацию каждые полтакта. Например, операция сложения

может выполняться и на нулевом и на первом модулях. Это означает, что за один такт процессор может выполнить целых 4 операции сложения. Модули 0 и 1 разбиты на три стадии: первая половина результата рассчитывается в первые полтакта, вторая половина результата рассчитывается во вторые полтакта, и ещё полтакта требуется на вычисление флагов. В итоге, всего на выполнение операции, конечно, потребуется 1.5 такта, однако хитрость заключается в том, что следующая микрооперация может поступать на выполнение сразу же, как только выполнится первая стадия, т.е. через полтакта после поступления первой микрооперации. Таким образом, нам необходимо учитывать при построении уравнения ещё и этот факт. Так как мы имеем информацию о том, сколько микроопераций и на какой порт, и даже более того, на какой подмодуль, генерирует каждая инструкция, то это не является проблемой.

Таким образом, мы построили уравнение и решили его, получив $C(I) \approx 39.66$ бит/такт. Теперь необходимо сравнить полученные данные с данными предыдущих процессоров. Итоговое сравнение всех рассмотренных процессоров мы проведём в конце данной работы, а здесь более интересным и информативным будет случай сравнения процессоров Pentium III и Pentium IV. В табл. 4 представлена информация об исследуемых процессорах и результаты выполнения бенчмарков на них. В последней строке описано отношение соответствующих характеристик друг к другу, при учёте оценки роста производительности процессора Pentium IV к процессору Pentium III.

Таблица 4. Данные оценки производительности процессоров Pentium IV и Pentium III.

Наименование	Частота, МГц	Вычислительная способность, Мбит/с	PassMark	3DMark	PCMark	SiSoft Sandra	CINT 2000	CFP 2000
Pentium III	1266	53200.28	270	1375	1922	4150	564	422
Pentium IV	2600	103115.51	405	2652	3070	6955	1030	1052
PIV / PIII		1.9383	1.5	1.9287	1.5973	1.6759	1.8262	2.4929

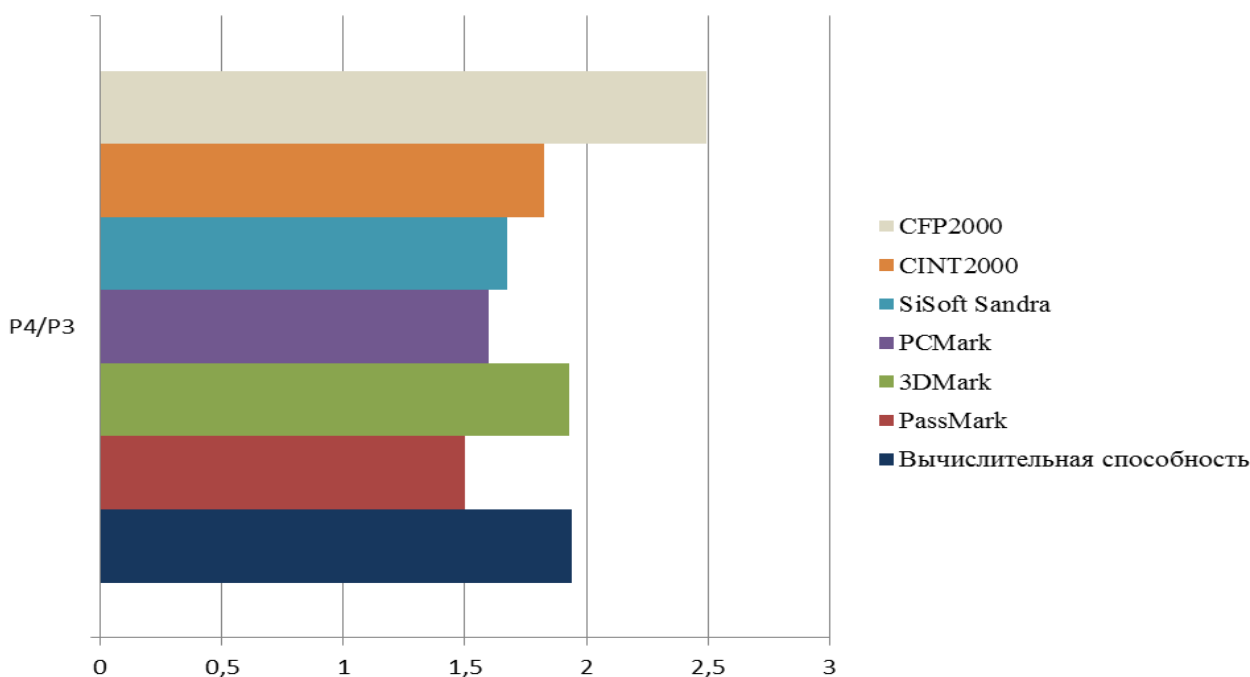


Рис. 3. Диаграмма отношения производительности процессора P4 к P3

На основе полученных в последней строке данных мы можем построить диаграмму, чтобы наглядно отобразить разброс между результатами различных бенчмарков. Диаграмма приведена на рис. 3. Здесь хорошо видно, что наша характеристика «Вычислительная способность» очень близка к результатам бенчмарка 3DMark и находится внутри интервала допустимой погрешности оценки производительности (под допустимой погрешностью мы будем подразумевать среднее значение отклонения математического ожидания значений бенчмарков от значения какого-либо из них). Более формально, допустимую погрешность опишем формулой:

$$\Delta = \frac{\sum_{i=1..6} |V_{avg} - V_i|}{6},$$

где Δ – допустимая погрешность, V_{avg} – математическое ожидание отношений значений оценки P4 к P3, V_i – значение отношения конкретной оценки производительности, если учитывать, что 1 – PassMark, 2 – 3DMark, 3 – PCMark, 4 – SiSoft Sandra, 5 – CINT2000, 6 – CFP2000. Таким образом, с учётом всего вышеописанного, получаем, что допустимый интервал будет задан как $[V_{avg} - \Delta, V_{avg} + \Delta]$. Подставим наши значения в формулы: $V_{avg} = 1.8368$, $\Delta = 0.2493$. Тогда допустимый интервал будет следующим: $[1.5875, 2.0861]$.

3.3. Оценка вычислительной способности для процессоров с микроархитектурой PM (Pentium M, Intel Core Solo)

Основа архитектуры процессоров серии PM такая же, как и у процессоров P6. Основные стадии процессора: предсказание переходов, выборка команд, декодирование инструкций, переименование регистров, переупорядочивание команд, очередь ожидания исполнения, исполнительные модули, обратное переупорядочивание и восстановление регистров.

Естественно, некоторые изменения и доработки в архитектуру были внесены. Так, был доработан блок предсказания переходов, некоторые изменения были внесены в исполнительные модули, но нас интересуют изменения, которые непосредственно влияют на построение уравнения и расчёт характеристик. В процессорах семейства PM появились так называемые «спаянные» микрооперации. Суть «спаянных» микроопераций заключается в том, чтобы повысить пропускную способность блоков, предшествующих исполнительным модулям и следующих за ними. Блок переименования регистров и блок восстановления имеют пропускную способность всего 3 микрооперации за один такт. Для того чтобы преодолеть этот предел и немного повысить пропускную способность, разработчики решили объединять некоторые микрооперации, которые в предыдущих процессорах были разделены. Их и называли «спаянными» микрооперациями. Эти операции делят одну микрооперацию в большей части процессора и одну ROB-запись. Эта запись описывает именно две микрооперации, которые выполняются в различных исполнительных модулях. Таким образом, эта запись для выполнения будет разделена на 2 разных порта, но при восстановлении будет рассматриваться как одно целое.

Техника «спайки» операций может быть применена только для двух типов комбинаций: операции записи в память и операции чтения-изменения.

Операции записи в память состоят из расчёта адреса памяти и непосредственно перемещения данных. Как мы помним из описания процессоров P6, за расчёт адреса отвечает порт 3, а для перемещения данных используется порт 4, т.е. эти две операции могут выполняться одновременно. Второй тип операций – это чтение-изменение. Здесь лучше будет пояснить на примере. Рассмотрим операцию ADD EAX [mem32] состоящую из 2-х операций. Первая – это чтение данных из ячейки памяти, вторая – суммирование считанных данных с регистром EAX. В предыдущих процессорах данная инструкция разбивалась на две микрооперации, но в процессорах PM они будут «спаяны» друг с другом. Спайка будет работать почти для всех инструкций модификации данных, кроме инструкций, которые работают с XMM-регистрами.

Добавление «спаянных» микроопераций даёт следующие преимущества:

- Декодирование становится более эффективным, т.к. если раньше инструкция образовывала 2 микрооперации, а теперь одну «спаянную» микрооперацию, то она может обрабатываться теперь любым из трёх декодеров.
- За счёт «спаянных» микроопераций увеличивается пропускная способность блока переименования регистров и блока восстановления.
- Пропускная способность блока переупорядочивания также возрастает за счёт использования одной ROB-записи для «спаянных» микроопераций.

Также в процессорах PM появился буфер цикла, размером 4×16 байт, в котором хранятся частично декодированные инструкции. Это даёт преимущество при выделении инструкций в циклах малого размера, так как декодер сможет брать уже выделенные инструкции из этого буфера. В случае использования нашего метода оценки этот буфер не оказывает существенного влияния на производительность и не меняет методику вычисления, тем не менее, о нём необходимо упомянуть, как о существенном дополнении и серьёзной модификации процессоров.

Инструкции для работы со стеком, такие как PUSH, POP, CALL и RET, вносят изменения в стековый указатель ESP. Процессоры предыдущих семейств использовали для обработки этих инструкций обычный ALU в исполнительном блоке. Рассмотрим пример: инструкция PUSH EAX генерирует 3 микрооперации. Две из них – для сохранения значения регистра EAX, ещё одна – для изменения значения ESP. В процессорах PM та же самая инструкция будет генерировать только одну микрооперацию, т.к. операции сохранения значения будут «спаяны» в одну, а микрооперацию по изменению стекового регистра выполнит так называемый стековый модуль. Стековый модуль – это специальный механизм, который обрабатывает микрооперации, изменяющие только стековый регистр. Стековый модуль располагается сразу же за декодером инструкций. Этот механизм позволяет обрабатывать три сложения за один такт. Однако существует определённая сложность, которая заключается в том, что значение ESP может понадобиться как в стековом модуле, так и в исполнительном блоке. Поэтому требуется некоторый механизм, который будет синхронизировать значение в стековом модуле и значение в исполнительном блоке. Истинное значение стекового указателя ESP_p получается из 32-битного значения ESP_0 , хранящегося в исполнительном блоке либо в постоянном регистровом файле, и знакового 8-битного значения ESP_d , которое хранится в стековом модуле, как $ESP_p = ESP_0 + ESP_d$.

Стековый модуль добавляет значение ESP_d в адресное слово каждой микрооперации, работающей со стеком, как сдвиг таким образом, что оно прибавляется к ESP_0 при расчёте адреса в порте 2 или 3. Очевидно, что значение ESP_d не может быть прибавлено к каждой микрооперации. Только к микрооперациям, генерируемым инструкциями PUSH, POP, CALL и RET. Если же встречается другая инструкция и при этом значение $ESP_d \neq 0$, то стековый модуль генерирует дополнительную микрооперацию, которая прибавляет это значение к ESP_0 , а в стековом модуле значение зануляет. Более подробное описание устройства процессора можно найти в работе [6].

Остальные блоки также претерпели некоторые изменения, но они несущественны для нашего метода оценки вычислительной способности и на него не влияют. Подведя итог описанию процессоров PM, мы можем сказать, что определение вычислительной способности для этих процессоров будет схожим с процессорами P6, но при построении уравнения необходимо будет учесть то, какие микрооперации будут «спаяны» и какие будут выполняться в стековом модуле.

Составив уравнения и решив их, получим для Pentium M $C(I) \approx 51.198$ бит/такт. Отличия в архитектуре процессоров Pentium M и Intel Core Solo минимальны и на уравнения не влияют, списки инструкций с их характеристиками также практически не отличаются, поэтому (основные изменения в процессорах Core связаны с энергосбережением) результат для Intel Core Solo будет $C(I) \approx 51.198$ бит/такт.

Для анализа процессоров серии PM сравним их с процессорами Pentium III и Pentium IV. Сравнительные данные приведены в табл. 5.

Таблица 5. Сравнительные данные процессоров Pentium III, Pentium M, Intel Core Solo и Pentium IV

Наименование процессора	Тактовая частота, МГц	PassMark	$C(I)$, бит/такт	Вычислительная способность, Мбит/с
Pentium III	1266	270	42.0223	53200.2843
Pentium M	2000	495	51.198	102396.0631
Intel Core Solo (T1350)	1860	443	51.198	95228.3388
Pentium IV	2600	405	39.6598	103115.5069

Для более наглядного отображения данных мы приведём сравнительную диаграмму, на которой выведем отношение следующего процессора к предыдущему по порядку их следования в табл. 5.

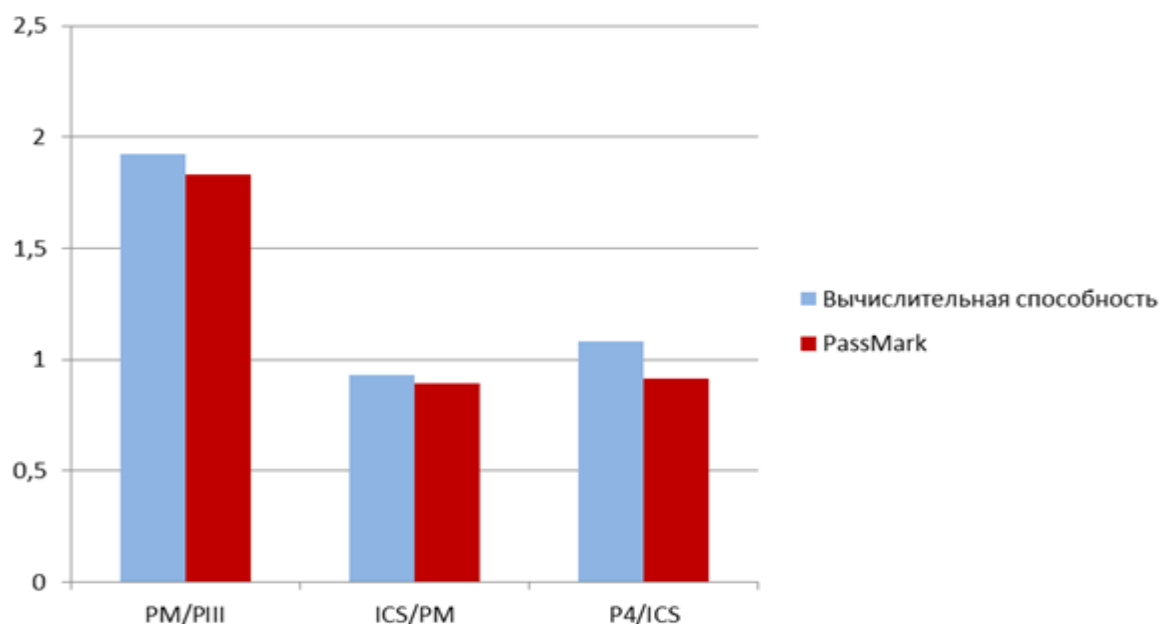


Рис. 4. Сравнительная диаграмма процессоров из табл. 5

Диаграмма, изображённая на рис. 4, в очередной раз подтверждает состоятельность нашей методики оценки производительности процессоров. На этот раз мы сравнивали 3 разных семейства процессоров с различной микроархитектурой.

3.4. Оценка вычислительной способности для процессоров Core 2 (Merom)

Микроархитектура Intel Core 2 – это более доработанный вариант архитектуры PM. Конвейер был улучшен до того, чтобы поддерживать одновременную обработку четырёх микроопераций. Процессоры микроархитектуры Core 2 на данный момент формируют основу процессоров Intel x86, включая мобильные, персональные и серверные версии. Процессоры этой серии имеют два и более вычислительных ядра с раздельным кэшем первого уровня и общим кэшем второго уровня. Рассмотрим подробнее изменения в этих процессорах.

Буфер переупорядочивания теперь имеет 96 записей. Блок ожидания – 32. Выборка инструкций была улучшена добавлением очереди между блоком предсказания переходов и блоком выборки. К сожалению, пропускная способность всё равно осталась 16 байт за один такт. Декодирование инструкций разделено между предекодером и декодером очередью. Основная задача предекодера – определить, где начинается каждая инструкция. Предекодер также определяет префиксы инструкций и прочие составляющие её компоненты. Максимальная пропускная способность предекодера 16 байт, либо 6 инструкций за один такт, в зависимости от того, что из этого будет меньше.

Непосредственно декодирование инструкций, как и вышеописанные блоки, претерпели некоторые изменения. Теперь вместо трёх декодеров в блоке декодирования находится 4 декодера и максимальная пропускная способность вырастает до 7 микроопераций за один такт (4 микрооперации может сгенерировать первый декодер и по одной ещё – каждый из трёх оставшихся). Минимальное количество микроопераций осталось по-прежнему равным двум, но такая ситуация практически невозможна, и т.к. максимальное количество генерируемых операций равно семи, то мы будем считать для верхней оценки вычислительной способности, что пропускная способность этого блока составляет 4 микрооперации.

Как и в предыдущих процессорах серии РМ, здесь тоже реализован механизм «спайки» микроопераций. Суть этого механизма полностью совпадает с реализацией на РМ. Еще две микрооперации рассматриваются как одна на всех стадиях, кроме стадии выполнения. Помимо этого, был придуман ещё один механизм, который позволяет объединять уже не микрооперации, а две инструкции в одну микрооперацию. Назовём его «спайкой» макроопераций. Этот вариант «спайки» возможен в определённых случаях:

- Первая инструкция CMP или TEST, а последующая инструкция – это условный переход (исключая инструкции JECXZ и LOOP).
- Инструкции CMP и TEST могут иметь в качестве операндов два регистра, либо регистр и константу, либо регистр и ячейку памяти, но ни в коем случае не ячейку памяти и константу.
- Core 2 может делать «спайку» макроопераций только в 16-битном или 32-битном режимах.
- Между этими двумя инструкциями не должно находиться других инструкций.
- Если несколько пар подобных инструкций одновременно достигнут декодера, то только первая пара будет «спаяна».

Любой из четырёх декодеров может выполнять «спайку» макроопераций, но не одновременно. Таким образом, мы видим, что декодер инструкций может в лучшем случае одновременно обработать 5 инструкций. «Спайка» микроопераций и макроопераций вполне может производиться одновременно.

Стековый модуль работает так же, как и в процессорах РМ, с несущественными дополнениями и доработками, которые на нашу методику определения вычислительной способности не влияют, а потому мы не будем их описывать.

Исполнительный блок был изменён и переработан. В него был добавлен ещё один исполнительный модуль для арифметических операций и теперь микрооперации распределяются по 6 портам (0,1 и 5 – это арифметические и логические операции). Таким образом, максимальная пропускная способность этого блока составляет 6 микроопераций за один такт, но с учётом пропускной способности предыдущих и следующих блоков, мы будем рассматривать общую пропускную способность в 4 микрооперации за один такт.

Формула определения вычислительной способности будет очень схожей с формулой, представленной в разделе про семейство процессоров Р6; тем не менее теперь у нас выполняется не 3, а 4 микрооперации одновременно, поэтому итоговая формула выглядит следующим образом:

$$C(I) = 4 \times C(I'),$$

где I' – это список инструкций процессора, последовательно выполняющего микрооперации, из которых состоят инструкции, затрачивая на выполнение каждой микрооперации равно

один такт. Помимо этого, данная формула относится только к одному вычислительному ядру процессора (она является абсолютно верной для процессоров Core 2 Solo), а чтобы определить вычислительную способность процессора с несколькими ядрами, мы воспользуемся формулой и методикой, описанной подробно в [1]. Данные, полученные в результате составления и решения уравнений, подробно представлены в табл. 6.

Таблица 6. Характеристики и оценка производительности процессоров Intel Core 2 Solo и Intel Core 2 Duo

Наименование процессора	Тактовая частота, МГц	Размер кэша L1, Кб	Размер кэша L2, Мб	$C(I)$, бит/такт	Вычислительная способность, Мбит/с	PassMark
Core 2 Solo U3500	1400	64	2	68.8469	96385.62	479
Core 2 Duo T7300	2000	64	4	137.6937	275387.48	1232

Сравним прирост производительности процессоров Core 2 Solo (C2S) и Core 2 Duo (C2D) относительно процессоров Pentium IV (PIV) и Pentium M (PM), а также между собой. Результаты вычислительной способности мы будем сравнивать с результатами самого распространённого и популярного бенчмарка PassMark.

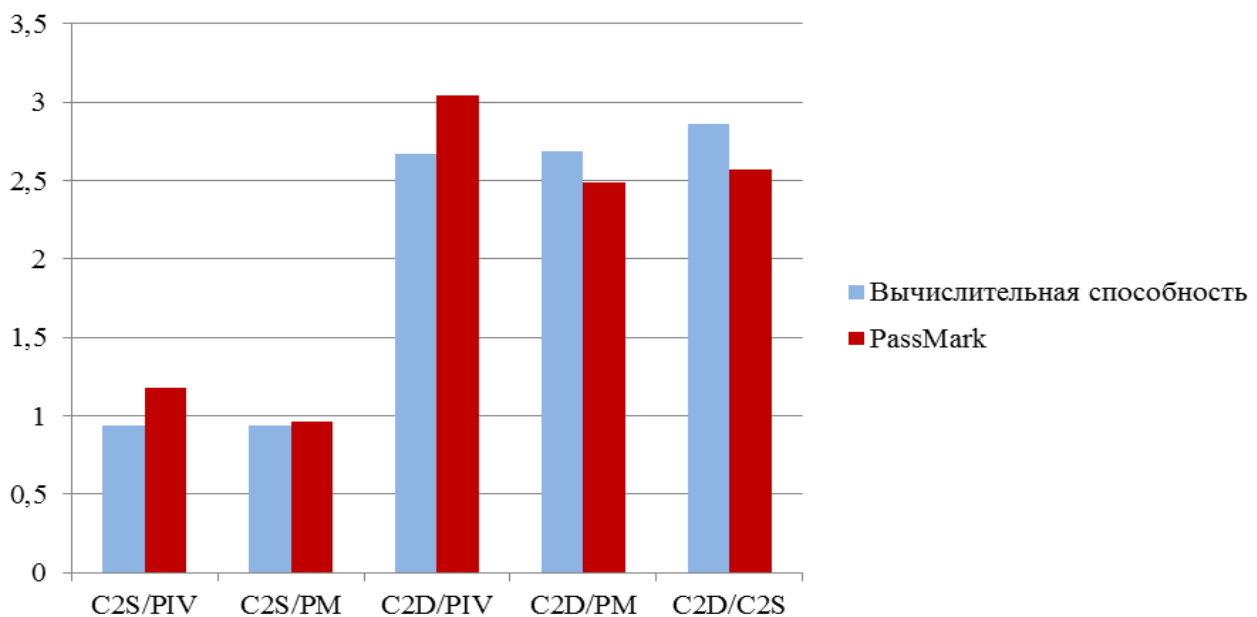


Рис. 5. Сравнение оценок производительности процессоров Intel Core 2 с процессорами предыдущих семейств

Для подведения итогов работы будет неплохо рассмотреть и сравнить между собой процессоры Intel Core 2, PM, P7 и P6. Из-за большого периода времени, в течение которого выпускались эти процессоры, для сводного сравнения хорошо подходит очень малое количество бенчмарков, поэтому мы будем сравнивать результаты нашей оценки производительности с помощью характеристики «вычислительная способность» с результатами оценки бенчмарком PassMark, который содержит в базе оценок достаточно большое для необходимого нам интервала количество процессоров. То есть он содержит сводную оценку всех описанных в табл. 7 процессоров.

Таблица 7. Сводная таблица характеристик процессоров (P6, P7, PM, Core 2)

Наименование процессора	Тактовая частота, МГц	PassMark	$C(I)$, бит/такт	Вычислительная способность, Мбит/с
Pentium III	1266	270	42.0223	53200.28
Pentium M	2000	495	51.198	102396.06
Core Solo T1350	1860	443	51.198	95228.34
Pentium IV	2600	405	39.6598	103115.51
Core 2 Solo U3500	1400	479	68.8469	96385.62
Core 2 Duo T7300	2000	1232	137.6937	275387.48

Для более качественного сравнения результатов, которые имеют различные единицы измерения, мы возьмём за единицу значение оценки для процессора Pentium III и построим графики оценок относительно этого значения. Полученный результат представлен на рис. 6.

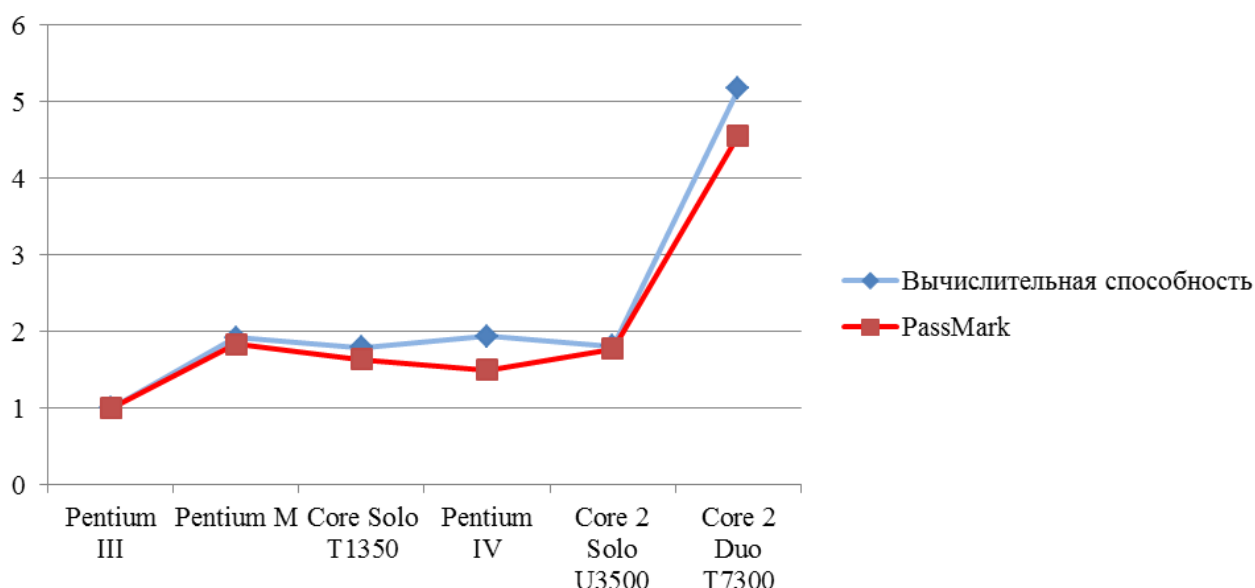


Рис. 6. Сравнительная диаграмма процессоров семейств P6, P7, PM, Core 2

Как видно на диаграмме, наша оценка имеет некоторые отклонения от кривой PassMark, но эти отклонения незначительны и вполне объяснимы, ведь оценка производительности процессора с помощью бенчмарков не является объективной в полной мере и результаты различных бенчмарков (как уже было показано выше) могут разительно отличаться друг от друга. Тем не менее общую тенденцию к росту наша характеристика отображает очень хорошо и поэтому может применяться на практике и является в достаточной степени объективной. Следующим шагом будет проверка состоятельности данной характеристики применимо к суперкомпьютерам, хотя частично она была уже подтверждена при оценке производительности процессоров с несколькими ядрами.

Литература

1. Ryabko B. An information-theoretic approach to estimate the capacity of processing units // Performance Evaluation. 2012. V. 69, P. 267-273.

2. *Shannon C. E.* A mathematical theory of communication // Bell Sys. Tech. J. 1948. V. 27, P. 379-423, P. 623-656.
3. *Ракитский А. А.* Практическое применение методов теоретической оценки вычислительной способности для процессоров Intel P5 // Вестник СибГУТИ. 2012.4. С. 50-61.
4. Intel x86 Quick Reference Instruction Manual - 8086/80186/80286/80386/80486. URL: <http://www.intel-assembler.it/portale/5/x86-instruction-reference-manual/x86-instruction-reference-manual.asp> (Дата обращения: 04.12.2012).
5. *Касперски К.* Секреты поваров компьютерной кухни или ПК: решение проблем. BHV, 2003. 560 с.
6. *Fog A.* The microarchitecture of Intel, AMD and VIA CPUs An optimization guide for assembly programmers and compiler makers. Copenhagen University College of Engineering. 2012-02-29. URL: <http://www.agner.org/optimize/> (Дата обращения: 04.12.2012).
7. *Fog A.* Lists of instruction latencies, throughputs and microoperation breakdowns for Intel, AMD and VIA CPUs. Copenhagen University College of Engineering. 2012. URL: <http://www.agner.org/optimize/> (Дата обращения: 04.12.2012).
8. Intel 64 and IA-32 Architectures Software Developers Manual Volume 1: Basic Architecture. Intel Corp. URL: <http://www.intel.ru/content/www/ru/ru/architecture-and-technology/64-ia-32-architectures-software-developer-vol-1-manual.html> (Дата обращения: 04.12.2012).
9. Intel 64 and IA-32 Architectures Software Developers Manual Volume 2. Intel Corp. URL: <http://www.intel.ru/content/www/us/en/architecture-and-technology/64-ia-32-architectures-software-developer-vol-2a-2b-instruction-set-a-z-manual.html> (Дата обращения: 15.03.2012).
10. *Tanenbaum A.S.* Structured computer organization. Prentice Hall PTR, 2001. 514 p.
11. *Marr D. T., Binns F., Hill D. L., Hinton G., Koufaty D. A., Miller J. A. and Upton M.* Hyper-threading technology architecture and microarchitecture // Intel Technology Journal. 2001. V. 06, Issue 01.

*Статья поступила в редакцию 14.03.2013;
переработанный вариант — 30.03.2013*

Ракитский Антон Андреевич

ассистент кафедры прикладной математики и кибернетики СибГУТИ,

тел.+7-923-101-1936, e-mail: rakitsky.anton@gmail.com

Theoretical evaluation of computational power of Intel processors

A. Rakitskiy

In this paper, we describe the application of innovative method of theoretical benchmark test of processors and computing systems to major Intel processors. This method is based on determining the computational power of the processor – characteristic based on the basic concepts of information theory. This method is applied to determine computational power of the main Intel processors (from 80486 to Core 2 Duo). Based on the characteristic values the processors are compared with each other and the final results are compared with the generally accepted benchmark.

Keywords: performance, computational power.