Министерство общего и профессионального образования Свердловской области

Профессионально-педагогический колледж

Кафедра дизайна сервиса и информационных технологий

Реферат

по архитектуре ЭВМ

Процессор

Специальность 230103 – Автоматизированные системы обработки информации и управления

Выполнил:

Студент группы 211 Т

И.Р.Гатауллин

Руководитель:

М. С. Огородов

Екатеринбург 2009

Содержание

Введение

1. Архитектура фон Неймана

2. Устройство центрального процессора

3. Системная шина

4. CISC, RISC, MISC процессоры

5. Конвейеры

6. Суперскалярные архитектуры

7. Кэш-память

8. Процессоры семейства AMD Phenom II

9. Процессоры семейства Intel Core i7

10. Core i7 920, Phenom II X4 920, Phenom X4 9950

Заключение

Введение

Сегодня мир без компьютера — это немыслимое явление. А ведь мало кто задумывается об устройстве этих "существ". И уж точно никто не знает, насколько умными стали данные аппараты за последние 50 лет. Для многих людей искусственный интеллект и компьютер, который стоит на вашем столе, — это одно и тоже. Но как люди просвещенные, мы знаем, что до разума человека, или даже собаки любой, даже самой умной, машине еще далеко. А ведь отличие все-таки есть: в мозге живых существ идет параллельная обработка видео, звука, вкуса, ощущений, и т. д., не говоря уже о такой элементарной вещи, как мыслительный процесс, который сопровождает многих от рождения и до самой смерти. Сегодня любой прорыв в информационных технологиях встречается как нечто особо выдающееся. Люди хотят создать себе младшего брата, который, если еще не думает, то хотя бы соображает быстрее их. Понятно, что никакими гигагерцами не измеришь уникум человеческого мозга, но никто и не измеряет, и мы проведем краткую экскурсию в недалекое прошлое и, конечно, в непонятное настоящее развития главной части компьютера, его мозга, его сердца — его центрального процессора. В данный момент эта тема очень актуальна, т.к. современные технологии развиваются стремительно, особенно процессоры. Цель моего реферата познакомиться с устройством центрального процессора, рассмотреть некоторые процессоры.

Для достижения этой цели я поставил перед собой следующие задачи:

* Узнать основные части процессора
* Для чего они нужны
* Познакомиться с линейкой процессоров Intel core i7 и AMD Phenom II X4.
* Сравнить некоторые процессоры.

# 1. Архитектура фон Неймана

Большинство современных процессоров для персональных компьютеров в общем основаны на той или иной версии циклического процесса последовательной обработки информации, изобретённого Джоном фон Нейманом. Д. фон Нейман придумал схему постройки компьютера в 1946 году.

Важнейшие этапы этого процесса приведены ниже. В различных архитектурах и для различных команд могут потребоваться дополнительные этапы. Например, для арифметических команд могут потребоваться дополнительные обращения к памяти, во время которых производится считывание операндов и запись результатов. Отличительной особенностью архитектуры фон Неймана является то, что инструкции и данные хранятся в одной и той же памяти.

Этапы цикла выполнения:

1. Процессор выставляет число, хранящееся в регистре счётчика команд, на шину адреса, и отдаёт памяти команду чтения;
2. Выставленное число является для памяти адресом; память, получив адрес и команду чтения, выставляет содержимое, хранящееся по этому адресу, на шину данных, и сообщает о готовности;
3. Процессор получает число с шины данных, интерпретирует его как команду (машинную инструкцию) из своей системы команд и исполняет её;
4. Если последняя команда не является командой перехода, процессор увеличивает на единицу (в предположении, что длина каждой команды равна единице) число, хранящееся в счётчике команд; в результате там образуется адрес следующей команды;
5. Снова выполняется п. 1.

Данный цикл выполняется неизменно, и именно он называется процессом (откуда и произошло название устройства).

Во время процесса процессор считывает последовательность команд, содержащихся в памяти, и исполняет их. Такая последовательность команд называется программой и представляет алгоритм полезной работы процессора. Очерёдность считывания команд изменяется в случае, если процессор считывает команду перехода — тогда адрес следующей команды может оказаться другим. Другим примером изменения процесса может служить случай получения команды остановки или переключение в режим обработки аппаратного прерывания.

Команды центрального процессора являются самым нижним уровнем управления компьютером, поэтому выполнение каждой команды неизбежно и безусловно. Не производится никакой проверки на допустимость выполняемых действий, в частности, не проверяется возможная потеря ценных данных. Чтобы компьютер выполнял только допустимые действия, команды должны быть соответствующим образом организованы в виде необходимой программы.

Скорость перехода от одного этапа цикла к другому определяется генератором тактовых импульсов. Генератор тактовых импульсов – генерирует последовательность электрических импульсов, частота которых определяет тактовую частоту процессора, промежуток времени между соседними импульсами, определяет время одного такта или просто такт работы машины. Частота генератора тактовых импульсов является одной из основных характеристик компьютера и во многом определяет скорость его работы, поскольку каждая операция выполняется за определенное количество тактов.

# 2. Устройство центрального процессора

Центральный процессор (ЦП; CPU — англ. céntral prócessing únit, дословно — центральное вычислительное устройство) — исполнитель машинных инструкций, часть аппаратного обеспечения компьютера или программируемого логического контроллера, отвечающая за выполнение арифметических операций, заданных программами операционной системы, и координирующий работу всех устройств компьютера.

На рис.1 показано устройство обычного компьютера. Центральный процессор — это мозг компьютера. Его задача — выполнять программы, находящиеся в основной памяти. Он вызывает команды из памяти, определяет их тип, а затем выполняет их одну за другой. Компоненты соединены шиной, представляющей собой набор параллельно связанных проводов, по которым передаются адреса, данные и сигналы управления. Шины могут быть внешними (связывающими процессор с памятью и устройствами ввода-вывода) и внутренними.

Процессор состоит из нескольких частей. Блок управления отвечает за вызов команд из памяти и определение их типа. Арифметико-логическое устройство выполняет арифметические операции (например, сложение) и логические операции (например, логическое И).

Внутри центрального процессора находится память для хранения промежуточных результатов и некоторых команд управления. Эта память состоит из нескольких регистров, каждый из которых выполняет определенную функцию. Обычно все регистры одинакового размера. Каждый регистр содержит одно число, которое ограничивается размером регистра. Регистры считываются и записываются очень быстро, поскольку они находятся внутри центрального процессора.

Самый важный регистр — счетчик команд, который указывает, какую команду нужно выполнять дальше. Название "счетчик команд" не соответствует действительности, поскольку он ничего не считает, но этот термин употребляется повсеместно. Еще есть регистр команд, в котором находится команда, выполняемая в данный момент. У большинства компьютеров имеются и другие регистры, одни из них многофункциональны, другие выполняют только какие-либо специфические функции.

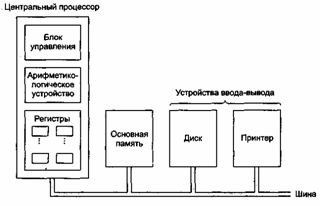


Рис.1 Схема устройства компьютера с одним центральным процессором и двумя устройствами ввода-вывода

# 3. Системная шина

Основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех устройств между собой, включая себя:

1. Кодовая шина данных (КШД) – содержит провода и схемы сопряжения для параллельной передачи всех разрядов машинного кода операнда.
2. Кодовая шина адреса (КША) – содержит провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода вывода внешнего устройства.
3. Кодовая шина инструкций (КШИ) – содержит провода и схемы сопряжения для передачи инструкций во все блоки машины.

Системная шина – обеспечивает три направления передачи информации:

1. Между процессором и основной памятью.

2. Между процессором и портами ввода вывода внешних устройств в режиме прямого доступа к памяти.

3. Между основной памятью и портами ввода вывода внешних устройств.

4. CISC, RISC, MISC процессоры

### CISC-процессоры

Complex Instruction Set Computer (CISC)— вычисления со сложным набором команд. Процессорная архитектура, основанная на усложнённом наборе команд. Типичными представителями CISC является семейство микропроцессоров Intel x86 (хотя уже много лет эти процессоры являются CISC только по внешней системе команд).

### RISC-процессоры

Reduced Instruction Set Computing (RISC) — вычисления с сокращённым набором команд. Архитектура процессоров, построенная на основе сокращённого набора команд. Характеризуется наличием команд фиксированной длины, большого количества регистров, операций типа регистр-регистр, а также отсутствием косвенной адресации. Концепция RISC разработана Джоном Коком из IBM, название придумано Дэвидом Паттерсоном .

Самая распространённая реализация этой архитектуры представлена процессорами серии PowerPC, включая G3, G4 и G5. Довольно известная реализация данной архитектуры — процессоры серий MIPS и Alpha.

### MISC-процессоры

Minimum Instruction Set Computing(MISC) — вычисления с минимальным набором команд. Дальнейшее развитие идей команды Чака Мура, который полагает, что принцип простоты, изначальный для RISC процессоров, слишком быстро отошёл на задний план. В пылу борьбы за максимальное быстродействие, RISC догнал и перегнал многие CISC процессоры по сложности. Архитектура MISC строится на стековой вычислительной модели с ограниченным числом команд (примерно 20-30 команд).

# 5. Конвейеры

Уже много лет известно, что главным препятствием высокой скорости выполнения команд является их вызов из памяти. Для разрешения этой проблемы разработчики придумали средство для вызова команд из памяти заранее, чтобы они имелись в наличии в тот момент, когда будут необходимы. Эти команды помещались в набор регистров, который назывался буфером выборки с упреждением. Таким образом, когда была нужна определенная команда, она вызывалась прямо из буфера, и не нужно было ждать, пока она считается из памяти. Эта идея использовалась еще при разработке IBM Stretch, который был сконструирован в 1959 году.

В действительности процесс выборки с упреждением подразделяет выполнение команды на два этапа: вызов и собственно выполнение. Идея конвейера еще больше продвинула эту стратегию вперед. Теперь команда подразделялась уже не на два, а на несколько этапов, каждый из которых выполнялся определенной частью аппаратного обеспечения, причем все эти части могли работать параллельно.

На рис. 2а изображен конвейер из 5 блоков, которые называются стадиями. Стадия С1 вызывает команду из памяти и помещает ее в буфер, где она хранится до тех пор, пока не будет нужна. Стадия С2 декодирует эту команду, определяя ее тип и тип операндов, над которыми она будет производить определенные действия. Стадия СЗ определяет местонахождение операндов и вызывает их или из регистров, или из памяти. Стадия С4 выполняет команду, обычно путем провода операндов через тракт данных. И наконец, стадия С5 записывает результат обратно в нужный регистр.

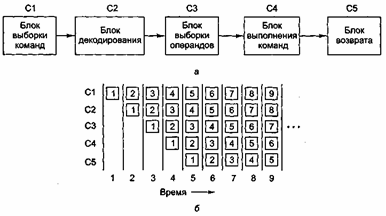


Рис.2 Конвейер из пяти стадий(а); состояния каждой стадии в зависимости от количества пройденных циклов(б) показано девять циклов

На рис. 2б мы видим, как действует конвейер во времени. Во время цикла 1 стадия С1 работает над командой 1, вызывая ее из памяти. Во время цикла 2 стадия С2 декодирует команду 1, в то время как стадия С1 вызывает из памяти команду 2. Во время цикла 3 стадия СЗ вызывает операнды для команды 1, стадия С2 декодирует команду 2, а стадия С1 вызывает третью команду. Во время цикла 4 стадия С4 выполняет команду 1, СЗ вызывает операнды для команды 2, С2 декодирует команду 3, а С1 вызывает команду 4. Наконец, во время пятого цикла С5 записывает результат выполнения команды 1 обратно в регистр, тогда как другие стадии работают над следующими командами.

Чтобы лучше понять принципы работы конвейера, рассмотрим аналогичный пример. Представим себе кондитерскую фабрику, на которой выпечка тортов и их упаковка для отправки производятся раздельно. Предположим, что в отделе отправки находится длинный конвейер, вдоль которого стоят 5 рабочих (или блоков обработки). Каждые 10 секунд (это время цикла) первый рабочий ставит пустую коробку для торта на ленту конвейера. Эта коробка отправляется ко второму рабочему, который кладет в нее торт. После этого коробка с тортом доставляется третьему рабочему, который закрывает и запечатывает ее. Затем она поступает к четвертому рабочему, который ставит на ней ярлык. Наконец, пятый рабочий снимает коробку с конвейерной ленты и помещает ее в большой контейнер для отправки в супермаркет. Примерно таким же образом действует компьютерный конвейер: каждая команда (в случае с кондитерской фабрикой — торт) перед окончательным выполнением проходит несколько шагов обработки.

Возвратимся к нашему конвейеру, изображенному на рис.2. Предположим, что время цикла у этой машины 2 нс. Тогда для того, чтобы одна команда прошла через весь конвейер, требуется 10 нс. На первый взгляд может показаться, что такой компьютер может выполнять 100 млн команд в секунду, в действительности же скорость его работы гораздо выше. Во время каждого цикла (2 нс) завершается выполнение одной новой команды, поэтому машина выполняет не 100 млн, а 500 млн команд в секунду.

Конвейеры позволяют найти компромисс между временем ожидания (сколько времени занимает выполнение одной команды) и пропускной способностью процессора (сколько миллионов команд в секунду выполняет процессор). Если время цикла составляет Т нс, а конвейер содержит п стадий, то время ожидания составит п\*Т нс, а пропускная способность — 1000/Т млн команд в секунду.

# 6. Суперскалярные архитектуры

Один конвейер — хорошо, а два — еще лучше. Одна из возможных схем процессора с двойным конвейером показана на рис.3. В основе разработки лежит конвейер, изображенный на рис.2. Здесь общий отдел вызова команд берет из памяти сразу по две команды и помещает каждую из них в один из конвейеров. Каждый конвейер содержит АЛУ для параллельных операций. Чтобы выполняться параллельно, две команды не должны конфликтовать при использовании ресурсов (например, регистров), и ни одна из них не должна зависеть от результата выполнения другой. Как и в случае с одним конвейером, либо компилятор должен следить, чтобы не возникало неприятных ситуаций (например, когда аппаратное обеспечение выдает некорректные результаты, если команды несовместимы), либо же конфликты выявляются и устраняются прямо во время выполнения команд благодаря использованию дополнительного аппаратного обеспечения.

Сначала конвейеры (как двойные, так и одинарные) использовались только в компьютерах RISC. У 386-го и его предшественников их не было. Конвейеры в процессорах компании Intel появились только начиная с 486-й модели.

Необходимо отметить, что параллельное функционирование отдельных блоков процессора использовалось и в предыдущем — 386-м — микропроцессоре. Оно стало прообразом 5-стадийного конвейера микропроцессора 486.

486-й процессор содержал один конвейер, a Pentium — два конвейера из пяти стадий. Похожая схема изображена на рис. 3, но разделение функций между второй и третьей стадиями (они назывались декодирование 1 и декодирование 2) было немного другим. Главный конвейер (u-конвейер) мог выполнять произвольные команды. Второй конвейер (v-конвейер) мог выполнять только простые команды с целыми числами, а также одну простую команду с плавающей точкой (FXCH).

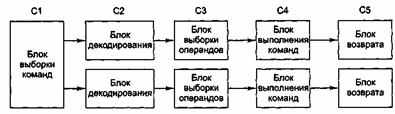


Рис.3 Двойной конвейер из пяти стадий с общим отделом вызова команд.

# 7. Кэш-память

Процессоры всегда работали быстрее, чем память. Процессоры и память совершенствовались параллельно, поэтому это несоответствие сохранялось. Поскольку на микросхему можно помещать все больше и больше транзисторов, разработчики процессоров использовали эти преимущества для создания конвейеров и суперскалярной архитектуры, что еще больше повышало скорость работы процессоров. Разработчики памяти обычно использовали новые технологии для увеличения емкости, а не скорости, что еще больше усугубляло проблему. На практике такое несоответствие в скорости работы приводит к следующему: после того как процессор дает запрос памяти, должно пройти много циклов, прежде чем он получит слово, которое ему нужно. Чем медленнее работает память, тем дольше процессору приходится ждать, тем больше циклов должно пройти.

Как мы уже говорили выше, есть два пути решения этой проблемы. Самый простой из них — начать считывать информацию из памяти, когда это необходимо, и при этом продолжать выполнение команд, но если какая-либо команда попытается использовать слово до того, как оно считалось из памяти, процессор должен приостанавливать работу. Чем медленнее работает память, тем чаще будет возникать такая проблема и тем больше будет проигрыш в работе. Например, если отсрочка составляет 10 циклов, весьма вероятно, что одна из 10 следующих команд попытается использовать слово, которое еще не считалось из памяти.

Другое решение проблемы — сконструировать машину, которая не приостанавливает работу, но следит, чтобы программы-компиляторы не использовали слова до того, как они считаются из памяти. Однако это не так просто осуществить на практике. Часто при выполнении команды загрузки машина не может выполнять другие действия, поэтому компилятор вынужден вставлять пустые команды, которые не производят никаких операций, но при этом занимают место в памяти. В действительности при таком подходе простаивает не аппаратное, а программное обеспечение, но снижение производительности при этом такое же.

На самом деле эта проблема не технологическая, а экономическая. Инженеры знают, как построить память, которая будет работать так же быстро, как и процессор, но при этом ее приходится помещать прямо на микросхему процессора (поскольку информация через шину поступает очень медленно). Установка большой памяти на микросхему процессора делает его больше и, следовательно, дороже, и даже если бы стоимость не имела значения, все равно существуют ограничения в размерах процессора, который можно сконструировать. Таким образом, приходится выбирать между быстрой памятью небольшого размера и медленной памятью большого размера. Мы бы предпочли память большого размера с высокой скоростью работы по низкой цене.

Интересно отметить, что существуют технологии сочетания маленькой и быстрой памяти с большой и медленной, что позволяет получить и высокую скорость работы, и большую емкость по разумной цене. Маленькая память с высокой скоростью работы называется кэш-памятью (от французского слова cacher "прятать"; В английском "cash" получило значение "наличные (карманные) деньги", то есть то, что под рукой. А уже из него и образовался термин "кэш", который относят к сверхоперативной памяти.). Ниже мы кратко опишем, как используется кэш-память и как она работает.

Основная идея кэш-памяти проста: в ней находятся слова, которые чаще всего используются. Если процессору нужно какое-нибудь слово, сначала он обращается к кэш-памяти. Только в том случае, если слова там нет, он обращается к основной памяти. Если значительная часть слов находится в кэш-памяти, среднее время доступа значительно сокращается.

Таким образом, успех или неудача зависит от того, какая часть слов находится в кэш-памяти. Давно известно, что программы не обращаются к памяти наугад. Если программе нужен доступ к адресу А, то скорее всего после этого ей понадобится доступ к адресу, расположенному поблизости от А. Практически все команды обычной программы (за исключением команд перехода и вызова процедур) вызываются из последовательных участков памяти. Кроме того, большую часть времени программа тратит на циклы, когда ограниченный набор команд выполняется снова и снова. Точно так же при манипулировании матрицами программа скорее всего будет обращаться много раз к одной и той же матрице, прежде чем перейдет к чему-либо другому.

То, что при последовательных отсылках к памяти в течение некоторого промежутка времени используется только небольшой ее участок, называется принципом локальности. Этот принцип составляет основу всех систем кэш-памяти. Идея состоит в следующем: когда определенное слово вызывается из памяти, оно вместе с соседними словами переносится в кэш-память, что позволяет при очередном запросе быстро обращаться к следующим словам. Общее устройство процессора, кэш-памяти и основной памяти показано на рис.4. Если слово считывается или записывается k раз, компьютеру понадобится сделать 1 обращение к медленной основной памяти и k-1 обращений к быстрой кэш-памяти. Чем больше k, тем выше общая производительность.

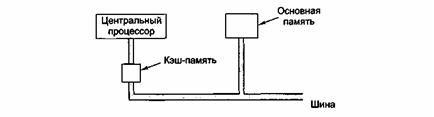


Рис.4 Кэш-память находиться между процессором и основной памятью.

Мы можем сделать более строгие вычисления. Пусть с — время доступа к кэш памяти, m — время доступа к основной памяти и h — коэффициент совпадения, который показывает соотношение числа ссылок к кэш-памяти и общего числа всех ссылок. В нашем примере h=(k-l)/k. Таким образом, мы можем вычислить среднее время доступа:

Среднее время доступа =c+(l-h)m.

Если h—"1 и все обращения делаются только к кэш-памяти, то время доступа стремится к с. С другой стороны, если h—>0 и каждый раз нужно обращаться к основной памяти, то время доступа стремится к с+т: сначала требуется время с для проверки кэш-памяти (в данном случае безуспешной), а затем время m для обращения к основной памяти. В некоторых системах обращение к основной памяти может начинаться параллельно с исследованием кэш-памяти, чтобы в случае неудачного поиска цикл обращения к основной памяти уже начался. Однако эта стратегия требует способности останавливать процесс обращения к основной памяти в случае результативного обращения к кэш-памяти, что делает разработку такого компьютера более сложной.

Основная память и кэш-память делятся на блоки фиксированного размера с учетом принципа локальности. Блоки внутри кэш-памяти обычно называют строками кэш-памяти. Если обращение к кэш-памяти нерезультативно, из основной памяти в кэш-память загружается вся строка, а не только необходимое слово. Например, если строка состоит из 64 байтов, обращение к адресу 260 повлечет за собой загрузку в кэш-память всей строки, то есть с 256-го по 319-й байт. Возможно, через некоторое время понадобятся другие слова из этой строки. Такой путь обращения к памяти более эффективен, чем вызов каждого слова по отдельности, потому что вызвать k слов 1 раз можно гораздо быстрее, чем 1 слово k раз. Если входные сообщения кэш-памяти содержат более одного слова, это значит, что будет меньше таких входных сообщений и, следовательно, меньше непроизводительных затрат.

Разработка кэш-памяти очень важна для процессоров с высокой производительностью. Первый вопрос – размер кэш-памяти. Чем больше размер, тем лучше работает память, но тем дороже она стоит. Второй вопрос – размер строки кэш-памяти. Кэш-память объемом 16 Кбайт можно разделить на 1К строк по 16 байтов, 2К строк по 8 байтов и т. д. Третий вопрос – как устроена кэш-память, то есть как она определяет, какие именно слова содержатся в ней в данный момент

Четвертый вопрос — должны ли команды и данные находиться вместе в общей кэш-памяти. Проще разработать смежную кэш-память, в которой хранятся и данные, и команды. При этом вызов команд и данных автоматически уравновешивается. Тем не менее в настоящее время существует тенденция к использованию разделенной кэш-памяти, когда команды хранятся в одной кэш-памяти, а данные — в другой. Такая структура также называется Гарвардской (Harvard Architecture), поскольку идея использования отдельной памяти для команд и отдельной памяти для данных впервые воплотилась в компьютере Магc III, который был создан Гавардом Айкеном в Гарварде. Современные разработчики пошли по этому пути, поскольку сейчас широко используются процессоры с конвейерами, а при такой организации должна быть возможность одновременного доступа и к командам, и к данным (операндам). Разделенная кэш-память позволяет осуществлять параллельный доступ, а общая — нет. К тому же, поскольку команды обычно не меняются во время выполнения, содержание командной кэш-памяти никогда не приходится записывать обратно в основную память.

Наконец, пятый вопрос — количество блоков кэш-памяти. В настоящее время очень часто кэш-память первого уровня располагается прямо на микросхеме процессора, кэш-память второго уровня — не на самой микросхеме, но в корпусе процессора, а кэш-память третьего уровня — еще дальше от процессора.

# 8. Процессоры семейства AMD Phenom II

В начале года, 8 января, компания AMD представила новую платформу AMD Dragon, основанную на процессоре нового семейства AMD Phenom II. Первоначально компания AMD продемонстрировала лишь два процессора данного семейства: AMD Phenom II Х4 940 и AMD Phenom II Х4 920, которые совместимы с разъемом АМ2+ и поддерживают память DDR2. Позднее были представлены процессоры семейства AMD Phenom II, совместимые с разъемом АМЗ и поддерживающие как DDR2, так и DDR3-память.

Модельный ряд процессоров семейства AMD Phenom II

Главное отличие новых процессоров семейства AMD Phenom II от процессоров семейства AMD Phenom заключается в том, что они выполнены по 45-нм техпроцессу с применением технологии S0I, в то время как процессоры семейства AMD Phenom выполняются по 65-нм техпроцессу.

Точно так же, как и процессоры семейства AMD Phenom, они представляют собой истинно многоядерные процессоры, то есть все ядра процессора выполнены на одном кристалле.

Среди нововведений, реализованных в новых процессорах AMD Phenom II, можно также отметить усовершенствованную технологию AMD Cool'&'Quiet 3.0. Она объединяет в себе ряд функций, позволяющих снизить энергопотребление процессора в те моменты, когда он недозагружен, а также предотвратить перегрев процессора.

При анонсе нового процессора семейства AMD Phenom IIХ4 компания AMD указывала и на другие преимущества в сравнении с предыдущим семейством. В частности, отмечалось, что новые процессоры выполняют больше инструкций за такт (Instruction Per Clock, IPC).

Семейство процессоров AMD Phenom II в настоящее время включает три серии: AMD Phenom II Х4 900, AMD Phenom II Х4 800 и AMD Phenom II ХЗ 700.

Процессоры серии AMD Phenom II Х4 900

Сейчас в 900-ю серию процессоров входят две четырехъядерные модели: AMD Phenom II Х4 940 и AMD Phenom IIХ4 920. Каждое ядро процессора AMD Phenom IIХ4 900-й серии имеет выделенный L-2-кэш размером 512 Кбайт и разделяемый между всеми ядрами L3-кэш размером 6 Мбайт.

Процессор AMD Phenom II Х4 940 имеет тактовую частоту 3,0 ГГц, а процессор AMD Phenom II Х4 920 — 2,8 ГГц. Эти процессоры оснащены интегрированным двухканальным контроллером памяти DDR2 и поддерживают память DDR2 667/800/1066.

Процессоры AMD Phenom II Х4 940 и AMD Phenom IIХ4 920 совместимы с разъемами Socke АМ2+/АМ2 и поддерживают шину HyperTransport 3.0 на скорости до 3600 МГц (двусторонняя) пропускной способностью до 16 Гбайт/с. Оба процессора имеют TDP 125 Вт.

Разница между моделями процессоров AMD Phenom IIХ4 940 и AMD Phenom IIХ4 920 заключается не только в тактовой частоте, но еще и том, что процессор AMD Phenom II Х4 940 имеет разблокированный множитель, что позволяет реализовывать его эффективный разгон. Вообще, если говорить о разгонном потенциал процессора AMD Phenom II Х4 940, то, по сообщениям независимых источников в Интернете, он достаточно большой. Так, есть данные что применение жидкого азота для охлаждения процессора позволило достичь рекордной тактовой частоты в 6 ГГц, а посредством обычного воздушного охлаждения этот процессор легко разгоняется до 4 ГГц.

Добавим также, что в скором времени ожидается появление процессора AMD Phenom IIХ4 910 которым будет иметь тактовую частоту 2,6 ГГц.

Процессоры серии AMD Phenom II Х4 800

На данный момент 800-я серия процессоров включает всего одну модель четырехъядерного процессора —AMD Phenom II Х4 810. Однако в скором времени ожидается появление еще одной модели AMD Phenom IIХ4 805.

Отличие процессоров 800-й серии от процессоров 900-й серии заключается в урезанном размере кэша L3 и в том, что в процессорах 800-й серии реализован контроллер памяти, поддерживающий память как DDR2, так и DDR3. Кроме того, процессоры 800-й серии совместимы как с разъемам Socket AM2+/AM2 так и с разъемом Socket AM3.

Каждое ядро процессора AMD Phenom IIX4 810 имеет выделенный L2-кэш размером 512 Кбайт и разделяемый между всеми ядрами L3-кэш размером 4 Мбайт. Процессор AMD Phenom II Х4 810 работает с тактовой частотой 2,6 ГГц. Он оснащен интегрированным двухканальным контроллером памяти DDR2 (поддерживается память DDR2-667/800/1066) и контроллером памяти DDR3 (поддерживается память DDR3-800/1066/1333). TDP процессора составляет 95 Вт.

Процессоры серии AMD Phenom II ХЗ 700

В настоящее время в 700-ю серию процессоров входят две модели: AMD Phenom II ХЗ 720 и AMD Phenom II ХЗ 710. Все процессоры 700-й серии являются трехъядерными. Каждое ядро процессора AMD Phenom II Х3 720 и AMD Phenom II ХЗ 710 имеет выделенный L2-кэш размером 512 Кбайт, а разделяемый между всеми ядрами L3-кэш имеет размер 6 Мбайт.

Как и процессоры 800-й серии, процессоры 700-й серии имеют интегрированный двухканальный контроллер памяти DDR2 (поддерживается память DDR2-667/800/1066) и контроллер памяти DDR3 (поддерживается память DDR3-800/1066/1333).

Процессор AMD Phenom II ХЗ 720 работает на тактовой частоте 2,8 ГГц, а процессор AMD Phenom II ХЗ 710 — на тактовой частоте 2,6 ГГц. Еще одно различие между AMD Phenom II ХЗ 720 и AMD Phenom II ХЗ 710 заключается в том, что в модели AMD Phenom II ХЗ 720 разблокирован множитель, а следовательно, его можно легко разгонять.

# 9. Процессоры семейства Intel Core i7



Рис.6 процессор Core i7-965

Процессоры семейства Nehalem, как и полагается первопроходцам новой платформы, будут представлены на рынке высокоуровневыми четырехъядерными решениями на базе ядра Bloomfield, а уже через год пополнятся доступными моделями, которые займут место прежних Core 2 Duo.

Процессор Nehalem

Новые процессоры, получившие название Core i7, изготовляются по технологическим нормам 45 нм с применением high-k диэлектрика и металлического затвора транзисторов, но в отличие от своих предшественников все четыре ядра расположены на одном кристалле. Если помните, Core 2 Quad состоит из двух ядер Core 2 Duo, объединенных в одном корпусе. Кроме того, процессоры Nehalem содержат кэш-память третьего уровня объемом 8 МБ, встроенный трехканальный контроллер памяти DDR3 и контроллер шины Quick Path Interconnect (QPI), которые потребовали значительное увеличение контактов – до 1366, из-за чего размеры CPU нового поколения стали больше и по форме он уже напоминает прямоугольник, а не квадрат как у Core 2. Естественно, ни о какой совместимости разъемов речи не идет.

Кстати, в название Core i7 отражено поколение процессоров, использующих архитектуру P6. Всего на данный момент доступно три модели новых CPU: Core i7-965 Extreme Edition, Core i7-940 и Core i7-920. Главное отличие между ними заключается в рабочей частоте ядер и шины QPI, которая пришла на смену "старушке" FSB, аналогично технологии HyperTransport от AMD. Естественно, экстремальная версия ориентирована на энтузиастов и оверклокеров, имеет более высокую частоту и разблокированный на повышение множитель. Также для Core i7-965 Extreme Edition характерно большее количество множителей для памяти, частота которой формируется путем их умножения на частоту тактового генератора (опорной частоты шины QPI или QPI bclk), равную в номинале 133 МГц. Частоты ядер, шины QPI и кэша L3 также формируются путем умножения определенных коэффициентов на опорную частоту. Если же разгонять процессор методом поднятия QPI bclk, то частоты всех блоков и памяти поднимутся в зависимости от их множителей. Обычные Intel Core i7 будут уже не столь дружелюбны к оверклокерам, но, возможно, со временем данную проблему все-таки решат.

Еще одним новшеством семейства Nehalem стало использование технологии Hyper-Threading (или Simultaneous Multithreading – SMT, технология "одновременной мультипоточности"), от которой отказались при переходе на архитектуру Core. Теперь же каждый процессор Core i7 определяется как восемь логических ядер, что может существенно повысить быстродействие оптимизированных под многопоточность приложений.

Несмотря на перенос части северного моста в CPU, уровень TDP не превышает 130 Вт, что даже ниже чем у 45-нм Intel Core 2 Extreme QX9770 на недавно вышедшем степпинге C0. Связано это как с монолитностью кристалла, так и с меньшим объемом кэша – у QX9770 он составляет 12 МБ, тогда как Core i7 довольствуется кэш-памятью общим объемом в 9 МБ. Но даже с таким уровнем TDP, системы охлаждения для новых процессоров немного выросли в размерах, а монтажные отверстия в материнских платах не совпадают с креплениями от кулеров под Socket LGA775. Учитывая, что сейчас процессоры в большинстве случаев поставляются в коробочном исполнении, то вряд ли стоит переживать на этот счет. Для разгона, конечно, придется подыскать кулер поэффективнее или крепление для старой, но мощной системы охлаждения.

Все основные характеристики процессоров Core i7 занесены в табл. 1, представленную ниже.

Табл.1 Характеристики процессоров Intel i7

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | Intel Core i7-965 Extreme Edition | Intel Core i7-940 | Intel Core i7-920 |
| Разъём | LGA1366 | LGA1366 | LGA1366 |
| Техпроцесс | 45-нм, с применением high-k диэлектриков | 45-нм, с применением high-k диэлектриков | 45-нм, с применением high-k диэлектриков |
| Число ядер | 4 (8 потоков) | 4 (8 потоков) | 4 (8 потоков) |
| Частота | 3,20 ГГц | 2,93 ГГц | 2,66 ГГц |
| L1 кэш | 4 x 32+32 КБ | 4 x 32+32 КБ | 4 x 32+32 КБ |
| L2 кэш | 4 x 256 КБ | 4 x 256 КБ | 4 x 256 КБ |
| L3 кэш | 8 МБ | 8 МБ | 8 МБ |
| Напряжение | 1,20 В | 1,20 В | 1,20 В |
| TDP | 130 Вт | 130 Вт | 130 Вт |

# 10. Core i7 920, Phenom II X4 920, Phenom X4 9950

Многие говорят, что, "AMD уже не те, что раньше". Действительно, успешная линейка Intel Core 2, а также ее преемник Core i7 позволила Intel занять практически не штурмуемую позицию на рынке hi-end систем. Впрочем, списывать калифорнийцев со счетов пока еще рано – во-первых, AMD осваивает бюджетный сектор, а во-вторых, все агрессивнее продвигает Phenom II, представленной на платформе Dragon. Что важнее — быть "быстрее, выше, сильнее!", или скромно предлагать недорогие и практичные решения для масс? Похоже, Advanced Micro Devices, представляя платформу Dragon, делали упор на второе. Противопоставляется Dragon не чему-нибудь, а платформе с Core i7; при этом AMD особо акцентирует внимание на соотношении цены и производительности, легкости модернизации и доступность простому пользователю.

Табл.2 Характеристики процессоров Core i7 920, Phenom II X4 920, Phenom X4 9950

|  |  |  |  |
| --- | --- | --- | --- |
|  | Core i7 920 | Phenom II X4 920 | Phenom X4 9950 |
| Ядро | Bloomfield | Deneb | Agena |
| Микроархитектура | Nehalem | Stars | Stars |
| Техпроцесс, нм | 45 | 45 | 65 |
| Кол-во транзисторов, млн. шт. | 731 | 758 | 450 |
| Площадь кристалла, кв. мм. | 263 | 258 | 285 |
| Тактовая частота, МГц | 2660 | 2800 | 2600 |
| Кэш L1, Кбайт | 4 x 32+32 | 4 x 64+64 | 4 x 64+64 |
| Кэш L2, Кбайт | 4 x 256 | 4 x 512 | 4 x 512 |
| Кэш L3, Мбайт | 8 | 6 | 2 |
| Тепловыделение, Вт | 130 | 125 | 125 |
| Цена, $ | 302 | 249 | 200 |

Не секрет, что желающие заиметь в составе домашнего ПК Core i7 вместе с процессором вынуждены в довесок обновлять материнскую плату и память. Совершенно не факт, что приживутся старые модули: камни под LGA1366 не выносят высоких напряжений на оперативке. Даже после падения цен, покупка такой системы влетает в копеечку. Материнские платы дешевле двухсот долларов найти сложно, а цены на младший Core i7 начинаются от 274$ (не забываем про выросший курс). Правда, цены на память значительно снизились – раньше двухгигабайтный комплект стоил почти 200$, сейчас эта цифра упала почти до 50$. Итого мы получаем около 580$ за базовый комплект на iCore7 (и это если не потребуется менять, например, блок питания или корпус).

Ценовой расклад по версии AMD:

AMD же предлагает платформу ценой 230$ (Phenom II X4 920) + 100$ (Gigabyte GA-MA790X-DS4) + 80$ (2 x 1024 Мбайт DDR2-1066) = 410$. Действительно, значительный отрыв (40%) сохраняется даже в пересчете на сегодняшние московские цены, а 170$ - немаленькая сумма для среднестатистического столичного жителя, не говоря уже о регионах. С точки зрения финансовой выгоды для потребителя AMD обыгрывает своего конкурента. Сравнивать производительность новинки мы будем с младшим представителем Core i7, 920-й моделью, Phenom X4 9950 и Phenom II X4 920.

Табл.3 Характеристика компьютера, на котором производится тестирование

|  |  |  |
| --- | --- | --- |
|  | Core i7 920 | Phenom II X4 920 / Phenom X4 9950 |
| Материнская плата | MSI X58 Platinum | MSI DKA790GX Platinum |
| Видеокарта | BFG GTX 295 | BFG GTX 295 |
| Оперативная память | Kingson HyperX DDR3-1333 | Corsair Dominator DDR2-1066 |
| Блок питания | BFG 800 Вт | BFG 800 Вт |
| Жесткий диск | WD Raptor 150 Гбайт | WD Raptor 150 Гбайт |
| Операционная система | Windows Vista SP1 | Windows Vista SP1 |

Подбирая тестовые приложения, мы старались охватить все области: игры, работу в двумерных и трехмерных редакторах, кодирование видео, архивацию данных.

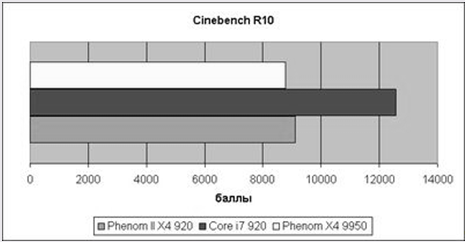


Рис.6 Результаты Cinebench R10

Cinebench R10 отдает свой голос за Core i7 – спасибо улучшенному Hyper-threading и оптимизации под многопоточные приложения. Обновленный Phenom обходит предшественника из-за более высокой тактовой частоты; изменения в ядре и возросший кэш не помогают совершенно.

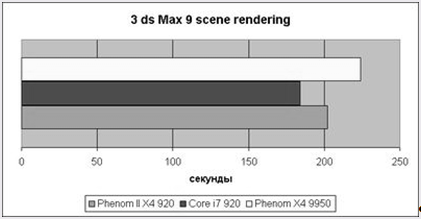


Рис.7 Результаты 3ds Max 9

В 3ds Max 9 отставание Deneb от Nehalem уже не столь велико, менее десяти процентов. А вот недостатки Agena проявляются сильнее – десятипроцентный прирост меньше объявленного AMD в пресс-релизе (25%), но уже что-то.

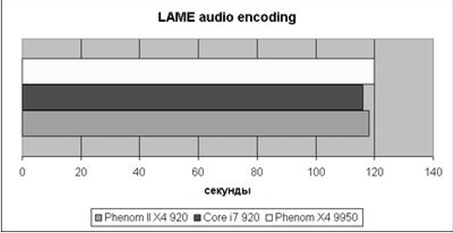


Рис.8 Результаты LAME encoding

Перекодирование аудио – однопоточная задача, так что разница между всеми тремя участниками теста невелика.

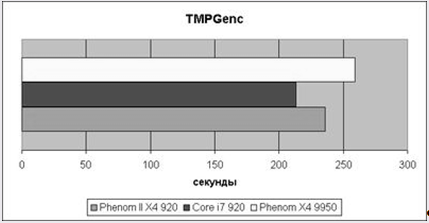


Рис.9 Результаты TMPGenc encoding

С конвертированием видео ситуация лучше, около 10% отрыва между соревнующимися. Расстановка сил не претерпела изменений – лидирует Core i7, за ним Phenom II X4, и в хвосте "старичок" 9950.

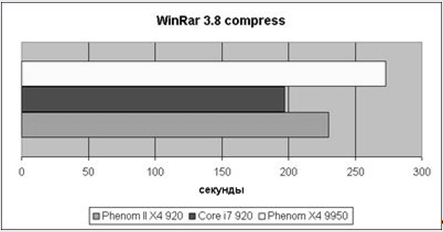


Рис.10 Результаты WinRar 3.80

WinRar 3.80 никаких сюрпризов не подносит – та же картина с незначительными изменениями.

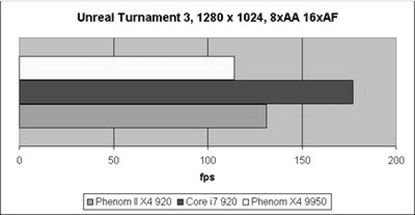


Рис.11 Результаты UT3

Unreal Tournament 3 отдал предпочтение продукции Intel, видимо, за счет разницы в архитектуре.

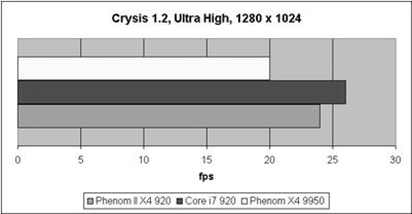


Рис.12 Результаты Crysis 1.2

В Crysis отставание Phenom X4 от новичка обусловлено ростом тактовой частоты; Core i7, как всегда, на первом месте.

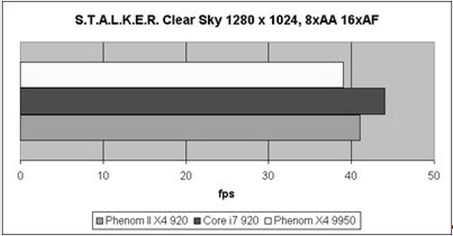


Рис.13 Результаты S.T.A.L.K.E.R. Clear Sky

S.T.A.L.K.E.R. Clear Sky не выбивается из общего порядка

Общий итог по тестам – Phenom II X4 шустрее первого на 10%, и примерно на столько же проигрывает младшему Core i7. Если же принять во внимание стоимость, то картина меняется: платформа Dragon обходит самую дешевую конфигурацию на Nehalem по соотношению цена/производительность.

Что же, удался новый процессор у AMD? Да, без сомнения. Хоть он и не конкурент по скорости Core i7 (да и четырехядерники на Core 2 Quad по отдельным показателям лучше), но в сравнении с Phenom X4 сделан значительный шаг вперед. Прибавим к этому грамотное позиционирование продукта (в составе платформы), совместимость с уже вышедшими материнскими платами, и дешевизну. У Dragon определенно есть будущее.

# Заключение

Познакомившись с центральным процессором, я понял что это одно из самых важных устройств компьютера от которого зависит работа всего компьютера, состоящее из множества компонентов, которые постоянно модернизируются и улучшаются.

Если говорить о сравнении процессоров Intel core i7 920, AMD Phenom II X4 920 и AMD Phenom X4 9950, то можно сказать, что в целом AMD Phenom II X4 по производительности стоит вровень с процессорами Intel предыдущего поколения (Core 2 Quad) и весьма значительно отстает от Intel Core i7. Но в сравнении AMD Phenom X4 и AMD Phenom II X4 сделан значительный шаг вперёд. Говоря о цене; Intel, то он стоит почти в два раза больше, чем AMD.