Лекция 24.10

У МАЕЙ ДЭВУШКЕ ДЭНЬ РАЖДЭЭНИЯ

Пентиум pro ’95 год

1. Инструкции попадают в конвейер последовательно,
2. затем разбиваются на мелки операции,
3. выполняются ядром неупорядоченно,
4. на выходе упорядочиваются

в 97 году MMX (multimedia extension)

ядро называлось п5 (5ое поколение)

что добавили: регистры mmx (появился блок работы с командами mmx) (регистры работали с целыми) (производительность увеличилась в 60 %), кэш был раздельным, 57 инструкций для работы с mmx (57 «симдов» (simd – single instruction multiply data), 4 конвейера по 14 ступеней.

Пентиум 2 (’97) - объединенная версия mmx и pro, увеличился кэш

Пентиум 3 (’99) – аналог пентиум 2, но появился новый набор регистров SSE (streaming simd extension), xmm регистры 8 регистров 128 разрядов. Обрабатывались числа по 32 бита (число с плавающей точкой большой точности), был добавлен кэш 2го уровня, 5 конвейеров по 11 ступеней. Шина данных 64, шина адреса 66 (начиная с пентиум про)

Пентиум 4 (’00) (первый процессор 7го поколения и последний 32х разрядный) – совершенно новая архитектура, 144 новых команд для работы с ссм. Регистры те же самые, но стало возможно вычислять целые числа + два числа с плавающей точкой сразу (т.е. 64 уже).

Net bvrst черты:   
1. Гиперконвейеризация   
2. Применение кэша последовательности микроопераций вместо микроинструкций  
3. АЛУ было разбито на блоки (3 шт) – два быстрых АЛУ и одно медленное. Быстрое – выполняло только простые целочисленные операции, а медленное могло выполнять сразу большое количество операций. Быстрые АЛУ работали на удвоенной частоте ядра (это было достигнуто изза работы на спаде импульса и на подъеме импульса)  
4. Повторное исполнение микроопераций (в кэше лежит блок микроопераций, они быстро загружаются и исполняются, но иногда они бывают не готовы чтоб операция было запущенна, тогда программа «планировщик», которая рассчитывает время, через которое будет выполненная эта операция. И если ничего не получилось, планировщик плохо посчитал, то такая микрооперация отправляется в очередь на повторное исполнение.

В связи с этим есть недостатки:

1. Задачи должны быть оптимизированы под эту архитектуру, иначе выигрыша во времени не будет.
2. Тактовая частота увеличивается – отсюда увеличивается тепловыделение (поэтому потребовались системы экстремального охлаждения)

Поэтому пентиум 4 стал последним одноядерным процессором

Пентиум dual (та же самая архитектура net bvrst, но теперь 2 ядра от пентиум 4) – 64х разрядная архитектура.

Гипертрейдинг технолоджи (ht/htt (hyper Threading technology)). – многопоточность, когда у ядра два набора регистров и два контроллера прерывания (отсюда экономится место в кристалле и увеличивается производительность). Впервые это использовалось в 2002 году в процессорах xeon и Pentium 4. Площадь увеличилась на 5%, а прозводтельность аж на 15-30%.

Цикл команды.

(-это выборка декодирования исполнения.)

1. Определение исполнительного адреса команды (инстракшн поинтер) (адрес текущей команды плюс длина команды)  
2. Выборка команды (адрес из айпи выставляется на шину адреса, а на шину управления код команды чтения из памяти) а память читает данные и выставляет их на шину данных, процессор получает шину данных, она декодируется и исполняется.  
3. Вычисление адреса операнда (если есть). Имя переменной оно является адресом ячейки памяти, то есть по нему все делаеца. То есть происходит выборка операнда.   
4. Выполнение операции с данными.   
5. Запись результата либо в память, либо в устройство вывода.

Особенности цикла команды:

1.По адресу айпи считывается блок команды, пишется в кэш, команды выполняются конвейером одновременно (параллельно) в порядке удобном процессору  
2. Если предсказания конвейера не верно, то цикл перегружается.   
3. Выборка операндов считывается в блок данных, отправляется в кэш данных и данные считываются уже из кэша.

Существует вторичное и первичное декодирование  
первичное – определяет тип команды и ее длину  
вторичное – получает команду и сразу знает что с ней делать по ее коду.

В любой момент может произойти прерывание от устройства. Обработка прерывания происходит следующим образом: у процессора есть специальный вход, на который устройство передает информацию о прерывании.   
Процессор принимает этот сигнал и   
1. Сохраняет контекст программы (состояние айпи, состояние аккумулятора)  
2. По сигналу понимает от какого устройства произошел сигнал и   
3. идет в таблицу прерываний,   
4. читает адрес, по которому находится обработчик прерываний,   
5. переходит и начинает выполнять программу, которая находится по этому адресу. 6.Восстанавливается контекст программы   
7. продолжается ее выполнение.

Это всё были особенности. =)

Какая то хуйня хз, но я пофоткал

