ОБЗОР УРОВНЕЙ РАБОТЫ ПРОЦЕССОРА, ФУНКЦИИ АРХИТЕКТУРЫ КОМАНД.

Подготовили:
Бутабаева А.,Ержанулы Е.,
Исмаилова Ж.,Кабидоллаева А.,
Каленов А.

Процессор

– это основное устройство ЭВМ, выполняющее логические и арифметические операции, и осуществляющее управление всеми компонентами ЭВМ. Процессор представляет собой миниатюрную тонкую кремниевую пластинку прямоугольной формы, на которой размещается огромное количество транзисторов, реализующих все функции, выполняемые процессором. Кремневая пластинка – очень хрупкая, а так как ее любое повреждение приведет к выходу из строя процессора, то она помещается в пластиковый или керамический корпус.

Принцип работы процессора

Процессор является одним из тех устройств, которые все время должен работать. Процессор ПК не может быть выключен. Даже если на наш взгляд процессор ничего не делает, все равно выполняется какая-то программа.

Процессор работает, по сравнению с другими устройствами компьютера, с наибольшей скоростью. И самыми медленными по сравнению с ним являются внешние устройства, в том числе и человек. Так, например, работая с клавиатурой, человек отправляет в компьютер в среднем один байт в секунду (нажимает на одну клавишу в секунду) . Процессор обрабатывает такую и формацию за 0,000001 секунды.

А что же делает процессор в остальное время, если он не может выключаться? А в остальное время он может получать сигналы от мыши, от других компьютеров, от гибких и жестких дисков. Он успевает несколько раз в течение секунды подзарядить оперативную память, обслужить внутренние часы компьютера, отдать распоряжение, как правильно отображать информацию на экране, и выполнить множество прочих дел.

Система команд процессора

Процессор обрабатывает информацию, выполняя определенные команды. Таких команд может быть более тысячи. У каждой команды есть свой код (номер) . Например, есть команда 000, 001, 002 и т. д. Коды всех команд процессора записаны в двоичной форме в специальном документе, который называется системой команд процессора.

У каждого процессора своя система команд, поэтому один и тот же код для, разных процессоров может обозначать разные команды. Если же процессоры имеют ограниченную совместимость, то их рассматривают как семейство. Примером семейства процессоров являются все процессоры Intel. Их родоначальником был процессор Intel 8086, на базе которого был сделан первый IBM PC. Процессоры семейства совместимы «сверху вниз» , т. е. новый процессор понимает» все команды своих предшественников, но не наоборот.

Уровень архитектуры команд имеет особое значение: он является связующим звеном между программным и аппаратным обеспечением. Конечно, можно было бы сделать так, чтобы аппаратное обеспечение сразу непосредственно выполняло программы, написанные на C, C+, FORTRAN 90 или других языках высокого уровня, но это не очень хорошая идея. Преимущество компиляции перед интерпретацией было бы тогда потеряно. Кроме того, из чисто практических соображений компьютеры должны уметь выполнять программы, написанные на разных языках, а не только на одном.

Уровень архитектуры команд связывает компиляторы и аппаратное обеспечение. Это язык, который понятен и компиляторам, и аппаратному обеспечению.

Набор команд ***уровня архитектуры команд*** полностью отличается от набора команд микроархитектурного уровня. И сами операции, и форматы команд на этих двух уровнях различны. Наличие нескольких одинаковых команд случайно.

Уровень архитектуры команд расположен между микроархитектурным уровнем и уровнем операционной системы. Исторически этот уровень развился прежде всех остальных уровней и изначально был единственным.

Однако когда программа выполняет команду уровня архитектуры команд, эта команда выполняется непосредственно микроархитектурным уровнем без участия операционной системы.

Некоторые команды уровня операционной системы идентичны командам уровня архитектуры команд. Эти команды сразу выполняются микропрограммой, а не операционной системой.

Набор команд уровня операционной системы содержит большую часть команд из уровня архитектуры команд, а также несколько новых очень важных команд. Некоторые ненужные команды в уровень операционной системы не включаются. **Ввод-вывод - это одна из областей, в которых эти два уровня различаются очень сильно.**

Причина такого различия проста. Во-первых, пользователь, способный выполнять команды ввода-вывода уровня архитектуры команд, сможет считать конфиденциальную информацию, которая хранится где-нибудь в системе, и вообще будет представлять угрозу для самой системы. Во-вторых, обычные нормальные программисты не хотят осуществлять ввод-вывод на уровне команд, поскольку это слишком сложно и утомительно. Вместо этого для осуществления ввода-вывода нужно установить определенные поля и биты в ряде регистров устройств, затем подождать, пока операция закончится, и проверить, что произошло. Диски обычно содержат биты регистров устройств для обнаружения следующих ошибок.

Уровень команд - это промежуточное звено между компиляторами и аппаратным обеспечением.

Все это вовсе не значит, что разработка уровня команд не имеет никакого значения. Хорошо разработанный уровень архитектуры команд имеет огромные преимущества перед плохим, особенно в отношении вычислительных возможностей и стоимости.

Над цифровым логическим уровнем находится микроархитектурный уровень. Его задача - интерпретация команд уровня 2 (уровня архитектуры команд.Строение микроархитектурного уровня зависит от того, каков уровень архитектуры команд, а также от стоимости и предназначения компьютера. В других системах (например, в системах Pentium II) на этом уровне имеются более сложные команды; выполнение одной такой команды занимает несколько циклов.

Чтобы выполнить команду, нужно найти операнды в памяти, считать их и записать полученные результаты обратно в память. Управление уровнем команд со сложными командами отличается от управления уровнем команд с простыми командами, так как в первом случае выполнение одной команды требует определенной последовательности операций.

Эта машина должна была иметь встроенный неизменяемый интерпретатор (микропрограмму), функция которого заключалась в выполнении программ посредством интерпретации. Так как аппаратное обеспечение должно было теперь вместо программ уровня архитектуры команд выполнять только микропрограммы с ограниченным набором команд, требовалось меньшее количество электронных схем. Поскольку электронные схемы тогда делались из электронных ламп, такое упрощение должно было сократить количество ламп и, следовательно, увеличить надежность.

Хотя самые первые компьютеры работали в основном с числами, современные компьютеры часто используются для нечисловых приложений, например, для обработки текстов или управления базой данных. Для этих приложений нужны другие, нечисловые, типы данных. Они часто поддерживаются командами уровня архитектуры команд. Здесь очень важны символы, хотя не каждый компьютер обеспечивает аппаратную поддержку для них. Наиболее распространенными символьными кодами являются ASCII и UNICODE. Они поддерживают 7-битные и 16-битные символы соответственно.

Архитектура RISC-процессоров характеризуется наличием команд фиксированной длины, большого количества регистров, операций типа регистр-регистр, а также отсутствием косвенной адресации.

Главные усилия в архитектуре RISC направлены на построение максимально эффективного конвейера команд, то есть такого, где все команды извлекаются из памяти и поступают в центральный процессор (ЦП) на обработку в виде равномерного потока, причем ни одна команда не должна находиться в состоянии ожидания, а ЦП должен оставаться загруженным на протяжении всего времени.

Унификация набора команд, ориентация на конвейерную обработку, унификация размера команд и длительности их выполнения, устранение периодов ожидания в конвейере - все эти факторы положительно сказываются на общем быстродействии.
Недостатки RISC прямо связаны с некоторыми преимуществами этой архитектуры.
Принципиальный недостаток - сокращенное число команд: на выполнение ряда функций приходится тратить несколько команд вместо одной в CISC. Это удлиняет код программы, увеличивает загрузку памяти и трафик команд между памятью и ЦП.

CISC (Complete Instruction Set Computing) - тип архитектуры процессора с полным набором команд. Основоположником CISC-архитектуры считается фирма IBM с архитектурой IBM/360. При этом подходе выполнение любой сколь угодно сложной команды из системы команд процессора реализовывается аппаратно внутри самого процессора.

Основную идею CISC-архитектуры отражает ее название - «полный набор команд». В данной архитектуре стремятся иметь отдельную машинную команду для каждого возможного (типового) действия по обработке данных.

Исторически CISC-архитектура была одной из первых. Совершенствование процессоров шло по пути создания ВМ, способных выполнять как можно больше разных команд.

EPIC (Explicitly Parallel Instruction Computing) - архитектура процессора с явным параллелизмом команд. Термин введен в 1997 году фирмами HP и Intel для разрабатываемой архитектуры Intel Itanium. EPIC позволяет микропроцессору выполнять инструкции параллельно, опираясь на работу компилятора, а не выявляя возможность параллельной работы инструкций при помощи специальных схем. Это могло упростить масштабирование вычислительной мощности процессора без увеличения тактовой частоты.

В архитектуре EPIC, которая в изделиях Intel получила название IA-64 (Intel Architecture - 64), предполагается наличие в процессоре ста двадцати восьми 64-разрядных регистров общего назначения и ста двадцати восьми 80-разрядных регистров с плавающей запятой. Кроме того, процессор IA-64 содержит 64 однобитовых регистра предикатов.

VLIW (Very Long Instruction Word - очень длинная машинная команда) - архитектура процессоров, характеризующаяся возможностью объединения нескольких простых команд в так называемую связку. Входящие в нее команды должны быть независимы друг от друга и выполняться параллельно. Таким образом, из нескольких независимых машинных команд транслятор формирует одно «очень длинное командное слово».

Цель анализа: обнаружить все команды, которые могут быть выполнены одновременно, причем так, чтобы между командами не возникали конфликты. В ходе анализа компилятор может даже частично имитировать выполнение рассматриваемой программы. На следующем этапе компилятор пытается объединить такие команды в пакеты (связки), каждый из которых рассматривается как одна сверхдлинная команда.

Вопрос 1

- Что такое процессор?

- Основное устройство ЭВМ, выполняющее логические и арифметические операции, и осуществляющее управление всеми компонентами ЭВМ. Процессор представляет собой миниатюрную тонкую кремниевую пластинку прямоугольной формы, на которой размещается огромное количество транзисторов, реализующих все функции, выполняемые процессором.

Вопрос 2

- Расскажите про принцип работы процессора?

Процессор работает, по сравнению с другими устройствами компьютера, с наибольшей скоростью. И самыми медленными по сравнению с ним являются внешние устройства, в том числе и человек. Так, например, работая с клавиатурой, человек отправляет в компьютер в среднем один байт в секунду (нажимает на одну клавишу в секунду) . Процессор обрабатывает такую и формацию за 0,000001 секунды.

Вопрос 3

- Расскажите подробнее про **Систему команд процессора.**

Процессор обрабатывает информацию, выполняя определенные команды. Таких команд может быть более тысячи. У каждой команды есть свой код (номер) . Например, есть команда 000, 001, 002 и т. д. Коды всех команд процессора записаны в двоичной форме в специальном документе, который называется системой команд процессора.

Вопрос 4

- Что значит уровень архитектуры команд?

- ***Уровень архитектуры команд*** связывает компиляторы и аппаратное обеспечение. Это язык, который понятен и компиляторам, и аппаратному обеспечению.

Вопрос 5

- Где расположен уровень архитектуры команд?

- Уровень архитектуры команд расположен между микроархитектурным уровнем и уровнем операционной системы. Исторически этот уровень развился прежде всех остальных уровней и изначально был единственным

Вопрос 6

- Расскажите про набор команд уровня.

Набор команд уровня операционной системы содержит большую часть команд из уровня архитектуры команд, а также несколько новых очень важных команд. Некоторые ненужные команды в уровень операционной системы не включаются. **Ввод-вывод - это одна из областей, в которых эти два уровня различаются очень сильно.**

Вопрос 7

- В чем состоит задача микроархитектурного уровня?

Над цифровым логическим уровнем находится микроархитектурный уровень. Его задача - интерпретация команд уровня 2 (уровня архитектуры команд. Строение микроархитектурного уровня зависит от того, каков уровень архитектуры команд, а также от стоимости и предназначения компьютера. В других системах (например, в системах Pentium II) на этом уровне имеются более сложные команды; выполнение одной такой команды занимает несколько циклов.

Вопрос 8

- Чем характеризуется Архитектура RISC-процессоров ?

Архитектура RISC-процессоров характеризуется наличием команд фиксированной длины, большого количества регистров, операций типа регистр-регистр, а также отсутствием косвенной адресации.

Вопрос 9

- Основная идея CISC-архитектуры

Основную идею CISC-архитектуры отражает ее название - «полный набор команд». В данной архитектуре стремятся иметь отдельную машинную команду для каждого возможного (типового) действия по обработке данных.

Исторически CISC-архитектура была одной из первых. Совершенствование процессоров шло по пути создания ВМ, способных выполнять как можно больше разных команд.

Вопрос 10

- Цель архитектуры **VLIW** (Very Long Instruction Word)

Цель анализа: обнаружить все команды, которые могут быть выполнены одновременно, причем так, чтобы между командами не возникали конфликты. В ходе анализа компилятор может даже частично имитировать выполнение рассматриваемой программы. На следующем этапе компилятор пытается объединить такие команды в пакеты (связки), каждый из которых рассматривается как одна сверхдлинная команда.

Вопрос 11

- Дайте определение архитектуре **VLIW** (Very Long Instruction Word)

VLIW (Very Long Instruction Word - очень длинная машинная команда) - архитектура процессоров, характеризующаяся возможностью объединения нескольких простых команд в так называемую связку.