02.03.2022

PCI express отличалась свой архитектурой: была изменена топология подключения внешних устройств

В данной архитектуре используются мосты и переключатели, что соответствует сетевой топологии и позволяет выполнять маршрутизацию передаваемой информации.

USB – универсальная последовательная шина для передачи последовательной информации.

Основные требования:

Пользователи не должны открывать компьютер, чтобы установить новое устройство

Должен существовать один вид кабеля, подсоединяющий все устройства

Устройства должны получать питание через кабель

Должна поддерживать устройства реального времени

Возможность подключения достаточно большого количества устройств

Шины USB оперируют стандартными кадрами 4-х типов:

Кадры управления для передачи команд конфигурации устройств

Изохронные для передачи информации в режиме реального времени

Передачи больших массивов данных

Кадры прерывания (нужны, так как шина не поддерживает прерывания)

Кадры состоят из нескольких стандартных пакетов, в которых основным пакетом являются данные, размер которых ограничен

Характеристики запоминающих устройств

Размещение

Внутренняя память – регистры, кэш, оперативная память

Внешняя память – различные виды дисковой памяти

Емкость – основная единица – байт, вводится понятие слова и их количества

Передаваемая порция – то, что передается при каждом обращении к внутренним ЗУ

Метод доступа

Последовательный, информация разделена на элементы – записи, чтобы считать информацию, необходимо пройти по всем предыдущим записям

Прямой – каждая запись имеет свой уникальный адрес, доступ осуществляется к определенной зоне носителя, в которой находится эта запись

Произвольный – определение местоположения осуществляется путем совмещения адресов в месте хранения информации и задаваемых процессором (кэш памяти и ОЗУ)

Ассоциативный – информация ищется не по адресу, а по содержимому

Производительность

Время доступа – длительность между интервалами, когда устройство передали адрес и когда данные зафиксированы в памяти или переданы в другое устройство

Длительность цикла обращения к памяти – временной интервал между последовательными сеансами доступа к памяти, это время включает время доступа и дополнительные операции, связанные с подготовкой устройств считывания

Скорость передачи – интенсивность информационного потока между ЗУ и другими устройствами

Физические типы

Полупроводниковый – внутренняя память

Магнитные – используется эффект перемагничивания магнитного материала, нанесенного на какую-либо поверхность

Оптические – изменение оптических свойств

Магнитно-оптические

Физические характеристики

Энергозависимые и энергонезависимые

Стираемые и нестираемые

Организация – архитектура, связана с конкретным видом ЗУ, по мере развития она менялась

Иерархия памяти

При разработке любой вычислительной системы относительно памяти возникает 3 основных вопроса:

Объем

Каково быстродействие

Стоимость

Эти параметры взаимосвязаны, для них характерны следующие отношения:

Чем выше быстродействие, тем выше относительная стоимость ЗУ в пересчете на 1 бит информации

Чем больше объем памяти в системе, тем ниже относительная стоимость ЗУ

Чем больше объем памяти, тем ниже быстродействие

По мере перехода от верхней иерархии к нижней наблюдается следующие изменения:

Снижается относительная стоимость хранения информации

Повышается емкость отдельного модуля

Увеличивается время доступа

Снижается частота обращения к памяти со стороны процессора

09.03.2022

Типы полупроводниковой памяти с произвольным доступом

Ядро ОЗУ – полупроводниковый конденсатор

Полупроводниковые конденсаторы способны хранить информацию определенный период времени, поэтому необходимо постоянно осуществлять процесс регенерации – периодическое считывание с последующей перезаписью.

Полупроводниковые транзисторы

Ядра ОЗУ структурированы в матрице, в которой выделяются строки и столбцы, которые чаще всего носят название страницы

Считывание информации происходит не посредством ядра, а посредством страницы

Разновидности DRAM

«Обычная» DRAM

FRM DRAM (Fast page mode) – поддержка сокращенных адресов

EDO-DRAM (Extended Data Out) – требуется меньше времени за счет того, что процесс чтение выполняется параллельно с перезарядкой внутренней цепи, и номер столбца может быть установлен еще до завершения считывания данных

BEDO (Burst EDO) – данные считываются пакетами, для каждого следующего пакета не нужно устанавливать адреса, они устанавливаются автоматически

SDRAM (Synchronous DRAM)

DDR SDRAM (Double data rate) – данные передаются как по фронту, так и по спаду тактового импульса

RDRAM (Rambus) – увеличение тактовой частоты за счет сокращения разрядности шины, одновременная передача номеров строки и столбца, увеличение кол-ва банков для усиления параллелизма

Формула памяти была введена, чтобы сравнить быстродействие разных видов ОЗУ применительно не ко времени, а к количеству тактов. Чтобы реализовать, используется частота системной шины.

Статическая RAM

Основу SRAM составляет триггер, в их основе – полевые транзисторы по технологии CMOS

Преимущество – не требуется регенерация, потребляется меньше энергии, триггеры работают на более высоких частотах, чем конденсаторы

Недостаток – проигрывает по быстродействию

Структурирована в матрице

Может быть многопортовой, данные могут считываться по разным линиям адреса и данных

Память бывает синхронной и асинхронной

Основные этапы ее развития соответствуют DRAM

16.03.2022

Устройство микросхемы памяти

Способы организации динамической памяти могут быть разные, но основу в любом случае составляет матрица запоминающих элементов. Как правило количество таких матриц определяет размер шины данных. Количество строк и столбцов в матрице определяет размер адресной шины.

Адресная шина может быть мультиплексной (по ней можно одновременное передавать и данные, и адреса), может быть разделена на отдельные адреса строк и столбцов либо последовательно передавать адреса строк и столбцов.

Буферы данных – запоминающий устройства, как правило триггеры, структурированные в регистры, для кратковременного хранения данных.

Кэширование памяти

Основные понятия

Кэш-память – промежуточный вид памяти между процессором и основной памятью. Не является адресуемой, не имеет собственного адресного пространства.

Данные в кэше хранятся определенный момент времени. При каждом обращении к основной памяти контроллер кэш-памяти по своему каталогу определяет, имеются ли эти данные в кэше. Если она там есть, то это называется кэш-попаданием. Если их там нет, то кэш-промах. Кэш-память хранит определенные блок данных, и эти данные сопоставлены с адресами основной памяти. Значения этих адресов хранятся в кэш-каталогов.

Обращение к основной памяти может начинаться одновременно с поиском данных в каталогах кэша, а в случае кэш-попадания обращение к основной памяти сокращается. Такая архитектура называется look aside.

Обращение к внешней памяти может начинаться и с только фиксации промаха. Такая архитектура называется look through.

Существуют разные виды уровней кэша: L1 – ближе всего к ядру процессора, L2, L3

Существуют 2 способа обращения к строкам кэша. Они называются политиками записи:

Сквозная запись write through – при таком виде записи происходит запись в кэш-память и одновременно в основную память. То есть затрачивает время на запись в основную память.

Обратная запись write back – позволяет уменьшать количество операций записи, если такой блок данных есть в кэше. В основную память такой блок не записывается сразу, а будет записан по мере освобождения шины. А в кэше такой блок помечается как модифицированный.

Кэш-память состоит из кэш-каталога и непосредственно кэш-памяти. Память разделена на строки. Размер строки соответствует размеру данных, считываемых из основной памяти. Каждая строка соотносится с определенным блоком данных основной памяти. Кэш-каталог хранит определенную информацию о адресах этих блоков – тэги – плюс дополнительные биты, которые указывают действительность строки и ее модифицируемость.

В зависимости от способа определения строки в кэше определяют 3 архитектуры кэш-памяти

Кэш прямого отображения

Полный адрес ячейки основной памяти определяется из 3 составляющих: 25-18 биты – тэг, 17-5 – индекс, 4-0 смещение в строке

Недостаток такого вида памяти – при обращении к одной и той же строке, расположенной на разных страницах, будут кэш-промахи

Разновидностью этой архитектуры является секторируемый кэш

Каждая строка – совокупность секторов, 4 строк

Наборно-ассоциативный кэш

Кэш-память разделена на банки

Размер каждого банка соответствует размеру памяти

Каждый банк имеет кэш-каталог

Дополнительный элемент – LRU – механизм, который определяет, в какой банк будут загружаться данные из основной памяти

Возможно использовать механизм FIFO или случайный выбор банка

Такая архитектура частично уменьшает недостаток кэша прямого отображения

Полностью ассоциативный кэш

Любая строка из основной памяти загружается в любую строку кэша. В этом случае кэш-каталог должен хранить полный адрес основной памяти.

23.03.2022

Кэширование процессоров старших поколений

Кэш первого уровня делится на 2 составляющие: кэш данных и кэш команд или инструкций

Вводятся дополнительные блоки, которые способны хранить часто используемую информацию

TLB – буфер ассоциативной трансляции, хранит информацию о вхождении в каталог и страницы таблиц, к которым обращались в последнее время

Буферы записи связаны с процессором и позволяют на некоторое время откладывать записи либо в кэш второго уровня, либо в основную память, при этом уступая шину для выполнения следующих команд

Блок выбора инструкций хранит часто используемые инструкции

Для проверки согласованности кэшей разного уровня в процессорах пятого поколения поддерживается протокол MESI

M – данная строка кэша была изменена, причем изменения не отражены в основной памяти и актуальны только для этого кэша

E – строка кэша содержит те же данные, что и блок основной памяти, причем они присутствуют только в данном кэше и отсутствуют во всех остальных

S – строка содержит те же данные, что и блок основной памяти, причем они присутствуют не только в данном кэше, но и в каких-то других

I – эта строка содержит недостоверные данные

Коррекция ошибок

В процессе работу полупроводниковых ЗУ неизбежны ошибки, их делят на 2 категории:

Неустранимые

Порождаются дефектом физического характера, элементы микросхем перестают менять свое состояние при записи

Корректируемые

Носят случайный характер, могут быть вызваны помехами по цепям питания, внешней радиацией или температурной нестабильностью работы микросхем

Для коррекции и обнаружения ошибок исходные данные M бит пропускаются через некую систему f, в которой по определенному алгоритму формируется код, дополняющий полученные данные K бит, который также заносится в память. На выходе из памяти основные данные M бит пропускаются через аналогичный алгоритм f, который должен сформировать код, аналогичный K. Эти два кода сравниваются:

Коды совпадают – отсутствие ошибки

Код не совпадают, но при этом ошибка может быть исправлена

Код не совпадают, но ошибка неисправима, формируется соответствующий сигнал

После сравнения компаратором входов формируется корректирующий код, с помощью которого идет исправление ошибки

Пример

Для 8 бит формируется 4 бита корректирующего код, который позволяет не только обнаружить, но и исправить ошибку. Для этих 4 бит должны выполняться следующие свойства:

Все биты кода равны 0, если ошибка не обнаружена

Если ошибка содержится в одном из четырех контрольных битов, то код признака ошибки содержит единицу только в одном бите и в этом случае нет необходимости корректировать основную информацию

Если ошибка содержится в информационном виде, то числовое значение кода признака ошибки должно указывать номер ошибочного бита, корректор должен инвертировать двоичный код этого бита и тем самым исправлять ошибку

Код признака ошибки вырабатывается на выходе компаратора

Допустим, храним M бит.

00111001

С1 = 1

С2 = 1

С4 = 1

С8 = 0

На выходе из тока памяти получили контрольный код 0001

Сравнение идет по модулю 2 и формируется код признака

Существует коды, которые позволяют исправлять одиночные ошибки и выявлять, но не исправлять, двойные (SED-DED)

Устройство ввода-вывода (УВВ) обеспечивают взаимодействие компьютерной системы с внешним миром. Существует 3 разных способа выполнения операции ВВ:

Программируемый – операция выполняется под контролем ПО

ВВ по прерыванию – программа только запускает процесс обмена, а затем переключается на выполнение других задач, пока ее не прервет подсистема ВВ и не известит о завершении процесса обмена

Прямой доступ к памяти – специализированный процессор модуля ВВ принимает на себя задачи по передачи информации между основной памятью и внешним устройством

В настоящий момент можно выделить следующие причины использования УВВ:

Существует большая номенклатура внешних устройств разного типа и назначений, которые выполняют разные операции, поэтому нецелесообразно нагружать процессор выполнением некоторых операций управления внешним устройством

Скорость обмена данными с внешним устройством значительно ниже пропускной способности процессора и основной памяти, поэтому нецелесообразно загружать системную шину медленным процессом взаимодействия с внешним устройством

Внешнее устройство может иметь формат представления информации, отличный от используемого в компьютерах

Модули ВВ должны выполнять

Взаимодействие с процессором и оперативной памятью посредством системной шины

Взаимодействие с одним или несколькими внешними устройствами через специализированные линии передачи данных

30.03.2022

Внешние устройства

Обмен информацией между вычислительной системой и внешним миром производится через внешний или периферийные устройства

Такие устр-ва можно разделить на 3 группы:

Предназначенные не непосредственной работы с человеком – дисплеи, принтеры устройства ввода

Предназначенные для работы с каким-либо оборудованием – внешняя память, различные датчики, исполнительные механизмы

Коммуникационные – предназначены для связи с удаленными абонентами – другие компьютеры, вычислительные системы

Взаимодействие с модулем ввода-вывода внешних устройств осуществляется с использованием сигналов 3 типов:

Управляющий – передаются от МВВ на внешние устройства, как правило такие сигналы связаны с чтением и записью

Информация о текущем состоянии – позволяет МВВ отслеживать состояние внешнего устройства, готовность – неготовность

Данные – циркулируют в двух направлениях и зависят от специфики внешнего устройства

Во внешнем устройстве должны быть следующие блоки:

Преобразователь, необходимый для преобразования информации из одной формы в другую

Буфер для временного хранения информации

Внутренний блок управления всем устройством – локальный контроллер – для приема и выработки управляющего сигнала

Функции МВВ

Управление и синхронизация – координация потоков данных между внешним устройством и внутренними ресурсами вычислительной системы, процесс передачи данных от внешних устройств в процессор может включать такую последовательность операций:

Процессор передает МВВ команду о проверке состояния подключенного внешнего устройства

МВВ возвращает процессору информацию о текущем состоянии

Если устройство находится в рабочем состоянии и готово передавать данные, то процессор дает команду МВВ начать передачу данных

МВВ считывает с внешнего устройства элементарную порцию данных

Данные передаются от МВВ в процессор

Связь с процессором – реализуются следующие операции

Расшифровка команд – МВВ определяет, какую команду он должен выполнить

Данные, которые передаются между процессором и МВВ посредством системной шины

Информация о текущем состоянии – готовность-занятость

Распознавание адреса – внешние устройства характеризуются уникальным адресом аналогично строке в памяти

Взаимодействие с внешним устройством – передача команд, обмен данными, передача информации о текущем состоянии

Временная буферизация данных

Обнаружение ошибок и сбоев – существует 2 типа:

Электромеханические, механические, электронные сбои внешних устройств

Случайные искажения кода

Структура МВВ

Регистры данных для временного хранения информации

Регистры состояния – хранится информация о текущем состоянии устройств

Регистры управления – хранятся управляющие команды

Логические схемы ввода-вывода для определения адресов внешних устройств

Логика управления внешними устройствами

Программирование ввода-вывода

Для выполнения ввода-вывода существует 4 типа команд

Команды управления – служит для активизации внешнего устройства и задает тип операции, которое это устройство должно выполнить

Команды опроса состояний – для анализа отдельных параметров текущего состояния модуля или подключенного устройства

Команды чтения – прием данных от внешних устройств

Команды записи – передача данных внешнему устройству

В вычислительных системах для обмена информации со внешними устройствами используются 2 типа адресации

Совмещенный режим – память и внешнее устройство используют единое адресное пространство, процессор вырабатывает управляющие сигналы и передает их по шинам управления независимо ото того, к чему обращается: к памяти или внешнему устройству

Изолированный режим – выделяются дополнительные линии для взаимодействия с МВВ

Ввод-вывод по прерыванию

Программирование ввода-вывода обладает следующими недостатками

Нерациональное использование времени процессора

Данный механизм дает возможность процессору переключиться на выполнение другой задачи или фрагмента программы во время выполнения команды ВВ

ВВ по прерыванию обеспечивает выполнение следующих действий

МВВ формирует сигнал запроса прерывания, которые передается процессору

Процессор завершает выполнение текущей команды

Процессор анализирует, поступил ли сигнал запроса прерывания, и, обнаружив его, посылает сигнал подтверждения МВВ

Процессор выполняет действие, предшествующее передачи управления подпрограмме обработки прерывания

Сохранят текущее состояние процессора

Сохраняет адрес следующей выполняемой команды

Формирует точку прерывания

Процессор загружает в счетчик команд начальный адрес обработки прерывания, начинается выполнение подпрограммы

Анализируется состояние МВВ, инициализируется обмен данными

Завершается выполнение этой подпрограммы и процессор восстанавливает информацию об основной программе

Прямой доступ к памяти (ПДП)

ПДП позволяет более эффективно использовать ресурсы процессора в том случае, когда перемещение данных идет от внешних устройств в оперативную память

Для организации такого механизма используется специальный элемент – контроллер ПДП

13.04.2022

Подход мультипроцессорных систем с шинной реализации отражает основные этапы развития вычислительных систем:

2 и более процессора используют общую память и взаимодействуют с ней по одной и той же шине. Недостатком является невозможность расширения данной системы

Каждому процессору добавлена кэш-память разного уровня. Это способствует уменьшению передачи информации по шине

Каждой процессор имеет свою локальную память. При это компилятор должен оптимально ее загружать, например, задавая процессору определенную программу со своим набором данных

Мультипроцессоры UMA с координатными коммутаторами

Представляет собой матрицу, посредством которой происходит соединение любого процессора с любым блоком памяти через коммутатор. Такой коммутатор представляет собой неблокируемую сеть, т.е. процессор всегда связан с нужным блоком памяти и другой процессор не может прервать эту связь. Недостаток: при увеличении количества процессоров, сеть увеличивается.

Мультипроцессор Sun Enterprise 10000 состоит из 64 процессоров с частотой работы 333 МГц и памятью 4 ГБ. Процессоры соединены с помощью координатного коммутатора 16х16 для обмена информацией. Кроме этого в системе имеются дополнительные адресные шины, которые позволяют отслеживать требуемую информацию в основной памяти и локальной памяти каждого модуля.

Процессоры с многоступенчатыми сетями

В основе лежит коммутатор 2 на 2.

Соединение входа и выхода коммутатора зависит от адреса ячейки памяти процессора. Например, процессор с номером 011 коммутируется с блоком памяти 100 через коммутаторы 1D, 2D, 3C

Недостаток: сеть является блокируемой. Это связано с тем, что на любом шаге всего 4 коммутатора.

Мультипроцессоры NUMA

Non-uniform memory access – процессоры с не унифицированным доступом к памяти

Так же, как и UMA, использует единое адресное пространство, но неоднородность проявляется в скорости обращения к каждому блоку памяти, т.е. в локальной памяти скорость будет больше, чем в общей, так как используется разная система команд и разные шины с разной пропускной способностью.

Выделяется 2 вида систем:

NC-NUMA – no cash NUMA, без кэширования

CC-NUMA – covering cash NUMA, согласованная кэш-память

Один из вариантов построения таких систем является мультипроцессорная система на основе каталогов, которая должна обеспечивать когерентность, согласованность данных. Каждый узел содержит процессор и локальную память. Узлы связаны через сеть. Каждый узел содержит каталог, который имеет основным назначением определение адреса хранения локальной памяти данных.

DASH – архитектура на основе каталога для памяти совместного использования

В данной структуре имеются матрицы 4х4, элементов матрицы является кластер, содержащий 4 стандартных процессора MIPS и локальную память. Каждый кластер содержит каталог для того, чтобы определить, какие кластеры используют блоки памяти в определенный момент, и информацию о состоянии этого блока

Состояния:

Блок памяти находится только в этой локальной памяти

Память содержит измененные данные, которые могут находиться в других кластерах

Данные неверны и находятся только лишь в одном кластере

NUMA-Q – содержит 4 узла, каждый содержит по 4 стандартных процессора и локальную память. Эти узлы соединены с помощью так называемого интерфейса SCI – масштабируемый когерентный интерфейс. Этот интерфейс связывает между собой дополнительные платы каждого узла.

Мультикомпьютерные системы

Основное отличие мультикомпьютера в том, что каждый узел этой систем имеет выход на УВВ. Мультипроцессорные системы ограничены в количестве процессоров (256) из-за того, что невозможно полноценно организовать обмен информации между отдельными узлами и памятью.

Выделяют:

MPP – процессор с массовым параллелизмом

COW – кластер рабочей станции

MPP система использует стандартные процессоры разных производителей (Intel, IBM, SUN и т.д.). В основе своей используют сеть, по которой передаются сообщения с низким временем ожидания и высокой пропускной способностью. Низкое время ожидания достигается тем, что эти сообщения небольшие по размеру. MPP используют большие внешние массивы данных, поэтому они имеют высокую производительность процесса ВВ. MPP также имеет высокую отказоустойчивость.

Примером MPP является машина Cray Research T3E. Это машина для научных исследований. Использует стандартные процессоры Alpha от компании DEC. Каждый узел сети содержит локальную память, процессор передачи данных и специальные регистры. Каждый узел соединен посредством топологии дуплексного трехмерного тора, например, это может быть куб 8х8х8. Каждый узел имеет 6 каналов связи с соседними узлами. Узлы соединены с подсистемой ВВ, обладающей высокой пропускной способностью. Это внешняя память, внешняя сеть, внешние устройства.

COW – другое название – NOW – сеть рабочих станций

Данный подход состоит в использовании стандартных компьютеров и соединений их между собой высокоскоростной сетью.

20.04.2022

Микропроцессор Intel 80386

32-разрядный микропроцессор (32-разрядная адресная шина)

1983 – год появления

Количество транзисторов – 275 тысяч

Технология – 1.5 мкм

Используется технология CHMOS (КМДП)

Используемые регистры:

Регистры общего назначения

Регистры системных флагов (32-разрядная структура полностью копирует версию аналогичного 16-разрядного регистра предыдущего поколения процессора, но введены дополнительные флаги: уровень привилегий ввода-вывода, флаг вложенной задачи, режим виртуального МП 8086)

Регистры сегмента – 6 16-разрядных регистров, которые содержат значения селекторов сегментов, указывающее на текущие адресуемые сегменты памяти: сегмент кода, сегмент стека и 4 сегмента данных

Регистры управления сегментированной памятью – выделяют 4 основных типа регистров: регистр таблицы глобальных дескрипторов – GDTR, регистр таблицы локальных дескрипторов – LDTR, регистр таблицы дескрипторов прерываний – IDTR, регистр задачи – TR (указывает на информацию, необходимую процессору для определения текущей задачи)

Указатель команд EIP – 32-разрядный, содержит относительный адрес следующей команды

Регистры управления Cr0-Cr3 – содержат информацию о текущем состоянии выполняемой задачи. Данный регистр содержит следующую информацию: PG – бит разбиения на страницы, ET – отвечает за наличие сопроцессора, TS – бит переключения задач, PE – бит включения защиты

Регистры отладки – их 6. Обозначаются DR. Содержат информацию о точках останова защитного процесса и позволяют возобновить процесс выполнения задачи

Режимы работы

Реальный – в этом режиме МП имеет такую же базовую архитектуру, что и МП 8086, но обеспечивает доступ к 32-разрядным регистрам. Механизм адресации, размеры памяти и обработка прерываний в данном режиме полностью совпадает с процессором 8086. Для формирования логического адреса в этом режиме используется 2 компоненты – это селектор для определения базового адреса сегмента и смещение внутри него. Рассмотрим сегментацию памяти в процессорах 8086: физический адрес получается в результате суммирования содержимого регистра, умноженного на 16, и смещения внутри него. Сегментация памяти давала возможность адресовываться к одному мегабайту памяти (220). Размер сегмента – 216

Защищенный – этот режим устанавливается битом PE и позволяет использовать все возможности 32-разрядной архитектуры.

Сегментная организация памяти – в процессоре 8086 адресные пространства – 1 МБ – и доступны 4 сегмента по 64 КБ. В 80386 адресные пространства – 232 байт – и допускается создание практически любого числа сегментов с размером, не превышающем 232 байт. Для такой реализации используется специальная структура – дескриптор – 64-битная структура, в которой выделяются следующие поля:

Базовый адрес – 32 бита. Определяет место сегмента внутри адресного пространства (начальный адрес).

Предел – 20 бит. Определяет размер сегмента, но с учетом бита гранулярности.

Бит гранулярности G. При G = 0 размер сегмента задается в байтах, при G = 1 в страницах по 4 килобайт. Таким образом, сегмент может варьироваться от 1 МБ до 4 ГБ

Бит пользователя U

Поле прав доступа. Бит DPL – уровень привилегий дескриптора. Бит P – бит присутствия (если равен 0, бит дескриптора не может использоваться)

Дескрипторы хранятся в памяти и группируются в дескрипторные таблицы. Выделяют 3 вида таблиц:

GDT – глобальная

LDT – локальная

IDT – таблица прерываний

GDT и IDT являются общесистемными и присутствуют в единственном экземпляре. LDT может создавать для каждой задачи. Максимальный размер таблицы – 213 байт. На эти таблицы ссылаются соответствующие регистры GDTR, LDTR, IDTR.

Изначально определяется адрес GDT по содержимому DGTR. Внутри этой таблицы выбирается дескриптор, по содержимому которого и по значению регистра LDTR определяется локальная таблица дескрипторов и определенный дескриптор внутри нее. По дескриптору LDT может быть получен линейный адрес.

Страничная организация памяти – в данной организации используется элемент памяти – страница, – которая имеет фиксированный размер (4 КБ) и фиксированное положение в линейном адресном пространстве. Граница страниц выравнены по 4 КБ-ным фрагментам. Такой механизм задается битом PG. Один из способов организации – разбиение на разделы, каждый раздел делится на страницы, и внутри страницы определяется внутренний физический адрес. Каждая страница имеет свои атрибуты, которые определяют следующую информацию:

Бит P – бит присутствия – присутствует ли страница в ОЗУ

Бит R/W – бит разрешения на чтение/запись

Бит U/S (user/supervisor) – определяет уровень привилегий

27.04.2022

Многозадачность – способность процессора выполнять несколько задач «одновременно»

Для поддержания многозадачности используются:

Сегмент состояния задач (TSS)

Дескриптор TSS

Регистр задач

Для реализации многозадачности необходимо:

Располагать быстродействующим процессором

Процессор должен аппаратно поддерживать механизм быстрого переключения задач

Процессор должен аппаратно поддерживать механизм защиты памяти

Сегмент состояния задач – структура данных, которая характеризует состояние виртуального процесса задач. Состоит из 2 частей:

Динамический набор – данные, которые обновляются процессором при каждом переключении задач: различные виды регистров

Статический набор, который микропроцессор считывает, но не обновляет: базовая карта ввода-вывода, содержимое некоторых регистров управления, определение стека для программы обработки прерываний

Сегмент определяется дескриптором, который расположен в глобальной таблице дескрипторов и соответствует по своей структуре дескриптору кода

Сегмент TSS имеет минимальный размер 104 байта, но возможно его увеличение для размещения дополнительной информации, связанной с информацией, полученной от сопроцессора

Переключение задач вызывается определенными событиями:

Определенный набор команд для перехода на другую задачу

С помощью определенных видов регистров флагов

Использование прерываний

При переключении задачи выполнятся следующие действия:

Процессор сохраняет контекст старой задачи в сегменте TSS

Процессор загружает в регистр задачи TR сегмент новой задачи

Процессор загружает в сегмент TSS новой задачи ссылку на такой же сегмент старой задачи

Получив доступ к TSS новой задачи, процессор загружает контекст новой задачи в соответствующий регистр

Защита памяти

В процессоре выделяют 2 защиты памяти:

На уровне сегментов

Предусмотрено 4 уровня привилегий для сегментов, которые задаются значением от 0 до 3, 0 – самый высокий

Приоритет или уровень защиты задается в разных структурах. Выделяют уровень привилегий дескриптора (DPL)

Текущий уровень привилегий (CPL) – контекст выполняемых задач

RPL – запрашиваемый уровень привилегий, связан с конкретным сегментом и определяет уровень на возможно выполняемую программу

В зависимости от значений этих содержимых этих 3 полей возможно переключение определенных задач

Существуют различные варианты защит доступа к данным или командам. При реализации этих защит происходит сравнение значений 3 полей: DPL, CPL, RPL

На уровне страниц

При реализации страничной памяти каждая страница имеет определенный набор данных, связанный с различными блоками защиты

Бит R/W – ограничивает возможности записи, т.е. страница возвожна только для чтения или еще и записи

Бит U/S – при 0 имеется высокий приоритет, соответствующий нулевому приоритету от 0 до 2 сегментной реализации памяти. При 1 это уровень пользователя

Процессоры 486

Появились в 1989

Технология – 1 микрон

Количество транзисторов 1.2 млн

Основные особенности:

В процессор введен внутренний кэш первого уровня и возможность подключения кэша второго уровня

Для повышения производительности использовалось так называемое RISС-ядро. Сам процессор представлял собой CISC-технологию

Введены буферы отложенной записи, которые позволяют задерживать запись в память до освобождения шины

В состав процессора введен высокопроизводительный математический сопроцессор

Введен новый набор команд

В связи с вводом кэш-памяти, используется пакетный режим передачи данных

Процессор пятого поколения

Появился 1993

Технология – 0.8 микрона

Количество процессоров – 3.1 млн

Частота доходит до 200 МГц

Основные особенности:

Супер-скалярная архитектура: процессор имеет 2 параллельно работающих конвейера обработки: U-конвейер с полным набором команд и V-конвейер с ограниченным набором команд. Стало возможным выполнение двух команд одновременно

Использовалась технология динамического предсказания переходов для обеспечений максимальной загрузки конвейера

Двухуровневый кэш, который поддерживает протокол MESI

Появилась возможность построения двухпроцессорной системы с избыточным контролем функциональности

Встроенный сопроцессор

В следующих этапах развития пятого поколения для реализации программ мультимедиа использовалась MMX-технология

MMX-технология

Использовалась для задач обработки 2D и 3D графики

Основная идея – обработка нескольких элементов данных за одну команду (SIMD)

Появились новые структуры данных – 64-битные целые. Для обработки таких структур нужна была MMX. В этой технологии были задействованы новые виды регистров (MMX0-7), которые способны управлять 64-разрядными, и набор команд следующего вида:

Арифметические

Сравнение

Преобразование форматов

Логические команды

Сдвиги

Пересылки данных между регистрами

MMX поддерживает арифметику с насыщением: при возникновении переполнения не происходит перенос в старший разряд, а фиксируется макс значение типа

Подобная технология появлялась и у AMD – 3DNOW!

04.05.2022

Двухпроцессорные системы

Процессоры пятого поколения – появилась возможность подключения второго процессора с помощью специальных интерфейсных сред, причем 2 процессора используют одноименные выводы и расположены параллельно

Первый способ построения – симметричная мультипроцессорная обработка (SMP). Данная архитектура обеспечивает возможность выполнения процессорами разных задач, но используя одни и те же ресурсы (память, внешние устройства). Такая архитектура должна обеспечивать согласованность данных оперативной памяти и различных уровней кэш-памяти.

Вторая архитектура – двухпроцессорная функционально избыточная (FRC). 2 процессора – функционально избыточные пары (мастер-помощник). Представляет собой 1 логический процессор – мастер, – который работает в обычном режиме. Второй процессор – проверочный, помощник – выполняет те же операции, но в хвосте. Его предназначение – проверка работы мастера.

Особенности процессоров шестого поколения Intel

Технология изготовления – 0.35 микрон

7.5 млн транзисторов

Частота работы – от 150 до 300 МГц

Особенности:

Динамическое исполнение – комбинация методов предсказания переходов, анализ прохождения данных и виртуальное выполнение. Команды, не зависящие от результатов предыдущих операций, могут выполняться в измененном порядке, но выгрузка результатов в память или внешнее устройство будет соответствовать исходному коду (выполнение инструкций не в том порядке – out of order)

Архитектура двойной независимой шины. Использование этой шины связано с динамическим исполнением, поскольку оно повышает частоту запросов процессора по шине к памяти. Поэтому была использована двойная шина, первая ее часть – связь процессора с кэш-памятью, а вторая – обычная системная шина.

Шина поддерживает ECC-контроллер (возможность обнаружения и исправления ошибок). Архитектура рассчитана на многопроцессорные системы по схеме SMP до 4 процессоров.

Параллелизм на уровне команд

Одним из способов повышения производительности выполнения команд – увеличение их скорости выполнения. Скорость напрямую зависит от технологий изготовления и связана с частотой работы процессора. Другой способ при данной скорости обработки – параллелизм, то есть возможность использовать несколько операций одновременно. Существует 2 формы параллелизма: параллелизм на уровне команд и параллелизм на уровне процессора.

Конвейеры

1959 – идея создания простого конвейера – предварительная загрузка команд из памяти и размещение их в специальном регистре – буфер выборки с упреждением. В дальнейшем процесс выполнения команды разбился на несколько этапов:

Извлечение команды

Декодирование

Извлечение операндов

Выполнение

Возврат, размещение результатов

Конвейер дает возможность увеличить быстродействие в пять раз.

Второй шаг – супер-скалярная архитектура. Имеет два и более конвейера. Такая архитектура позволяла использовать стандартные элементы на всех стадиях выполнения команды, кроме четвертой. Как правило, здесь были разные исполнительные блоки.

Следующий шаг – супер-скалярная архитектура с несколькими функциональными блоками. Такая архитектура дает возможность увеличить скорость обработки за счет выполнения определенных команд на своих исполнительных блоках.

Двухъядерные процессоры Intel

Производительность связана со скорость выполнения команд

IPC – количество команд за один такт

Частота связана с технологией изготовления

IPC связан с архитектурой: количество исполнительных блоков, наличие конвейера, его длина, эффективность заполнения конвейера (зависит от компилятора)

Для производительности IPC – использование многопроцессорных SMP систем или использование многоядерных систем

Технология Hyper-Threading

Появилась в 2002

Компания Intel

Многопоточная обработка команд

Первый шаг к многоядерности

Является нечто средним между многопоточной обработкой мультипроцессорных систем и параллелизме на уровне команд в однопроцессорных системах

Позволяет реализовать 2 логических процессора в одном физическом

Данная технология позволяет создавать 2 логических процессора, каждый из которых имеет свои регистры и свои контроллеры прерываний, то есть каждый процессор работает со своими собственными данными, но при этом использует одни и те же ресурсы процессора. Команды от двух параллельных потоков одновременно посылаются на ядро процессора, процессор, используя технология выполнения команд не в том порядке, способен параллельно обрабатывать два потока за счет нескольких исполнительных модулей. В идеальном случае производительность увеличивается в 2 раза, но реально возможно появление конфликтных ситуаций и уменьшение скорости. Поэтому второй шаг этой технологии – многоядерность.

Многоядерность

Позволяет избежать конфликтных ситуаций технологии Hyper-Threading

Производительность может быть увеличена в 2 раза, но в идеальных ситуациях, поскольку потоки, размещенные на каждом ядре, должны быть независимы друг от друга

Компьютеры с сокращенным и полным набором команд (RISC, CISC)

Компьютеры с RISC архитектурой имеет следующие особенности:

Выполнение одной команды за один машинный цикл. Машинный цикл – время, за которое процессор извлекает из регистров два операнда, выполняет над ними определенные действия и записывает результат в регистры. В таких процессорах для выполнения команды не требуется использовать микропрограммирвоание, а логика операций реализована в виде одной электронной схемы.

Большое количество операций типа регистр-регистр. В таких архитектурах минимизируют обращения к оперативной памяти, поэтому все данные должны быть загружены в регистр, туда же размещается результат. Это упрощает процесс вычисления.

Использование простых видов адресации. Под адресацией понимают способ определения адреса операнда. Используется регистровая адресация, а также адресация со смещением и адресация относительно текущего счетчика команд.

Простой формат команд. Длина команд фиксирована и равна длине машинного слова.

Основные преимущества RISC:

Повышение производительности:

Для таких архитектур разрабатывают более эффективные компиляторы с языком высокого уровня.

RISC процессоры просты, нет необходимости микропрограммирование.

Использование конвейеров.

RISC архитектуры быстрее реагируют на прерывания, поскольку они анализируются более простыми операциями.

Упрощение электронной схемы процессора:

Под каждую операцию может быть использована своя логическая структура. Нет необходимости в микропрограммировании.

CISC архитектура подразумевает использование полного набора команд для обработки информации.

Основные особенности:

Каждая команда разбивается на ряд простых машинных команд в RISC архитектуре, выполнение которых требует определенного времени. При достаточной частоте работы процессор с полным набором команд быстрее выполнит одну сложную, чем множество простых.

В таких архитектурах с полным набором команд формируется программный код меньшего объема, чем в RISC архитектурах, меньше объем памяти, более эффективно память используется

Архитектура IA-64

Была разработана Intel вместе с HP, существенным образом отличалась от стандартной 32-разрядной архитектуры.

Основные особенности:

Параллелизм на уровне команд, который связан не только с методом выполнения команд, но и с проектированием самого набора команд.

Применение длинных (LIW) и сверхдлинных (VLIW) форматов команд

Новый способ опережающего выполнения ветвящегося потока команд

Выборочная загрузка

Эти пункты относятся к EPIC – вычисления на основе явного параллелизма команд

Сравним традиционную архитектуру с IA-64:

|  |  |
| --- | --- |
| Традиционная | IA-64 |
| Используются RISC команды по одной в слове | Используются RISC команды, но сгруппированные по 3 в один пакет |
| Несколько параллельно функционирующих исполнительных блоков | |
| Изменение оптимизации потока команд происходит в ходе выполнения программы | Изменение оптимизации потока команд происходит на этапе компиляции программы |
| Прогнозирование ветвления и выборочное выполнение команд одной ветви | Выполнение команд с опережением в обеих ветвях после команды перехода |
| Данные загружаются из памяти только тогда, когда в них возникает необходимость, изначально они отыскиваются в кэше | Данные выборочно загружаются из памяти еще до того, как в них возникает необходимость, изначально они отыскиваются в кэше |

Структурная организация:

Большое количество регистров: 256 64-битных, из 128 регистры общего назначения и 128 для хранения данных с плавающей точкой

Большое количество исполнительных блоков, причем каждый исполнительный блок входит в состав конвейера. В конвейере используется до 4 параллельных блоков

Новый формат команд, который подразумевает включение трех команд и поле шаблонов. Одновременно из памяти извлекаются 3 команды, и формируется пакет.

Опережающее выполнение ветвящегося процессора

Данный процесс заключается в следующем:

Если встречается оператор if, то компилятор формирует два предиката, соответствующий двум ветвям: истина или ложь. Значения этих предикатов хранятся в соответствующих регистрах. После того, как известен результат операции if, результаты выполнения команд определенной ветви буду использованы

Предварительная загрузка данных

Если в операции ветвления имеется команда загрузки данных, то она может быть реализована раньше, чем выполнение других команд, и в случае, если вычислительный процесс идет по ветви, где эта команда находилась, то ее результат уже будет известен

Микроархитектура Net Burst

Технология – 0.13 микрон

Количество транзисторов – 42-125 млн

Частота – 1.25 – 3.4 ГГц

Увеличение быстродействия за счет увеличения параллелизма выполнения команд и, в частности, разбиения команд на более простые операции

В этой микроархитектуре выделяют следующие наборы блоков:

Группа исполнительных устройств и обслуживающие их элементы – back-end

Группа устройств, которая отвечает за декодирование команд и своевременную подачу первой группе – front-end

Группа устройств, отвечающая за загрузку и подачу данных на исполнительные блоки – подсистема памяти (memory subsystem)

Back-end содержит множество исполнительных устройств: 5 – вычислительное АЛУ, 2 – загрузка и извлечение. Исполнительный блок связан с обработкой графики и размещением результатов.

Основные особенности АЛУ в том, что они могут обрабатывать команды не в порядке их расположения в программе, а использовать технологию не очередного исполнения. Для этих целей используются дополнительные элементы – планировщики, – которые формируют вычислительный процесс.

Группа front-end – основное назначение – подготовить исполняемые команды. Помимо декодирования используется специальный модифицированный кэш команд. Каждая команда разбивается на ряд простых микроопераций (). Данные команды хранятся в отдельных кэшах первого уровня. Имеется быстродействующий кэш второго уровня.