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