1. Направления развития ЭВМ. История механических и электромеханических приборов для вычислений. Электронные приборы для вычислений

История вычислительных машин начинается с механических устройств, таких как счетные автоматы и арифмометры. В 19 веке Чарльз Бэббидж создал концепцию аналитической машины, которая включала принципы программирования. С 20 века, с изобретением электронных приборов, таких как ENIAC, началась новая эра в вычислениях, где используются электронные компоненты (лампы, транзисторы) для выполнения операций.

2. Поколения ЭВМ, ЭВМ Фон Неймана

ЭВМ делят на несколько поколений:

- **1-е поколение**: Вакуумные лампы (ENIAC).
- **2-е поколение**: Транзисторы – меньшие, надежнее.
- **3-е поколение**: Интегральные схемы.
- **4-е поколение**: Микропроцессоры.
- **5-е поколение**: Искусственный интеллект и параллельные вычисления.

ЭВМ Фон Неймана основывают свою архитектуру на принципе хранимой программы, где и данные, и программы хранятся в одной памяти.

3. Типы архитектур вычислительных систем

- **Классификация Флинна**:
  - SISD (однопроцессорная, одноканальная).
  - SIMD (однопроцессорная, многоканальная).
  - MISD (многопроцессорная, одноканальная).
  - MIMD (многопроцессорная, многоканальная).
- **Классификация Хокни** включает в себя такие типы, как унифицированные и специализированные архитектуры, в зависимости от способа обработки данных.

4. Система ввода-вывода

Система ввода-вывода (I/O) отвечает за взаимодействие компьютера с внешними устройствами. Она имеет адресное пространство, позволяющее обращаться к устройствам ввода-вывода, что позволяет использовать их совместно с основной памятью.

5. Шинная организация ЭВМ

Шина — это ключевая концепция в архитектуре ЭВМ, представляющая собой набор проводников, которые служат для передачи данных между различными компонентами системы, такими как процессор, память и периферийные устройства.

Основные типы шин:

Адресные шины:

- Предназначены для передачи адресов, по которым осуществляются операции чтения или записи.
- Ширина адресной шины определяет максимальный объем адресуемой памяти. Например, 32-битная шина может адресовать до 4 ГБ памяти.

Шины данных:

- Используются для передачи фактических данных между устройствами.
- Чем шире шина данных (например, 32 бита против 64 бит), тем больше данных может передаваться за один такт.

Управляющие шины:

- Передают управляющие сигналы, которые синхронизируют работу различных компонентов, указывая на операции чтения/записи и активируя нужные компоненты.

Примеры взаимодействия:

- При выполнении операции чтения, адрес передается через адресную шину, затем управляющие сигналы определяют, что данные должны быть считаны, и результаты перемещаются через шину данных в процессор.

6. Иерархия шин

Иерархия шин устанавливает уровни взаимодействия между компонентами системы, обеспечивая более оптимальное и эффективное использование ресурсов.

Уровни иерархии:

Уровень системной шины:

- Основная шина, соединяющая процессор, оперативную память и контроллеры.

Сегментированные и специализированные шины:

- Позволяют оптимизировать взаимодействие, уменьшив нагрузки на системную шину.
- Например, шины для видеокарт, жестких дисков и других периферийных устройств.

Арбитраж шин:

- механизмы, которые контролируют доступ к общей шине для предотвращения конфликтов между устройствами, пытающимися использовать шину одновременно.
- Часто используется метод временной multiplexing (разделение времени), где каждая шина имеет определенное время для передачи данных.

В результате такая иерархия обеспечивает более эффективное использование ресурсов и повышает общую производительность системы.

7. Характеристики памяти. Классификация запоминающих устройств

Память в ЭВМ играет критически важную роль в производительности и эффективности работы системы.

Классификация по характеристикам:

По скорости:

- **Кэш-память**: Быстрая, используемая для временного хранения данных, которые чаще всего запрашиваются процессором.
- **Оперативная память (RAM)**: Быстрая, но менее быстрая, чем кэш.

По доступности:

- **Постоянная память**: Хранит данные при отключении питания (например, ROM, флеш-память).
- **Временная память**: Используется для активной обработки данных (например, RAM).

По стоимости:

- **Дорогая память**: Например, кэш и оперативная память.
- **Дешевая память**: Жесткие диски и SSD.

Виды памяти:

- **Быстрая**: Кэш и RAM
- **Медленная**: Жесткие диски, SSD
- **Постоянная**: ROM, флеш-память
- Каждое устройство имеет свои преимущества и недостатки, и их выбор зависит от конкретных условий использования.

8. RAID-массивы

RAID (Redundant Array of Independent Disks) — метод вычисления и управления данными, который улучшает производительность и надежность хранения данных за счет объединения нескольких жестких дисков в массивы.

Основные уровни RAID:

RAID 0: Данные разбиваются на блоки и распределяются между всеми дисками. Повышает скорость, но не обеспечивает избыточности.
RAID 1: Создает точные копии данных на двух (или более) дисках. Управляет избыточностью, но снижает общую емкость.
RAID 5: Использует метод чередования с избыточностью распределенной по всем дискам, обеспечивая как скорость, так и надежность.
RAID 6: Похож на RAID 5, но с дополнительным слоем избыточности, что позволяет восстанавливать данные даже в случае выхода из строя двух дисков.
RAID 10 (или 1+0): Комбинация RAID 1 и RAID 0, обеспечивающая высокую производительность и надежность.

RAID позволяет значительно улучшить доступность и защиту данных, что особенно важно для серверов и рабочих станций.

9. Способы ввода-вывода информации

Процессы ввода-вывода (I/O) критически важны для работы системы. Они обеспечивают взаимодействие между процессором и внешними устройствами.

Основные методы:

Прямой доступ к памяти (DMA):

- Позволяет устройствам ввода-вывода получать доступ к памяти без участия процессора, что увеличивает скорость передачи данных.

Прерывания:

- Процессор может временно прервать выполнение текущей задачи для обработки события ввода-вывода. После обработки прерывания он возвращается к предыдущей задаче.

Программное управление:

- Все операции ввода-вывода выполняются непосредственно под контролем процессора. Это менее эффективно, чем DMA, но более простое в реализации.

Каждый из этих методов имеет свои преимущества и недостатки, и выбор зависит от задач, которые необходимо решить.

10. Машинная команда. Основные группы команд. Структура машинной команды

Машинная команда представляет собой минимальную единицу работы, которую может выполнять процессор.

Основные группы команд:

Арифметические команды:

- Выполняют математические операции, такие как сложение, вычитание, умножение и деление.

Логические команды:

- Выполняют логические операции, такие как AND, OR, NOT, которые применяются к битам.

Переносные команды:

- Занимаются перемещением данных, например, копированием значений между регистрами или в память.

Структура машинной команды:

- **Код операции (Opcode)**: Определяет, какую операцию нужно выполнить.
- **Операнды**: Это данные, над которыми будет производиться операция, или адреса, где эти данные находятся.

Точная структура и количество операндов могут варьироваться в зависимости от архитектуры процессора (например, RISC или CISC).

11. Показатели производительности, единицы измерения

Производительность вычислительных систем измеряется различными показателями, которые помогают оценить эффективность работы.

Основные единицы измерения:

MIPS (Million Instructions Per Second):

- Измеряет количество миллионов инструкций, выполняемых процессором за секунду.

FLOPS (Floating Point Operations Per Second):

- Измеряет количество операций с плавающей точкой, что особенно важно для научных и инженерных приложений.

Эти показатели помогают сравнивать производительность разных процессоров и систем, а также оптимизировать их использование на практике.

12. Процессор. Диаграмма операций в процессоре. Архитектура набора команд

Процессор — это сердцевина ЭВМ, который выполняет команды программ.

Основные стадии операции:

Извлечение: Команда загружается из памяти.
Декодирование: Команда интерпретируется, определяются операнды.
Исполнение: Выполнение операции над данными.
Запись результата: Запись результата операции в память или регистр.

Архитектура набора команд:

- **RISC (Reduced Instruction Set Computer)**: Имеет относительно небольшой набор простых команд, что позволяет увеличивать частоту тактов и упрощать аппаратное обеспечение.
- **CISC (Complex Instruction Set Computer)**: Имеет более сложный набор команд, что позволяет выполнять более трудоемкие операции за меньшее количество команд, но усложняет аппаратную реализацию.

Эти архитектуры влияют на производительность и эффективность выполнения программ, что является важным аспектом при выборе процессоров для различных применений.

13. Типы архитектур процессоров CISC и RISC, гибридные архитектуры.

- **CISC** (Complex Instruction Set Computer) имеет сложные команды.
- **RISC** (Reduced Instruction Set Computer) имеет упрощенный набор команд, что позволяет увеличивать скорость выполнения.

14. Способы адресации

Адресация включает в себя прямую, косвенную, индексную и другие способы, позволяющие процессору находить необходимые данные.

15. Устройство управления. Жесткое и микропрограммное управление. Микрооперации.

Устройство управления определяет порядок выполнения команд. Жесткое управление задается аппаратно, микропрограммное - с помощью программ.

16-20. Периферийные устройства

Эти устройства обеспечивают взаимодействие с ЭВМ:

- Ввод: клавиатура, мышь, сканер.
- Вывод: монитор, принтер, динамики.
- Процессор i8086 имеет набор входов и выходов для работы с памятью и внешними устройствами.

21. Схема демультипликации шин

Демультипликаторы играют важную роль в архитектуре компьютерных систем, позволяя эффективно управлять передачей данных и адресов между устройствами.

Основные функции и принципы:

Разделение линий шины:

- Демультипликаторы разбивают шины адреса и данных на несколько линий, что позволяет подключать несколько устройств к одной шине.
- Это увеличивает возможности системы, позволяя одновременно взаимодействовать с несколькими устройствами без конфликтов.

Функционирование:

- Когда устройство отправляет адрес, демультипликатор определяет, к какому устройству этот адрес относится, и направляет к нему данные.
- Это происходит на уровне логики, обеспечивая правильное взаимодействие по поверхностям (например, для нескольких микросхем).

Преимущества:

- Упрощение маршрутизации сигналов и оптимизация использования шины.
- Снижение временных задержек за счет уменьшения числа необходимых выборок адреса.

Демультипликаторы являются важной частью архитектуры шины, особенно в многопроцессорных системах.

22. Модель памяти. Организация памяти i8086

i8086 — это один из первых процессоров Intel, введенный в 1978 году, который положил начало архитектуре x86.

Основные характеристики модели памяти i8086:

Сегментация:

- Модель памяти i8086 использует сегментацию для управления памятью, разбивая её на сегменты (например, код, данные и стек).
- Это позволяет организовать доступ к памяти более эффективно и безопасно.

Сегментарные регистры:

- Процессор содержит сегментарные регистры (CS, DS, SS, ES) для указания начал сегментов кода, данных и стека.
- Каждый адрес содержит 16-битное значение сегмента и 16-битное смещение, что позволяет объединить их в один 20-битный адрес, обеспечивающий доступ к 1 МБ памяти.

Управление доступом:

- Сегментация также позволяет реализовать различные уровни доступа к памяти (например, чтение, запись), что важно для защиты данных и предотвращения их случайного изменения.

Эта модель важна для понимания особенностей работы процессоров и операционных систем на архитектуре x86.

23. Сегментация памяти

Сегментация — это метод организации памяти, который позволяет более гибко управлять ею и эффективно работать с защищенным режимом.

Основные аспекты сегментации:

Гибкость управления памятью:

- Позволяет динамически выделять и освобождать память, что дает возможность адаптироваться к различным потребностям программ и пользователей.
- Каждому сегменту можно задать различные права доступа.

Защищенный режим:

- В защищенном режиме операционная система может управлять доступом к каждому сегменту, что позволяет предотвратить случайные или вредоносные операции над памятью.
- Сегментация помогает избежать конфликтов между процессами в многозадачных операционных системах.

Таблицы дескрипторов:

- Сегментация используется совместно с таблицами дескрипторов для управления доступом к сегментам.
- Дескрипторы содержат информацию о базе, размере сегмента и правах доступа, что позволяет процессору быстро проверять запрашиваемые операции.

Сегментация упрощает архитектуру управления памятью и служит базой для многих современных операционных систем.

24. Организация прерываний на примере i8086

Прерывания — это механизм, позволяющий процессору временно приостанавливать выполнение текущей программы для обработки событий или запросов от внешних устройств.

Основные функции организации прерываний:

Обработка событий:

- Прерывания позволяют процессору реагировать на события, такие как ввод с клавиатуры или сигнал от устройства, без постоянной проверки состояния устройств (опрашивания).
- Это ускоряет выполнение программ и делает систему более отзывчивой.

Различие типов прерываний:

- **Аппаратные прерывания**: Генерируются внешними устройствами (например, клавиатурой, мышью).
- **Программные прерывания**: Генерируются сами программами, как результат выполнения определённых операций (например, деление на ноль).

Стек прерываний:

- Когда происходит прерывание, процессор сохраняет текущее состояние выполнения программы (адрес возврата) в стек, что позволяет после обработки прерывания вернуться к оригинальной программе.
- Это гарантия, что прерывания не нарушают текущий поток выполнения.

Вектор прерываний:

- В i8086 реализована таблица векторов прерываний, которая содержит адреса обработчиков для различных прерываний.
- Это позволяет быстро находить и вызывать соответствующий обработчик для каждого типа прерывания.

Организация прерываний является важной частью архитектуры процессора, обеспечивая взаимодействие между устройствами и программами, а также улучшая эффективность и отзывчивость системы.

25. Конвейерные вычисления: общая структура и показатели

Конвейерные вычисления — это техника, позволяющая улучшить производительность процессора за счет параллельного выполнения нескольких инструкций. Концепция аналогична конвейерному производству на заводе, где каждое звено выполняет свою часть работы.

Общая структура:

Этапы конвейера:

- **Извлечение (Fetch)**: Получение инструкции из памяти.
- **Декодирование (Decode)**: Интерпретация инструкции и определение операндов.
- **Исполнение (Execute)**: Выполнение арифметической или логической операции.
- **Завершение (Write-back)**: Запись результата в память или регистр.

Поток данных: Инструкции проходят через этапы конвейера последовательно, но на каждом этапе может обрабатываться несколько инструкций одновременно.

Показатели:

- **Пропускная способность**: Количество инструкций, обрабатываемых за единицу времени. Повышение этой метрики — основная цель конвейеризации.
- **Задержка**: Время, необходимое для выполнения одной инструкции. Увеличивается из-за билетов (так называемых "провалов" в конвейере).
- **Уровень параллелизма**: Определяет, сколько инструкций может выполняться одновременно.

Таким образом, конвейерные вычисления значительно увеличивают производительность системы, однако они также приносят определенные сложности, такие как зависимость данных и структурные конфликты.

26. Конвейер команд: конфликты

Конвейер команд — это уровень, на котором осуществляется выполнение инструкций в процессоре. Однако параллельное выполнение может привести к различного рода конфликтам.

Виды конфликтов:

Конфликты данных:

- Возникают, когда одна инструкция зависит от результата предыдущей.
- Например, если первая инструкция загружает данные в регистр, а вторая использует этот регистр, может случиться ошибка, если вторая команда будет выполнена до завершения первой.

Структурные конфликты:

- Происходят, когда несколько инструкций требуют доступа к одному ресурсу (например, к памяти или к аппаратному блоку).
- Это может привести к задержкам при ожидании доступа к этому ресурсу.

Контрольные конфликты:

- Возникают в случае изменения потока управления (например, при выполнении условных переходов).
- Если процессор не может заранее предсказать, какая инструкция должна быть выполнена следом, он может начать выполнять неправильные инструкции, что потребует дальнейших затрат времени на откат.

Чтобы минимизировать эти конфликты, применяются различные приемы, такие как:

- **Предварительное вычисление**.
- **Использование методов предсказания переходов**.
- **Реорганизация инструкций** (инструкция с меньшей зависимостью может быть помещена раньше в очередь).

Эти меры помогают эффективно управлять конвейером и снижать влияние конфликтов на общую производительность.

27. Суперконвейеризация и суперскалярные вычисления

Суперконвейеризация и **суперскалярные вычисления** — это более продвинутые техники, которые используются для еще большего увеличения производительности современных процессоров.

Суперконвейеризация:

- Придаёт конвейеру большее количество этапов, что позволяет уменьшать время каждой операции, но увеличивает количество операций, которые могут быть выполнены одновременно.
- Например, в двухстадийной суперконвейерной архитектуре могут быть задействованы отдельные элементы вычислений, которые более эффективно разграничивают стадии. Это позволяет быстрее обрабатывать инструкции, упрощая каждый этап.

Суперскалярные вычисления:

- На основе концепции суперконвейеризации, суперскалярные архитектуры позволяют процессорам одновременно выполнять несколько инструкций на одном такте.
- Процессор имеет несколько исполнительных единиц, которые позволяют ему одновременно выполнять несколько операций (например, два целых числа и одно вычисление с плавающей запятой одновременно).

Основные характеристики:

- **Многопоточность**: Процессоры могут обрабатывать несколько потоков инструкций, что значительно увеличивает эффективность.
- **Динамическое планирование**: Процессоры могут перераспределять инструкции в зависимости от доступности ресурсов и конфликта.