**Содержание**

**Архитектура RISC……………………………………………………………..…………….…2**

**История развития архитектуры RISC……………………………………………...………..3** **Архитектура RISC для чего было создано……………………………………………..……4**

**Плюсы и минусы RISC………………………………………………………………………...5**

**Последние годы RI…………….………………………………………………………….…….6**

**Выводы про архитектуру RISC…………………………………………………………….…8  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
¶Архитектура RISC** - (Reduced Instruction Set Computing) — это принцип построения процессора, в котором используется набор простых и однородных инструкций. Главной идеей архитектуры RISC является упрощение набора инструкций, что позволяет достичь более высокой производительности и упрощает процессорную реализацию.Основные принципы архитектуры RISC включают:1.Малый набор инструкций: В архитектуре RISC используется ограниченный набор инструкций, часто фиксированного размера и однородной структуры. Это позволяет упростить декодирование и исполнение инструкций, а также улучшить производительность за счет более простой аппаратной реализации.2.Простые инструкции: Инструкции архитектуры RISC обычно выполняют простые операции, такие как арифметические операции, операции с памятью и операции перехода. Сложные операции могут быть построены из комбинаций этих простых инструкций. Это упрощает реализацию и ускоряет выполнение инструкций.3.Загрузка/хранение в память: В архитектуре RISC операции чтения и записи данных в память выполняются только специальными инструкциями, называемыми "загрузка" (load) и "хранение" (store). Это отличается от других архитектурных подходов, где операции чтения и записи могут быть встроены в другие инструкции. Такой подход позволяет сократить число инструкций и упростить управление памятью.4.Использование регистрового файла: В архитектуре RISC используется набор регистров для хранения операндов и результатов операций. Операции выполняются непосредственно над регистрами, что позволяет достичь более высокой скорости выполнения инструкций и уменьшить нагрузку на память.5.Прямая адресация памяти: В архитектуре RISC адресация памяти происходит непосредственно через указание адреса в инструкции. Относительная адресация и другие сложные формы адресации обычно не используются. Это упрощает исполнение инструкций и увеличивает скорость доступа к памяти.Примеры архитектур RISC включают MIPS (Microprocessor without Interlocked Pipeline Stages), SPARC (Scalable Processor Architecture), ARM (Advanced RISC Machine) и RISC-V (RISC-Five). Эти архитектуры широко применяются в различных устройствах, включая компьютеры, мобильные устройства, микроконтроллеры и встроенные системы.

**История развития архитектуры RISC** - началась в 1970-х годах и продолжается до сегодняшнего дня. Вот основные этапы и вехи в развитии RISC:1.Открытие концепции RISC: В начале 1970-х годов профессор Джон Кокс (John Cocke) из IBM провел исследования и предложил идею упрощенной архитектуры, в которой использовался набор простых инструкций. Эта идея стала основой для разработки архитектуры RISC.2.Проект IBM 801: В 1980 году IBM начала разработку первого коммерческого процессора на основе архитектуры RISC — IBM 801. Процессор IBM 801 был успешно разработан и использовался в системах IBM, но не получил широкого распространения.3.Stanford MIPS: В 1981 году команда из университета Стэнфорда (Stanford) под руководством профессора Джона Хеннесси (John Hennessy) начала работу над проектом MIPS (Microprocessor without Interlocked Pipeline Stages). MIPS был одним из первых коммерчески успешных процессоров на базе архитектуры RISC. Архитектура MIPS получила широкое признание и использовалась во многих системах, включая компьютеры, рабочие станции и встроенные системы.4.Sun Microsystems SPARC: В 1984 году компания Sun Microsystems представила архитектуру SPARC (Scalable Processor Architecture). SPARC стал одним из наиболее распространенных процессорных семейств на основе архитектуры RISC. Он был широко использован в серверах и рабочих станциях Sun Microsystems, а также в других системах.5.Развитие ARM: В 1985 году была создана компания ARM (Advanced RISC Machines) в Великобритании, которая занималась разработкой и лицензированием процессоров на основе архитектуры RISC. Процессоры ARM получили широкое применение в мобильных устройствах, планшетах, смартфонах и других энергоэффективных системах.6.RISC-V: В 2010 году команда из Университета Беркли (University of California, Berkeley) начала разработку открытой архитектуры RISC-V. RISC-V стал проектом с открытым исходным кодом, что позволяет разработчикам свободно использовать, изменять и распространять эту архитектуру. RISC-V получил значительное внимание и стал популярным в академическом и промышленном сообществах.Сегодня архитектура RISC остается актуальной и используется во многих системах, включая вычислительные устройства, встраиваемые системы, мобильные устройства и другие. Она продолжает эволюционировать, улучшаясь и приспосабливаясь к современным требованиям высокой производительности и энергоэффективности.

**Архитектура RISC для чего было создан -** была создана с целью повышения производительности процессоров и упрощения их проектирования. Во время разработки ранних компьютерных систем, применявших архитектуру CISC (Complex Instruction Set Computing), были замечены некоторые ограничения и сложности:1.Сложность декодирования инструкций: В архитектуре CISC инструкции могут иметь различную длину и сложную структуру. Декодирование таких инструкций требовало сложных и медленных алгоритмов, что снижало производительность процессоров.2.Низкая скорость выполнения инструкций: Из-за сложности и разнообразия инструкций архитектуры CISC, процессорам требовалось больше тактовых циклов для выполнения каждой инструкции, что ограничивало скорость работы.3.Сложность управления памятью: Архитектура CISC требовала сложных и гибких механизмов адресации памяти, что увеличивало сложность реализации и замедляло выполнение инструкций.Архитектура RISC была разработана для преодоления этих ограничений. Она предлагает следующие преимущества:1.Простота инструкций: Архитектура RISC использует набор простых и однородных инструкций, обычно фиксированного размера и однотипной структуры. Это упрощает декодирование инструкций и позволяет процессору выполнять их быстрее.2.Высокая производительность: Благодаря простым инструкциям и однородной структуре, процессоры с архитектурой RISC способны выполнять инструкции с большей скоростью. Это достигается за счет сокращения числа тактовых циклов, необходимых для выполнения каждой инструкции.3.Упрощенное управление памятью: В архитектуре RISC используется принцип "загрузка/хранение" (load/store), где операции чтения и записи данных выполняются только специальными инструкциями. Это упрощает управление памятью и повышает производительность.4.Более простая аппаратная реализация: Упрощенные инструкции RISC позволяют создавать более простые и эффективные процессоры. Более простая аппаратура обычно требует меньше транзисторов, занимает меньше места на чипе и потребляет меньше энергии.Архитектура RISC была эффективной в повышении производительности процессоров и стала основой для многих успешных процессорных семейств, таких как MIPS, SPARC, ARM и RISC-V.

**Плюсы и минусы RISC -** Архитектура RISC (Reduced Instruction Set Computing) имеет ряд преимуществ и некоторые недостатки. Вот основные плюсы и минусы архитектуры RISC:

**Плюсы RISC:**

1.Высокая производительность: Упрощенные инструкции и однородная структура архитектуры RISC позволяют процессорам выполнять инструкции быстрее, сокращая число тактовых циклов. Это обеспечивает высокую производительность во многих приложениях.

2.Простота аппаратной реализации: Простые инструкции RISC требуют меньше сложной логики и меньшего количества транзисторов для их реализации. Это делает процессоры на базе RISC более компактными, более энергоэффективными и проще в проектировании.

3.Легкая оптимизация компиляторов: Упрощенные инструкции RISC облегчают работу компиляторов при оптимизации кода. Компиляторы могут более эффективно генерировать код для RISC-процессоров, что способствует повышению производительности программ.

4.Улучшенная предсказуемость выполнения: Благодаря однородной структуре и простым инструкциям, выполнение инструкций в архитектуре RISC более предсказуемо. Это позволяет более точно оценивать время выполнения программ и облегчает реализацию механизмов предсказания ветвлений и других оптимизаций.

**Минусы RISC:**

1.Больший объем кода: В некоторых случаях, из-за простоты инструкций RISC, для выполнения сложных операций может потребоваться больше инструкций по сравнению с архитектурой CISC (Complex Instruction Set Computing). Это может привести к увеличению объема кода и памяти, требуемых для выполнения программ.

2.Ограниченная функциональность инструкций: RISC инструкции обычно предназначены для базовых операций и не включают некоторые сложные операции, которые могут быть реализованы в архитектуре CISC. В некоторых приложениях это может потребовать дополнительных операций и реализаций.

3.Усложнение программистской работы: Из-за более простых инструкций и отсутствия некоторых сложных операций, программистам может потребоваться больше усилий для реализации некоторых функций и оптимизаций. Это может привести к увеличению сложности программирования и разработки.

4.Возможные ограничения в архитектуре: В некоторых реализациях архитектуры RISC могут отсутствовать некоторые расширения и функции, которые могут быть доступны в архитектуре CISC. Это может ограничить некоторые возможности и функциональность процессоров на базе RISC.

5.В целом, архитектура RISC имеет множество преимуществ, включая высокую производительность, простоту реализации и улучшенную предсказуемость выполнения. Однако она также имеет свои ограничения и недостатки, которые нужно учитывать при выборе и применении этой архитектуры.**Последние годы RISC**Как оказалось в начале 1990-х годов, RISC-архитектуры позволяют получить большую производительность, чем CISC, за счёт использования суперскалярного и VLIW-подхода, а также за счёт возможности серьёзного повышения тактовой частоты и упрощения кристалла с высвобождением площади под кэш, достигающий огромных ёмкостей. Также RISC-архитектуры позволили сильно снизить энергопотребление процессора за счёт уменьшения числа транзисторов.Первое время RISC-архитектуры с трудом принимались рынком из-за отсутствия программного обеспечения для них. Эта проблема была решена переносом UNIX-подобных операционных систем (SunOS) на RISC-архитектуры.В настоящее время многие архитектуры процессоров являются RISC-подобными, к примеру, ARM, DEC Alpha, SPARC, AVR, MIPS, POWER и PowerPC. Наиболее широко используемые в настольных компьютерах процессоры архитектуры x86 ранее являлись CISC-процессорами, однако новые процессоры, начиная с Intel Pentium Pro (1995 г.), являются CISC-процессорами с RISC-ядром[. Они непосредственно перед исполнением преобразуют CISC-инструкции x86-процессоров в более простой набор внутренних инструкций RISC.После того, как процессоры архитектуры x86 были переведены на суперскалярную RISC-архитектуру, можно сказать, что большинство существующих ныне процессоров основано на архитектуре RISC.

**Выводы про архитектуру RISC -** является эффективным подходом к проектированию процессоров и имеет ряд преимуществ. Выводы по архитектуре RISC включают:1.Высокая производительность: RISC-процессоры благодаря упрощенным инструкциям и однородной структуре демонстрируют высокую производительность и способность обрабатывать инструкции быстрее.2.Простота аппаратной реализации: Упрощенные инструкции RISC требуют меньше сложной логики и меньше транзисторов для их реализации, что делает процессоры на базе RISC более компактными, энергоэффективными и проще в проектировании.3.Легкая оптимизация компиляторов: Простота инструкций RISC облегчает работу компиляторов при оптимизации кода, что способствует повышению производительности программ.4.Улучшенная предсказуемость выполнения: Благодаря однородной структуре и простым инструкциям RISC, выполнение инструкций более предсказуемо, что позволяет более точно оценивать время выполнения программ и улучшает реализацию механизмов предсказания ветвлений и других оптимизаций.Однако у архитектуры RISC также есть некоторые ограничения:1.Больший объем кода: В некоторых случаях, из-за простоты инструкций RISC, для выполнения сложных операций может потребоваться больше инструкций, что может привести к увеличению объема кода и памяти.2.Ограниченная функциональность инструкций: RISC инструкции не включают некоторые сложные операции, доступные в архитектуре CISC, что может потребовать дополнительных операций и реализаций.3.Усложнение программистской работы: Из-за простых инструкций и отсутствия некоторых сложных операций, программистам может потребоваться больше усилий для реализации некоторых функций и оптимизаций.В целом, архитектура RISC представляет собой эффективный и широко применяемый подход к проектированию процессоров. Ее преимущества включают высокую производительность, простоту реализации и улучшенную предсказуемость выполнения. Однако при выборе архитектуры следует учитывать и ее ограничения, чтобы они соответствовали требованиям конкретных приложений и систем.