RISC

Материал из Википедии — свободной энциклопедии

RISC (англ. *Reduced Instruction Set Computer*; неправильно — *Reduced Instruction Set Computing*) — вычисления с сокращённым набором команд.

Это концепция проектирования процессоров (ЦПУ), которая во главу ставит следующий принцип: более компактные и простые инструкции выполняются быстрее. Простая архитектура позволяет удешевить процессор, поднять тактовую частоту, а также распараллелить исполнение команд между несколькими блоками исполнения (т. н. суперскалярные архитектуры процессоров). Многие ранние RISC-процессоры даже не имели команд умножения и деления. Идея создания RISC процессоров пришла после того, как в 1970-х годах ученые из IBM обнаружили, что многие из функциональных особенностей традиционных ЦПУ игнорировались программистами. Отчасти это был побочный эффект сложности компиляторов. В то время компиляторы могли использовать лишь часть из набора команд процессора. Следующее открытие заключалось в том, что, поскольку некоторые сложные операции использовались редко, они как правило были медленнее, чем те же действия, выполняемые набором простых команд. Это происходило из-за того, что создатели процессоров тратили гораздо меньше времени на улучшение сложных команд, чем на улучшение простых.

Первые RISC-процессоры были разработаны в начале 1980-х годов в Стэнфордском и Калифорнийском университетах США. Они выполняли небольшой (50−100) набор команд, тогда как обычные CISC (Complex Instruction Set Computer) выполняли 100—200.

Содержание

[убрать]

Характерные особенности RISC-процессоров

- Фиксированная длина машинных инструкций (например, 32 бита) и простой формат команды.
- Специализированные команды для операций с памятью — чтения или записи. Операции вида «прочитать-изменить-записать» отсутствуют. Любые операции «изменить» выполняются только над содержимым регистров (т. н. load-and-store архитектура).
- Большое количество регистров общего назначения (32 и более).

- Отсутствие поддержки операций вида «изменить» над укороченными типами данных — байт, 16-битное слово. Так, например, система команд DEC Alpha содержала только операции над 64-битными словами, и требовала разработки и последующего вызова процедур для выполнения операций над байтами, 16- и 32-битными словами.
- Отсутствие микропрограмм внутри самого процессора. То, что в CISC процессоре исполняется микропрограммами, в RISC процессоре исполняется как обыкновенный (хотя и помещенный в специальное хранилище) машинный код, не отличающийся принципиально от кода ядра ОС и приложений. Так, например, обработка отказов страниц в DEC Alpha и интерпретация таблиц страниц содержалась в так называемом PALCode (Privileged Architecture Library), помещенном в ПЗУ. Заменой PALCode можно было превратить процессор Alpha из 64-битного в 32-битный, а также изменить порядок байт в слове и формат входов таблиц страниц виртуальной памяти.

Архитектуры, обычно обсуждаемые в связи с RISC

- Суперскалярные архитектуры (первоначально Sun SPARC, начиная с Pentium использованы в семействе x86). Распараллеливание исполнения команд между несколькими устройствами исполнения, причем решение о параллельном исполнении двух или более команд принимается аппаратурой процессора на этапе исполнения. Эффективное использование такой архитектуры требует специальной оптимизации машинного кода в компиляторе для генерации пар независимых (результат одной не является входом другой) команд.
- Архитектуры VLIW (Very Long Instruction Word — Очень Длинное Слово Команды). Отличаются от суперскалярной архитектуры тем, что решение о распараллеливании принимается не аппаратурой на этапе исполнения, а компилятором на этапе генерации кода. Команды очень длинны, и содержат явные инструкции по распараллеливанию нескольких субкоманд на несколько устройств исполнения. Элементы архитектуры содержались в серии PA-RISC. VLIW-процессором в его классическом виде является Itanium, долгое время бывший самым мощным процессором в мире. Разработка эффективного компилятора для VLIW является сложнейшей задачей, решить которую не получалось долгое время. Преимущество VLIW перед суперскалярной архитектурой — компилятор является более сложной и «умной», чем устройства управления процессора, системой, способной хранить больше контекстной информации и принимать более верные решения об оптимизации.

Иные архитектурные решения, типичные для RISC

- Спекулятивное исполнение. При встрече с командой условного перехода процессор исполняет (или по крайней мере читает в кэш инструкций) сразу обе ветви, до тех пор, пока не окончится вычисление управляющего выражения перехода. Позволяет отказаться от простоев конвейера при условных переходах.
- Переименование регистров. Каждый регистр процессора на самом деле представляет собой несколько параллельных регистров, хранящих несколько версий значения. Используется для реализации спекулятивного исполнения.

Начало развития RISC-архитектуры

Первая система, которая может быть названа RISC-системой, - суперкомпьютер CDC 6600, который был создан в 1964 году, за десять лет до появления соответствующего термина.

CDC 6600 имел RISC-архитектуру всего с двумя режимами адресации (регистр+регистр и регистр+immediate_constant) и 74 кодами команд (тогда как Intel 8086 имел 400 кодов команд). В CDC 6600 было 11 конвейерных устройств арифметической и логической обработки, а также 5 load units и 2 store units. Память была многоблочной, поэтому все устройства загрузки-хранения могли работать одновременно. Базовая тактовая частота/частота выдачи команд была в 10 раз выше, чем время доступа к памяти. Джим Торнтон и Сеймор Крэй, разработчики CDC 6600, создали в нем мощный процессор, позволявший быстро обрабатывать большие объемы цифровых данных. Главный процессор поддерживался 10-ю простыми периферийными процессорами, выполнявшими операции ввода/вывода и другие функции ОС. [1] Позднее появилась шутка, что термин RISC на самом деле расшифровывается как "**R**eally **I**nvented by **S**eymour **C**ray" ("На самом деле придуман Сеймуром и Крэйем"). Еще одна ранняя RISC-машина - миникомпьютер Data General Nova , разработанный в 1968.

Первая попытка создать RISC-процессор на чипе была предпринята в IBM в 1975. Эта работа привела к созданию семейства процессоров IBM 801, которые широко использовались в различных устройствах IBM. 801-ый в конце концов был выпущен в форме чипа под именем ROMP в 1981 году. ROMP расшифровывается как Research OPD(Office Product Division) Micro Processor, т.е. Исследовательский Микро Процессор, разработанный в Департанменте офисных разработок. Как следует из названия, процессор был разработан для "мини"-задач, и когда в 1986 IBM выпустила на его базе компьютер IBM RT-PC, он работал не слишком хорошо. Однако, за выпуском 801-ого процессора последовало несколько исследовательских проектов, в результате одного из которых появилась система POWER.

Однако наиболее известные RISC-системы были разработаны в рамках университетских исследовательских програм, финансировавшихся программой DARPA VLSI. Программа VLSI, практически неизвестная даже сегодня, на самом деле поспособствовала большому количеству открытий в области архитектуры чипов, технологий производства и даже компьютерной графики.

Проект RISC в Университете Беркли был начат в 1980 году под руководством Дэвида Паттерсона и Карло Секвина. Исследования базировались на использовании конвейерной обработки и агрессивного использования техники регистрового окна. В обычном процессоре имеется небольшое количество регистров и программа может использовать любой регистр в любое время. В процессоре, использующем технологии регистрового окна, очень большое количество регистров (например, 128), но программы могут использовать ограниченное количество (например, только 8 в каждый момент времени).

Программа, ограниченная лишь 8-ю регистрами для каждой процедуры, может выполнять очень быстрые вызовы процедур: "окно" просто сдвигается к 8-ми регистровому блоку нужной процедуры, а при возврате из процедуры "окно" сдвигается обратно, к регистрам вызвавшей процедуры. (В обычном процессоре большинство процедур при вызове вынуждены сохранять значения некоторых регистров в стеке для того, чтобы пользоваться этими регистрами при исполнении процедуры. При возврате из процедуры значения регистров восстанавливаются из стека).

Проект RISC произвел на счет процессор RISC-I в 1982 году. В нем было всего 44420 транзисторов (для сравнения: в наиболее современных CISC-процессорах того времени было около 100000 транзисторов). RISC-I имел всего 32 инструкции, но превосходил по скорости работы любой одночиповый процессорв того времени. Через год, в 1983, был выпущен RISC-II, который состоял из 40760-ти транзисторов, использовал 39 инструкций и работал в 3 раза быстрее RICS-I.

Практически к то же время, в 1981 году, Джон Хеннесси начал аналогичный проект, названный "MIPS Архитектура" в Стэнфордском университете. MIPS практически полностью сфокусировался на конвейерной обработке, попытавшись "выжать все" из этой

технологии. Конвейерная обработка использовалась и в других продуктах, некоторые идеи, реализованные в MIPS, позволили разработанному чипу работать значительно быстрее аналогов. Наиболее важным было требование выполнения любой из инструкций процессора за один такт. Это требование позволило конвейеру работать на гораздо бо́льших скоростях передачи данных и привело к значительному ускорению работы процессора. С другой стороны, исполнение этого требования имело негативный попбочный эффект в виде удаления из набора инструкций таких полезных операций, как умножение или деление.

В первые годы попытки развития RISC-архитектуры были хорошо известны, однако оставались в рамках породивших их университетских исследовательских лабораторий. Многие в компьютерной индустрии считали, что преимущества RISC-процессоров не проявятся при использовании в реальных продуктах из-за низкой эффективности использования памяти в составных инструкциях. Однако с 1986 года исследовательские проекты RISC начали выпускать первые работающие продукты.

Последние годы

Как оказалось в начале 1990-х годов, RISC-архитектуры позволяют получить большую производительность, чем CISC, за счет использования суперскалярного и VLIW подхода, а также за счет возможности серьезного повышения тактовой частоты DEC Alpha и за счет упрощения кристалла с высвобождением площади под кеш-память,достигающую огромных размеров. Также RISC-архитектуры позволили сильно снизить энергопотребление процессора за счет уменьшения числа транзисторов (ARM).

Первое время RISC-архитектуры с трудом принимались рынком из-за отсутствия программного обеспечения для них. Эта проблема была быстро 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 Intel486DX, являются CISC-процессорами с RISC-ядром. Они непосредственно перед исполнением преобразуют CISC-инструкции x86-процессоров в более простой набор внутренних инструкций RISC.

С отказом компаний Apple и Sun от использования серии CISC-процессоров Motorola 68xxx (в пользу PowerPC у Apple и в пользу SPARC у Sun), приведшем к фактическому прекращению производства серии, а также с переводом внутренней архитектуры серии x86 на суперскалярную RISC-архитектуру, подавляющее большинство существующих процессоров используют архитектуру RISC. Позже Apple перешла на x86-архитектуру, внешне являющуюся CISC.

См. также

- Усовершенствованные RISC-вычисления
- URISC

Примечания

1. ↑ Grishman, Ralph. Assembly Language Programming for the Control Data 6000 Series. Algorithmics Press. 1974. pg 12

Ссылки

- RISC в DMOZ