

Особенности архитектуры MIPS компании MIPS Technology

Архитектура MIPS была одной из первых RISC-архитектур, получившей признание со стороны промышленности. Она была анонсирована в 1986 году. Первоначально это была полностью 32-битовая архитектура, которая включала 32 регистра общего назначения, 16 регистров плавающей точки и специальную пару регистров для хранения результатов выполнения операций целочисленного умножения и деления. Размер команд составлял 32 бит, в ней поддерживался всего один метод адресации, и пользовательское адресное пространство также определялось 32 битами. Выполнение арифметических операций регламентировалось стандартом IEEE 754. В компьютерной промышленности широкую популярность приобрели 32-битовые процессоры R2000 и R3000, которые в течение достаточно длительного времени служили основой для построения рабочих станций и серверов компаний Silicon Graphics, Digital, Siemens Nixdorf и др. Процессоры R3000/R3010 работали на тактовой частоте 33 или 40 МГц и обеспечивали производительность на уровне 20 SPECint92 и 23 SPECfp92.

Затем на смену микропроцессорам семейства R3000 пришли новые 64-битовые микропроцессоры R4000 и R4400. (MIPS Technology была первой компанией выпустившей процессоры с 64-битовой архитектурой). Набор команд этих процессоров (спецификация MIPS II) был расширен командами загрузки и записи 64-разрядных чисел с плавающей точкой, командами вычисления квадратного корня с одинарной и двойной точностью, командами условных прерываний, а также атомарными операциями, необходимыми для поддержки мультипроцессорных конфигураций. В процессорах R4000 и R4400 реализованы 64-битовые шины данных и 64-битовые регистры. В этих процессорах применяется метод удвоения внутренней тактовой частоты.

Процессоры R2000 и R3000 имели стандартные пятиступенчатые конвейеры команд. В процессорах R4000 и R4400 применяются более длинные конвейеры (иногда их называют суперконвейерами). Количество ступеней в процессорах R4000 и R4400 увеличилось до восьми, что объясняется прежде всего увеличением тактовой частоты и необходимостью распределения логики для обеспечения заданной пропускной способности конвейера. Процессор R4000 может работать с тактовой частотой 50/100 МГц и обеспечивает уровень производительности в 58 SPECint92 и 61 SPECfp92. Процессор R4400 может работать на частоте 50/100 МГц, или 75/150 МГц, показывая уровень производительности 94 SPECint92 и 105 SPECfp92.

Внутренняя кэш-память процессора R4000 имеет емкость 16 Кбайт. Она разделена на 8-Кб кэш команд и 8-Кб кэш данных. С точки зрения реализации кэш-памяти процессор R4400 имеет более развитые возможности. Он выпускается в трех модификациях: PC (Primary Cache) - имеет внутренние кэши команд и данных емкостью по 16 Кбайт. Процессор в такой конфигурации предназначен главным образом для дешевых моделей рабочих станций. SC (Secondary Cache) содержит логику управления кэш-памятью второго уровня. MC (Multiprocessor Cache) - использует специальные алгоритмы обеспечения когерентности и согласованного состояния памяти для мультипроцессорных конфигураций.

В середине 1994 года компания MIPS анонсировала процессор R8000, который прежде всего был ориентирован на научные прикладные задачи с интенсивным использованием операций с плавающей точкой. Этот процессор построен на двух кристаллах (выпускается в виде многокристальной сборки) и представляет собой первую суперскалярную реализацию архитектуры MIPS. Теоретическая пиковая производительность процессора для тактовой частоты 75 МГц составляет 300 MFLOPs (до четырех команд и шести операций с плавающей точкой в каждом такте). Реализация большой кэш-памяти данных емкостью 16 Мбайт,

высокой пропускной способности доступа к данным (до 1.2 Гбайт/с) в сочетании с высокой скоростью выполнения операций позволяет R8000 достигать 75% теоретической производительности даже при решении больших задач типа LINPACK с размерами матриц 1000x1000 элементов. Аппаратные средства поддержки когерентного состояния кэш-памяти вместе со средствами распараллеливания компиляторов обеспечивают возможность построения высокопроизводительных симметричных многопроцессорных систем. Например, процессоры R8000 используются в системе Power Challenge компании Silicon Graphics, которая вполне может сравниться по производительности с известными суперкомпьютерами Cray Y-MP, имеет на порядок меньшую стоимость и предъявляет значительно меньшие требования к подсистемам питания и охлаждения. В однопроцессорном исполнении эта система обеспечивает производительность на уровне 310 SPECfp92 и 265 MFLOPs на пакете LINPACK (1000x1000).

В 1994 году MIPS Technology объявила также о создании своего нового суперскалярного процессора R10000, начало массовых поставок которого ожидалось в конце 1995 года. По заявлениям представителей MIPS Technology R10000 обеспечивает пиковую производительность в 800 MIPS при работе с внутренней тактовой частотой 200 МГц за счет обеспечения выдачи для выполнения четырех команд в каждом такте синхронизации. При этом он обеспечивает обмен данными с кэш-памятью второго уровня со скоростью 3.2 Гбайт/с.

Чтобы обеспечить столь высокий уровень производительности в процессоре R10000 реализованы многие последние достижения в области технологии и архитектуры процессоров. На рисунке 1 показана блок-схема этого микропроцессора.

Иерархия памяти

При разработке процессора R10000 большое внимание было уделено эффективной реализации иерархии памяти. В нем обеспечиваются раннее обнаружение промахов кэш-памяти и параллельная перезагрузка строк с выполнением другой полезной работой. Реализованные на кристалле кэши поддерживают одновременную выборку команд, выполнение команд загрузки и записи данных в память, а также операций перезагрузки строк кэш-памяти. Заполнение строк кэш-памяти выполняется по принципу "запрошенное слово первым", что позволяет существенно сократить простои процессора из-за ожидания требуемой информации. Все кэши имеют двухканальную множественно-ассоциативную организацию с алгоритмом замещения LRU.

Кэш-память данных первого уровня

Кэш-память данных первого уровня процессора R10000 имеет емкость 32 Кбайт и организована в виде двух одинаковых банков емкостью по 16 Кбайт, что обеспечивает двукратное расслоение при выполнении обращений к этой кэш-памяти. Каждый банк представляет собой двухканальную множественно-ассоциативную кэш-память с размером строки (блока) в 32 байта. Кэш данных индексируется с помощью виртуального адреса и хранит теги физических адресов памяти. Такой метод индексации позволяет выбрать подмножество кэш-памяти в том же такте, в котором формируется виртуальный адрес. Однако для того, чтобы поддерживать когерентность с кэш-памятью второго уровня, в кэше первого уровня хранятся теги физических адресов памяти.

Массивы данных и тегов в каждом банке являются независимыми. Эти четыре массива работают под общим управлением очереди формирования адресов памяти и схем внешнего интерфейса кристалла. В очереди адресов могут одновременно находиться до 16 команд

загрузки и записи, которые обрабатываются в четырех отдельных конвейерах. Команды из этой очереди динамически выдаются для выполнения в специальный конвейер, который обеспечивает вычисление исполнительного виртуального адреса и преобразование этого адреса в физический. Три других параллельно работающих конвейера могут одновременно выполнять проверку тегов, осуществлять пересылку данных для команд загрузки и завершать выполнение команд записи в память. Хотя команды выполняются в строгом порядке их расположения в памяти, вычисление адресов и пересылка данных для команд загрузки могут происходить неупорядоченно. Схемы внешнего интерфейса кристалла могут выполнять заполнение или обратное копирование строк кэш-памяти, либо операции просмотра тегов. Такая параллельная работа большинства устройств процессора позволяет R10000 эффективно выполнять реальные многопроцессорные приложения.

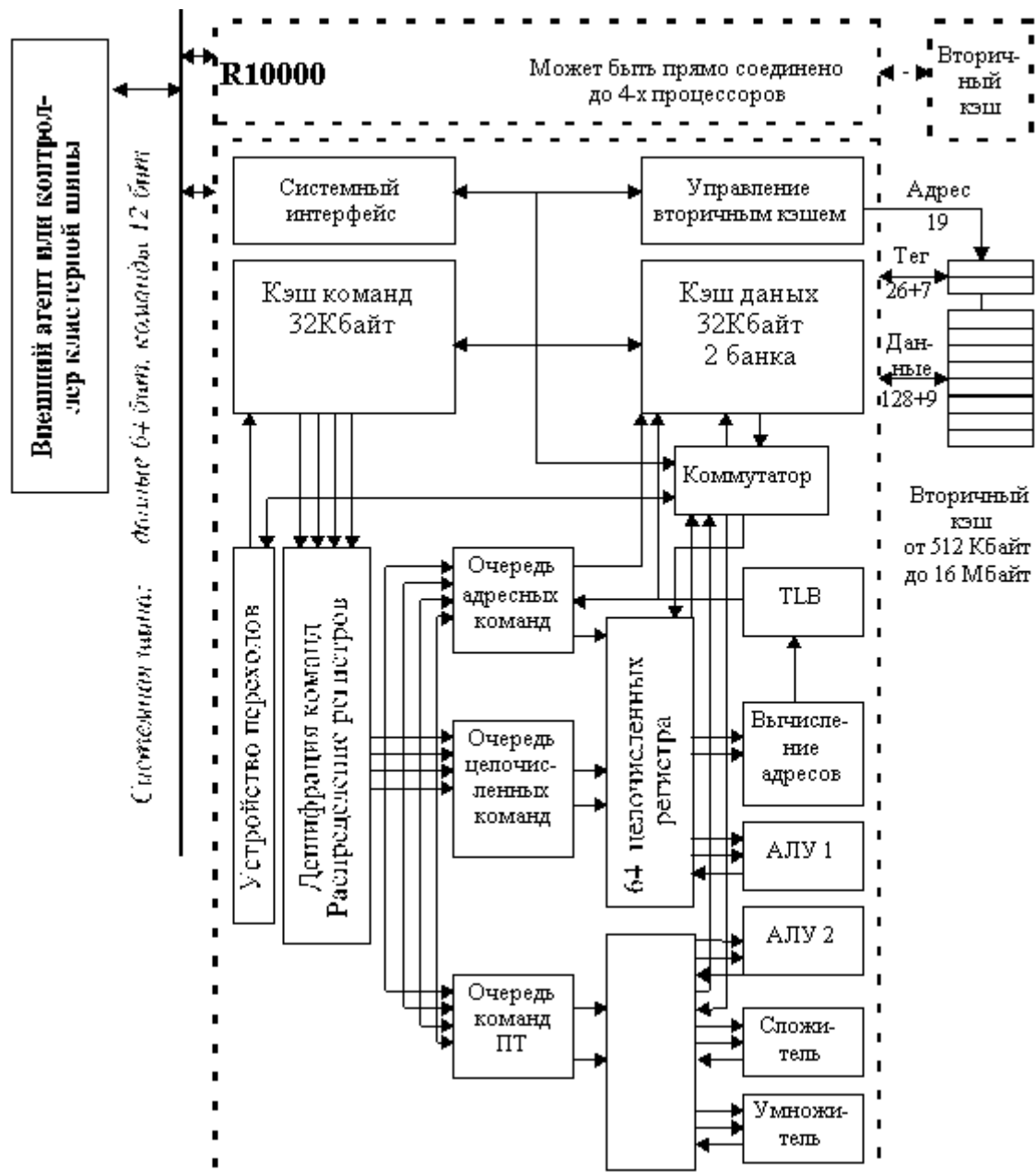


Рис. 1. Блок-схема микропроцессора R10000

Работа конвейеров кэш-памяти данных тесно координирована. Например, команды загрузки могут выполнять проверку тегов и чтение данных в том же такте, что и преобразование адреса. Команды записи сразу же начинают проверку тегов, чтобы в случае необходимости

как можно раньше инициировать заполнение требуемой строки из кэш-памяти второго уровня, но непосредственная запись данных в кэш задерживается до тех пор, пока сама команда записи не станет самой старой командой в общей очереди выполняемых команд и ей будет позволено зафиксировать свой результат ("выпустится"). Промах при обращении к кэш-памяти данных первого уровня иницирует процесс заполнения строки из кэш-памяти второго уровня. При выполнении команд загрузки одновременно с заполнением строки кэш-памяти данные могут поступать по цепям обхода в регистровый файл.

При обнаружении промаха при обращении к кэш-памяти данных ее работа не блокируется, т.е. она может продолжать обслуживание следующих запросов. Это особенно полезно для уменьшения такого важного показателя качества реализованной архитектуры как среднее число тактов на команду (CPI - clock cycles per instruction). На рисунке 2 представлены результаты моделирования работы R10000 на нескольких программах тестового пакета SPEC. Для каждого теста даны два результата: с блокировкой кэш-памяти данных при обнаружении промаха (вверху) и действительное значение CPI R10000 (внизу). Выделенная более темным цветом правая область соответствует времени, потерянному из-за промахов кэш-памяти. Верхний результат отражает полную задержку в случае, если бы все операции по перезагрузке кэш-памяти выполнялись строго последовательно. Таким образом, стрелка представляет потери времени, которые возникают в блокируемом кэше. Эффект применения неблокируемой кэш-памяти сильно зависит характеристик самих программ. Для небольших тестов, рабочие наборы которых полностью помещаются в кэш-памяти первого уровня, этот эффект не велик. Однако для более реальных программ, подобных тесту tomcatv или тяжелому для кэш-памяти тесту compress, выигрыш оказывается существенным.

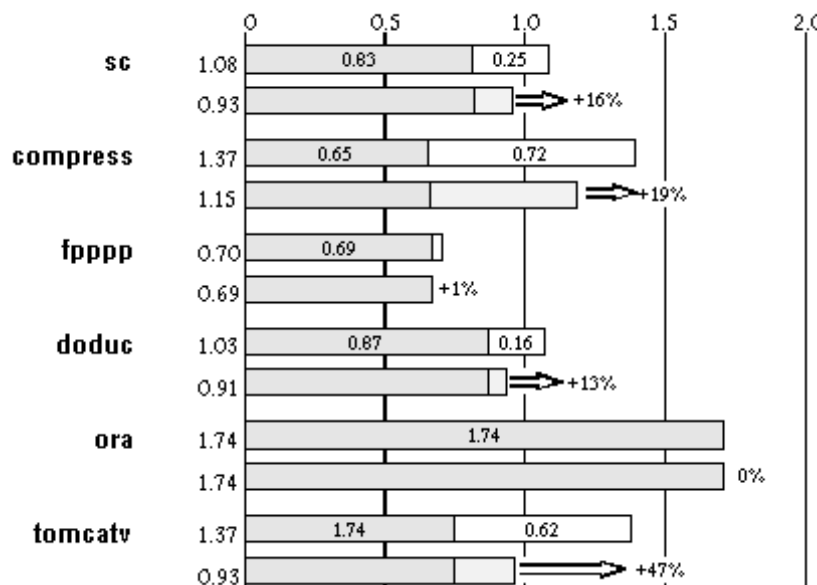


Рис. 2. Моделирование работы R10000 на нескольких компонентах пакета SPEC

Кэш-память второго уровня

Интерфейс кэш-памяти второго уровня процессора R10000 поддерживает 128-битовую магистраль данных, которая может работать с тактовой частотой до 200 МГц, обеспечивая скорость обмена до 3.2 Гбайт/с (для снижения требований к быстродействию микросхем памяти предусмотрена также возможность деления частоты с коэффициентами 1.5, 2, 2.5 и 3). Все стандартные синхронные сигналы управления статической памятью вырабатываются внутри процессора. Не требуется никаких внешних интерфейсных схем. Минимальный

объем кэш-памяти второго уровня составляет 512 Кбайт, максимальный размер - 16 Мбайт. Размер строки этой кэш-памяти программируется и может составлять 64 или 128 байт.

Одним из методов улучшения временных показателей работы кэш-памяти является построение псевдо-множественно-ассоциативной кэш-памяти. В такой кэш-памяти частота промахов находится на уровне частоты промахов множественно-ассоциативной памяти, а время выборки при попадании соответствует кэш-памяти с прямым отображением. Кэш-память R10000 организована именно таким способом, причем для ее реализации используются стандартные синхронные микросхемы памяти (SRAM). В одном наборе микросхем памяти находятся оба канала кэша. Информация о частоте использования этих каналов хранится в схемах управления кэшем на процессорном кристалле. Поэтому после обнаружения промаха в первичном кэше из наиболее часто используемого канала вторичного кэша считываются две четырехсловные строки. Их теги считываются вместе с первой четырехсловной строкой, а теги альтернативного канала читаются одновременно со второй четырехсловной строкой (это осуществляется простым инвертированием старшего разряда адреса).

При этом возможны три случая. Если происходит попадание по первому каналу, то данные доступны немедленно. Если происходит попадание по альтернативному каналу, происходит повторное чтение вторичного кэша. Если отсутствует попадание по обоим каналам, вторичный кэш должен перезаполняться из основной памяти.

Для обеспечения целостности данных в кэш-памяти большой емкости обычной практикой является использование кодов исправляющих одиночные ошибки (ECC-кодов). В R10000 с каждой четырехсловной строкой хранится 9-битовый ECC-код и бит четности.

Дополнительный бит четности позволяет сократить задержку, поскольку проверка на четность может быть выполнена очень быстро, чтобы предотвратить использование некорректных данных. При этом, если обнаруживается корректируемая ошибка, то чтение повторяется через специальный двухтактный конвейер коррекции ошибок.

Кэш-память команд

Объем внутренней двухканальной множественно-ассоциативной кэш-памяти команд составляет 32 Кбайт. В процессе ее загрузки команды частично декодируются. При этом к каждой команде добавляются 4 дополнительных бита, которые указывают исполнительное устройство, в котором она будет выполняться. Таким образом, в кэш-памяти команды хранятся в 36-битовом формате. Размер строки кэш-памяти команд составляет 64 байта.

Обработка команд перехода

При реализации конвейерной обработки возникают ситуации, которые препятствуют выполнению очередной команды из потока команд в предназначенном для нее такте. Такие ситуации называются конфликтами. Конфликты снижают реальную производительность конвейера, которая могла бы быть достигнута в идеальном случае. Одним из типов конфликтов, с которыми приходится иметь дело разработчикам высокопроизводительных процессоров, являются конфликты по управлению, которые возникают при конвейеризации команд перехода и других команд, изменяющих значение счетчика команд.

Конфликты по управлению могут вызывать даже большие потери производительности суперскалярного процессора, чем конфликты по данным. По статистике среди команд управления, меняющих значение счетчика команд, преобладают команды условного перехода. Таким образом, снижение потерь от условных переходов становится критически

важным вопросом. Имеется несколько методов сокращения приостановок конвейера, возникающих из-за задержек выполнения условных переходов. В процессоре R10000 используются два наиболее мощных метода динамической оптимизации выполнения условных переходов: аппаратное прогнозирование направления условных переходов и "выполнение по предположению" (speculation).

Устройство переходов процессора R10000 может декодировать и выполнять только по одной команде перехода в каждом такте. Поскольку за каждой командой перехода следует слот задержки, максимально могут быть одновременно выбраны две команды перехода, но только одна более ранняя команда перехода может декодироваться в данный момент времени. Во время декодирования команд к каждой команде добавляется бит признака перехода. Эти биты используются для пометки команд перехода в конвейере выборки команд.

Направление условного перехода прогнозируется с помощью специальной памяти (branch history table) емкостью 512 строк, которая хранит историю выполнения переходов в прошлом. Обращение к этой таблице осуществляется с помощью адреса команды во время ее выборки. Двухбитовый код прогноза в этой памяти обновляется каждый раз, когда принято окончательное решение о направлении перехода. Моделирование показало, что точность двухбитовой схемы прогнозирования для тестового пакета программ SPEC составляет 87%.

Все команды, выбранные вслед за командой условного перехода, считаются выполняемыми по предположению (условно). Это означает, что в момент их выборки заранее не известно, будет ли завершено их выполнение. Процессор допускает предварительную обработку и прогнозирование направления четырех команд условного перехода, которые могут разрешаться в произвольном порядке. При этом для каждой выполняемой по предположению команды условного перехода в специальный стек переходов записывается информация, необходимая для восстановления состояния процессора в случае, если направление перехода было предсказано неверно. Стек переходов имеет глубину в 4 элемента и позволяет в случае необходимости быстро и эффективно (за один такт) восстановить конвейер.

Структура очередей команд

Процессор R10000 содержит три очереди (буфера) команд (очередь целочисленных команд, очередь команд плавающей точки и адресную очередь). Эти три очереди осуществляют динамическую выдачу команд в соответствующие исполнительные устройства. С каждой командой в очереди хранится тег команды, который перемещается вместе с командой по ступеням конвейера. Каждая очередь осуществляет динамическое планирование потока команд и может определить моменты времени, когда становятся доступными операнды, необходимые для выполнения каждой команды. Кроме того, очередь определяет порядок выполнения команд на основе анализа состояния соответствующих исполнительных устройств. Как только ресурс оказывается свободным очередь выдает команду в соответствующее исполнительное устройство.

Очередь целочисленных команд

Очередь целочисленных команд содержит 16 строк и выдает команды в два арифметико-логических устройства. Целочисленные команды поступают в свободные строки этой очереди, причем в каждом такте в нее могут записываться до 4 команд. Целочисленные команды остаются в очереди до тех пор, пока они не будут выданы в одно из АЛУ.

Очередь команд плавающей точки

Очередь команд плавающей точки также содержит 16 строк и выдает команды в исполнительные устройства сложения и умножения с плавающей точкой. Команды плавающей точки поступают в свободные строки очереди, причем в каждом такте в нее могут записываться до 4 команд. Команды остаются в очереди до тех пор, пока они не будут выданы в одно из исполнительных устройств. Очередь команд плавающей точки содержит также логику управления команд типа "умножить-сложить". Эта команда сначала направляется в устройство умножения, а затем прямо в устройство сложения.

Адресная очередь

Очередь адресных команд выдает команды в устройство загрузки/записи и содержит 16 строк. Очередь организована в виде циклического буфера FIFO (first-in first-out). Команды могут выдаваться в произвольном порядке, но должны записываться в очередь и изыматься из нее строго последовательно. В каждом такте в очередь могут поступать до 4 команд. Буфер FIFO поддерживает первоначальную последовательность команд, что упрощает обнаружение зависимостей по адресам. Выполнение выданной команды может не закончиться при обнаружении зависимости по адресам, кэш-промаха или конфликта по ресурсам. В этих случаях адресная очередь должна заново повторять выдачу команды до тех пор, пока ее выполнение не завершится.

Переименование регистров

Одним из аппаратных методов минимизации конфликтов по данным является метод переименования регистров (register renaming). Он получил свое название от широко применяющегося в компиляторах метода переименования - метода размещения данных, способствующего сокращению числа зависимостей и тем самым увеличению производительности при отображении необходимых исходной программе объектов (например, переменных) на аппаратные ресурсы (например, ячейки памяти и регистры).

При аппаратной реализации метода переименования регистров выделяются логические регистры, обращение к которым выполняется с помощью соответствующих полей команды, и физические регистры, которые размещаются в аппаратном регистровом файле процессора. Номера логических регистров динамически отображаются на номера физических регистров посредством таблиц отображения, которые обновляются после декодирования каждой команды. Каждый новый результат записывается в новый физический регистр. Однако предыдущее значение каждого логического регистра сохраняется и может быть восстановлено в случае, если выполнение команды должно быть прервано из-за возникновения исключительной ситуации или неправильного предсказания направления условного перехода.

В процессе выполнения программы генерируется множество временных регистровых результатов. Эти временные значения записываются в регистровые файлы вместе с постоянными значениями. Временное значение становится новым постоянным значением, когда завершается выполнение команды (фиксируется ее результат). В свою очередь, завершение выполнения команды происходит когда все предыдущие команды успешно завершились в заданном программой порядке. Программист (или компилятор) имеет дело только с логическими регистрами. Реализация физических регистров от него скрыта.

Таким образом, аппаратный метод переименования регистров, используемый в процессоре R10000, имеет три основных достоинства. Во-первых, результаты "выполняемых по предположению" команд могут прямо записываться в регистровый файл. Во-вторых, этот метод устраняет все конфликты типа "запись после чтения" и "запись после записи", которые

часто возникают при неупорядоченном выполнении команд. И, наконец, метод переименования регистров упрощает контроль зависимостей по данным. Поскольку процессор обеспечивает выдачу для выполнения до четырех команд в каждом такте, в процессе переименования регистров их логические номера сравниваются для определения зависимостей между четырьмя командами, декодированными в одном и том же такте.

Реализованная в микропроцессоре R10000 схема отображения команд состоит из двух таблиц отображения, списка активных команд и двух списков свободных регистров (для целочисленных команд и команд плавающей точки имеются отдельные таблицы отображения и списки свободных регистров). Чтобы поддерживать последовательный порядок завершения выполнения команд, существует только один список активных команд, который содержит как целочисленные команды, так и команды плавающей точки.

Микропроцессор R10000 содержит по 64 физических регистра (целочисленных и плавающей точки). В любой момент времени значение физического регистра содержится в одном из указанных выше списков. На рисунке 3 показана упрощенная блок-схема отображения целочисленных команд.

Команды выбираются из кэша команд и помещаются в таблицу отображения. В любой момент времени каждый из 64 номеров физических регистров находится в одном из трех указанных на рисунке блоков. Список активных команд длиной 32 элемента может хранить упорядоченную в соответствии с программой последовательность команд, которые могут находиться в обработке в любой данный момент времени. Команды из очереди целочисленных команд могут выполняться неупорядоченно и записывать результаты в физические регистры, но порядок их окончательного завершения определяется списком активных команд.

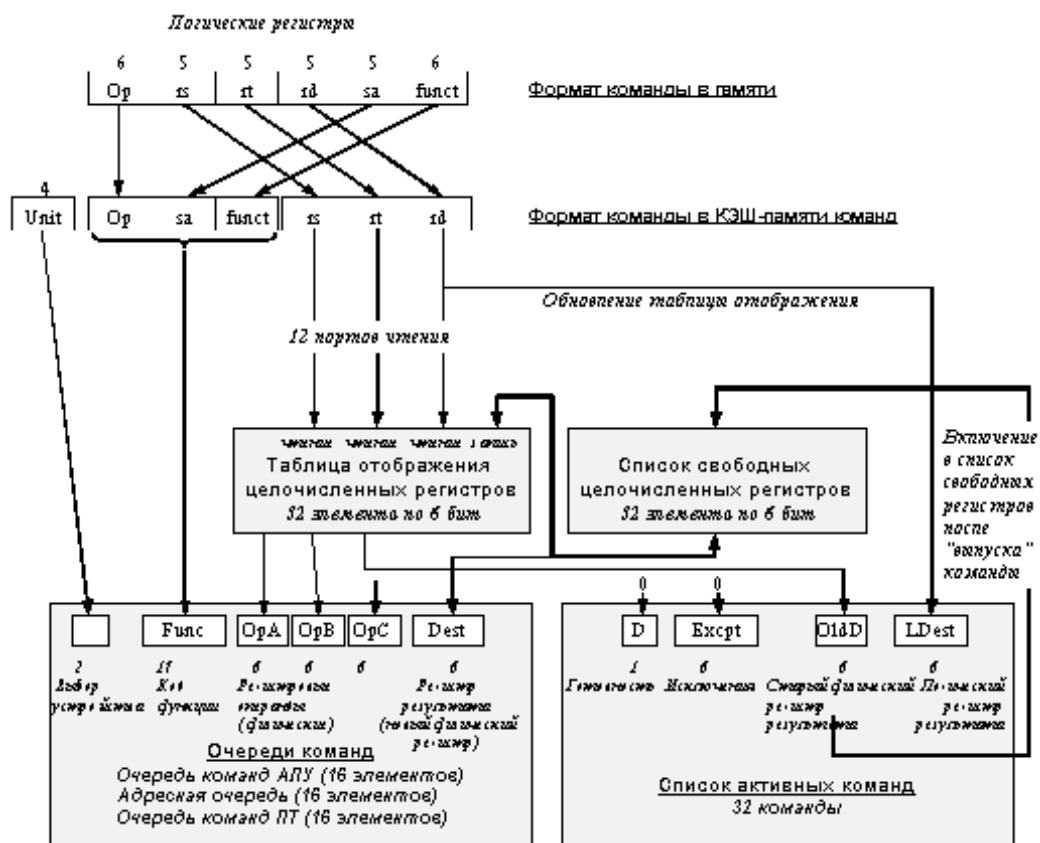


Рис. 3. Упрощенная блок-схема отображения целочисленных команд

Каждая команда может уникально идентифицироваться своим положением в списке активных команд. Поэтому каждую команду в очереди и в соответствующем исполнительном устройстве сопровождает 5-битовая метка, называемая тегом команды. Этот тег и определяет положение команды в списке активных команд. Когда в исполнительном устройстве заканчивается выполнение команды, тег позволяет очень просто ее отыскать в списке активных команд и пометить как выполненную. Когда результат операции из исполнительного устройства записывается в физический регистр, номер этого физического регистра становится больше не нужным и может быть затем возвращен в список свободных регистров, а соответствующая команда перестает быть активной.

Когда в процессе переименования из списка свободных регистров выбирается очередной номер физического регистра, он передается в таблицу отображения, которая обновляется. При этом старый номер регистра, соответствующий определенному в команде логическому регистру результата, помещается из таблицы отображения в список активных команд. Этот номер остается в списке активных команд до тех пор, пока соответствующая команда не "выпустится" (graduate), т.е. завершится в заданном программой порядке. Команда может "выпуститься" только после того, как успешно завершится выполнение всех предыдущих команд.

Микропроцессор R10000 содержит 64 физических и 32 логических целочисленных регистра. Список активных команд может содержать максимально 32 элемента. Список свободных регистров также может максимально содержать 32 значения. Если список активных команд полон, то могут быть 32 "зафиксированных" и 32 временных значения. Отсюда потребность в 64 регистрах.

Исполнительные устройства

В процессоре R10000 имеются пять полностью независимых исполнительных устройств: два целочисленных АЛУ, два основных устройства плавающей точки с двумя вторичными устройствами плавающей точки, которые работают с длинными операциями деления и вычисления квадратного корня, а также устройство загрузки/записи.

Целочисленные АЛУ

В микропроцессоре R10000 имеются два целочисленных АЛУ: АЛУ1 и АЛУ2. Время выполнения всех целочисленных операций АЛУ (за исключением операций умножения и деления) и частота повторений составляют один такт.

Оба АЛУ выполняют стандартные операции сложения, вычитания и логические операции. Эти операции завершаются за один такт. АЛУ1 обрабатывает все команды перехода, а также операции сдвига, а АЛУ2 - все операции умножения и деления с использованием итерационных алгоритмов. Целочисленные операции умножения и деления помещают свои результаты в регистры EntryHi и EntryLo.

Во время выполнения операций умножения в АЛУ2 могут выполняться другие одноктактные команды, но сам множитель оказывается занятым. Однако когда множитель заканчивает свою работу, АЛУ2 оказывается занятым на два такта, чтобы обеспечить запись результата в два регистра. Во время выполнения операций деления, которые имеют очень большую задержку, АЛУ2 занято на все время выполнения операции.

Целочисленные операции умножения вырабатывают произведение с двойной точностью. Для операций с одинарной точностью происходит распространение знака результата до 64

бит прежде, чем он будет помещен в регистры EntryHi и EntryLo. Время выполнения операций с двойной точностью примерно в два раза превосходит время выполнения операций с одинарной точностью.

Устройства плавающей точки

В микропроцессоре R10000 реализованы два основных устройства плавающей точки. Устройство сложения обрабатывает операции сложения, а устройство умножения - операции умножения. Кроме того, существуют два вторичных устройства плавающей точки, которые обрабатывают длинные операции деления и вычисления квадратного корня.

Время выполнения команд сложения, вычитания и преобразования типов равно двум тактам, а скорость их поступления в устройство составляет 1 команда/такт. Эти команды обрабатываются в устройстве сложения. Команды преобразования целочисленных значений в значения с плавающей точкой с однократной точностью имеют задержку в 4 такта, поскольку они должны пройти через устройство сложения дважды.

В устройстве умножения обрабатываются все операции умножения с плавающей точкой. Время их выполнения составляет два такта, а скорость поступления - 1 команда/такт. Устройства деления и вычисления квадратного корня выполняют операции с использованием итерационных алгоритмов. Эти устройства не конвейеризованы и не могут начать выполнение следующей операции до тех пор, пока не завершилось выполнение текущей команды. Таким образом, скорость повторения этих операций примерно равна задержке их выполнения. Порты умножителя являются общими и для устройств деления и вычисления квадратного корня. В начале и в конце операции теряется по одному такту (для выборки операндов и для записи результата).

Операция с плавающей точкой "умножить-сложить", которая в вычислительных программах возникает достаточно часто, выполняется с использованием двух отдельных операций: операции умножения и операции сложения. Команда "умножить-сложить" (MADD) имеет задержку 4 такта и скорость повторения 1 команда/такт. Эта составная команда увеличивает производительность за счет устранения выборки и декодирования дополнительной команды.

Устройства деления и вычисления квадратного корня используют отдельные цепи и могут работать одновременно. Однако очередь команд плавающей точки не может выдать для выполнения обе команды в одном и том же такте.

Устройство загрузки/записи и TLB

Устройство загрузки/записи содержит очередь адресов, устройство вычисления адреса, устройство преобразования виртуальных адресов в физические (TLB), стек адресов, буфер записи и кэш-память данных первого уровня. Устройство загрузки/записи выполняет команды загрузки, записи, предварительной выборки, а также команды работы с кэш-памятью.

Выполнение всех команд загрузки и записи начинается с трехтактной последовательности, во время которой осуществляется выдача команды, вычисление виртуального адреса и его преобразование в физический. Преобразование адреса осуществляется во время выполнения команды только однажды. Производится обращение к кэш-памяти данных, и пересылка требуемых данных завершается при наличии данных в кэш-памяти первого уровня.

В случае промаха, или в случае занятости разделяемого порта регистрового файла, обращение к кэшу данных и к тегу должно быть повторено после получения данных либо из кэш-памяти второго уровня, либо из основной памяти.

TLB содержит 64 строки и выполняет преобразование виртуального адреса в физический. Виртуальный адрес для преобразования поступает либо из устройства вычисления адреса, либо из счетчика команд.

Интерфейс кэш-памяти второго уровня

Внешняя кэш-память второго уровня управляется с помощью внутреннего контроллера, который имеет специальный порт для подсоединения кэш-памяти. Специальная магистраль данных шириной в 128 бит осуществляет пересылки данных на внутренней тактовой частоте процессора 200 МГц, обеспечивая максимальную скорость передачи данных кэш-памяти второго уровня 3.2 Гбайт/с. В процессоре имеется также 64-битовая шина данных системного интерфейса.

Кэш-память второго уровня имеет двухканальную множественно-ассоциативную организацию. Максимальный размер этой кэш-памяти - 16 Мбайт. Минимальный размер - 512 Кбайт. Пересылки осуществляются 128-битовыми порциями (4 32-битовых слова). Для пересылки больших блоков данных используются последовательные циклы шины:

- Четырехсловные обращения (128 бит) используются для команд кэш-памяти (CACHE);
- Восьмисловные обращения (256 бит) используются для перезагрузки первичного кэша данных;
- Шестнадцатисловные обращения (512 бит) используются для перезагрузки первичного кэша команд;
- Тридцатидвухсловные обращения (1024 бит) используются для перезагрузки кэш-памяти второго уровня.

Системный интерфейс

Системный интерфейс процессора R10000 работает в качестве шлюза между самим процессором, связанным с ним кэшем второго уровня и остальной системой. Системный интерфейс работает с тактовой частотой внешней синхронизации (SysClk). Возможно программирование работы системного интерфейса на тактовой частоте 200, 133, 100, 80, 67, 57 и 50 МГц. Все выходы и входы системного интерфейса синхронизируются нарастающим фронтом сигнала SysClk, позволяя ему работать на максимально возможной тактовой частоте.

В большинстве микропроцессорных систем в каждый момент времени может происходить только одна системная транзакция.

Процессор R10000 поддерживает протокол расщепления транзакций, позволяющий осуществлять выдачу очередных запросов процессором или внешним абонентом шины, не дожидаясь ответа на предыдущий запрос. Максимально в любой момент времени поддерживается до четырех одновременных транзакций на шине.

Поддержка многопроцессорной организации

Процессор R10000 допускает два способа организации многопроцессорной системы. Один из способов связан с созданием специального внешнего интерфейса (агента) для каждого процессора системы. Этот интерфейс обычно реализуется с помощью заказной интегральной схемы, которая организует шлюз к основной памяти и подсистеме ввода/вывода. При таком типе соединений процессоры не связаны друг с другом непосредственно, а взаимодействуют через этот специальный интерфейс. Хотя такая реализация общепринята, ее стоимость, а также общая сложность системы достаточно высоки. (поскольку по крайней мере один внешний агент должен сопровождать каждый процессор.

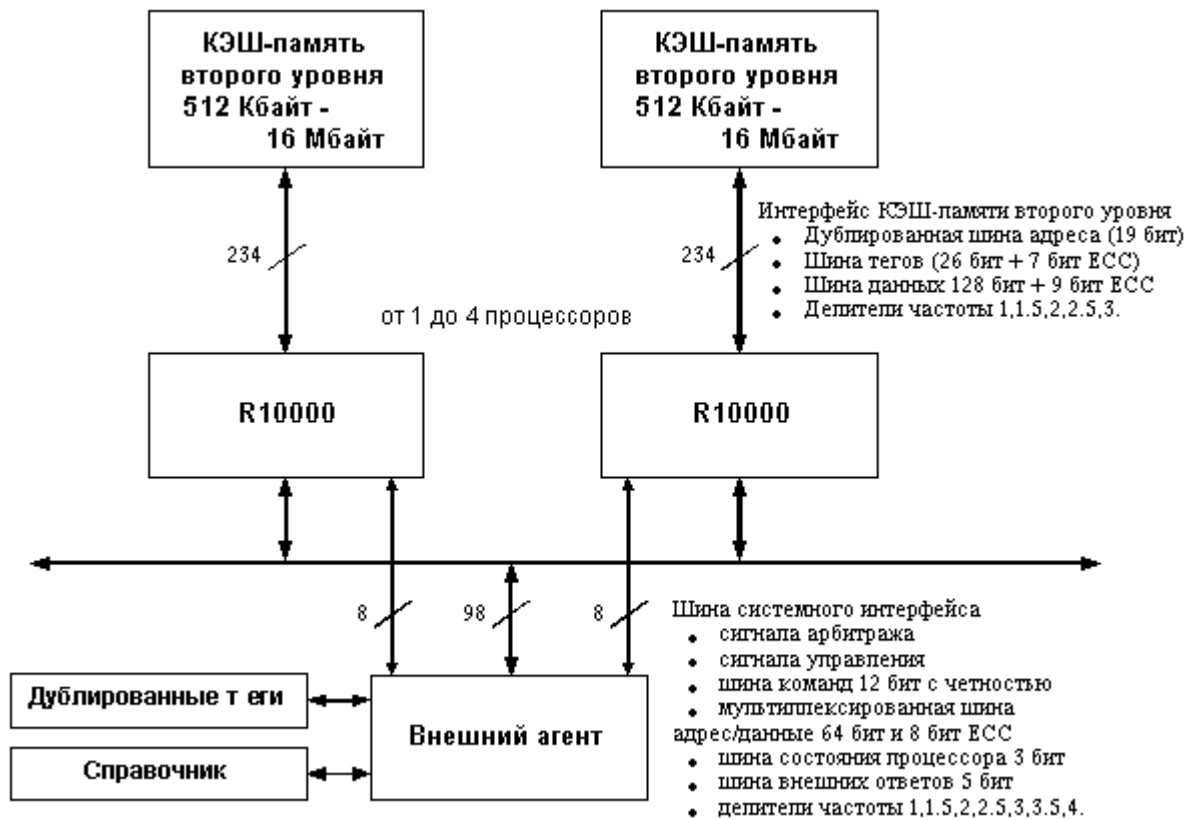


Рис. 4. Построение многопроцессорной системы на базе кластерной шины

Второй способ предназначен для достижения максимальной производительности минимальными затратами. Он подразумевает использование от двух до четырех процессоров, объединенных шиной Cluster Bus. В этом случае необходим только один внешний интерфейс для взаимодействия с другими ресурсами системы. Все процессоры связаны с одним и тем же внешним агентом. Реализация кластерной шины не только снижает сложность, но и количество заказных интегральных схем, а следовательно и стоимость системы, требуя только одного внешнего агента на каждые четыре процессора.

В дополнение к 64-битовой мультиплексированной шине адреса/данных имеется двухбитовая шина состояний, которая используется для выдачи ответов о состоянии процессорной когерентности. Кроме того, используется 5-битовая шина системных ответов внешним агентом для выдачи внешних ответов подтверждения. На рисунке 4 показана блок-схема конфигурации кластерной шины.

Сводка по микропроцессорам архитектуры MIPS

MIPS (Microprocessor without Interlocked Pipeline Stages — «микропроцессор без блокировок в конвейере») — семейство RISC-микропроцессоров, разработанное компанией MIPS Technologies. Архитектура MIPS использовалась в старых компьютерах SGI, а также во встроенных системах и в игровых консолях Nintendo 64, Sony PlayStation, Sony PlayStation 2 и Sony PSP.

Первые процессоры MIPS были 32-битными, позже была разработана 64-битная архитектура. Основная идея — сильно упростив внутреннее устройство процессора и используя очень длинный (по тем временам) конвейер, можно получить процессор, не умеющий выполнять сравнительно сложные инструкции, зато работающий на очень высоких тактовых частотах, позволяющих скомпенсировать потери производительности на эмуляцию этих сложных инструкций. Изначально предполагалось, что MIPS-процессоры не будут аппаратно поддерживать даже операции умножения и деления, благодаря чему можно было обойтись без сложных в реализации блокировок конвейера.

R10000 — последний процессор, созданный MIPS Technologies. Фирма, не желая конкурировать с Intel, собиравшейся разработать для этого рынка Itanium, свернула разработки и лицензировала архитектуру MIPS другим фирмам.

В настоящий момент процессоры архитектуры MIPS широко используются во встраиваемых устройствах с критичной производительностью.

Семейство процессоров с архитектурой MIPS

- R1000 (не выпускался: лабораторный образец). У него отсутствовало умножение и деление (они выполнялись программно)
- R2000
- R3000
- R4000
- R5000
- RM7000
- RM9000
- R8000
- R10000 (последний созданный MIPS); его дальнейшие модификации с увеличенным кэшем и частотой:
 - R12000
 - R14000
 - R16000
- существуют различные модификации других фирм.