

ПРОЦЕССОРЫ ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

А. ФРУНЗЕ, г. Москва

В предыдущих статьях А. Фрунзе [1, 2] рассказывалось о микропроцессорах, установленных в 90 % компьютеров, выпущенных к настоящему времени. Это — процессоры семейства x86. Однако, помимо них, существует большое число “компьютерных” процессоров, принципиально отличных от x86. Они существенно отличаются друг от друга как деталями архитектуры, так и системами команд, но всех их объединяет то, что они относятся к группе так называемых RISC-процессоров. Что же это за процессоры, где они применяются сегодня и каковы их перспективы на будущее? Ответы на эти вопросы — в публикуемой ниже статье.

RISC-ПРОЦЕССОРЫ: ПРЕДПОСЫЛКИ СОЗДАНИЯ

Аббревиатура RISC расшифровывается как Reduced Instruction Set Computer — компьютер с сокращенным набором команд, т. е. речь идет о процессорах, система команд которых существенно сокращена в сравнении с системой команд x86. Для того чтобы понять принципы, лежащие в основе RISC-процессоров, нужно вспомнить некоторые особенности развития микропроцессорных архитектур, проявившиеся в 80-е годы.

Аппаратные возможности процессоров определялись уровнем развития микроэлектронной технологии, который ограничивал число логических элементов, размещаемых на кристалле. Фактически его размеры и число размещаемых на нем транзисторов почти целиком определяли тот набор аппаратных устройств, которые включались в создаваемый процессор. Разработчики стремились ввести в его состав как можно большее число различных арифметических и логических устройств, сопроцессоров и т. д., поскольку очевидно, что аппаратная реализация какой-либо функции обеспечивала более быстрое ее выполнение, чем программная. Да и длина кода для реализуемых аппаратным путем функций существенно меньше. Поэтому в 80-х годах процессоры “росли вверх”, оснащались новыми узлами, совершенствуясь уже имеющиеся.

Кроме того, разработчикам процессоров приходилось постоянно подстраиваться под стремление программистов иметь возможно более мощную и гибкую систему команд. Такие системы команд с разнообразными способами адресации и наборами условных переходов и вызовов подпрограмм, строковыми операциями и префиксациями обеспечивали создание коротких программ, гарантировали меньшее число ошибок в них. Они также позволяли программистам создавать настоящие программные шедевры за счет нестандартного (а порой, и совершенно неожиданного) использования возможностей тех или иных команд. Нельзя сказать, что эти шедевры серьезным образом повлияли на тенденции развития программирования, но наличие или отсутствие таких возможностей в немалой степени определяло отношение программистов к тому или иному процессору. И разработчики не могли не считаться с этим.

В результате появились системы команд, подобные системе семейства x86, включающие в себя тысячи различных инструкций. Так сформировались CISC-процессоры (CISC — Complex Instruction Set Computer — компьютер с комплексным набором инструкций). Фактически в этой структуре шел перенос “центра тяжести” обработки с программного уровня системы на аппаратный.

Однако начиная с некоторого момента CISC-идеология стала не ускорителем, а тормозом роста производительности систем. Дело в том, что большое разнообразие команд и различная их длина существенно усложняют аппаратуру управления внутри процессора. Так, например, она должна определить длину команды (может варьироваться в пределах от одного до пяти-шести байт), выделить код операции, подготовить операнды, которые могут находиться как в регистрах, так и в оперативной памяти. Результат вычислений также нужно перенести или в память, или в какой-либо из регистров.

Большое число действий, которое должен совершить микропроцессор при выполнении команды, заставляло отводить под системы управления большую часть кристалла микросхемы (до 70 % в процессорах x86). В результате для арифметико-логических устройств (АЛУ), сопроцессоров и других устройств, повышающих производительность процессора, оставалось всего 30...40 % площади кристалла.

В то же время размеры кристаллов ограничиваются рядом причин. В первую очередь, это причины экономического характера: чем крупнее кристалл, тем вероятнее возможность появления дефектов, т. е. ниже процент выхода годных изделий, да и цена такого кристалла выше. Кроме того, есть и физические ограничения — с ростом размеров кристалла увеличивается тепловыделение и снижается максимально допустимая тактовая частота. Добавим к этому, что все возрастающее число логических элементов в устройстве управления увеличивает время выполнения им своих функций. Причина этого также заключается в ограничениях, накладываемых уровнем развития технологии. Прохождение сигнала через каждый логический элемент характеризуется некоторой задержкой по времени, и рост числа элементов, естественно, сопровождается увеличением суммарной задержки, которую не удается скомпенсировать соответствующим уменьшением ее, обусловленным прогрессом технологии.

Все вышеизложенное убедительно подтверждает тот факт, что повышение производительности CISC-процессоров превращается во все более и более сложную задачу.

ПРИНЦИПЫ RISC

Осознав указанные факты, многие исследователи уже в 80-е годы начали поиски альтернативных путей повышения производительности процессоров. Было ясно, что нужно попытаться уменьшить “накладные расходы”, отъедающие до двух третей полезной площади кристалла, снизив число логических элементов в них до минимума. Для этого было необходимо перенести “центр тяжести” на программные средства, оставив минимально возможное число команд, имеющих простую и регулярную структуру. Другими словами, начала формироваться стратегия “длинная программа — короткие команды” в противовес господствующей “короткая программа — длинные команды”. Оставалось только найти разумный компромисс между ресурсами, отводимыми основным вычислительным узлом процессора и устройствам управления.

И вскоре этот компромисс был найден. Вначале Джон Кук из IBM Research Labs на своем экспериментальном компьютере “model 801” показал, что при использовании в программе лишь простейших команд формата “регистр—регистр” (операнды извлекаются исключительно из регистров и после выполнения операции помещаются только в регистры) скорость выполнения большинства вычислительных задач возрастает в два-три раза. Затем учеными Стенфордского университета было показано, что исключение из набора команд сложных операций уменьшает число элементов процессора на порядок за счет упрощения схем управления ресурсами процессора (при этом производительность системы остается практически неизменной).

Опыт многочисленных исследований статистических свойств программ выявил, что 80 % процессорного времени обычно тратится на выполнение всего 20 % от общего числа инструкций обычных CISC-процессоров. Причем все эти 20 % приходится на команды типа “регистр—регистр”. И наконец, в работах Д. Паттерсона и К. Секуина были сформулированы четыре основных принципа, которые должны быть положены в основу процессоров группы RISC:

а) любая операция, к какому бы типу она ни принадлежала, должна выполняться за один такт;

б) операции обработки данных реализуются только в формате “регистр—регистр”. Обмен между памятью и регистрами (т. е. модификация памяти и чтение из нее необходимых данных) осуществляется только лишь с помощью специальных команд чтения/записи;

в) система команд должна содержать минимальное число наиболее часто используемых простейших команд одинаковой длины;

г) состав системы команд должен быть оптимизирован с учетом требований компиляторов языков высокого уровня.