

Информатика

Лекция 10



Суперскалярность

- Суперскалярность – архитектура вычислительного ядра, при которой наиболее нагруженные блоки могут входить в нескольких экземплярах. Скажем, в ядре процессора блок выборки инструкций может нагружать сразу несколько блоков декодирования.
- В этом случае блоки, выполняющие более сложные действия и работающие дольше, за счет параллельной обработки сразу нескольких инструкций не будут задерживать весь конвейер.
- Однако параллельное выполнение инструкций возможно, только если эти инструкции – независимые.

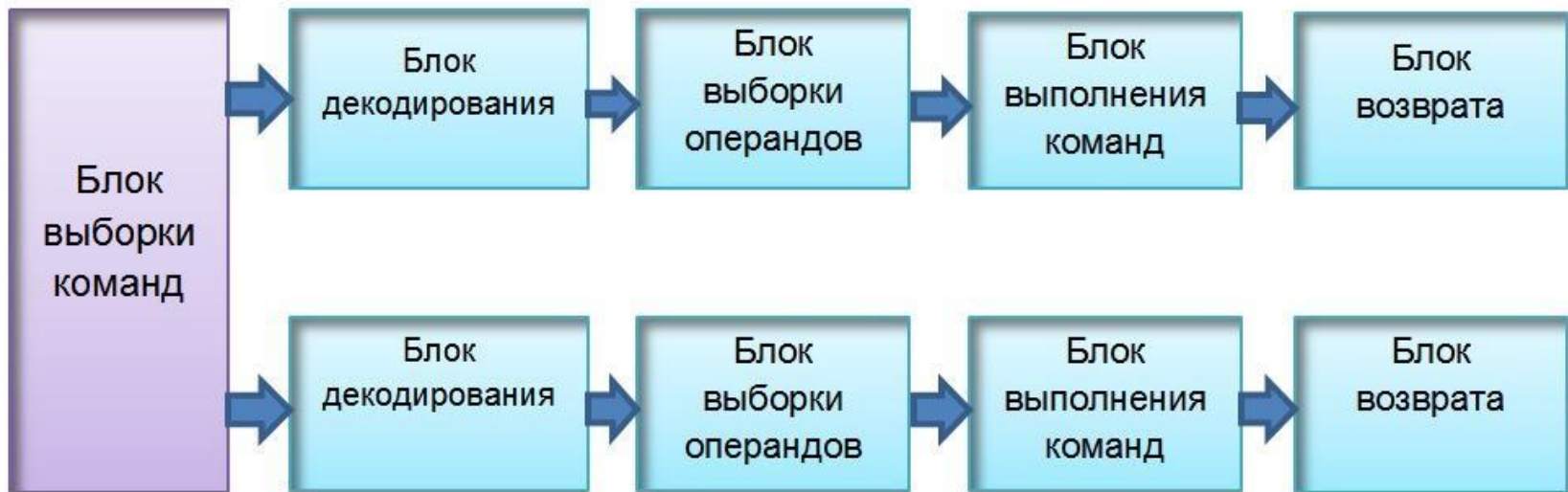
Конвейер x86

Процессоры x86 имеет 5-ти ступенчатый конвейер.

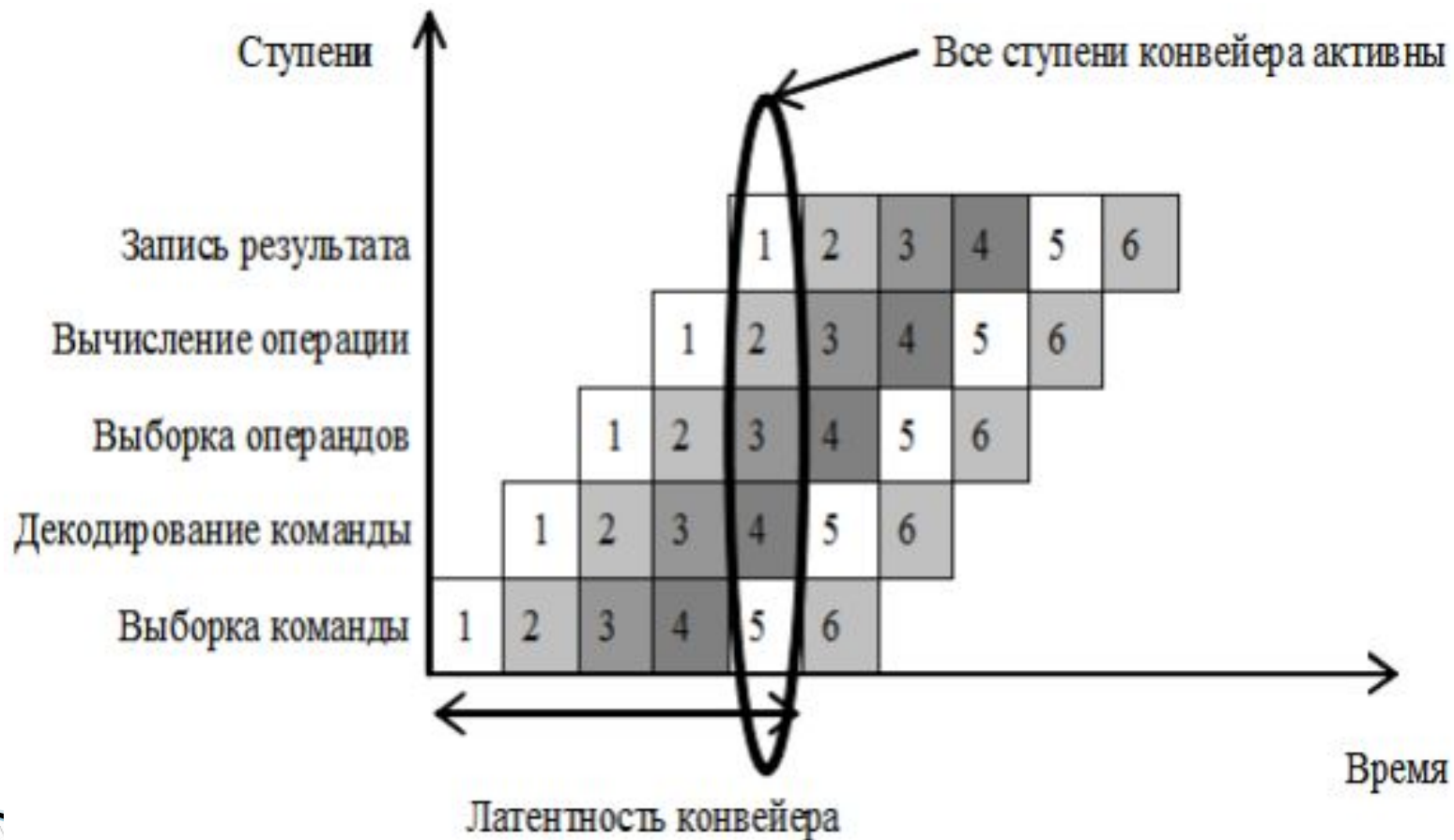
Соответствующие этапы включают:

1. выборку команд из кэш-памяти или оперативной памяти;
2. декодирование команды;
3. генерацию адреса, в процессе которой определяются адреса операндов в памяти;
4. выполнение операции с помощью АЛУ (арифметико-логического устройства);
5. запись результаты (адрес определяется конкретной машинной командой).

Конвейер



Конвейер



Конвейер

- ▣ Обработка команды, или цикл процессора, может быть разделена на несколько основных этапов, которые можно назвать микрокомандами.
- ▣ Каждая операция требует для своего выполнения времени, равного такту генератора процессора. Конвейеризация осуществляет многопоточную параллельную обработку команд, так что в каждый момент одна из команд считывается, другая декодируется и т. д., и всего в обработке одновременно находится пять команд.

Конвейер

- Таким образом, на выходе конвейера на каждом такте процессора появляется результат обработки одной команды (одна команда в один такт). Первая инструкция может считаться выполненной, когда завершат работу все пять микрокоманд.
- Такая технология обработки команд носит название конвейерной (pipeline) обработки. Каждая часть устройства называется ступенью конвейера, а общее число ступеней — длиной линии конвейера. С ростом числа линий конвейера и увеличением числа ступеней на линии увеличивается пропускная способность процессора при неизменной тактовой частоте.

КЭШ-память

- Несмотря на все технологии и уловки разработчиков, производительность процессора все-таки напрямую зависит от скорости выборки команд и данных из памяти.
- И даже, если процессор имеет сбалансированный и продуманный конвейер, использует технологию Hyper-Threading и так далее, но не обеспечивает должную скорость выборки данных и команд из памяти, то, в результате, общая производительность ЭВМ не оправдает ваших ожиданий.
- Поэтому один из важнейших параметров устройства процессора — это КЭШ-память, призванная сократить время выборки команд и данных из основной оперативной памяти и выполняющая роль промежуточного буфера с быстрым доступом между процессором и основной оперативной памятью.

КЭШ-память

- ❑ КЭШ-память строится на базе дорогой SRAM-памяти (static random access memory), обеспечивающей доступ к ячейкам памяти гораздо более быстрый, чем к ячейкам DRAM-памяти (dynamic random access memory), на базе которой построена оперативная память.
- ❑ SRAM-память не требует постоянной регенерации, что так же увеличивает ее быстродействие.
- ❑ КЭШ-память делится на несколько уровней. В современных процессорах, обычно, бывает три уровня, а в некоторых топовых моделях процессоров иногда встречается и четыре уровня КЭШ-памяти.

Типы КЭШ-памяти

- КЭШ-память более высокого уровня всегда больше по размеру и медленнее КЭШ-памяти более низкого уровня.
- Самая быстрая и самая маленькая КЭШ-память – это КЭШ-память первого уровня. Она обычно работает на частоте процессора, имеет объем несколько сотен килобайт и располагается в непосредственной близости от блоков выборки данных и команд.
- При этом она может быть единой (Принстонская архитектура) или разделяться на две части (Гарвардская архитектура): на память команд и память данных.
- В большинстве современных процессоров используют разделенную КЭШ-память первого уровня, так как это позволяет одновременно с выборкой команд осуществлять выборку данных, что крайне важно для работы конвейера.

Типы КЭШ-памяти

- КЭШ-память второго уровня – более медленная (время доступа, в среднем, 8-20 тактов процессора), но зато имеет объем несколько мегабайт.
- КЭШ-память третьего уровня – еще медленнее, но имеет сравнительно большой объем.
- В многоядерных процессорах, обычно, последний уровень КЭШ-памяти делают общим для всех ядер. Причем, в зависимости от нагрузки на ядра, может динамически изменяться отведенный ядру объем КЭШ-памяти последнего уровня. Если ядро имеет высокую нагрузку, то ему выделяется больше КЭШ-памяти, за счет уменьшения объема КЭШ-памяти для менее нагруженных ядер.

Принцип работы

- ❑ Процессор считывает из основной оперативной памяти данные и заносит их в КЭШ-память всех уровней, замещая данные, к которым давно и наиболее редко обращались.
- ❑ В следующий раз, когда процессору понадобятся эти же данные, они будут считаны уже не из основной оперативной памяти, а из КЭШ-памяти первого уровня, что значительно быстрее.
- ❑ Если к этим данным процессор долго не будет обращаться, то они будут постепенно вытеснены из всех уровней КЭШ-памяти, вначале из первого, так как он самый маленький по объему, затем из второго и так далее.
- ❑ Даже если эти данные останутся только в третьем уровне КЭШ-памяти, все равно обращение к ним будет быстрее, чем к основной памяти.

Особенности работы

- Однако, чем больше уровней КЭШ-памяти, тем сложнее алгоритм замещения устаревших данных и тем больше времени тратится на согласования данных во всех уровнях КЭШ-памяти.
- В результате, выигрыш от скорости работы КЭШ-памяти быстро сходит на нет. К тому же SRAM-память — очень дорогая, и при больших объемах, а, как помните, каждый новый уровень КЭШ-памяти должен быть больше предыдущего, быстро снижается показатель цена-качество, что крайне негативно сказывается на конкурентоспособности процессора.
- Поэтому на практике больше четырех уровней КЭШ-памяти не делают.

Прирост производительности

- Эксперименты свидетельствуют, что в среднестатистическом "домашнем" процессоре влияние размера кэша на производительность находится в пределах 10 %, и его вполне можно компенсировать, например, высокой частотой.
- Наличие кэша L3 обеспечивает прирост производительности 8%.

Архитектура ЭВМ

- Архитектура ЭВМ - это многоуровневая иерархия аппаратурнопрограммных средств, из которых строится ЭВМ. Каждый из уровней допускает многовариантное построение и применение. Конкретная реализация уровней определяет особенности структурного построения ЭВМ.
- В **широком** смысле архитектура охватывает понятие организации системы, включающее такие аспекты разработки компьютера как систему памяти, структуру системной шины, организацию ввода вывода и т.п.
- В **узком** смысле под архитектурой понимают архитектуру набора команд.

Основные архитектуры набора команд

- На современном этапе развития вычислительной техники существуют две основные архитектуры набора команд, используемые компьютерной промышленностью. - архитектуры CISC' (Complete Instruction Set Computer) и RISC (Restricted (reduced) Instruction Set Computer).
- Основоположником CISC-архитектуры - архитектуры с полным набором команд можно считать фирму IBM с ее базовой архитектурой IBM 360, ядро которой используется с 1964 г. и дошло до наших дней, например, в таких современных мейнфреймах, как IBM ES/9000.
- Лидером в разработке микропроцессоров с полным набором команд считается компания Intel с микропроцессорами X86. Это практически стандарт для рынка микропроцессоров.

Регистро-ориентированная RISC - архитектура

- При проектировании супер-миникомпьютеров на базе последних достижений СБИС-технологии оказалось невозможным полностью перенести в нее архитектуру удачного компьютера, выполненного на другой элементной базе. Такой перенос был бы очень неэффективен из-за технических ограничений на ресурсы кристалла: площадь, количество транзисторов, мощность рассеивания и т. д.
- Для снятия указанных ограничений в Беркли (США. Калифорния) была разработана регистро-ориентированная RISC - архитектура. Компьютеры с такой архитектурой иногда называют компьютерами с сокращенным набором команд.
- Суть ее состоит в выделении наиболее употребительных операций и создании архитектуры, приспособленной для их быстрой реализации. Это позволило в условиях ограниченных ресурсов разработать компьютеры с высокой пропускной способностью.

Основные принципы RISC-архитектуры:

- каждая команда независимо от ее типа выполняется за один *машинный цикл*, длительность которого должна быть максимально короткой:
- все команды должны иметь *одинаковую длину* и использовать минимум адресных форматов, что резко упрощает логику центрального управления процессором:
- обращение к памяти происходит только при выполнении операций записи и чтения, вся *обработка данных* осуществляется исключительно в *регистровой структуре* процессора:
- система команд должна обеспечивать *поддержку языка высокого уровня* (имеется в виду подбор системы команд, наиболее эффективной для различных языков программирования).

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 Read-Modify-Write («прочитать-изменить-записать») отсутствуют. Любые операции «изменить» выполняются только над содержимым регистров (т. н. архитектура load-and-store).
- Большое количество регистров общего назначения (32 и более).

Характерные особенности RISC-процессоров

- Отсутствие поддержки операций вида «изменить» над укороченными типами данных — байт, 16-разрядное слово. Так, например, система команд DEC Alpha содержала только операции над 64-разрядными словами, и требовала разработки и последующего вызова процедур для выполнения операций над байтами, 16- и 32-разрядными словами.

□

Характерные особенности RISC-процессоров

- Отсутствие микропрограмм внутри самого процессора.
- То, что в CISC-процессоре исполняется микропрограммами, в RISC-процессоре исполняется как обыкновенный (хотя и помещённый в специальное хранилище) машинный код, не отличающийся принципиально от кода ядра ОС и приложений.

Главные отличия архитектуры RISC от CISC

- RISC - Набор команд в процессорах построенных на данной архитектуре упрощенный. Что позволяет поднять частоту, снизить стоимость производства и оптимально распараллеливать задачи. Это основное отличие.
- В CISC архитектуре соответственно длина команды не ограничена, одна инструкция содержит (может содержать) несколько арифметических действий. Как следствие - требуются суперскалярные вычисления и использование в процессоре конвейера. По сути процессор построенный по CISC архитектуре выполняет те же команды что и процессор на RISC архитектуре, но внутри себя содержит командный интерпретатор, "переделывающий" сложные команды в несколько простых.

Достоинства RISC-процессоров

- В результате, хотя RISC-процессоры и требуют выполнения большего количества инструкций для решения одной и той же задачи, по сравнению с CISC-процессорами, они, в общем случае, показывают более высокую производительность.
- Во-первых, выполнение одной RISC-инструкции занимает гораздо меньше времени, чем выполнение CISC-инструкции.
- Во-вторых, RISC-процессоры более широко используют возможности параллельной работы.
- В-третьих, RISC-процессоры могут иметь более высокую тактовую частоту, по сравнению с CISC-процессорами.

Распространение RISC-процессоров

- ❑ Однако, несмотря на явное преимущество RISC, процессоры не получили столь серьезного распространения, как CISC.
- ❑ Правда, связано это в основном не с тем, что они по каким-то параметрам могли быть хуже CISC-процессоров. Они не хуже.
- ❑ Дело в том, что CISC-процессоры появились первыми, а программное обеспечение для CISC - процессоров – несовместимо с RISC-процессорами.

Распространение RISC-процессоров

- В результате, экономически крайне невыгодно переписывать все программы, которые уже разработаны, отлажены и используются огромным количеством пользователей. Вот так и получилось, что теперь мы вынуждены использовать CISC-процессоры.
- Разработчики нашли компромиссное решение данной проблемы, и уже очень давно в CISC-процессорах используют RISC-ядро и замену сложных команд на микропрограммы. Это позволило несколько сгладить ситуацию. Но все же RISC-процессоры по большинству параметров выигрывают даже у CISC-процессоров с RISC-ядром.

- На сегодняшний день CISC- процессоры почти монополюно занимают на компьютерном рынке сектор персональных компьютеров.
- RISC-процессорам нет равных в секторе высокопроизводительных серверов и рабочих станций.

Архитектура ARM

- ▣ **ARM** (от англ. *Advanced RISC Machine* — усовершенствованная RISC-машина; иногда — *Acorn RISC Machine*) — семейство лицензируемых 32-битных и 64-битных микропроцессорных ядер разработки компании ARM Limited.

Популярность

- В 2007 году около 98 % из более чем миллиарда мобильных телефонов, продаваемых ежегодно, были оснащены, по крайней мере, одним процессором ARM.
- По состоянию на 2009 на процессоры ARM приходилось до 90 % всех встроенных 32-разрядных процессоров.
- Процессоры ARM широко используются в потребительской электронике — в том числе КПК, мобильных телефонах, цифровых носителях и плеерах, портативных игровых консолях, калькуляторах и компьютерных периферийных устройствах, таких, как жесткие диски или маршрутизаторы.
- Эти процессоры имеют низкое энергопотребление.

Влияние ARM-технологии на рынок

- В основном процессоры семейства завоевали сегмент массовых мобильных продуктов (сотовые телефоны, карманные компьютеры) и встраиваемых систем средней и высокой производительности (от сетевых маршрутизаторов и точек доступа до телевизоров).
- Отдельные компании заявляют о разработках эффективных серверов на базе кластеров ARM процессоров, но пока это только экспериментальные проекты с 32-битной архитектурой

MISC

- ▣ **MISC (Minimal Instruction Set Computer)** – дальнейшее развитие архитектуры RISC, основанное на еще большем упрощении инструкций и уменьшении их количества.
- ▣ Так, в среднем, в MISC-процессорах используется 20-30 простых инструкций.
- ▣ Такой подход позволил еще больше упростить устройство процессора, снизить энергопотребление и максимально использовать возможности параллельной обработки данных.

VLIW

- ▣ **VLIW (Very long instruction word)** – архитектура процессоров, использующая инструкции большой длины, содержащие сразу несколько операций, объединенных компилятором для параллельной обработки.
- ▣ В некоторых реализациях процессоров длина инструкций может достигать 128 или даже 256 бит.

Преимущества и недостатки

- Подход VLIW сильно упрощает архитектуру процессора, перекладывая задачу распределения вычислительных устройств на компилятор. Поскольку отсутствуют большие и сложные узлы, сильно снижается энергопотребление.
- В то же время код для VLIW обладает невысокой плотностью. Из-за большого количества пустых инструкций для простаивающих устройств программы для VLIW-процессоров могут быть гораздо длиннее, чем аналогичные программы для традиционных архитектур.

Преимущества и недостатки

- Архитектура VLIW выглядит довольно экзотической и непривычной для программиста.
- Из-за сложных внутренних зависимостей кода программирование вручную, на уровне машинных кодов для VLIW-архитектур, является достаточно сложным.
- Приходится полагаться на оптимизацию компилятора.