

Архитектура по регистрам в ядре

- Регистровая архитектура
- Аккумуляторная архитектура
- Стековая архитектура

Регистровая архитектура

- характеризуется свободным доступом к регистрам для выборки всех аргументов и записи результата. Элементарны арифметико-логические операции в таких процессорах кодируются в двух-, или трёхоперандные инструкции (регистр+регистр→регистр, иногда регистр результата совпадает с источником одного из аргументов).
- Все регистры кроме PC, SP и состояния равноправны и могут содержать операнд, адрес операнда
- Чем больше регистров, тем быстрее процессор и короче код. В регистрах (а не в ОЗУ) можно размещать локальные переменные, например счётчики циклов (счётчики i , j и т.п)
- Характерна для RISC процессоров.

Аккумуляторная архитектура

- **Аккумуляторная архитектура** — из регистров выделяется один - аккумулятор (или несколько регистров-аккумуляторов). Регистр-аккумулятор является источником одного из аргументов и **приёмником результата** вычислений. Операции кодируются как правило в однооперандные инструкции (аккумулятор+операнд→аккумулятор). Такая архитектура характерная для многих CISC-процессоров (напр. Z80, x86 (i3,i7)).

Стековая архитектура

- **Стековая архитектура** — определяется организацией регистрового файла в виде стека, и косвенной адресацией регистров через указатель стека, который определяет положение вершины стека, операции производятся над значениями на вершине стека и результат кладётся также на вершину. Арифметические операции кодируются в нуль-операндные инструкции. Стековая архитектура является неотъемлемой частью MISC-процессоров

Введение в ядро КЭШ памяти

- Кэширование — это использование дополнительной быстродействующей памяти (кеш-памяти) для хранения копий блоков информации из основной (оперативной) памяти, вероятность обращения к которым в ближайшее время велика.
- Уровни КЭШ: L1-на каждом ядре, L2- общая для ядер раньше была на плате процессора, сейчас на кристалле МП, L3-дополнительная медленная

Расширение шины данных

- За 1 такт читается больше требуемой длины данных . Так, процессоры AMD K8 умели производить одновременно 64-битные запись и чтение, либо два 64-битных чтения за такт, AMD K8L может производить два 128-битных чтения или записи в любой комбинации. Процессоры Intel Core 2 могут производить 128-битные запись и чтение за такт.
- В итоге процессор может реже читать/писать память, выполняя несколько операций по результатам одного чтения из ПЗУ/ОЗУ.

Упреждающее чтение и фиктивное выполнение

- Процессор при свободной шине ОЗУ считывает в КЭШ из ОЗУ наперёд несколько (иногда десятков, сотен) команд. При выполнении команды извлекаются из КЭШ обычно за 1 такт.
- При появлении команд ветвления, фиктивно выполняются команды по разным веткам ветвления, пока реально не выполнится команда, результат которой определит реальный путь ветвления. Результаты выполнения (накопленные в параллельных регистрах процессора) ложных веток отбрасываются, а реальной ветки принимаются как выполненные за 1 такт.

Задачи и блоки ядра

В течение каждого цикла команды **ЦП** выполняет много управляющих функций:

- 1) помещает адрес команды в адресную шину памяти;
- 2) получает команду из шины ввода данных и дешифрует ее;
- 3) выбирает адреса и данные, содержащиеся в команде; адреса и данные могут находиться в памяти или в регистрах;
- 4) выполняет операцию, определенную в коде команды. Операцией может быть арифметическая или логическая функция, передача данных или функция управления;
- 5) следит за управляющими сигналами, такими как прерывание, и реагирует соответствующим образом;
- 6) генерирует сигналы состояния, управления и времени, которые необходимы для нормальной работы УВВ и памяти.

Обычно каждую функцию ядра выполняет блок ядра.

Выполнение операций (логических, целочисленных, с плав. запятой) осуществляют **исполнительные блоки**

Уменьшение тактов на одну команду

- В ранних МК (в МК Intel 8051) на 1 простейшую команду требовалось 3 цикла (чтение, декодирование и исполнение, запись результата) по 4 такта. Итого команда выполнялась за 12 тактов. При частоте 12 МГц тот процессор имел производительность 1 MIPS а его современные аналоги имеют при той же частоте 12 MIPS и более.
- Затем при введении RISC MISC простые команды (логические, целочисленные + - потом * и /) научились выполнять за 1 такт
- Сейчас в ядрах топовых RISC и MISC МП и МК любые команды, в том числе и с плав. запятой выполняются за 1 такт (и в среднем менее чем за 1).

Конвейер

- Процессор, выполняющий каждую команду за четыре такта (выбор команды, декодирование и выбор операндов, исполнение, запись результата) обеспечивает работой исполнительное устройство только в такте «исполнение». Для увеличения столь низкой (25 %) эффективности используется конвейерная организация: такты последовательно выбираемых команд совмещаются во времени так, что в каждый период одна из команд обязательно находится на этапе «исполнение», загружая исполнительское устройство.
- В итоге за 1 такт выполняется до 4 (и более) команд. В ARM CM3 в среднем 1,25... 1,5 команды за такт.

Реализация конвейера с 4-х тактными инструкциями

- | | T1 | T2 | T3 | T4 | T5 | T6 | |
|------|--------|---------|------------|------------|------------|------------|-------------------|
| • И1 | Чтение | декодир | исполнение | запись | | | |
| • И2 | | Чтение | декодир | исполнение | запись | | |
| • И3 | | | Чтение | декодир | исполнение | запись | |
| • И4 | | | | Чтение | декодир | исполнение | запись |
| • И5 | | | | | Чтение | декодир | исполнение запись |
| • | | | | | | | |
- В итоге за один такт (напрмер T4) выполняется одновременно 4 инструкции=команды.

Суперскалярные архитектуры

- **Суперскалярность** — архитектура вычислительного ядра, использующая несколько декодеров команд, которые могут **нагружать работой множество исполнительных блоков**. (в простом конвейере- по одному блоку)
- **Суперскалярность** -Способность выполнения нескольких машинных инструкций за один такт процессора. Появление этой технологии привело к существенному увеличению производительности

Многоядерность

- Вершина суперскалярности, **когда всех компонентов ядра** (не только исполнительных блоков а также узлов чтения, декодирования, записи, синхронизации, прерываний и т.д.) **более 1** и можно выделить автономные ядра с полным набором компонентов ядра, которые могут работать автономно и независимо друг от друга

VLIW-архитектура

- 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 длинные, 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 (и его аналог Эльбрус-3).
- Исследования показывают, что VLIW DSP (TMS320C6x фирмы Texas Instruments) выполняет мультимедийные задачи в 0,63 – 9 раз быстрее, чем суперскалярный процессор Pentium II без μ SIMD-расширения на той же тактовой частоте

CUDA , GPGPU

- **CUDA** (англ. *Compute Unified Device Architecture*) — технология GPGPU (англ. *General-Purpose computing on Graphics Processing Units*), позволяющая программистам реализовывать на языке программирования Си алгоритмы, выполнимые на графических процессорах ускорителей GeForce восьмого поколения и старше (GeForce 8 Series, GeForce 9 Series, GeForce 200 Series), Nvidia Quadro и Tesla компании Nvidia. Технология CUDA разработана компанией nVidia.
- В современных графических процессорах 128-256 ядер! Которые можно использовать для обычных вычислений, когда плата не сильно загружена графикой.