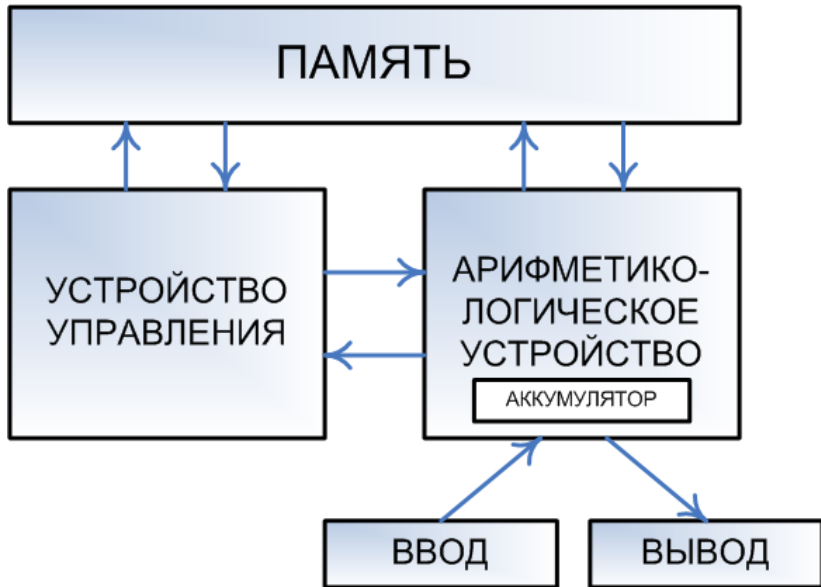


Введение

¹Кафедра информационных технологий и систем
Национальная металлургическая академия Украины

4 сентября 2012 г.

Архитектура фон Неймана — широко известный принцип совместного хранения программ и данных в памяти компьютера. Вычислительные системы такого рода часто обозначают термином «машина фон Неймана», однако соответствие этих понятий не всегда однозначно. В общем случае, когда говорят об архитектуре фон Неймана, подразумевают физическое отделение процессорного модуля от устройств хранения программ и данных.



Принцип двоичного кодирования

Согласно этому принципу, вся информация, поступающая в ЭВМ, кодируется с помощью двоичных сигналов (двоичных цифр, битов) и разделяется на единицы, называемые словами.

Принцип однородности памяти

Программы и данные хранятся в одной и той же памяти.

Поэтому ЭВМ не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Принцип адресуемости памяти

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к хранящимся в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

Принцип последовательного программного управления
Предполагает, что программа состоит из набора команд,
которые выполняются процессором автоматически друг за
другом в определенной последовательности.

Принцип жесткости архитектуры

Неизменяемость в процессе работы топологии, архитектуры, списка команд.

Компьютеры, построенные на этих принципах, относят к типу фон-неймановских.

Совместное использование шины для памяти программ и памяти данных приводит к узкому месту архитектуры фон Неймана, а именно ограничению пропускной способности между процессором и памятью по сравнению с объемом памяти. Из-за того, что память программ и память данных не могут быть доступны в одно и то же время, пропускная способность является значительно меньшей, чем скорость, с которой процессор может работать. Это серьезно ограничивает эффективное быстроедействие при использовании процессоров, необходимых для выполнения минимальной обработки на больших объемах данных. Процессор постоянно вынужден ждать необходимых данных, которые будут переданы в память или из памяти. Так как скорость процессора и объем памяти увеличивались гораздо быстрее, чем пропускная способность между ними, узкое место (бутleneck) стало большой проблемой, серьезность которой возрастает с каждым новым поколением процессоров.

SISD (англ. Single Instruction, Single Data) или ОКОД (Одиночный поток Команд, Одиночный поток Данных) — архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя одним потоком данных. Относится к фон-Неймановской архитектуре.

SISD компьютеры это обычные, ”традиционные” последовательные компьютеры, в которых в каждый момент времени выполняется лишь одна операция над одним элементом данных (числовым или каким-либо другим значением). Большинство персональных ЭВМ до последнего времени, например, попадает именно в эту категорию. Иногда сюда относят и некоторые типы векторных компьютеров, это зависит от того, что понимать под потоком данных.

SIMD (англ. single instruction, multiple data — одиночный поток команд, множественный поток данных, ОКМД) — принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных.

SIMD-компьютеры состоят из одного командного процессора (управляющего модуля), называемого контроллером, и нескольких модулей обработки данных, называемых процессорными элементами. Управляющий модуль принимает, анализирует и выполняет команды. Если в команде встречаются данные, контроллер рассылает на все процессорные элементы команду, и эта команда выполняется на нескольких или на всех процессорных элементах. Каждый процессорный элемент имеет свою собственную память для хранения данных.

Одним из преимуществ данной архитектуры считается то, что в этом случае более эффективно реализована логика вычислений. До половины логических инструкций обычного процессора связано с управлением выполнением машинных команд, а остальная их часть относится к работе с внутренней памятью процессора и выполнению арифметических операций. В SIMD компьютере управление выполняется контроллером, а "арифметика" отдана процессорным элементам.

MISC (англ. minimal instruction set computer «минимальный набор команд компьютера») процессорная архитектура. Увеличение разрядности процессоров привело к идее укладки нескольких команд в одно большое слово (связку, bound). Это позволило использовать возросшую производительность компьютера и его возможность обрабатывать одновременно несколько потоков данных. Кроме этого MISC использует стековую модель вычислительного устройства и основные команды работы со стеком языка Forth. MISC-принцип может лежать в основе микропрограммы выполнения Java- и .Net- программ, хотя по количеству используемых команд они нарушают принцип MISC.

CISC (англ. Complex instruction set computing, или англ. Complex instruction set computer компьютер с комплексным набором команд) концепция проектирования процессоров, которая характеризуется следующим набором свойств:

- нефиксированное значение длины команды;
- арифметические действия кодируются в одной команде;
- небольшое число регистров, каждый из которых выполняет строго определённую функцию.

RISC (англ. Restricted (reduced) instruction set computer компьютер с сокращённым набором команд) архитектура процессора, в которой быстродействие увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения короче. Первые RISC-процессоры даже не имели инструкций умножения и деления. Это также облегчает повышение тактовой частоты и делает более эффективной суперскалярность (распараллеливание инструкций между несколькими исполнительными блоками).

VLIW (англ. very long instruction word «очень длинная машинная команда») архитектура процессоров с несколькими вычислительными устройствами. Характеризуется тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно. Фактически это «видимое программисту» микропрограммное управление, когда машинный код представляет собой лишь немного свёрнутый микрокод для непосредственного управления аппаратурой.