

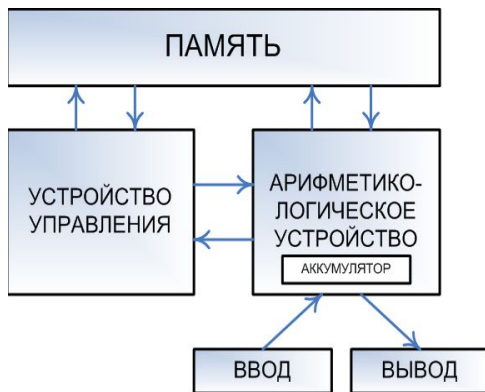
Lecture

№3

Принципы фон Неймана.

Архитектура по фон Нейману.

Что делает процессор в компьютере?



Джон фон Нейман, John von Neumann (родился 28.12.1903 в Будапеште; умер 08.02 1957 в Вашингтоне) - работал в областях математики, физики, химии, астрономии, биологии, экономики.

Известность фон Нейману принесли работы по математическому обоснованию квантовой механики. Он доказал, что нельзя исключить недетерминистские элементы в процессе измерений. Интересовался проблемами прогнозирования вероятностных процессов: исхода азартных игр, изменений климата и др. Идея фон Неймана о создании надежной машины из ненадежных элементов стала принципом создания электронных вычислительных машин и сетей.

В 1930 году фон Нейман эмигрировал из Европы в США, где работал в основном в Принстонском университете. Участвовал в разработке атомной бомбы. В 1946 году вместе с Клаусом Фуком запатентовал термоядерное взрывное устройство, приводимое в действие атомным взрывом - т.е. саму идею водородной бомбы.

Погиб от рака, вызванного облучением при испытаниях водородного боеприпаса на атолле Бикини (Маршалловы острова).



Neumann J. and co. "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument"
— Institute for Advanced Study, Princeton, N. J., July 1946

Принцип двоичности:

Для представления данных и команд используется двоичная система счисления

Принцип адресуемости памяти:

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка

Принципы фон Неймана

Принцип программного управления:

Программа состоит из набора команд, которые выполняются процессором друг за другом в определённой последовательности

Принцип последовательного программного управления:

Все команды располагаются в памяти и выполняются последовательно, одна после завершения другой

Принцип однородности памяти:

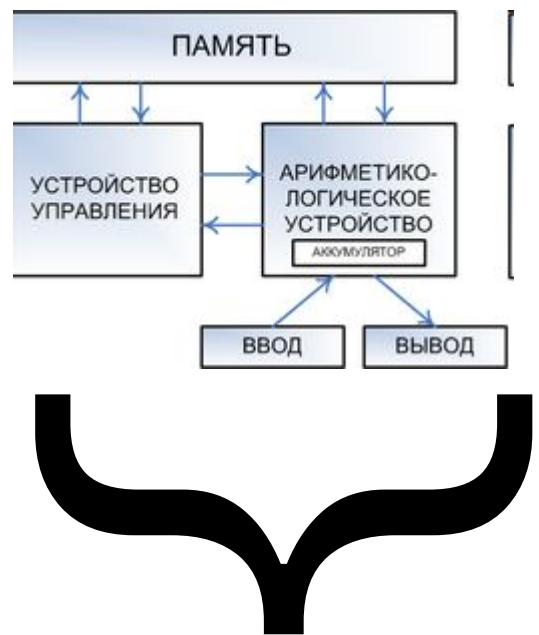
Как программы (команды), так и данные хранятся в одной и той же памяти (и кодируются в одной и той же системе счисления — чаще всего двоичной). Над командами можно выполнять такие же действия, как и над данными

Принцип условного перехода:

Команды из программы не всегда выполняются одна за другой. Возможно присутствие в программе команд условного перехода, которые изменяют последовательность выполнения команд в зависимости от значений данных



Машина фон Неймана



Под архитектурой фон Неймана, подразумевают физическое отделение процессорного модуля от устройств хранения программ и данных.

CPU: центральный процессор

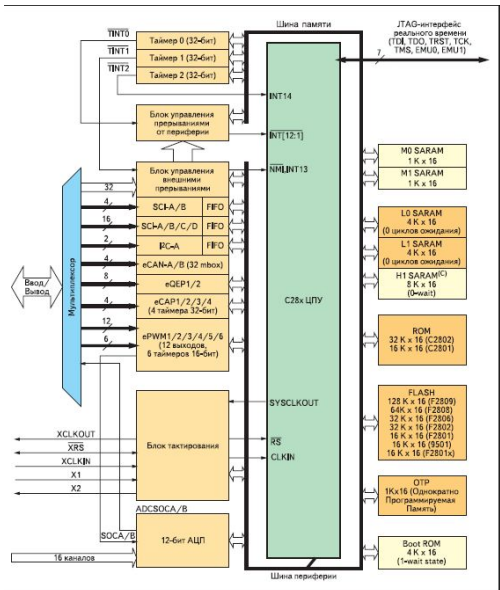
выбирает и исполняет

Имеет


команды из памяти последовательно, а адрес очередной команды задается «счетчиком адреса» в блока управления.

Выполняемые действия определяются блоком управления и АЛУ, которые вместе являются основой центрального процессора(CPU)

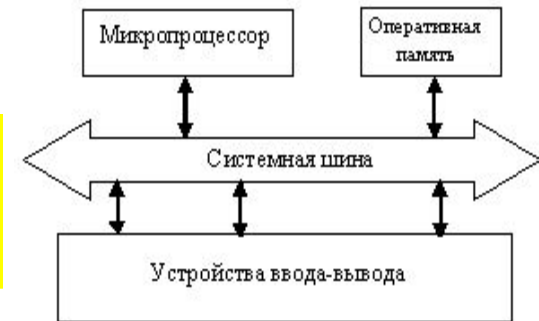
Набор регистров, часть которых доступна для хранения операндов, выполнения действий над ними и формирования адреса инструкций и операндов в памяти, другая часть – для системных целей



Оперативная память(ОЗУ)



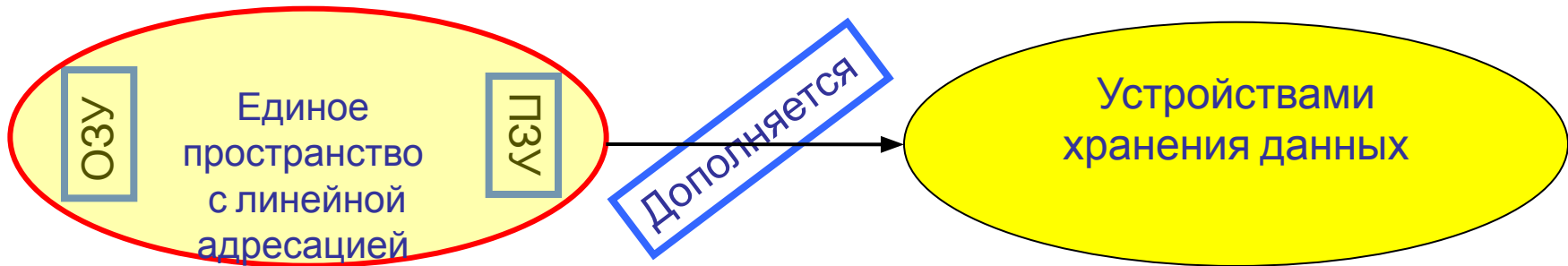
Оперативная память – самый большой массив ячеек памяти со смежными адресами – реализуется, как правило, на модулях (микросхемах) динамической памяти.



Для повышения производительности обмена данными(включая и считывание команд) оперативная память кэшируется сверхоперативной памятью(CACHE)

Cache I,II территориально располагаются в микропроцессоре

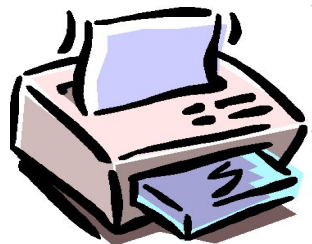
Оперативная память вместе с кэшем всех уровней представляет собой единый массив памяти, непосредственно доступный процессору для R/W data + R program code



Устройства ввода-вывода

Служат для преобразования информации из внутреннего представления в компьютере (биты и байты) в форму, доступную для окружению, и обратно.

Input/output devices



Периферия

Пространства памяти и ввода-вывода
неравнозначны не только по объёму,
но и по способам обращения

С помощью устройств хранения
и paging

создается

Виртуальная адресация:
Иллюзия создания ROM
гигантского размера
(если ОС поддерживает)

Реальная адресация к памяти – в этом случае физический
адрес совпадает с логическим.

Логический адрес текущей выполняемой инструкции хранится в указателе инструкций(IP),
который соответствует счетчику команд фон-неймановской машины

Способы адресации

ПАМЯТИ


I/O

>9

2

К портам I/O
обращаются
только по реальным
адресам, а
виртуализация
возможна только
программными
средствами

Архитектура и микроархитектура процессора



Архитектура процессора – это его программная модель, то есть программно-видимые свойства.

Микроархитектура процессора – это внутренняя реализация этой программной модели.

Микроархитектура

Переименование регистров(register renaming)

Исполнение по предложению
(Speculative execution)

Конвейеризация
(pipelining)

Продвижение данных
(data forwarding)

Предсказание переходов
(branch prediction)

Исполнение с изменением последовательности инструкции(out-of-order execution)

Конвейеризация (pipelining):

Предполагает разбивку выполнения каждой инструкции на несколько этапов, причем каждый этап выполняется на своей ступени конвейера процессора

Одновременно может обрабатываться несколько инструкций, и производительность процессора можно оценивать темпом выхода инструкций со всех его конвейеров

Для достижения максимальной производительности процессора
Надо обеспечить полную загрузку конвейеров с минимальным числом лишних штрафных циклов (penalty cycles).

Суперконвейерная архитектура в настоящее время имеет от 20 конвейеров

При записи промежуточных результатов устанавливается соответствие логических имен и физических регистров

Переименование регистров (register renaming):

Позволяет обойти архитектурное ограничение на возможность параллельного исполнения инструкций (доступно всего лишь 8 общих регистров)

Т.о., одновременно может исполняться несколько инструкций, ссылающихся на одно и тоже логическое имя регистра, при условии, что между ними нет фактических зависимостей по данным.

Продвижение данных (data forwarding):

Подразумевает начало исполнения инструкции до готовности всех операндов. При этом выполняются все возможные действия, и декодированная инструкция с одним операндом помещается в исполнительное устройство, где дожидается готовности второго операнда, выходящего с другого конвейера.

Предсказание переходов (branch prediction):

Позволяет продолжать выборку и декодирование потока инструкций после выборки инструкций ветвления (условного перехода), не дожидаясь проверки условия.



Иркутский
государственный университет





Иркутский
государственный университет





Иркутский
государственный университет





Иркутский
государственный университет





Иркутский
государственный университет





Иркутский
государственный университет

