

# Архитектура



*Уровень внешний,  
концептуальный и внутренний*



# Архитектура ЭВМ

# Пользователи компьютеров:

- **Конечные пользователи >90%**
- **Прикладные программисты <10%**
- **Системные программисты <1%**

# Способы описания архитектуры компьютеров

- Словесные описания, чертежи, блок-схемы и т.п.
- *Язык машины , язык Ассемблера*
- *Формальные языки для описания учебной ЭВМ*

# Какие ЭВМ будем изучать?

1. Абстрактная машина (машина Фон Неймана)
2. Специальные **учебные ЭВМ**
3. Архитектура первой (младшей) модели **конкретного** компьютера фирмы Intel
4. Отличительные особенности архитектуры современных компьютеров

# Вопросы и упражнения

1. Для чего необходимо выделять различные уровни видения архитектуры компьютера?
2. Сформулируйте различия в уровнях видения архитектуры компьютера для конечного пользователя и прикладного программиста.

# Машина Фон Неймана



Джон фон Нейман 1903-57.

В 1946 работающий в то время в Англии венгерский математик Джон фон Нейман (с соавторами) описал в техническом докладе архитектуру некоторого абстрактного вычислителя, который сейчас принято называть *машиной Фон Неймана*. Эта машина является *абстрактной моделью ЭВМ*

Машина Фон Неймана не поддаётся реализации по тому, что многие детали в архитектуре этого исполнителя алгоритма *не конкретизированы*

На каком уровне видения архитектуры рассмотрена ЭВМ?

# Схема машины фон Неймана

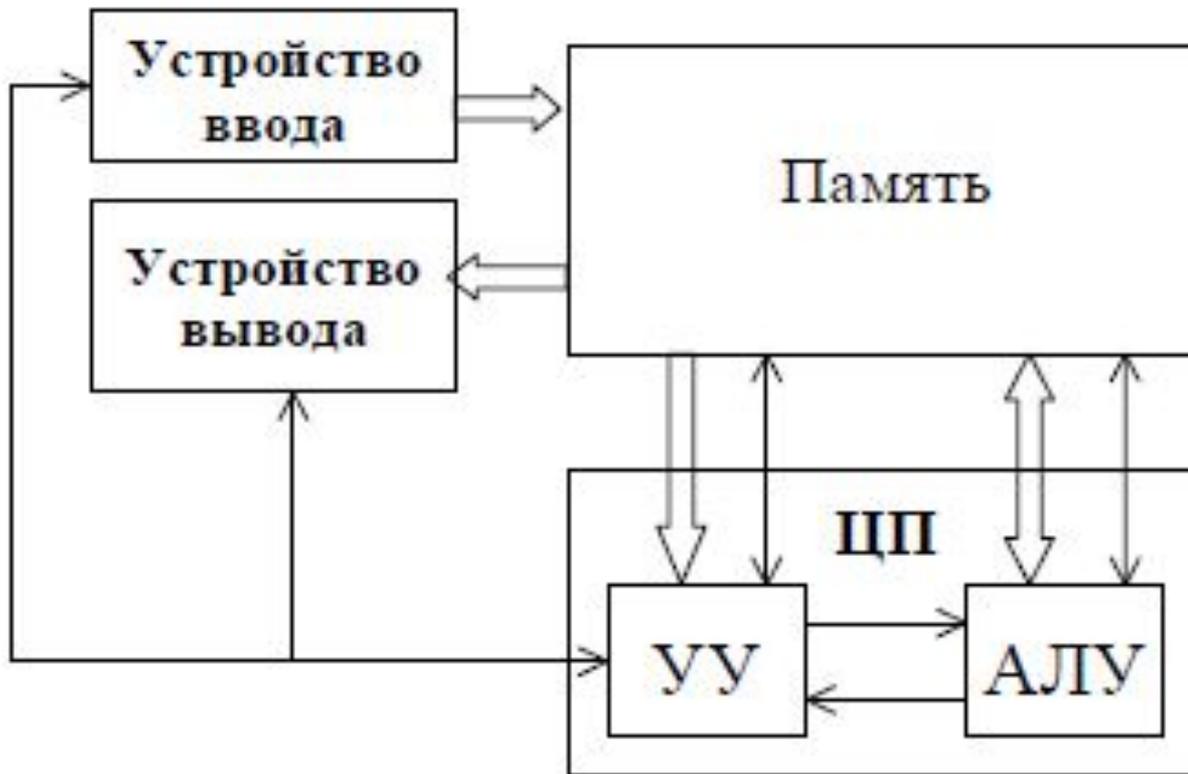
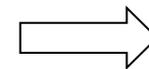


Схема машины фон Неймана.



- Потоки команд и данных



- передача между отдельными устройствами компьютера управляющих и информационных сигналов

# Принципы Фон Неймана

## 1. Принцип линейности и однородности памяти

*Память* машины Фон Неймана – это линейная (упорядоченная) и однородная последовательность некоторых элементов, называемых *ячейками*

*память с произвольным доступом* (Random Access Memory – RAM). Некоторые области памяти поддерживают только чтение информации (по-английски эта память называется Read Only Memory, ROM), данные в такую память записываются один раз при изготовлении этой памяти. Другие области памяти могут допускать запись, но за значительно большее время, чем в обычную память (это так называемая *полупостоянная* память, такой памятью комплектуется популярные в настоящее время карты флэш-памяти)

- Ячейки памяти в машине Фон Неймана нумеруются от нуля до некоторого положительного числа  $N$  ( $N$  часто является степенью двойки, минус единица). *Адресом ячейки* называется её номер. Каждая ячейка состоит из более мелких частей, именуемых *разрядами* и нумеруемых также от нуля и до определённого числа.
- Количество разрядов в ячейке обозначает *разрядность памяти*. Каждый разряд может хранить одну *цифру* в некоторой системе счисления. В большинстве ЭВМ используется двоичная система счисления, т.к. это более выгодно с точки зрения аппаратной реализации.
- В этом случае каждый разряд хранит одну двоичную цифру или один *бит* информации. Восемь бит составляют один *байт*. Сам Фон Нейман тоже был сторонником использования двоичной системы счисления, что позволяло хорошо описывать архитектуру узлов ЭВМ с помощью логических (булевских) выражений.

- Содержимое ячейки называется *машинным словом*. С точки зрения архитектуры, машинное слово – это минимальный объём данных, которым могут обмениваться между собой различные узлы машины по толстым стрелкам на схеме (не надо, однако, забывать о передаче управляющих сигналов по тонким стрелкам). Из каждой ячейки памяти можно считать *копию* машинного слова и передать её в другое устройство компьютера, при этом оригинал не меняется. При записи в память старое содержимое ячейки пропадает и заменяется новым машинным словом.

- Динамическая память(через каждые несколько миллисекунд) приходится восстанавливать содержимое этой памяти.
- Статическая память( по сравнению с динамической, работает быстрее, однако она дороже и требует при реализации больше электронных схем)

# Типичные характеристики памяти современных ЭВМ

1. Объём памяти – от сотен миллионов до нескольких миллиардов ячеек (обычно восьмиразрядных).
2. Скорость работы памяти: это *время доступа* (access time – минимальная задержка на чтение слова из памяти на некоторый регистр) и *время цикла* (cycle time – минимальная задержка на повторное чтение из памяти) – порядка единиц наносекунд (1 секунда =  $10^9$  наносекунд).
  - Заметим, что для упомянутой выше динамической памяти время цикла *больше*, чем время доступа, так как надо ещё восстановить разрушенное при чтении содержимое ячейки.
3. Стоимость. Для основной памяти ЭВМ пока достаточно знать, что чем быстрее такая память, тем она, естественно, дороже.

# Принципы Фон Неймана

2. Принцип неразличимости команд и данных.

# Принципы Фон Неймана

## 3. Принцип хранимой программы

- Одна программа может в качестве результата своей работы поместить в память компьютера другую программу. Именно так и работают *компиляторы*, переводящие программы с одного языка на другой.
- Следствием принципа хранимой программы является то, что программа, может *изменяться* во время счёта самой этой программы. Говорят также, что программа может *самомодифицироваться*

# ***Устройство Управления***

- **Принцип автоматической работы**
- **Принцип последовательного выполнения команд**

# Арифметико-Логическое Устройство

1. Читать содержимое некоторой ячейки памяти (машинное слово), т.е. поместить копию этого машинного слова в некоторую другую ячейку, расположенную в самом АЛУ. Если ячейки памяти расположены не в основной памяти, а в других устройствах ЭВМ, то они называются *регистровой памятью* или просто *регистрами*. Таким образом, АЛУ может читать машинное слово из памяти на один из своих регистров.
2. Записать машинное слово в некоторую ячейку памяти – поместить копию содержимого одного из своих регистров в эту ячейку памяти. Когда не имеет значения, какая операция (чтение или запись) производится, говорят, что происходит *обмен* машинным словом между регистром и основной памятью ЭВМ. Таким образом, машинное слово – это минимальная порция данных для обмена с основной памятью.
3. АЛУ может также выполнять различные *операции* над данными в своих регистрах, например, сложить содержимое двух регистров, обычно называемых регистрами первого R1 и второго R2 операндов, и поместить результат этой операции на третий регистр (называемый сумматором S)

# Взаимодействие УУ и АЛУ

- Революционность идей Джона Фон Неймана заключалась в строгой *специализации*
- Устройство управления тоже имеет свои регистры, оно может считывать команды из памяти на специальный *регистр команд* RK (IR – instruction register), на котором всегда хранится *текущая* выполняемая команда. Регистр УУ с именем RA называется *счётчиком адреса* или *регистром адреса* (в англоязычной литературе его часто обозначают IP – instruction pointer), при выполнении текущей команды в него по определённым правилам записывается адрес *следующей* выполняемой команды

$R1 := \text{ПАМ}[x]; R2 := \text{ПАМ}[y]; S := R1+R2; \text{ПАМ}[z] := S;$

или

$R1 := \langle x \rangle; R2 := \langle y \rangle; S := R1+R2; \langle z \rangle := S;$

Опишем теперь более формально шаги выполнения одной команды в машине Фон Неймана:

1.  $RK := \langle RA \rangle$ ; считать из ячейки памяти с адресом  $RA$  команду на регистр команд  $RK$ ;
2.  $RA := RA+1$ ; увеличить счётчик адреса на единицу;
3. Выполнить очередную команду, хранящуюся в регистре  $RK$ .

Затем по такой же схеме из трёх шагов выполняется следующая команда и т.д.

# Современные ЭВМ

- Современные ЭВМ в той или иной степени нарушают практически все принципы Фон Неймана. Исключение, пожалуй, составляют только принцип автоматической работы, он лежит в самой основе определения ЭВМ как устройства для *автоматической* обработки данных, и принцип хранимой программы.
- Например, существуют компьютеры, которые различают команды и данные. В них каждая ячейка основной памяти кроме собственно машинного слова хранит ещё специальный признак, называемый *тэгом*
- 1 В схеме от АЛУ к УУ тоже ведёт тонкая стрелка, однако она определяет не управляющий сигнал (так как АЛУ не может "командовать" УУ), а информационный, с помощью таких сигналов АЛУ "рапортует" УУ, что заданное действие выполнено, или при его выполнении возникла ошибка.
- (tag), который и определяет, чем является это машинное слово.

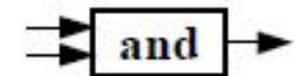
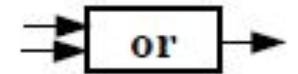
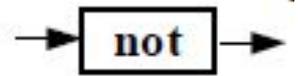
- Первая ЭВМ, построенная на основе принципов Фон Неймана, называлась EDVAC (Electronic Delay Storage Automatic Calculator – автоматический вычислитель с электронной памятью на линиях задержки<sup>4</sup>).
- Компьютер EDVAC был построен в 1949 году в Англии М.Уилксом. EDVAC была одноадресной ЭВМ, которая работала в двоичной системе счисления со скоростью примерно 100 операций в секунду. Заметим, что именно от этой машины принято отсчитывать **первое** поколение ЭВМ (все предшествующие "не совсем настоящие" компьютеры можно условно отнести к нулевому поколению).

# И чуть уровня инженера-конструктора

- Аппаратура современных ЭВМ конструируется из некоторых относительно простых элементов, называемых *вентильями* (по-английски – *circuits*). Каждый вентиль является достаточно простой (электронной) схемой и реализует одну из логических операций, у него есть один или два *входа* (аргументы операции) и один *выход* (результат). На входах и выходе могут быть электрические сигналы двух видов: низкое напряжение (трактруется как ноль или логическое значение **false**) и высокое (ему соответствует единица или логическое значение **true**)

# Основные вентили

1. *Отрицание*, этот вентиль имеет один вход и один выход, (**not** (НЕ) языка Паскаль).
2. *Дизъюнкция* или логическое сложение, **or** (ИЛИ)
3. И, наконец, вентиль, реализующий *конъюнкцию* или логическое умножение, **and** (И)



- Каждый клапан срабатывает (т.е. преобразует входные сигналы в выходные) не непрерывно, а только тогда, когда на этот клапан по специальному управляющему проводу приходит так называемый *тактовый импульс*. Заметим, что по этому принципу работают ЭВМ, которые называются *дискретными*, в отличие от *аналоговых* компьютеров, схемы в которых работают непрерывно (всё время). Подавляющее число современных ЭВМ являются дискретными,

- Из вентиляей строятся так называемые *интегральные схемы* (по-английски *chips*) – это набор вентиляей, соединённых проводами и такими радиотехническими элементами, как сопротивления, конденсаторы и индуктивности, знакомые Вам из курса физики. Каждая интегральная схема тоже имеет свои входы и выходы (их называют внешними *контактами* схемы) и реализует какую-нибудь функцию узла компьютера. Интегральные схемы, которые содержат порядка 1000 вентиляей, называются малыми интегральными схемами (МИС), порядка 10000 вентиляей – средними (СИС), порядка 100000 – большими (БИС), а число вентиляей в сверхбольших интегральных схемах (СБИС) исчисляется уже миллионами

**КОМПЬЮТЕРЫ НАЗЫВАЮТСЯ  
МНОГОЯДЕРНЫМИ, ЕСЛИ НА  
ПЛАСТИНКЕ ЕСТЬ НЕСКОЛЬКО ПОЧТИ  
НЕЗАВИСИМЫХ ЦЕНТРАЛЬНЫХ  
ПРОЦЕССОРОВ**

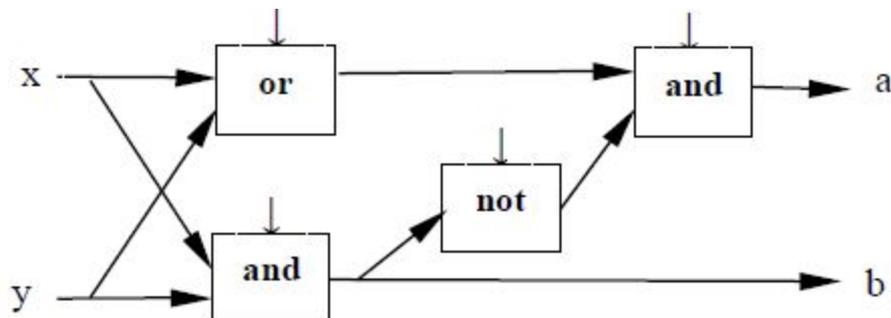
# Пример простой интегральной схемы, которая реализует функцию сложение двух одноразрядных двоичных целых чисел

x	y	b	a
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Легко вычислить, что выходные величины a и b будут определяться такими логическими формулами:

$$a = x \oplus y = (x \text{ or } y) \text{ and not } (x \text{ and } y)$$

$$b = x \text{ and } y$$



Скорость работы интегральной схемы напрямую зависит от частоты прихода тактовых импульсов, называемой *тактовой частотой* схемы. У современных ЭВМ не очень высокой производительности тактовые импульсы приходят на схемы основной *памяти* с частотой несколько сотен миллионов раз в секунду, а на схемы *центрального процессора* – ещё примерно в 10 раз чаще.



Пример ИС двоичного сумматора.

# Вопросы и упражнения

1. Почему машина Фон Неймана является *абстрактной* ЭВМ?
2. В чём заключается принцип линейности и однородности памяти?
3. Объясните разницу между понятиями *ячейка*, *адрес ячейки* и *машинное слово*.
4. Чем отличаются статическая и динамическая память компьютера?
5. Сформулируйте принцип неразличимости команд и данных.
6. Что такое язык машины?
7. Чем отличается регистровая и основная память компьютера?
8. В чём различие между регистром адреса и счётчиком адреса?
9. В чём заключается принцип хранимой программы?
10. Что такое вентиль и интегральная схема?
11. Что такое тактовая частота?