Лекция 1

8 февраля

Цель курса «Архитектура ЭВМ и язык ассемблера»

1. Формирование связного представления об организации современных вычислительных систем.
   - Аппаратура, системные программы, прикладные программы
2. Понимание взаимосвязей между архитектурными решениями на уровнях аппаратуры ЭВМ и ПО.
3. Понимание факторов, влияющих на качественные и количественные характеристики ЭВМ, производительность и безопасность всей вычислительной системы в целом.

- Язык Си как средство демонстрации …
  - того, как работает аппаратура …
  - и того, как следует использовать аппаратуру для реализации языка программирования
- Результат
  - Понимание причин хорошей/плохой производительности программы
  - Понимание причин ошибок
  - Основа для других курсов: ОС, сети, компиляторы, параллельная обработка данных …

Мотивация

- Почему не ставится задача научить программировать на языке ассемблера
  - Промышленное программирование на ассемблере?
  - Не, не слышал ★

```
int x = 5000;
int y = 50000;
...
if (y * y > x * x) {
// попадем сюда?
}
```

- Проблема №1 Свойства чисел
  - Всегда ли свойства математических абстракций выполняются вычислительной техникой?

```
double fun(int i) {
  volatile double d[1] = {3.14};
  volatile long int a[2];
  a[i] = 1073741824;
  return d[0];
}
```

```
fun(0)  →  3.14
fun(1)  →  3.14
fun(2)  →  3.1399998664856
fun(3)  →  2.00000061035156
fun(4)  →  3.14, а потом segmentation fault
```

- Проблема №2 Отладка
  - Где на самом деле хранятся переменные?
  - Как сделать отладчик?
- Проблема №3 Производительность
  - Почему алгоритм Штрассена $O(n^{2.81})$ на практике быстрее алгоритма Копперсмита-Винограда $O(n^{2.3727})$?
  - Может ли стандартное умножение матриц $O(n^3)$ работать быстрее?
- Проблема №4 Безопасность
  - Что на самом деле выполняет компьютер?
  - Можно ли гарантировать, что Си-код который я написал, действительно будет выполняться?

★ Не все так однозначно

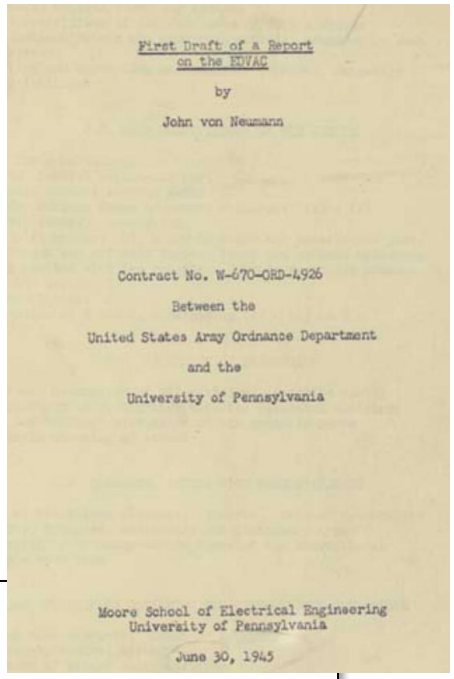
Зачем сейчас нужен язык ассемблера?

- Язык ассемблера позволяет понимать поведение машины

- Системное программное обеспечение создается людьми, в том числе выпускниками ВМК
  - Компиляторы и многие другие инструменты разработки программ
  - Операционные системы
  - Средства защиты информации
- Отладка ошибок разработчиком программы
- Настройка производительности программы
  - Почему оптимизация программы компилятором не дает ожидаемого результата
- Безопасность программного обеспечения (software security, cyber security)
  - Сертификация программ
  - Средства защиты
  - Вредоносный код: ботнеты, вирусы/черви, руткиты, …

История первых компьютеров

- Чарлз Бэббидж, Ада Лавлайс 1820 – 1833
  - Разработка первых счетных машин
- Табуляторы – электромеханические машины для суммирования и категоризации, применялись с конца XIX века.
  - Tabulating Machine Company → IBM
- Конрад фон Цузе – Германия
  - Z1 1938г., Z2 1939 г., Z3 1941г., Z4 1945г.
- Экерт и Моучли – США
  - ENIAC 1946 г.
  - фон Нейман
    - First Draft of a Report on the EDVAC, 30 июня 1945
- Атанасов и Берри – США
  - ABC 1942 г.
- Mark I – Великобритания
  - 1949 г.
- Исаак Брук – СССР, Москва
  - М-1 1951 г.
- Сергей Лебедев – СССР, Киев
  - МЭСМ 1950 г.
  - БЭСМ-1 1952 г.



First Draft of a Report
on the EDVAC

by

John von Neumann

Contract No. W-670-ORD-4926

Between the

United States Army Ordnance Department

and the

University of Pennsylvania

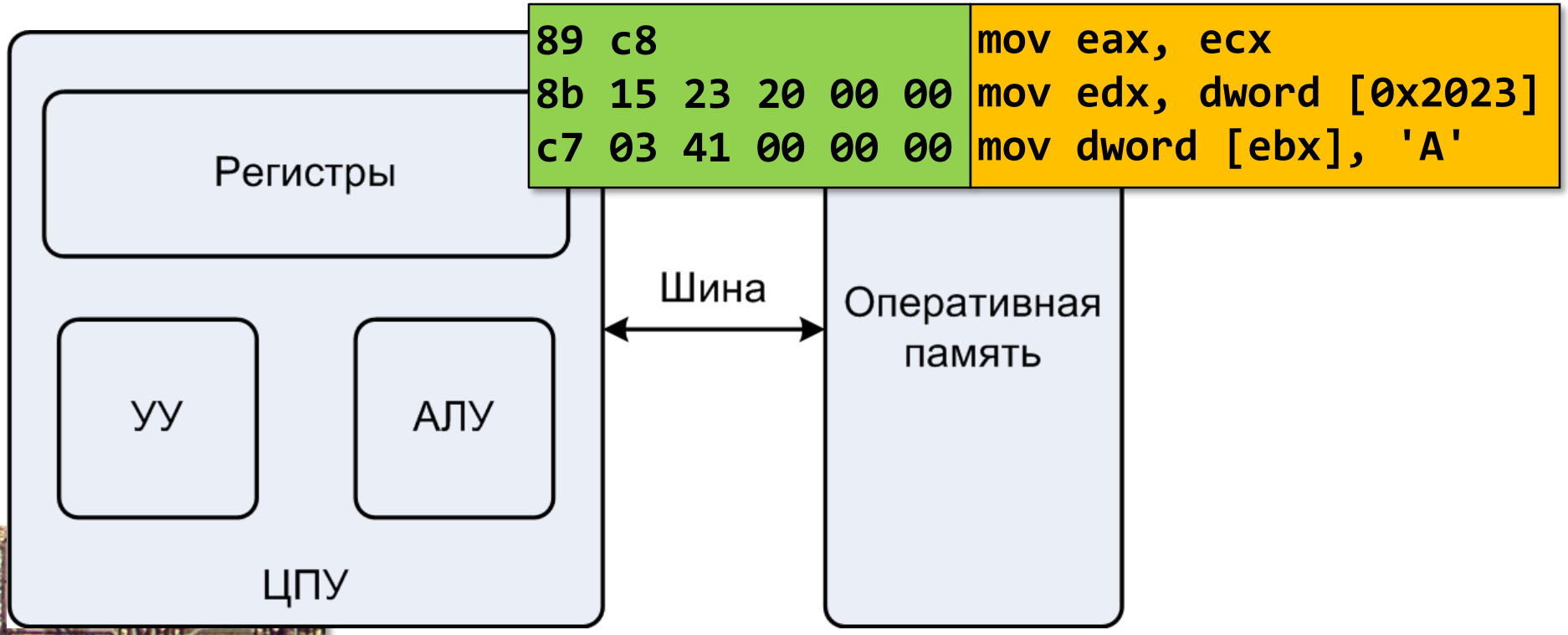
Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

Применение первых компьютеров
- задачи математической физики
- шифрование/дешифрование сообщений
- управление (военной) техникой

Современное применение компьютеров
- хранение информации и предоставление удобного доступа для работы с ней
- игры!!!
- передача информации
- управление технологическими процессами
- работа во всем, что нас окружает …

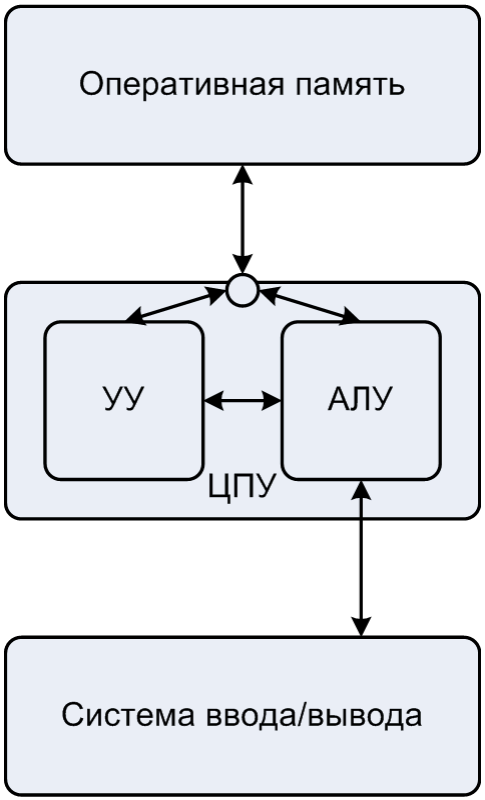
Архитектура фон Неймана



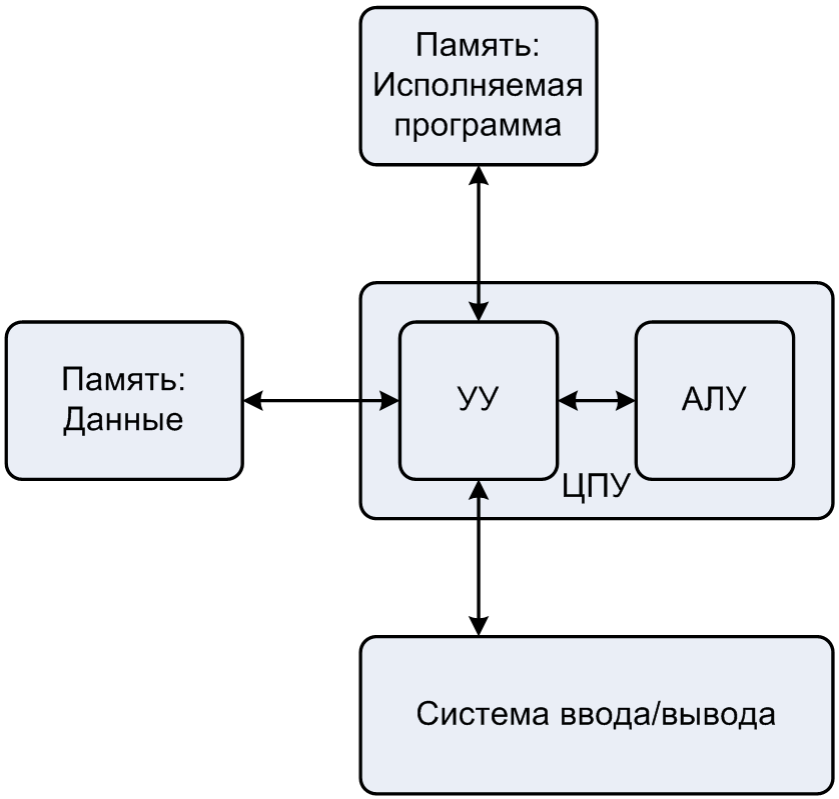
1. Двоичное кодирование информации
2. Неразличимость команд и данных
3. Адресуемость памяти
4. Последовательное выполнение команд

Вспоминаем Си-машину…

Единственный вариант архитектуры?



Еще один вариант изображения архитектуры фон Неймана…



Гарвардская архитектура

Вспоминаем Си-машину…

(Модельный) цикл работы ЭВМ

1. **Извлечение инструкции из памяти**
   Используя текущее значение *счетчика команд,* процессор извлекает некоторое количество байт из памяти и помещает их в *буфер команд*
2. **Декодирование команды**
   Процессор просматривает содержимое *буфера команд* и определяет *код операции* и ее *операнды*
   Длина декодированной команды прибавляется к текущему значению *счетчика команд*
3. **Загрузка операндов**
   Извлекаются значения операндов
   Если операнд размещен в ячейках памяти – вычисляется *исполнительный адрес*
4. **Выполнение операции над данными**
5. **Запись результата**
   Результат может быть записан в том числе и в *счетчик команд* для изменения естественного порядка выполнения

Промежуточные итоги: ключевые термины и понятия

- Регистр, имя регистра, значение регистра, разрядность регистра, машинное слово
- Память, ячейка памяти, адрес ячейки памяти, время доступа к памяти
- Цикл, тактовая частота, счетчик команд
- Машинная команда, код операции, операнд, адресность команды
  Исполнительный (действительный) адрес
  англ. *effective (executive) address*
- Способ адресации
  прямая, непосредственная, косвенная