Машинно-зависимые языки программирования

Лекция 1

Краткая история компьютеров

Изначально слово компьютер означало не устройство, а профессию. Первоначально это были люди, которые решали какие-то сложные задачи (типа для карт для ориентации по звездам). При усложнении задач постепенно начали появляться вычислительные машины (арифмометры и тд).

Эти первые машины были механическими. К началу 20 века научились делать электрические (выполняли действия быстрее). Но и у тех и у других была проблема с тем, что они решали только одну конкретную задачу. Поэтому в середине 20 века назрел вопрос о создании универсальной машины, которая могла бы выполнять произвольные алгоритмы без перестраивания машины.

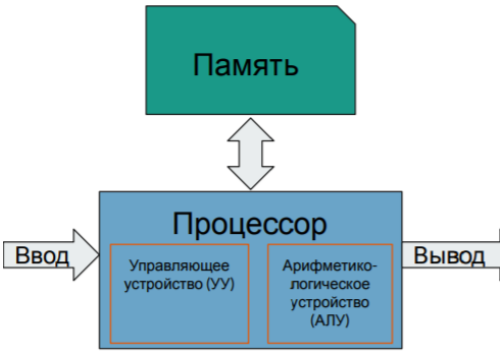
Американский ученый фон Нейман работал над этой проблемой. Ему удалось сформулировать некие основные принципы.

Принципы фон Неймана:

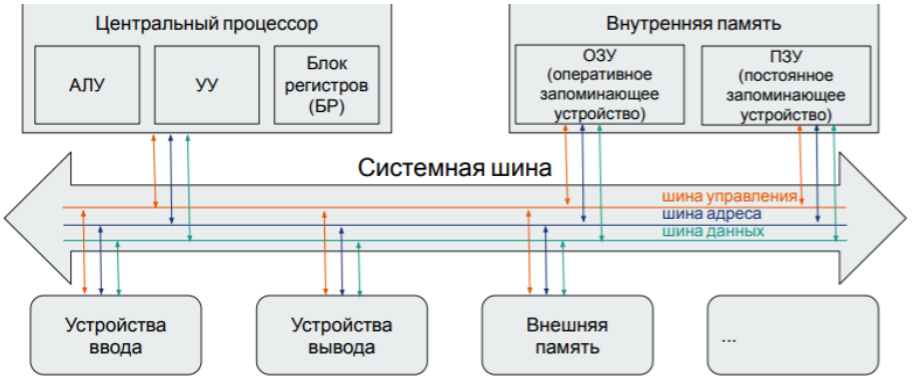
1. Использование двоичной системы счисления
2. Программное управление ЭВМ
3. Принцип однородности памяти (память используется не только для хранения данных, но и для хранения программ)
4. Принцип адресности (все ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы)
5. Возможность условного перехода в процессе выполнения программы (чтобы можно было создавать не только линейные алгоритмы)

Схема архитектуры машины фон Неймана:

- Абстрактный процессор – вычислитель
    - Управляющее устройство (интерпретирует команды и их выполняет)
    - Арифметико-логическое устройство (выполняет арифметические и некоторые логические бинарные операции над данными)
- Абстрактная память
  (из памяти процессор берет значения, код, и что-то туда пишет)
- Также обозначены абстрактный ввод и вывод



Структурная схема ЭВМ:



- Центральным устройством является системная шина (это некоторая абстракция, которая обеспечивает взаимодействие всех устройств)
    - шина управления
    - шина адреса
    - шина данных

  Все три составляющие используются для любого обмена. Например, если процессору необходимо считать какое-то значение из памяти, то он выставляет на шину адреса номер ячейки памяти и по шине управления отправляет сигнал. Память считывает значение с шины адреса, ищет нужную ячейку, выставляет ее значение на шину данных. Данные получены.

  В компьютере функции системной шины по сути выполняет материнская плата. Также на практике шин обычно несколько (быстрая между процессором и оперативкой, шины для подключения внешних устройств [pci, usb])

- Центральный процессор
    - Блок регистров
      (регистр – внутренняя ячейка памяти процессора. Обычно таких ячеек не очень много, десяток-полтора ячеек из нескольких байтов)
    - АЛУ
    - УУ

  Процессор много взаимодействует с оперативкой. В частности, во время работы программа должна быть загружена в оперативную память. Это единственная память, к которой процессор имеет прямой доступ.

- Внутренняя память
    - ОЗУ (RAM) // очищается при выключении компьютера
    - ПЗУ // нужна для хранения данных

  В ПЗУ также хранится стартовая программа для загрузки компьютера.
  (как это работает: при включении процессор всегда начинает работать с фиксированного адреса. Производители материнских плат этим пользуются и настраивают ПЗУ так, чтобы данные из ПЗУ копировались в ОЗУ по нужным адресам, таким образом добиваются того, что при включении процессор начинает читать именно стартовую программу (bios))

  BIOS:
    1. Выполняет первичную диагностику устройства
    2. Определяет какие устройства подключены
    3. Находит внешнюю память, находит загрузочный диск
    4. Загружает операционную систему

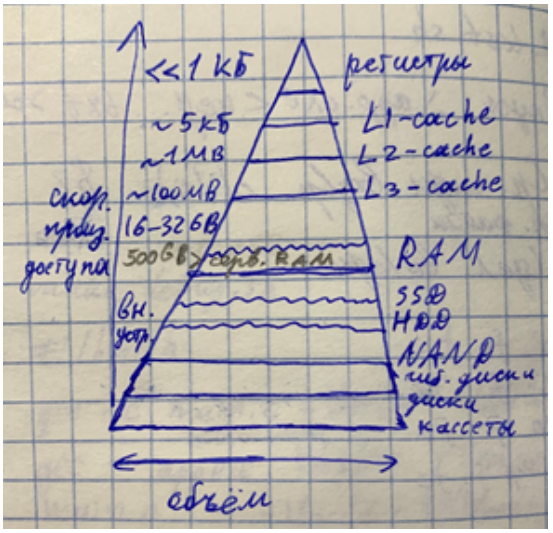
- Устройства ввода (клавиатура, мышка и тд)
- Устройства вывода (монитор, принтер и тд)
- Внешняя память (все возможные виды накопителей – дискеты, магнитные ленты, жесткие диски, оптические диски, флешки, SSD-диски)

Память

Почему так много разных видов памяти?
Чем более быстродействующая память, тем она дороже стоит => тем меньше ее можно вставить.

Пирамидка быстродействия:

- регистры процессора
- кэш-память процессора (несколько уровней)
- оперативная память
- внешние



Минимальная адресуемая единица памяти[1] – байт

Побитово работа с памятью никогда не осуществляется (это нужно для сокращения адресных линий)

(доп: байт очень удобно представлять в 16-ричной системе. В этом случае он будет выглядеть как ровно две 16-ричные цифры)

Машинное слово[2] – машинно-зависимая величина, измеряемая в битах, равна разрядности регистров процессора и шины данных.

16-ти разрядный процессор => размер его регистров 16 бит = 2 байта => шина данных такая же

Параграф – 16 байт

ASCII – однобайтовая символьная таблица (7-битная кодировка)
Первые 32 байта такой таблицы занимают различные непечатные символы.

---

[1] Минимальная адресуемая единица памяти по костру – минимальная разница между двумя адресами в памяти

[2] Машинное слово по костру – сообщение фиксированной длины, с помощью которых компьютер общается со своими составными частями

Системы счисления

Двоичная (binary)

- 0, 1, 10, 11, 100, 101…
- $2^8 = 256$
- $2^{10} = 1024$
- $2^{16} = 65536$
- Суффикс - b. Пример: 1101b

Шестнадцатеричная (hexadecimal)

- 0, 1, …, 8, 9, A, B, C, D, E, F, 10, 11, 12, ….., 19, 1A, 1B, …
- $2^4 = 10_{16}$
- $2^8 = 100_{16}$
- $2^{16} = 10000_{16}$
- Суффикс - h (10h - 16). Некоторые компиляторы требуют префикса 0x (0x10)

$$1011011011111000_2 = B6F8_{16}$$

Так как первое время мы будем изучать 16-ти разрядный процессор, то нам важно понимать, что 2^16 = 64 Кбайт.

Представление отрицательных чисел

Знак - в старшем разряде (0 - “+”, 1 - “-”).

Возможные способы:

- прямой код
- обратный код (инверсия)
- **дополнительный код (инверсия и прибавление единицы)**

Примеры доп. кода на 8-разрядной сетке

-1:

1. 00000001
2. 11111110
3. 11111111

Смысл: -1 + 1 = 0 (хоть и с переполнением):

11111111 + 1 = (1)00000000

-101101:

1. 00101101
2. 11010010
3. 11010011

Плюс хранения в дополнительном коде – в плане арифметических выражений ничего не меняется.

Виды современных архитектур ЭВМ

- x86-64 8086 (16-разр.) ⟹ 86 (32-разр.) ⟹ 6-64 (64-разр.)
- ARM
- IA64
- MIPS (включая Байкал)
- Эльбрус

Семейство процессоров x86 и x86-64

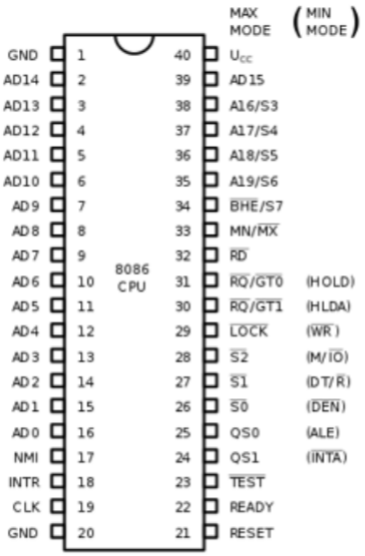
- Микропроцессор 8086: 16-разрядный, 1978 г., 5-10 МГц, 3000 нм
- Предшественники: 4004 - 4-битный, 1971 г.; 8008 - 8-битный, 1972 г.; 8080 - 1974 г.
- Требует микросхем поддержки
- 80186 - 1982 г., добавлено несколько команд, интегрированы микросхемы поддержки
- 80286 - 1982 г., 16-разрядный, добавлен защищённый режим
- 80386, 80486, Pentium, Celeron, AMD, ... - 32-разрядные, повышение быстродействия и расширение системы команд
- x86-64 (x64) - семейство с 64-разрядной архитектурой
- Отечественный аналог - К1810ВМ86, 1985 г.

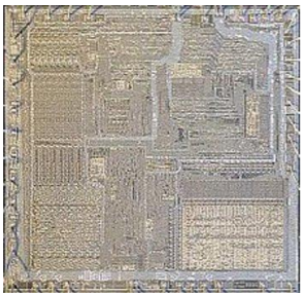
Устройство 8086

Если с процессора 8086 снять корпус, то внутри окажется небольшой квадратик.

// сложная система, переслушивайте сами

A – шина адреса (здесь 20 => нам доступно 2^20 адресных ячеек)
D – шина данных (их количество определяет разрядность, здесь 16)





Структура блока регистров (архитектура 8086 с точки зрения программиста)

1. IP – указатель команд (хранит смещение команды, которая будет выполнена следующей)
2. Регистры общего назначения – используются в арифметических, логический и тд операциях (для большинства операций являются взаимозаменяемыми. Базово AX – умножение и деление, BX – вычисление адресов, CX – циклы, DX – обмен с внешними устройствами)
3. Регистры в центре близки к регистрам общего назначения
4. Сегментные регистры (отвечают за сегменты памяти [вторая составл адреса])

Язык ассемблера

Машинная команда – инструкция (в двоичном коде) из аппаратного определенного набора, которую способен выполнять процессор.

Машинный код – система команд конкретной вычислительной машины, которая интерпретируется непосредственно процессором.

Язык ассемблера – машинно-зависимый язык программирования низкого уровня, команды которого прямо соответствуют машинным командам.

Чтобы получить программу в машинном коде необходимо ее из исходного кода скомпилировать и получить исполняемый файл.

Исполняемый файл – файл, содержащий программу в виде, в котором она может быть исполнена компьютером (то есть в машинном коде).

В dos и windows расширения у исполняемых файлов exe и com

Компилятор – программа для преобразования исходного текста программы в объектный модуль.

Компоновщик – программа для связывания нескольких объектный файлов в исполняемый.

Последовательность запуска программы операционной системой:

- Определение формата файла
- Чтение и разбор заголовка
- Считывание разделов исполняемого модуля в ОЗУ по необходимым адресам
- Подготовка к запуску (загрузка библиотек)
- Передача управления на точку входа

Отладчик – программа для автоматизации процесса отладки. Может выполнять трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода, устанавливать или удалять контрольные точки или условия остановки.

.COM – простейший формат исполняемого файла в dos и ранней винде.
Особенности:

- не имеет заголовка
- состоит из одной секции, не превышающей 64Кб
- загружается в ОЗУ без изменений
- начинает выполняться с 1-го байта (точка входа всегда в начале)

Последовательность запуска COM программы:

1. Система выделяет свободный сегмент памяти нужного размера и заносит его адрес во все сегментные регистры (CS, DS, ES, FS, GS, SS)
2. В первые 256 (100h) байт этого сегмента записывает служебная структура DOS, описывающая программу – PSP (префикс программного обеспечения)
3. Непосредственно за ним загружается содержимое COM-файла без изменений
4. Указатель стека (регистр SP) устанавливается на конец сегмента.
5. В стек записывается 0000h (начало PSP – адрес возврата для возможности завершения командой ret)
6. Управление передается по адресу CS:0100h.

Классификация команд процессора 8086:

- Команды пересылки данных
- Арифметические и логические команды
- Команды переходов
- Команды работы с подпрограммами
- Команды управления процессором

Команды пересылки данных $MOV\ <\text{приемник}>,\ <\text{источник}>$
(типа оператор присваивания)

Источник: непосредственный операнд (константа, включенная в машинный код) / РОН (?) / регистр общего назначения / сегментный регистр / переменная
Приемник: все то же самое, кроме константы

```
MOV AX, 5
MOV BX, DX
MOV [1234h], CH
MOV DS, AX
```

Ограничения:

- Нельзя скопировать переменную в переменную
- В сегментные регистры напрямую нельзя записывать константы (записать можно только из регистра общего назначения)

Целочисленная арифметика

- ADD <приемник>, <источник> – арифметическое сложение приемника и источника. Сумма помещается в приемник, источник не меняется
- SUB <приемник>, <источник> – арифметическое вычитание источника из приемника
- MUL <источник> – беззнаковое умножение. Умножаются источник и AL/AX, в зависимости от размера источника. Результат помещается в AX либо DX:AX.
- DIV <источник> – целочисленное беззнаковое деление. Делится AL/AX на источник. Результат помещается в AL/AX, остаток – в AH/DX.
- INC <приемник> – инкремент на 1
- DEC <приемник> – декремент на 1

Побитовая арифметика

- AND <приемник>, <источник> – побитовое И (AND al, 00001111b)
- OR <приемник>, <источник> – побитовое ИЛИ
- XOR <приемник>, <источник> – побитовое исключающее ИЛИ (XOR AX, AX)
- NOT <приемник> – инверсия

Команда безусловной передачи управления JMP $JMP\ <\text{операнд}>$

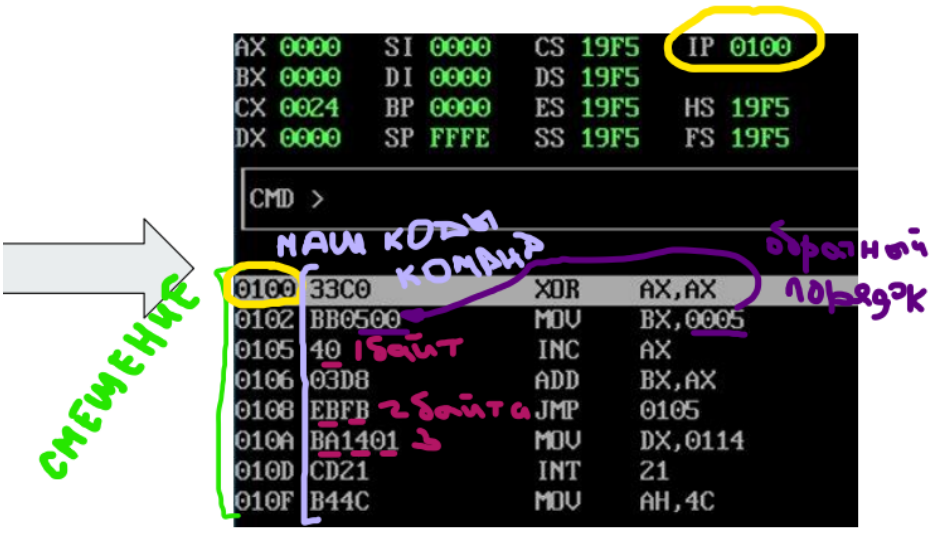
- Передает управление в другую точку программы (на другой адрес памяти), не сохраняя какой-либо информации для возврата.
- Операнд – непосредственный адрес (вставленный в машинный код), адрес в регистре или адрес в переменной.

Команда, которая ничего не делает NOP

- Занимает место и время
- Размер – 1 байт, код – 90h
- Назначение – задержка выполнения либо заполнения памяти, например, для выравнивания

Пример

```
...
    XOR AX, AX
    MOV BX, 5
label1:
    INC AX
    ADD BX, AX
    JMP label 1
```



Взаимодействие программы с внешней средой

Прерывания – аппаратный механизм для приостановки выполнения текущей программы и передачи управления специальной программе – обработчику прерывания.

Основные виды:
- аппаратные
- программные

Вызов – int <номер>

21h – прерывание DOS. Передается через регистр AH. Параметры функций передаются собственым способом, он описан в документации.

Память в реальном режиме работы процессора

Реальный режим работы - режим совместимости современных процессоров с 8086.

Доступен 1 Мб памяти (2^{20} байт), то есть разрядность шины адреса - **20 разрядов**.

Физический адрес получается **сложением** адреса **начала сегмента** (на основе сегментного регистра) и **смещения**.

Сегментный регистр хранит в себе **старшие 16 разрядов** (из 20) адреса начала сегмента. 4 младших разряда в адресе начала сегмента всегда нулевые. Говорят, что сегментный регистр содержит в себе **номер параграфа начала сегмента**.

Память в реальном режиме работы процессора - пример

Номер параграфа начала сегмента

19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Смещение

[SEG]:[OFFSET] => физический адрес:
1. SEG необходимо побитово сдвинуть на 4 разряда влево (или умножить на 16, что тождественно)
2. К результату прибавить OFFSET

5678h:1234h =>
56780
+1234
579B4

Вычисление физического адреса выполняется процессором аппаратно, без участия программиста.
Распространённые пары регистров: CS:IP, DS:BX, SS:SP

Структура памяти программы. Виды сегментов. Назначение отдельных сегментных регистров

- Сегмент кода - регистр CS. Командой MOV изменить невозможно, меняется автоматически по мере выполнения команд.
- Сегмент данных. Основной регистр - DS, при необходимости дополнительных сегментов данных задействуются ES, FS, GS.
- Сегмент стека - регистр SS