

Лекция 1-2  
Архитектура управляющего  
устройства

# Содержание

- Понятие архитектуры микроконтроллера
- Понятие машинного кода
- Компиляция машинного кода
- Представление процессора
- Структура ассемблера

# Архитектура

- Архитектура компьютера — это модель компьютерной системы, воплощённая в её *компонентах, их взаимодействии* между собой и окружением, включающая также принципы её проектирования и развития. Аспекты *реализации* (например, технология, применяемая при реализации памяти) не являются частью архитектуры

# Уровни архитектуры

- 5 — языки высокого уровня
- 4 — прикладные программы
- 3 — ОС (драйвера, файловая система)
- 2 — система команд
- 1 — машинно-зависимое исполнение команд
- 0 — аппаратное обеспечение

# Архитектура по типу процессора

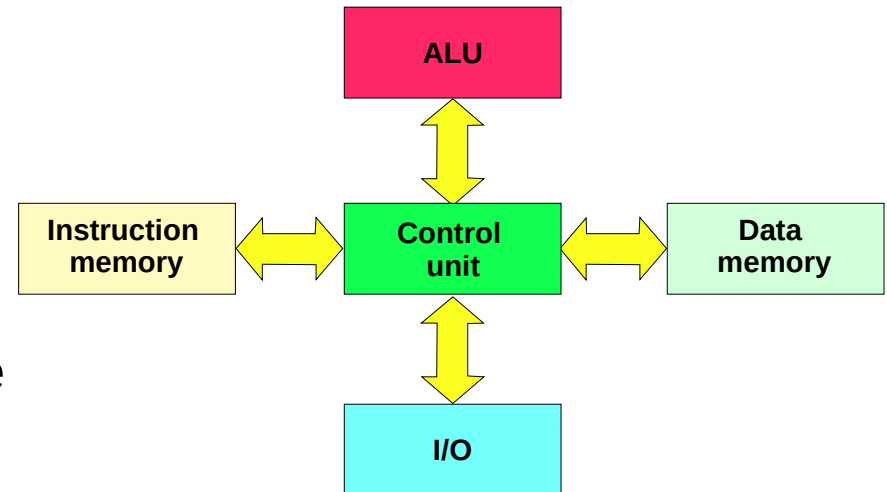
- CISC (complex instruction set computing) — архитектура с полным набором команд
- RISC (reduced instruction set computing) — сокращенный набор команд
- MISC (minimal instruction set computing) — минимальный набор команд
- VLIW (very long instruction word) — очень длинная машинная команда

# Архитектура по разделению памяти

Гарвардская архитектура — архитектура ЭВМ, отличительными признаками которой являются:

- хранилище инструкций и хранилище данных представляют собой разные физические устройства
- канал инструкций и канал данных также физически разделены

Архитектура фон Неймана — принцип построения ЭВМ, когда данные и программа объединены в одном массиве памяти



# Машинный код

Машинный код (платформенно-ориентированный код), машинный язык — система команд (набор кодов операций) конкретной вычислительной машины, которая исполняется непосредственно процессором или микропрограммами этой вычислительной машины.

# Компиляция

Компилятор — программа, переводящая написанный на языке программирования текст в набор машинных кодов для непосредственного выполнения процессором

Процесс компиляции:

- Трансляция — преобразование исходных языков программирования в целевые (близкие к) машинные коды
- Сборка — объединение полученных элементов в единую программу



# Представление процессора

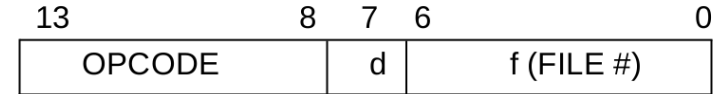
Ключевые компоненты процессора

- Аккумулятор  $W$  — текущий регистр, с которым выполняются все операции
- Счетчик команд  $PC$  — указатель на текущую инструкцию в памяти программ
- Регистр состояний  $STATUS$  — состояние процессора после выполнения последней команды

# Ассемблер Microchip

- Инструкция — 14-битное слово, разделенное на инструкцию и операнд
- Байтовые инструкции работают с целым регистром
- Битовые инструкции оперируют только одним битом в байте
- Литеральные и управляющие инструкции оперируют константами и переходами

## Byte-oriented file register operations

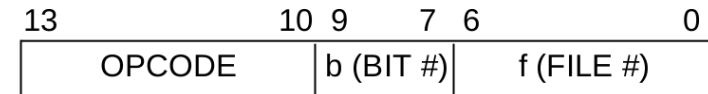


d = 0 for destination W

d = 1 for destination f

f = 7-bit file register address

## Bit-oriented file register operations

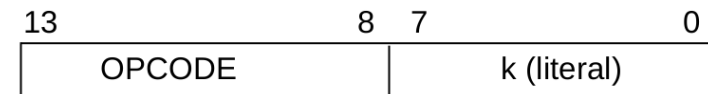


b = 3-bit bit address

f = 7-bit file register address

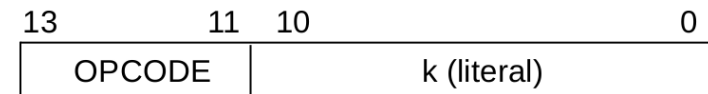
## Literal and control operations

### General



k = 8-bit immediate value

### CALL and GOTO instructions only



k = 11-bit immediate value

# Некоторые команды ассемблера

**bsf** *R, B* – установить бит *B* в регистре *R* в 1

**bcf** *R, B* – установить бит *B* в регистре *R* в 0

**btfsc** *R, B* – проверить бит *B* в регистре *R*, если 0 – пропустить следующую операцию

**btfss** *R, B* – проверить бит *B* в регистре *R*, если 1 – пропустить следующую операцию

**goto** *L* – перейти к метке *L*

**incfsz** *R, D* – увеличить значение в регистре *R* на единицу, сохранив результат в *D*; если результат 0, то пропустить следующую операцию

**call** *L* – вызвать подпрограмму по метке *L*

**return** – вернуться из подпрограммы

# Пример программы

```
init:
    bsf STATUS, RP0
    bcf TRISA, 4
    bcf STATUS, RP0

start:
    btfsc PORTA, 3
    goto end
count:
    incfsz COUNTER, F
    goto end
    call switch
end:
    goto start
```

```
switch:
    btfss PORTA, 4
    goto turn_on
turn_off:
    bcf PORTA, 4
    goto switch_end
turn_on:
    bsf PORTA, 4
switch_end:
    return
```

# Заключение

- Основная архитектура микроконтроллеров — гарвардская / RISC
- Все языки для микроконтроллеров являются компилируемыми из ассемблера или более высоких уровней