

# Аппаратное обеспечение IoT/CPS

## Лекция 8

А. А. Подшивалов

[apodshivalov@miem.hse.ru](mailto:apodshivalov@miem.hse.ru)

Память микроконтроллера

# Несколько общих слов

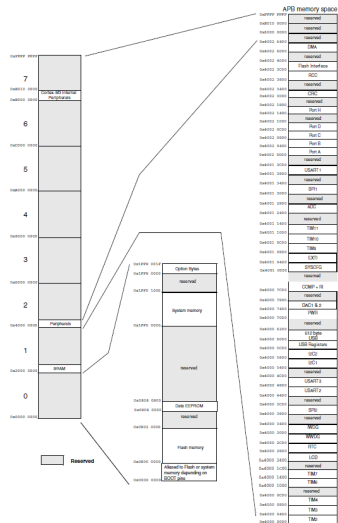
- ▶ Гарвардская архитектура — отдельная память программ и данных
- ▶ Фон-неймановская архитектура — общая память программ и данных
- ▶ Ввод-вывод с отображением на память (memory-mapped I/O)

# Организация памяти МК

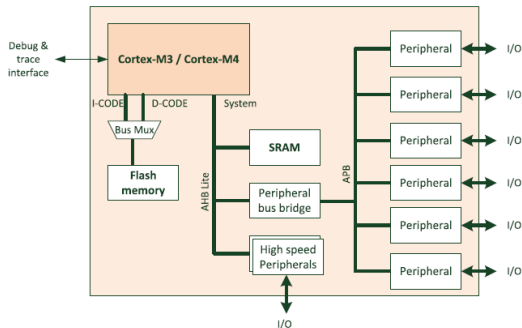
- ▶ PIC10/12/16/18
  - ▶ Типичный пример гарвардской архитектуры
  - ▶ Раздельные ROM и RAM
  - ▶ Special Function Registers в адресном пространстве RAM
- ▶ 8051
  - ▶ IRAM — внутренняя RAM, 128 или 256 байт, 8-битная адресация
  - ▶ SFR в адресном пространстве IRAM
  - ▶ XRAM — до 64 кБайт (16-битная адресация)
  - ▶ PMEM — память программ, до 64 кБайт (или больше с переключаемыми страницами)

# ARM Cortex-M

- ▶ Общий размер адресного пространства у 32-битного ядра — 4 Гб
- ▶ В единое адресное пространство проецируются:
  - ▶ ROM (встроенная Flash-память)
  - ▶ RAM
  - ▶ Регистры периферии
- ▶ С точки зрения программиста — типичнейший пример фон-неймановской архитектуры



# ARM Cortex-M

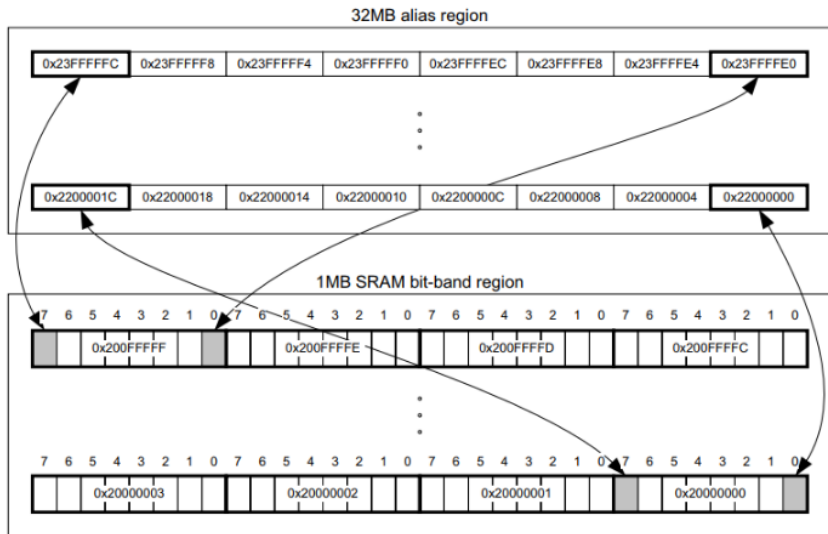


- ▶ «Внутри» ядро Cortex-M имеет гарвардскую архитектуру памяти
- ▶ Это позволяет одновременно с загрузкой инструкции из ROM загружать данные из RAM
- ▶ Выполнение кода из RAM возможно с замедлением примерно в 2 раза
- ▶ Для встроенной периферии используются шины АНВ (AMBA High-performance Bus) или АРВ (AMBA Peripheral Bus), описанные в стандарте AMBA (Advanced Microcontroller Bus Architecture)

# DMA — контроллер прямого доступа к памяти

- ▶ «Асинхронный метсру»
- ▶ Может копировать данные с одного адреса в памяти на другой без участия процессорного ядра
- ▶ Автоинкремент адресов источника и приемника (настраивается)
- ▶ Может управляться от периферии (UART, SPI, I<sup>2</sup>C, ADC, ...)
- ▶ Счетчик количества операций
- ▶ При достижении середины и конца буфера генерирует прерывание

# ARM Cortex-M: bit-banding





# Виды памяти

# Встроенная память МК

- ▶ RAM

- ▶ Static RAM — дорогая (6 транзисторов/бит), но не требует «регенерации» и имеет низкое энергопотребление

- ▶ ROM

- ▶ Flash
  - ▶ EEPROM

# Flash-память

- ▶ Чтение происходит «строками» по 32, 64 или 128 бит, довольно медленное
  - ▶ Контролер памяти скрывает это от программиста
  - ▶ Между ядром процессора и flash часто находится «ускоритель flash» — небольшой объем кеш-памяти
- ▶ Возможна только запись 0 на место 1
  - ▶ Стирание памяти (запись 1 во все биты) производится «страницами», типичный размер — от 256 байт до 64 кБайт
  - ▶ Ресурс выражается в циклах стирание-запись, обычно около 10 000 циклов
- ▶ Контролер flash-памяти — отдельное периферийное устройство, позволяет стирать flash и записывать туда данные
- ▶ Типичный объем — десятки-сотни кБайт (256 кБ в STM32L151CC)

# EEPROM

- ▶ Возможно побайтовое чтение и запись
- ▶ Существенно больший ресурс, чем у flash, 100 000 – 1 000 000 циклов
- ▶ Типичный объем — единицы кБайт (8 кБ в STM32L151CC)

# FSMC — Flexible Static Memory Controller

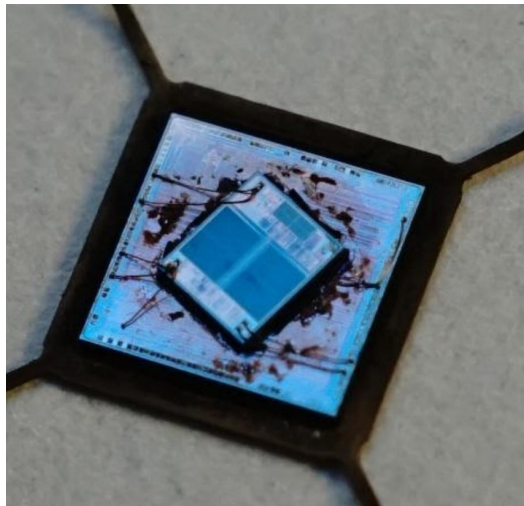
- ▶ Внешняя память проецируется в адресное пространство микроконтроллера (адреса с 0x60000000 по 0x9FFFFFFF)
- ▶ Шина адреса до 26 бит
- ▶ Шина данных до 16 бит
- ▶ «Служебные» сигналы read strobe, write strobe, chip select
- ▶ Можно подключать не только память, но и все, имеющее схожий интерфейс (например, ЖК-дисплеи с параллельной шиной)

# Экзотика

- ▶ TI CC3200
  - ▶ Встроенная флеш-память очень мала и программисту недоступна
  - ▶ Программа хранится во внешней микросхеме памяти с последовательным интерфейсом QSPI, при запуске МК считывается оттуда в оперативную память
- ▶ TI CC3220SF
  - ▶ 1 Мб встроенной флеш-памяти, XIP (eXecute In Place)
  - ▶ Ускоритель не работает, память очень медленная, производительность в 1,5 раза ниже предыдущей модели
  - ▶ Внешняя микросхема памяти все равно нужна

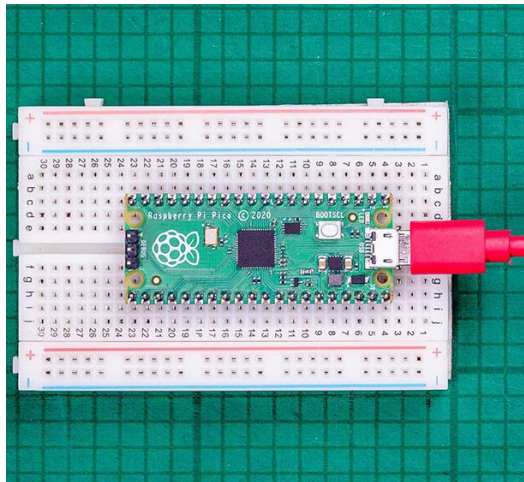
## Экзотика-2

- ▶ GigaDevice — китайский производитель микросхем флеш-памяти и микроконтроллеров
- ▶ GD32 — клоны STM32 на ядре Cortex-M
- ▶ GD32V — собственные разработки на базе RISC-V
- ▶ В одном корпусе собраны микроконтроллер с периферией и отдельный чип flash-памяти с последовательным интерфейсом



# Экзотика-3

- ▶ RP2040 — попытка Raspberry Pi Foundation сделать микроконтроллер
- ▶ Встроенная flash-память отсутствует, программа загружается в SRAM из внешней микросхемы памяти с интерфейсом QSPI





С точки зрения программиста

\*.ld и \*.map

- ▶ Linker script — файл с расширением `ld`
  - ▶ Описывает расположение данных в памяти
  - ▶ Обычно секция `.text` и копия `.data` помещаются в ROM
  - ▶ Задача обработчика сброса — обнулить `.bss` и скопировать `.data` в RAM
- ▶ Map file — файл с расширением `map`

# Bootloader

- ▶ Выполняющаяся на микроконтроллере программа для записи «прошивки» в собственный Flash
- ▶ Очень часто bootloader «зашит» в микроконтроллер уже при изготовлении, и выбор программы для запуска (bootloader либо штатная прошивка) осуществляется переключением логических уровней на нескольких специально выделенных выводах
- ▶ Bootloader в STM32 умеет работать с UART и USB (по протоколу DFU, Device Firmware Update)

# Постоянная память и RIOT OS

- ▶ Драйверы Flash и EEPROM
- ▶ Имитация EEPROM в Flash-памяти (для некоторых семейств процессоров)
- ▶ В EEPROM/Flash можно сохранять настройки устройства, сессионные ключи, идентификаторы, ...

`drivers/include/periph/eeprom.h`

# Файловые системы для Flash

- ▶ Когда flash-памяти много, можно организовать в неиспользуемой ее части полноценную файловую систему
- ▶ Основная проблема — ограниченное количество циклов перезаписи
- ▶ Решение — стирать данные только при «переполнении» ФС, поддерживать версии файлов

## И еще про внешние запоминающие устройства

- ▶ «Флешки» с I<sup>2</sup>C и SPI, объем до нескольких Мбит
- ▶ SD-карты — 1-битный интерфейс, похожий на SPI; модуль `sdcard_spi` и пакет `fatfs` в RIoT

