

Лекция 4 Ядро ARM Cortex-M3.

Микроконтроллер Миландр 1986BE92QI

План курса «Встраиваемые микропроцессорные системы»:

Лекция 1: Введение. Язык программирования C

Лекция 2: Язык программирования C, применение для встраиваемых систем

Лекция 3: Стандартная библиотека языка C

Лекция 4: Ядро ARM Cortex-M3. Микроконтроллер Миландр K1986BE92QI

Лекция 5: Этапы разработки микропроцессорных систем

Лекция 6: Разработка и отладка программ для встраиваемых систем

Лекция 7: Архитектура программного обеспечения

Лекция 8: Периферийные модули: Timer, DMA, ADC, DAC

Лекция 9: Периферийные модули: CAN, USB, Ethernet, SDIO

Архитектура центрального процессора

Архитектура с точки зрения разработчика процессора (микро-архитектура):

- Внутренние шины;
- Количество стадий конвейера;
- Время исполнения команд;
- и т.д.

Знание микро-архитектуры нужно разработчикам микроконтроллеров и разработчиками компиляторов.

Архитектура с точки зрения программиста:

- Набор команд (то, из чего состоят программы);
- Программно-логическая модель (набор регистров и карта памяти)
(то, что видят программы);
- Методы отладки (то, что видят отладчики);

Знание архитектуры нужно программистам на ассемблере и частично программистам на С.

Архитектура ARM Cortex-M3

Все ARM Cortex-M являются 32 разрядным RISC (Reduced Instruction Set Computing) процессорами:

- 32 разрядные регистры;
- 32 разрядные внутренние магистрали;
- 32 разрядные шинные интерфейсы.

Микро-архитектуры ARM Cortex-M:

Cortex-M0, Cortex-M0+, Cortex-M1
Cortex-M3, Cortex-M4, Cortex-M7,
Cortex-M23, Cortex-M33

Микро-архитектура **ARM Cortex-M3**, в документах также называется **ARMv7-M**.

Особенности ARM Cortex-M3 (ARMv7-M):

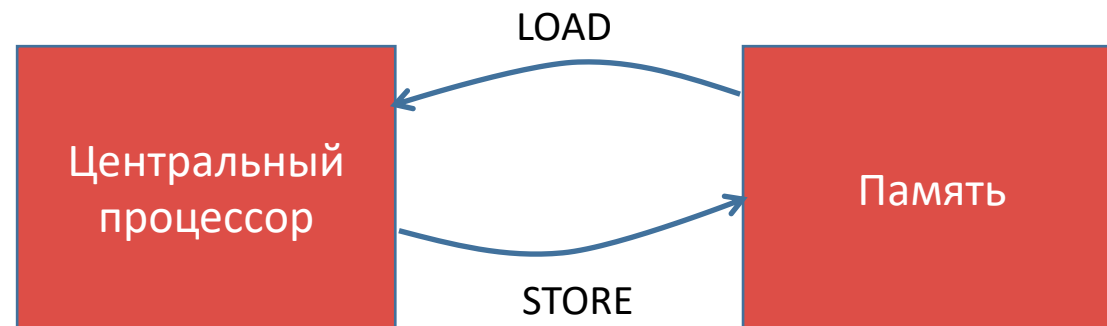
- 3 ступенчатый конвейер (Fetch, Decode, Execute);
- набор команд Thumb-2 (совместное использование 16-ти разрядных и 32-разрядных команд).

Архитектура Load-Store

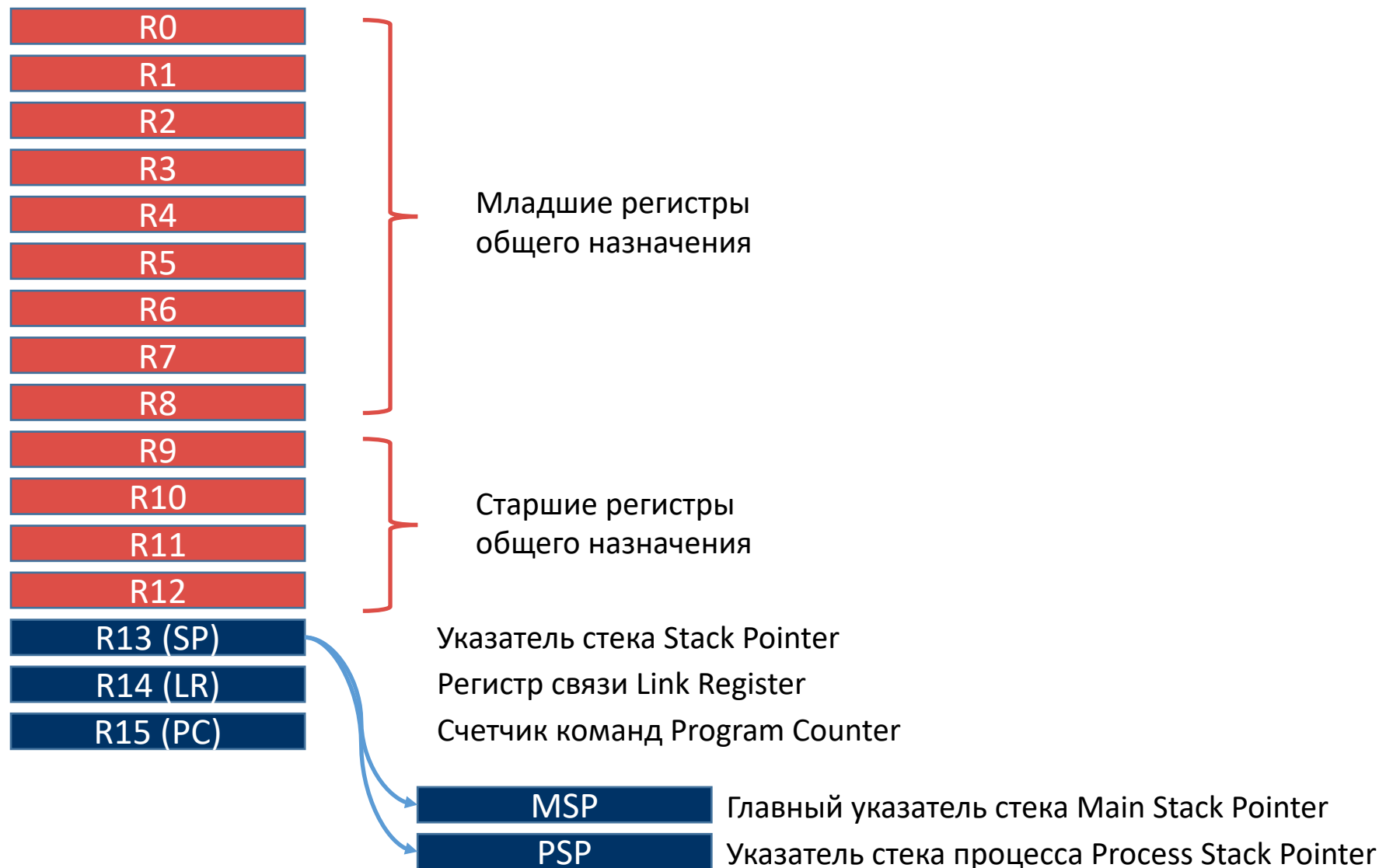
Некоторое количество регистров внутри центрального процессора требуется для выполнения операций. Если необходимо сделать какие-либо операции с данными, то данные необходимо вначале загрузить из памяти в регистр, провести обработку данных и сохранить результат в памяти (Load-Store Architecture).

Как правило, процессоры с архитектурой RISC (такие как ARM, MIPS, RISC-V) вообще не имеют команд для манипуляции данным прямо в памяти, только через загрузку в регистры центрального процессора.

Банк регистров центрального процессора содержит шестнадцать 32-х разрядных регистров. Из них 13 регистров – это регистры общего назначения (РОН), остальные – специальные регистры.



Программно-логическая модель ARM Cortex-M3



Программно-логическая модель ARM Cortex-M3

PRIMASK
FAULTMASK
BASEPRI
CONTROL
xPSR

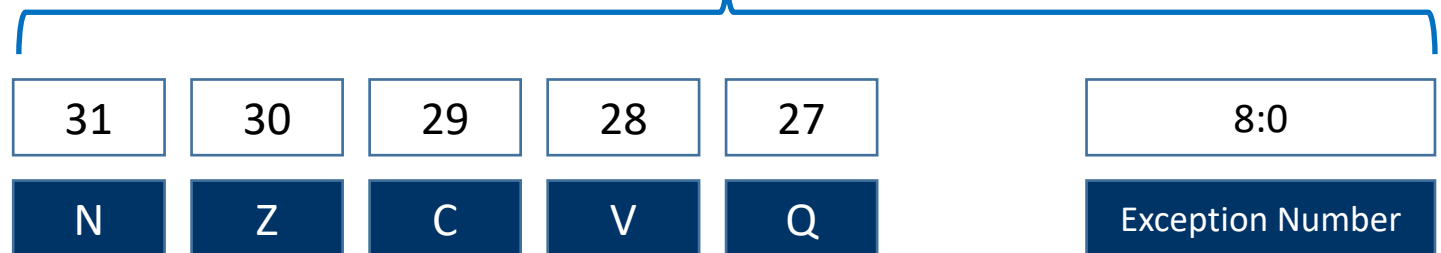
Регистр – разрешения прерывания

Регистр разрешения исключения

Регистр маски прерываний

Регистр управления процессором

Регистр состояние программы



N – отрицательное число

Z – ноль

C – перенос при сложении беззнаковых чисел

V – переполнение при сложении чисел со знаком

Q – насыщение

Exception Number –
номер текущего
исключения/
прерывания

Команды ARM Cortex-M3

Команды перемещения данных:

MOV R4, R0

Команды загрузки/выгрузки:

LDR R0, [R1, #0x8]

STR R0, [SP]

PUSH {R0-R5}

POP {R1}

Арифметические, логические команды и
команды сдвига:

ADD R0, R0, R1

AND R4, R4, R3

ASRS R2, R1, #2

REV R0, R1

Команды перехода:

BL func

B loop

BEQ m1

BNE m2

Специальные команды:

SEV

ISB

DSB

SVC #10

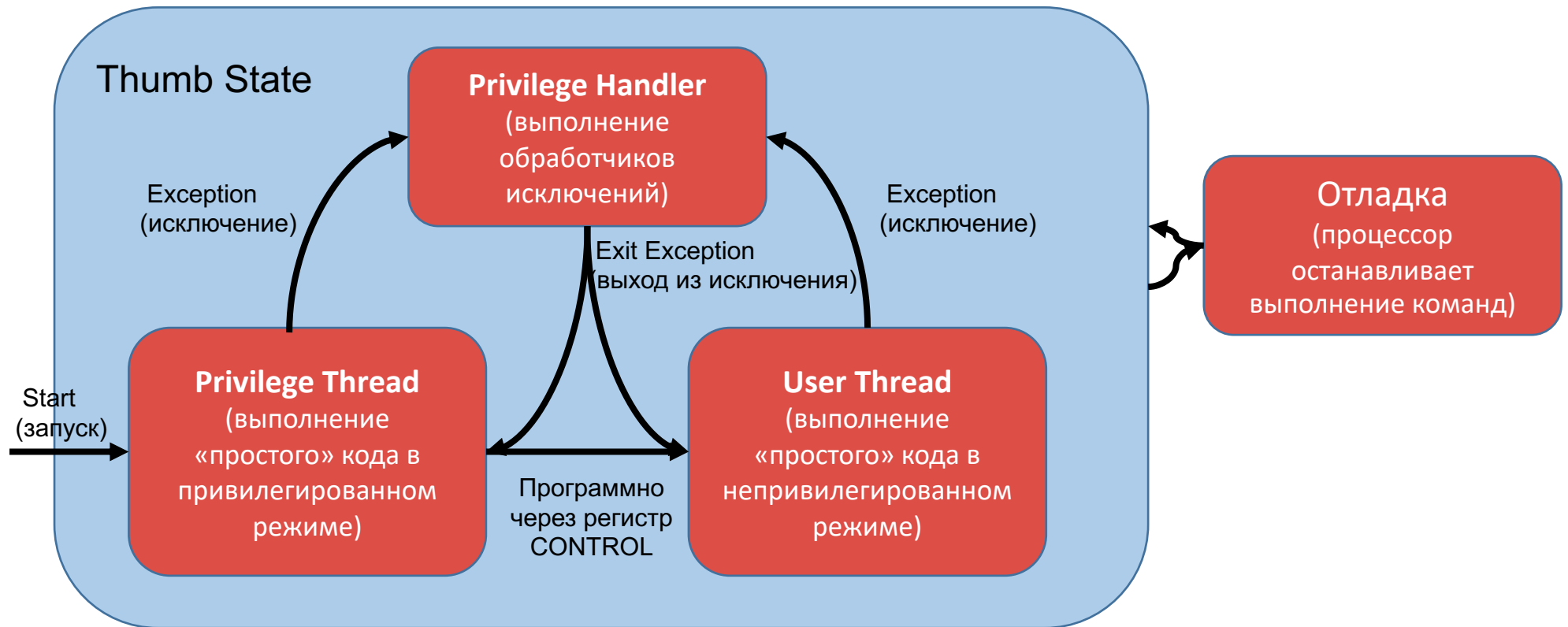
Карта распределения памяти ARM Cortex-M3

Особенности распределения памяти:

- 4 GB адресное пространство;
- Поддерживает Big-Endian и Little Endian;
- Память разделенная на области.



Состояния ARM Cortex-M3



	Привилегированный режим	Непривилегированный режим
Выполняются код исключения (Handler)	Подпрограммы обработчики прерываний	-
Выполняется основной код (Thread)	Код операционной системы	Код задачи/процесса

Исключения и прерывания ARM Cortex-M3

№ исключения	Название	Приоритет	Описание
0	-	-	Нет активного исключения
1	Reset	-3 (макс.)	Сброс
2	NMI	-2	Немаскируемое прерывание (внешняя линия)
3	Hard Fault	-1	Все аппаратные аварии
4	MemManage Fault	Программ.	Обращение к адресам под защитой MPU
5	Bus Fault	Программ.	Ошибка обращения к шине
6	Usage Fault	Программ.	Ошибка исполнения команды
7-10	Reserved	-	
11	SVCall	Программ.	Вызов супервизора (программное прерывание)
12	Debug Monitor	Программ.	Монитор отладки (breakpoint, watchpoint и т.д.)
13	Reserved	-	
14	PendSV	Программ.	Отложенная обработка супервизора
15	SYSTICK	Программ.	Системный таймер
16	IRQ #1	Программ.	Прерывание №1 от периферии
17	IRQ #2	Программ.	Прерывание №2 от периферии
...
255	IRQ #239	Программ.	Прерывание №239 от периферии

Сброс и последовательность запуска ARM Cortex-M3

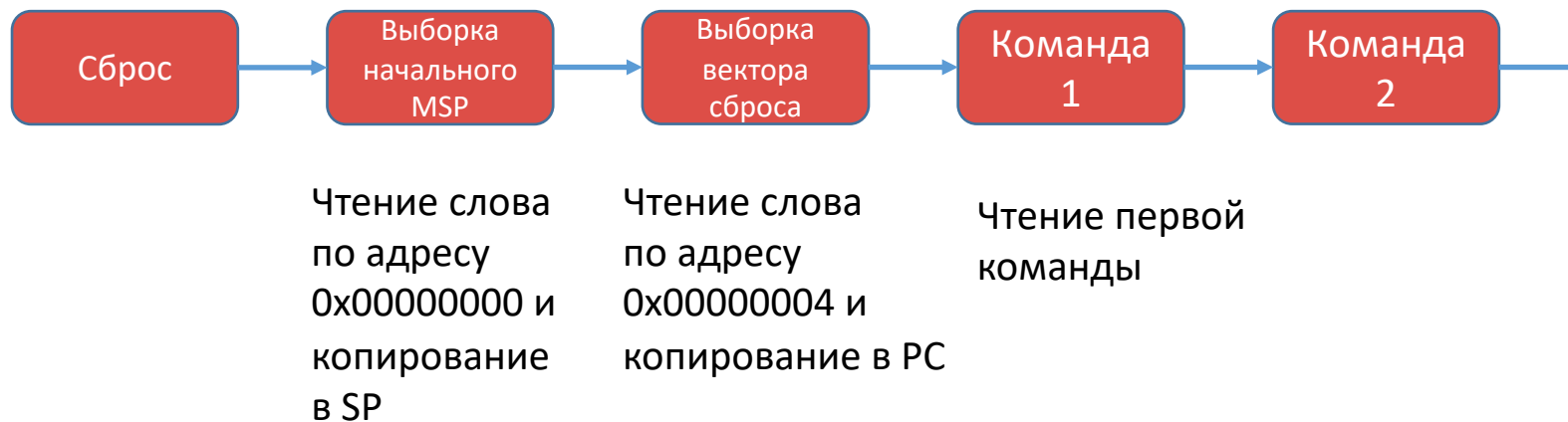
Виды сброса (Reset):

- Power on Reset – сброс по питанию, сбрасывает все в микроконтроллере – процессор, периферию и модуль отладки
- System Reset – сбрасывает только процессор, периферию, но не модуль отладки
- Processor Reset – сбрасывает только процессор

Startup файл:

```
__Vectors DCD __initial_sp ; Вершина стека  
          DCD Reset_Handler ; Вектор сброса  
...
```

Последовательность сброса:



Двоичный интерфейс приложений

Для того, чтобы подпрограммы и программы понимали друг друга разработан двоичный интерфейс приложений (ABI – Application Binary Interface).

В этом интерфейсе описано то, как использовать регистры и стек между вызовами, как осуществлять вызовы супервизора операционной системы и так далее.

ABI для ARM:

<https://github.com/ARM-software/abi-aarch64>

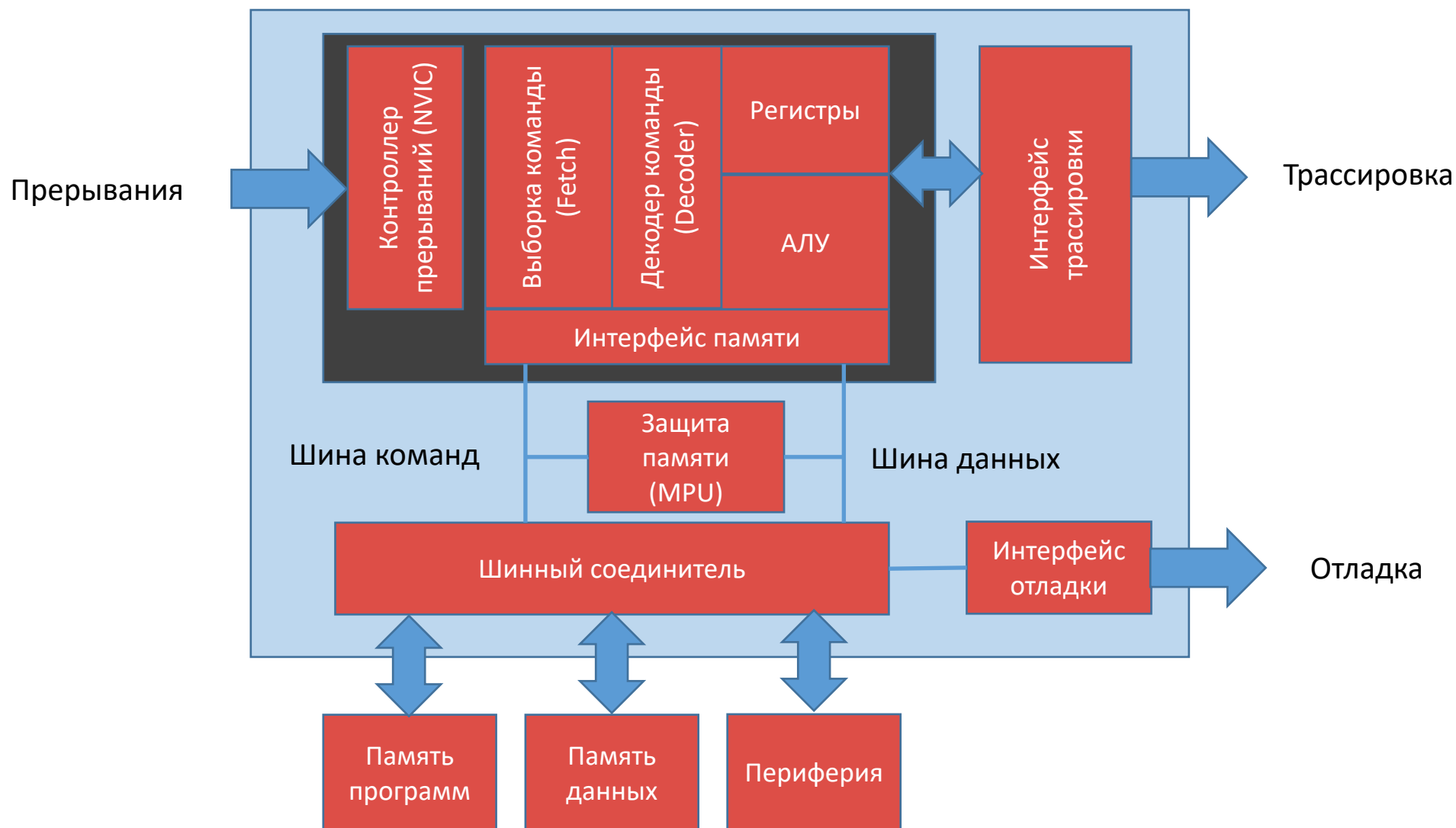
Разработчики компиляторов в обязательном порядке соблюдают ABI. В противном случае программы собранные в разных компиляторах не понимали друг друга.

Если требуется написать подпрограмму на ассемблере с последующим вызовом из языка C, то также необходимо соблюдать требования ABI.

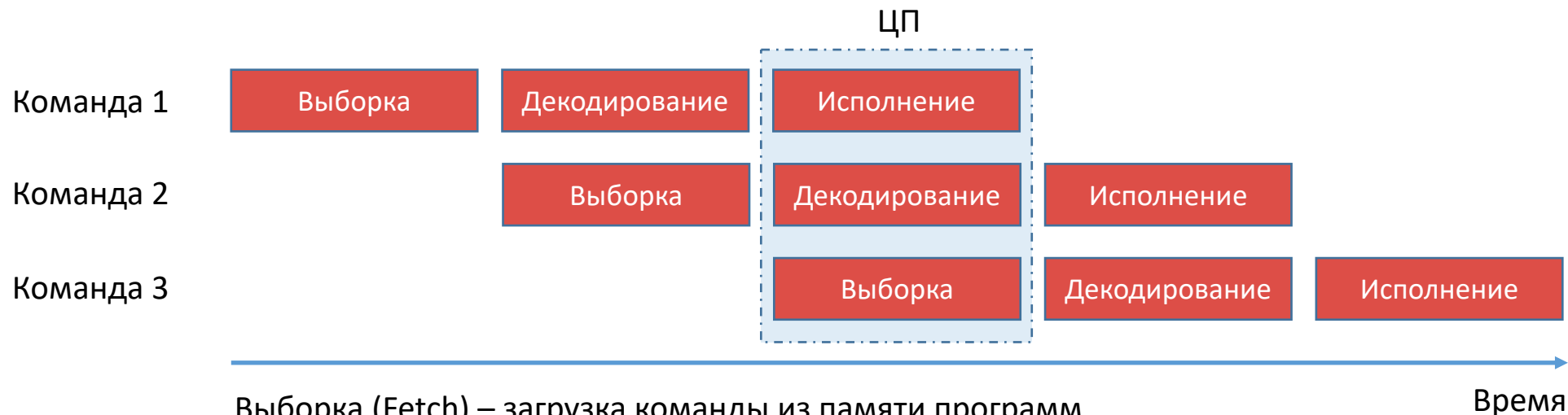
Двоичный интерфейс приложений

R0	Аргумент / возвращаемое значение / временный регистр 1
R1	Аргумент / возвращаемое значение / временный регистр 2
R2	Аргумент / временный регистр 3
R3	Аргумент / временный регистр 4
R4	Регистр для переменных 1
R5	Регистр для переменных 2
R6	Регистр для переменных 3
R7	Регистр для переменных 4
R8	Регистр для переменных 5
R9	Регистр для переменных 6
R10	Регистр для переменных 7
R11	Регистр для переменных 8
R12	Временный регистр (между вызовов процедур)
R13 (SP)	Указатель стека Stack Pointer
R14 (LR)	Регистр возврата Link Register
R15 (PC)	Счетчик команд Program Counter

Структура ARM Cortex-M3



Конвейер ARM Cortex-M3



Выборка (Fetch) – загрузка команды из памяти программ

Декодирование (Decode) – декодирование команды

Исполнение (Execute) – чтение данных из регистров, передача в АЛУ, запись в регистры

Исходный код:

```
LDR  R0, [R1]
LDR  R2, [R3]
ADDS R0, R0, R2
ANDS R0, R0, R2
STR  R0, [R3]
STR  R2, [R1]
```

Выборка	Декодирование	Исполнение
LDR R0, [R1]	–	–
LDR R2, [R3]	LDR R0, [R1]	–
ADDS R0, R0, R2	LDR R2, [R3]	LDR R0, [R1]
ANDS R0, R0, R2	ADDS R0, R0, R2	LDR R2, [R3]
STR R0, [R3]	ANDS R0, R0, R2	ADDS R0, R0, R2
STR R2, [R1]	STR R0, [R3]	ANDS R0, R0, R2
–	STR R2, [R1]	STR R0, [R3]

Оптимальная функция для Cortex-M3

Для написания оптимальной функции на языке C следует понимать архитектуру процессора.

В оптимальной функции:

- До четырех 32-х разрядных аргументов (остальные через стек)
- Возвращаемое значение одно 32-х разрядное или одно 64-х разрядное значение
- До 8 – 12 32-х разрядных локальных переменных (остальные на стеке)
- Оптимальный тип данных: `uint32_t` или `int32_t`

```
int32_t perfect_func(int32_t a, int32_t b, int32_t c, int32_t d)
{
    int32_t e, f, g, h;
    uint32_t i, j, k, l;
    ...
    return e;
}
```

Ограничение на количество локальных переменных и аргументов также соотносится с рекомендацией писать функции имеющие не более 4 – 7 переменных. Человек способен в уме удерживать одновременно не более 4 – 7 элементов, что позволяет «отлаживать и запускать программу в уме».

Другие особенности Cortex-M3

Типы данных float/double медленные в Cortex-M3. В процессоре отсутствует аппаратная поддержка чисел с плавающей запятой (нет FPU), только программная эмуляция.

Программы для многоядерных систем должны учитывать наличие конвейера при работе с общей памятью. Один процессор может прочитать память до того как полностью исполнилась команда другого процессора, которая обращалась к той же памяти.

Центральный процессор имеет модуль защиты памяти (MPU), который позволяет защитить критически важные участки кода и данных.

Центральный процессор имеет некоторую поддержку операционных систем: команды вызова супервизора (SVC), два стека (MSP, PSP), режимы работы процессора (Thread/Process, Privilege/User).

Что внутри микроконтроллера

Центральный процессор (ядро) может занимать менее 10% от площади кристалла.

Остальную часть занимают:

- Постоянная память (Flash, EEPROM)
- Оперативная память (SRAM)
- Периферия (таймеры, интерфейсы и т.д.)
- Генератор тактового сигнала (PLL), система распределения тактового сигнала по кристаллу
- Регулятор напряжения и цепи управления питанием
- Внутренние шины
- Дополнительные цепи для тестирования на производстве



Микроконтроллер 1986BE92QI. Структура

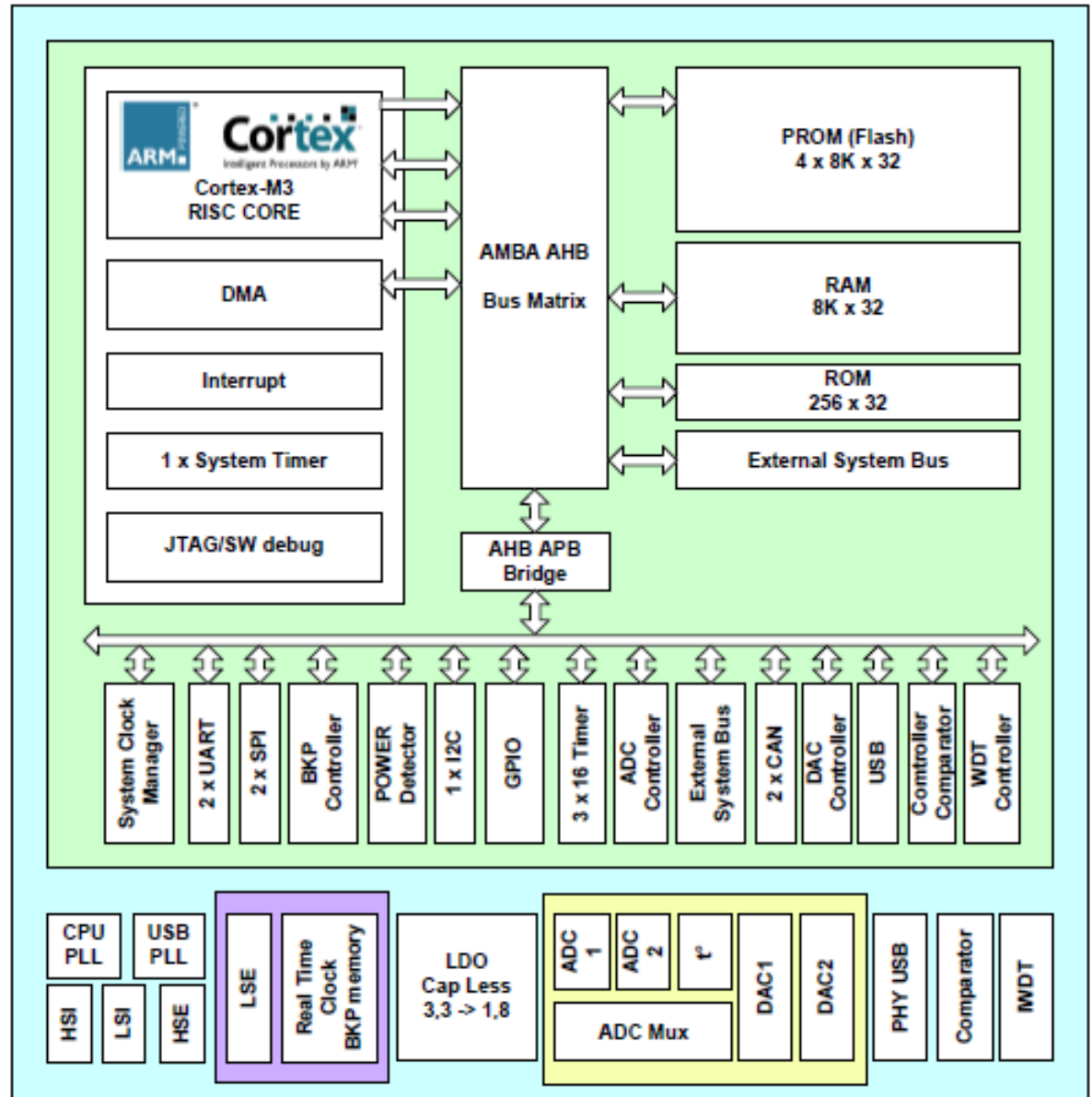
Миландр 1986BE92QI – 32-х разрядный МК общего назначения.

Сфера применения:
системы управления

AMBA - Advanced
Microcontroller Bus Architecture

AHB-Lite (Advanced High-performance Bus) –
высокопроизводительная шина для выборки команд из памяти программ и работы с оперативной память центральным процессором

APB (Advanced Peripheral Bus) –
шина периферийных модулей, имеет ограниченную пропускную способность (меньше разрядность и частота тактирования)



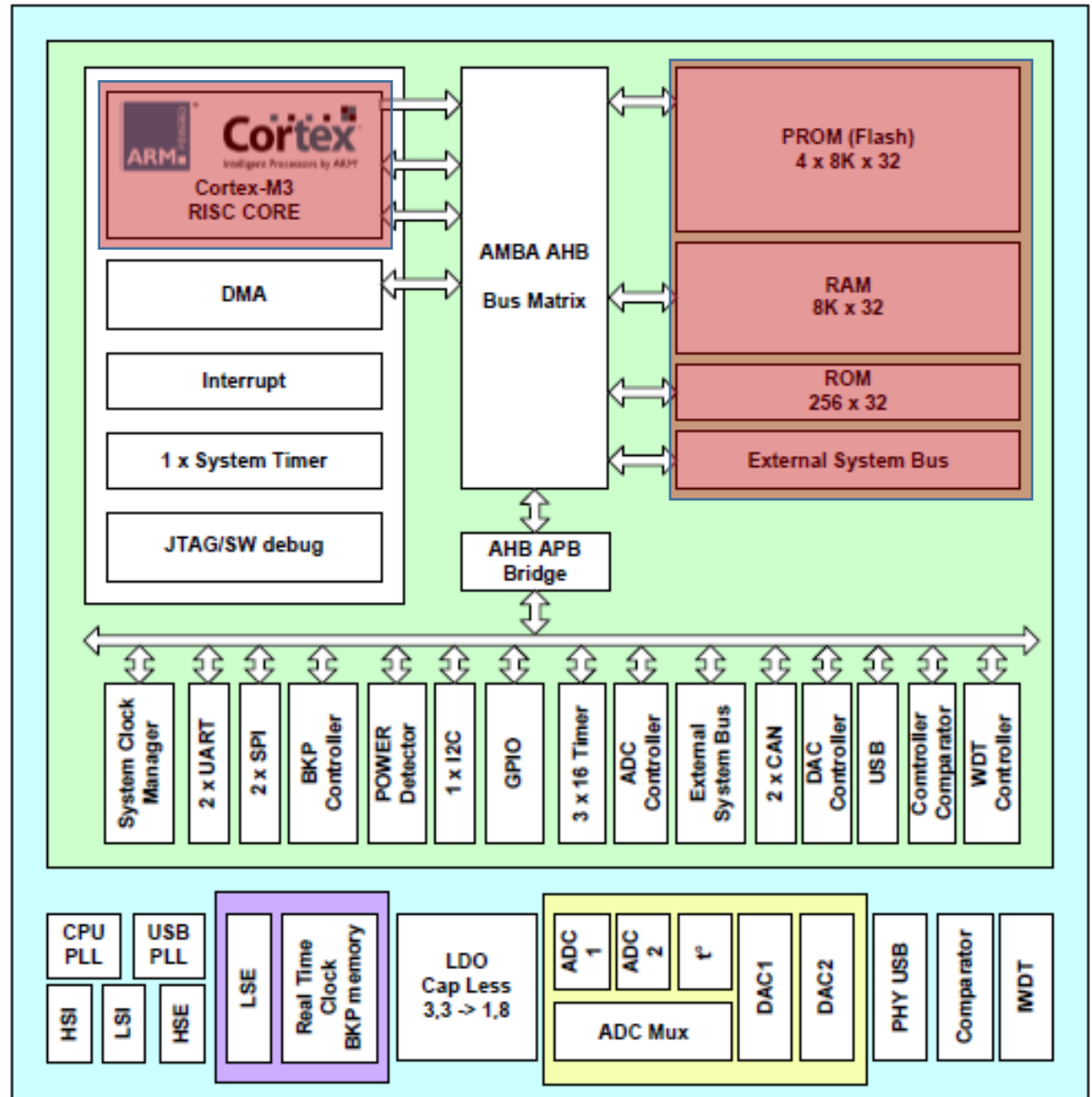
Микроконтроллер 1986BE92QI. Структура

Ядро:

- ARM 32-битное RISC-ядро Cortex-M3, тактовая частота до 80 МГц;
- блок аппаратной защиты памяти MPU;
- умножение за один цикл, аппаратная реализация деления.

Память:

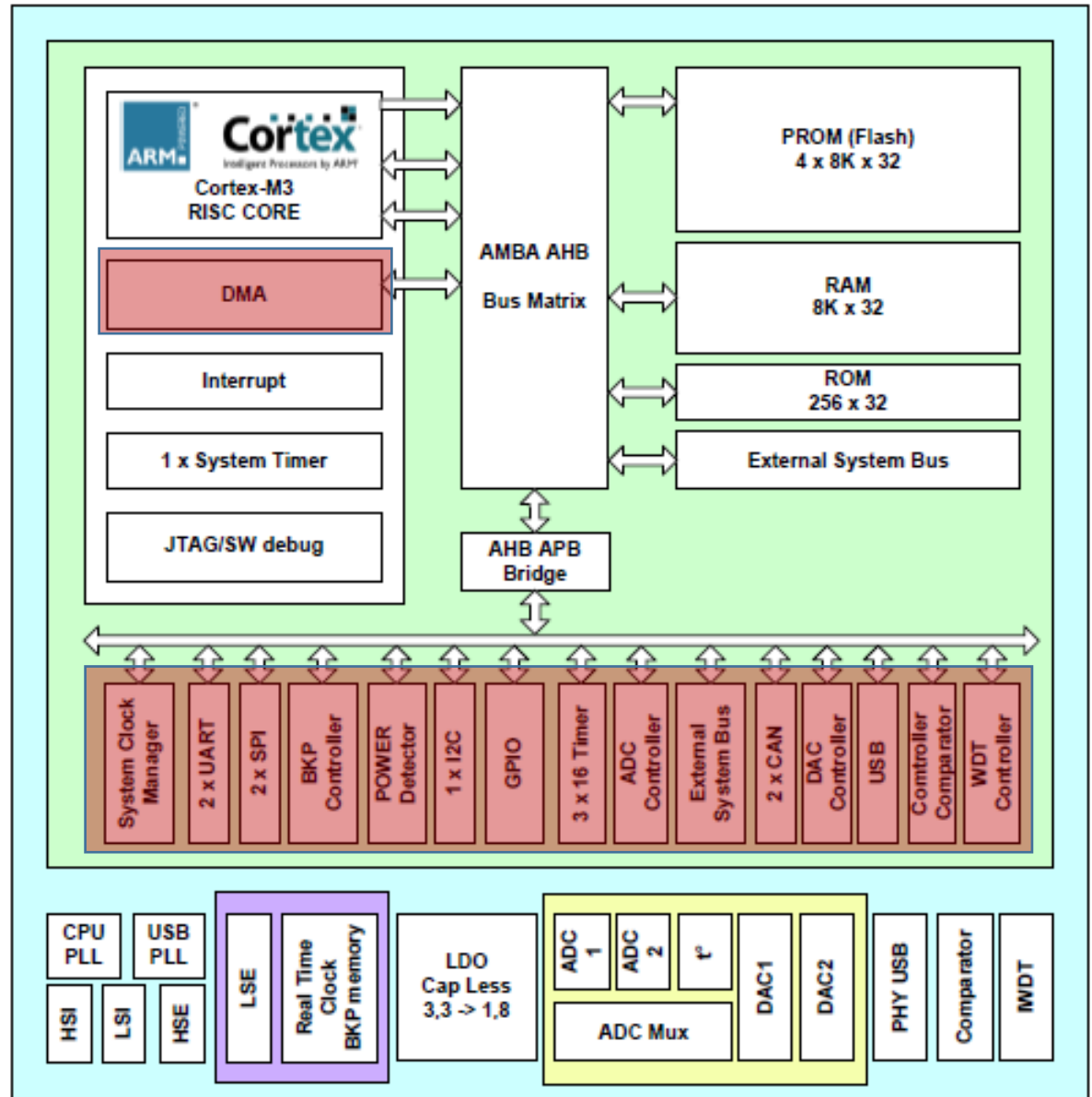
- встроенная энергонезависимая Flash-память программ размером 128 Кбайт;
- встроенное ОЗУ размером 32 Кбайт;
- контроллер внешней шины с поддержкой памяти СОЗУ, ПЗУ, NAND Flash.



Микроконтроллер 1986BE92QI. Структура

Периферия:

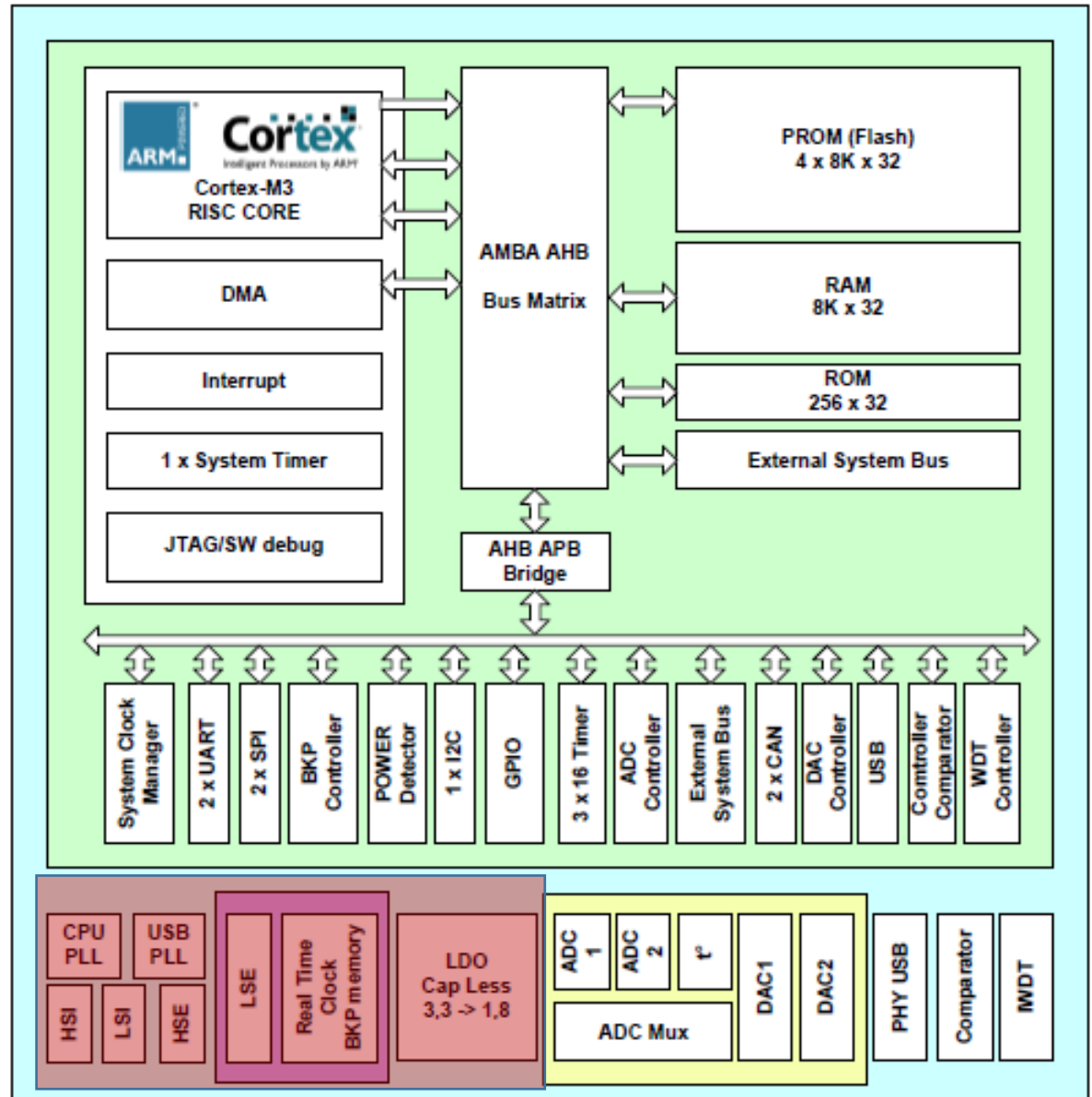
- контроллер DMA с функциями передачи Периферия-Память, Память-Память;
- контроллер USB интерфейса с функциями работы Device и Host;
- контроллеры интерфейсов UART, SPI, I2C, CAN;
- три 16-разрядных таймер-счетчика с функциями ШИМ и регистрации событий;
- до 96 пользовательских линий ввода-вывода.



Микроконтроллер 1986BE92QI. Структура

Питание и тактовые генераторы:

- внешнее питание $2,2 \div 3,6$ В;
- встроенный регулируемый стабилизатор напряжения на 1,8 В для питания ядра;
- встроенные подстраиваемые RC генераторы 8 МГц и 40 кГц;
- внешние кварцевые резонаторы на $2 \div 16$ МГц и 32,768 кГц;
- встроенный умножитель тактовой частоты PLL для ядра;
- встроенный умножитель тактовой частоты PLL для USB.



Микроконтроллер 1986BE92QI. Структура

Аналоговые модули:

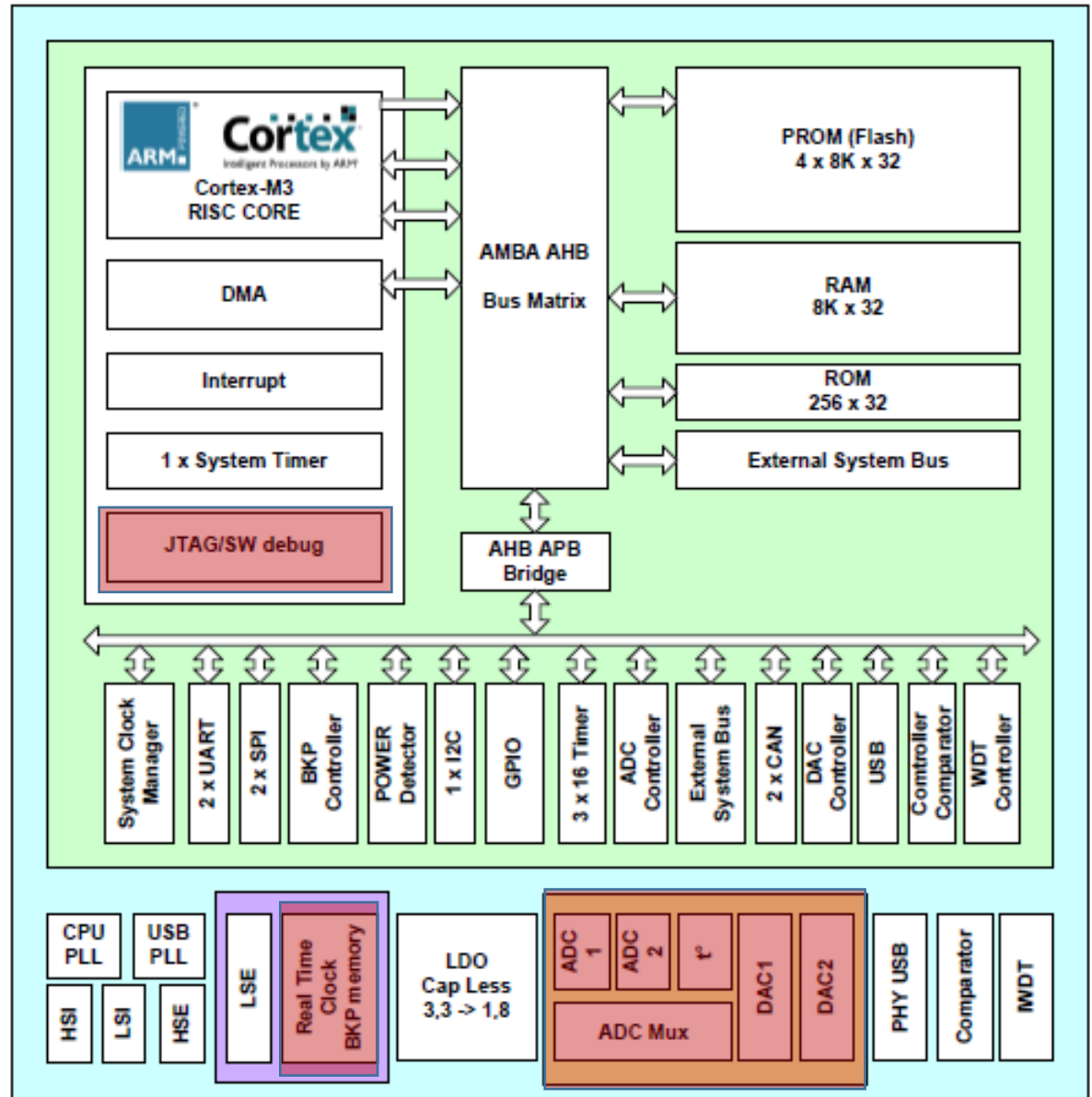
- два 12-разрядных АЦП (до 16 каналов);
- температурный датчик;
- двухканальный 12-разрядный ЦАП;
- встроенный компаратор.

Режим пониженного энергопотребления:

- режимы Sleep, Deep Sleep и Standby;
- батарейный домен с часами реального времени и регистрами аварийного сохранения.

Отладочные интерфейсы:

- последовательные интерфейсы SWD и JTAG.



Заключение

1. Архитектура ARM Cortex-M
 - Современная 32 разрядная RISC архитектура;
 - Большинство производителей микроконтроллеров имеют изделия с ядром Cortex-M.
2. Понимание архитектуры процессора помогает писать оптимальный код на языке C.
3. Микроконтроллер 1986BE92QI
 - 32 разрядный микроконтроллер общего назначения с ядром Cortex-M3;
 - Широкий набор периферии.