Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule

Chapter 2: Stellaris® family of microcontrollers

Stellaris® Cortex™-M3 - Microcontroller Family

Texas Instruments

Texas Instruments - University Program

Heilbronn University, Campus Künzelsau

Prof. Dr.-Ing. Ralf Gessler

Rev. 1.0

Content

Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule

- Chapter 2: Stellaris® family of microcontrollers
  - 2.1 Overview MCU
  - 2.2 Cortex™ -M3: Processor and Peripherals
  - 2.3 Cortex ™ -M3: Programmer model (ISA)
  - 2.4 Peripherals

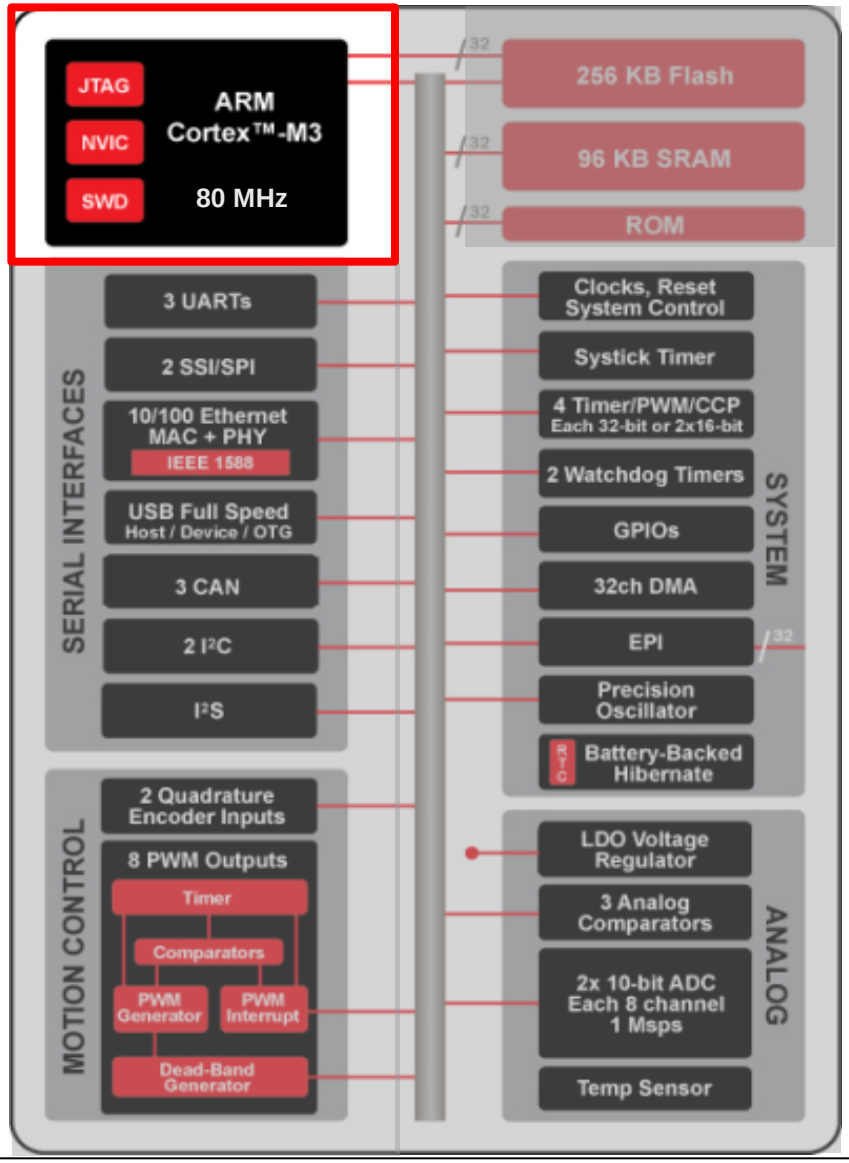
- **Topics:** ARM® Cortex™-M3 **core**, Harvard Architecture, AMBA, **on-chip memory**, Stellaris® Peripherals, APB, AHB

Learning Objectives

Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule

- The chapter describes the basics (theory) of the:
- ARM® Cortex™ -M3 core
  - Processor
  - (Internal) Peripherals
  - Busses: AMBA (System Bus)
- Stellaris® Peripherals
  - (Internal) Memories
  - Busses: APB, AHB
- Structure and questions:
  - What does the Stellaris® architecture looks like?
  - What are the interfaces (busses) between ARM® Processor Core (Cortex™ -M3) and Stellaris® Peripherals
  - What is AMBA?
  - What are the Cortex™ -M3 Peripherals?

Stellaris® Family Overview



Introduction Cortex™ -M3

- The ARM® Cortex™ -M3 processor provides a high-performance, low-cost platform that meets the system requirements of minimal memory allocation, reduced pin count and low power consumption. At the same time it provides enhance computational performance and fast system response to interrupts.
- The Stellaris® family of microcontrollers builds on this core to bring high-performance 32-bit computing to cost-sensitive embedded microcontroller applications.

Main Features Cortex™ -M3

- Compact core.
- Thumb-2 instruction set, delivering the high-performance expected of an ARM® core in the memory size usually associated with 8- and 16-bit devices; typically in the range of a few kilobytes of memory for microcontroller applications.
- Rapid application execution through Harvard architecture characterized by separate buses for instruction and data.
- Fast interrupt handling, by implementing the register manipulations required for handling an interrupt in hardware.
- Deterministic, fast interrupt processing: always 12 cycles, or just 6 cycles with tail-chaining.
- Memory protection unit (MPU) to provide a privileged mode of operation for complex applications.

Main Features Cortex™ -M3 (cont.)

- Migration from the ARM7TDMI processor family for better performance and power efficiency.
- Fully-featured debug interface
- Optimized for single-cycle flash usage
- Three sleep modes with clock gating for low power
- Single-cycle multiply instruction and hardware divide
- Atomic (un-interruptible) operations
- ARM® Thumb2 mixed 16-bit/32-bit instruction set
- 1.25 DMIPS/MHz

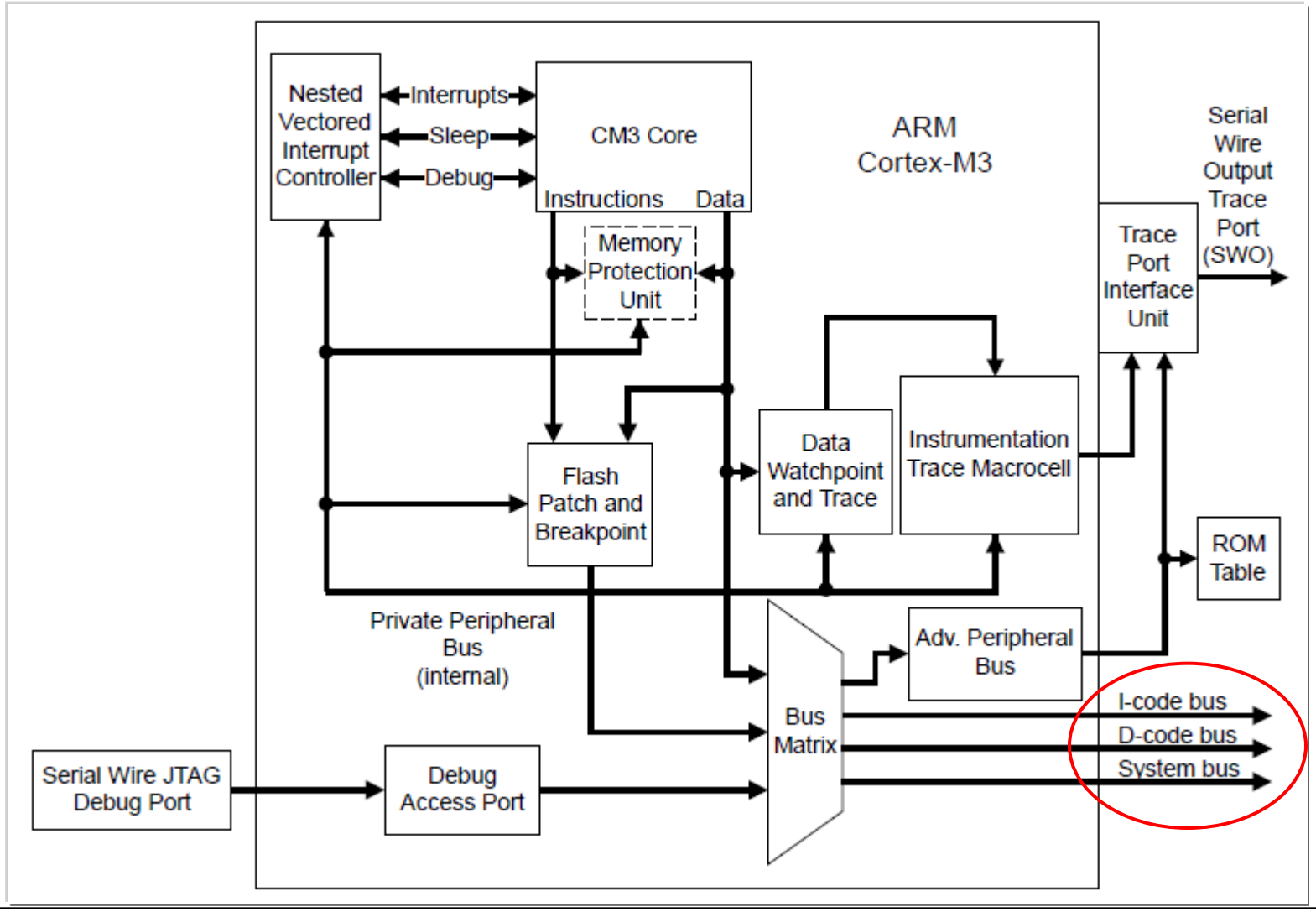
Cortex™ -M3 CPU

- System-Level Interface
  - The Cortex™ -M3 processor provides multiple interfaces using AMBA technology to provide high-speed, low-latency memory accesses. The core supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks, and thread-safe Boolean data handling.
- Integrated Configurable Debug
  - The Cortex™ -M3 processor implements a complete hardware debug solution, providing high system visibility of the processor and memory through either a traditional JTAG port or a 2-pin Serial Wire Debug (SWD) port that is ideal for microcontrollers and other small package devices.
  - The Stellaris® implementation replaces the ARM SW-DP and JTAG-DP with the ARM® CoreSight-compliant Serial Wire JTAG Debug Port (SWJ-DP) interface. The SWJ-DP interface combines the SWD and JTAG debug ports into one module. See the ARM Debug Interface V5 Architecture Specification for details on SWJ-DP.

Cortex™-M3 CPU (cont.)

- Full-featured debug solution
  - Serial Wire JTAG Debug Port (SWJ-DP)
  - Flash Patch and Breakpoint (FPB) unit for implementing breakpoints
  - Data Watchpoint and Trigger (DWT) unit for implementing watchpoints, trigger resources, and system profiling
  - Instrumentation Trace Macrocell (ITM) for support of printf style debugging
  - Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer

CPU Block Diagram



Cortex™-M3 Peripherals

- Cortex™ -M3 processor peripherals, including:
  - SysTick
    - Provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism.
  - Nested Vectored Interrupt Controller (NVIC)
    - Facilitates low-latency exception and interrupt handling
    - Controls power management
    - Implements system control registers
    - Link: Chapter 4.2 “Interrupt functions”
  - System Control Block (SCB)
    - Provides system implementation information and system control, including configuration, control, and reporting of system exceptions.
  - Memory Protection Unit (MPU)
    - Supports the standard ARMv7 Protected Memory System Architecture (PMSA) model. The MPU provides full support for protection regions, overlapping protection regions, access permissions, and exporting memory attributes to the system.

Cortex™ -M3 Peripherals

- Core Peripheral Register Regions

Address	Core Peripheral
0xE000.E010-0xE000.E01F	System Timer
0xE000.E100-0xE000.E4EF 0xE000.EF00-0xE000.EF03	Nested Vectored Interrupt Controller
0xE000.ED00-0xE000.ED3F	System Control Block
0xE000.ED90-0xE000.EDB8	Memory Protection Unit

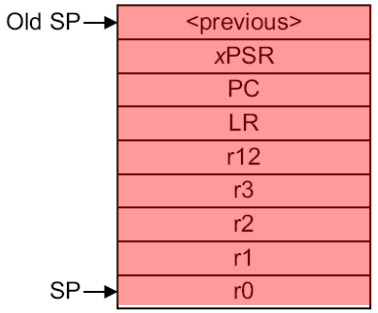
Cortex™ -M3 System Timer

- Cortex™ -M3 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:
  - An RTOS tick timer which fires at a programmable rate (for example, 100 Hz) and invokes a SysTick routine.
  - A high-speed alarm timer using the system clock.
  - A variable rate alarm or signal timer - the duration is range-dependent on the reference clock used and the dynamic range of the counter.
  - A simple counter. Software can use this to measure time to completion and time used.
  - An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

Cortex™ -M3 Interrupts

Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule

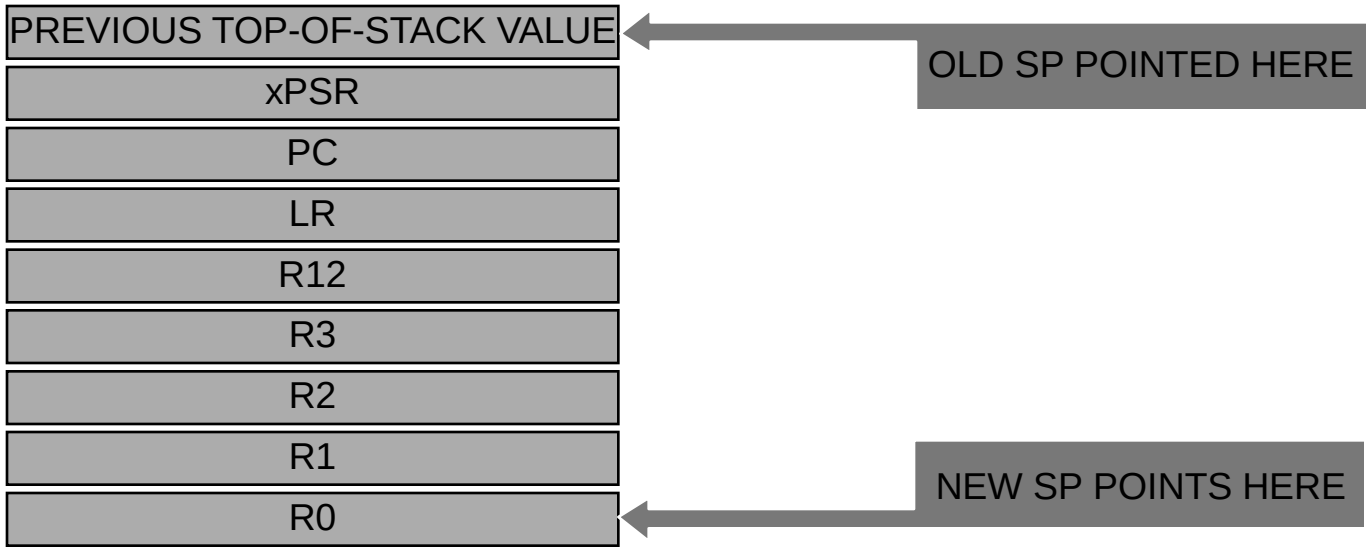
- Automatic state saving/restoring and reading of the vector table entry
- Nested Vectored Interrupt Controller (NVIC) prioritizes and handles all exceptions
- Tail-chaining support
- Pre-emptive/Nested interrupts implementation
- Priority grouping
- Vector table relocation possibility
- Process and Main (Handler) stacks only fully access the NVIC from privileged mode
- 10 Cortex™-M3 core exceptions types with programmable priority except Reset, NMI, Hard fault (highest priorities)
- Up to 53 peripherals interrupts (GPIOs, PWMs, ADCs etc.) on Stellaris®



Exception Model

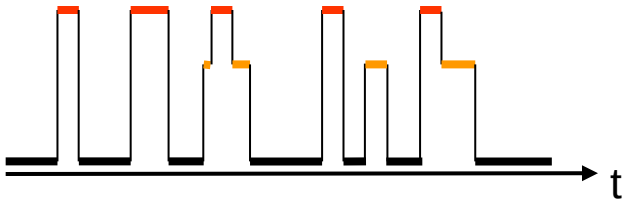
- Exception Model handles all interrupts, synchronous faults and SVC exceptions
  - Exceptions cause current machine state to be stacked
  - Stacked registers conform to Embedded Application Binary Interface (EABI)
- Exception handlers are trivial as register manipulation is carried out in hardware
  - No assembler code required
  - Simple ‘C’ interrupt service routines

    void IRQ(void) { /* my handler */ }



Cortex™ -M3 Interrupts

- Tasks and priorities
- Main application runs as foreground (base level)
  - Easy to write since no “factoring” – just normal application or RTOS based
  - Can use PLC style state-machine poll loop safely: ISRs keep data available
- ISRs for Motor control have highest priority (or priorities)
  - PWM, ADCs, Timer(s), Fault (may be highest), Temp sensor, etc.
- ISRs for communications below that
  - Ethernet, CAN, and/or serial
- May use other priorities as needed
  - Very fast interrupt response time, true nested interrupts, priority masking, easy ISR setup all contribute to making an easy solution
  - Application uses priority masking vs. interrupt-disable if needs critical region



Motor control ISRs (e.g. PWM, ADC)

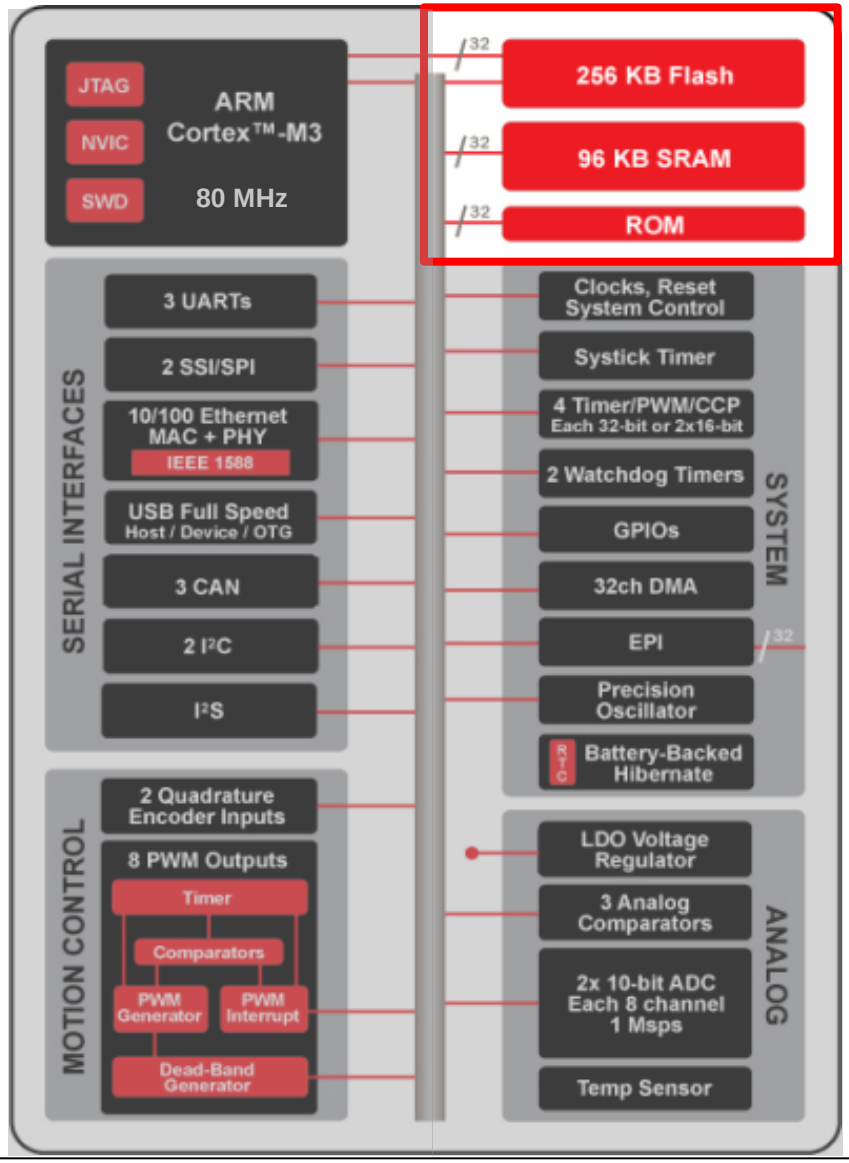
Communication ISRs (e.g. ENET, CAN)

Main application (foreground)

Cortex™ -M3 MPU

- Memory Protection Unit (MPU) → Silicon vendor option
- Benefits
  - Enforce privilege rules
  - Separate processes
  - Enforce access rules
- Features
  - 8 Protection regions (no access, R/W, Read only) from 32B to 4GB range
  - Access permissions (privileged/user)
  - Overlapping protection regions with region priority
  - MPU mismatches and permission violations invoke “MemManage” fault handler

Stellaris® Family Overview



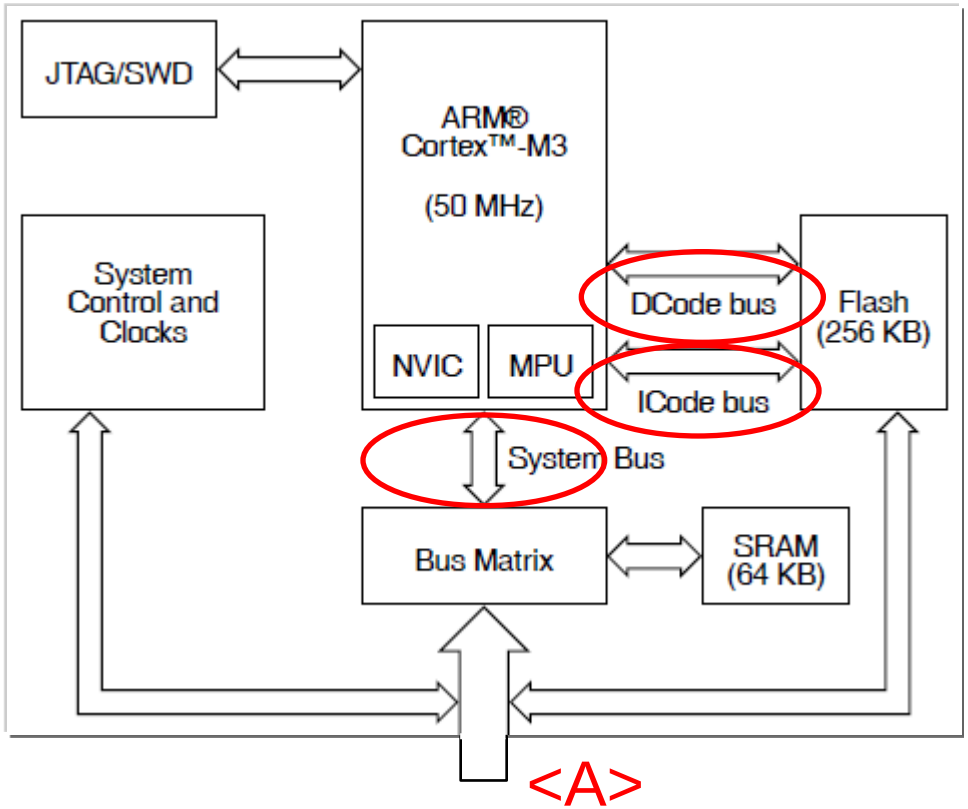
Memory Map

0x0000 0000	**Internal FLASH**
0x0100 0000	**On-chip ROM**
	Boot Loader, DriverLib, AES+CRC software
0x2000 0000	**Bit-banded on-chip SRAM**
0x2200 0000	**Bit-band alias of 0x2000.0000**
0x4000.0000	**Peripherals**
0x4200 0000	**Bit-band alias of 0x4000.0000**
0x6000 0000	**EPI0 mapped peripheral and RAM**
0xE000 0000	**Private peripherals bus**
	Instrumentation Trace Macrocell (ITM)
	Data Watchpoint and Trace (DWT)
	Flash Patch and Breakpoint (FPB)
	Nested Vectored Interrupt Controller (NVIC)
	Trace Port Interface Unit (TPIU)

Buses

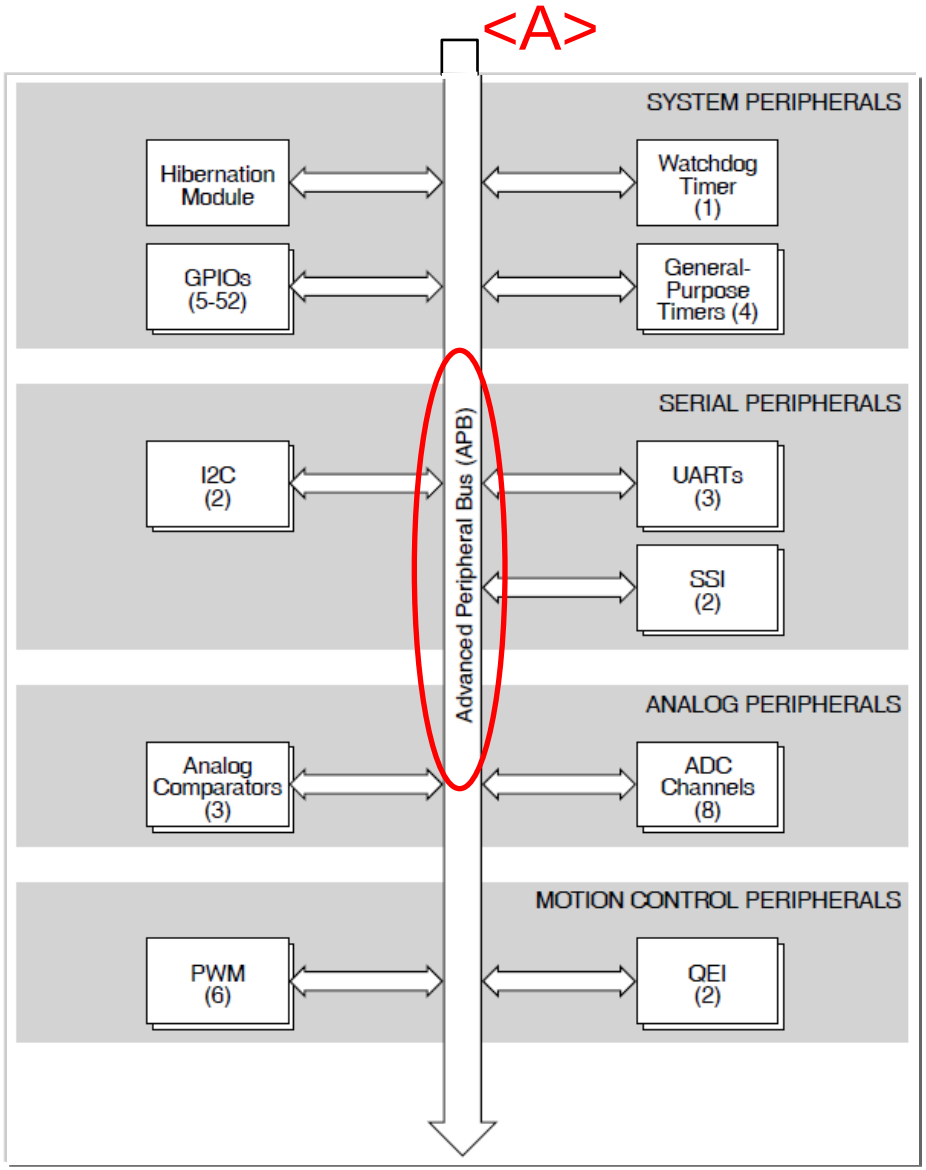
- Cortex™-M3 core is designed for System on a Chip (SOC) approaches.
- The System-Level bus interfaces from the Cortex™-M3 core to microcontroller manufacturer are based on the AHB and APB protocols.
- Advanced Microcontroller Bus Architecture (AMBA) describes these protocols.
- Advanced High-performance Bus (AHB)
  - Instruction bus (I-code bus): 32 bit
  - Data bus (D-code bus): 32 bit
  - System Bus: 32 bit
- (Internal) Advanced Peripheral Bus (APB): 32 bit
  - External Private Peripheral Bus (External PPB): 32 bit

Buses: Stellaris® LM3S1968



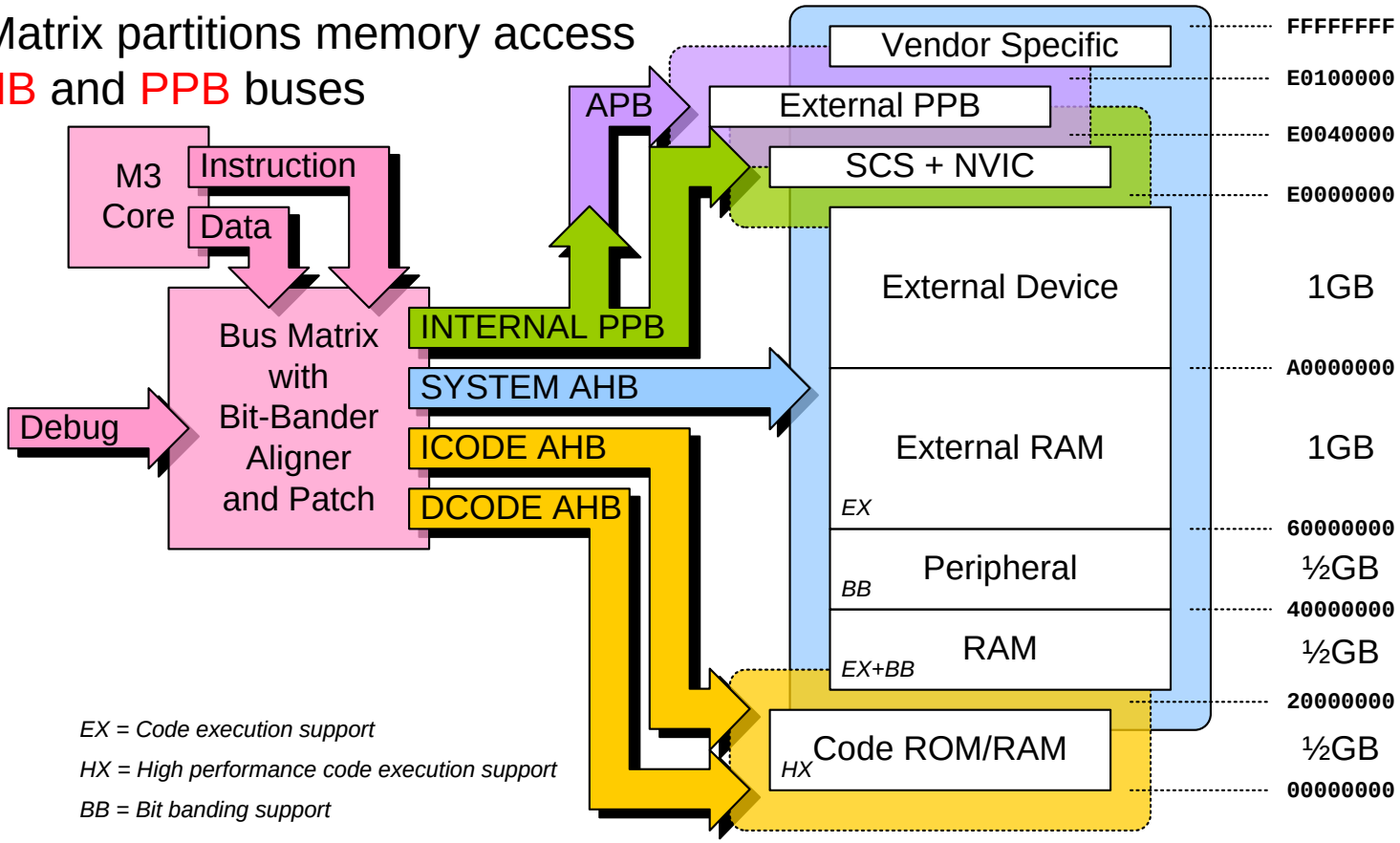
Buses: Stellaris® LM3S1968

Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule



Memory Map

- Very simple linear 4 GB memory map
- Fixed map required to host system components and simplify implementation
- The Bus Matrix partitions memory access via the AHB and PPB buses



Internal memories

- FLASH
  - Flash programming
    - Complete 1-KB blocks can be erased
    - 32-bit word can be programmed individually
    - The write buffer allows continuous 32 words programming
  - Flash memory protection
    - 2-KB Flash memory block granularity
    - “No protection”, “read only”, “execute only” options (committed or not)
    - Possibility to make protection permanent (sensitive data protection)

Internal memories (cont.)

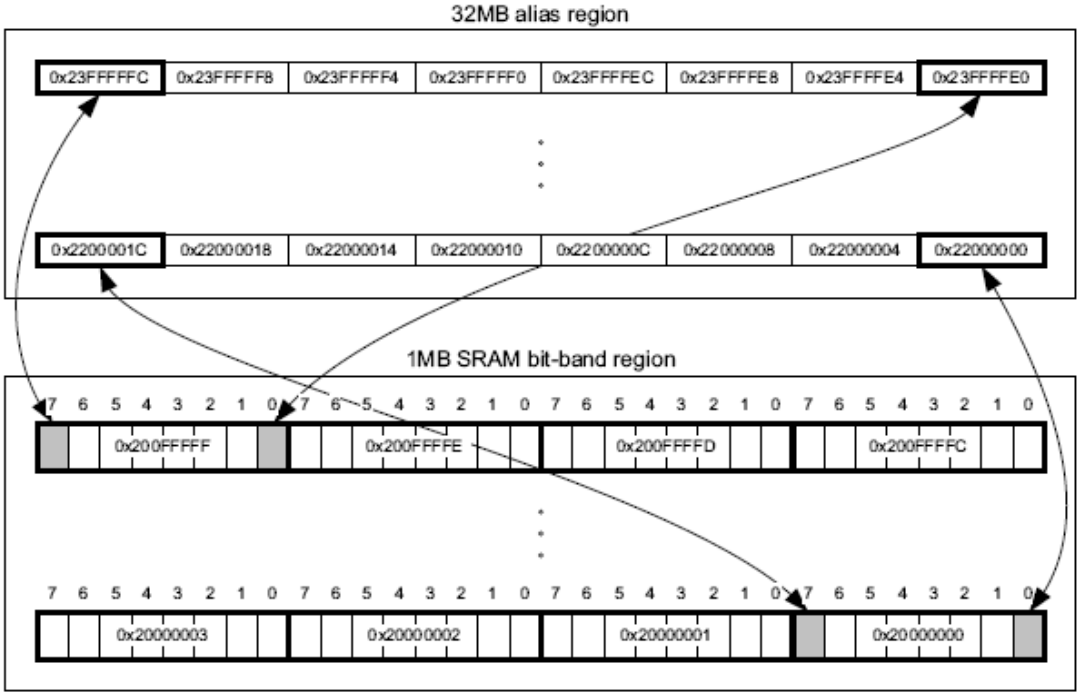
- SRAM
  - Bit-band capability
- ROM
  - Stellaris® Bootloader and vector table
  - Stellaris® Peripheral Driver Library (DriverLib) release for product-specific peripherals and interfaces
  - Advanced Encryption Standard (AES) cryptography tables
  - Cyclic Redundancy Check (CRC) error detection functionality

Bit-band

- A bit-band region maps each word in a bit-band alias region to a single bit in the bit-band region.
- The bit-band regions occupy the lowest 1 MB of the SRAM and peripheral memory regions. Accesses to the 32-MB SRAM alias region map to the 1-MB SRAM bit-band region.
- Bit_word_addr =
  bit_band_base + (byte_offset x 32) + (bit_number × 4)
- Note: A word access to the SRAM or the peripheral bit-band alias region maps to a single bit in the SRAM or peripheral bit-band region.

Bit-band (cont.)

- Example:
  - The alias word at 0x23FFFFE0 maps to bit [0] of the bit-band byte at 0x200FFFFF: 0x23FFFFE0 = 0x22000000 + (0xFFFFF*32) + 0*4.



Questions and Exercises

Hochschule Heilbronn
Heilbronn University
ENGINEERING BUSINESS INFORMATICS
Campus Künzelsau
Reinhold-Würth-Hochschule

1. Explain a SOC.
2. What is AMBA?
3. What is AHB and PPB?
4. Explain Bit-Banding.

Summary and Outlook

- Summary
  - From Stellaris® High-level Block Diagram to Cortex™-M3 Core
  - From the core peripherals to the Stellaris® peripherals
  - From the core to the internal and external busses
- Outlook/How to go on?
  - Chapter 2.3 “Cortex™-M3: Programmer model (ISA)” shows the Programming Interface

References

- [1] Texas Instruments: *Data Sheet - Stellaris® LM3S1968 Microcontroller.* Chapter 2: “Cortex™-M3 Processor”, Chapter 3: “Cortex™ -M3 Peripherals”, Chapter 7: “Internal Memory”, spms037f.pdf, 2011.
- [2] Henri, G.; Texas Instruments: *MCU Training Module - ARM® Cortex™-M3/M4 cores*. EMEA; Oct. 7, 2010.
- [3] Henri, G.; Texas Instruments: *MCU Training Module - System and Peripherals*. EMEA; Oct. 7, 2010.
- [4] Yiu, J.: *The Definitive Guide to the ARM Cortex-M3.* Elsevier Inc., 2007, ISBN: 978-0-7506-8534-4.