Exam questions of ARM Cortex Core Microcontrollers lecture (VIMIAV07)

1. Describe the evolution and main trends of the microcontroller market until the appearance of ARM Cortex core micro controllers. What were the main microcontroller families and what new features they had?
2. Briefly compare the properties of Cortex M0, M3, M4, M7 cores. Characterize the capabilities of typical microcontrollers containing such cores: use a flash memory – pin count graph. Specify typical operating frequency, typical peripherals of microcontrollers with these cores!
3. Describe the main features of the Cortex M3 processor core: architecture, instruction set, instruction execution, pipe-line, major internal core blocks, operating modes! What's new comparing to the ARM7 core?
4. Briefly describe the features of the Cortex M3 based microcontrollers memory organization. What are the major address ranges? What is bit banding, what is non-aligned memory access. What are the main differences comparing to ARM7-based controllers?
5. Present briefly the characteristics of the Cortex M0 core. What are the main differentials comparing to the Cortex M3 and ARM7 cores? What is the Wake-up Interupt Controller (WIC) and why is it important for energy saving?
6. Describe the features of Cortex M4 and Cortex M7 cores. What are SIMD and MAC instructions and why can they be effective for signal processing? What are the additional features of the M7 core comparing to the M4 core?
7. Describe the improvement of the internal architecture of the Cortex M core based microcontrollers in the past 15 years! What type of internal bus structures introduced, what are the most important internal architectural changes?
8. What is the Flash Accelerator Module, why is it necessary? Explain briefly its benefits and its operation!
9. Describe a typical clock tree of a Cortex M core microcontroller. Explain the meaning and necessity of each clock signal source as well as clock signal divisions.
10. Present briefly the structure and goals of the CMSIS (Cortex Microcontroller Software Interface Standard). What are the components of CMSIS Core and what are the role of these files and functions?
11. Introduce a GPIO pin handling with a Cortex M core controller using the CMSIS standard. What kind of files and how do they contain a description of the peripheral registers and how to access these registers? What kind of typical problems presents during such peripheral programing in ARM Cortex-core controller, which were not exists for 8-bit microcontrollers?
12. Describe the features offered by a modern Timer blocks (Capture, Compare, PWM …). Why is the System Timer in an ARM Cortex core important, is it simplex or more complex than a normal timer block?
13. Compare the features of SPI and I2C communication, analyze their advantages and disadvantages. Show an example of peripheral communication using these buses. What are the typical peripherals using these buses?
14. Introduce the USART periphery and its typical applications. How to use USART peripherals for "printf" in a C language environment? Write down this process!

15. Compare Cortex M core NVIC with ARM7 interrupt handling options! What is tail-chaining, what's happening at this time? What are the priority options for Cortex M core interrupt handling?
16. Describe the main features of the NVIC interrupt vector table (you do not have to know the structure of the IT table). How many peripheral interrupts are supported by NVIC, all of these are used? How does CMSIS support IT handling? What's the meaning of the "weak" keyword in CMSIS IT handling?
17. What are the roles of SVC, Pend SVC, Systick, and NMI interrupts in a Cortex M controller? What is the use of the Hard Fault Interrupt? What happens to a typical reset vector, what's the meaning of the Vector Table offset register?
18. Give an example of how to use DMA. What kind of options are offered by a general purpose DMA block? What parameters should a programmer usually set for a DMA transmission?
19. Describe shortly where the DMA block in the system architecture is located in a Cortex M core microcontroller, and how does it connect to the rest of the system. What solutions are used to increase DMA effectiveness in new generation of Cortex M core controllers? How do the designers solve the possible collision of multiple masters' transfer?
20. What is the purpose of automatic address incrementation, and circular buffer option for DMA? Show an example of a typical circular buffer based peripheral handling. What is Scatter and gather type DMA operation?
21. Present briefly the structure and characteristics of FreeRTOS. What Tasks states do you have in such an operating system (do not need to know exactly the names)? What interrupts need to be ported for FreeRTOS to work on an ARM Cortex core controller? What is the structure of the FreeRTOS folder library?
22. Introduce briefly the synchronization methods of FreeRTOS or CMSIS OS abstraction! Show a lifelike example for using each synchronization method (semaphore, mutex, queue)?
23. What is the purpose of Heap_1, Heap_2 .. implementations in FreeRTOS? When a Task is created how and from where its stack memory is allocated? What's the problem with using the Standard C's malloc() function?
24. How can we defend against stack overflows in FreeRTOS, what are the basis of these methods, and do they always provide protection? What's the purpose of Trace Hooks, and what can they be used for? What good is the so-called. Idle hook, what do we use it for?
25. Describe the possibilities we have to influence the active, running mode power consumption of 8-bit and 32-bit microcontrollers. Compare the active power consumption of 8-bit and 32-bit microcontrollers over the past few years.
26. Introduce the typical energy-saving modes of 8-bit and 32-bit microcontrollers. What kind of extra energy saving modes have come up with 32-bit controllers and why?
27. What are the advantages of a 32-bit mictocontrollers against an 8-bit controller in energy-saving applications? Why is it important to use the appropriate clock source in energy-saving modes? What are the main differences between RC and quartz crytal based clock sources? Why there is usually a special low frequency 32.768kHz quartz connection possibility in microcontrollers, show an example for its usage?

28. What is the ROM monitor (also known as GDB stub), traditionally how and where to use it, why it is not the best solution for modern 32 bit microcontrollers?
29. Describe briefly, what steps are being taken when trying to query the value of a variable during debugging (assume OpenOCD and an Eclipse based IDE with GDB debugger). What is GDB, what kind of commands are there, what is the GDB RSP and what is the role of OpenOCD?
30. Describe the blocks of the Coresight debug and trace system of the Cortex M core microcontrollers. How much are these blocks more than traditional debugging: what is the AHB-AP for, what are the ITM, DWT, ETM blocks?
31. Describe the structure of the STM32 Cube! What parts of the CMSIS standard is used by STM32 Cube? What's the difference between Generic and Extensions APIs? What is the CubeMx Application good for? What is the use of _MspInit, _MspDeInit functions when initializing peripherals? Briefly summarize what is the difference between STM32 Cube's three main peripherals handling mode: Pollig, IT, DMA!
32. What are the main differences between USB versions? Which versions are typically used in a microcontroller environment? Describe the USB's four types of transfers. Show examples for using transfer types. What are the rules for making frames out of transfers?
33. Introduce the USB descriptor files! What is the Device, Configuration, Interface, and Endpoint Descriptor? What are their roles? What is the meaning of the Class code in the configuration description? When custom driver is needed, and how to make it for a Vendor specific device?