ARM CM3 architecture

- Architecture = ISA (Programmer's model + Interrupt handling) + Instruction set

CM3 states

- CM3 doesn't have ARM or Jazzelle state.
- Thumb state
    - T=1 (always)
    - Support Thumb2 Instruction set
    - All instructions are half-word aligned. PC last bit is always 0.
    - Eliminates overheads of ARM state (state switching)
- Debug state
    - Used while debugging.

CM3 modes

- Thread mode
    - Bare-metal programming
        - User code runs in thread mode.
    - OS level programming
        - User task code runs in thread mode.
        - System (OS) code runs in thread mode.
- Handler mode
    - Exception/Interrupt handling code is executed in handler mode.

CM3 privilege levels

- Privileged level (PL=0)
    - Privileged level allows access to all registers & execution of all instructions and also have access to NVIC.
    - Handler mode always run in privileged level.
    - Bare-metal programming
        - User code runs in thread mode with privileged level.
    - OS level programming
        - System (OS) code runs in thread mode with privileged level.
- Non-Privileged level (PL=1)
    - Non-privileged mode doesn't allow

            - accessing NVIC registers
            - accessing special registers like PRIMASK, FAULTMASK, BASEPRI, CONTROL.
            - execution of MSR & MRS instructions
            - access to certain memory regions.
        - OS level programming
            - User task code runs in thread mode with non-privileged level.

CM3 registers

- r0-r12 -- general purpose registers
- r13 (sp) -- banked register
    - MSP (Main SP) (SPSel=0)
        - Bare metal programming
            - During exception handling as well as user code execution
        - OS level programming
            - During exception handling as well as system (OS) code execution
    - PSP (Program SP) (SPSel=1)
        - OS level programming
            - During user task code execution
- r14 (lr)
    - to store return address while fn call ~~and exception~~
    - for exception LR stores a magic value i.e. EXC_RETURN - 0xFFFFFFFX
- r15 (pc) -- address of next instruction
    - current instruction + 4 bytes (due to pipeline).
- xPSR -- Program status register
    - Not directly accessible in program.
    - Divided into three parts for programming.
        - IPSR: Interrupt PSR -- Current Interrupt level
        - EPSR: Execution PSR -- IT bits, T bit & GE bits
        - APSR: Application PSR -- NZCVQ ALU flags.
- PRIMASK
    - bit0: To disable all peripheral interrupts and exceptions (other than Reset, NMI & Faults).
- FAULTMASK

        - bit0: To disable all peripheral interrupts and exceptions (other than Reset and NMI).

- BASEPRI
    - bit[8-0]: Interrupt number. All interrupts with priority lower or equal to this number will be disabled.
- CONTROL
    - bit0 (PL)
        - PL=0 -- Privileged level
        - PL=1 -- Non-Privileged level
    - bit1 (SPSel)
        - SPSel=0 -- MSP
        - SPSel=1 -- PSP

CM3 Exception/Interrupt

- CM3 have single vector table which contains addresses of exception handlers as well as interrupt handlers.
- Slot0: Initial address of MSP
- Slot1-15: Exception handlers
- Slot15 onwards: (Peripheral) Interrupt handler

CM3 Exceptions

- Reset
    - When CPU is reset, Reset handler (slot1) is executed in thread mode.
    - Reset handler does system initialization (clock setup, peripheral init, calling main()).
    - Highest priority (-3) (cannot be changed).
- NMI
    - Non-Maskable interrupt is high priority interrupt.
    - It can never be disabled (masked).
    - Higher priority (-2) (cannot be changed).
- Faults
    - MemManage Fault
    - Bus Fault
    - Usage Fault
    - Hard Fault
        - If any of above fault is not handled, hard fault is raised.
        - It must be handled.
        - High priority (-1) (cannot be changed).
- Debug Monitor

        - While debugging (when breakpoint is hit).
- SysTick
    - Due to SysTick hardware
- SVC & PendSVC
    - For implementing System calls in OS.
        - System Calls are the functions exposted by the OS/Kernel, so that user programs/tasks can access those functionalities.
    - SVC is for Software Interrupt.

CM3 Interrupts

- Peripheral Interrupts
    - Slot 16 onwards is used for peripheral interrupt.
    - Number of peripherals is decided by the manufacturer of controller chip.
    - e.g. LPC1768 has 35 peripherals.

CM3 initialization steps

- When reset vector table slot0 (MSP address) is copied in MSP.
- The reset handler address is fetched from vector table (slot1) and jump to that address (PC).
- Reset handler is executed in thread mode (it does initialization) and it calls main().
- main() is executed.

CM3 architecture

- CM3 core
    - Execution engine
        - Registers
        - Pipeline + ALU
        - Memory interface
    - NVIC
    - SysTick timer
    - Trace (debug) interface
- CM3 processor
    - CM3 core
    - MPU (optional)
    - Debug related blocks
    - Bus connections: I-Bus, D-Bus, System Bus

CM3 NVIC

- ISER0 & ISER1
    - 1 bit per peripheral -- 35 bits (32 + 3)
    - Interrupt Set Enable Register -- to enable interrupt write 1 to the bit corresponding to the peripheral.
    - NVIC_EnableIRQ() internally set this register for peripheral interrupt.
- ICER0 & ICER1
    - 1 bit per peripheral -- 35 bits (32 + 3)
    - Interrupt Clear Enable Register -- to disable interrupt write 1 to the bit corresponding to the peripheral.
    - NVIC_DisableIRQ() internally set this register for peripheral interrupt.
- ISPR0 & ISPR1
    - 1 bit per peripheral -- 35 bits (32 + 3)
    - When peripheral interrupt arrives, ISPR bit is set by hw indicating that this interrupt is pending for execution. Then interrupt is given to CM3 core.
    - If ISPR bit is set programmatically, it will force to execute ISR. This is Software triggered interrupt.
    - When handler execution begins, this bit is cleared (0).
- ICPR0 & ICPR1
    - 1 bit per peripheral -- 35 bits (32 + 3)
    - The pending interrupt can be cleared programmatically before it is notified to CM3 core using bit in ICPR register.
- IABR0 & IABR1
    - 1 bit per peripheral -- 35 bits (32 + 3)
    - When Core begin execution of ISR for current interrupt, IABR bit is set. This register is read-only. When ISR is completed, the bit is auto cleared.
- IPR0 to IPR8
    - Per peripheral 5 bits priority (32 priority levels) (+3 bits unused).
    - Each IPRx register have priorities for 4 peripherals.
    - NVIC_SetPriority() function internally does this.

Operating System Functions

- CPU scheduling
- Task (process) management
- Memory management
- File & IO management
- Hardware abstraction