Features,Accesslevel and Modes of the Processor

Features of Corex M0/M3/M4 processor

1. Operational Mode of the processor
2. Different access levels of the processor
3. Register set of the processor
4. The banked stack design of the processor
5. Exceptions and exception handling
6. Interrupt Handling
7. Bus Interfaces and Bus Matrix
8. Memory Architecture of the processor, bit banding, memory endianness
9. Aligned and Unaligned data transfer
10. Bootloader and IAP

Operational Mode of the Processor

M0/M3/M4

1. Processor gives 2 operational modes
    - Thread Mode
    - Handler Mode

2. All your application code will execute under “Thread mode” of the processor. This is also called as “User Mode”.

3. All the exception handlers or interrupt handlers will run under the “Handler mode” of the processor.

4. Whenever the core meets with the system exceptions or any external interrupts then the core will change its mode to handler mode in order to service the ISR associated with that system exception or the interrupt.

Operational Mode of the Processor Contd…

Practical Demonstration!!!

Access Levels of the Processor

M0/M3/M4

1. Processor offers 2 access levels
    a. Privileged Access Level (PAL)
    b. Non-Privileged Access Level (NPAL)
2. If your code is running with PAL, then your code has full access to all the processor specific resources and restricted registers.
3. If your code is running with NPAL, then your code may have access to some of the restricted registers of the processor.
4. By default your code will run in PAL
5. When the processor is in “thread mode”, it's possible to move the processor to NPAL. Once you move out of the PAL to NPAL being in thread mode, then it's not possible to come back to PAL unless you change the processor’s operational mode to “handler mode”.
6. “Handler mode” code execution is always with PAL
7. Use the CONTROL register of the processor if you want to switch between the access levels.

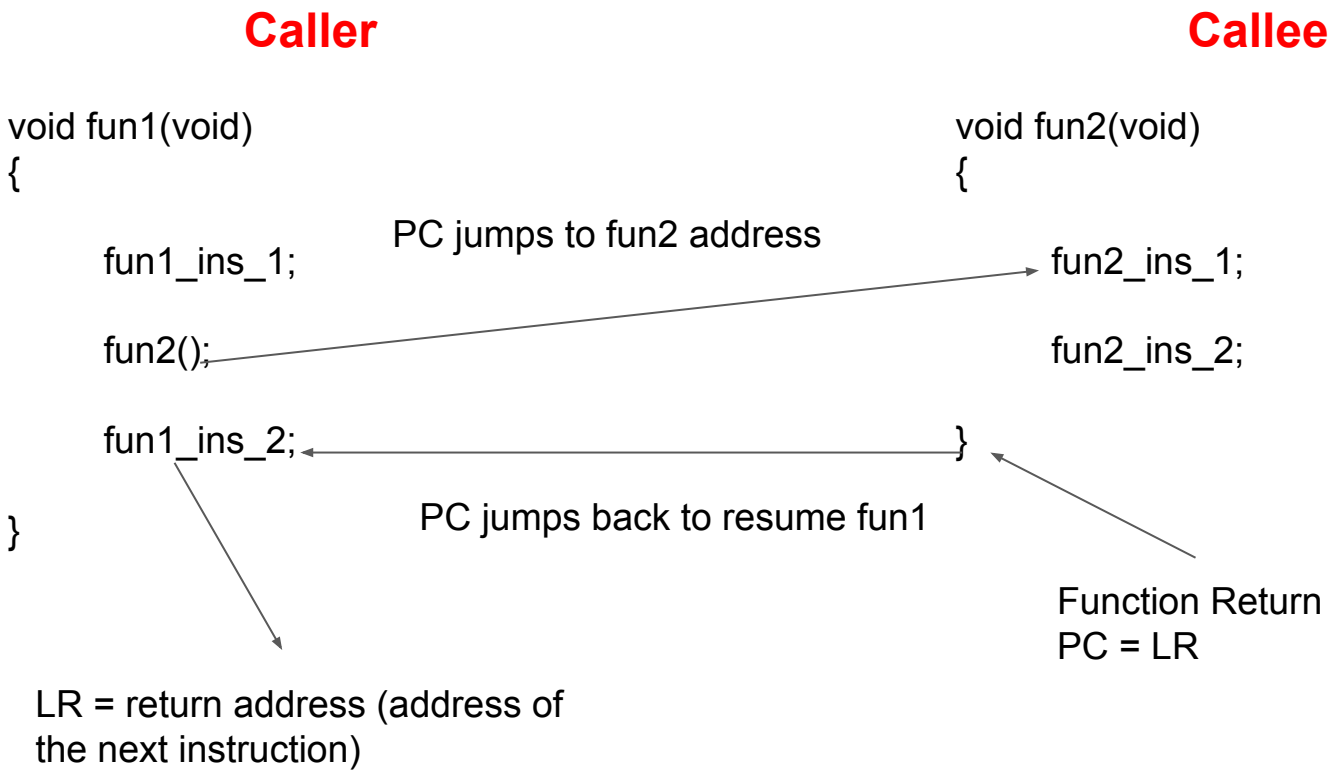
Cortex M Processor Core Registers

Let’s refer to the core registers section from the generic user guide of the Cortex M processor

1. The core registers are actually residing in the Cortex-M4 core
2. R0 to R12 (total 13) registers are for general purpose.
3. All the core registers are 32 bit wide.
4. The register R13 is called as SP (Stack Pointer).
    a. PSP : Process Stack Pointer
    b. MSP : Main Stack Pointer

    Note : PSP and MSP are called as Banked version of SP

5. The Register R14 is called as LR (link Register) ans used to hold the return information during function call and exception handling.
6. The Register R15 is called PC which holds the address of the next instruction to be executed.

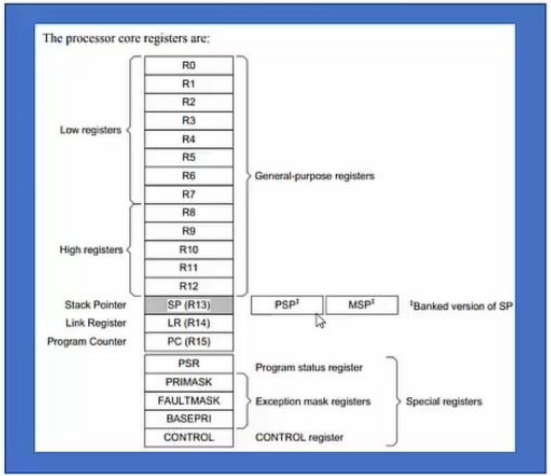

Caller
Callee
void fun1(void)
{
fun1_ins_1;
fun2();
fun1_ins_2;
}
PC jumps to fun2 address
void fun2(void)
{
fun2_ins_1;
fun2_ins_2;
}
PC jumps back to resume fun1
Function Return
PC = LR
LR = return address (address of the next instruction)

Cortex M Processor Core Registers : Special Registers

Let’s refer to the core registers section from the generic user guide of the Cortex M processor

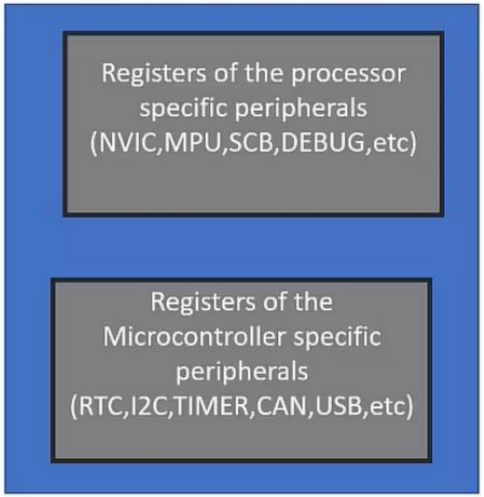
Memory Mapped and Non-Memory Mapped registers

Non-memory mapped registers



- The registers do not have unique addresses to access them. Hence there are not part of the processor memory map.
- You cannot access these registers in a 'C' program using address dereferencing
- To access these registers, you have to use assembly instructions

Memory mapped registers



- Every register has its address in the processor memory map
- You can access these registers in a 'C' program using address dereferencing.

Thank you!!!