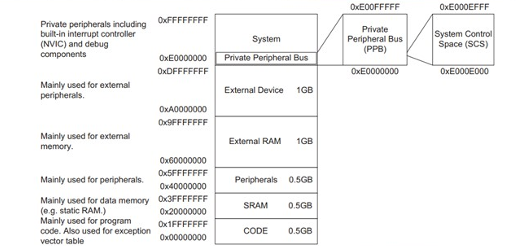
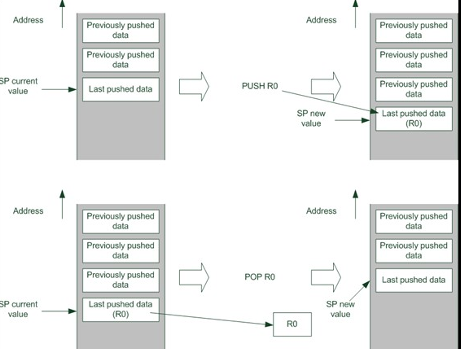
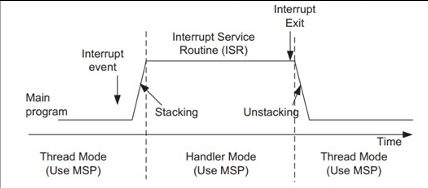
By: Abdelrahman matarawy

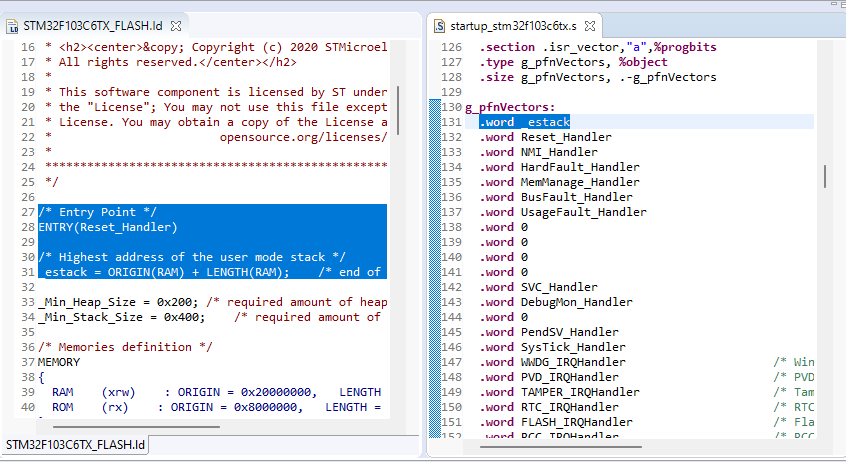
**Lecture5**

**Stack memory**

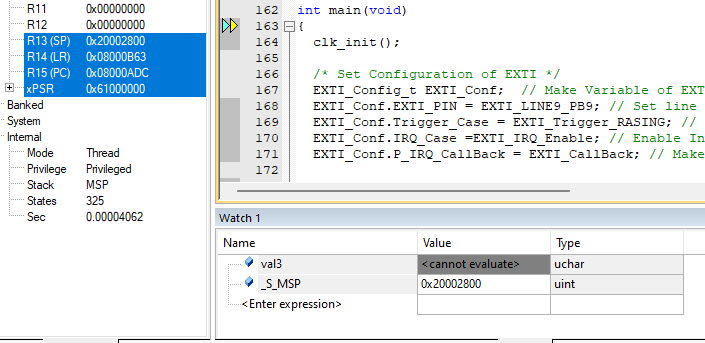
* **Memory Map:**
* Cortex M processor contain 4Gbytes of address space is portioned into 4 groups:
  + Program code (Code Region)
  + Data access (SRAM Region)
  + Peripherals (Peripheral region)
  + Processor internal control and debug components

****

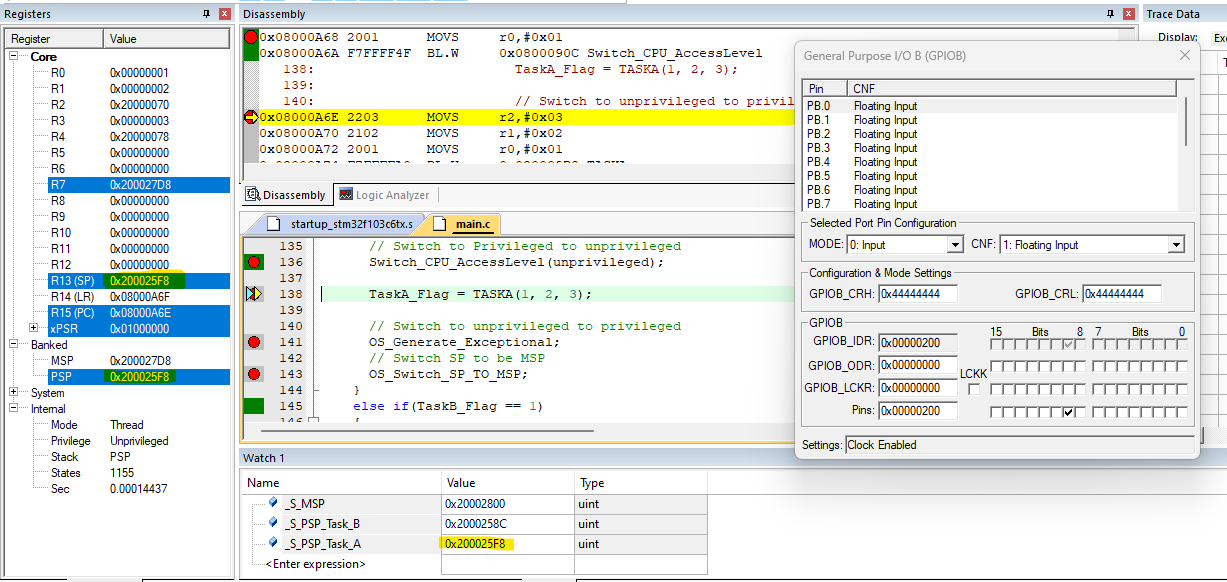
* **Stack Memory:**
  + Stack is kind of Memory use mechanism LIFO data storage buffer.
  + ARM processors use main system memory for stack memory operations and push instructions to store data in stack and pop instruction to retrieve data from stack.
  + ****Cortex M Processor use stack memory model called “Full Descending Stack”
  + For each Push:
    1. Processor decrement SP.
    2. Store Value in Memory.
  + For each POP:
    1. Reading Value from memory location as SP point.
    2. Then increment value of SP.
* **Physically There are two stack pointers in Cortex M Processor** 
  + Main Stack Pointer:
    - This is the default stack pointer used after reset/power ON and used in Exceptional Handlers.
  + Process Stack Pointer:
    - This is alternative stack pointer used only on Thread mode.
    - It’s usually used in Application Tasks in embedded system running on Embedded OS. 
* **Lab:**
* We Will use estack as stack top and start of main stack pointer.

****

* **Notices**
  + that when we start the Firmware SP stopped at address 0x20002800
  + We work for STM32F103C6 SRAM length is 10K bytes (10 \* 1024 = 0x2800)
  + Start of SRAM(0x20000000) the result is 0x200002800
  + Start of MSP set on same address.



* **After first trigger:**
  + PSP carry address of Start\_TaskA.
  + SP switch from MSP to PSP.
  + switch from privileged to unprivileged.



* **After second trigger**
  + PSP carry address of Start\_TaskA
  + SP switch from MSP to PSP
  + Switch from privileged to unprivileged.

