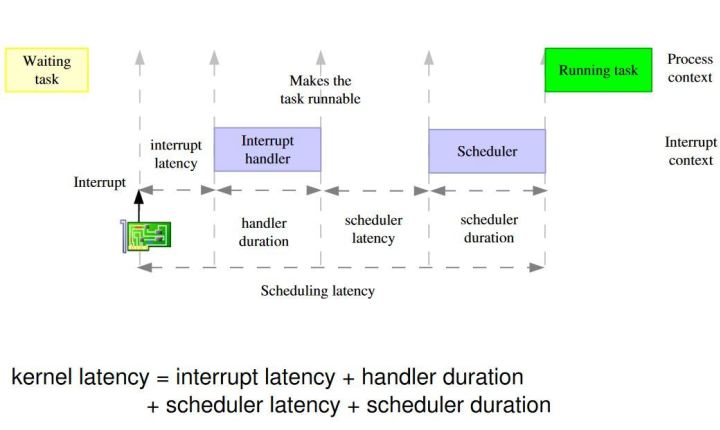
**15. Interrupt Latency**

**1. Interrupt performance measure**

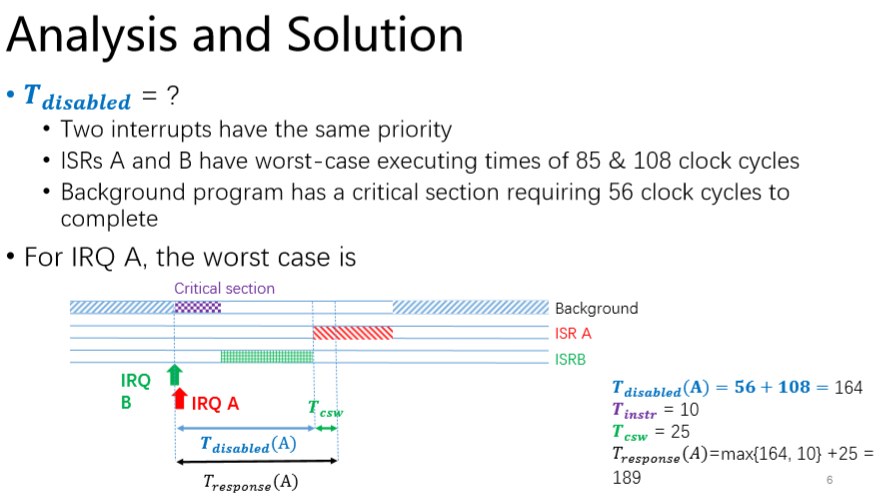
Performance of Fore/Back process model:

1. Interrupt latency = Interval between interrupt event and start of activity

**max** { 𝑻𝒅𝒊𝒔𝒂𝒃𝒍𝒆𝒅, 𝑻𝒊𝒏𝒔𝒕𝒓 }

1. Interrupt response = Interval between interrupt event and start of executing the ISR,

response = latency + context switch time

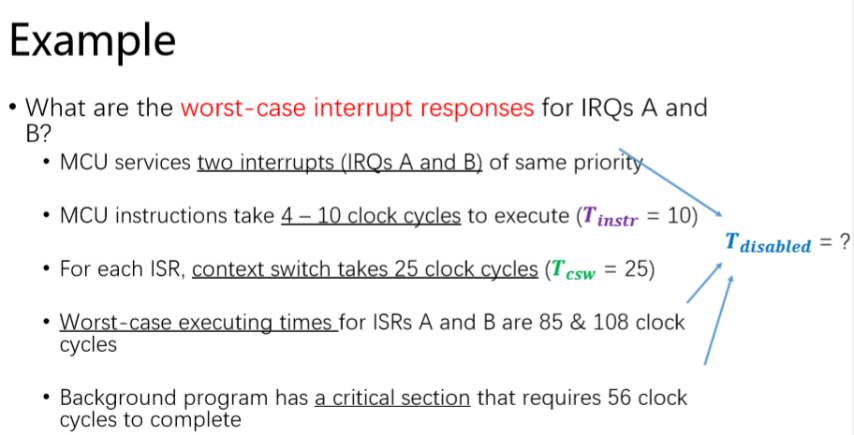
𝑇𝑟𝑒𝑠𝑝𝑜𝑛𝑠𝑒 = **max** { 𝑻𝒅𝒊𝒔𝒂𝒃𝒍𝒆𝒅, 𝑻𝒊𝒏𝒔𝒕𝒓 } + 𝑻𝒄𝒔𝒘 (Worst case)

𝑻𝒅𝒊𝒔𝒂𝒃𝒍𝒆𝒅 = Period that Interrupts are disabled

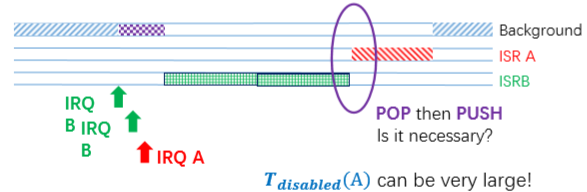
𝑻𝒊𝒏𝒔𝒕𝒓 = Longest execution time for any instruction

Assumptions made:

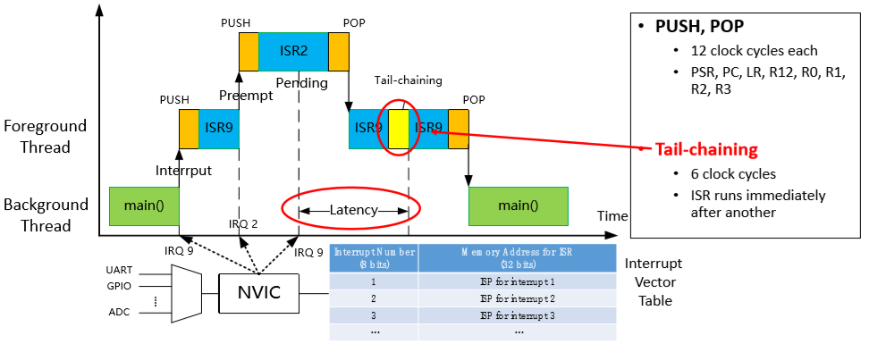
1. Int. not frequent
2. No tail chaining

𝑻𝒄𝒔𝒘 = Context switch time – Storing the state of a process/Thread

𝑻𝒅𝒊𝒔𝒂𝒃𝒍𝒆𝒅 = ***T****exec time (Worst case Int.)* +***T****critical*

**Tail Chaining**

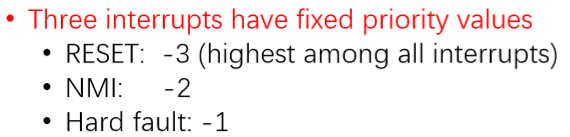
Tail-chaining:

* Back-to-back Interrupt (Int.) processing
* No state saving and restoration between interrupts
* Skips the pop
* Reduce Int. response time (**6** clk cycles)

**No** tail-chaining **PUSH** and **POP** = require **12** clk cycles each

**2. Interrupt priority and preemption**

Preemptive: (within ISR) preempt = take action

* Interrupts with higher priority (i.e lower priority value) can preempt and run before other ISR’s complete

**ISR2** has Higher priority than **ISR9**

* Higher priority value = lower priority
* Lower priority value = higher priority
* Configurable priority values = 0-7

Non – Preemptive: (within ISR) (**default** on many MCU’s)

* All other interrupts are globally disabled
* Priority value = 0

Interrupts with **higher priority** (lower value) preempt interrupts with **lower priority** (higher value)via NVIC

Example: (Take the 3 highest bits)

IntPrioritySet (IRQ\_A, 0x60); // IRQ A has a priority value = 0110 0000 = 011 = 3

IntPrioritySet (IRQ\_B, 0x20); // IRQ A has a priority value = 0010 0000 = 001 = 1 (Runs before A)

