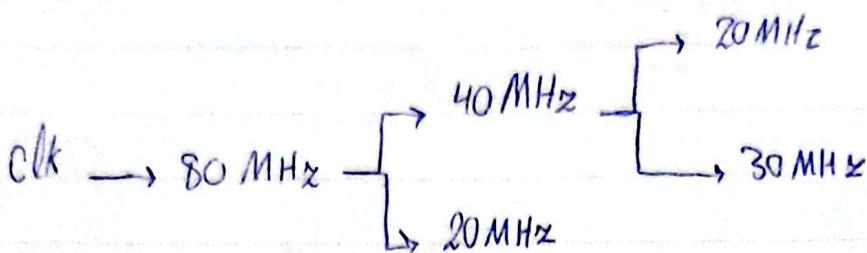


* To test the code performance per time \Rightarrow HIL

Hardware in loop

Technical
reference
manual

preipheral abilities \Rightarrow clock tree \Rightarrow ratio between clock tree \Rightarrow



حسب احتياجات
الـ preipheral clocking

$f_{\text{clk}} = f_{\text{osc}} - \text{freescale}$ \Rightarrow preipheral \Rightarrow



- Power management unit
- Clock

No Clock management system

For P & C customization

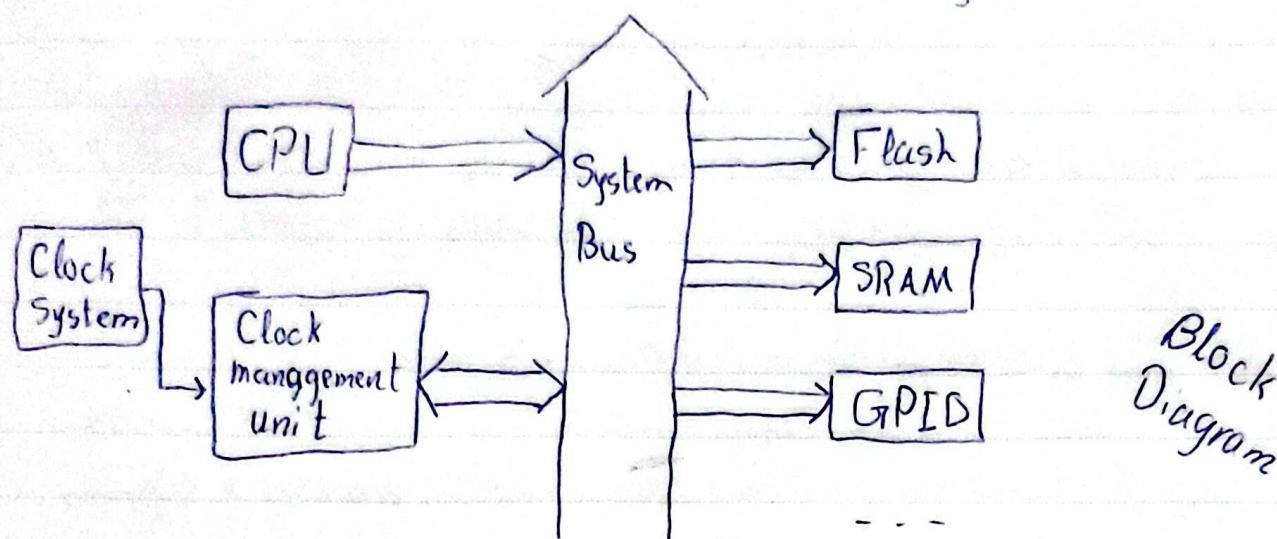
according to peripherals' needs

Default \Rightarrow No clock for All peripherals

\Rightarrow او انتبه حاجة افتح الملوكي

Reset & clock control (CPU nice)

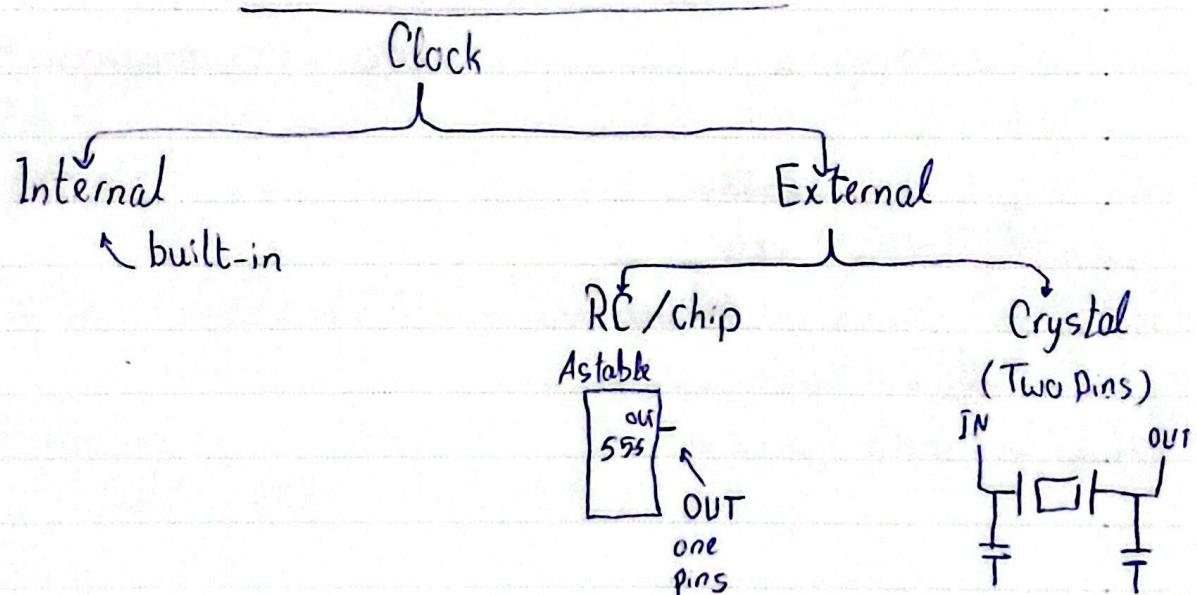
→ General Topology of the clock Architecture sys-clock (SYSCLK)



Peripheral = hardware circuit + Registers

peripheral يُدعى بـ bus أي تربط بين الـ TRM أو في memory map كل الأجهزة

لذلك نشان تقليل دائرة الـ block diagram في المقدمة قبل الـ bus



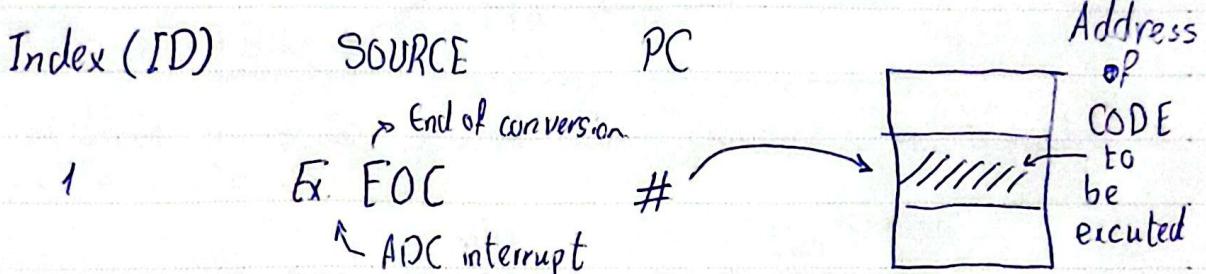
أشارات المهام التي يدخلها بمحظة external events أو حدوث events أو peripheral events routine

Interrupt handlers / Interrupt service routine ويسعى Actions لبيان ما ناقشه لازم

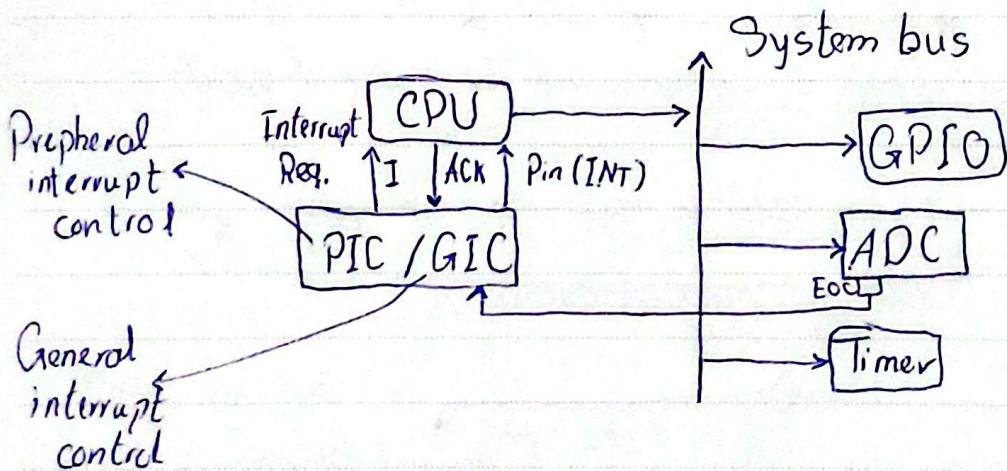
Interrupt → CPU → ISR



* Interrupt vector table (IVT)



ISR لـ ECU العوان هو الذي يجد startup code



Interrupts → HW INT → MP II بـ peripheral acc int.
SW INT (Exceptional INR) → Fault trap → SW instruction

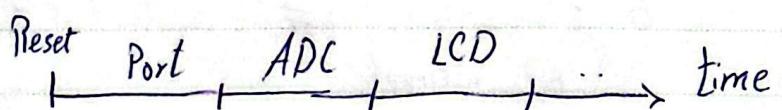
purpose لـ يـ قادر على one pin (ازاي) ↪
 GPIO ↪ UART ↪ ...

عن طريق الـ Routing ↪ AFIO shows Multi Routing

↳ Peripheral (inside or outside) GPIO



* Determinism: Knowing what happens at every point in Timeline

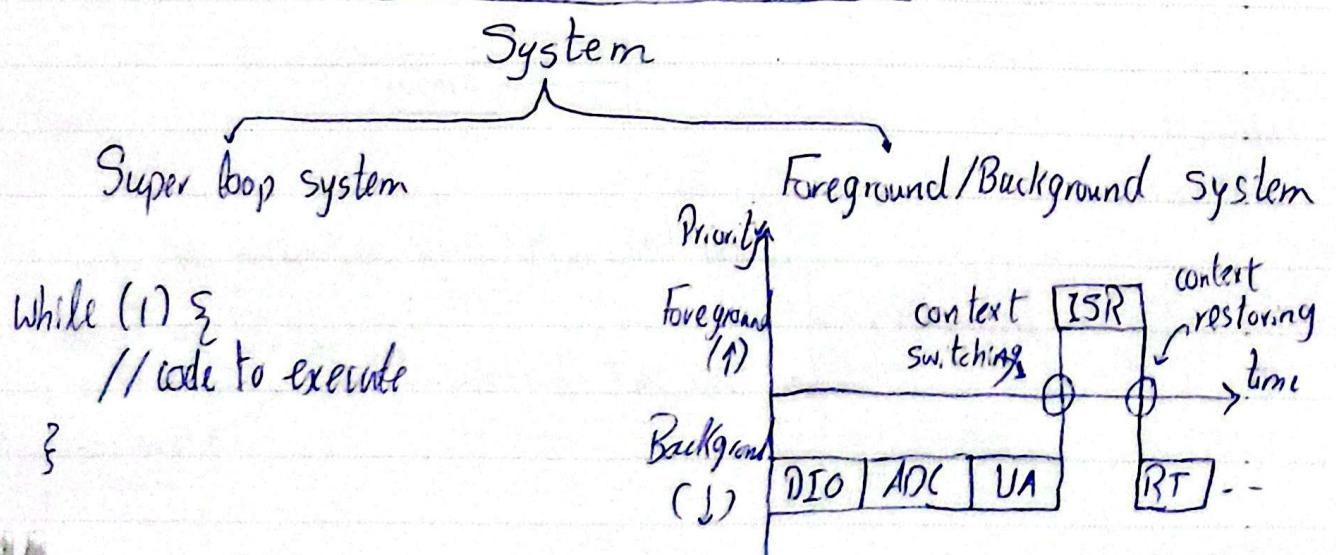


و) فيـ system الـ وقت يـ فـ يـ لـ

* — * — * — *

* Responsiveness: How fast I can respond to external event

مثلـ الـ CPU يـ يـ لـ يـ يـ يـ يـ يـ



Super loop system:

Adv: ① Determinism

② Simple hardware

③ Simple software

Disadvantages:

① low responsiveness

② High power consumption

polling routine. كل من كثر من condition check ، يطلب software للـ U ←

Foreground / Background system

Adv: ① high responsiveness

② low power consumption

Disadvantage: ① low determinism

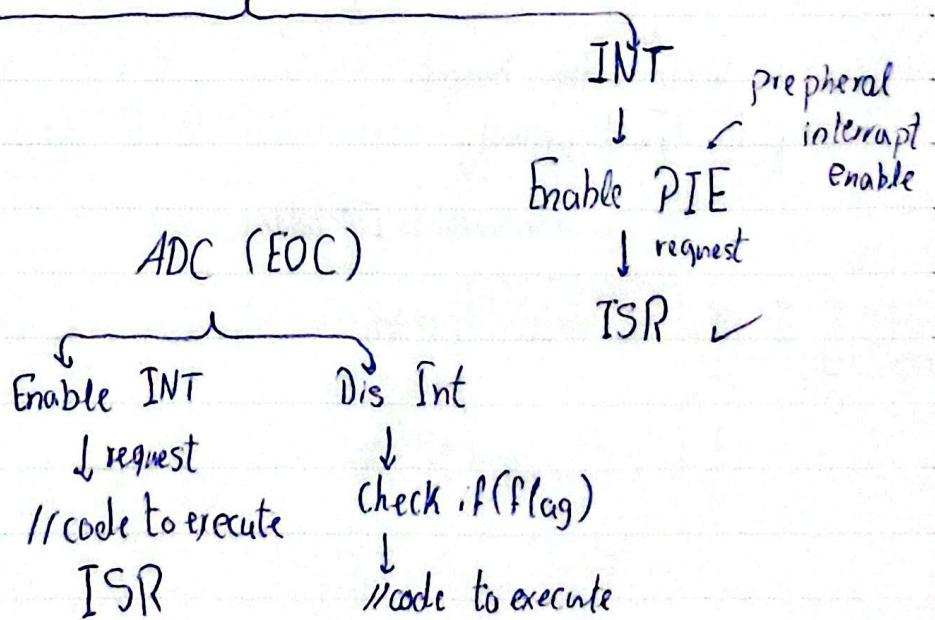
② Extra hardware (GIC)

③ Complex hardware software

Software mechanism (Event handling)

polling

Keep checking



Polling

INT

① Waste time

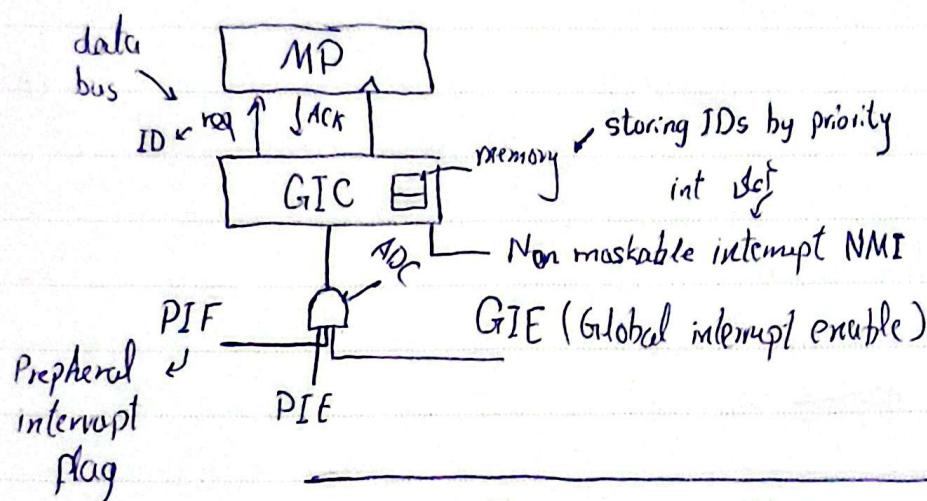
① No wasting time

② Bad design (Bug)

logical error

مهم check if called

② No bugs



* Servicing INT

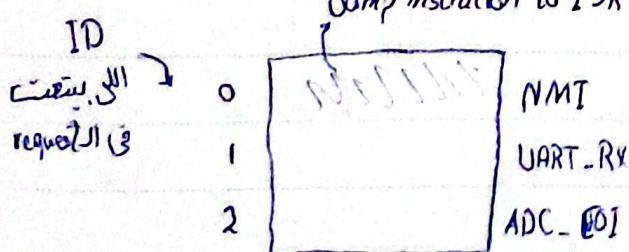
Vector table based

OR Fixed priority

Non-vectorized priority

OR Flexible priority

Jump instruction to ISR address



Steps for fixed priority

- ① Peripheral sends pulse to GIC
- ② GIC writes ID in the register
- ③ GIC sends request to MP with the ID in the data bus
- ④ MP reads the ID
- ⑤ Jumps to the address of ID in vector table
- ⑥ " " of ISR and execute it

Steps for flexible priority (Preferred in the small number of INT)

- ① GIC sends pulse to MP
- ② MP jumps to a location of one vector always
- ③ MP jumps to ISR

⇒ Shape of ISR : ISR()

```
if (ADC_EOC == 1) {  
    // code to execute      ← first priority
```

```
    }  
    if (UART_RX == 1) {  
        // code to execute  
    }
```

}

fixed Priority

Flexible priority

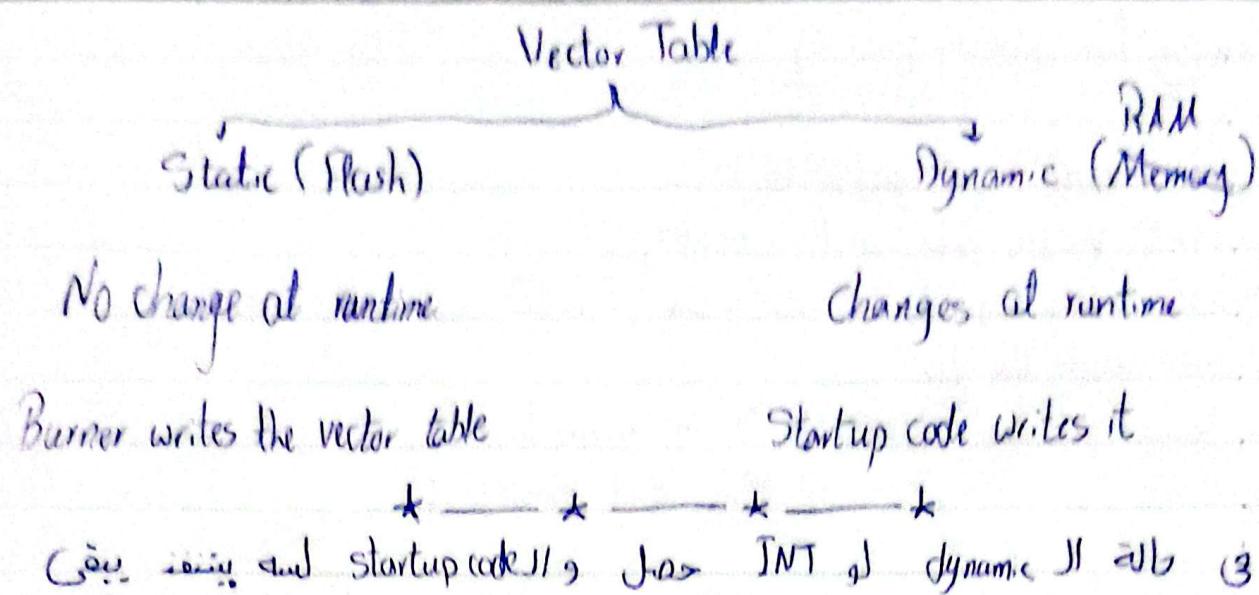
Hardware Priority

Software Priority

Time latency (1)

Time latency (1)

SENA



Undefined behavior ↪ Garbage (junk) vector table (Junk)

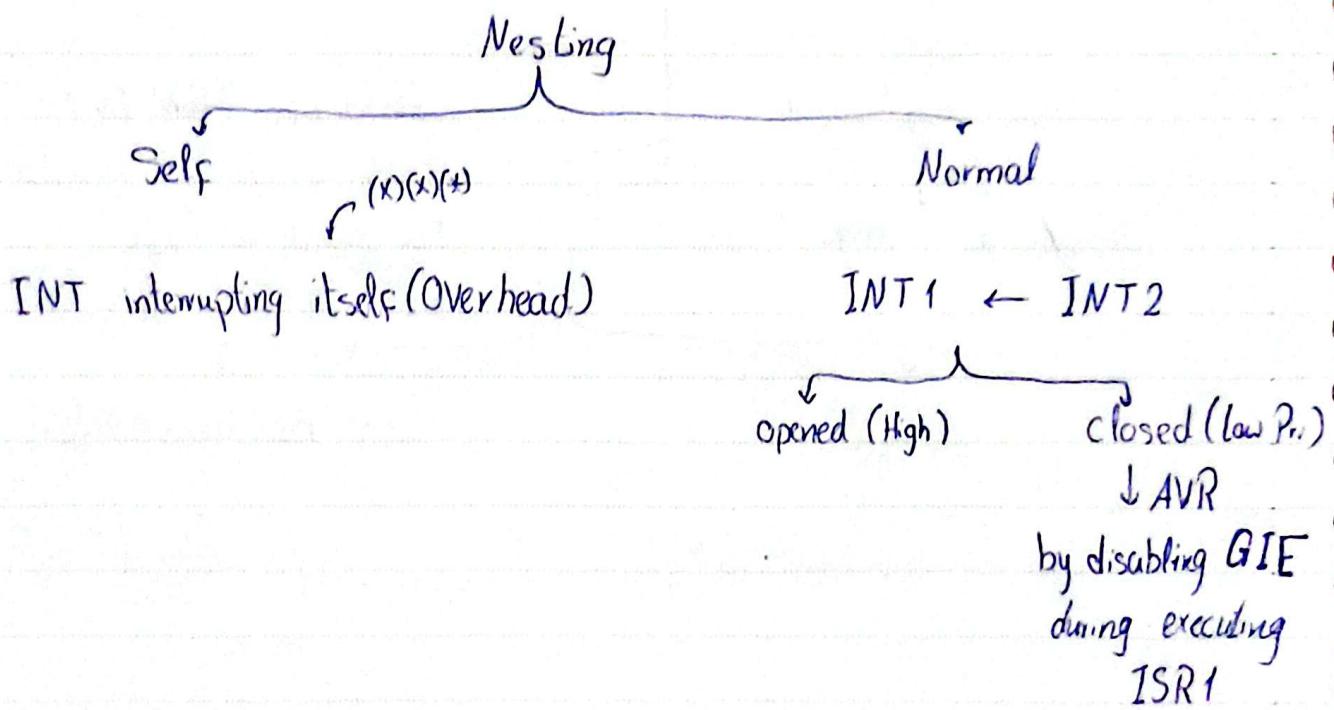
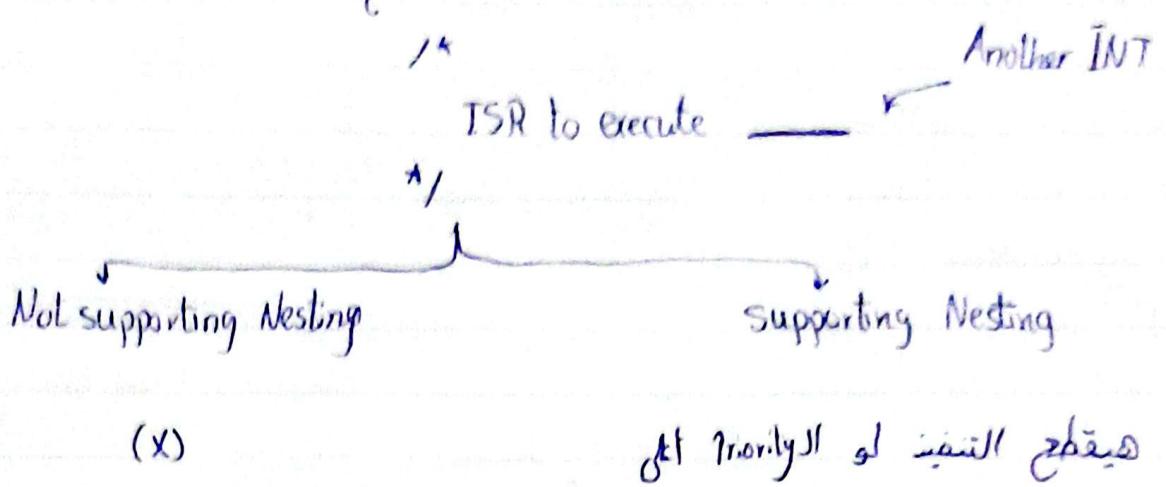
Solution → Disable INT → [Startup code] → Enable INT

∴ Static is faster than dynamic

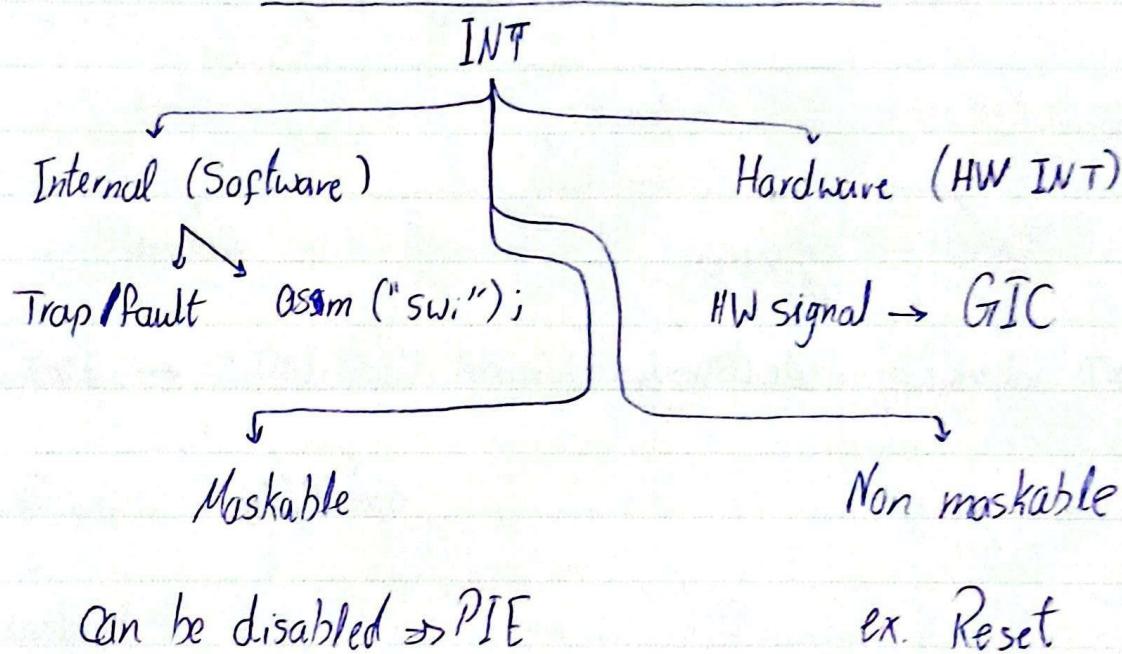
Startup code :

- ① load (.data) Flash $\xrightarrow{\text{copy}}$ RAM
- ② Allocate (.bss) in \Rightarrow Flash RAM
- ③ Initialize vector table on RAM (Dynamic)
- ④ Registers initialization ex. Stack register
- ⑤ Call main (Jump / branch)

★ Interrupt - Nesting



* Flag clearance (PIF)



* Instruction cycle with INT

- ① Complete execution of the current inst^{Assembly}
- ② Stop main program & clear PIF
- ③ Disable global Interrupt enable
- ④ HW
↓
 - Store /Push program counter → Stack
 - SW
compiler
 - Store /push register files
- ⑤ Context sw /stacking
- ⑥ Jump to vector table
- ⑦ Jump to ISR & execute it
- ⑧ Enable GIE by SW
- ⑨ Pop register files then PC \Rightarrow Unstacking

Func_calling

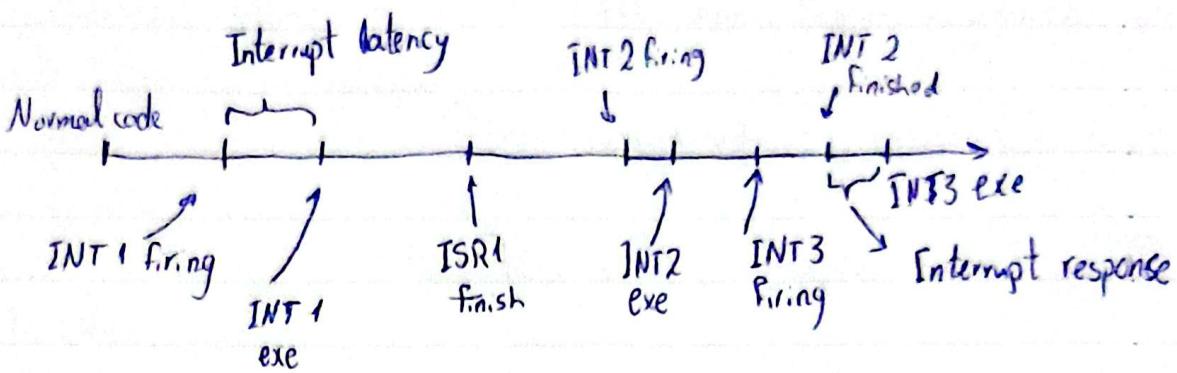
ISR

SW context switching

can be INT

HW context switching

Can't be INT \rightarrow self nesting
 \downarrow loss priority



- * INT response \Rightarrow Fixed time
- * Latency time $>$ Response time
- * Recovery time is time taken to unstack

