TAPiT

KHÓA HỌC
LẬP TRÌNH VI ĐIỀU KHIỂN

Giảng viên
NGUYỄN HUỲNH NHẬT THƯƠNG

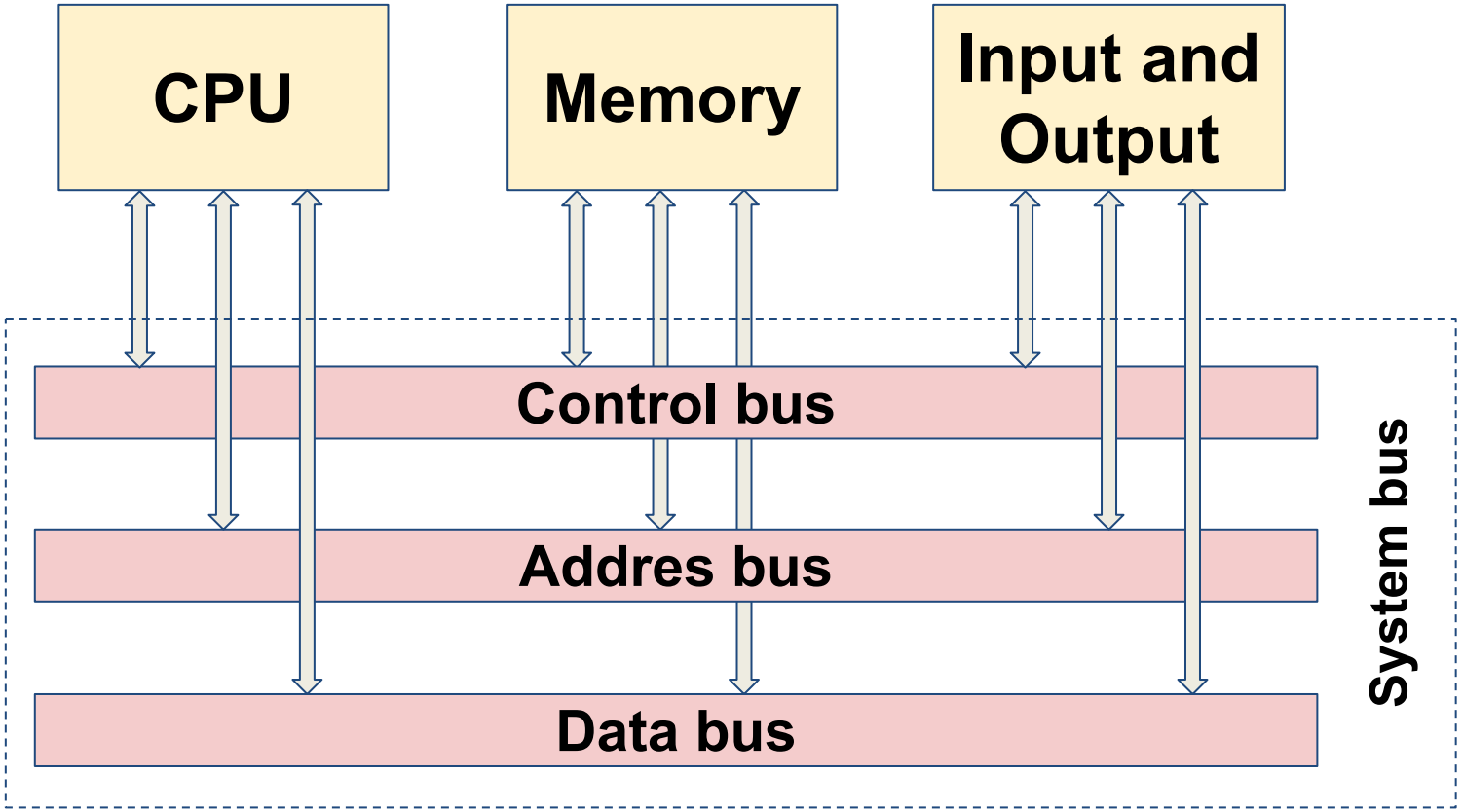
LỊCH HỌC:
19h30 - 22h00 thứ 4 và thứ 6

ĐỊA ĐIỂM:
Online qua nền tảng Zoom/Google Meet

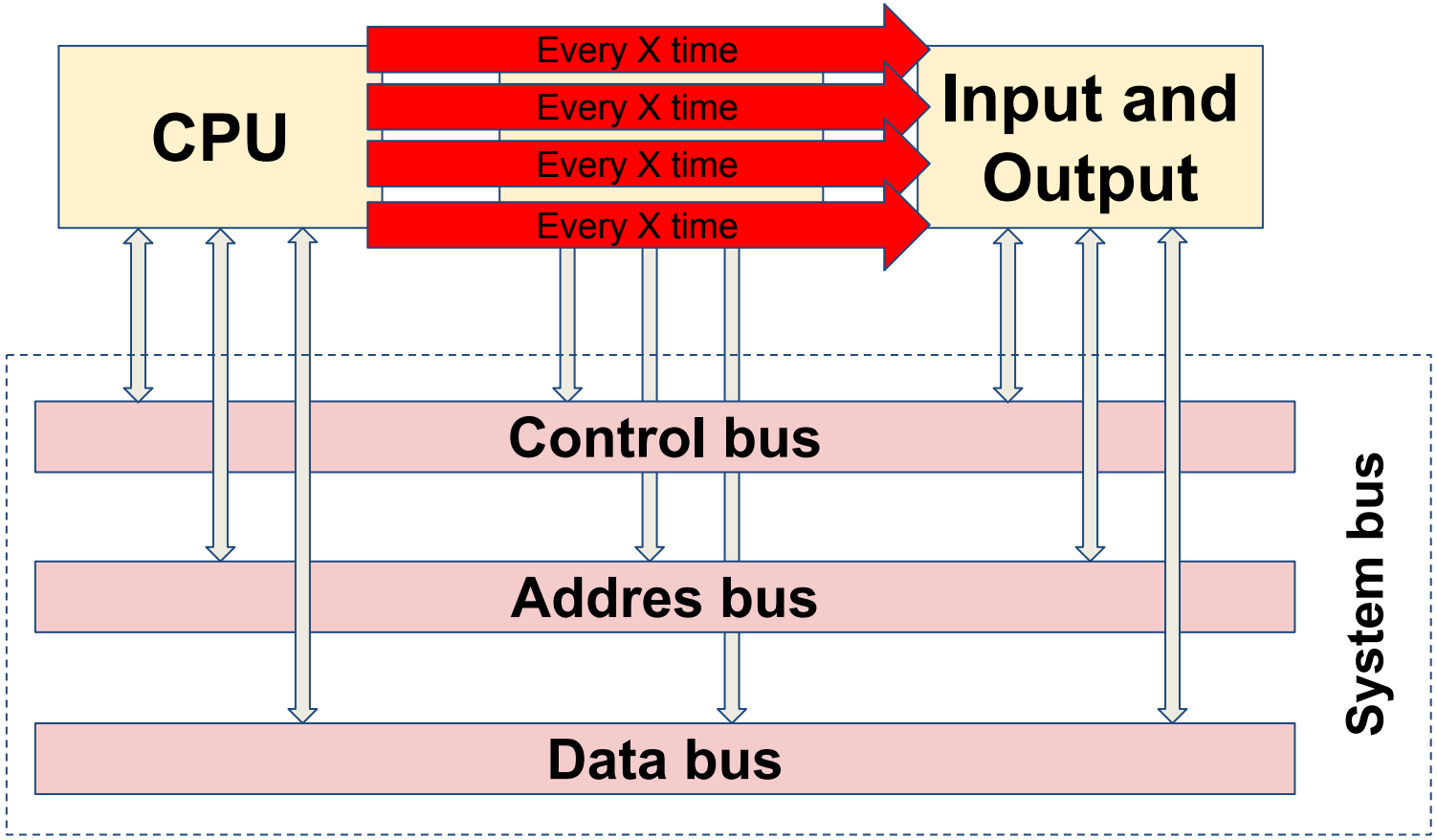
MODULE 3

- SYSTEM EXCEPTIONS & INTERRUPTS
- EXTERNAL INTERRUPT
- THỰC HÀNH, DEBUG
- PHÂN TÍCH DỰ ÁN THỰC TẾ

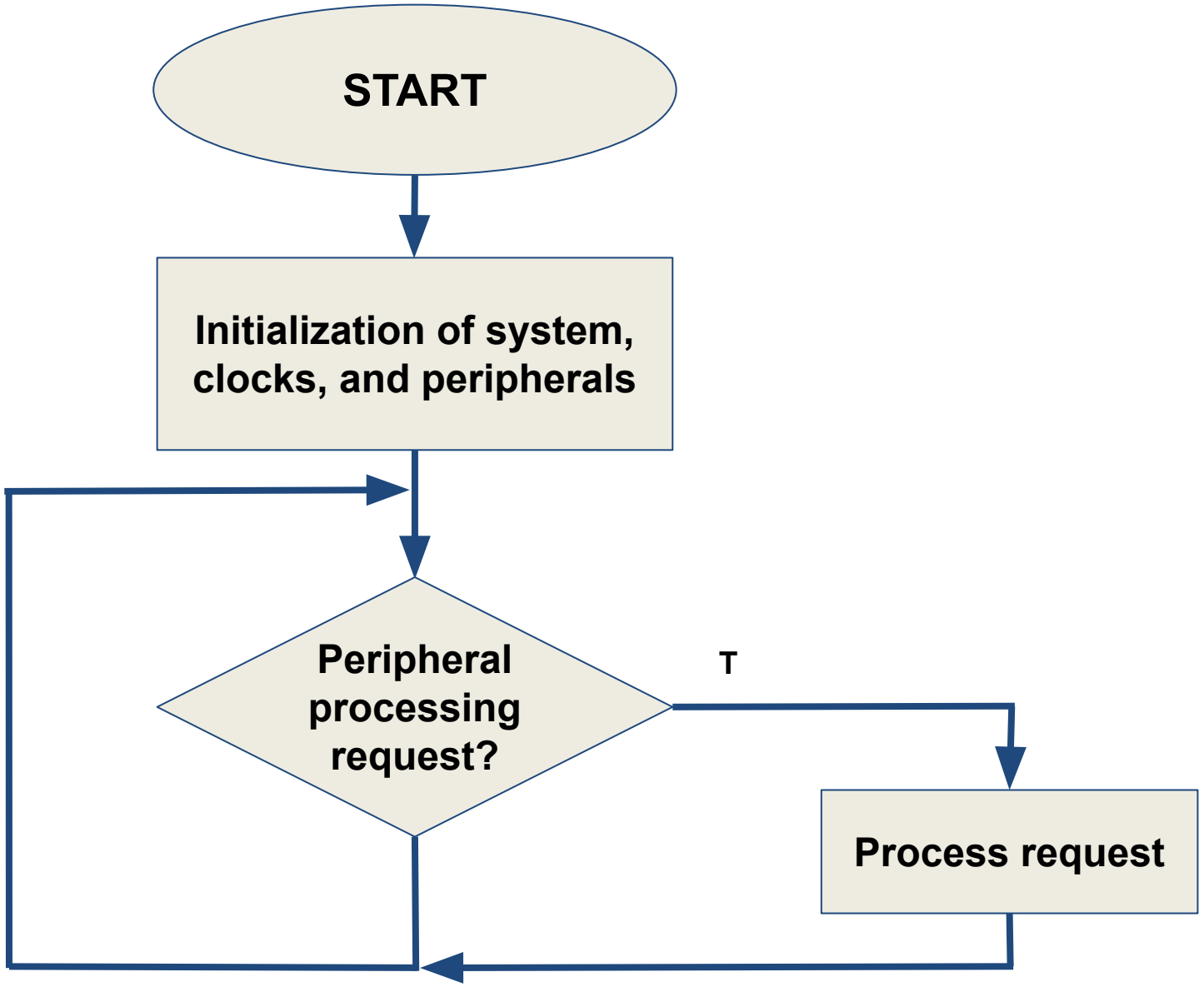
Kiến trúc cơ bản của Vi điều khiển



Hoạt động của cơ chế Polling



Polling: Lưu đồ thuật toán



Một số vấn đề gặp phải với Polling

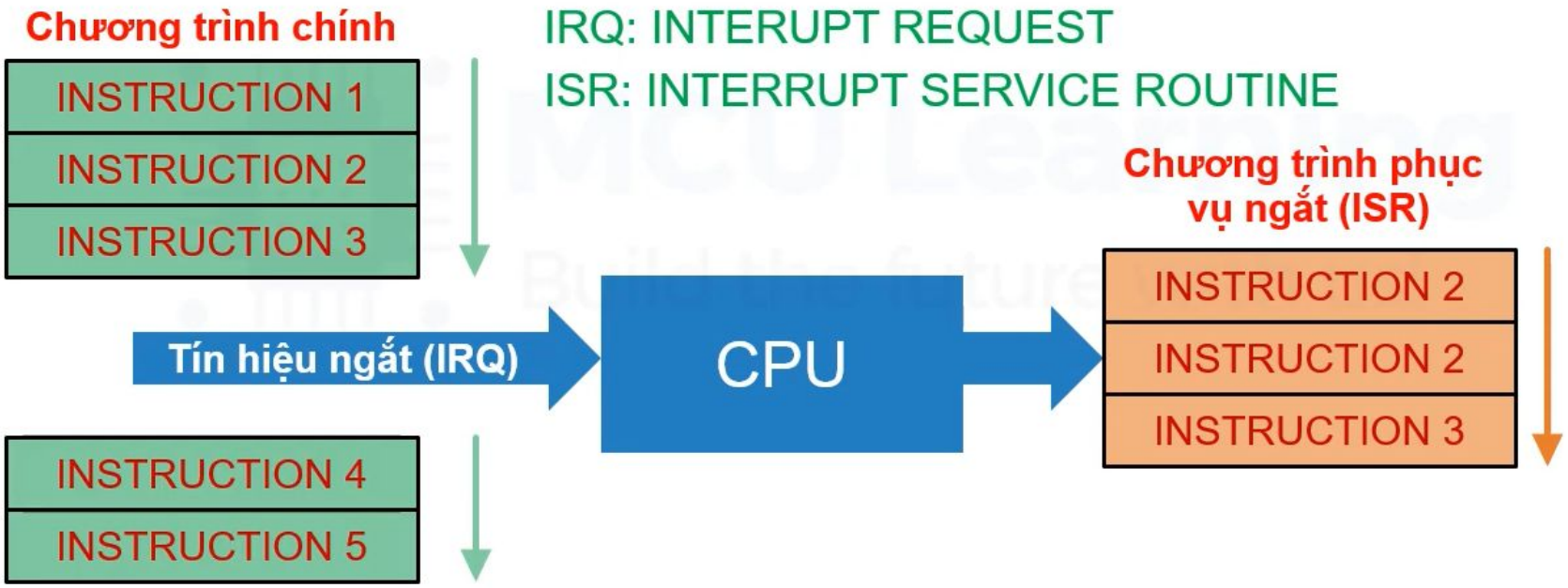
Exception

- Là một điều kiện làm thay đổi thứ tự thực hiện bình thường của một chương trình
- Gồm 2 loại: System Exceptions và Interrupts
- Vi xử lý ARM Cortex - M4 hỗ trợ 15 System Exception và 240 Interrupt
- https://developer.arm.com/documentation/dui0553/b/ (2.3 Exception model)

Tổng quan về Interrupt

Tương tác với ngoại vi: **Serving Peripherals**

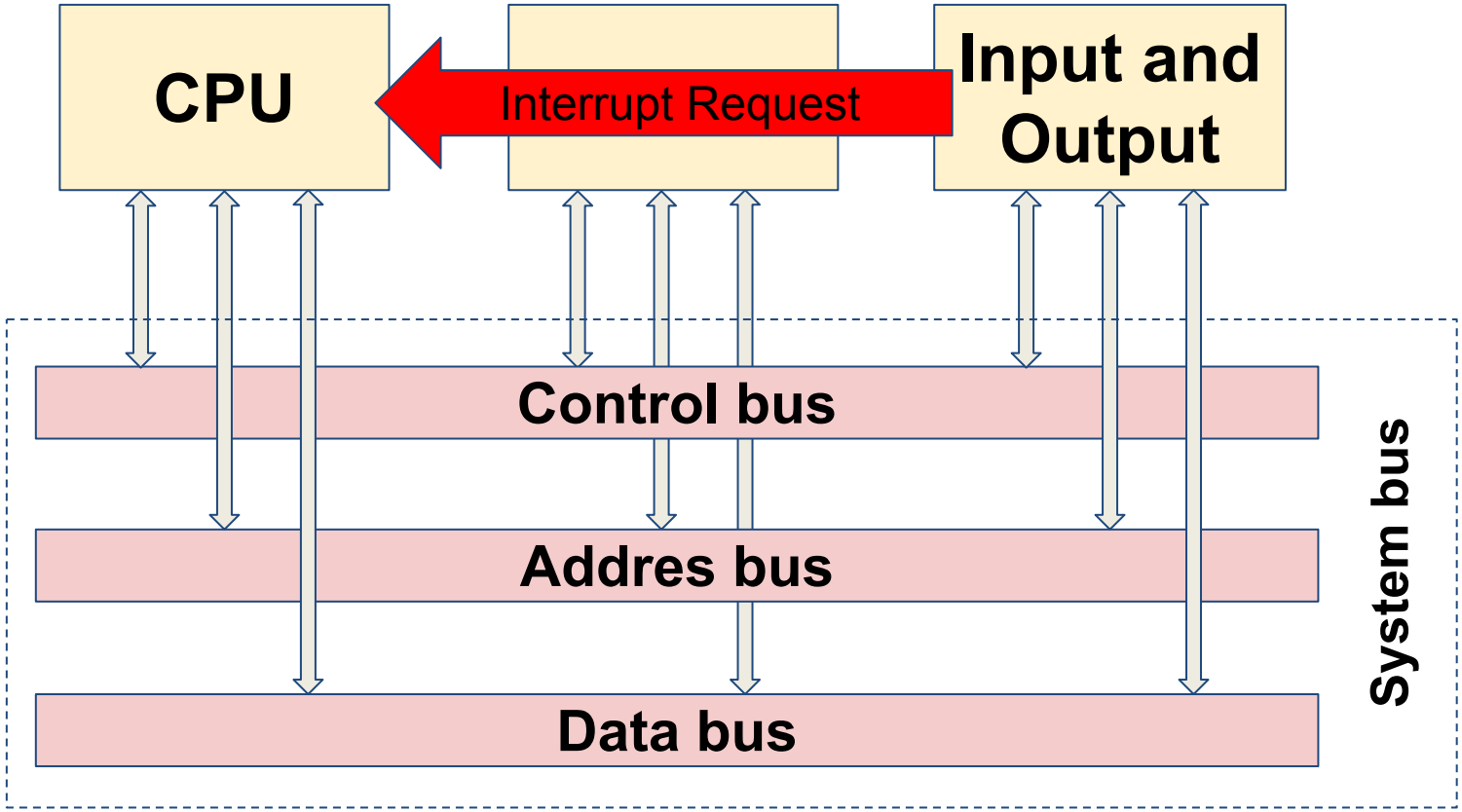
+ **Service by Interrupt** (ngắt) : Ngoại vi sẽ chủ động **gửi yêu cầu ngắt(IRQ)** đến VXL mỗi khi có **sự kiện** để VXL **thực hiện đoạn chương trình** phục vụ ngắt (ISR)



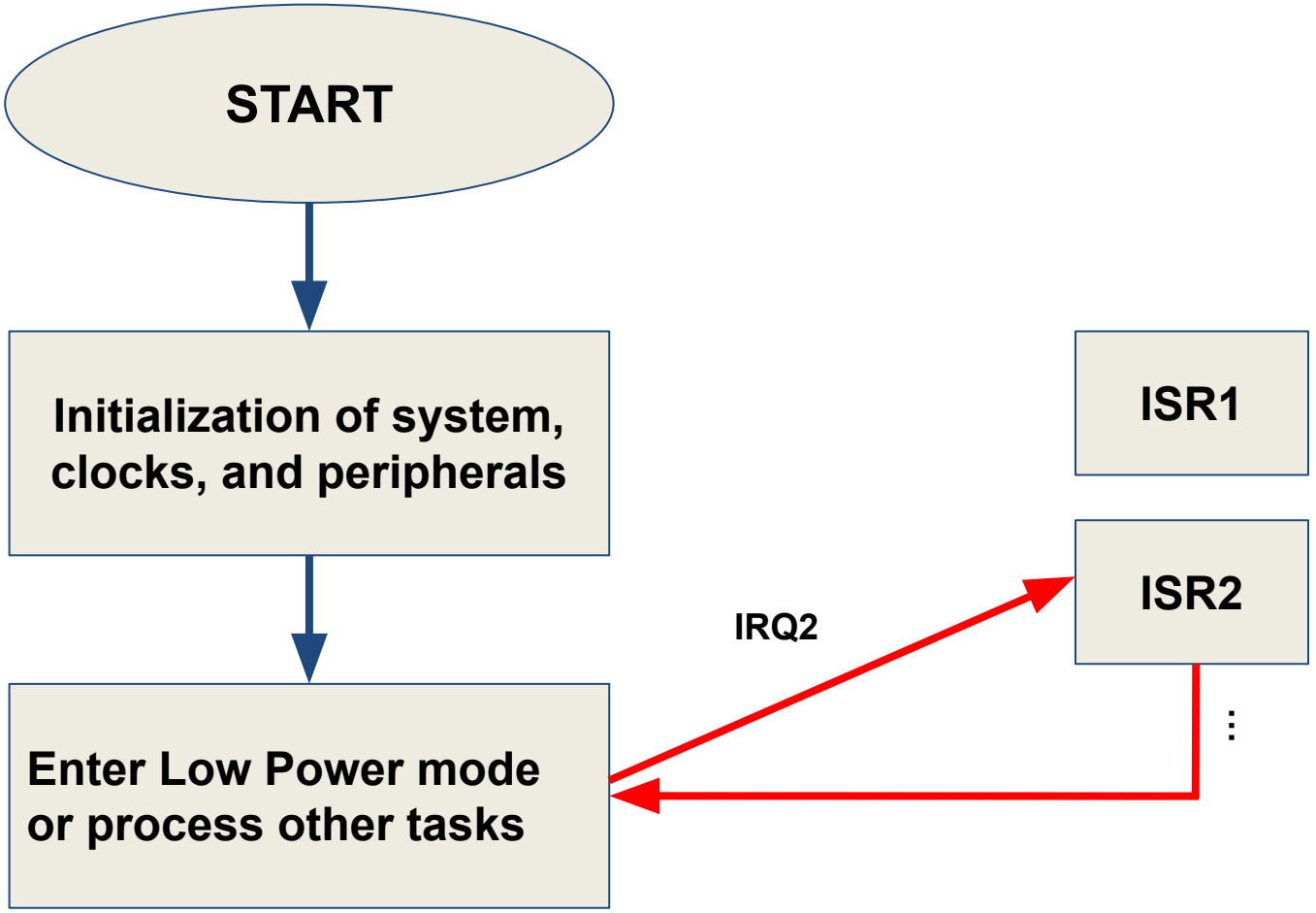
Interrupt

- Các tín hiệu Interrupt được tạo ra bởi các ngoại vi của vi điều khiển và được gửi đến vi xử lý. Các tín hiệu này được gọi là Interrupt Request (IRQ)
- Vi xử lý tạm ngưng thực thi chương trình bình thường để thực thi một đoạn chương trình đặc biệt là Interrupt Service Routine (ISR) – Interrupt Handler
- Các tín hiệu yêu cầu Interrupt (IRQ) được gửi vào khối NVIC của vi xử lý
- https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303x6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf (14.1.3)

Hoạt động của cơ chế Interrupt



Interrupt: Lưu đồ thuật toán



Nested VIC – Bộ xử lý ngắt

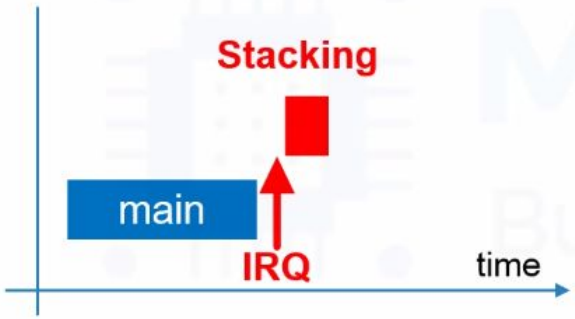
- Là một ngoại vi của lõi vi xử lý ARM-Cortex M
- Cấu hình **enable/disable/pend** các ngắt
- Đọc trạng thái của các ngắt (active/pend)
- Cấu hình độ ưu tiên các ngắt
- https://developer.arm.com/documentation/100166/0001/Functional-Description/About-the-functions?lang=en

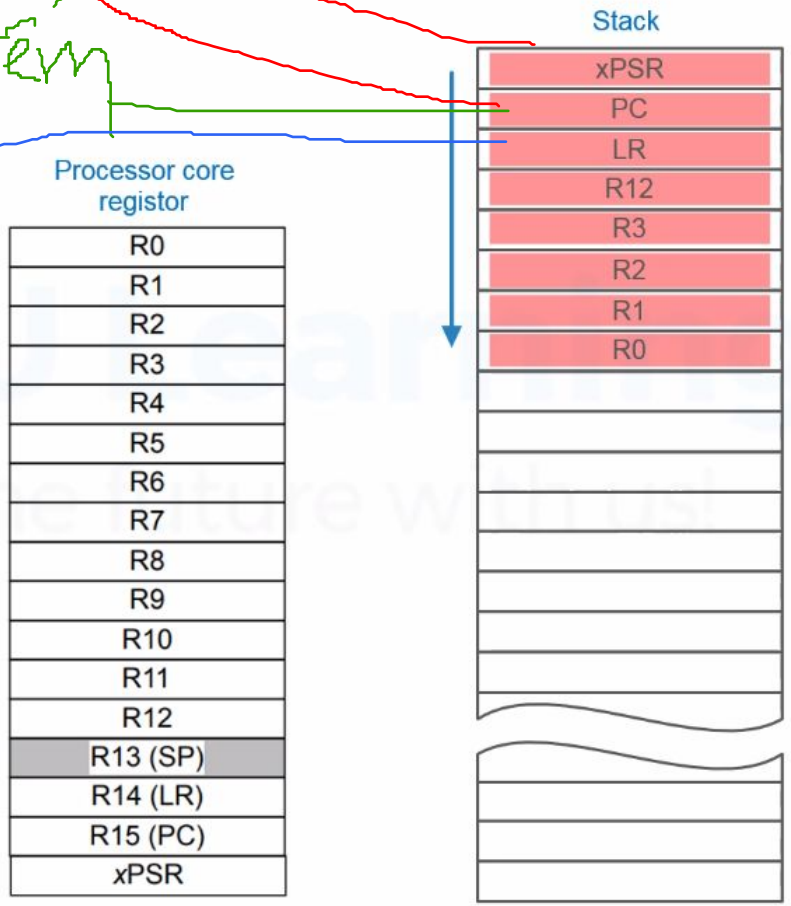
Quy trình vào và thoát ngắt của CPU

- Stacking và Unstacking

Stacking: quá trình lưu ngữ cảnh.

- Stacking

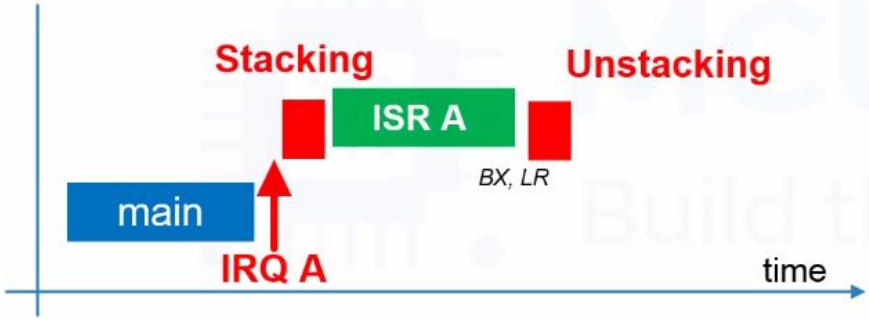


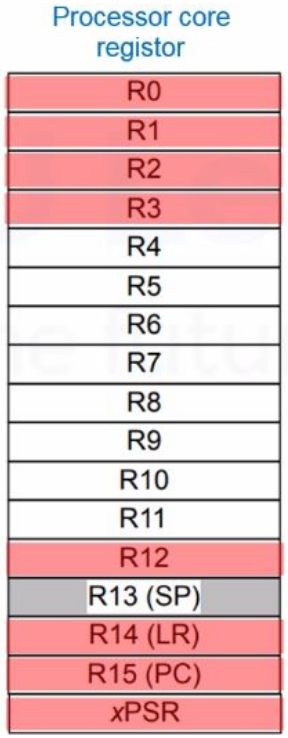


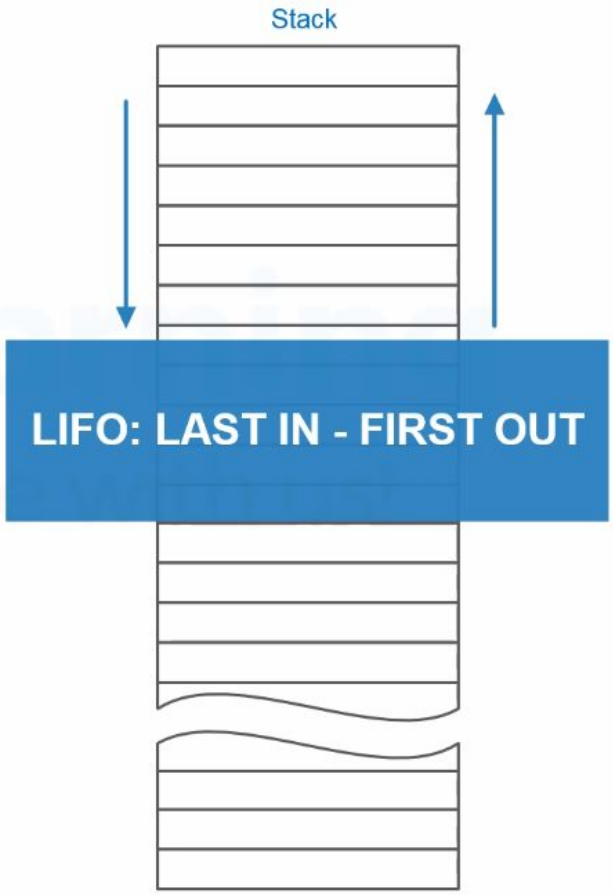
SRAM

FILO

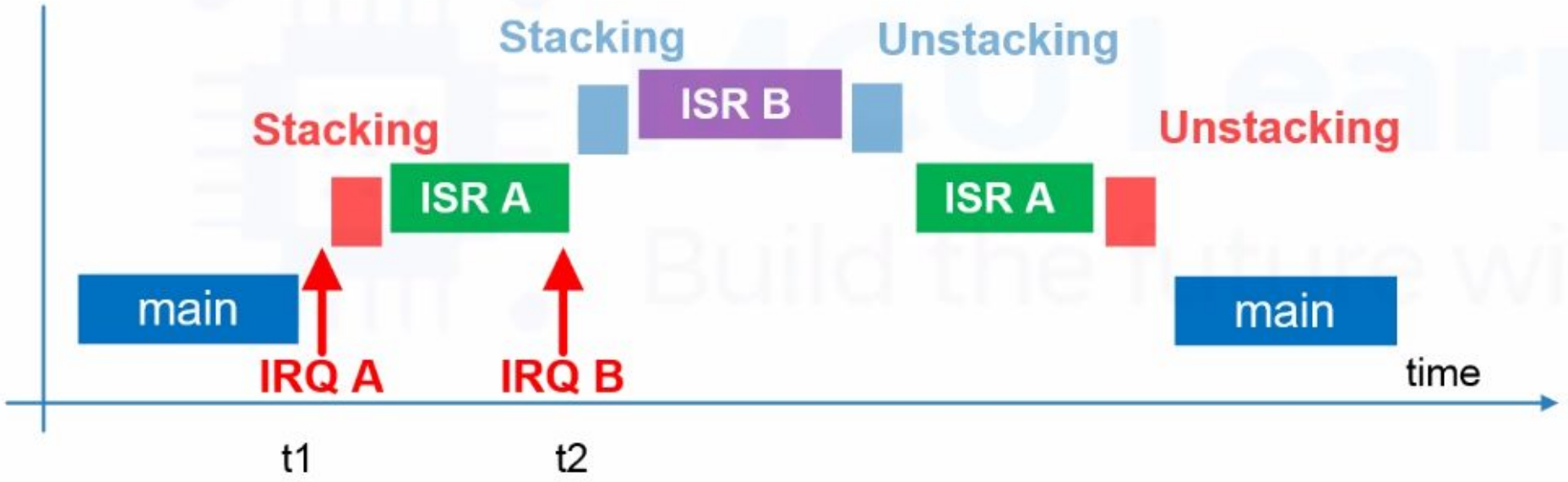
- Unstacking


Stacking
Unstacking
ISR A
BX, LR
main
IRQ A
time


Processor core registor
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13 (SP)
R14 (LR)
R15 (PC)
xPSR

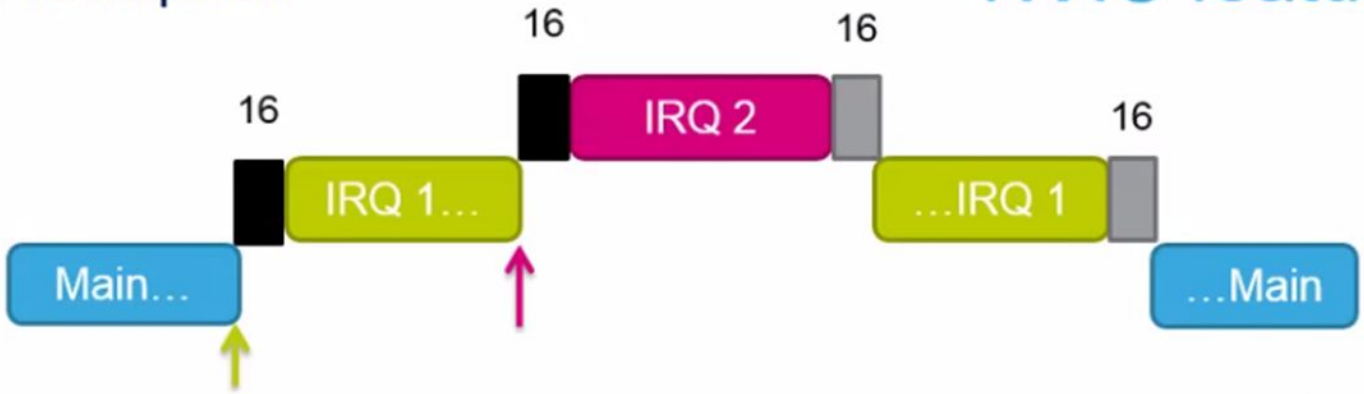

Stack
LIFO: LAST IN - FIRST OUT

Stacking và Unstacking khi ngắt chồng ngắt (nested)



Interrupts
NVIC features

- Preemption

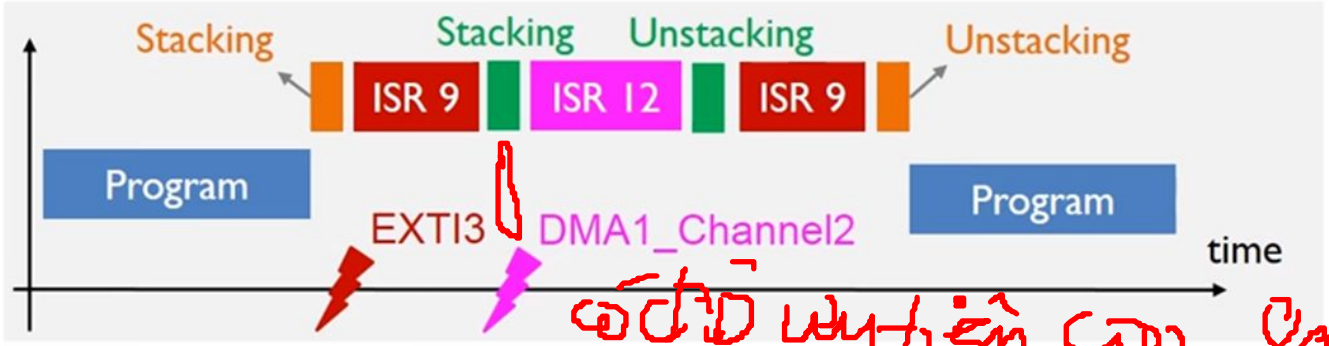


- Tail-chaining

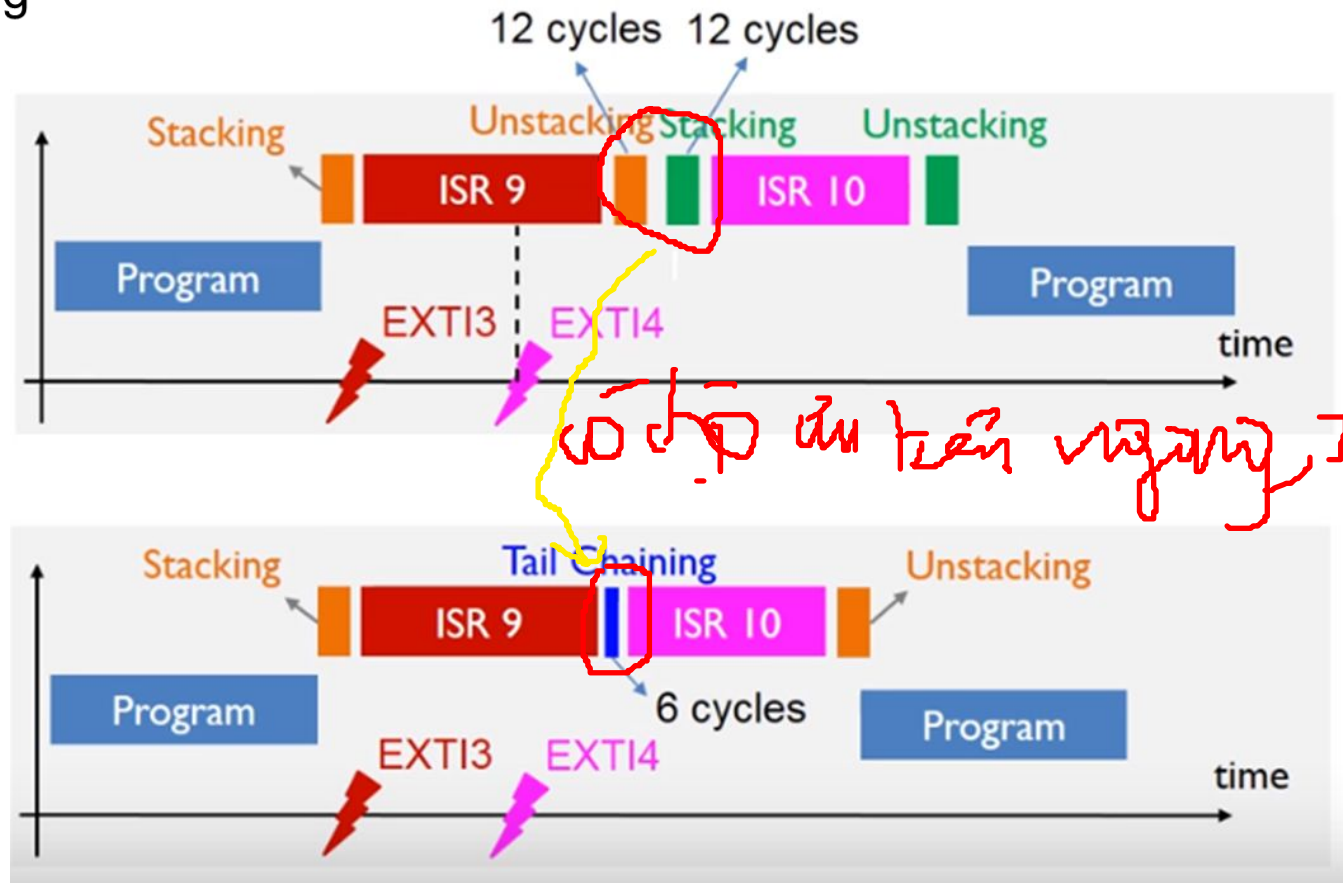
Priority:
IRQ2 > **IRQ1**

- Late arrival

Preemption



Tail chaining



Độ ưu tiên Priority

Trong thực tế, có những trường hợp sau:
1 ngắt đang thực thi thì xuất hiện 1 yêu cầu ngắt khác.
(Ngắt EXTI đang được thực thi thì có yêu cầu ngắt từ System Timer)

Yêu cầu ngắt mới có độ ưu tiên thấp hơn ngắt đang thực thi
=> Phải chờ (ở trạng thái Pending)
Yêu cầu ngắt mới có độ ưu tiên bằng ngắt đang thực thi
=> Phải chờ (ở trạng thái Pending)
Yêu cầu ngắt mới có độ ưu tiên cao hơn ngắt đang thực thi
=> Chiếm dụng ngắt (thực thi ngắt mới,trạng thái active, ngắt cũ sẽ ở trạng thái inactive)

Các trạng thái:
pending: Chưa được chấp nhận xử lý, đang chờ
active: Đang được phục vụ, ISR đang được thực thi
inactive: Đã được chấp nhận xử lý rồi, đã thực thi rồi nhưng bị giành quyền bởi 1 ngắt khác có độ ưu tiên cao hơn.

Hãng ST dùng 4 bit để cài đặt độ ưu tiên cho các ngắt
=> Có 16 độ ưu tiên khác nhau
Và 4 bit này được chia làm 2 nhóm: Preemption, Sub
Preemption: Được xét trước, nếu có độ ưu tiên preemtion cao hơn (giá trị nhỏ hơn) thì được chiếm dụng ngắt
Khi 2 preemption bằng nhau, thì xét sub-priority, nếu có độ ưu tiên sub cao hơn (giá trị sub nhỏ hơn) thì được chiếm dụng ngắt.
Nếu cả preemption và sub ngang nhau thì đến sau phải chờ.

External Interrupt (Ngắt ngoài)

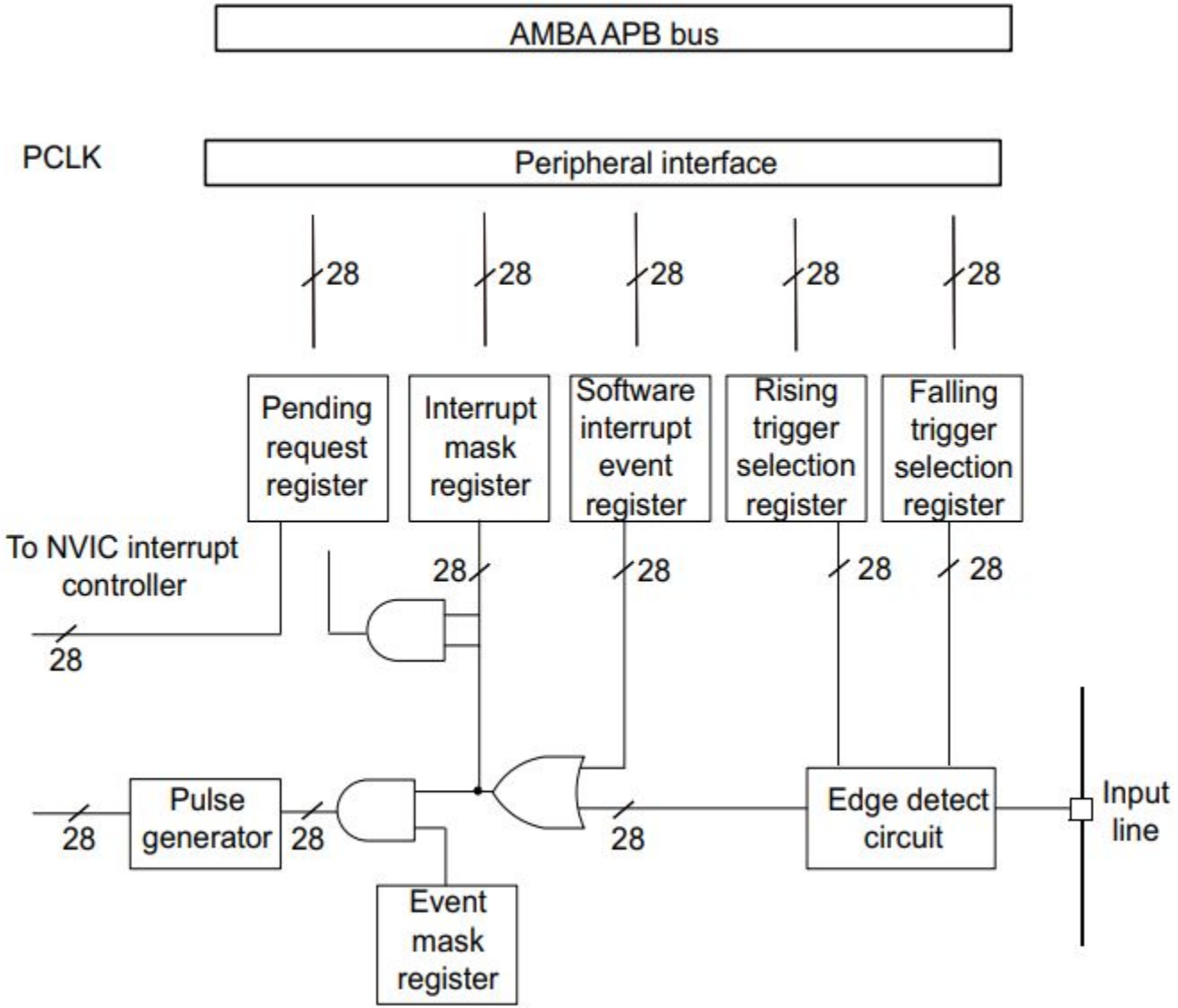
Tín hiệu từ bên ngoài tác động lên chân vi điều khiển như nút nhấn, cảm biến, module truyền nhận dữ liệu, vi điều khiển khác, IC chức năng khác.

+ Sườn lên RISING
+ Sườn xuống FALLING
+ Cả hai CHANGE

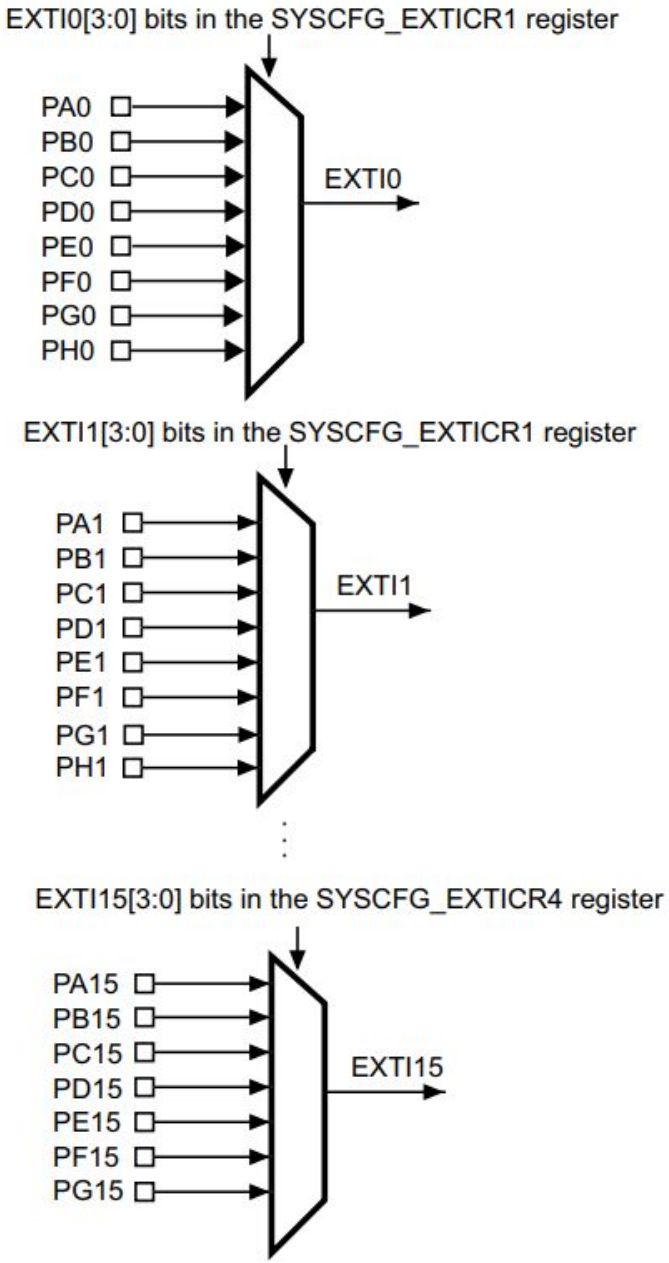
Extended interrupts and events controller (EXTI)

Xem lý thuyết tại đây: https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303x6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf (Mục 14.2)

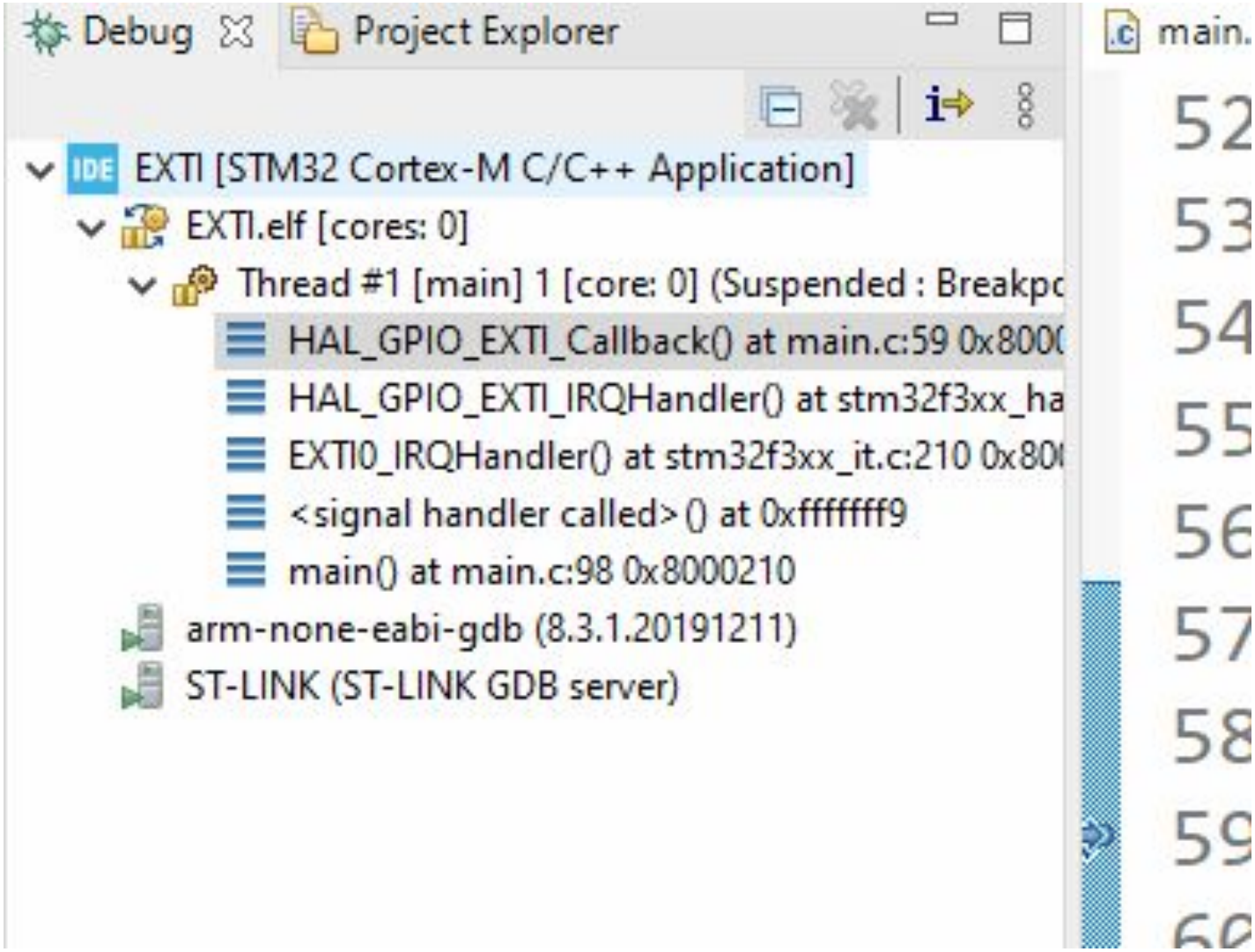
Figure 50. External interrupt/event block diagram



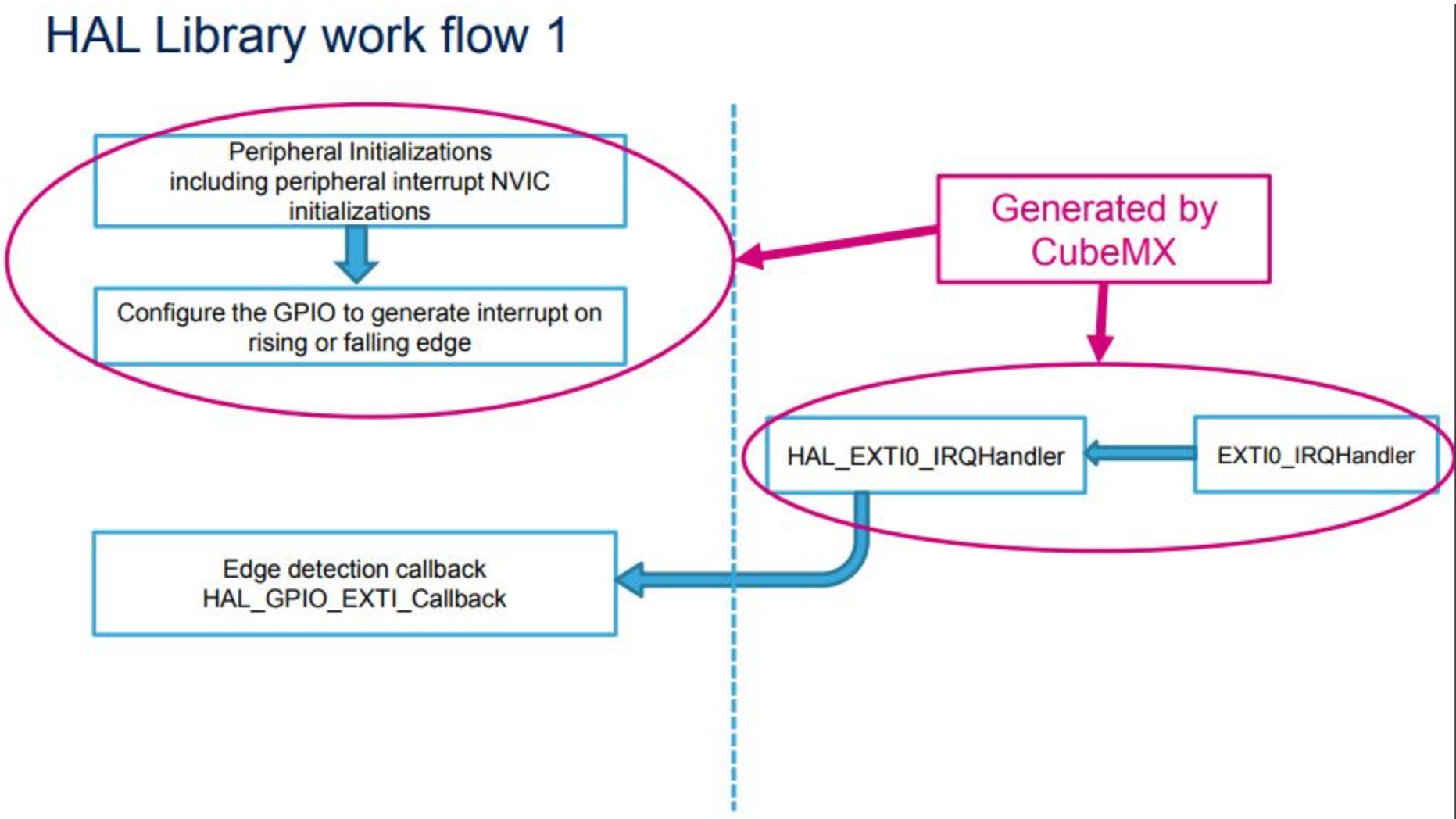
Các vấn đề cần quan tâm



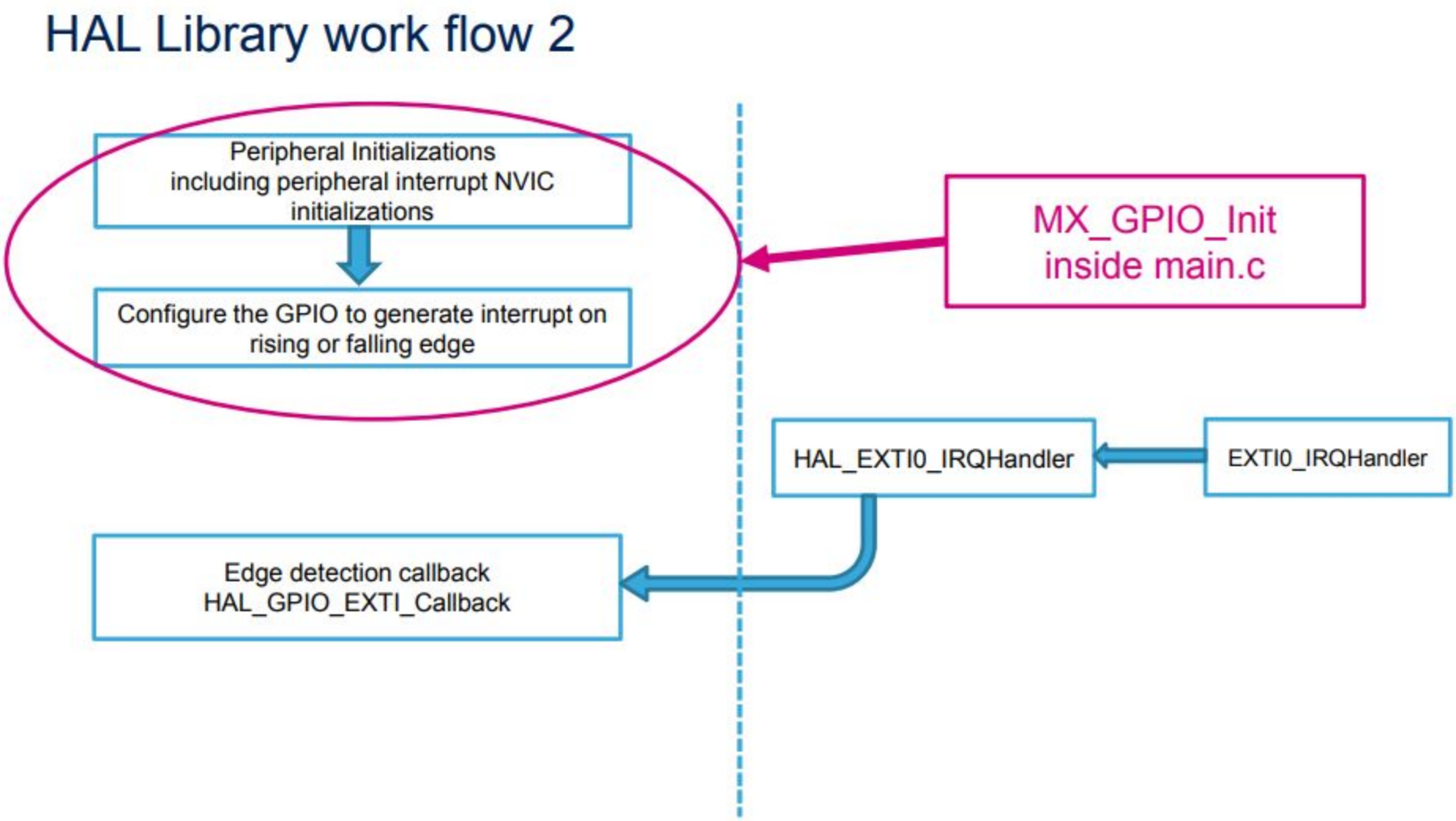
- Cấu hình độ ưu tiên của các ngắt
- Cấu hình cho phép ngoại vi sinh ngắt phù hợp vì có thể có nhiều nguồn sinh ra yêu cầu ngắt
- Viết chương trình ISR tương ứng với yêu cầu ngắt IRQ


Debug
Project Explorer
main.
EXTI [STM32 Cortex-M C/C++ Application]
EXTI.elf [cores: 0]
Thread #1 [main] 1 [core: 0] (Suspended : Breakpc
HAL_GPIO_EXTI_Callback() at main.c:59 0x800
HAL_GPIO_EXTI_IRQHandler() at stm32f3xx_ha
EXTI0_IRQHandler() at stm32f3xx_it.c:210 0x80
<signal handler called>() at 0xffffffff9
main() at main.c:98 0x8000210
arm-none-eabi-gdb (8.3.1.20191211)
ST-LINK (ST-LINK GDB server)
52
53
54
55
56
57
58
59

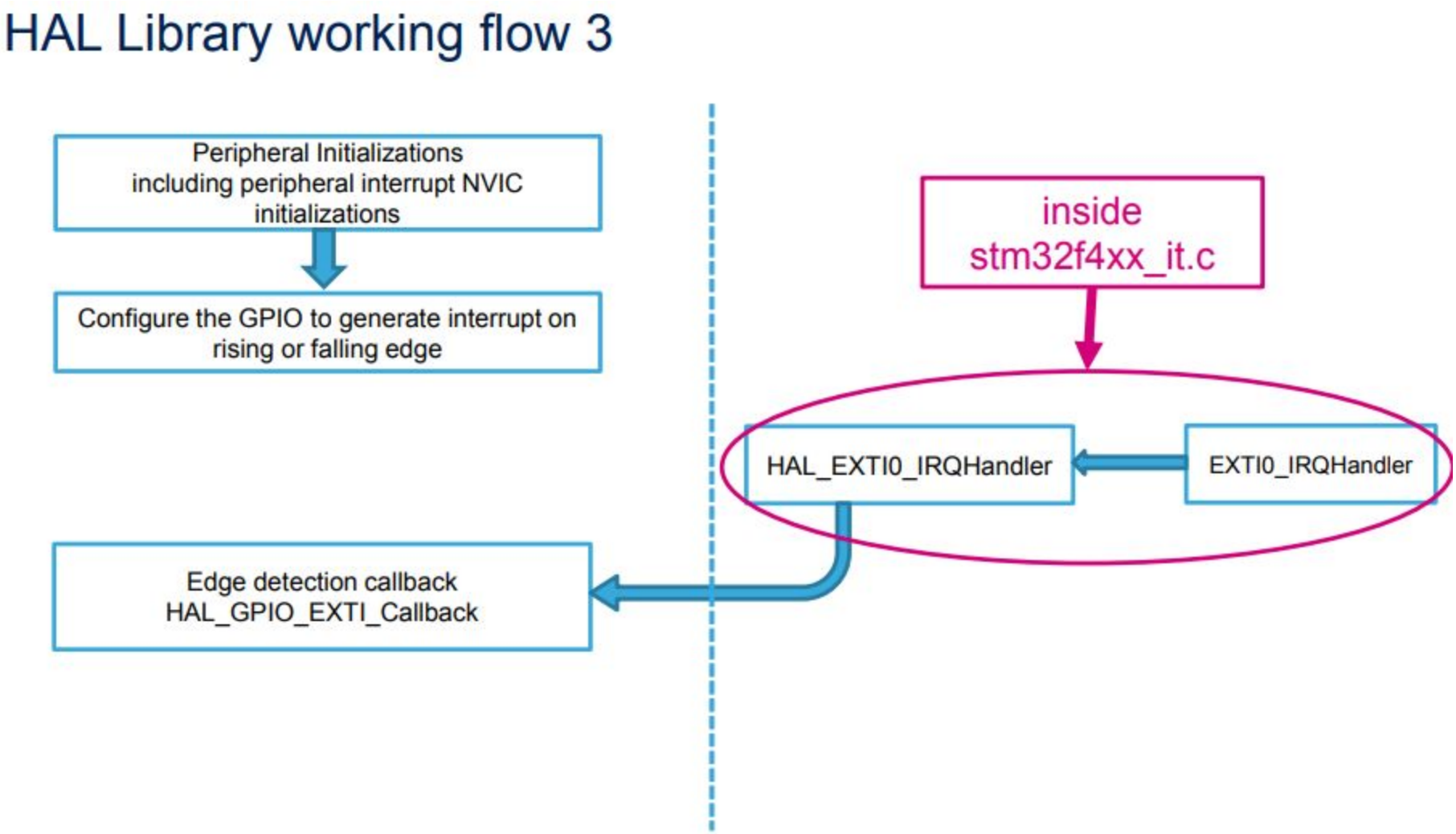
HAL Library work flow



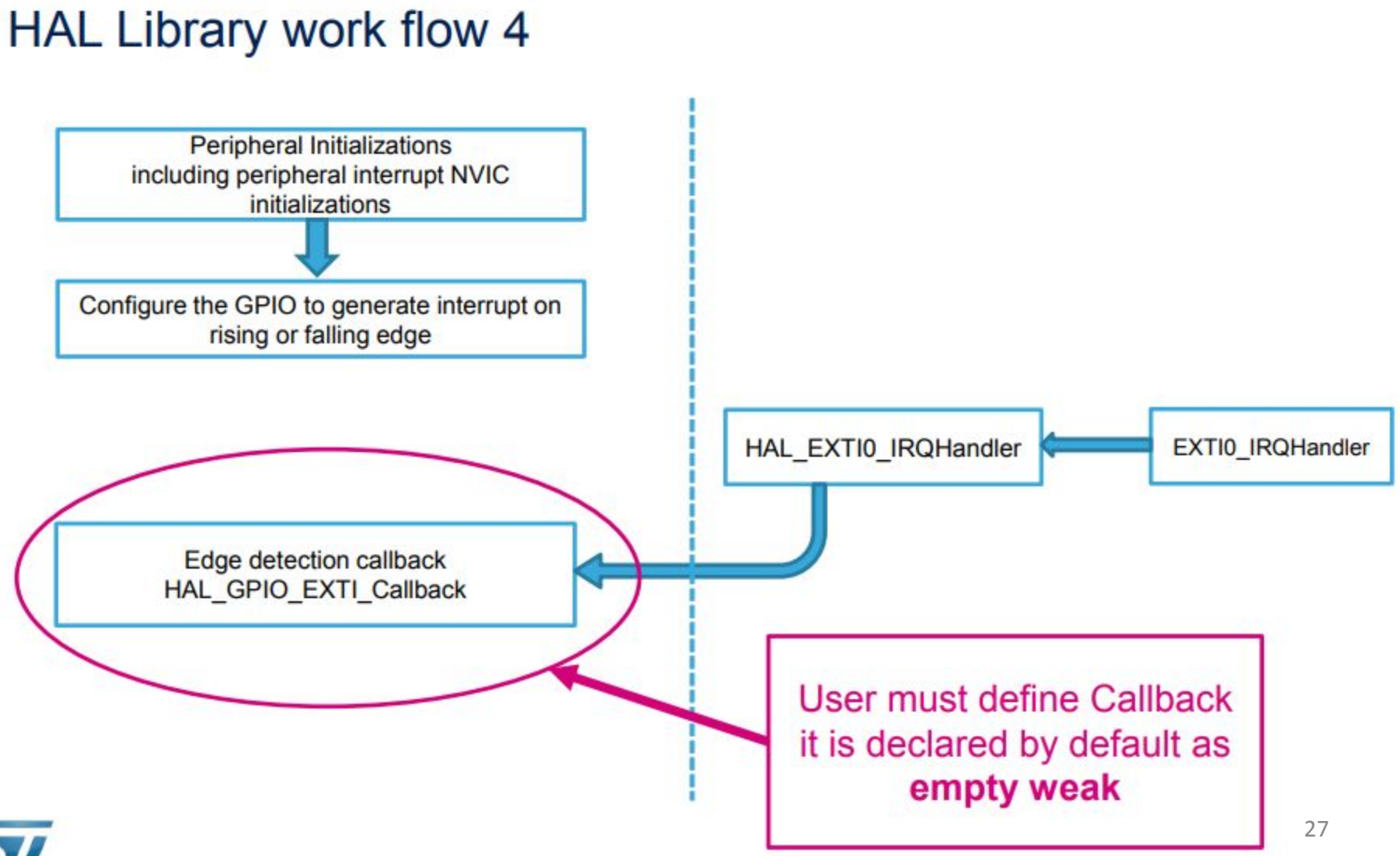
HAL Library work flow



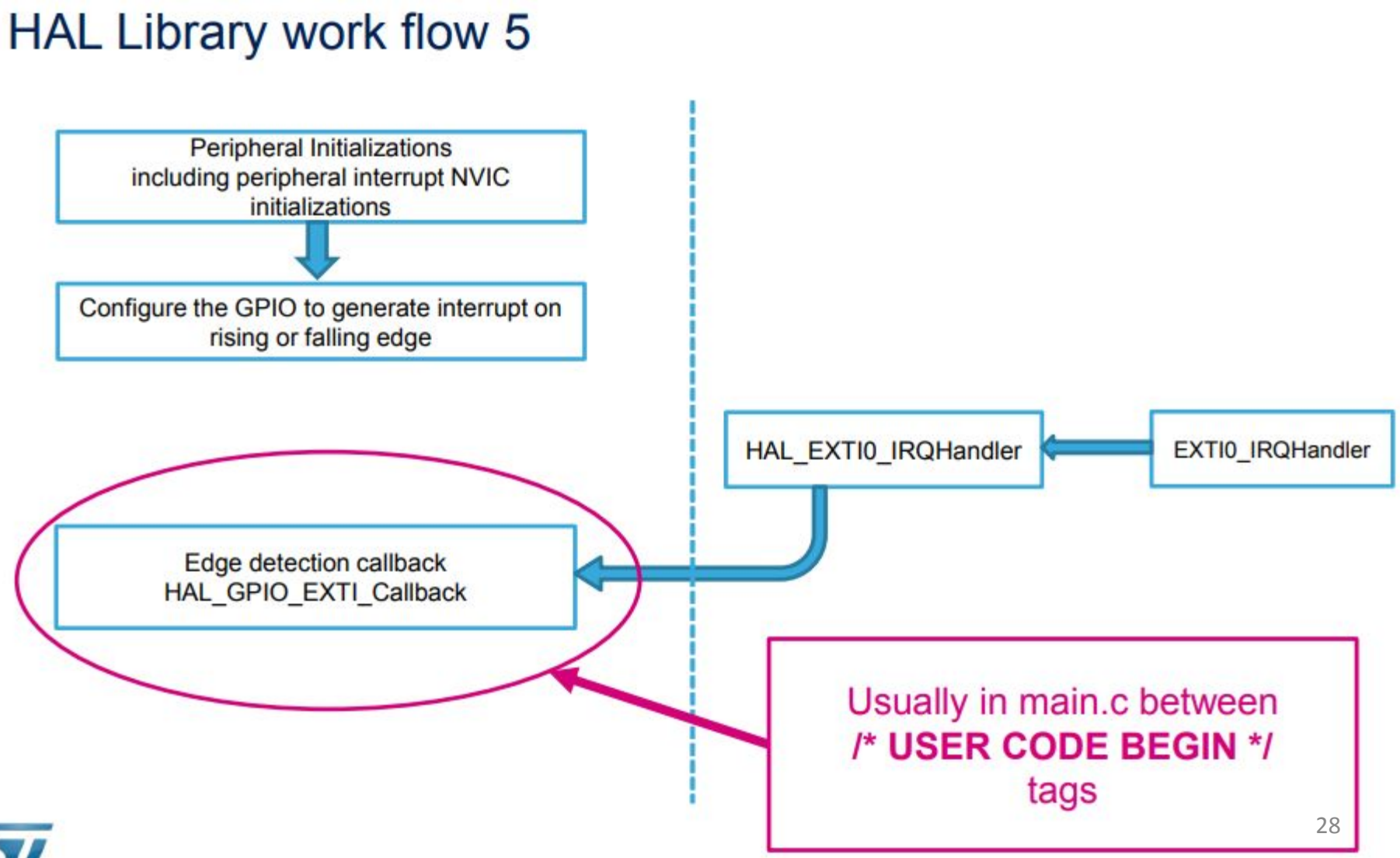
HAL Library work flow



HAL Library work flow



HAL Library work flow



HAL Library work flow