STM32WL5- EXTI

Extended interrupts and events controller

Revision 1.0

Hello, and welcome to this presentation of the STM32WL5 Extended Interrupts and Events Controller (EXTI).

Overview

- 47 events / interrupt lines
  - 23 configurable events
  - 24 direct events
- Independent masks and configuration
- Independent wakeup for both STM32WL5 CPUs.

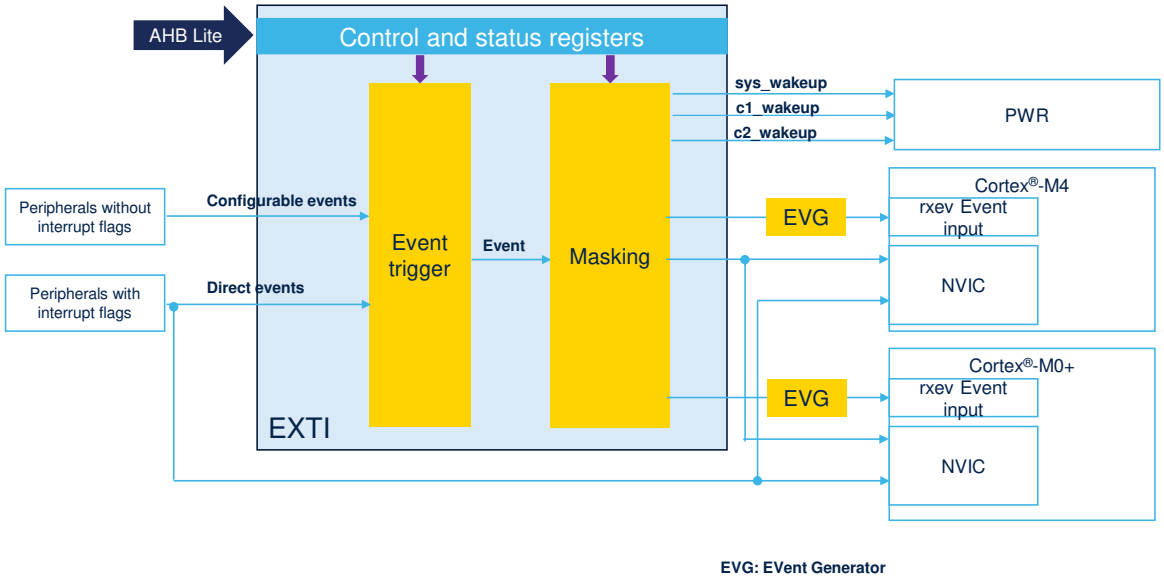
Application benefits

- Manage external and internal wakeup events and interrupts
- Provide pending flag for Configurable events.

life.augmented

The Extended interrupt and event controller (EXTI) provides 47 independent events, split into two categories – configurable events and direct events.
Applications benefit through smarter use of low-power modes, taking advantage of the STM32WL5's capability to wake up each CPU independently via external communication or requests.

EXTI block diagram



life.augmented

This is the block diagram of the extended interrupt and event controller .
Configurable events are generated by peripherals without interrupt capability, but which are able to issue a pulse. The EXTI controller provides interrupt detection, masking and software trigger.
Direct events are generated by peripherals supporting interrupt requests. In this case, the EXTI controller is used to generate events to the CPU and to request system wakeups.
The extended interrupt and event controller provides a single interrupt per configurable event to both CPUs. Individual CPU wakeups are provided to allow independent wakeup of both processors.

Key features

- Wake-up from Stop mode, interrupts and events generation
  - Independent interrupt and event masks

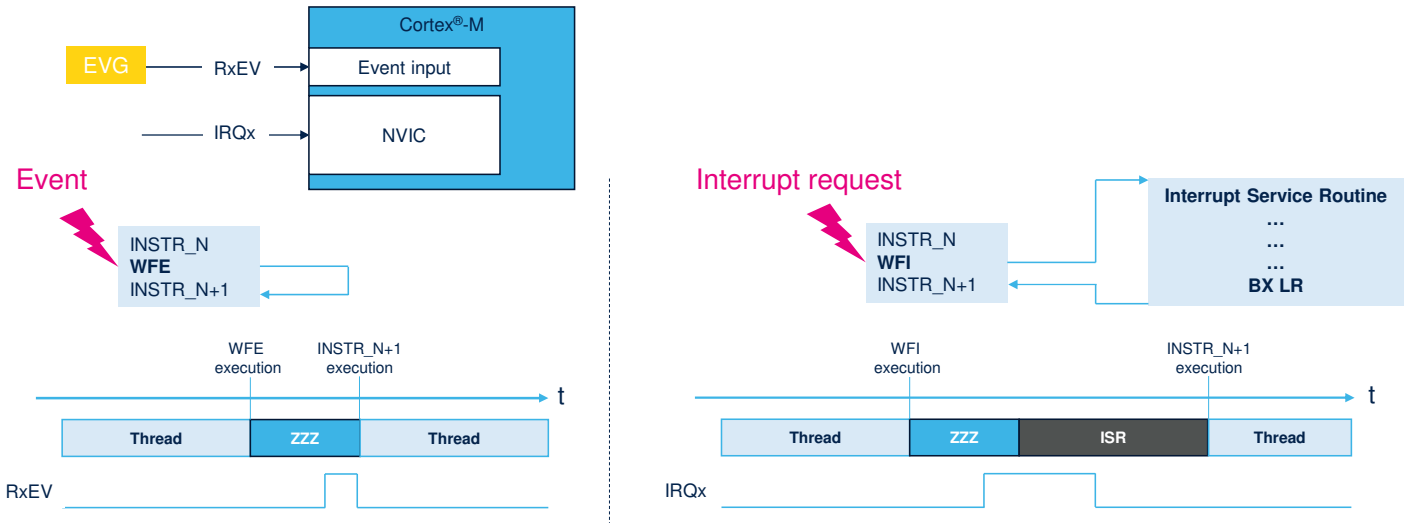
- Configurable events
  - Active edge selection
  - Single pending flag
  - Trigger-able by software
  - Linked to:
    - GPIO, PVD, PVM, COMP, CPU SEV, and sub-GHz radio busy.

- Direct events
  - Status flag provided by related peripheral
  - Linked to:
    - RTC, TAMP, I2C, USART, LPUART, LPTIM, sub-GHz radio, IPCC HSEM, Flash, and DEBUG

life.augmented

The extended interrupt and event controller can generate interrupts and event as well as wake up the processors from Stop modes.
Configurable events are linked with external interrupts from GPIOs, PVD, PVM, comparators COMP, CPU sent event, and sub-giga Hertz radio busy.
Direct events are linked with RTC, TAMP, I2C, USARTS, LPUART, LPTIM, sub-giga Hertz radio, IPCC, HSEM, Flash, and DEBUG.

Cortex®-m event vs interrupt



- Events are available on:
  - GPIO EXTI, RTC, TAMP, COMP, and CPU SEV

life.augmented

5

The Cortex®-M supports two ways to enter a low-power state:

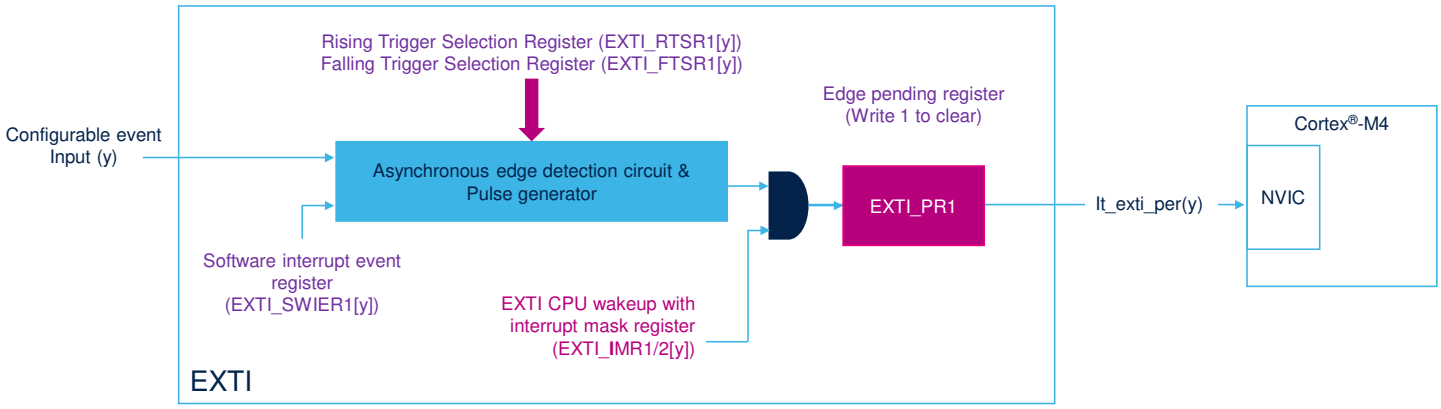
1. Executing the Wait For Event (WFE) instruction
2. Executing the Wait For Interrupt (WFI) instruction.

With WFE, the first instruction executed after a wake-up event is the next sequential one, INSTR_N+1 in the sequence on the left.
By implementing WFI, the processor jumps to the Interrupt Service Routine when an enabled interrupt request is received.

Note that an interrupt request is a WFE exit condition, but an event received on RXEV is not a WFI exit condition.
For the STM32WL5 series, the event generation is only available from the listed peripherals.

Interrupt generation

- Using configurable events as interrupt requests:



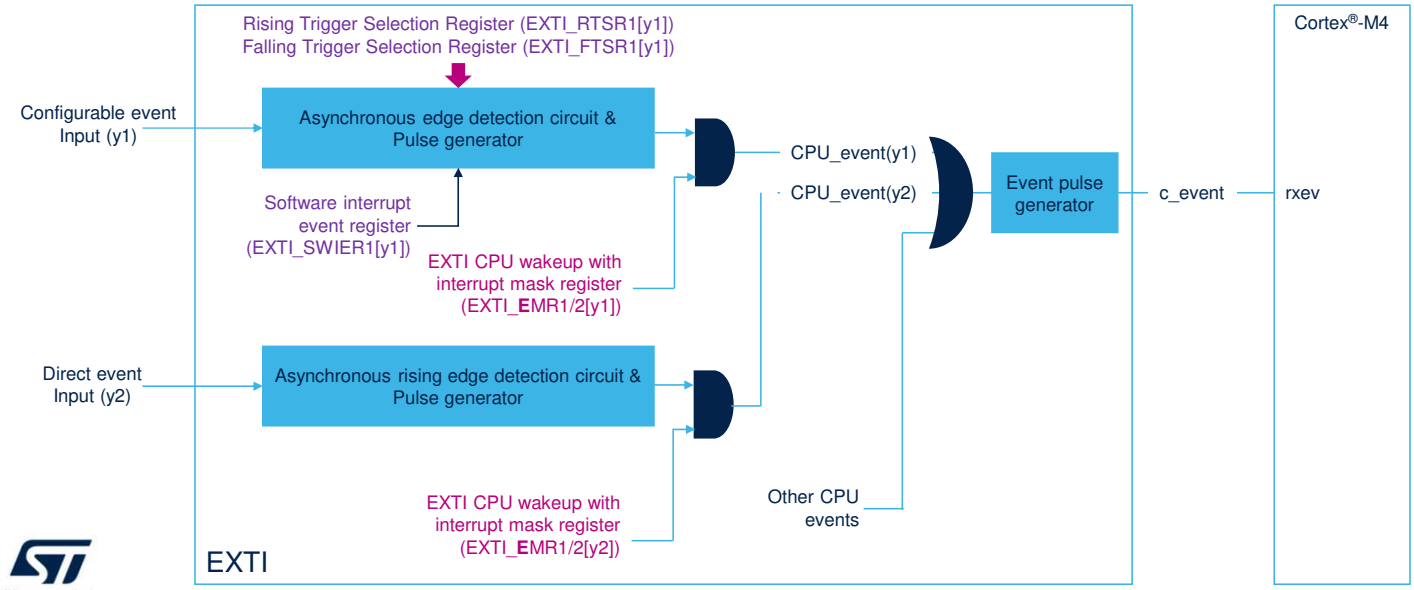
life.augmented

6

This figure aims to explain the various stages enabling the conversion of a configurable event active edge into an interrupt request.
The first stage is the asynchronous edge detection circuit configured by two registers EXTI_RTSR1 and EXTI_FTSR1. Any edge, possibly both, can be chosen.
The software can emulate a configurable event by setting the corresponding bit in the EXTI_SWIER register. The bit is auto-cleared by hardware.
An AND gate is used to mask or enable the generation of the interrupt to the CPU NVIC.
Finally, a flag is set in the EXTI_PR1 register when the interrupt is generated to the NVIC. This flag enables the software to determine the cause of the interrupt.
This flag is expected to be cleared by the interrupt service routine.

CPU event generation

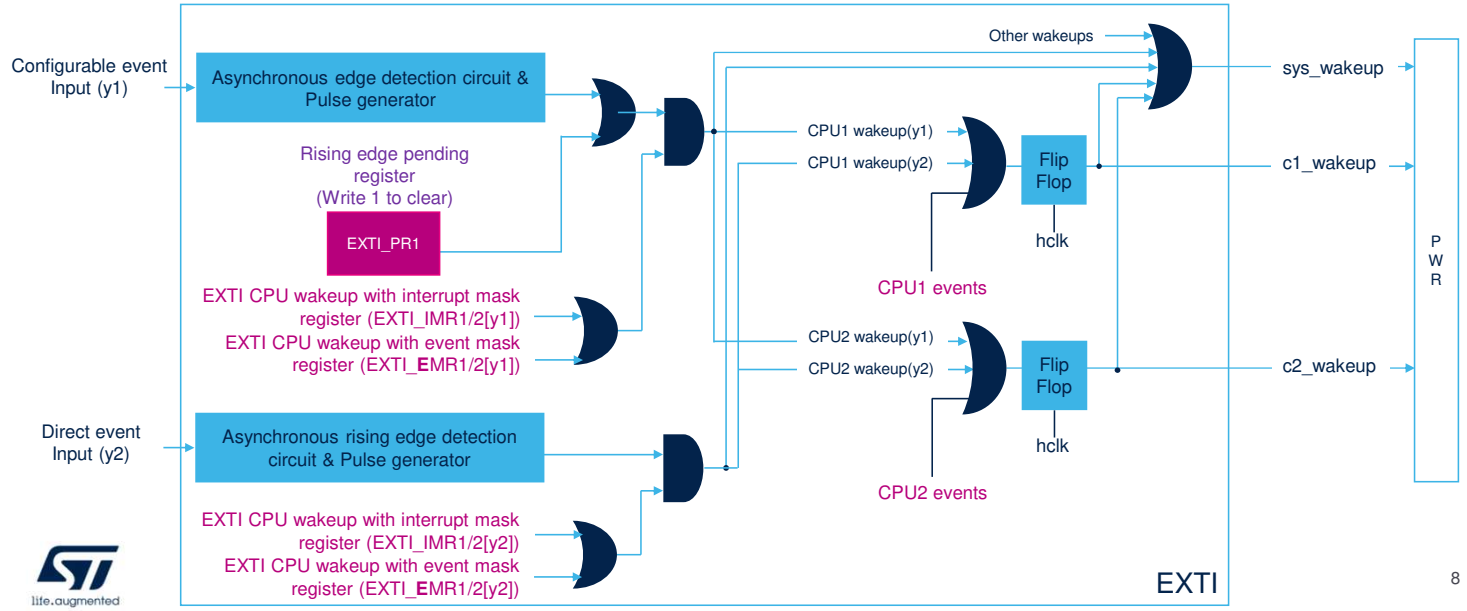
- Using configurable and direct events as CPU event request:



This figure aims to explain the various stages enabling the conversion of a configurable event active edge into a processor event.
Both Configurable and Direct events can be configured to issue events to the CPU, steered to its rxev input.
Unlike interrupt requests, the CPU has a unique event input, so all event requests are ORed together before entering the Event pulse generator.
The registers used to mask the generation of events are different from the ones used to mask the generation of interrupts: EXTI_EMR instead of EXTI_IMR.
The dual-core STM32WL5 microcontroller has independent event generation logic for each CPU.

Wakeup event generation

- Using configurable and direct events as core and system wakeup requests:



The CPUs wakeup signals generated by the EXTI block are connected to the PWR block and are used to wake up the system and CPU sub-systems bus clocks.
Both configurable and direct events are able to request a wakeup.
A wakeup occurs when an asynchronous edge detection circuit has detected an active edge.
Consequently, software is expected to clear the flag in the EXTI_PR1 register to disable the wakeup request when the source of the wakeup is a configurable event. For direct events, the flag is located in the peripheral unit.
These flags enable the software to find the cause of the wakeup.
The wakeup indication is asserted when either the interrupt or the event generation is enabled, see the OR gate combining EXTI_IMR and EXTI_EMR registers.
All CPU wakeup signals are ORed together and then ORed with the event requests.

sys_wakeup is asynchronous and wakes up the clocks. Once hclk is running, the synchronous c1_wakeup and or c2_wakeup is generated to wake up the respective CPU.

Direct event trigger logic CPU wakeup

- The direct events do not have an associated EXTI interrupt
  - The EXTI only wakes up the system and CPU sub-systems clocks and may generate a CPU wakeup event
  - The peripheral synchronous interrupt, associated with the direct wakeup event wakes up the CPU
- The EXTI direct event is able to generate a CPU event

life.augmented

A direct event is able through the EXTI controller to generate a CPU event and trigger a system wakeup.
The active edge of direct events is the rising edge.
Direct events do not rely on the EXTI controller to assert interrupt requests, because they have their dedicated lines to the NVIC.
Otherwise the same circuit as the one described in the previous slides are implemented. Direct events can be independently masked for event generation and interrupt generation. The interrupt mask in the EXTI controller is only used as a wakeup mask.

Related peripherals

- Refer to the training material for the following peripherals linked to the timers:
  - NVIC
    - Handling the CPU interrupt request.
  - Cortex®-M4
    - The CPU implements an exception mechanism used to handle both software and hardware exceptions.
  - Cortex®-M0+
    - The CPU implements an exception mechanism used to handle both software and hardware exceptions.

life.augmented

The Extended Interrupt and event Controller is linked with the Nested Vector Interrupt Controller of the Cortex-M4 CPU and Cortex-M0+ CPU. Please refer to the related presentations.

References

- For more details, please refer to the following documents:
  - PM0214 Programming manual for STM32F3, F4, L4 and L4+ series.
  - STM32WL5 MCU reference manuals

life.augmented

For detailed information, please refer to the programming manual for the STM32F3, F4, L4 and L4+ series and the reference manual of the STM32WL5 series.