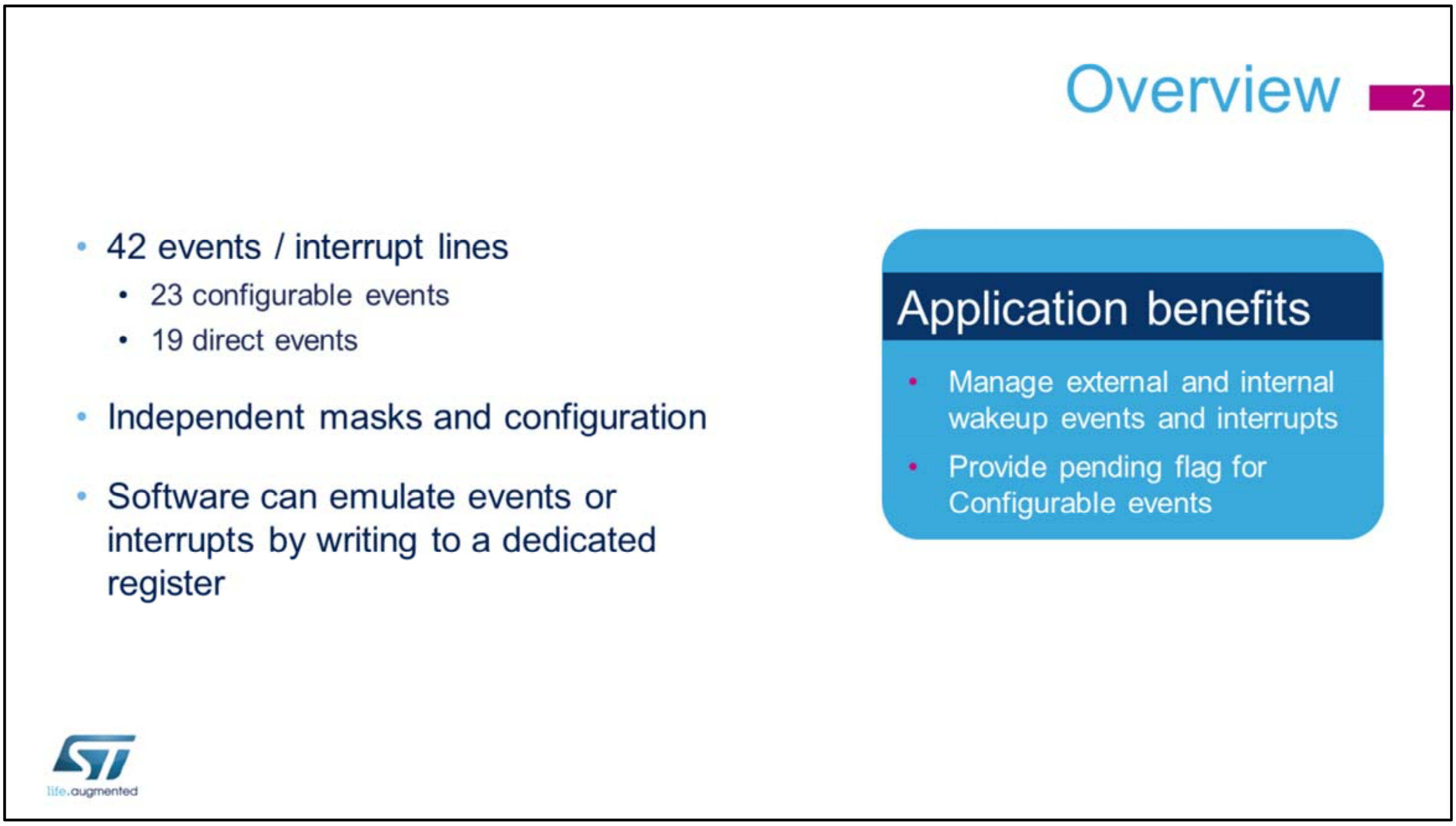
STM32L5- EXTI

Extended interrupts and events controller

Revision 1.0

life.augmented

Hello, and welcome to this presentation of the STM32L5 Extended Interrupts and Events Controller.
We will be presenting the features of the EXTI controller.



Overview

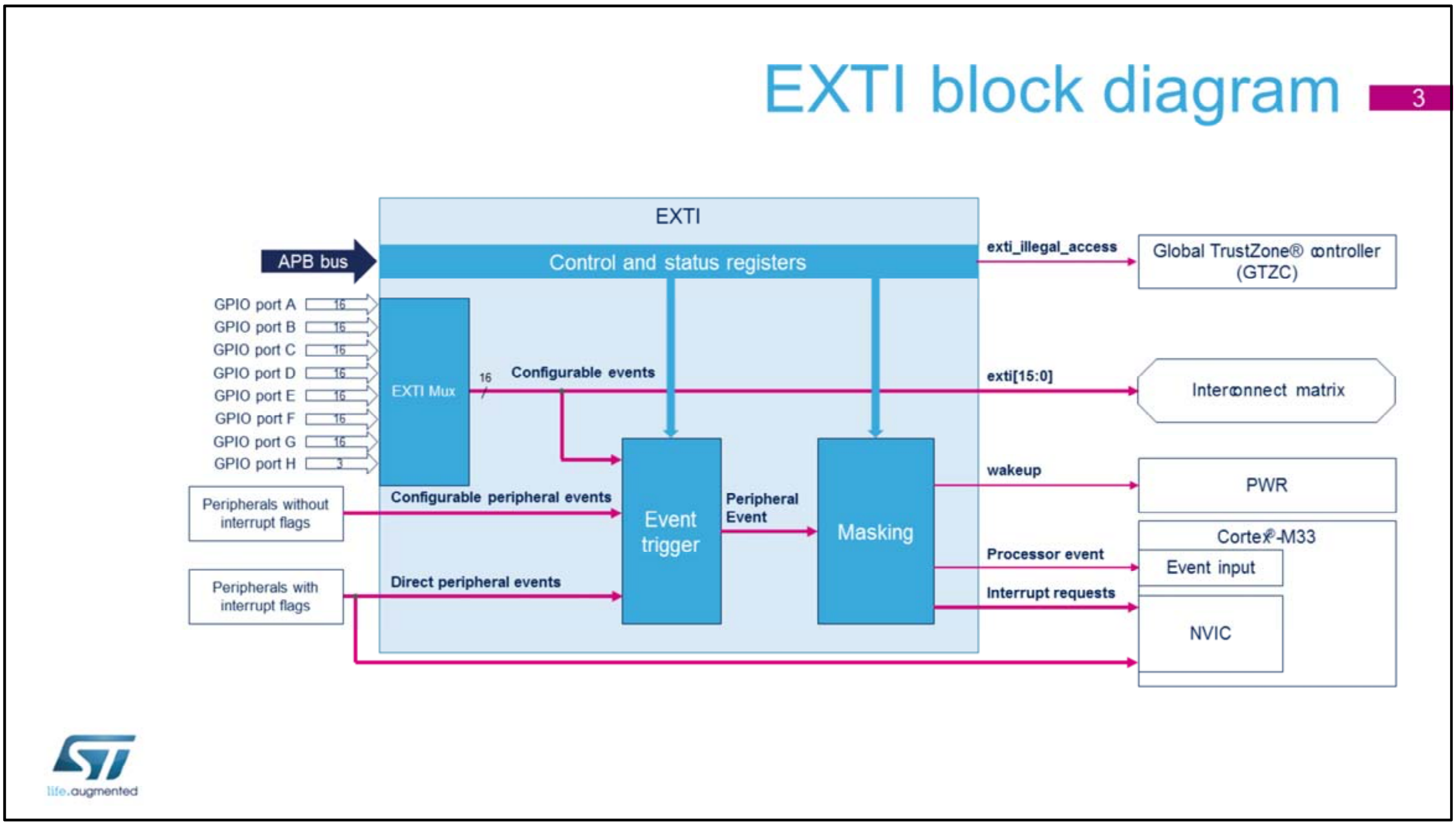
- 42 events / interrupt lines
  - 23 configurable events
  - 19 direct events
- Independent masks and configuration
- Software can emulate events or interrupts by writing to a dedicated register

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 up to 42 independent events, split into two categories – configurable events and direct events.
Applications benefit from smarter use of low-power modes, taking advantage of the STM32L5’s capability to wake up via external communication or requests.



EXTI block diagram

This is the block diagram of the extended interrupt and event controller.

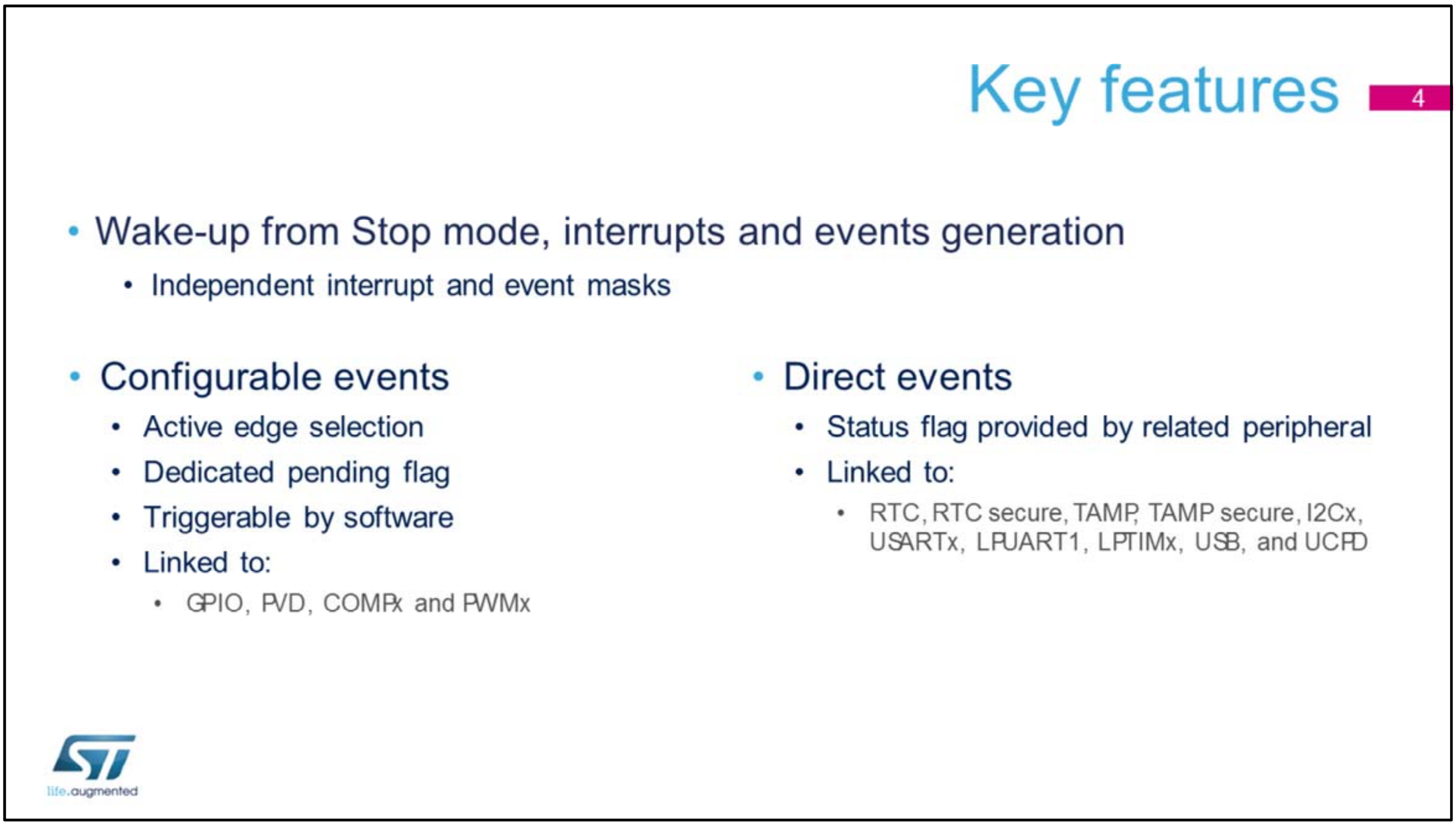
Configurable peripheral events are generated by peripherals without interrupt capability, but which are able to issue a pulse. The EXTI controller provides interrupt detection, masking and software trigger.

Direct peripheral 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.

Do not confuse peripheral events and the processor event. Peripheral events are used by peripherals to indicate that they require processor attention.

The processor event is a pulse signal used by ARM CPUs to exit the Wait For Event low power state.

The EXTI is TrustZone aware. The access to control and configuration bits of secure input events can be made secure and or privileged. When a non-secure master attempts to access a secure resource, the EXTI illegal access is reported to the Global TrustZone controller, GTZC.



Key features

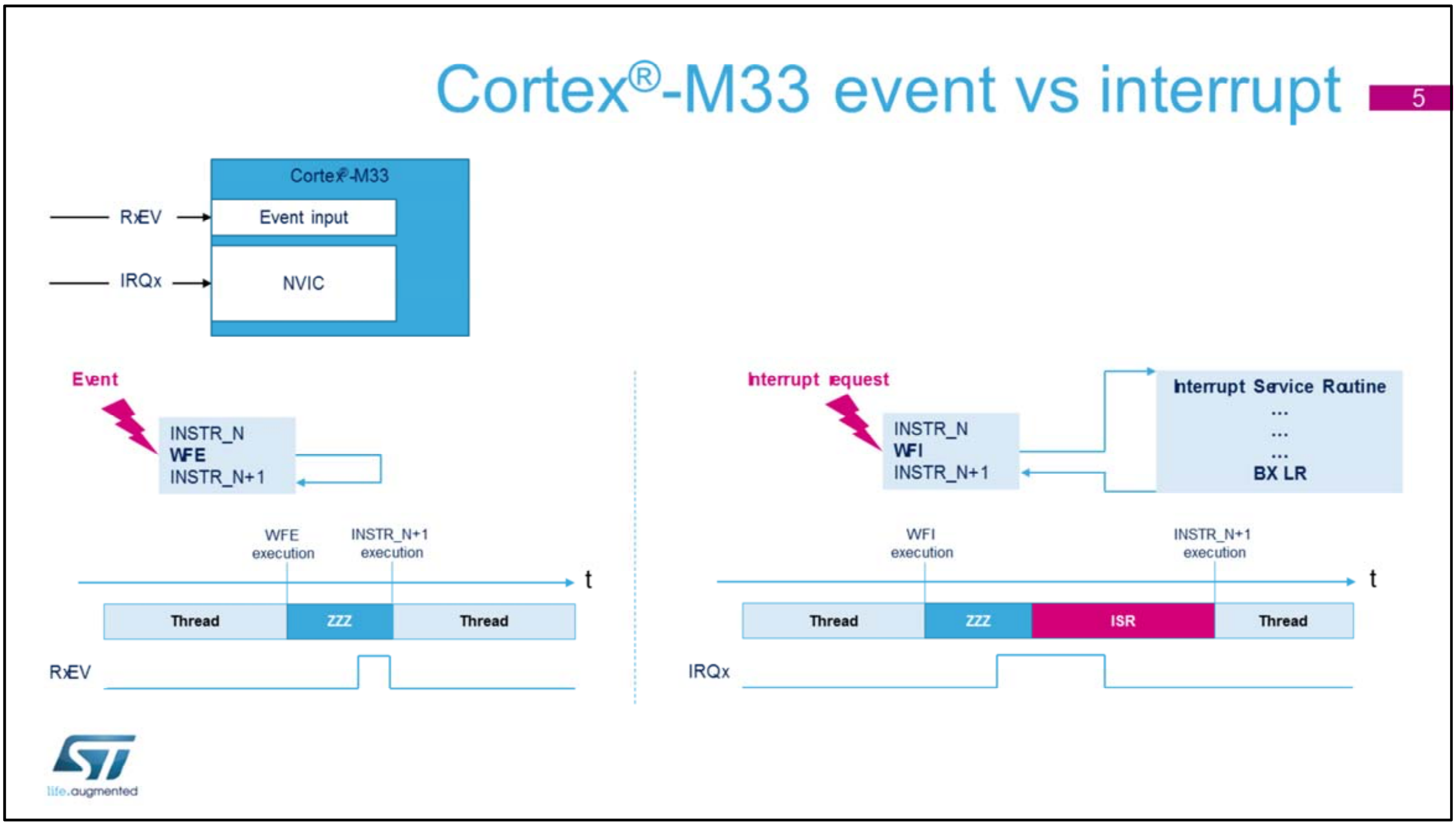
4

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

- Configurable events
  - Active edge selection
  - Dedicated pending flag
  - Triggerable by software
  - Linked to:
    - GPIO, PVD, COMPx and PWMx

- Direct events
  - Status flag provided by related peripheral
  - Linked to:
    - RTC, RTC secure, TAMP, TAMP secure, I2Cx, USARTx, LPUART1, LPTIMx, USB, and UCPD

The Extended Interrupt and event controller can generate interrupts and events as well as wake up the processor from Stop modes.
Configurable events are linked with external interrupts from GPIOs, PVD, comparators and PWM modules.
Direct events are linked with RTC (secure and non secure), TAMP (secure and non-secure), I2C, USART, LPUART1, LPTIM, USB and UCPD.



Cortex®-M33 event vs interrupt

5

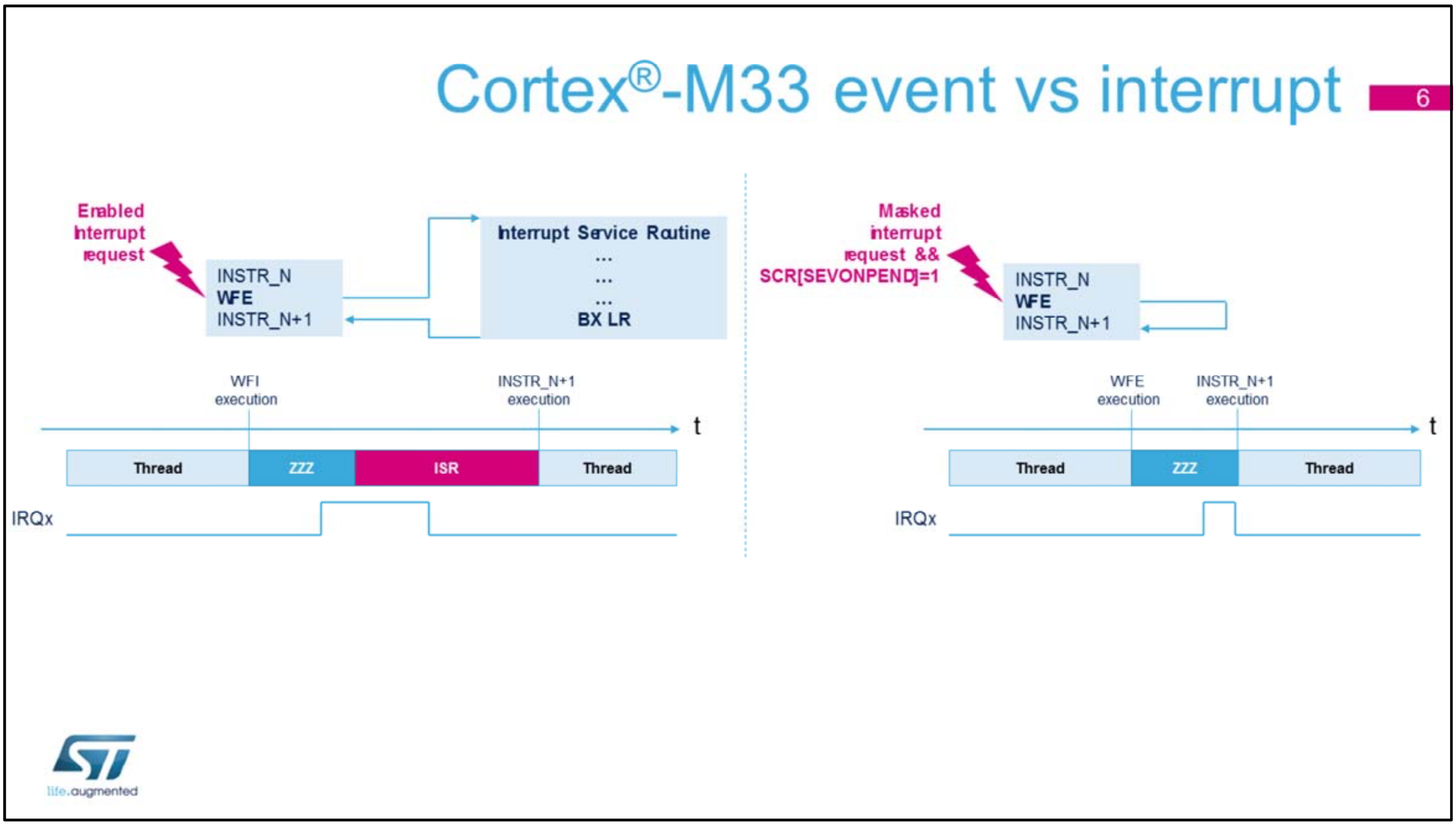
The Cortex®-M33 supports two ways to enter a low-power state:

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.



Cortex®-M33 event vs interrupt

Enabled Interrupt request

INSTR_N
WFE
INSTR_N+1

Interrupt Service Routine
…
…
…
BX LR

WFI execution

INSTR_N+1 execution

t

Thread | zzz | ISR | Thread

IRQx

Masked interrupt request && SCR[SEVONPEND]=1

INSTR_N
WFE
INSTR_N+1

WFE execution

INSTR_N+1 execution

t

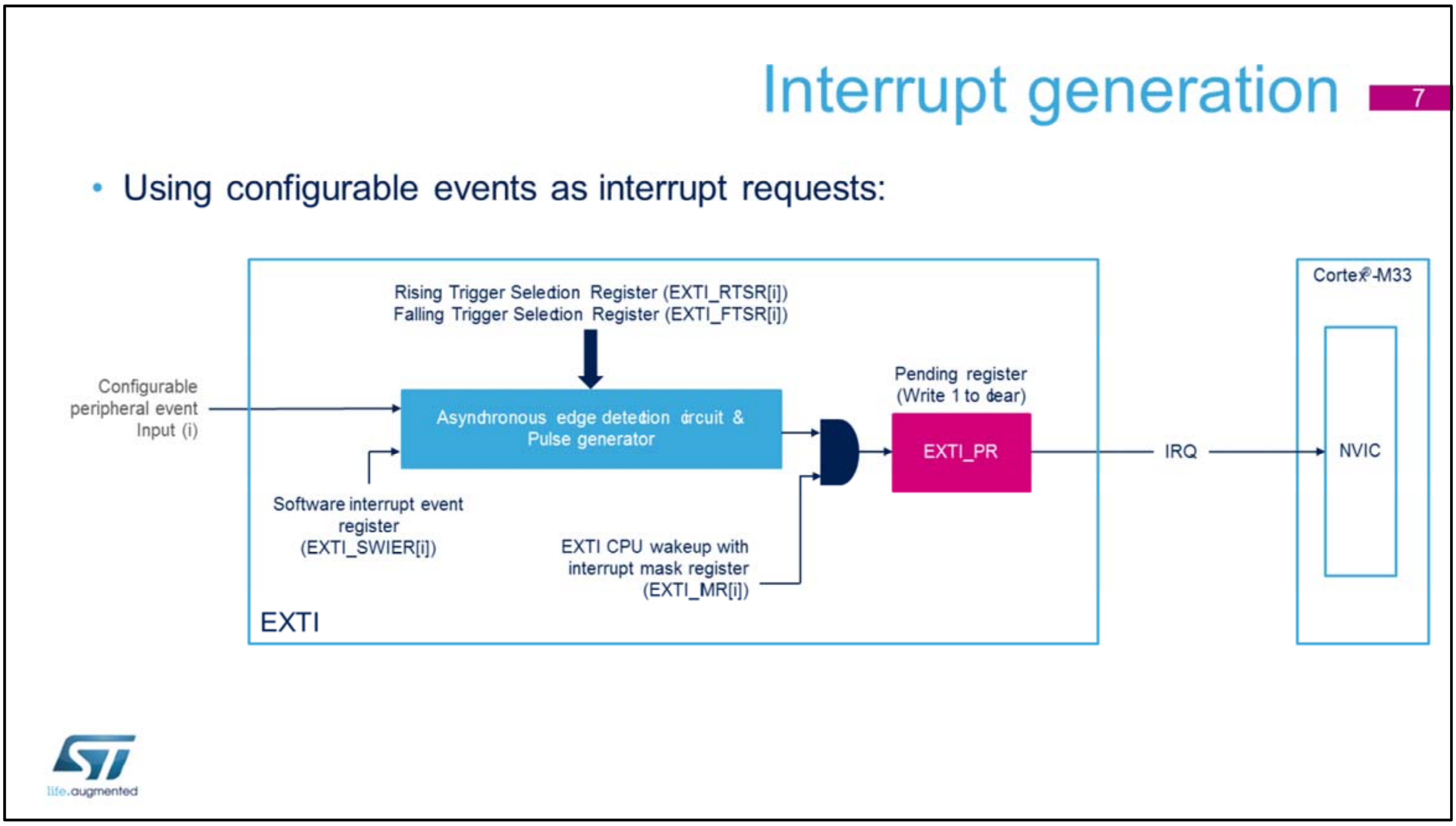
Thread | zzz | Thread

IRQx

life.augmented

Receiving an enabled interrupt request while the processor is in the WFE state causes the processor to wakeup and execute the interrupt service routine.

When the Cortex-M33 control bit called SEVONPEND, which means Send Event on Pending, is set to one, receiving an interrupt request related to a masked interrupt causes a wakeup event. In this case the processor executes the next sequential instruction. Software may later decide to enable the next interrupt to be served.



Interrupt generation

- Using configurable events as interrupt requests:

Rising Trigger Selection Register (EXTI_RTSR[i])
Falling Trigger Selection Register (EXTI_FTSR[i])
Configurable peripheral event Input (i)
Asynchronous edge detection circuit & Pulse generator
Software interrupt event register (EXTI_SWIER[i])
EXTI CPU wakeup with interrupt mask register (EXTI_MR[i])
Pending register (Write 1 to clear)
EXTI_PR
EXTI
IRQ
Cortex®-M33
NVIC
life.augmented

This figure explains the various stages enabling the conversion of a configurable peripheral event active edge into an interrupt request.

The first stage is the asynchronous edge detection circuit configured by two registers EXTI_RTSR and EXTI_FTSR. Any edge, possibly both, can be chosen.

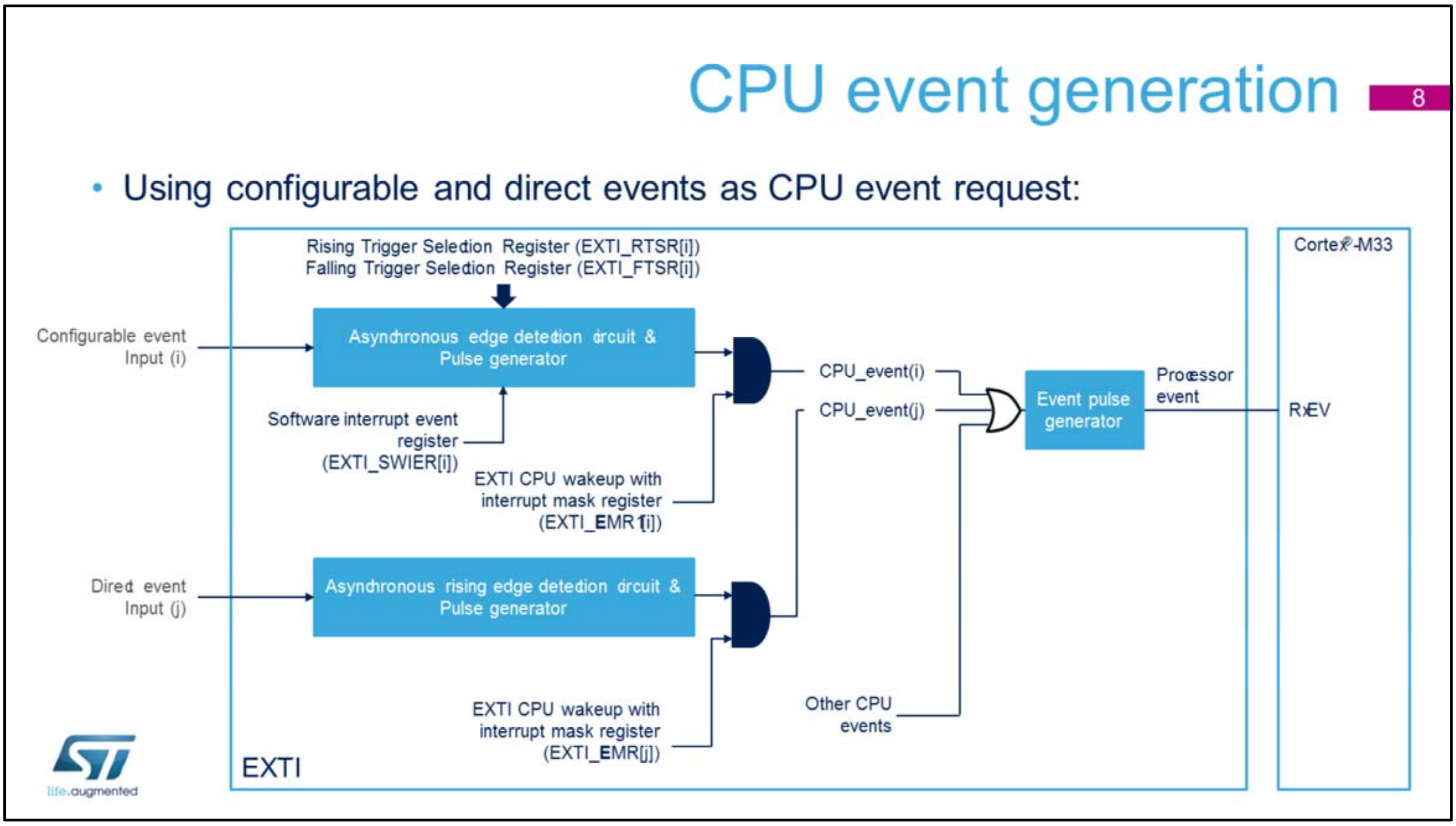
The software can emulate a configurable event by setting the corresponding bit in the EXTI_SWIER register.

The bit is auto-cleared by hardware.

An AND gate is used to mask or enable the generation of the interrupt to the NVIC.

Finally, a flag is set in the EXTI_PR 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

8

- Using configurable and direct events as CPU event request:

Rising Trigger Selection Register (EXTI_RTSR[i])
Falling Trigger Selection Register (EXTI_FTSR[i])
Configurable event Input (i)
Asynchronous edge detection circuit & Pulse generator
Software interrupt event register (EXTI_SWIER[i])
EXTI CPU wakeup with interrupt mask register (EXTI_EMR1[i])
CPU_event(i)
CPU_event(j)
Event pulse generator
Processor event
Cortex®-M33
RxEV
Direct event Input (j)
Asynchronous rising edge detection circuit & Pulse generator
EXTI CPU wakeup with interrupt mask register (EXTI_EMR[j])
Other CPU events
EXTI
life.augmented

This figure explains the various stages enabling the conversion of a peripheral event active edge into a processor event.

Both Configurable and Direct peripheral events can be configured to issue events to the CPU, steered to its RxEV input.

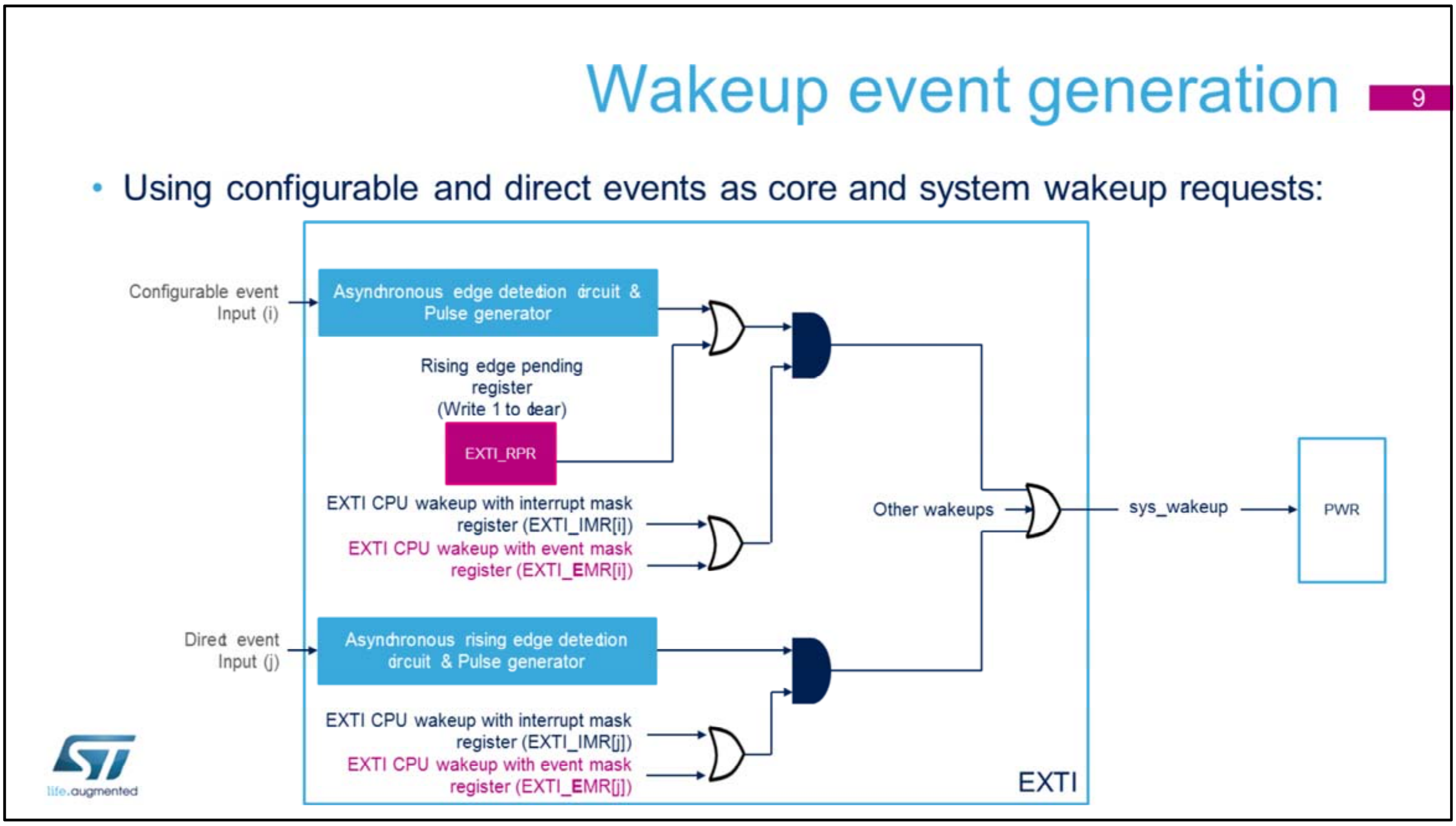
Configurable event active edge is programmable in the EXTI_RTSR and EXTI_FTSR registers while direct events are always sensitive to a rising edge.

Software can emulate a configurable event by writing to the EXTI_SWIER register.

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.

Note that unlike interrupt requests, the pending bit, corresponding to the peripheral event line, is not set.



Wakeup event generation

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

Configurable event Input (i)
Asynchronous edge detection circuit & Pulse generator
Rising edge pending register (Write 1 to clear)
EXTI_RPR
EXTI CPU wakeup with interrupt mask register (EXTI_IMR[i])
EXTI CPU wakeup with event mask register (EXTI_EMR[i])
Direct event Input (j)
Asynchronous rising edge detection circuit & Pulse generator
EXTI CPU wakeup with interrupt mask register (EXTI_IMR[j])
EXTI CPU wakeup with event mask register (EXTI_EMR[j])
Other wakeups
sys_wakeup
PWR
EXTI
life.augmented

The CPU wakeup signals generated by the EXTI block are connected to the PWR block, and are used to wake up the system and CPU sub-system bus clocks.

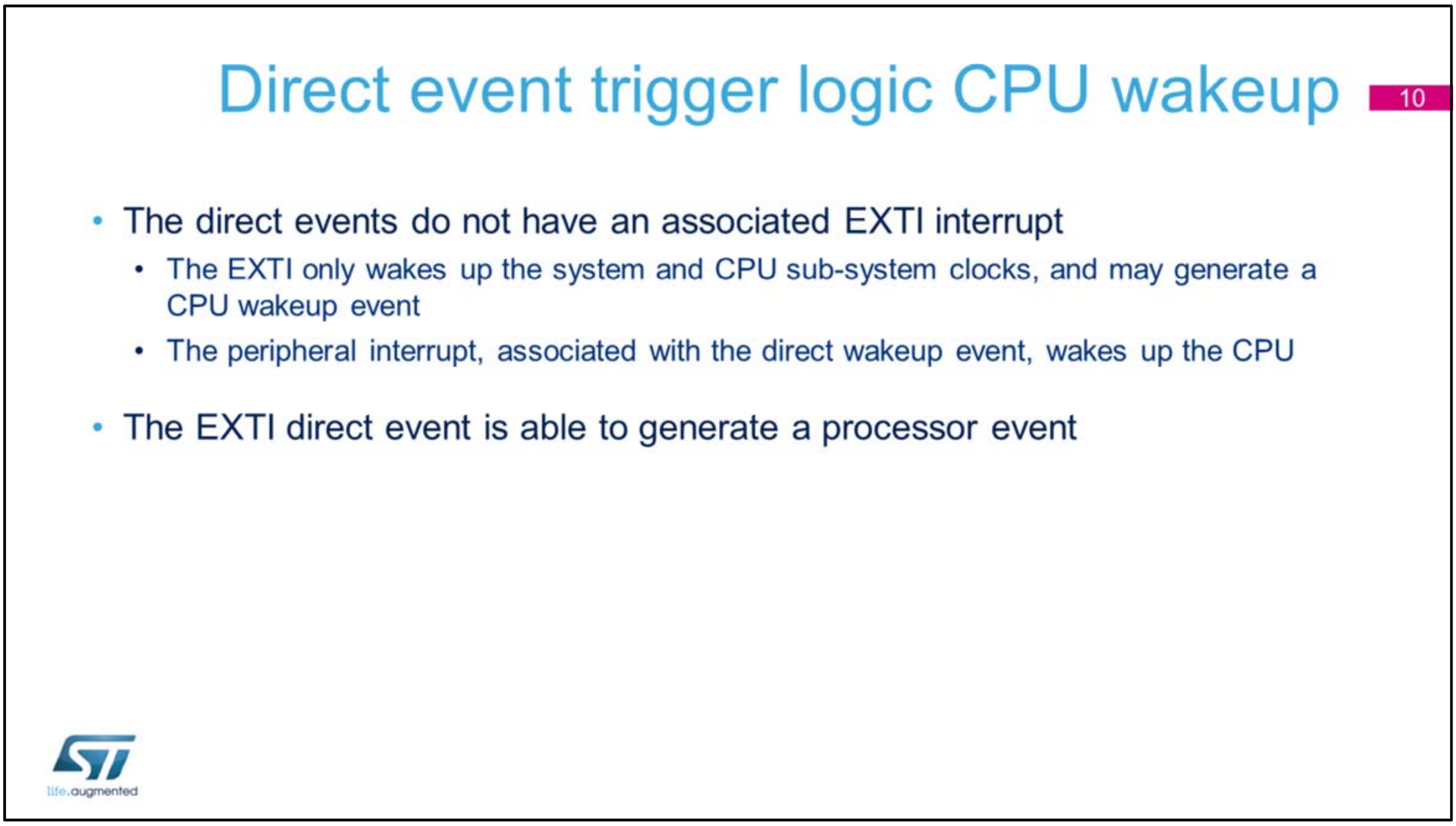
Both configurable and direct peripheral events are able to request a wakeup.

A wakeup occurs when an asynchronous edge detection circuit has detected an active edge or a flag is set to one in the EXTI_RPR register.

Consequently, software is expected to clear the flag in the EXTI_RPR 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.

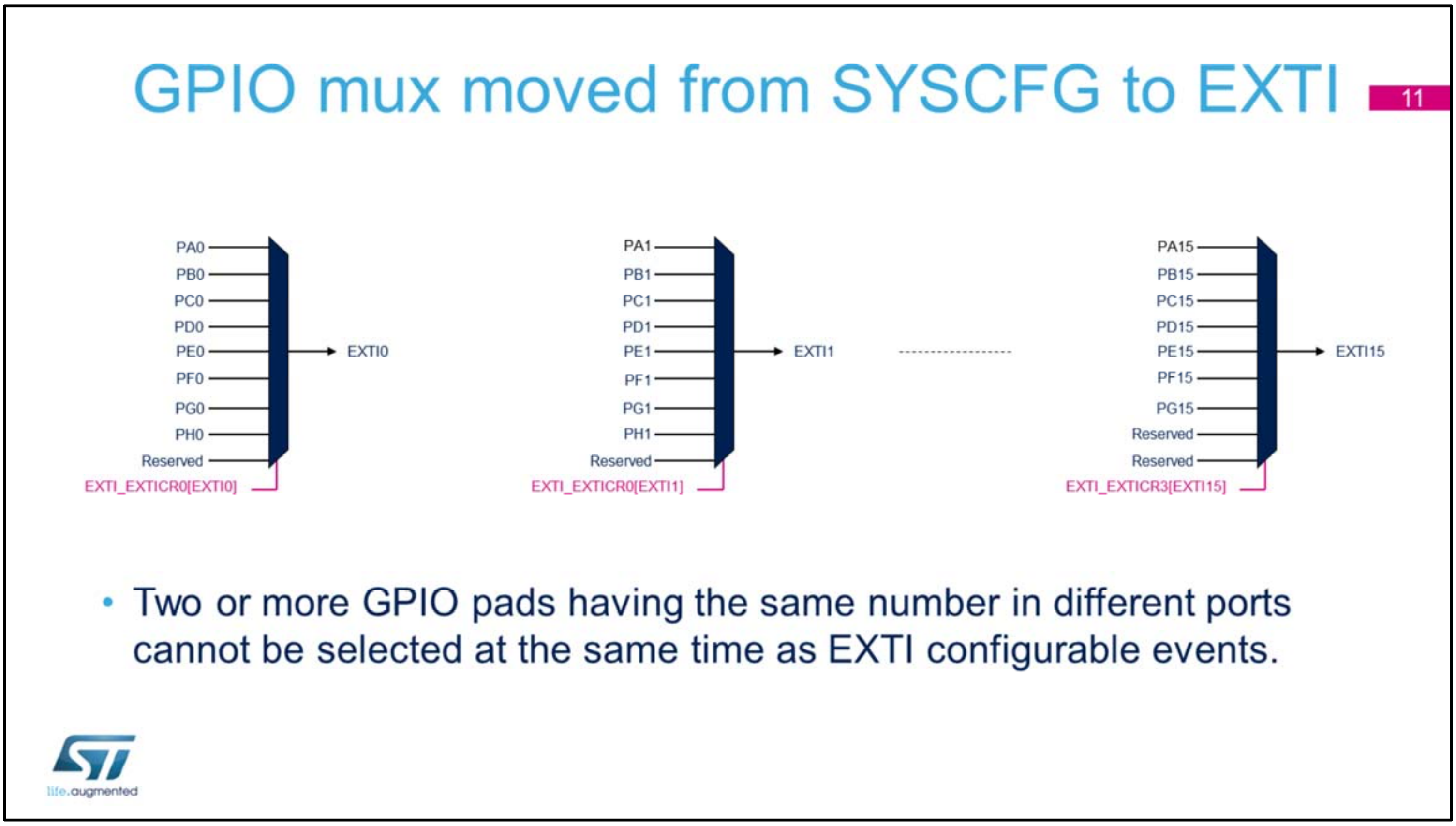
The wakeup indication is asserted when either the interrupt or the event generation is enabled, see the OR gate combining EXTI_IMR and EXTI_EMR registers.



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-system clocks, and may generate a CPU wakeup event
  - The peripheral interrupt, associated with the direct wakeup event, wakes up the CPU
- The EXTI direct event is able to generate a processor event

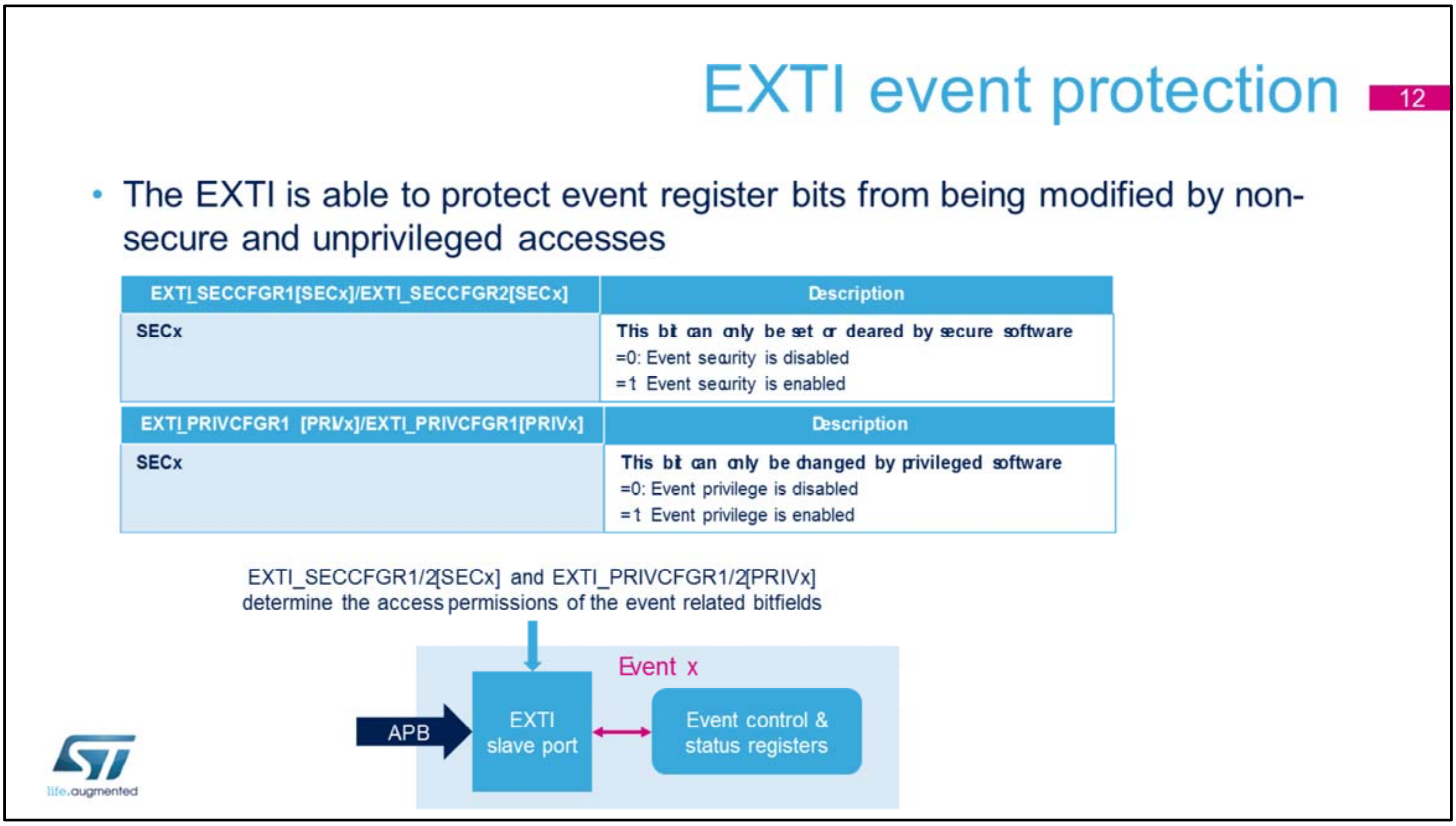
A direct event is able to generate a CPU event and trigger a system wakeup through the EXTI controller.
The active edge of direct events is the rising edge.
Direct events do not rely on the EXTI controller to assert interrupt requests because they have their dedicated lines to the NVIC.
Otherwise, the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



GPIO mux moved from SYSCFG to EXTI

- Two or more GPIO pads having the same number in different ports cannot be selected at the same time as EXTI configurable events.

The STM32L5 has 8 IO ports: Ports A to G are 16-pin wide, port H is 3-pin wide.
Each of the 16 EXTI configurable events related to GPIO ports has an independent multiplexer.
The EXTI multiplexer outputs are available as output signals from the EXTI block to trigger other IPs. The EXTI multiplexer outputs are available independently from any masks defined in the EXTI_IMR and EXTI_EMR registers.



EXTI event protection

- The EXTI is able to protect event register bits from being modified by non-secure and unprivileged accesses

EXTI_SECCFGR1[SECx]/EXTI_SECCFGR2[SECx]	Description
SECx	This bit can only be set or cleared by secure software =0: Event security is disabled =1 Event security is enabled

EXTI_PRIVCFGR1 [PRIVx]/EXTI_PRIVCFGR1[PRIVx]	Description
SECx	This bit can only be changed by privileged software =0: Event privilege is disabled =1 Event privilege is enabled

EXTI_SECCFGR1/2[SECx] and EXTI_PRIVCFGR1/2[PRIVx] determine the access permissions of the event related bitfields

Two levels of protection are programmable per event source: security and privileged.

At EXTI level the protection consists in preventing unauthorized write access to:

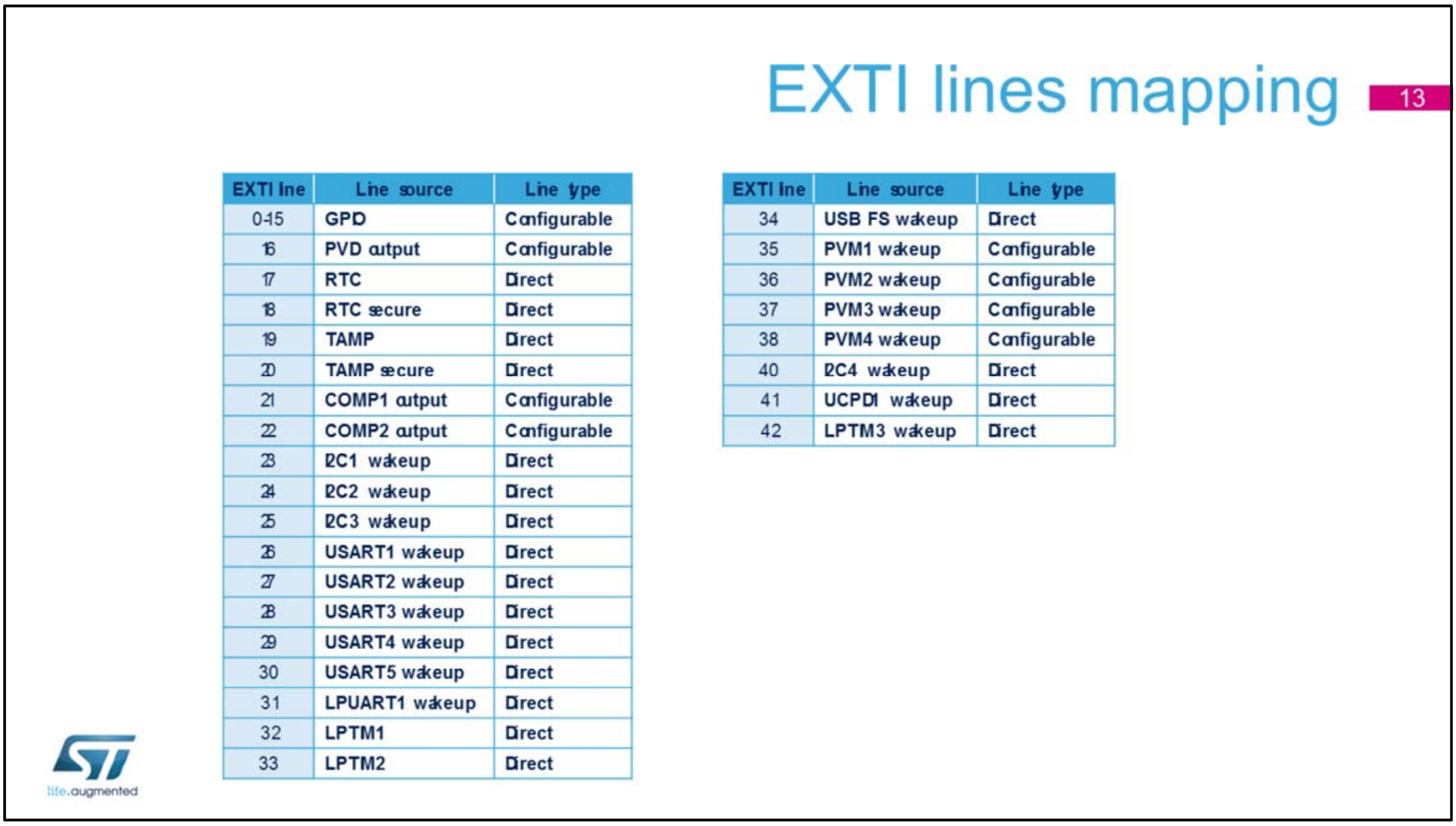
Change the settings of the secure and/or privileged configurable events

Change the masking of the secure and/or privileged input events

Clear pending status of the secure and/or privileged input events.

Security violations are reported to the Global TrustZone Controller, GTZC.

The security and privilege configuration can be globally locked after reset by setting the lock bit in the EXTI_LOCKR register.



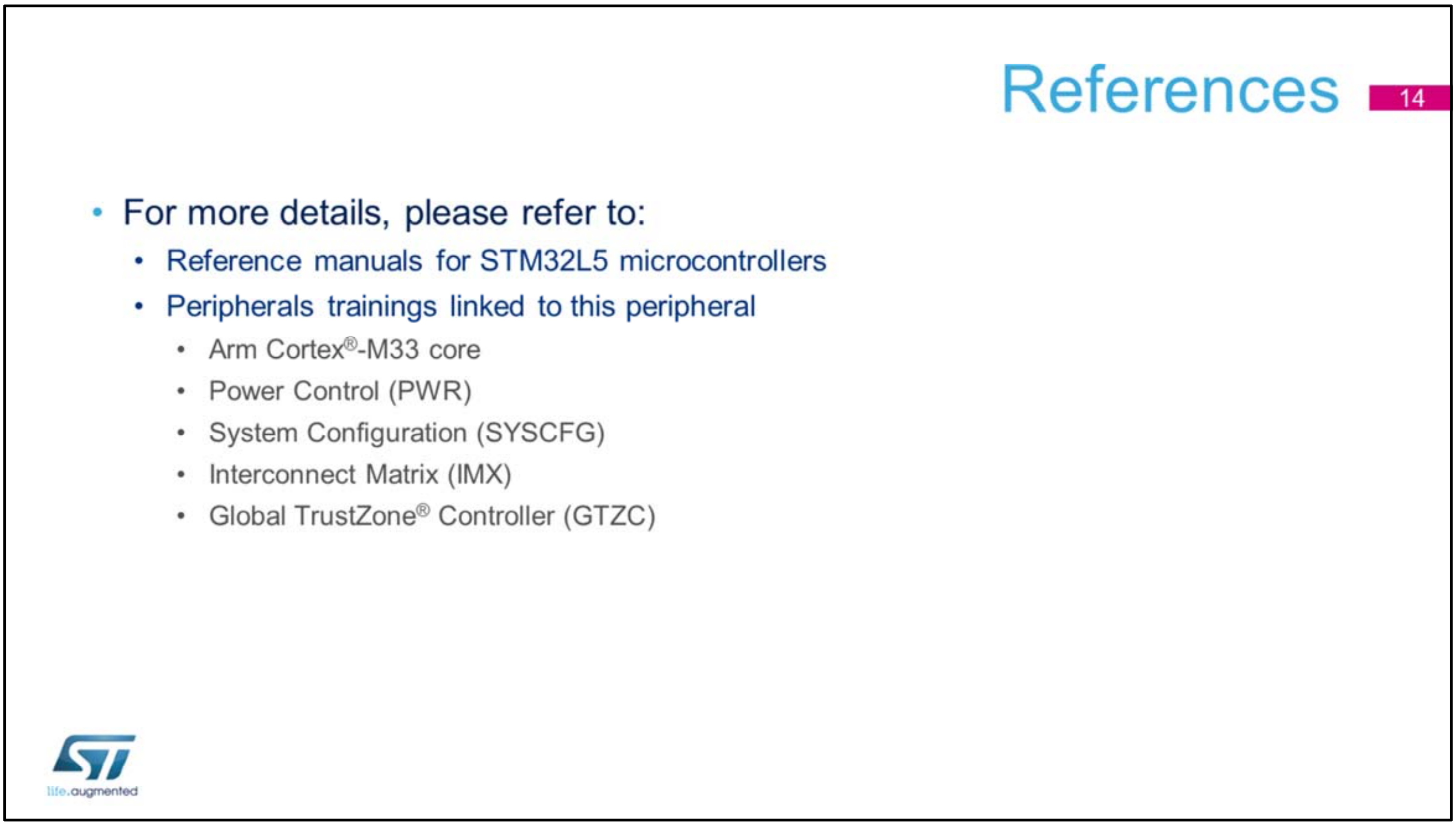
EXTI lines mapping

EXTI line	Line source	Line type
0-15	GPIO	Configurable
16	PVD output	Configurable
17	RTC	Direct
18	RTC secure	Direct
19	TAMP	Direct
20	TAMP secure	Direct
21	COMP1 output	Configurable
22	COMP2 output	Configurable
23	I2C1 wakeup	Direct
24	I2C2 wakeup	Direct
25	I2C3 wakeup	Direct
26	USART1 wakeup	Direct
27	USART2 wakeup	Direct
28	USART3 wakeup	Direct
29	USART4 wakeup	Direct
30	USART5 wakeup	Direct
31	LPUART1 wakeup	Direct
32	LPTM1	Direct
33	LPTM2	Direct

EXTI line	Line source	Line type
34	USB FS wakeup	Direct
35	PVM1 wakeup	Configurable
36	PVM2 wakeup	Configurable
37	PVM3 wakeup	Configurable
38	PVM4 wakeup	Configurable
40	I2C4 wakeup	Direct
41	UCPD1 wakeup	Direct
42	LPTM3 wakeup	Direct

life.augmented

This table provides all inputs of the EXTI block present in the STM32L5 microcontroller and indicates for each of them whether it is a configurable event input or a direct event input.



References

14

- For more details, please refer to:
  - Reference manuals for STM32L5 microcontrollers
  - Peripherals trainings linked to this peripheral
    - Arm Cortex®-M33 core
    - Power Control (PWR)
    - System Configuration (SYSCFG)
    - Interconnect Matrix (IMX)
    - Global TrustZone® Controller (GTZC)

For more details about the System Configuration module, refer to the reference manual for STM32L5 microcontrollers. If needed, refer also to these trainings for more information:

- Arm Cortex-M33 core
- Power control (PWR)
- System Configuration (SYSCFG)
- Interconnect Matrix (IMX)
- Global TrustZone® Controller (GTZC).