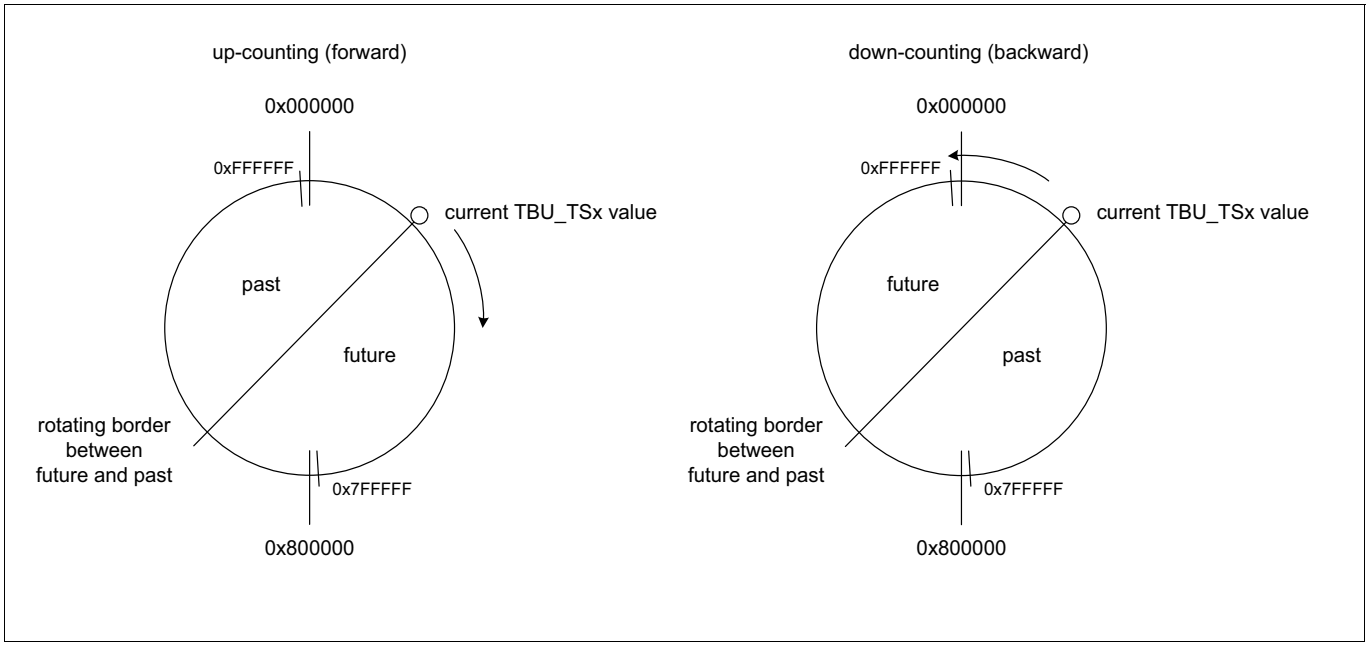

Figure 15 Cyclic event counter representing time or angle

Inside different submodules of GTM a grater-equal compare (in case of up-counting) or a less-equal compare (in case of down-counting) against a TBU base value (representing time or angle) always means that it is checked if the reference value is in relation to the current TBU value in the future or in the past.

28.4.5 GTM Interrupt Concept

The sub-modules of the GTM can generate thousands of interrupts on behalf of internal events. This high amount of interrupts is combined inside the Interrupt Concentrator Module (ICM) into interrupt groups. In these interrupt groups the GTM sub-module interrupt signals are bundled to a smaller set of interrupts. From these interrupt sets, a smaller amount of interrupt signals is created and signaled outside of the GTM as a signal *GTM_<MOD>_IRQ,* where <MOD> identifies the name of the corresponding GTM sub-module.

Moreover, each output signal *GTM_<MOD>_IRQ* has a corresponding input signal *GTM_<MOD>_IRQ_CLR* that can be used for clearing the interrupts. These input signals can be used by the surrounding micro controller system as:

- acknowledge signal from a DMA controller
- validation signal from ADC
- clear signal from a GTM-external interrupt controller to do an atomic clear while entering an ISR routine

The controlling of the individual interrupts is done inside the sub-modules. If a sub-module consists of several sub-module channels that are most likely to work independent from each other (like TIM, PSM, MCS, TOM, and ATOM), each sub-module channel has its own interrupt control and status register set, named as interrupt set in the following. Other sub-modules (SPE, ARU, DPLL, BRC, CMP and global GTM functionality) have a common interrupt set for the whole sub-module.

The interrupt set consists of four registers: The **IRQ_EN** register, the **IRQ_NOTIFY** register, the **IRQ_FORCINT** register, and the **IRQ_MODE** register. While the registers **IRQ_EN**, **IRQ_NOTIFY,** and **IRQ_FORCINT** signalize the status and allow controlling of each individual interrupt source within an interrupt set, the register **IRQ_MODE** configures the interrupt mode that is applied to all interrupts that belong to the same interrupt set.

In order to support a wide variety of micro controller architectures and interrupt systems with different interrupt signal output characteristics and internal interrupt handling the following four modes can be configured:

- Level mode,

- Pulse mode,
- Pulse-Notify mode,
- Single-Pulse mode.

These interrupt modes are described in more details in the following subsections.

The register **IRQ_EN** allows the enabling and disabling of an individual interrupt within an interrupt set. Independent of the configured mode, only enabled interrupts can signalize an interrupts on its signal *GTM_<MOD>_IRQ*.

The register **IRQ_NOTIFY** collects the occurrence of interrupt events. The behavior for setting a bit in this register depends on the configured mode and thus it is described later on in the mode descriptions.

Independent of the configured mode any write access with value '1' to a bit in the register **IRQ_NOTIFY** always clears the corresponding **IRQ_NOTIFY** bit.

Moreover, the enabling of a disabled interrupt source with a write access to the register **IRQ_EN** also clears the corresponding bit in the **IRQ_NOTIFY** register but only if the error interrupt source **EIRQ_EN** is disabled. However, if the enabling of a disabled interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the register **IRQ_NOTIFY** is not cleared.

Additionally, each write access to the register **IRQ_MODE**, clears all bits in the **IRQ_NOTIFY** register. It should be notified that the clearing of **IRQ_NOTIFY** is applied independently of the written data (e.g. no mode change).

Thus, a secure way for reconfiguring the interrupt mode of an interrupt set, is to disable all interrupts of the interrupt set with the register **IRQ_EN**, define the new interrupt mode by writing register **IRQ_MODE**, followed by enabling the desired interrupts with the register **IRQ_EN**.

Thus, a secure way for reconfiguring the interrupt mode of an error interrupt set, is to disable all error interrupts of the error interrupt set with the register **EIRQ_EN**, define the new interrupt mode by writing register **IRQ_MODE**, followed by enabling the desired error interrupts with the register **EIRQ_EN**.

The register **IRQ_FORCINT** is used by software for triggering individual interrupts with a write access with value '1'. Since a write access to **IRQ_FORCINT** only generates a single pulse, **IRQ_FORCINT** is not implemented as a true register and thus any read access to **IRQ_FORCINT** always results with a value of '0'.

The mechanism for triggering interrupts with **IRQ_FORCINT** is globally disabled after reset. It has to be explicitly enabled by clearing the bit **RF_PROT** in the register **GTM_CTRL** (see **Chapter 28.4.9.3**)

For the modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE and CMP each interrupt may configured to raise instead of the normal interrupt an error interrupt if enabled by the corresponding error interrupt enable bit in register **EIRQ_EN**.It is possible for one source to enable the normal interrupt and the error interrupt in parallel. Because both interrupt clear signals could reset the notify bit this is expected to cause problems in a system and therefore it is strongly recommended to not enable both interrupt types at the same point in time.

Similar to enabling an interrupt, the enabling of a disabled error interrupt source with a write access to the register **EIRQ_EN** also clears the corresponding bit in the **IRQ_NOTIFY** register only if the interrupt source **IRQ_EN** is disabled. However, if the enabling of a disabled error interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the register **IRQ_NOTIFY** is not cleared.

All enabled error interrupts are OR-combined inside the ICM and assigned to the dedicated GTM port *gtm_err_irq*. A corresponding input *gtm_err_irq_clr* allows the reset of this error interrupt from outside the GTM (hardware clear).

To be able to detect the module source of the error interrupt the ICM provides the register **ICM_IRQG_MEI**.

The error interrupt causing channel can be determined for the module FIFO by evaluating the ICM register **ICM_IRQG_CEI0**.

The error interrupt causing channel can be determined for the modules TIM by evaluating the ICM register **ICM_IRQG_CEI1...2**.

The error interrupt causing channel can be determined for the modules MCS with all possible channel by evaluating the ICM register **ICM_IRQG_MCS[i]_CEI**. In case of usage only the first 8 channels of each MCS the error interrupt causing channel can be determined by evaluating the ICM register **ICM_IRQG_CEI3...4**.

28.4.5.1 Level interrupt mode

The default interrupt mode is the Level Interrupt Mode. In this mode each occurred interrupt event is collected in the register **IRQ_NOTIFY**, independent of the corresponding enable bit of register **IRQ_EN** and **EIRQ_EN**.

An interrupt event, which is defined as a pulse on the signal *Int_out* of **Figure 16**, may be triggered by the interrupt source of the sub-module or by software performing a write access to the corresponding register **IRQ_FORCINT**, with a disabled bit **RF_PROT** in register **GTM_CTRL**.

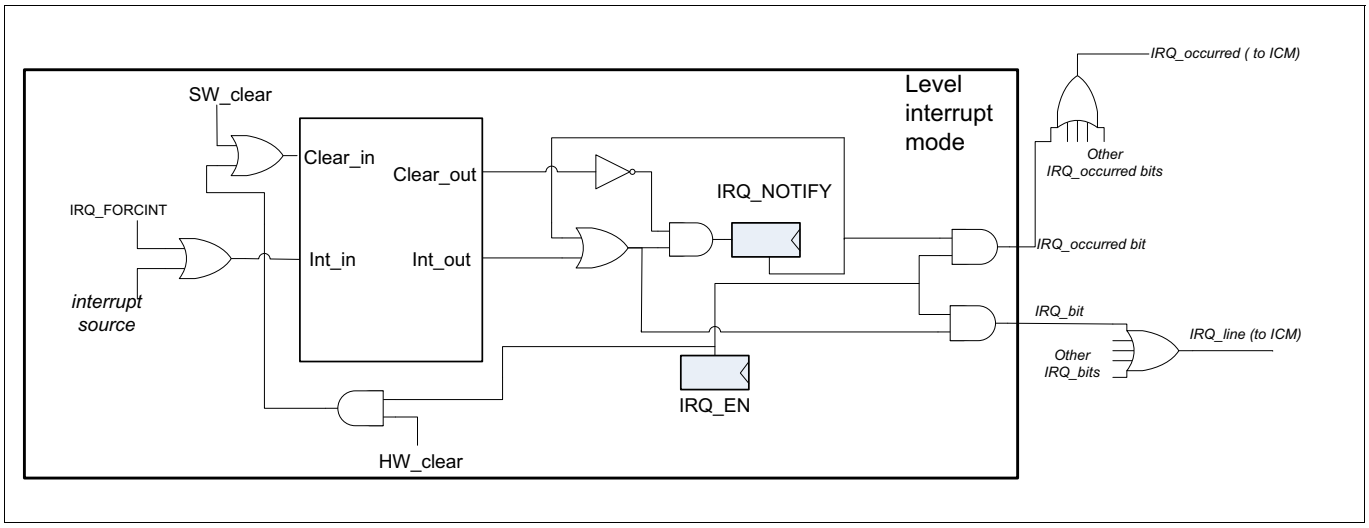

Figure 16 Level interrupt mode scheme

A collected interrupt bit in register **IRQ_NOTIFY** may be cleared by a clear event, which is defined as a pulse on signal *Clear_out* of **Figure 16**. A clear event can be performed by writing '1' to the corresponding bit in the register **IRQ_NOTIFY** leading to a pulse on signals *SW_clear*. A clear event may also result from an externally connected signal *GTM_<MOD>_IRQ_CLR*, which is routed to the signal *HW_clear* of **Figure 16**. However, the hardware clear mechanism is only possible, if the corresponding interrupt is enabled by register **IRQ_EN**.

As **Table 9** shows, interrupt events are dominant in the case of a simultaneous interrupt event and clear event.

Table 9 Priority of Interrupt Events and Clear Events

Int_in	Clear_in	Int_out	Clear_out
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

As shown in **Figure 16** an occurred interrupt event is signaled as a constant signal level with value 1 to the signal *IRQ_bit*, if the corresponding interrupt is enabled in register **IRQ_EN**.

With exception of the sub-modules ARU and DPLL, the signal *IRQ_bit* is OR-combined with the neighboring *IRQ_bit* signals of the same interrupt set and they are routed as a signal *IRQ_line* to the interrupt concentrator module (ICM). The interrupt signals *IRQ_bit* of the sub-modules DPLL and ARU are routed directly as a signal *IRQ_line* to the sub-module ICM. In some cases (sub-modules TOM and ATOM) the ICM may further OR-combine

several *IRQ_line* signals to an outgoing interrupt signal *GTM_<MOD>_IRQ*. In the other cases the *IRQ_line* signals are directly connected to the outgoing signals *GTM_<MOD>_IRQ,* within the sub-module ICM.

The signal *IRQ_occurred* is connected in a similar way as the signal *IRQ_line*, however this signal is used for monitoring the interrupt state of the register **IRQ_NOTIFY** in the registers of the ICM.

The additional error interrupt enable mechanism for level interrupt is shown below.

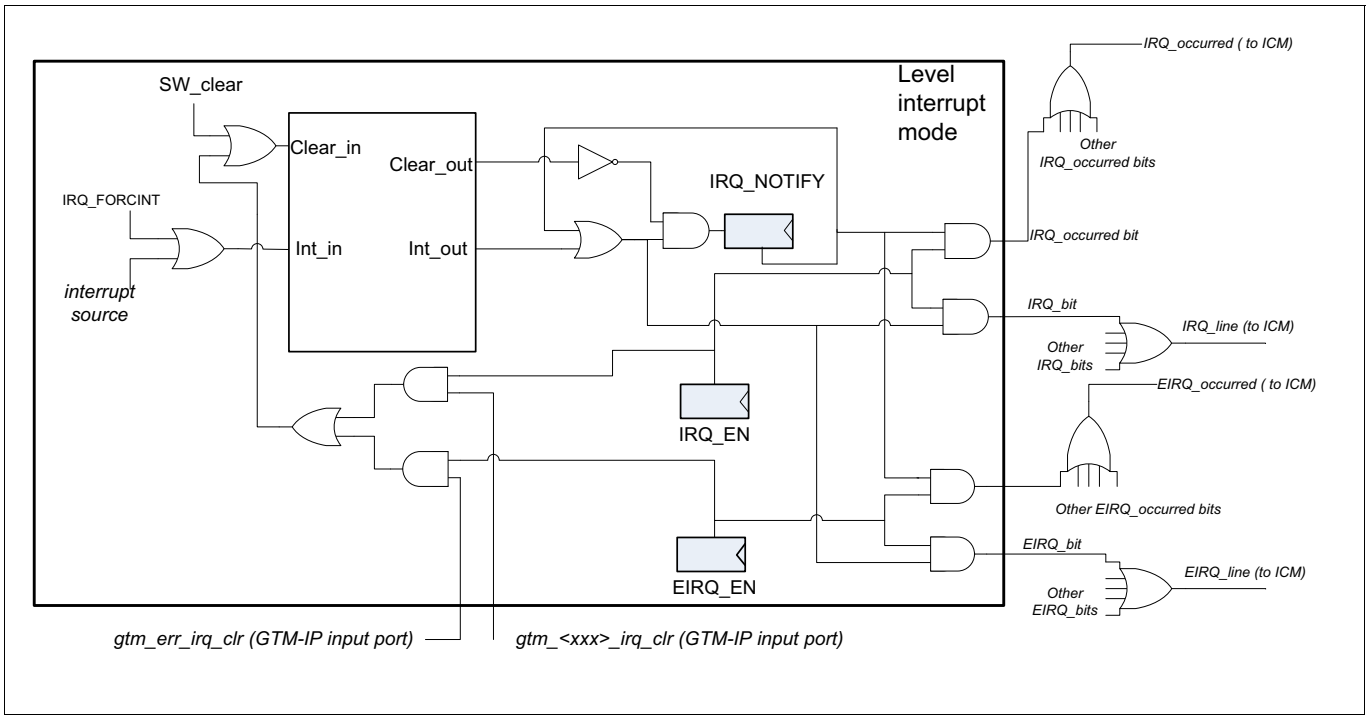

Figure 17 Level interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

A collected interrupt bit in register **IRQ_NOTIFY** may be cleared by a clear event, which is defined as a pulse on signal *Clear_out* of **Figure 17**. A clear event can be performed by writing '1' to the corresponding bit in the register **IRQ_NOTIFY** leading to a pulse on signals *SW_clear*. A clear event may also result from the externally connected signal *gtm_<MOD>_irq_clr* or *gtm_err_irq_clr*, which is routed as an *HW_clear* to *Clear_in* of **Figure 17**. However, the hardware clear mechanism is only possible, if the corresponding interrupt or error interrupt is enabled by register **IRQ_EN** or **EIRQ_EN**.

As it can be seen from the **Figure 17** an occurred interrupt event is signaled as a constant signal level with value 1 to the signal *IRQ_bit*, if the corresponding interrupt is enabled in register **IRQ_EN**.

28.4.5.2 Pulse interrupt mode

The Pulse interrupt mode behavior can be observed from **Figure 18**.

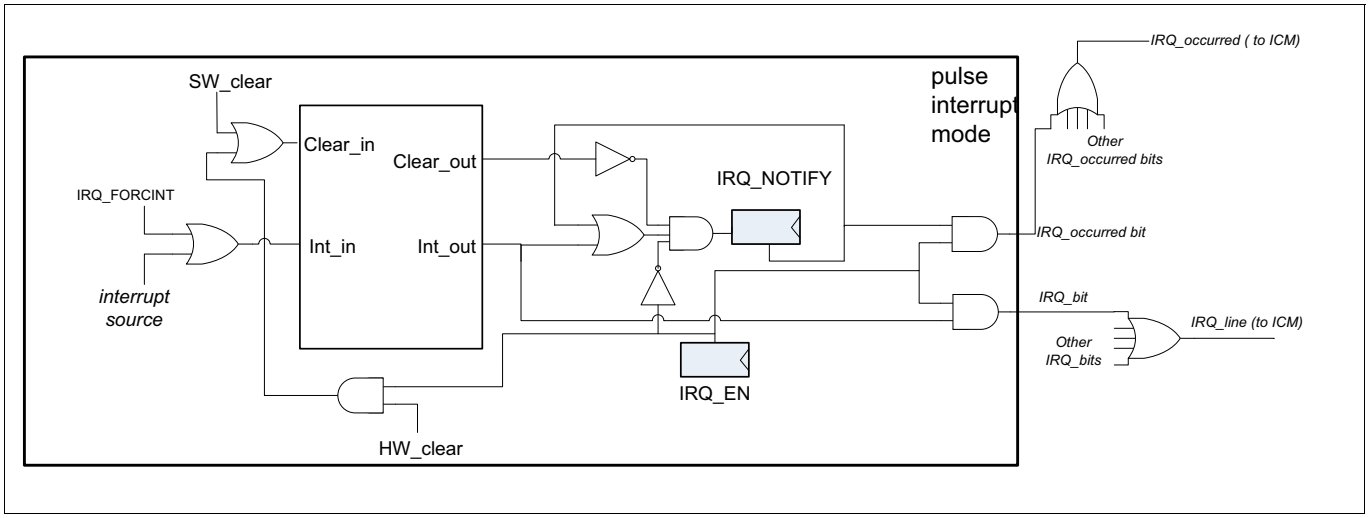

Figure 18 Pulse interrupt mode scheme

In Pulse Interrupt Mode each Interrupt Event will generate a pulse on the *IRQ_bit* signal if **IRQ_EN** is enabled.

As it can be seen from the figure, the interrupt bit in **IRQ_NOTIFY** register is always cleared if **IRQ_EN** is enabled.

However, if an interrupt is disabled in the register **IRQ_EN**, an occurred interrupt event is captured in the register **IRQ_NOTIFY**, in order to allow polling for disabled interrupts by software.

Disabled interrupts may be cleared by an interrupt clear event.

In Pulse interrupt mode, the signal IRQ_occurred is always 0.

The additional error interrupt enable mechanism for pulse interrupt is shown below.

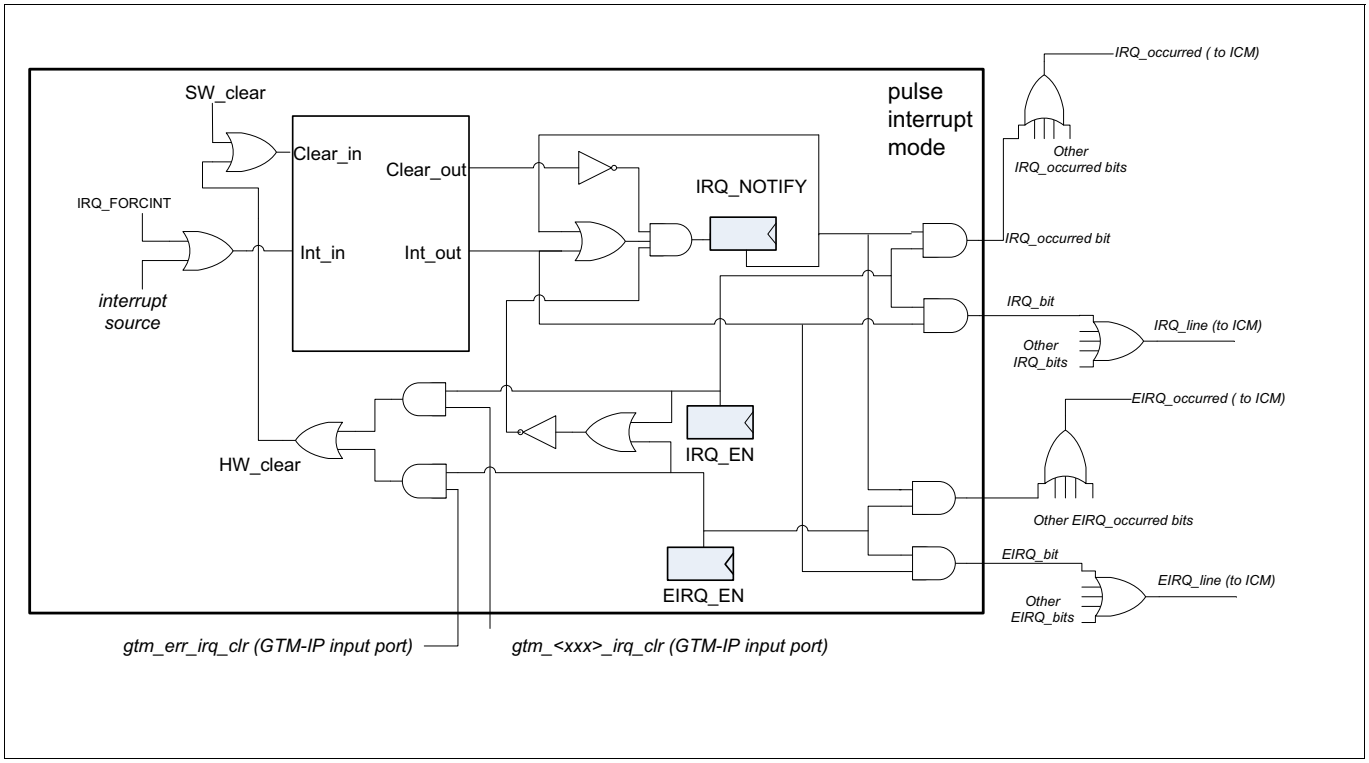

Figure 19 Pulse interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse Interrupt Mode each Interrupt Event will generate a pulse on the *EIRQ_bit* signal if **EIRQ_EN** is enabled.

As it can be seen from the figure, the interrupt bit in **IRQ_NOTIFY** register is always cleared if **EIRQ_EN** or **IRQ_EN** are enabled.

However, if an error interrupt is disabled in the register **EIRQ_EN**, an occurred error interrupt event is captured in the register **IRQ_NOTIFY**, in order to allow polling for disabled error interrupts by software.

Disabled error interrupts may be cleared by an error interrupt clear event.

In Pulse interrupt mode, the signal EIRQ_occurred is always 0.

28.4.5.3 Pulse-notify interrupt mode

In Pulse-notify Interrupt mode, all interrupt events are captured in the register **IRQ_NOTIFY**. If an interrupt is enabled by the register **IRQ_EN**, each interrupt event will also generate a pulse on the *IRQ_bit* signal. The signal *IRQ_occurred* will be high if interrupt is enabled in register **IRQ_EN** and the corresponding bit of register **IRQ_NOTIFY** is set. The Pulse-notify interrupt mode is shown in **Figure 20**.

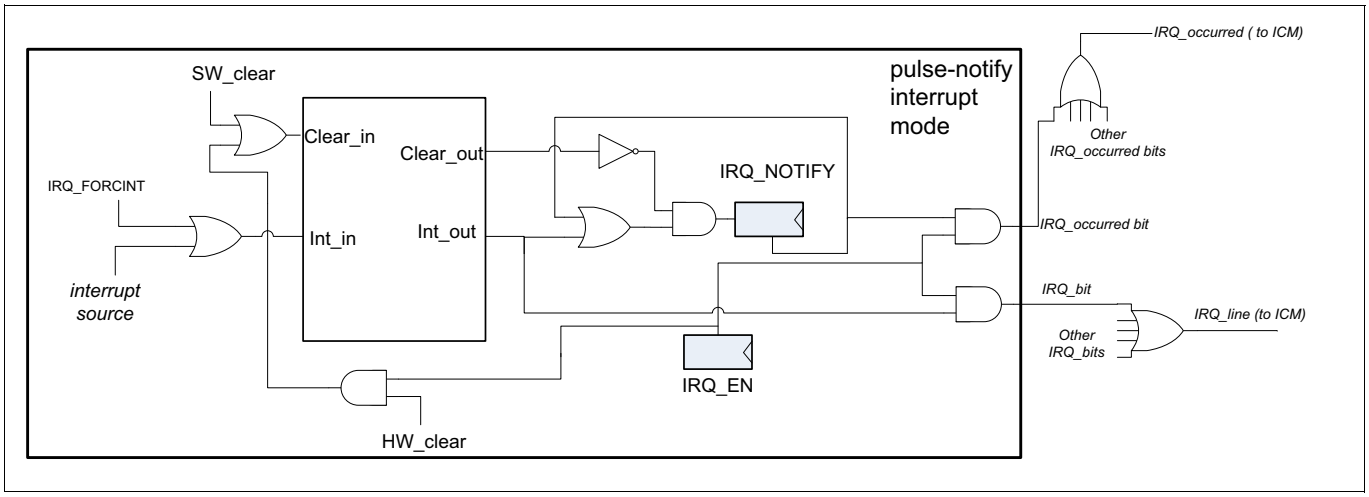

Figure 20 Pulse-notify interrupt mode scheme

The additional error interrupt enable mechanism for pulse-notify interrupt is shown below

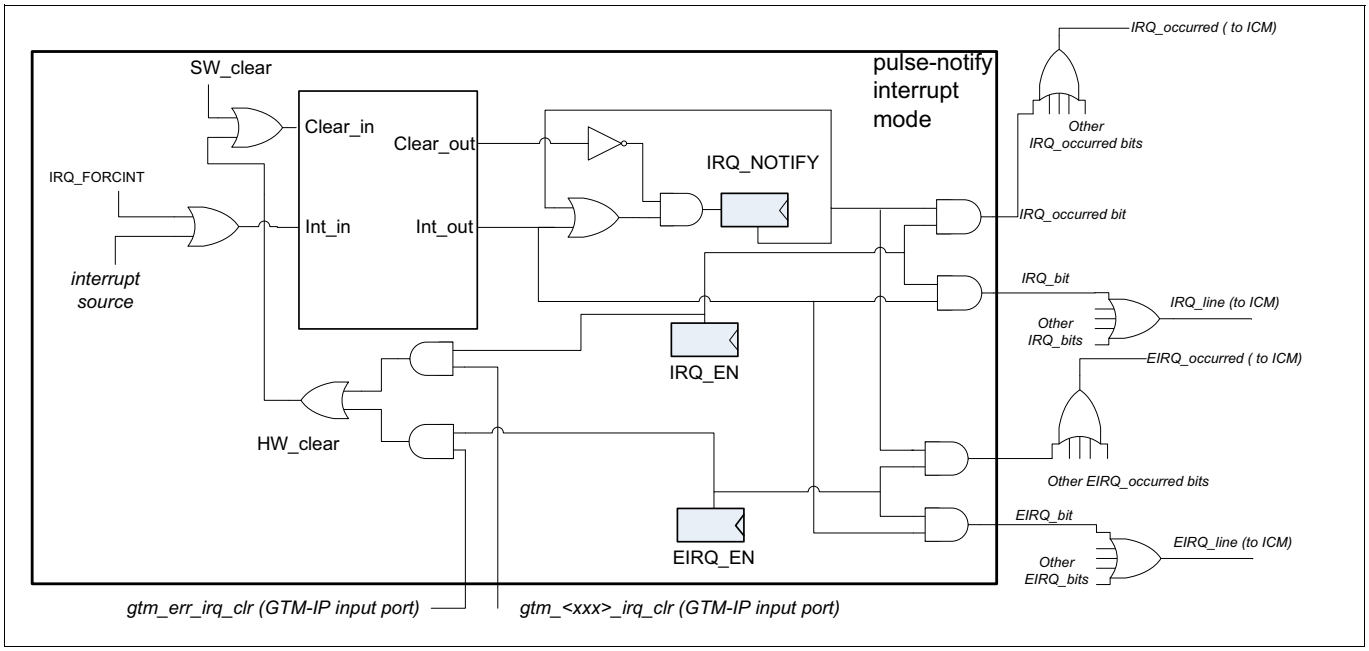

Figure 21 Pulse-notify interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse-notify Interrupt mode, all error interrupt events are captured in the register **IRQ_NOTIFY**. If an error interrupt is enabled by the register **EIRQ_EN**, each error interrupt event will also generate a pulse on the *EIRQ_bit* signal. The signal *EIRQ_occurred* will be high if error interrupt is enabled in register **EIRQ_EN** and the corresponding bit of register **IRQ_NOTIFY** is set. The Pulse-notify interrupt mode for error interrupts is shown in **Figure 21**.

28.4.5.4 Single-pulse interrupt mode

In Single-pulse Interrupt Mode, an interrupt event is always captured in the register **IRQ_NOTIFY**, independent of the state of **IRQ_EN**. However, only the first interrupt event of an enabled interrupt within a common interrupt set is forwarded to signal *IRQ_line.* Additional interrupt events of the same interrupt set cannot generate pulses on the signal *IRQ_line*, until the corresponding bits in register **IRQ_NOTIFY** of enabled interrupts are cleared by a clear event. The *IRQ_occurred* signal line will be high, if the **IRQ_EN** and the **IRQ_NOTIFY** register bits are set. The Single-pulse interrupt mode is shown in **Figure 22**.

The only exceptions are the modules ARU and DPLL. In these modules the *IRQ_occurred* bit of each interrupt is directly connected (without OR-conjunction of neighboring *IRQ_occurred* bits) to the inverter for suppressing further interrupt pulses.

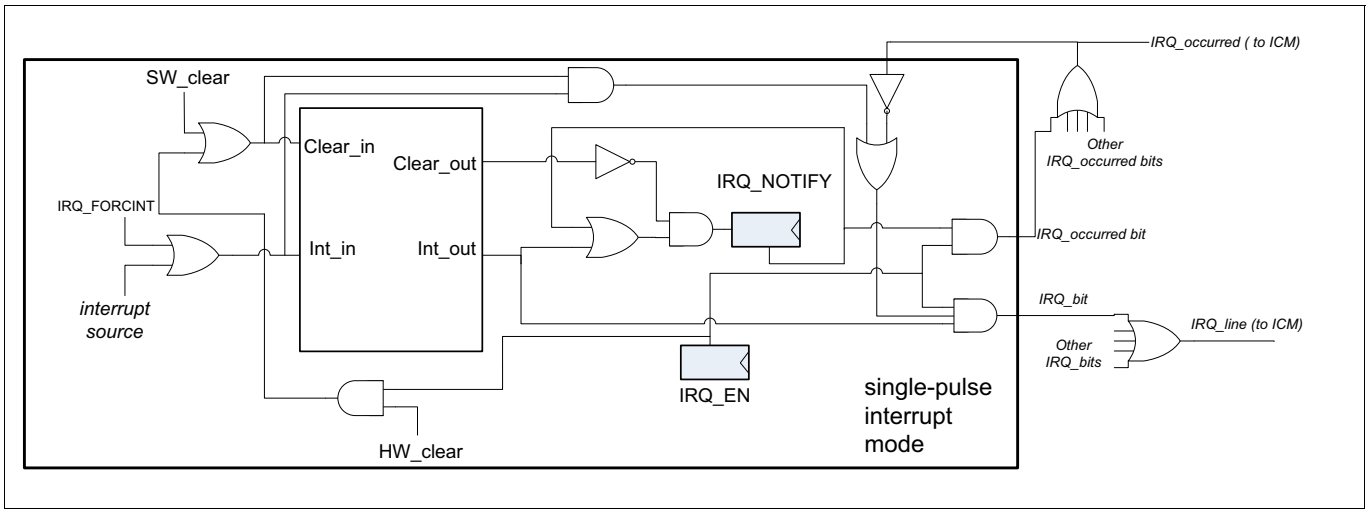

Figure 22 Single-pulse interrupt mode scheme

To avoid unexpected IRQ behavior in the single pulse mode, all desired interrupt sources should be enabled by a single write access to **IRQ_EN** and the notification bits should be cleared by a single write access to the register **IRQ_NOTIFY**.

The additional error interrupt enable mechanism for single-pulse interrupt is shown below.