28.13 Timer Input Module (TIM)

28.13.1 Overview

The Timer Input Module (TIM) is responsible for filtering and capturing input signals of the GTM. Several characteristics of the input signals can be measured inside the TIM channels. For advanced data processing the detected input characteristics of the TIM module can be routed through the ARU to subsequent processing units of the GTM.

Input characteristics mean either time stamp values of detected input rising or falling edges together with the new signal level or the number of edges received since channel enable together with the actual time stamp or PWM signal duration for a whole PWM period.

The architecture of TIM is shown in **Figure 31**.

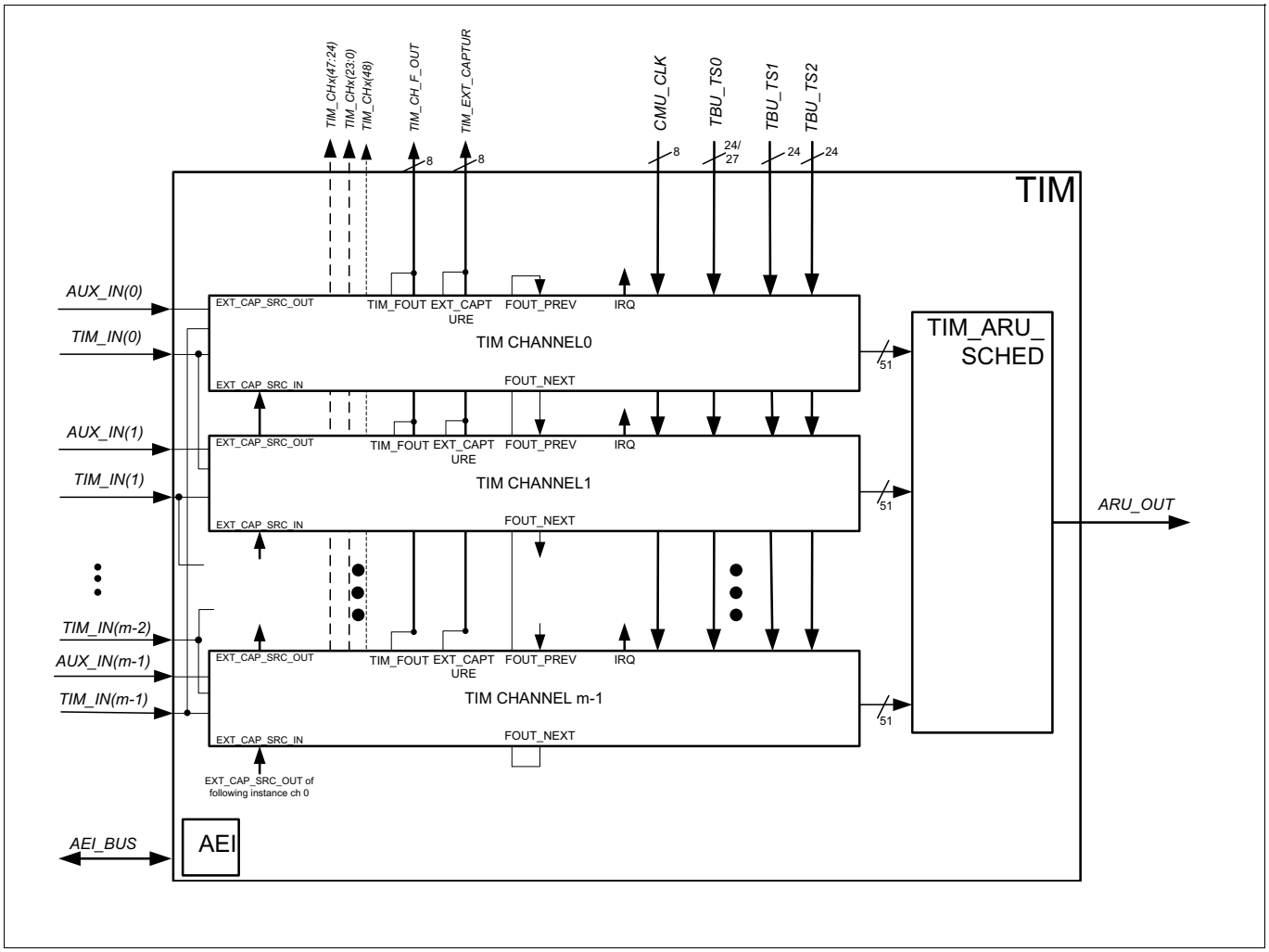


Figure 31 TIM Block Diagram

The number of channels *m* inside a TIM sub-module depends on the device.

Each of the *m* dedicated input signals are filtered inside the FLTx sub-unit of the TIM Module. It should be noted that the incoming input signals are synchronized to the clock SYS_CLK, resulting in a delay of two SYS_CLK periods for the incoming signals.

The measurement values can be read by the CPU directly via the AEI-Bus or they can be routed through the ARU to other sub-modules of the GTM.

For the GTM TIM0 sub-module only, the dashed signal outputs *TIM[i]_CH[x](23:0)*, *TIM[i]_CH[x](47:24)* and *TIM[i]_CH[x](48)* come from the TIM0 sub-module channels zero (0) to five (5) and are connected to MAP sub-module. There, they are used for further processing and for routing to the DPLL.

The two (three) time bases coming from the TBU are connected to the TIM channels to annotate time stamps to incoming signals. For TIM0 the extended 27 bit width time base TBU_TS0 is connected to the TIM channels, and the user has to select if the lower 24 bits (*TBU_TS0(23…0)*) or the higher 24 bits (*TBU_TS0(26…3)*) are stored inside the **GPR0** and **GPR1** registers.

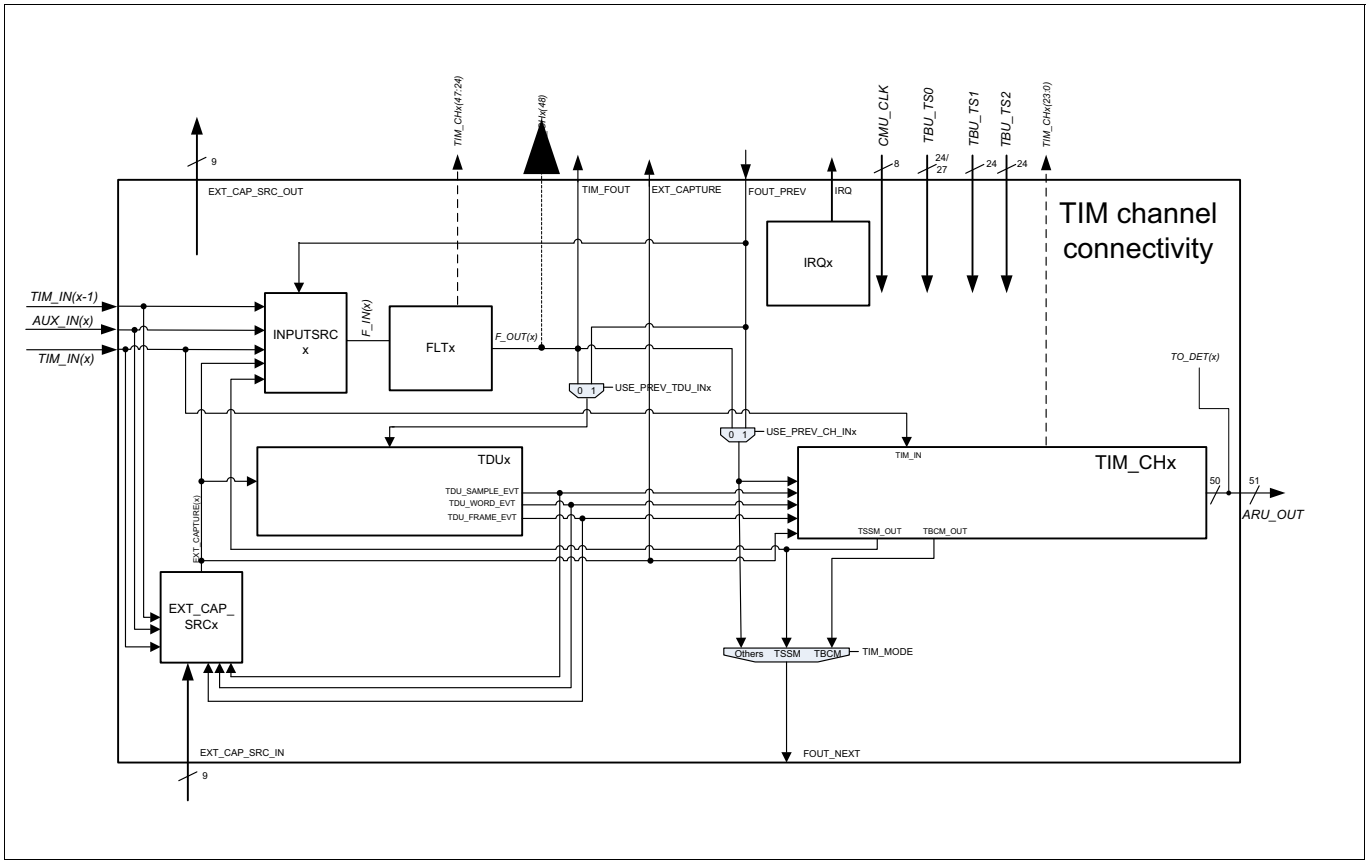


Figure 32 TIM channel internal connectivity

Above figure gives an overview of the channel internal connectivity of the sub units. The sub units with the major functionality are listed next:

- INPUT_SRCx: Select signal for processing by the Filter unit FLTx
- FLTx: The filter unit provides different filter mechanisms described in more detail in **Section 28.13.2**.
- TDUx: Timeout detection unit (no subsequent edge detected during a specified duration
- TIM_CHx: Measurement unit; different measurements strategies configurable on the filtered signal
- IRQx: Local interrupt controller (enabling, status, …)
- EXT_CAP_SRCx: Selects a local signal ext_capture(x) which is needed by certain functions

Details are given in the next chapters.

Depending on the values of the configuration bit fields **USE_PREV_TDU_INx**, **USE_PREV_CH_INx** it is possible to operate on the signal of the local channel x or the previous channel x-1.

Depending on the value of the configuration bit field **TIM_MODEx** it is possible to provide different signals (via FOUT_NEXT) to the next channel.

In TBCM mode each capture event selected by the sensitive edges (CNTS) will be forwarded with the value of ECNT[0] to the following channel (via FOUT_NEXT).

28.13.1.1 Input source selection INPUTSRCx

It can be configured which source shall be used for processing in the FLT,TDU,TIM_CH units. It can be selected by the bit fields **CICTRL** and **MODE_x**, **VAL_x** in the register **TIM[i]_IN_SRC** which source is in use.

Alternatively the signal **F_IN(x)** can be generated by a 8 bit lookup table, which allows to define any function of 3 input sources.

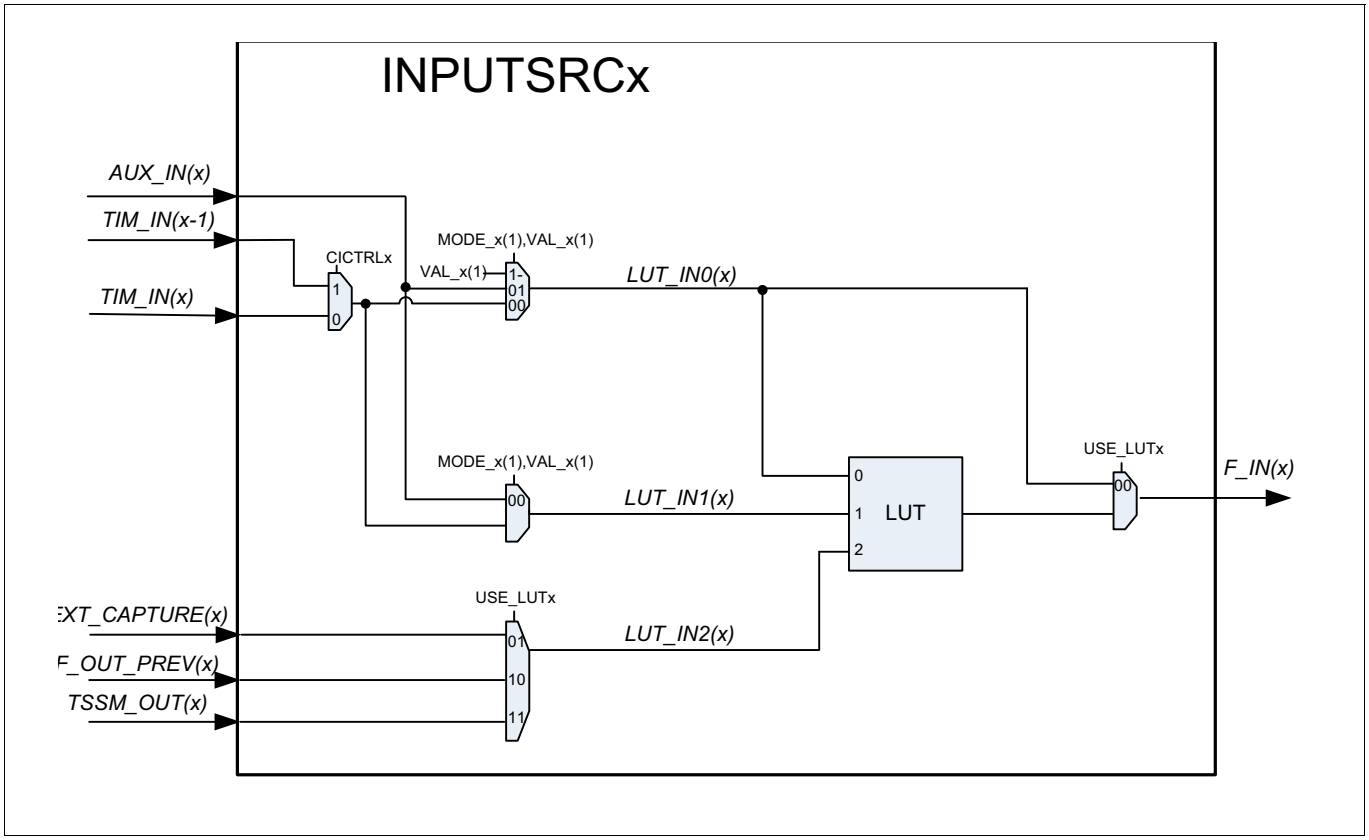


Figure 33 INPUTSRC Block Diagram

If **USE_LUT**=b00 is set the lookup table signal generation is bypassed and the signal selection is performed as follows:

In a certain **MODE_x**, **VAL_x** combination the input signal *F_IN(x)* can be driven by **VAL_x(1)** with 0 or 1 directly.

Due to the fact that all 8 channels are bundled in the register **TIM[i]_IN_SRC** a synchronous control of all 8 input channels is possible.

Two adjacent channels can be combined by setting the **CICTRL** bit field in the corresponding **TIM[i]_CH[x]_CTRL** register. This allows for a combination of complex measurements on one input signal with two TIM channels.

The additional signal **AUX_IN[x]** can be selected as an input signal. The source of this signal is defined in the subchapter “TIM auxiliary input multiplexing”.

If **USE_LUT** !=0b00 is set, the lookup table signal generation with following inputs is in use. See **Figure 33**:

Input **LUT_INO(x)** selection:

TIM_IN(x) if **CICTRLx**=0 and **MODE_x**(1)=0 and **VAL_x**(1)=0
TIM_IN(x-1) if **CICTRLx**=1 and MODE_x(1)=0 and VAL_x(1)=0

AUX_IN(x) if MODE_x(1)=0 and VAL_x(1)=1
VAL_x(1) if MODE_x(1)=1

Input **LUT_IN1(x)** selection:

AUX_IN(x) if **MODE_x**(1)=0 and **VAL_x**(1)=0
TIM_IN(x) if **CICTRLx**=0
TIM_IN(x-1) if **CICTRLx**=1

Input **LUT_IN2(x)** selection:
EXT_CAPTURE(x) if **USE_LUT**=0b01
FOUT_PREV(x) if **USE_LUT**=0b10
TSSM_OUT(x) if **USE_LUT**=0b11

The lookup table is defined by the contents of the bit field **TO_CNT2x**. The lookup_table_index is defined by **LUT_IN2**(x) & **LUT_IN1**(x) & **LUT_IN0**(x). The signal **F_IN**(x) is generated by **TO_CNT2x**[lookup_table_index].

If **USE_LUT** !=0b00 is set, only limited functionality is available in the TDU. See bit field Slicing (**SLICING**) in the register TIM[i]_CH[x]_TDUV.

28.13.1.2 Input observation

It is possible to observe for all channels of one instance by reading **TIM_INP_VAL** the actual signal values of the following processing stages:

- TIM_IN(7:0) signals after TIM input synchronization
- TIM F_IN(7:0) signals after TIM INPUTSRC selection (input to TIM_FLT)
- TIM F_OUT(7:0) signals after TIM filter functionality (output of TIM_FLT)

28.13.1.3 External capture source selection EXTCAPSRCx

Each channel can operate on an external capture signal **EXT_CAPTURE**. The source to use for this signal can be configured by the bit field **EXT_CAP_SRCx** in the register TIM[i]_CH[x]_ECTRL

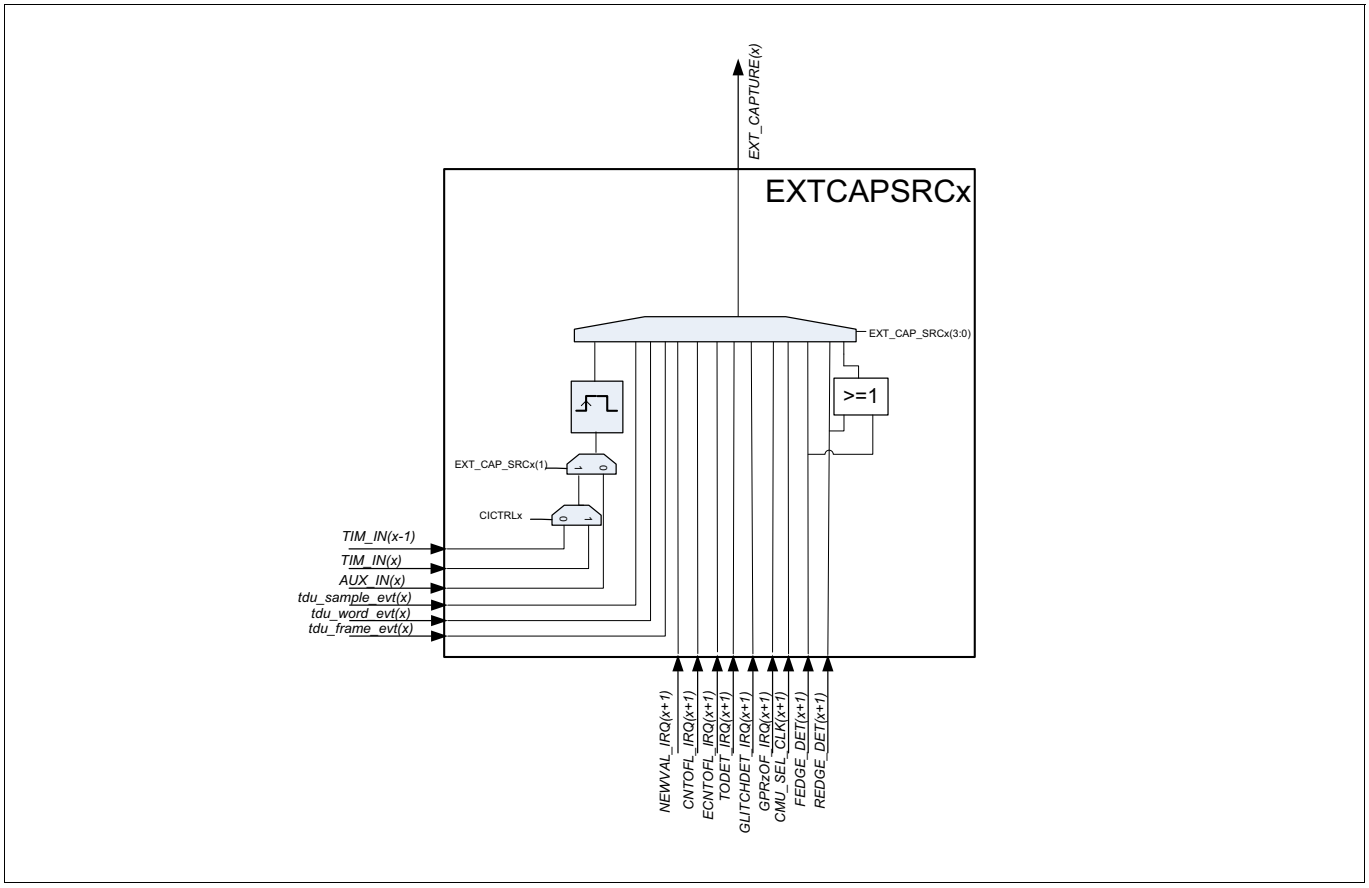


Figure 34 EXTCAPSRC Block Diagram

The external capture functionality can be enabled for the TIM channel **x** with the bit **EXT_CAP_EN** in the register **TIM[i]_CH[x]_CTRL**, it will trigger on each rising edge. A pulse generation for each rising edge of the selected input signal **TIM_IN[x]** and **AUX_IN[x]** is applied.

The six TIM channel interrupt sources can be triggered by the operation in the certain TIM channel modes. Alternatively they can be issued by a soft trigger using the corresponding bits in the register **TIM[i]_CH[x+1]_FORCINT**.

28.13.2 TIM Filter Functionality (FLT)

28.13.2.1 Overview

The TIM sub-module provides a configurable filter mechanism for each input signal. These filter mechanism is provided inside the FLT sub-unit. The FLT architecture is shown in **Figure 35**.

The filter includes a clock synchronization unit (CSU), an edge detection unit (EDU), and a filter counter associated to the filter unit (FLTU).

The CSU is synchronizing the incoming signal *F_IN* to the selected filter clock frequency, which is controlled with the bit field **FLT_CNT_FRQ** of register **TIM[i]_CH[x]_CTRL**.

The synchronized input signal *F_IN_SYNC* is used for further processing within the filter.

It should be noted that glitches with a duration go less than the selected CMU clock period is lost.

The filter modes can be applied individually to the falling and rising edges of an input signal. The following filter modes are available:

- immediate edge propagation mode,
- individual de-glitch time mode (up/down counter), and
- individual de-glitch time mode (hold counter).
- individual de-glitch time mode (reset counter).

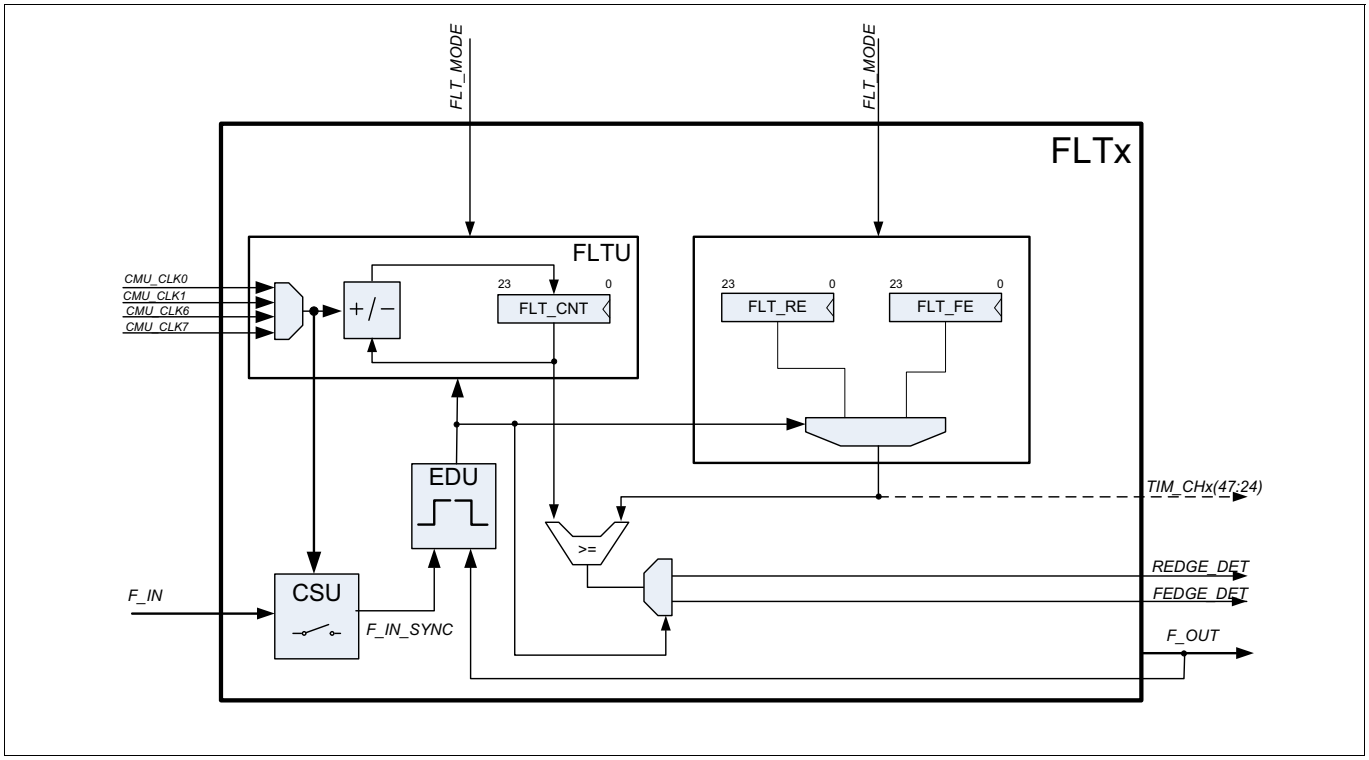


Figure 35 FLT Architecture

The filter parameters (deglitching and acceptance time) for the rising and falling edge can be configured inside the two filter parameter registers **FLT_RE** (rising edge) and **FLT_FE** (falling edge). The exact meaning of the parameter depends on the filter mode.

However the delay time T of both filter parameters **FLT_xE** can always be determined by:

T=(FLT_xE+1)*T_{FLT_CLK},

When a glitch is detected on an input signal a status flag **GLITCHDET** is set inside the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

Table 28 gives an overview about the meanings for the registers **FLT_RE** and **FLT_FE**. In the individual deglitching time modes, the actual filter threshold for a detected regular edge is provided on the *TIM[i]_CH[x](47:24)* output line. In the case of immediate edge propagation mode, a value of zero is provided on the *TIM[i]_CH[x](47:24)* output line.

The *TIM[i]_CH[x](47:24)* output line is used by the MAP sub-module for further processing (please see chapter “TIM0 Input Mapping Module (MAP)”).

Table 28 Filter Parameter summary for the different Filter Modes

Filter mode	Meaning of FLT_RE	Meaning of FLT_FE
Immediate edge propagation	Acceptance time for rising edge	Acceptance time for falling edge
Individual de-glitch time (up/down counter)	De-glitch time for rising edge	De-glitch time for falling edge

Table 28 Filter Parameter summary for the different Filter Modes (cont'd)

Filter mode	Meaning of FLT_RE	Meaning of FLT_FE
Individual de-glitch time (hold counter)	De-glitch time for rising edge	De-glitch time for falling edge
Individual de-glitch time (reset counter)	De-glitch time for rising edge	De-glitch time for falling edge

A counter **FLT_CNT** is used to measure the glitch and acceptance times.

The frequency of the **FLT_CNT** counter is configurable in bit field **FLT_CNT_FRQ** of register **TIM[i]_CH[x]_CTRL**.

The counter **FLT_CNT** can either be clock with the *CMU_CLK0, CMU_CLK1, CMU_CLK6* or the *CMU_CLK7* signal. These signals are coming from the CMU sub-module.

The **FLT_CNT**, **FLT_FE** and **FLT_RE** registers are 24-bit width. For example, when the resolution of the *CMU_CLK0* signal is 50ns this allows maximal de-glitch and acceptance times of about 838ms for the filter.

28.13.2.2 TIM Filter Modes

28.13.2.2.1 Immediate Edge Propagation Mode

In immediate edge propagation mode after detection of an edge the new signal level on *F_IN_SYNC* is propagated to *F_OUT* with a delay of one T_{period} and the new signal level remains unchanged until the configured acceptance time expires.

For each edge type the acceptance time can be specified separately in the **FLT_RE** and **FLT_FE** registers.

Each signal change on the input *F_IN_SYNC* during the duration of the acceptance time has no effect on the output signal level *F_OUT* of the filter but it sets the glitch **GLITCHDET** bit in the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

After it expires an acceptance time the input signal *F_IN_SYNC* is observed and on signal level change the filter raises a new detected edge and the new signal level is propagated to *F_OUT*.

Independent of a signal level change the value of *F_OUT* is always set to *F_IN_SYNC,* when the acceptance time expires (see also **Figure 37**).

Figure 36 shows an example for the immediate edge propagation mode, in the case of rising edge detection. Both, the signal before filtering (*F_IN*) and after filtering (*F_OUT*) are shown. The acceptance time *at1* is specified in the register **FLT_RE**.

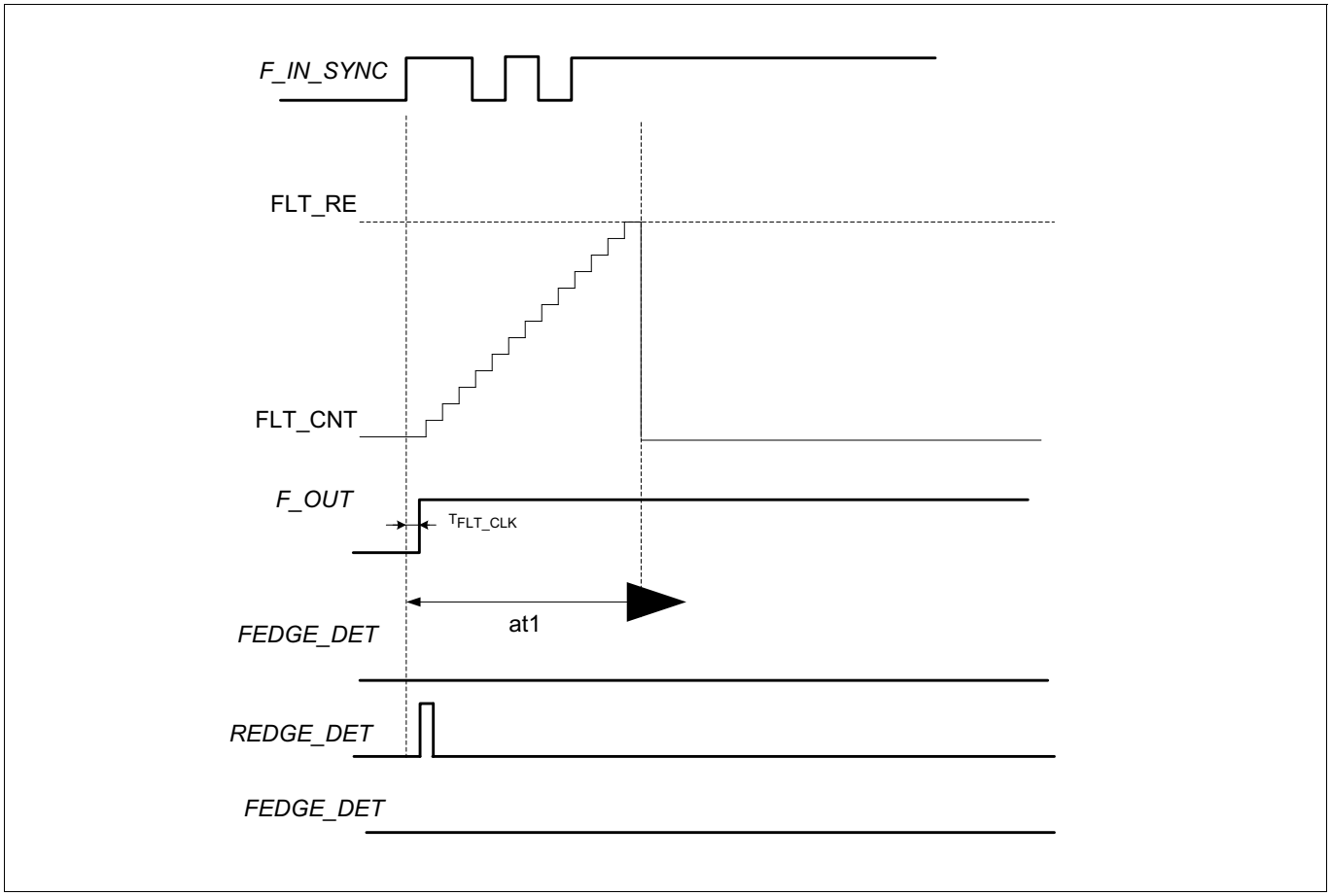


Figure 36 Immediate Edge Propagation Mode in the case of a rising edge

In immediate edge propagation mode the glitch measurement mechanism is not applied to the edge detection. Detected edges on *F_IN_SYNC* are transferred directly to *F_OUT*.

The counter **FLT_CNT** is incremented until acceptance time threshold is reached.

Figure 37 shows a more complex example of the TIM filter, in which both, rising and falling edges are configured in immediate edge propagation mode.

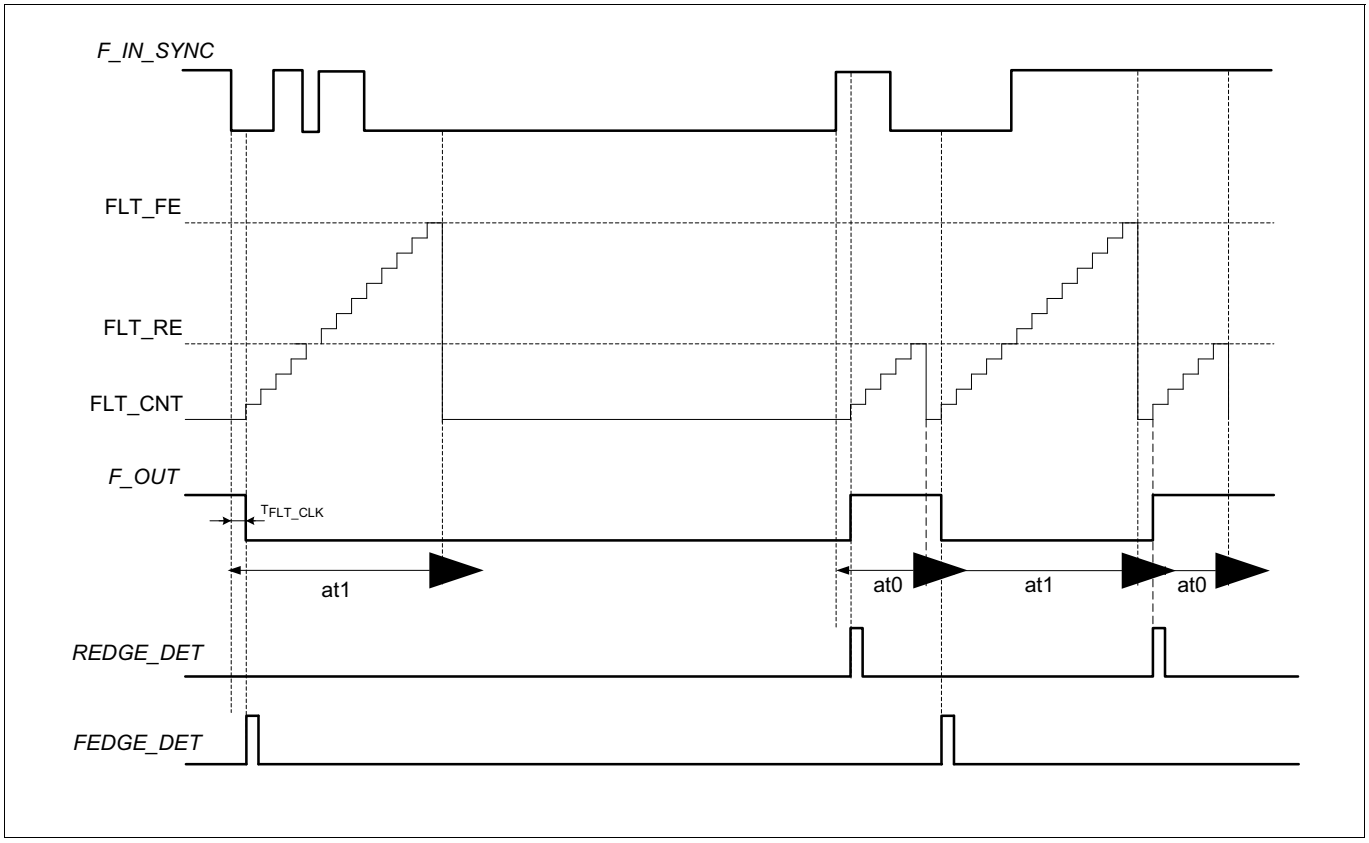


Figure 37 Immediate Edge Propagation Mode in the case of a rising and falling edge

If the **FLT_CNT** has reached the acceptance time for a specific signal edge and the signal *F_IN_SYNC* has already changed to the opposite level of *F_OUT*, the opposite signal level is set to *F_OUT* and the acceptance time measurement is started immediately. **Figure 37** shows this scenario at the detection of the first rising edge and the second falling edge.

28.13.2.2.2 Individual De-glitch Time Mode (up/down counter)

In individual de-glitch time mode (up/down counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers **FLT_RE** and **FLT_FE,** respectively.

The filter counter register **FLT_CNT** is incremented when the signal level on *F_IN_SYNC* is unequal to the signal level on *F_OUT* and decremented if *F_IN_SYNC* equals *F_OUT*.

After **FLT_CNT** has reached a value of zero during decrementing the counter is stopped immediately.

If a glitch is detected a glitch detection bit **GLITCHDET** is set in the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

The detected edge signal together with the new signal level is propagated to *F_OUT* after the individual de-glitch threshold is reached. **Figure 38** shows the behavior of the filter in individual de-glitch time (up/down counter) mode in the case of the rising edge detection.

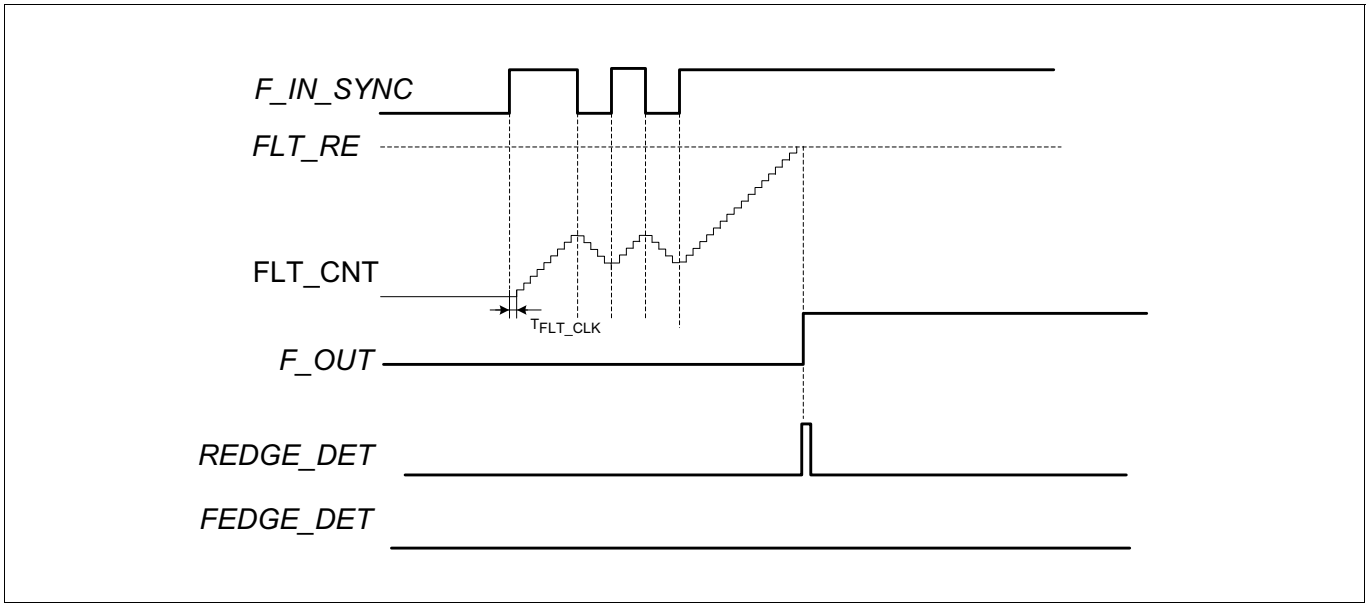


Figure 38 Individual De-glitch Time Mode (up/down counter) in the case of a rising edge

28.13.2.2.3 Individual De-glitch Time Mode (hold counter)

In individual de-glitch time mode (hold counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers **FLT_RE** and **FLT_FE,** respectively.

The filter counter register **FLT_CNT** is incremented when the signal level on *F_IN_SYNC* is unequal to the signal level on *F_OUT* and the counter value of **FLT_CNT** is hold if *F_IN* equals *F_OUT*.

If a glitch is detected the glitch detection bit **GLITCHDET** is set in the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

The detected edge signal together with the new signal level is propagated to *F_OUT* after the individual de-glitch threshold is reached. **Figure 39** shows the behavior of the filter in individual de-glitch time (hold counter) mode in the case of the rising edge detection.

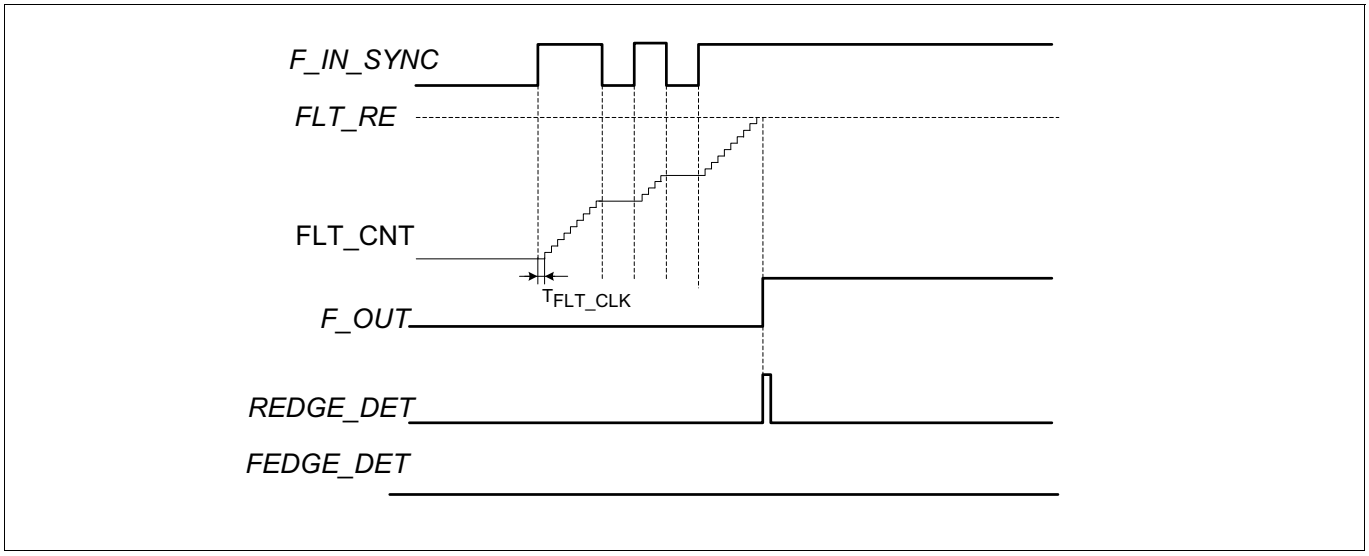


Figure 39 Individual De-glitch Time Mode (hold counter) in the case of a rising edge

28.13.2.2.4 Individual De-glitch Time Mode (reset counter)

In individual de-glitch time mode (reset counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers **FLT_RE** and **FLT_FE,** respectively.

The filter counter register **FLT_CNT** is incremented when the signal level on *F_IN_SYNC* is unequal to the signal level on *F_OUT* and the counter value of **FLT_CNT** is reset to 0x000000 if *F_IN* equals *F_OUT*.

If a glitch is detected the glitch detection bit **GLITCHDET** is set in the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

The detected edge signal together with the new signal level is propagated to *F_OUT* after the individual de-glitch threshold is reached. **Figure 40** shows the behavior of the filter in individual de-glitch time (reset counter) mode in the case of the rising edge detection.

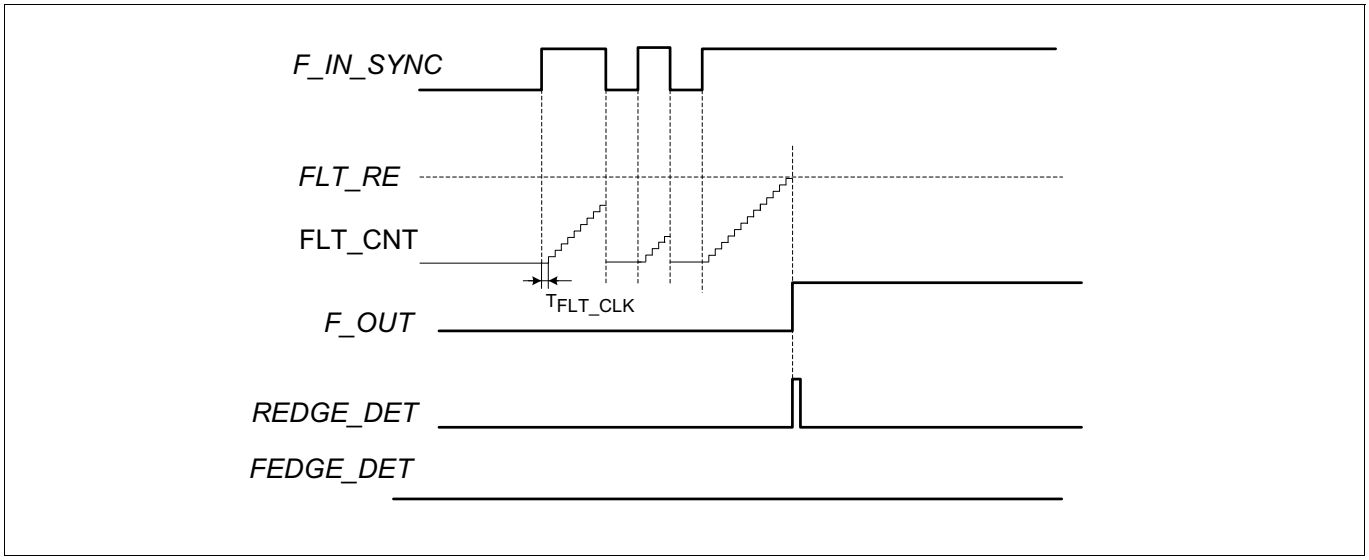


Figure 40 Individual De-glitch Time Mode (reset counter) in the case of a rising edge

28.13.2.2.5 Immediate Edge Propagation and Individual De-glitch Mode

As already mentioned, the four different filter modes can be applied individually to each edge of the measured signal.

However, if one edge is configured with immediate edge propagation and the other edge with an individual deglitching mode (whether up/down counter, hold counter or reset counter) a special consideration has to be applied.

Assume that the rising edge is configured for immediate edge propagation and the falling edge with individual deglitching mode (up/down counter) as shown in **Figure 41**.

If the falling edge of the incoming signal already occurs during the measuring of the acceptance time of the rising edge, the measurement of the deglitching time on the falling edge is started delayed, but immediately after the acceptance time measurement phase of the rising edge has finished.

Consequently, the deglitching counter cannot measure the time T_{ERROR}, as shown in **Figure 41**.

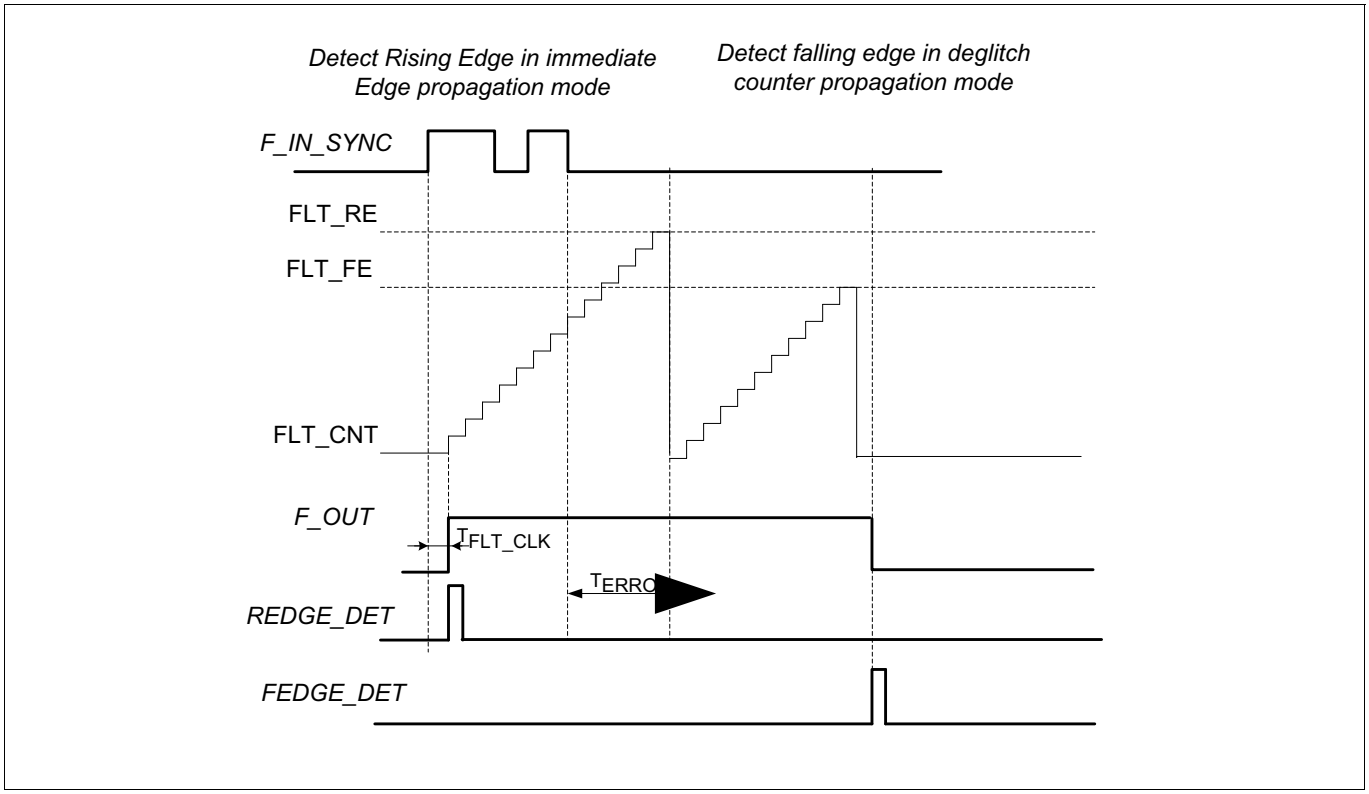


Figure 41 Mixed mode measurement

28.13.2.3 TIM Filter re-configuration

If **FLT_EN**=1 a change of or **FLT_FE** will take place immediately.

If **FLT_EN**=1 a change of **FLT_MODE_RE** or **FLT_MODE_FE** will be used with the next occurring corresponding edge. If the mode is changed while the filter unit is processing a certain mode, it will end this edge filtering in the mode as started.

If **FLT_EN**=1 a change of **FLT_CTR_RE**, **FLT_CTR_FE**, **EFLT_CTR_RE** or **EFLT_CTR_FE** will take place immediately.

28.13.3 Timeout Detection Unit (TDU)

The Timeout Detection Unit (TDU) is responsible for timeout detection of the TIM input signals.

Each channel of the TIM sub-module has its own Timeout Detection Unit (TDU) where a timeout event can be set up on the filtered input signal of the corresponding channel.

In each timeout unit exist 3 8 bit counter/comparator slices. A counter/comparator slice is shown below. The counter TO_CNT will increment by signal *INC*. The counter can be loaded with the value *LOAD_VAL* if *LOAD_VAL*= 1. GT_EVT will be 1 if TO_CNT > TOV is fulfilled. EQ_EVT will be 1 if TO_CNT = TOV is fulfilled.

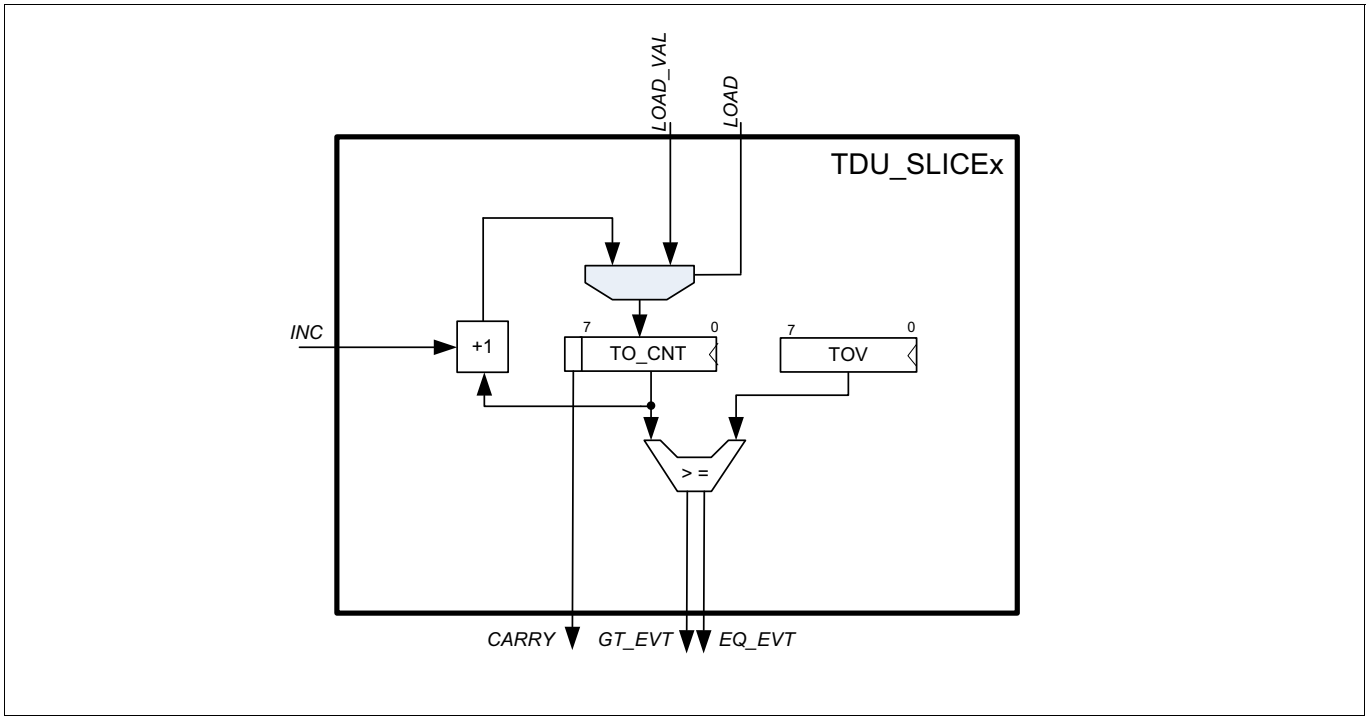


Figure 42 Counter/comparator slice

The counter/comparator slices can be cascaded depending on the application needs to operate as:

- 3x 8-bit counter
- 1x 16-bit counter and 1x 8-bit counter
- 1x 24-bit counter
- 2x 8-bit counter

This allows the user to use the functions:

- timeout on input signals
- local CMU clock prescaler 8 bit
- trigger event generation 8-bit (external capture, todet_irq)

in parallel.

With usage of the 3x 8-bit counter it is possible to define different timeout values for the 2 signal levels.

Following table shows which functions can be used in parallel.

28.13.3.1 Used parallel functions

Table 29 Used parallel functions)

Counter type	Timeout functionality	Generate local TIM CMU clk	Source for external capture to previous channel	Source for TODET_IRQ
24 bit	24 bit	no	tdu_timeout_evt tdu_sample_evt	tdu_timeout_evt tdu_sample_evt
1 x 8 bit 1 x 16 bit	16 bit local clk tdu_sample_evt usable	yes	tdu_timeout_evt, tdu_frame_evt, tdu_sample_evt	tdu_timeout_evt, tdu_frame_evt, tdu_sample_evt
3x 8 bit	8 bit local clk tdu_sample_evt usable	yes	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt
3x 8 bit	no	yes	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt
2x 8 bit	no	no	tdu_timeout_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_word_evt, tdu_frame_evt

Next table shows which of the available 8 bit resources are cascaded with a chosen **SLICING**.

28.13.3.2 Which of the available 8 bit resources are cascaded with a chosen SLICING

Table 30 Which of the available 8 bit resources are cascaded with a chosen SLICING

Counter type	Counters count on	Counter resource generates	CLK selection
24 bit	CNT on TCS	CNT= TO_CNT2 & TO_CNT1 & TO_CNT; TCMP = TOV2 & TOV1 & TOV; CNT >= TCMP generates tdu_sample_evt tdu_timeout_evt = tdu_sample_evt tdu_frame_evt = 0 tdu_word_evt = 0	TCS selected
3x 8 bit	TO_CNT2 on TCS TO_CNT on tdu_sample_evt TO_CNT1 on tdu_word_evt	TO_CNT2 >= TOV2 generates tdu_sample_evt TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt	TO_CNT2: TCS selected TO_CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EVT=1 TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK=0

Table 30 Which of the available 8 bit resources are cascaded with a chosen SLICING (cont'd)

Counter type	Counters count on	Counter resource generates	CLK selection
3x 8 bit	TO_CNT2 on TCS TO_CNT on tdu_sample_evt TO_CNT1 on tdu_sample_evt	TO_CNT2 >= TOV2 generates tdu_sample_evt TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt or tdu_frame_evt	TO_CNT2: TCS selected TO_CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EVT=1 TO_CNT1: tdu_sample_evt selected with TDU_SAME_CNT_CLK=1
3x 8 bit	TO_CNT2 on TCS TO_CNT on TCS TO_CNT1 on tdu_word_evt	TO_CNT2 >= TOV2 generates tdu_sample_evt TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt	TO_CNT2: TCS selected TO_CNT: TCS selected with TCS_USE_SAMPLE_EVT=0 TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK=0
3x 8 bit	TO_CNT2 on TCS TO_CNT on TCS TO_CNT1 on TCS	TO_CNT2 >= TOV2 generates tdu_sample_evt TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt or tdu_frame_evt	TO_CNT2: TCS selected TO_CNT: TCS selected with TCS_USE_SAMPLE_EVT=0 TO_CNT1: TCS selected with TDU_SAME_CNT_CLK=1
2x 8 bit	TO_CNT on TCS TO_CNT1 on tdu_word_evt	TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt tdu_sample_evt = 0	TO_CNT: TCS selected TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK=0
2x 8 bit	TO_CNT on TCS TO_CNT1 on TCS	TO_CNT >= TOV generates tdu_word_evt TO_CNT1 >= TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt tdu_sample_evt = 0	TO_CNT: TCS selected TO_CNT1: TCS selected with TDU_SAME_CNT_CLK=1

Table 30 Which of the available 8 bit resources are cascaded with a chosen SLICING (cont'd)

Counter type	Counters count on	Counter resource generates	CLK selection
1 x 8 bit 1 x 16 bit	TO_CNT2 on TCS CNT on TCS	TO_CNT2 >= TOV2 generates tdu_sample_evt CNT = TO_CNT1 & TO_CNT; TCMP = TOV1 & TOV; CNT >= TCMP generates tdu_frame_evt tdu_timeout_evt = tdu_frame_evt tdu_word_evt = 0	TO_CNT2: TCS selected CNT: TCS selected with TCS_USE_SAMPLE_EVT=0
1 x 8 bit 1 x 16 bit	TO_CNT2 on TCS CNT on tdu_sample_evt	[CDATA]TO_CNT2 >= TOV2 generates tdu_sample_evt CNT = TO_CNT1 & TO_CNT; TCMP = TOV1 & TOV; CNT >= TCMP generates tdu_frame_evt tdu_timeout_evt = tdu_frame_evt tdu_word_evt = 0	TO_CNT2: TCS selected CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EVT=1

Based on a chosen counter configuration by **SLICING** it is possible to control the start behavior of the counters by **TDU_START** in multiple ways. In addition the stopping of the counters can be controlled by **TDU_STOP**. Depending on the application needs it can be decided how the individual counter slices can be reset/reloaded by the configuration field **TDU_RESYNC**.

Depending on the counter configuration, up to 4 internal compare events tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt out of the 3 comparator slices can be generated. It can be chosen by **TODET_IRQ_SRC** which shall be used as *TIM_TODETx_IRQ* signal which will be accessible by the **TODET** bit inside the **TIM[i]_CH[x]_IRQ_NOTIFY** register

The TDU architecture is shown in **Figure 43**.

28.13.3.3 Architecture of the TDU Sub-unit

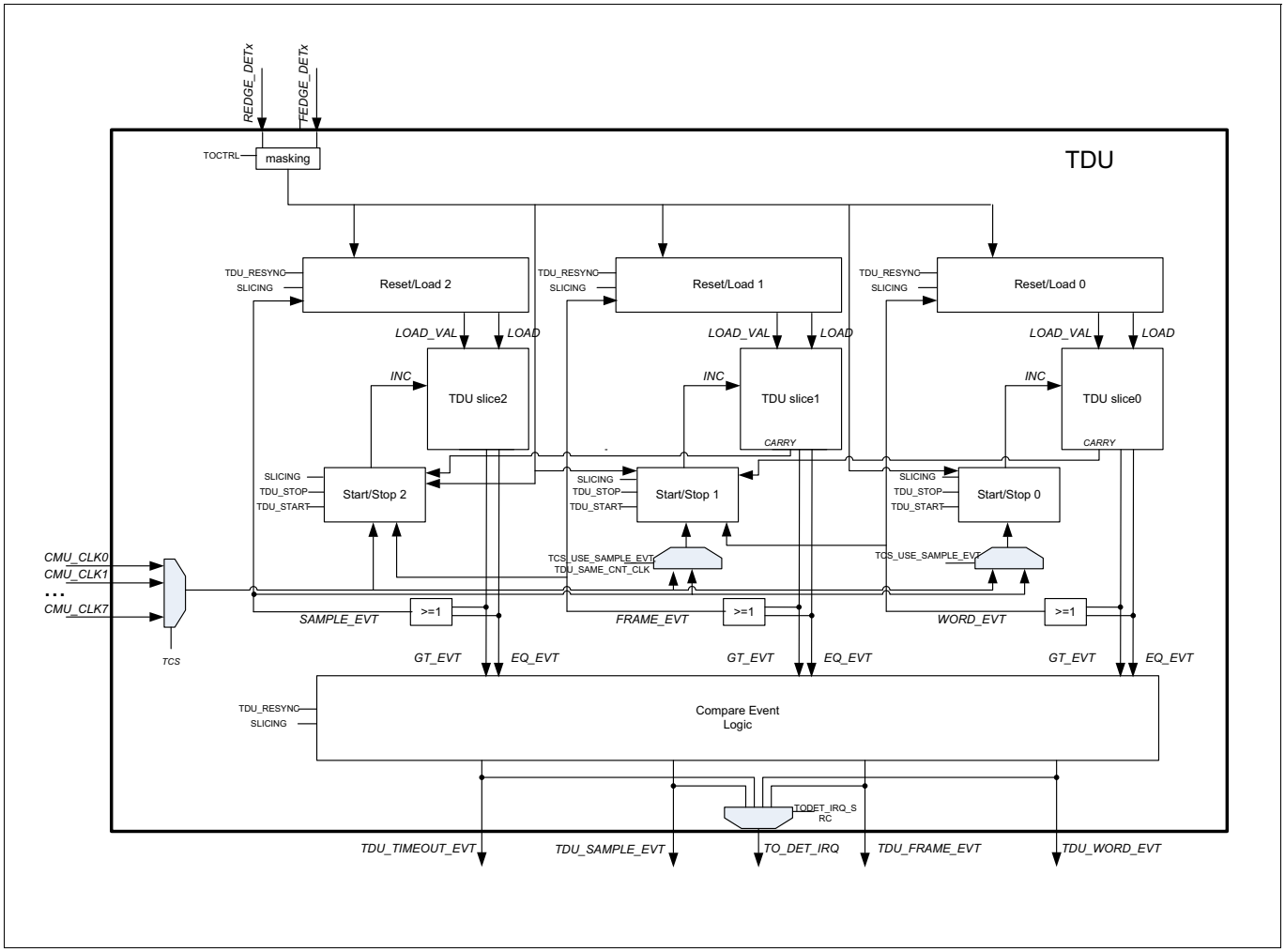


Figure 43 Architecture of the TDU Sub-unit

Each TDU_slice has its own start/stop control, based on the chosen configuration it will decide if the counter inside the TDU slice will increment on the resolution of the applied clock/event. The reset/load control decides based on the configuration settings and the compare result of the TDU slices those the counters **TO_CNT**,**TO_CNT1**,**TO_CNT2** have to be reloaded. Depending on the chosen counter/compare configuration the compare event logic will generate based on the compare results of the 3 TDU slices and the chosen resolution the events *tdu_sample_evt*, *tdu_word_evt*, *tdu_frame_evt*.

The primary resolution on which the TDU is working can be specified with the bit field **TCS** of the register **TIM[i]_CH[x]_TDUV**. The corresponding input signal *CMU_CLKx* will be used to clock the TDU. The individual timeout/counter values have to be specified in number of ticks of the selected input clock signal in the fields **TOV**, **TOV1**, **TOV2** of the timeout value register **TIM[i]_CH[x]_TDUV** of the TIM channel x.

In case of cascading the bit slices by usage of **SLICING** and **TCS_USE_SAMPLE_EVT** and **TDU_SAME_CLK** the resolution for counting can be switched to the events tdu_sample_evt or tdu_word_evt. More details see table above.

The counter compare units start operation on occurrence of the first "start event" configured by **TDU_START**. They continue their operation until the first "stop event" configured by **TDU_STOP** occurs.

In case of occurrence of a start event and a compare/count resolution event in the same clock cycle, the counters will increment or reload/reset based on **TDU_RESYNC** immediately. No *tdu_sample_evt*, *tdu_word_evt*, *tdu_frame_evt* will be generated.

In case of occurrence of a stop event the counters will not change their values. In case of occurrence of a stop event and a compare/count resolution event in the same clock cycle the corresponding events *tdu_sample_evt*, *tdu_word_evt*, *tdu_frame_evt* will be generated.

In case of occurrence of a start event and a stop event in the same clock cycle the counters will not change their values. No *tdu_sample_evt*, *tdu_word_evt*, *tdu_frame_evt* will be generated.

The function of the timeout unit (configured to **TDU_RESYNC**=0000,**TDU_START**=000) can be started or stopped inside the **TIM[i]_CH[x]_CTRL** register by setting/resetting the **TOCTRL** bit.

Timeout detection can be enabled to be sensitive to falling, rising or both edges of the input signal by writing the corresponding values to the bit field **TOCTRL**.

The TDU generates an interrupt signal *TIM_TODETx_IRQ* whenever a timeout is detected for an individual input signal, and the **TODET** bit is set inside the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

In addition, when the ARU access is enabled with the **ARU_EN** bit inside the **TIM[i]_CH[x]_CTRL** register, the actual values stored inside the registers **TIM[i]_CH[x]_GPR0** and **TIM[i]_CH[x]_GPR1** are sent together with the last stored signal level to the ARU if a timeout event *TDU_TIMEOUT_EVT* occurs.

To signal that a timeout occurred, the ARU_OUT(50) bit (ACB(2)) is set. The bit ACB(0) will be updated with the timeout event to the signal level on which the timeout was detected. Timeout signaling with ACB(2) is only possible with **TODET_IRQ_SRC**= 0000.

Thus, a destination could determine if a timeout occurred at the TIM input by evaluating ACB bit 2.

Since the TIM channel still monitors its input pin although the timeout happened, a valid edge could occur at the input pin while the timeout information is still valid at the ARU. In that case, the new edge associated data is stored inside the registers **TIM[i]_CH[x]_GPR0** and **TIM[i]_CH[x]_GPR1**, the GPR overflow detected bit is set together in the ACB field (ACB(1)) with the timeout bit (ACB(2)) and the values are marked as valid to the ARU.

The ACB bit 2 is cleared, when a successful ARU write access by the TIM channel took place.

The ACB bit 1 is cleared, when a successful ARU write access by the TIM channel took place.

When a valid edge initiates an ARU write access which has not ended while a new timeout occurs the GPR overflow detected bit (ACB(1)) is set. The bit ACB(0) will be updated to the level on which the timeout occurred.

When a timeout occurred and initiates an ARU write access which has not ended while a new timeout occurs the GPR overflow detected bit (ACB(1)) is not set.

The following table clarifies the meaning of the ACB Bits for valid data provided by a TIM channel:

Table 31 ACB Bits for valid data provided by a TIM channel

ACB4/3	ACB2	ACB1	ACB0	Description
dc	0	0	SL	Valid edge detected
dc	0	1	SL	Input edge overwritten by subsequent edge
dc	1	0	SL	Timeout detected without valid edge
dc	1	1	SL	Timeout detected with subsequent valid edge detected

28.13.4 TIM Channel Architecture

28.13.4.1 Overview

Each TIM channel consist of an input edge counter **ECNT**, a Signal Measurement Unit (SMU) with a counter **CNT**, a counter shadow register **CNTS** for SMU counter and two general purpose registers **GPR0** and **GPR1** for value storage.

The value **TOV** of the timeout register **TIM[i]_CH[x]_TDU** is provided to TDU sub-unit of each individual channel for timeout measurement. The architecture of the TIM channel is depicted in **Figure 44**.

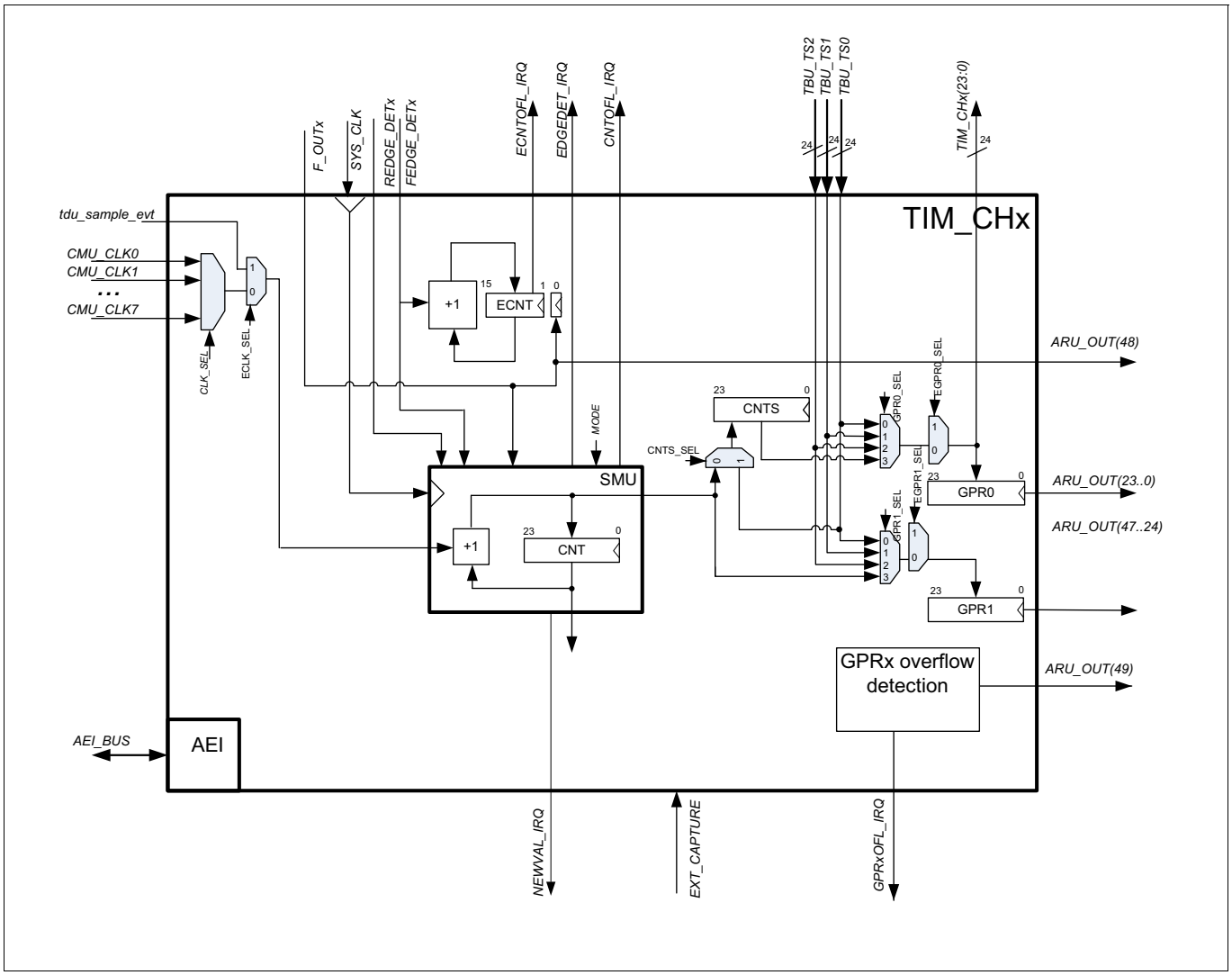


Figure 44 TIM Channel Architecture

Each TIM channel receives both input trigger signals *REDGE_DETx* and *FEDGE_DETx,* generated by the corresponding filter module in order to signalize a detected echo of the input signal *F_INx*. The signal *F_OUTx* shows the filtered signal of the channel's input signal *F_INx*.

The edge counter **ECNT** counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of **ECNT**. (However, the actual counter implementation counts only falling edges on ECNT[n:1] bits. It generates **ECNT** by composing the ECNT[n:1] bits with F_OUTx as bit 0).

Thus, the whole ECNT counter value is always odd, when a positive edge was received and always even, when a negative edge was received.

The current **ECNT[7:0]** register content is made visible on the bits 31 down to 24 of the registers **GPR0**, **GPR1,** and **CNTS**. This allows the software to detect inconsistent read accesses to registers **GPR0**, **GPR1**, and **CNTS**. However, the update strategy of these registers depends on the selected TIM modes, and thus the consistency check has to be adapted carefully.

It can be chosen with the bit field **FR_ECNT_OFL** when an **ECNT** overflow is signaled on **ECNTOFL**. An ECNT overflow can be signaled on 8 bit or full range resolution.

While reading the register **TIM[i]_CH[x]_ECNT** the bit **ECNT[0]** shows the input signal value F_OUTx independent of the state (enabled / disabled) of the channel. If a channel gets disabled (OSM mode or resetting TIM_EN) the content of **TIM[i]_CH[x]_ECNT** will be frozen until a read of the register takes place. This read will reset the **ECNT** counter. Continuing reads will show the input signal value in bit **ECNT[0]** again.

When new data is written into **GPR0** and **GPR1** the **NEWVAL** bit is set in **TIM[i]_CH[x]_IRQ_NOTIFY** register and depending on corresponding enable bit value the *NEWVALx_IRQ* interrupt is raised.

Each TIM input channel has an ARU connection for providing data via the ARU to the other GTM sub-modules. The data provided to the ARU depends on the TIM channel mode and its corresponding adjustments (e.g. multiplexer configuration).

The bit **ARU_EN** of register **TIM[i]_CH[x]_CTRL** decides, whether the measurement results of registers **GPR0** and **GPR1** are consumed by another sub-module via ARU (**ARU_EN** = 1) or the CPU via AEI (**ARU_EN** = 0).

To guarantee a consistent delivery of data from the **GPR0** and **GPR1** registers to the ARU or the CPU each TIM channel has to ensure that the data is consumed before it is overwritten with new values.

If new data was produced by the TIM channel (bit **NEWVAL** is set inside **TIM[i]_CH[x]_IRQ_NOTIFY** register) while the old data is not consumed by the ARU (**ARU_EN** = 1) or CPU (**ARU_EN** = 0), the TIM channel sets the **GPROFL** bit inside the status register **TIM[i]_CH[x]_IRQ_NOTIFY** and it overwrites the data inside the registers **GPR0** and **GPR1**. In addition when **ARU_EN**=1 the bit ACB(1) is set to 1 to indicate the overflow in the ARU data.

If the CPU is selected as consumer for the registers **GPR0** and **GPR1** (**ARU_EN** = 0), the acknowledge for reading out data is performed by a read access to the register **GPR0**. Thus, register **GPR1** should be read always before **GPR0**.

If the ARU is selected as consumer for the registers **GPR0** and **GPR1** (**ARU_EN** = 1), the acknowledge for reading out data is performed by the ARU itself. However, the registers **GPR0** and **GPR1** could be read by CPU without giving an acknowledge.

28.13.4.2 TIM Channel Modes

The TIM provides seven different measurement modes that can be configured with the bit field **TIM_MODE** of register **TIM[i]_CH[x]_CTRL**. The measurement modes are described in the following subsections. Besides these different basic measurement modes, there exist distinct configuration bits in the register **TIM[i]_CH[x]_CTRL** for a more detailed controlling of each mode. The meanings of these bits are as follows:

- **DSL**: control the signal level for the measurement modes (e.g. if a measurement is started with rising edge or falling edge, or if high level pulses or low level pulses are measured.
- **EGPR0_SEL, GPR0_SEL** and **EGPR1_SEL, GPR1_SEL**: control the actual content of the registers **GPR0** and **GPR1** after a measurement has finished.
- **CNTS_SEL**: control the content of the registers **CNTS.** The actual time for updating the **CNTS** register is mode dependent.
- **OSM**: activate measurement in one-shot mode or continuous mode. In one-shot mode only one measurement cycle is performed and after that the channel is disabled.
- **NEWVAL**: The NEWVAL IRQ interrupt is triggered at the end of a measurement cycle, signaling that the registers GPR0 and GPR1 are updated.
- **ARU_EN**: enables sending of the registers **GPR0** and **GPR1** together with the actual signal level (in bit 48) and the overflow signal GPROFL (in bit 49), and the timeout status information (bit 50) to the ARU.
- **EXT_CAP_EN**: forces an update of the registers **GPR0** and **GPR1** and **CNTS** (TIM channel mode dependent) only on each rising edge of the EXT_CAPTURE signal and triggers a NEWVAL IRQ interrupt. If this mode is disabled the NEWVAL IRQ interrupt is triggered at the end of each measurement cycle.

For each channel the source of the EXT_CAPTURE signal can be configured with the bit fields **EXT_CAP_SRC** in the register **TIM[i]_CH[x]_ECTRL.**

28.13.4.2.1 TIM PWM Measurement Mode (TPWM)

In TIM PWM Measurement Mode the TIM channel measures duty cycle and period of an incoming PWM signal. The **DSL** bit defines the polarity of the PWM signal to be measured.

When measurement of pulse high time and period is requested (PWM with a high level duty cycle, **DSL**=1) and **IMM_START**=0, the channel starts measuring after the first rising edge is detected by the filter.

If **IMM_START**=1 the measurement starts immediately after activating the channel by **TIM_EN**=1.

Measurement is done with the **CNT** register counting with the configured clock coming from **CMU_CLKx** until a falling edge is detected.

Assume: **SWAP_CAPTURE**=0, **ECNT_RESET**=0

Then the counter value is stored inside the shadow register **CNTS** (if **CNTS_SEL** = 0) and the counter **CNT** counts continuously until the next rising edge is reached.

On this following rising edge the content of the **CNTS** register is transferred to **GPR0** and the content of **CNT** register is transferred to **GPR1**, assuming settings for the selectors **EGPR0_SEL**=0,**GPR0_SEL**=11 and **EGPR1_SEL**=0,**GPR1_SEL**=11. By this, **GPR0** contains the duty cycle length and **GPR1** contains the period. It should be noted, that the bits 1 to 7 of the **ECNT** may be used to check data consistency of the registers **GPR0** and **GPR1**.

In addition the **CNT** register is cleared **NEWVAL** status bit inside of **TIM[i]_CH[x]_IRQ_NOTIFY** status register and depending on corresponding interrupt enable condition *TIM_NEWVALx_IRQ* interrupt is raised.

The CNTS register update is not performed until the measurement is started. Afterwards each edge leaving the level defined by DSL is performing a CNTS register update.

If a PWM with a low level duty cycle should be measured (**DSL** = 0) and **IMM_START**=0, the channel waits for a falling edge until measurement is started. On this edge the low level duty cycle time is stored first in **CNTS** and then finally in **GPR0** and the period is stored in **GPR1**.

When a PWM period was successfully measured, the data in the registers **GPR0** and **GPR1** is marked as valid for reading by the ARU when the **ARU_EN** bit is set inside **TIM[i]_CH[x]_CTRL** register, the **NEWVAL** bit is set inside the **TIM[i]_CH[x]_IRQ_NOTIFY** register, and a new measurement is started.

If the preceding PWM values were not consumed by a reader attached to the ARU (**ARU_EN** bit enabled) or by the CPU the TIM channel set **GPROFL** status bit in **TIM[i]_CH[x]_IRQ_NOTIFY** and depending on corresponding interrupt enable bit value raises a *GPROFL_IRQ* and overwrites the old values in **GPR0** and **GPR1**. A new measurement is started afterwards.

If the register **CNT** produces an overflow during the measurement, the bit **CNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_CNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

If the register **ECNT** produces an overflow during the measurement, the bit **ECNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_ECNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

If **ECNT_RESET**=0 the counter **CNT** will be reset to 0 on active edge (defined by **DSL**) of the input signal. If **ECNT_RESET**=1 the counter **CNT** will be reset to 0 on each edge of the input signal.

Assume **EXT_CAP_EN**=0 and **SWAP_CAPTURE**=0:On every input edge to the active level defined by **DSL** will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**. Every edge to the inactive level will capture the data selected by **CNTS_SEL** to the registers **CNTS**.

Assume **EXT_CAP_EN**=0 and **SWAP_CAPTURE**=1:On every input edge to the inactive level defined by **DSL** will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**. Every edge to the active level will capture the data selected by **CNTS_SEL** to the registers **CNTS**.

External capture TIM PWM Measurement Mode (TPWM)

If external capture is enabled **EXT_CAP_EN**=1, the PWM measurement is done continuously. The actual measurement values are captured to GPRx if an external capture event occurs.

On every external capture event the data selected by **CNTS_SEL, EGPR0_SEL**, **GPR0_SEL** will be captured to the registers **CNTS, GPR0.**

If **SWAP_CAPTURE**=0 every external capture event will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**. Every input edge to the level != **DSL** will capture the data selected by **CNTS_SEL** to the registers **CNTS**.

If **SWAP_CAPTURE**=1 every input edge to the inactive level != **DSL** will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**.

Assume **SWAP_CAPTURE**=0:

Operation is done depending on CMU clock, **ISL, DSL** bit and the input signal value defined in next table (Assume CNTS_SEL= 0):

Table 32 Operation depending on CMU clock, ISL, DSL and the input signal value (Assume CNTS_SEL= 0)

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description
0	1	0	-	0	CNT++
1	1	0	-	0	no
rising edge	-	0	0	0	capture CNT value in CNTS
falling edge	-	0	0	0	CNT=0
rising edge	-	0	1	0	no
falling edge	-	0	1	0	capture CNT value in CNTS; CNT=0
1	1	0	-	1	CNT++
0	1	0	-	1	no
falling edge	-	0	0	1	capture CNT value in CNTS
rising edge	-	0	0	1	CNT=0
falling edge	-	0	1	1	no
rising edge	-	0	1	1	capture CNT value in CNTS; CNT=0
-	-	rising edge	-	-	do GPRx capture; issue NEWVAL_IRQ
-	0	0	-	-	no

The CNTS register update is not performed until the measurement is started (first edge defined by DSL is detected). Afterwards the update of the CNTS register is defined by ISL,DSL combinations in the table above.

28.13.4.2.2 TIM Pulse Integration Mode (TPIM)

In TIM Pulse Integration Mode each TIM channel is able to measure a sum of pulse high or low times on an input signal, depending on the selected signal level bit **DSL** of register **TIM[i]_CH[x]_CTRL** register.

If **IMM_START**=0 the pulse integration measurement is started with occurrence of the first edge defined by DSL on the input signal. If **IMM_START**=1 the measurement starts immediately after activating the channel by **TIM_EN**=1.

The pulse times are measured by incrementing the TIM channel counter **CNT** until the counter is stopped with occurrence of a input signal edge to the opposite signal level defined by **DSL**.

The counter **CNT** counts with the *CMU_CLKx* clock specified by the *CLK_SEL* bit field of the **TIM[i]_CH[x]_CTRL** register.

The **CNT** register is reset at the time the channel is activated (enabling via AEI write access) and it accumulates pulses while the channel is staying enabled.

Assume **EXT_CAP_EN**=0 and **SWAP_CAPTURE**=0:After measurement is started, every falling(DSL=1) or rising(DSL=0) input edge will issue a *TIM_NEWVALx_IRQ* interrupt, and the registers **CNTS**, **GPR0** and **GPR1** are updated according to settings of its corresponding input multiplexers, using the bits **EGPR0_SEL**, **EGPR1_SEL**, **GPR0_SEL**, **GPR1_SEL** and **CNTS_SEL**. It should be noted, that the bits 1 to 7 of the **ECNT** may be used to check data consistency of the registers **GPR0** and **GPR1**.

Assume **EXT_CAP_EN**=0 and **SWAP_CAPTURE**=1:After measurement is started, every falling(DSL=1) or rising(DSL=0) input edge will issue a *TIM_NEWVALx_IRQ* interrupt, and the registers **CNTS**, **GPR0** are updated according to settings of its corresponding input multiplexers, using the bits **EGPR0_SEL**, **GPR0_SEL** and **CNTS_SEL**.

Every input edge to active level defined by **DSL** (rising DSL=1;falling DSL=0) will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**.

When the **ARU_EN** bit is set inside the **TIM[i]_CH[x]_CTRL** register the measurement results of the registers **GPR0** and **GPR1** can be send to subsequent sub-modules attached to the ARU.

External capture TIM Pulse Integration Mode (TPIM)

If external capture is enabled **EXT_CAP_EN**=1, the pulse integration is done until next external capture event occurs.

On every external capture event the data selected by **CNTS_SEL, EGPR0_SEL**, **GPR0_SEL** will be captured to the registers **CNTS, GPR0.**

If **SWAP_CAPTURE**=0 every external capture event will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**.

If **SWAP_CAPTURE**=1 every input edge to the inactive level != **DSL** will capture the data selected by **EGPR1_SEL**, **GPR1_SEL** to the registers **GPR1**.

Assume **SWAP_CAPTURE**=0; **IMM_START**=0:

Operation is done depending on CMU clock, **DSL** bit and the input signal value defined in next table (inc_cnt = false if TIM channel is enabled):

Table 33 Operation depending on CMU clock, DSL and the input signal value (inc_cnt = false if TIM channel is enabled)

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description
falling edge	-	0	-	0	inc_cnt = true
rising edge	-	0	-	0	if inc_cnt == true then {do capture GPRx, CNTS; issue NEWVAL_IRQ} inc_cnt = false
rising edge	-	0	-	1	inc_cnt = true
falling edge	-	0	-	1	if inc_cnt == true then {do capture GPRx, CNTS; issue NEWVAL_IRQ} inc_cnt = false

Table 33 Operation depending on CMU clock, DSL and the input signal value (inc_cnt = false if TIM channel is enabled) (cont'd)

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description
-	1	0	-	-	if inc_cnt == true then CNT++;
-	-	rising edge	-	-	do capture GPRx, CNTS; issue NEWVAL_IRQ; CNT=0
-	0	0	-	-	no

28.13.4.2.3 TIM Input Event Mode (TIEM)

In TIM Input Event Mode the TIM channel is able to count edges.

It is configurable if rising, falling or both edges should be counted. This can be done with the bit fields **DSL** and **ISL** in **TIM[i]_CH[x]_CTRL** register.

In addition, a *TIM[i]_NEWVAL[x]_IRQ* interrupt is raised when the configured edge was received and this interrupt was enabled.

The counter register **CNT** is used to count the number of edges, and the bit fields **EGPR0_SE**L, **EGPR1_SE**L,**GPR0_SEL**, **GPR1_SEL**, and **CNTS_SEL** can be used to configure the desired update values for the registers **GPR0**, **GPR1** and **CNTS**. These register are updated whenever the edge counter **CNT** is incremented due to the arrival of a desired edge.

If the preceding data was not consumed by a reader attached to the ARU or by the CPU the TIM channel sets **GPROFL** status bit and raises a *GPROFL[x]_IRQ* if it was enabled in **TIM[i]_CH[x]_IRQ_EN** register and overwrites the old values in **GPR0** and **GPR1** with the new ones.

If the register **CNT** produces an overflow during the measurement, the bit **CNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_CNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

If the register **ECNT** produces an overflow during the measurement, the bit **ECNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_ECNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

The TIM Input Event Mode does not depend on the bit field **CLK_SEL** of register **TIM[i]_CH[x]_CTRL**.

External capture TIM Input Event Mode (TIEM)

If external capture is enabled, capturing is done depending on the **DSL**, **ISL** bit and the input signal value defined in next table:

Table 34 Capturing depended on the DSL, ISL and the input signal value, if external capture is enabled

Input signal F_OUTx	External capture	ISL	DSL	Action description
-	rising edge	1	-	do capture; issue NEWVAL_IRQ; CNT++
-	0	1	-	no
1	rising edge	0	1	do capture; issue NEWVAL_IRQ; CNT++
0	-	0	1	no
0	rising edge	0	0	do capture; issue NEWVAL_IRQ; CNT++
1	-	0	0	no

28.13.4.2.4 TIM Input Prescaler Mode (TIPM)

In the TIM Input Prescaler Mode the number of edges which should be detected before a *TIM[i]_NEWVAL[x]_IRQ* is raised is programmable. In this mode it must be specified in the **CNTS** register after how many edges the interrupt has to be raised.

A value of 0 in **CNTS** means that after one edge an interrupt is raised and a value of 1 means that after two edges an interrupt is raised, and so on.

The edges to be counted can be selected by the bit fields **DSL** and **ISL** of register **TIM[i]_CH[x]_CTRL**.

With each triggered interrupt, the registers **GPR0** and **GPR1** are updated according to bits **EGPR0_SE**L, **EGPR1_SE**L,**GPR0_SEL** and **GPR1_SEL**.

If the register **ECNT** produces an overflow during the measurement, the bit **ECNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_ECNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

The TIM Input Prescaler Mode does not depend on the bit field **CLK_SEL** of register **TIM[i]_CH[x]_CTRL**.

External capture TIM Input Prescaler Mode (TIPM)

If external capture is enabled, the external capture events are counted instead of the input signal edges.

Operation is done depending on the external capture signal, **DSL**, **ISL** bit and the input signal value defined in next table:

Table 35 Operation depending on the external capture signal, DSL, ISL and the input signal value

Input signal F_OUTx	External capture	ISL	DSL	Action description
-	rising edge	1	-	if CNT >= CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
-	0	1	-	no
1	rising edge	0	1	if CNT >= CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	-	0	1	no
0	rising edge	0	0	if CNT >= CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
1	-	0	0	no

28.13.4.2.5 TIM Bit Compression Mode (TBCM)

The TIM Bit Compression Mode can be used to combine all filtered input signals of a TIM sub-module to a parallel *m* bit data word, which can be routed to the ARU, where *m* is the number of channels available in the TIM sub-module.

Figure 45 gives an overview of the TIM bit compression mode.

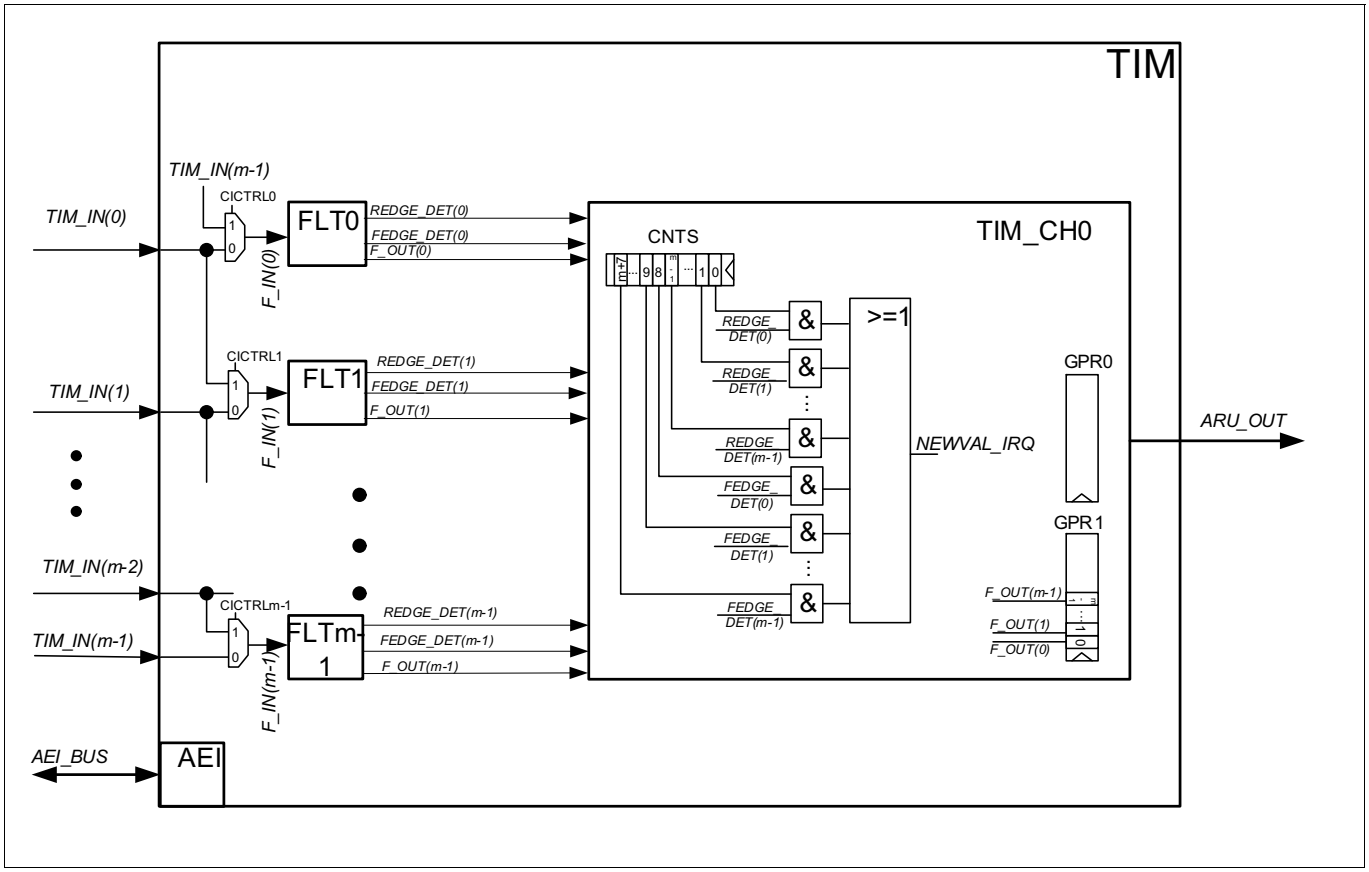


Figure 45 TIM Bit Compression Mode

The register **CNTS** of a channel is used to configure the event that releases the *NEWVAL_IRQ* and samples the input signals F_IN(0) to F_IN(*m*-1) in ascending order as a parallel data word in **GPR1**.

The bits 0 to *m-1* of the **CNTS** register are used to select the *REDGE_DET* signals of the TIM filters 0 to *m-1* as a sampling event, and the bits 8 to (7+*m*) are used to select the *FEDGE_DET* signals of the TIM filters 0 to *m-1*, respectively. If multiple events are selected, the events are OR-combined (see also **Figure 45**).

EGPR0_SEL,**GPR0_SEL** selects the timestamp value, which is routed through the ARU. **GPR1_SEL** is not applicable in TBCM mode.

If the bit **ARU_EN** of register **TIM[i]_CH[x]_CTRL** is set, the sampled data of register **GPR1** is routed together with a time stamp of register **GPR0** to the ARU, whenever the NEWVAL_IRQ is released.

In TIM Bit compression mode, the register **ECNT** increments with each *NEWVAL_IRQ*, which means that the value of ECNT may depend on all *m* input signals. Consequently, the LSB of **ECNT** does not reflect the actual level of the input signal TIM_IN(x).

If the register **ECNT** produces an overflow during the measurement, the bit **ECNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_ECNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

The TIM Bit Compression Mode does not depend on the bit field **CLK_SEL** of register **TIM[i]_CH[x]_CTRL**.

External capture Bit Compression Mode (TBCM)

If external capture is enabled, capturing is done depending on the **DSL**, **ISL** bit and the input signal value defined in next table:

Table 36 Capturing depended on the DSL, ISL and the input signal value, if external capture is enabled

Input signal F_OUTx	External capture	ISL	DSL	Action description
-	rising edge	1	-	do capture; issue NEWVAL_IRQ; CNT++
-	0	1	-	no
1	rising edge	0	1	do capture; issue NEWVAL_IRQ; CNT++
0	-	0	1	no
0	rising edge	0	0	do capture; issue NEWVAL_IRQ; CNT++
1	-	0	0	no

28.13.4.2.6 TIM Gated Periodic Sampling Mode (TGPS)

In the TIM Gated Periodic Sampling Mode the number of CMU clock cycles which should elapse before capturing and raising *TIM[i]_NEWVAL[x]_IRQ* is programmable. In this mode it must be specified in the **CNTS** register after how many CMU clock cycles the interrupt has to be raised.

A value of 0 in **TIM[i]_CH[x]_CNTS** means that after one **CLK_SEL** edge a trigger/interrupt is raised, and a value of 1 means that after two edges a trigger/interrupt is raised, and so on.

In the **TIM[i]_CH[x]_CNT** register the elapsed cycles were incremented and compared against **TIM[i]_CH[x]_CNTS**. If **TIM[i]_CH[x]_CNT** is greater or equal to **TIM[i]_CH[x]_CNTS** a trigger will be raised. This allows by writing a value to **TIM[i]_CH[x]_CNTS** that the actual period time can be changed on the fly.

Operation is done depending on CMU clock, **DSL**, **ISL** bit and the input signal value defined in next table:

Table 37 Operation depending on CMU clock, DSL, ISL and the input signal value

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description

-	1	0	1	-	if CNT >= CNTS then do capture
					  issue NEWVAL_IRQ;
					  CNT=0
					else
					  CNT++
					endif

0	0	0	0	1	no

1	1	0	0	1	if CNT >= CNTS then do capture;
					  issue NEWVAL_IRQ;
					  CNT=0
					else
					  CNT++
					endif

0	0	-	0	1	no

0	1	0	0	0	if CNT >= CNTS then do capture;
					  issue NEWVAL_IRQ;
					  CNT=0
					else
					  CNT++
					endif

Table 37 Operation depending on CMU clock, DSL, ISL and the input signal value (cont'd)

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description
1	0	0	0	0	no
-	0	0	-	-	no

In this mode the **TIM[i]_CH[x]_GPR1** operates as a shadow register for **TIM[i]_CH[x]_CNTS**. This would allow that the period for the next sampling period could be specified. The update of **TIM[i]_CH[x]_CNTS** will only take place once on a trigger if the **TIM[i]_CH[x]_GPR1** was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from **TIM[i]_CH[x]_GPR1** and afterwards the new sampling period for the next sampling period (the one after the actual sampling period) could be written.

With each triggered interrupt, the registers **GPR0** and **GPR1** are updated according to bits **GPR0_SEL**, **GPR1_SEL, EGPR0_SEL** and **EGPR1_SEL**.

When selecting **ECNT** as a source for the capture registers, GPRx will show the edge count and the input signal value at point of capture. Selecting **GPR0_SEL** = '11' and **EGPR0_SEL** = '0' for TIM channel 0 all 8 TIM input signals will be captured to **GPR0[7:0]**.

In the TGPS Mode the bit field **CLK_SEL** of register **TIM[i]_CH[x]_CTRL** will define the selected CMU clock which will be used.

The behavior of the **ECNT** counter is configurable by **ECNT_RESET**. If set to 1 on each interrupt (period expired) the **ECNT** will be reset. Otherwise it operates in wrap around mode.

If the register **ECNT** produces an overflow during the measurement, the bit **ECNTOFL** is set inside the register **TIM[i]_CH[x]_IRQ_NOTIFY** and interrupt *TIM_ECNTOFL[x]_IRQ* is raised depending on corresponding interrupt enable condition.

External capture TIM Gated Periodic Sampling Mode (TGPS)

If external capture is enabled, the external capture events will capture the GPRx, reset the counter **CNT** and issue a *NEWVAL_IRQ*.

Operation is done depending on the CMU clock, external capture signal, **DSL**, **ISL** bit and the input signal value defined in next table:

Table 38 Operation depending on the CMU clock, external capture signal, DSL, ISL and the input signal value

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description

-	1	0	1	-	if CNT >= CNTS then do
					capture;
					issue NEWVAL_IRQ;
					CNT=0
					else
					CNT++
					endif

0	0	0	0	1	no

Table 38 Operation depending on the CMU clock, external capture signal, DSL, ISL and the input signal value (cont'd)

Input signal F_OUTx	selected CMU Clock	External capture	ISL	DSL	Action description

1	1	0	0	1	if CNT >= CNTS then do
					capture;
					issue NEWVAL_IRQ;
					CNT=0
					else
					CNT++
					endif

0	0	-	0	1	no

0	1	0	0	0	if CNT >= CNTS then do
					capture;
					issue NEWVAL_IRQ;
					CNT=0
					else
					CNT++
					endif

1	0	0	0	0	no
-	0	0	-	-	no
-	-	rising edge	-	-	do capture; issue NEWVAL_IRQ; CNT =0

28.13.4.2.7 TIM Serial Shift Mode (TSSM)

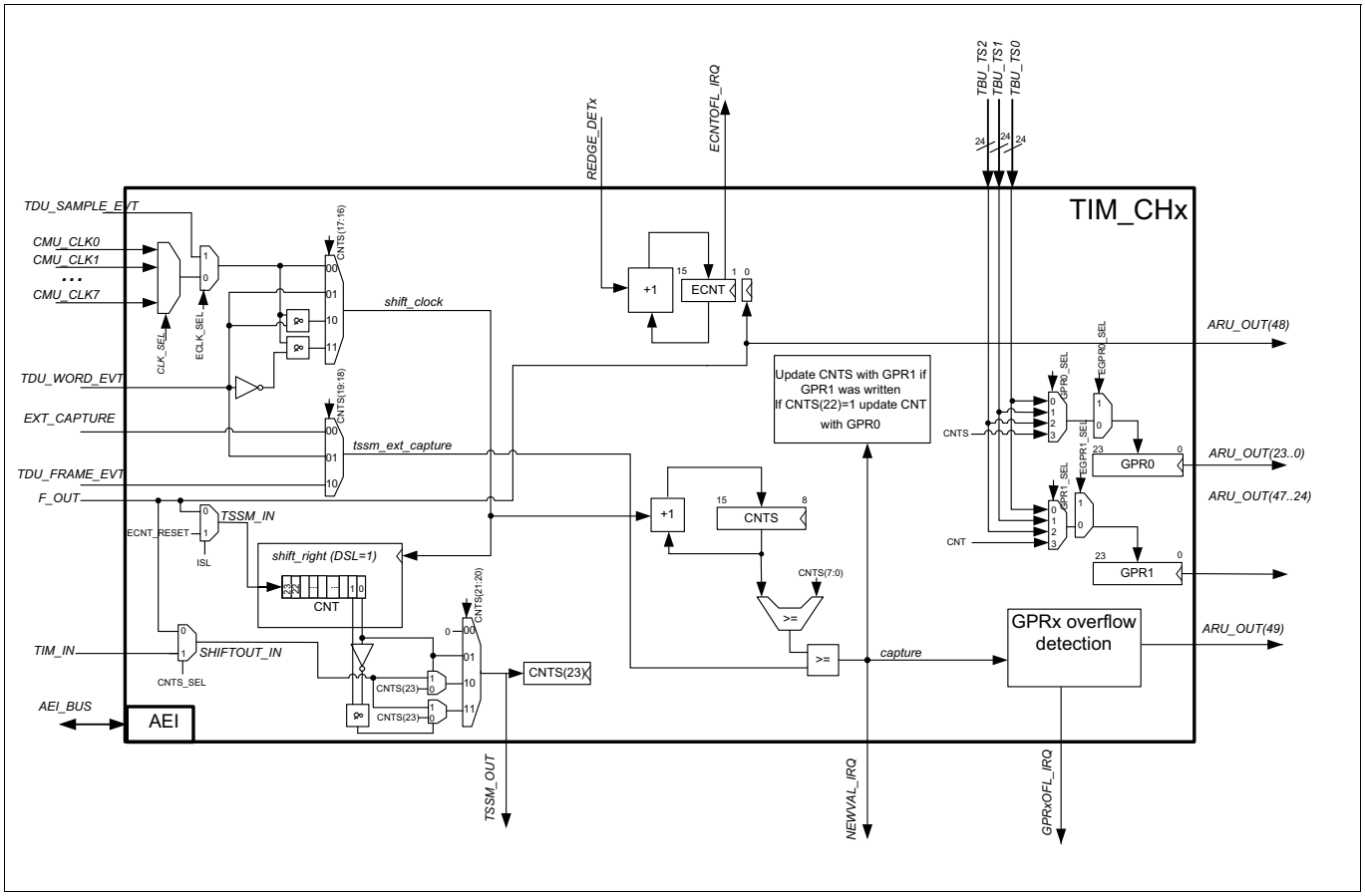


Figure 46 TIM Serial Shift Mode (TSSM)

In the TIM Serial Shift Mode on each shift clock event the actual value of the input signal **TSSM_INx** will be registered in dependence of **DSL** n the register **TIM[i]_CH[x]_CNT**.

If **ISL**=0 is set **FOUTx** will be used as shift in value **TSSM_INx**, with **ISL**=1 the bit field **ECNT_RESET** defines the value for **TSSM_INx**.

With **DSL**=0 **TSSM_INx** will be stored in **TIM[i]_CH[x]_CNT[0]** and **TIM[i]_CH[x]_CNT[22:0]** will be shifted left. With **DSL**</ b>=1 **TSSM_OUTx** will be stored in **TIM[i]_CH[x]_CNT[23]** and **TIM[i]_CH[x]_CNT[23:1]** will be shifted right.

Operation is done depending on the shift clock, external capture signal, **DSL**, **ISL** bit and the input signal value defined in next table:

Table 39 Operation depending on the shift clock, external capture signal, DSL, ISL and the input signal value

Input signal TSSM_INx	shift clock	tssm_ext_capture	ISL	DSL	Action description
-	0	0	-	-	no
-	-	1	-	-	if EXT_CAP_EN=1 then see function table in next chapter else no endif

Table 39 Operation depending on the shift clock, external capture signal, DSL, ISL and the input signal value (cont'd)

Input signal TSSM_INx	shift clock	tssm_ext_capture	ISL	DSL	Action description

value	1	0	0	0	CNT[23:1]= CNT[22:0];
					CNT[0]= value
					if CNTS[15:8] >= CNTS[7:0]
					then
					do capture;
					issue NEWVAL_IRQ;
					CNTS[15:8]=0
					else
					CNTS[15:8]++
					endif

value	1	0	0	1	CNT[22:0]= CNT[23:1];
					CNT[23]= value
					if CNTS[15:8] >= CNTS[7:0]
					then
					do capture;
					issue NEWVAL_IRQ;
					CNTS[15:8]=0
					else
					CNTS[15:8]++
					endif

value	1	0	1	0	CNT[23:1]= CNT[22:0]; CNT[0]= value if CNTS[15:8] >= CNTS[7:0] then do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET else CNTS[15:8]++ endif

value	1	0	1	1	CNT[22:0]= CNT[23:1];
					CNT[23]= value
					if CNTS[15:8] >= CNTS[7:0]
					then
					do capture;
					issue NEWVAL_IRQ;
					CNTS[15:8]=0
					CNT[23:0]=ECNT_RESET
					else
					CNTS[15:8]++
					endif

The register **TIM[i]_CH[x]_CNTS[7:0]** define the amount of bits which will be stored inside **TIM[i]_CH[x]_CNT.**

Each shift clock will increment the register **TIM[i]_CH[x]_CNTS[15:8].** If the condition **TIM[i]_CH[x]_CNTS[15:8] >= TIM[i]_CH[x]_CNTS[7:0]** is met a capture event is raised and *TIM[i]_NEWVAL[x]_IRQ* is asserted.

With each capture event the registers **GPR0** and **GPR1** are updated according to bits **GPR0_SEL**, **GPR1_SEL, EGPR0_SEL** and **EGPR1_SEL**.

If the bit field **ISL** is set to 1 the register bits **TIM[i]_CH[x]_CNT** are set to the value defined by **ECNT_RESET** in case of a capture event.

In this mode the **TIM[i]_CH[x]_GPR1** operates as a shadow register for **TIM[i]_CH[x]_CNTS**. This allows that the amount of bits to sample can be specified. The update of **TIM[i]_CH[x]_CNTS** will only take place once on a trigger if the **TIM[i]_CH[x]_GPR1** was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from **TIM[i]_CH[x]_GPR1** and afterwards the new amount of bits to sample for the next sampling period (the one after the actual sampling period) could be written.

The shift clock which will be in use is selectable by **TIM[i]_CH[x]_CNTS[17:16]**:

$\mathbf{00_B}$: source selection by **USE_TDU_CLK_SRC** is in use. It can be set to any CMU_CLK source or to the local TDU sample clock tdu_sample_evt.

$\mathbf{01_B}$: the tdu_word_evt signal will be used as shift clock source.

$\mathbf{10_B}$: the clk source selected by **USE_TDU_CLK_SRC** is used and gated with tdu_word_evt. If tdu_word_evt=0 then shift clock will be 0.

$\mathbf{11_B}$: the clk source selected by **USE_TDU_CLK_SRC** is used and gated with tdu_word_evt. If tdu_word_evt=1 then shift clock will be 0.

Signal Generation with TIM Serial Shift Mode

If **TIM[i]_CH[x]_CNTS[22]** is 1 the **TIM[i]_CH[x]_GPR0** operates as a shadow register for **TIM[i]_CH[x]_CNT**. This allows that the bits for shifting out can be specified. The update of **TIM[i]_CH[x]_CNT** will only take place once on a trigger if the **TIM[i]_CH[x]_GPR0** was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from **TIM[i]_CH[x]_GPR0** and afterwards the new bits to shift out could be written.

In addition the TIM Serial Shift Mode is able to generate a signal **TSSM_OUT** which can be used internally to the TIM channel.

On each system clock the value for **TSSM_OUT** is generated as defined next. The actual value can be read by the register bit **TIM[i]_CH[x]_CNTS[23]**.

Following functionality for **TSSM_OUTx** is selectable by **TIM[i]_CH[x]_CNTS[21:20]:**

$\mathbf{00_B}$: Constant output; **TSSM_OUTx** = 0.

$\mathbf{10_B}$:Shift output; If **DSL**=0 (shift left) then **TSSM_OUTx** = **TIM[i]_CH[x]_CNT[23]** else (shift right) **TSSM_OUTx** = **TIM[i]_CH[x]_CNT[0]**.

$\mathbf{10_B}$: Latched output; If **DSL**=0 and **TIM[i]_CH[x]_CNT[23]**=1 then **TSSM_OUTx** = **SHIFTOUT_INx** elsif **DSL**=1 and **TIM[i]_CH[x]_CNT[0]**=1 then **TSSM_OUTx** = **SHIFTOUT_INx**.

$\mathbf{11_B}$: Registered output; If **DSL**=0 and **TIM[i]_CH[x]_CNT[23:22]**=b01 then **TSSM_OUTx** = **SHIFTOUT_INx** elsif **DSL**=1 and **TIM[i]_CH[x]_CNT[1:0]**=b10 then **TSSM_OUTx** = **SHIFTOUT_INx**.

In case of registered or latched output mode the signal **SHIFTOUT_INx** is selectable by **CNTS_SEL**.

If **CNTS_SEL**=0 is set **FOUTx** will be used for **SHIFTOUT_INx**, with **CNTS_SEL**=1 the signal **TIM_INx** is in us for **SHIFTOUT_INx**.

External capture TIM Serial Shift Mode (TSSM)

If external capture is enabled (**EXT_CAP_EN**=1), the external capture events will capture the GPRx, reset the counter **CNT** depending on ISL and issue a *NEWVAL_IRQ*. Functionality from previous table will be applied.

The source which will be used as external capture event for TSSM mode is selectable by **TIM[i]_CH[x]_CNTS[19:18]**:

00_B: source selection by **EXT_CAP_SRC** is in use.

01_B: tdu_word_evt signal will be used as source.

10_B: tdu_frame_evt signal will be used as source.

11_B: reserved

Operation is done depending on the shift clock, external capture signal, **DSL**, **ISL** bit and the input signal value defined in next table:

Table 40 Operation depending on the shift clock, external capture signal, DSL, ISL and the input signal value

Input signal F_OUTx	shift clock	tssm_ext_capture	ISL	DSL	Action description
-	0	1	1	-	do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
-	0	1	0	-	do capture; issue NEWVAL_IRQ; CNTS[15:8]=0
value	1	1	1	0	CNT[23:1]= CNT[22:0]; CNT[0]= value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
value	1	1	1	1	CNT[22:0]= CNT[23:1]; CNT[23]= value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
value	1	1	0	0	CNT[23:1]= CNT[22:0]; CNT[0]= value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0
value	1	1	0	1	CNT[22:0]= CNT[23:1]; CNT[23]= value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0

28.13.5 MAP Submodule Interface

The GTM provides one dedicated TIM sub-module TIM0 where channels zero (0) to five (5) are connected to the MAP sub-module described in chapter “TIM0 Input Mapping Module ”. There, the TIM0 sub-module channels provide the input signal level together with the actual filter value and the annotated time stamp for the edge

together in a 49 bit wide signal to the MAP sub-module. This 49 bit wide data signal is marked as valid with a separate valid signal tim0_map_dval[x] (x: 0…5).

28.13.5.1 Structure of map data

Table 41 MAP Submodule Interface

tim0_map_data[x](48)	Signal level bit from tim0_ch[x]
tim0_map_data[x](47:24)	actual filter value TIM0_CH[x]_FLT_RE/ TIM0_CH[x]_FLT_FE if corresponding channel x bit field FLT_MODE_RE/ FLT_MODE_FE is 1 else 0 is assigned.
tim0_map_data[x](23:0)	time stamp value selected by TBU0_SEL, GRP0_SEL, EGPR0_SEL, CNTS_SEL of channel x if bit field TIM_EN= 1
tim0_map_dval[x]	mark tim0_map_data[x] valid for one clock cycle

*Note: With **TIM_EN**=1 the MAP interface starts operation, it is not dependent on the setting of the bit fields **TIM_MODE, ISL, DSL**.*

Note: While the MAP interface is in use the following guidelines have to be fulfilled, otherwise inconsistent filter values can be transferred.

Change **TIM0_CH[x]_FLT_RE** only between occurrence of rising and falling edge.

Change **TIM0_CH[x]_FLT_FE** only between occurrence of falling and rising edge.

28.13.6 TIM Interrupt Signals

Table 42 TIM Interrupt Signals

Signal	Description
TIM[i]_NEWVAL[x]_IRQ	New measurement value detected by SMU of channel x (x: 0…m-1)
TIM[i]_ECNTOFL[x]_IRQ	ECNT counter overflow of channel x (x: 0…m-1)
TIM[i]_CNTOFL[x]_IRQ	SMU CNT counter overflow of channel x (x: 0…m-1)
TIM[i]_GPROFL[x]_IRQ	GPR0 and GPR1 data overflow, old data was not read out before new data has arrived at input pin of channel x (x: 0…m-1)
TIM[i]_TODET[x]_IRQ	Time out reached for input signal of channel x (x: 0…m-1)
TIM[i]_GLITCHDET[x]_IRQ	A glitch was detected by the TIM filter of channel x (x: 0…m-1)

28.13.7 TIM Configuration Register Overview

Table 43 TIM Configuration Register Overview

Register Name	Description	see Page
TIM[i]_CH[x]_CTRL	TIMi channel x control register	178
TIM[i]_CH[x]_ECTRL	TIMi channel x extended control register	194
TIM[i]_CH[x]_FLT_RE	TIMi channel x filter parameter 0 register	182

Table 43 TIM Configuration Register Overview (cont'd)

Register Name	Description	see Page
TIM[i]_CH[x]_FLT_FE	TIMi channel x filter parameter 1 register	183
TIM[i]_CH[x]_TDUV	TIMi channel x TDU control register	192
TIM[i]_CH[x]_TDUC	TIMi channel x TDU counter register	193
TIM[i]_CH[x]_GPR0	TIMi channel x general purpose 0 register	183
TIM[i]_CH[x]_GPR1	TIMi channel x general purpose 1 register	184
TIM[i]_CH[x]_CNT	TIMi channel x SMU counter register	185
TIM[i]_CH[x]_ECNT	TIMi channel x SMU edge counter register	194
TIM[i]_CH[x]_CNTS	TIMi channel x SMU shadow counter register	185
TIM[i]_CH[x]_IRQ_NOTIFY	TIMi channel x interrupt notification register	186
TIM[i]_CH[x]_IRQ_EN	TIMi channel x interrupt enable register	187
TIM[i]_CH[x]_EIRQ_EN	TIMi channel x error interrupt enable register	191
TIM[i]_CH[x]_IRQ_FORCINT	TIMi channel x force interrupt register	188
TIM[i]_CH[x]_IRQ_MODE	TIMi interrupt mode configuration register	189
TIM[i]_RST	TIMi global software reset register	188
TIM[i]_IN_SRC	TIMi AUX IN source selection register	190
TIM[i]_INP_VAL	TIMi input value observation register	198

28.13.8 TIM Configuration Registers Description

28.13.8.1 Register TIM[i]_CH[x]_CTRL

TIMi Channel x Control Register

Table 44 Filter Modes for Rising Edge

FLT_MODE_RE		EFLT_CTR_RE[1)]	FLT_CTR_RE	Coding
0	Immediate edge propagation mode for rising edge	0	0	Immediate edge propagation mode
		0	1	Immediate edge propagation mode
		1	0	Reserved
		1	1	Reserved
1	Individual de-glitch mode for rising edge	0	0	Up-Down Counter individual de-glitch mode
		0	1	Hold Counter individual de-glitch mode
		1	0	Reset Counter individual de-glitch mode
		1	1	Reserved

1) Bit is located in register TIM[i]_CH[x]_ECTRL.

Table 45 Filter Modes for Falling Edge

FLT_MODE_FE		EFLT_CTR_FE[1)]	FLT_CTR_FE	Coding
0	Immediate edge propagation mode for rising edge	0	0	Immediate edge propagation mode
		0	1	Immediate edge propagation mode
		1	0	Reserved
		1	1	Reserved
1	Individual de-glitch mode for rising edge	0	0	Up-Down Counter individual de-glitch mode
		0	1	Hold Counter individual de-glitch mode
		1	0	Reset Counter individual de-glitch mode
		1	1	Reserved

1) Bit is located in register TIM[i]_CH[x]_ECTRL.

TIMi_CHx_CTRL (i=0-7;x=0-7)

TIMi Channel x Control Register **($001024_H+i*800_H+x*80_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOCTRL		EGPR1_SEL	EGPR0_SEL	FR_ECNT_OFL		CLK_SEL		FLT_CTR_FE	FLT_MODE_FE	FLT_CTR_RE	FLT_MODE_RE	EXT_CAP_EN	FLT_CNT_FRQ		FLT_EN
rw		rw	rw	rw		rw		rw	rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECNT_RESET	ISL	DSL	CNTS_SEL	GPR1_SEL		GPR0_SEL		TBU0_SEL	CICTRL	ARU_EN	OSM	TIM_MODE			TIM_EN
rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw			rw

Field	Bits	Type	Description
TIM_EN	0	rw	**TIM channel x enable** Enabling of the channel resets the registers **ECNT, TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_GPR0,** and **TIM[i]_CH[x]_GPR1** to their reset values. After finishing the action in one-shot mode the **TIM_EN** bit is cleared automatically. Otherwise, the bit must be cleared manually. 0_B Channel disabled 1_B Channel enabled
TIM_MODE	3:1	rw	**TIM channel x mode** If an undefined value is written to the TIM_MODE register, the hardware switches automatically to TIM_MODE = 0b000 (TPWM mode). The TIM_MODE register should not be changed while the TIM channel is enabled. If the TIM channel is enabled and operating in TPWM or TPIM mode after the first valid edge defined by DSL has occurred, a reconfiguration of DSL,ISL,TIM_MODE will not change the channel behavior. Reading these bit fields after reconfiguration will show the newly configured settings but the initial channel behavior will not change. Only a disabling of the TIM channel by setting TIM_EN= 0 and reenabling with TIM_EN= 1 will change the channel operation mode. 000_B PWM Measurement Mode (TPWM) 001_B Pulse Integration Mode (TPIM) 010_B Input Event Mode (TIEM) 011_B Input Prescaler Mode (TIPM) 100_B Bit Compression Mode (TBCM) 101_B Gated Periodic Sampling Mode (TGPS) 110_B Serial Shift Mode (TSSM)
OSM	4	rw	**One-shot mode** After finishing the action in one-shot mode the **TIM_EN** bit is cleared automatically. 0_B Continuous operation mode 1_B One-shot mode
ARU_EN	5	rw	**GPR0 and GPR1 register values routed to ARU** 0_B Registers content not routed 1_B Registers content routed
CICTRL	6	rw	**Channel Input Control** 0_B Use signal TIM_IN(x) as input for channel x 1_B Use signal TIM_IN(x-1) as input for channel x (or TIM_IN(m-1) if x is 0)
TBU0_SEL	7	rw	**TBU_TS0 bits input select for TIM0_CH[x]_GPRz (z: 0, 1)** This bit is only applicable for TIM0. 0_B Use TBU_TS0(23..0) to store in TIM0_CH[x]_GPR0/TIM0_CH[x]_GPR1 1_B Use TBU_TS0(26..3) to store in TIM0_CH[x]_GPR0/TIM0_CH[x]_GPR1

Field	Bits	Type	Description
GPR0_SEL	9:8	rw	**Selection for GPR0 register** If EGPR0_SEL =0 / EGPR0_SEL =1 : If a reserved value is written to the EGPR0_SEL, GPR0_SEL bit fields, the hardware will use TBU_TS0 input. 00_B Use TBU_TS0 as input / use ECNT as input 01_B Use TBU_TS1 as input / use TIM_INP_VAL as input 10_B Use TBU_TS2 as input / reserved 11_B Use CNTS as input; if TGPS mode in channel = 0 is selected, use TIM Filter F_OUT as input / reserved
GPR1_SEL	11:10	rw	**Selection for GPR1 register** If EGPR1_SEL =0 / EGPR1_SEL =1: If a reserved value is written to the EGPR1_SEL, GPR1_SEL bit fields, the hardware will use TBU_TS0 input. Note: In TBCM mode: EGPR1_SEL=1, GPR1_SEL=01 selects TIM_INP_VAL as input; in all other cases, TIM Filter F_OUT is used. 00_B Use TBU_TS0 as input / use ECNT as input 01_B Use TBU_TS1 as input / use TIM_INP_VAL as input 10_B Use TBU_TS2 as input / reserved 11_B Use CNT as input / reserved
CNTS_SEL	12	rw	**Selection for CNTS register** The functionality of the **CNTS_SEL** is disabled in the modes TIPM,TGPS and TBCM. **CNTS_SEL** in TSSM mode selects the source signal for registered or latched shift out operation. 0_B use F_OUTx 1_B use TIM_INx 0_B Use CNT register as input 1_B Use TBU_TS0 as input
DSL	13	rw	**Signal level control** In TIM_MODE=0b110 (TSSM), the bit field DSL defines the shift direction. 0_B Shift left 1_B Shift right 0_B Measurement starts with falling edge (low level measurement) 1_B Measurement starts with rising edge (high level measurement)
ISL	14	rw	**Ignore signal level** This bit is mode dependent and will have different meanings (see details in the TIM Channel mode description). 0_B Use DSL bit for selecting active signal level (TIEM) 1_B Ignore DSL and treat both edges as active edge (TIEM)

Field	Bits	Type	Description
ECNT_RESET	15	rw	**Enables resetting of counter in certain modes** If TIM_MODE=0b101 (TGPS) / TIM_MODE=0b000 (TPWM) else ECNT counter operating in wrap around mode; In TIM_MODE=0b110 (TSSM), the bit field ECNT_RESET defines the initial polarity for the shift register. 0_B ECNT counter operating in wrap around mode / ECNT counter operating in wrap around mode, CNT is reset on active input edge defined by DSL 1_B ECNT counter is reset with periodic sampling / ECNT counter operating in wrap around mode, CNT is reset on active and inactive input edge
FLT_EN	16	rw	**Filter enable for channel x** If the filter is disabled, all filter related units (including CSU) are bypassed, which means that the signal *F_IN* is directly routed to signal *F_OUT*. 0_B Filter disabled and internal states are reset 1_B Filter enabled
FLT_CNT_FRQ	18:17	rw	**Filter counter frequency select** 00_B FLT_CNT counts with CMU_CLK0 01_B FLT_CNT counts with CMU_CLK1 10_B FLT_CNT counts with CMU_CLK6 11_B FLT_CNT counts with CMU_CLK7
EXT_CAP_EN	19	rw	**Enables external capture mode** The selected TIM mode is only sensitive to external capture pulses the input event changes are ignored. 0_B External capture disabled 1_B External capture enabled
FLT_MODE_RE	20	rw	**Filter mode for rising edge** Coding see **Table 44**.
FLT_CTR_RE	21	rw	**Filter counter mode for rising edge** Coding see **Table 44**.
FLT_MODE_FE	22	rw	**Filter mode for falling edge** Coding see **Table 45**.
FLT_CTR_FE	23	rw	**Filter counter mode for falling edge** Coding see **Table 45**.
CLK_SEL	26:24	rw	**CMU clock source select for channel** If ECLK_SEL =0 / ECLK_SEL =1: 000_B CMU_CLK0 selected / tdu_sample_evt of TDU selected 001_B CMU_CLK1 selected / reserved 010_B CMU_CLK2 selected / reserved 011_B CMU_CLK3 selected / reserved 100_B CMU_CLK4 selected / reserved 101_B CMU_CLK5 selected / reserved 110_B CMU_CLK6 selected / reserved 111_B CMU_CLK7 selected / reserved

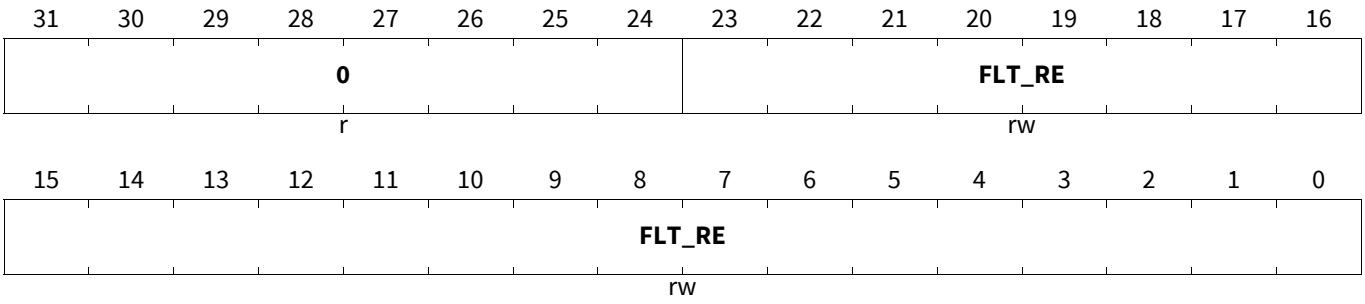
Field	Bits	Type	Description
FR_ECNT_OFL	27	rw	**Extended Edge counter overflow behavior** 0_B Overflow will be signaled on ECNT bit width = 8 1_B Overflow will be signaled on EECNT bit width (full range)
EGPR0_SEL	28	rw	**Extension of GPR0_SEL bit field** Details described in GPR0_SEL bit field.
EGPR1_SEL	29	rw	**Extension of GPR1_SEL bit field** Details described in GPR1_SEL bit field.
TOCTRL	31:30	rw	**Timeout control** It has to be mentioned that writing of TOCTRL= 0 will every time stop the TDU, independent of the previous state of TOCTRL. 00_B Timeout feature disabled 01_B Timeout feature enabled for rising edge only 10_B Timeout feature enabled for falling edge only 11_B Timeout feature enabled for both edges

28.13.8.2 Register TIM[i]_CH[x]_FLT_RE

TIMi Channel x Filter Parameter 0 Register

TIMi_CHx_FLT_RE (i=0-7;x=0-7)

TIMi Channel x Filter Parameter 0 Register($00101C_H$+i*800_H+x*80_H) Application Reset Value: 0000 0000_H



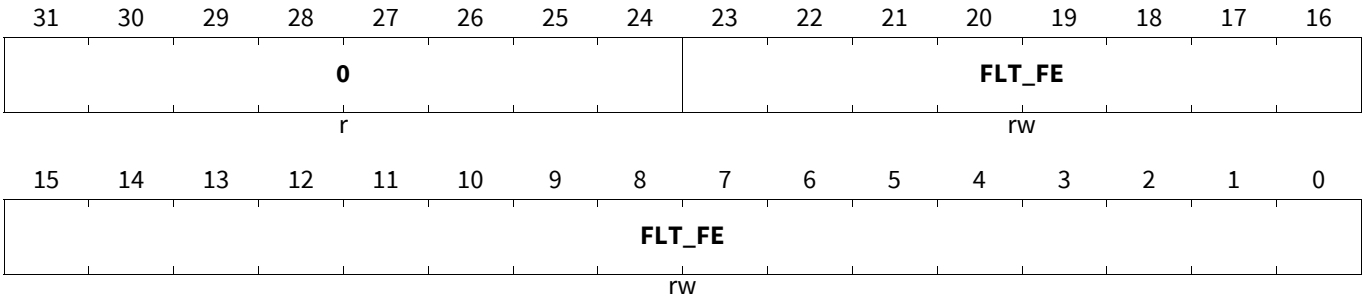
Field	Bits	Type	Description
FLT_RE	23:0	rw	**Filter parameter for rising edge** FLT_RE has different meanings in the various filter modes.Immediate edge propagation mode = acceptance time for rising edgeIndividual deglitch time mode = deglitch time for rising edge.
0	31:24	r	**Reserved** Read as zero, shall be written as zero

28.13.8.3 Register TIM[i]_CH[x]_FLT_FE

TIMi Channel x Filter Parameter 1 Register

TIMi_CHx_FLT_FE (i=0-7;x=0-7)

TIMi Channel x Filter Parameter 1 Register(001020_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H



Field	Bits	Type	Description
FLT_FE	23:0	rw	**Filter parameter for falling edge** FLT_FE has different meanings in the various filter modes.Immediate edge propagation mode = acceptance time for falling edge Individual deglitch time mode = deglitch time for falling edge.
0	31:24	r	**Reserved** Read as zero, shall be written as zero

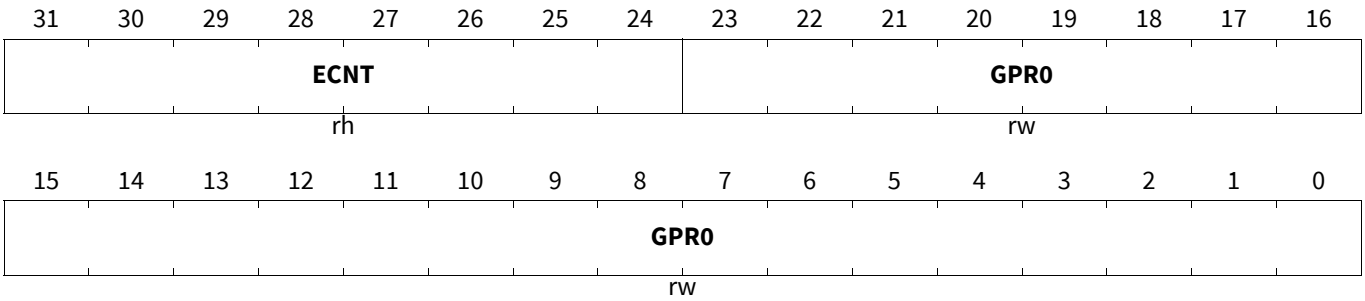
28.13.8.4 Register TIM[i]_CH[x]_GPR0

TIMi Channel x General Purpose 0 Register

Note: The ECNT register is reset to its initial value when the channel is enabled. Please note, that bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.

TIMi_CHx_GPR0 (i=0-7;x=0-7)

TIMi Channel x General Purpose 0 Register(001000_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H



Field	Bits	Type	Description
GPR0	23:0	rw	**Input signal characteristic parameter 0** The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit fields **EGPR0_SEL**, **GPR0_SEL** of register **TIM[i]_CH[x]_CTRL**. **Note:** The content of this register can only be written in TIM channel mode TSSM.
ECNT	31:24	rh	**Edge counter** The **ECNT** counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of **ECNT**.

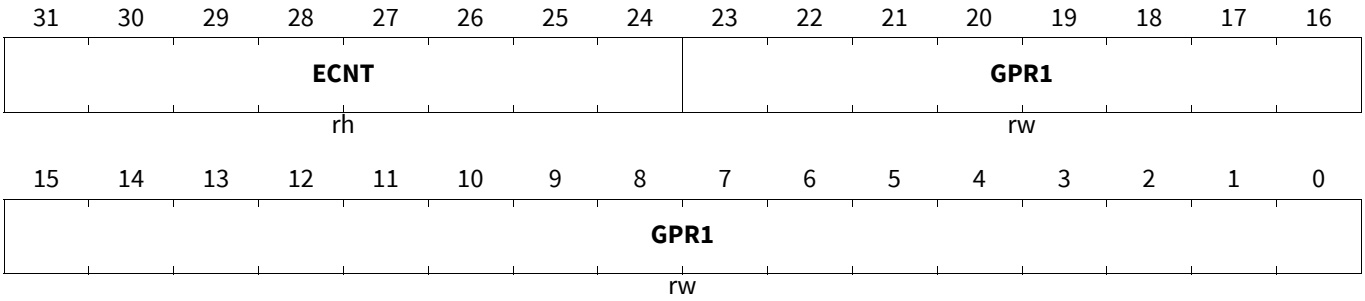
28.13.8.5 Register TIM[i]_CH[x]_GPR1

TIMi Channel x General Purpose 1 Register

Note: The ECNT register is reset to its initial value when the channel is enabled. Please note, that bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.

TIMi_CHx_GPR1 (i=0-7;x=0-7)

TIMi Channel x General Purpose 1 Register(001004_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H

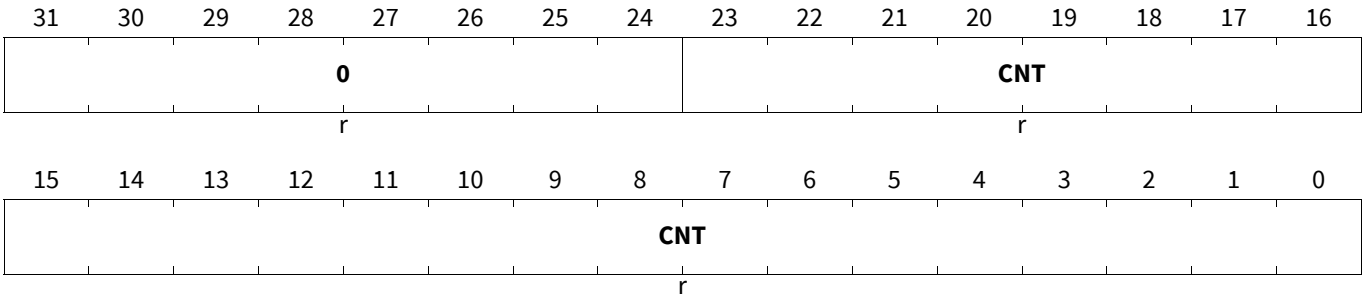


Field	Bits	Type	Description
GPR1	23:0	rw	**Input signal characteristic parameter 1** The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit fields **EGPR1_SEL**, **GPR1_SEL** of register **TIM[i]_CH[x]_CTRL**. In TBCM mode if EGPR1_SEL=1, GPR1_SEL=01 then TIM_INP_VAL is used as input in all other cases TIM Filter F_OUT is used as input and Bits GPR1(23:8) = 0 The content of this register can only be written in TIM channel mode TGPS and TSSM.
ECNT	31:24	rh	**Edge counter** The **ECNT** counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of **ECNT**.

28.13.8.6 Register TIM[i]_CH[x]_CNT

TIMi Channel x SMU Counter Register

TIMi_CHx_CNT (i=0-7;x=0-7)
TIMi Channel x SMU Counter Register (001008_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H



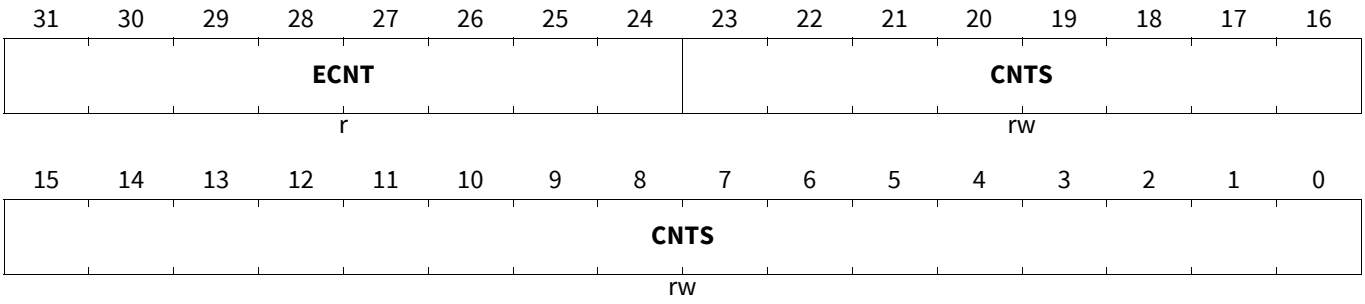
Field	Bits	Type	Description
CNT	23:0	r	**Actual SMU counter value** The meaning of this value depends on the configured mode: TPWM = actual duration of PWM signal.TPIM = actual duration of all pulses (sum of pulses).TIEM = actual number of received edges.TIPM = actual number of received edges.TGPS = elapsed time for periodic sampling.TSSM = shift data.
0	31:24	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.7 Register TIM[i]_CH[x]_CNTS

TIMi Channel x SMU Shadow Counter Register

Note: The ECNT register is reset to its initial value when the channel is enabled. Please note, that bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.

TIMi_CHx_CNTS (i=0-7;x=0-7)
TIMi Channel x SMU Shadow Counter Register(001010_H+i*800_H+x*80_H)Application Reset Value: 0000 0000_H



Field	Bits	Type	Description
CNTS	23:0	rw	**Counter shadow register** The content of this register has different meaning for the TIM channels modes. The content depends directly on the bit field **CNTS_SEL** of register **TIM[i]_CH[x]_CTRL**. The register **TIM[i]_CH[x]_CNTS** is only writable in TIPM,TBCM ,TGPS and TSSM mode.
ECNT	31:24	r	**Edge counter** The **ECNT** counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of **ECNT**.

28.13.8.8 Register TIM[i]_CH[x]_IRQ_NOTIFY

TIMi Channel x Interrupt Notification Register

TIMi_CHx_IRQ_NOTIFY (i=0-7;x=0-7)
TIMi Channel x Interrupt Notification Register($00102C_H$+i*800_H+x*80_H)Application Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										GLITCHDET	TODET	GPROFL	CNTOFL	ECNTOFL	NEWVAL
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
NEWVAL	0	rw	**New measurement value detected by in channel x** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0_B No event has occurred 1_B NEWVAL has occurred on the TIM channel
ECNTOFL	1	rw	**counter overflow of channel x** See bit 0.
CNTOFL	2	rw	**SMU CNT counter overflow of channel x** See bit 0.
GPROFL	3	rw	**GPR0 and GPR1 data overflow** Old data not read out before new data has arrived at input pin. See bit 0.
TODET	4	rw	**Timeout reached for input signal of channel x** See bit 0.

Field	Bits	Type	Description
GLITCHDET	5	rw	**Glitch detected on channel x, (x:0...m-1)** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0_B No glitch detected for last edge 1_B Glitch detected for last edge
0	31:6	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.9 Register TIM[i]_CH[x]_IRQ_EN

TIMi Channel x Interrupt Enable Register

TIMi_CHx_IRQ_EN (i=0-7;x=0-7)

TIMi Channel x Interrupt Enable Register(001030_H+i*800_H+x*80_H) **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										GLITCHDET_IRQ_EN	TODET_IRQ_EN	GPROFL_IRQ_EN	CNTOFL_IRQ_EN	ECNTOFL_IRQ_EN	NEWVAL_IRQ_EN
r										rw	rw	rw	rw	rw	rw

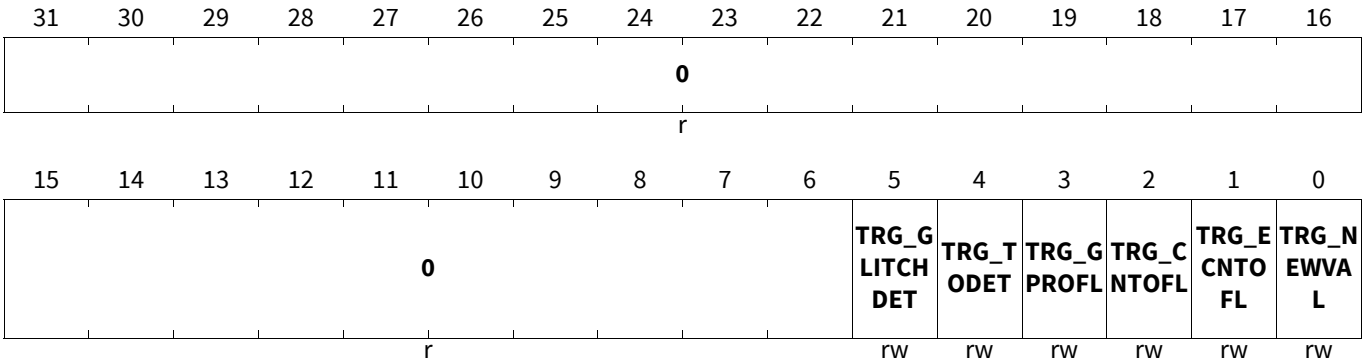
Field	Bits	Type	Description
NEWVAL_IRQ_EN	0	rw	**TIM_NEWVALx_IRQ interrupt enable** 0_B Disable interrupt, interrupt is not visible outside GTM 1_B Enable interrupt, interrupt is visible outside GTM
ECNTOFL_IRQ_EN	1	rw	**TIM_ECNTOFLx_IRQ interrupt enable** Coding see bit 0.
CNTOFL_IRQ_EN	2	rw	**TIM_CNTOFLx_IRQ interrupt enable** Coding see bit 0.
GPROFL_IRQ_EN	3	rw	**TIM_GPROFL_IRQ interrupt enable** Coding see bit 0.
TODET_IRQ_EN	4	rw	**TIM_TODETx_IRQ interrupt enable** Coding see bit 0.
GLITCHDET_IRQ_EN	5	rw	**TIM_GLITCHDETx_IRQ interrupt enable** Coding see bit 0.
0	31:6	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.10Register TIM[i]_CH[x]_IRQ_FORCINT

TIMi Channel x Force Interrupt Register

TIMi_CHx_IRQ_FORCINT (i=0-7;x=0-7)

TIMi Channel x Force Interrupt Register(001034_H+i*800_H+x*80_H) **Application Reset Value: 0000 0000_H**

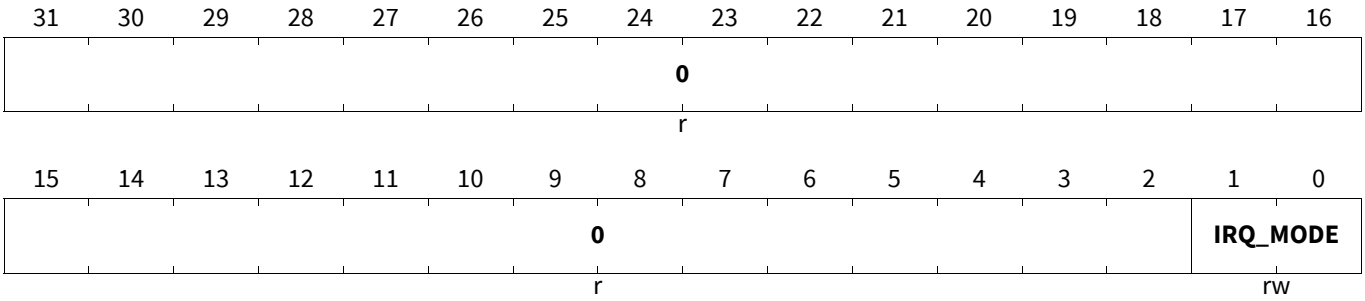


Field	Bits	Type	Description
TRG_NEWVAL	0	rw	**Trigger NEWVAL bit in TIM_CHx_IRQ_NOTIFY register by software** This bit is cleared automatically after write. This bit is write protected by bit RF_PROT of register GTM_CTRL. 0_B No interrupt triggering 1_B Assert corresponding field in TIM[i]_CH[x]_IRQ_NOTIFY register
TRG_ECNTOFL	1	rw	**Trigger ECNTOFL bit in TIM_CHx_IRQ_NOTIFY register by software** Coding see bit 0.
TRG_CNTOFL	2	rw	**Trigger CNTOFL bit in TIM_CHx_IRQ_NOTIFY register by software** Coding see bit 0.
TRG_GPROFL	3	rw	**Trigger GPROFL bit in TIM_CHx_IRQ_NOTIFY register by software** Coding see bit 0.
TRG_TODET	4	rw	**Trigger TODET bit in TIM_CHx_IRQ_NOTIFY register by software** Coding see bit 0.
TRG_GLITCHDET	5	rw	**Trigger GLITCHDET bit in TIM_CHx_IRQ_NOTIFY register by software** Coding see bit 0.
0	31:6	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.11Register TIM[i]_CH[x]_IRQ_MODE

TIMi Channel x Interrupt Mode Configuration Register

TIMi_CHx_IRQ_MODE (i=0-7;x=0-7)
TIMi Channel x Interrupt Mode Configuration Register(001038_H+i*800_H+x*80_H) Application Reset Value: $0000\ 0000_H$

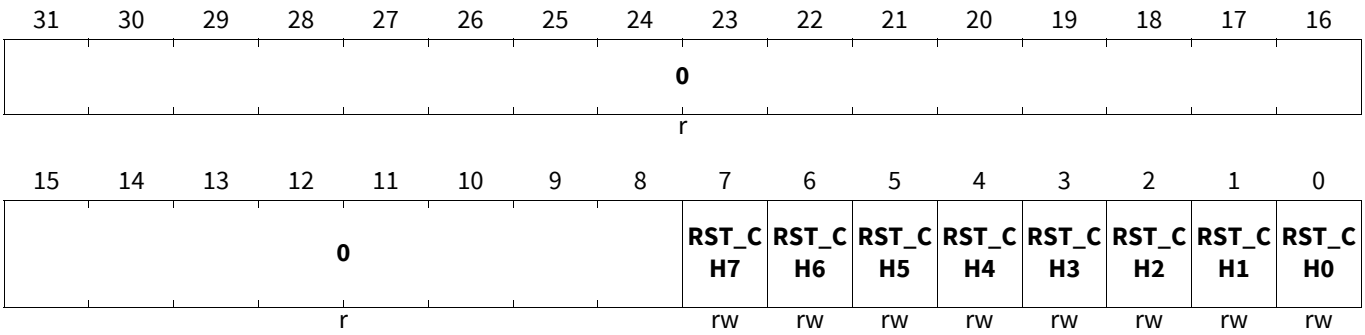


Field	Bits	Type	Description
IRQ_MODE	1:0	rw	**IRQ mode selection** The interrupt modes are described in **Section 28.4.5**. 00_B Level mode 01_B Pulse mode 10_B Pulse-Notify mode 11_B Single-Pulse mode
0	31:2	r	**Reserved** Read as zero, shall be written as zero

28.13.8.12Register TIM[i]_RST

TIMi Global Software Reset Register

TIMi_RST (i=0-7)
TIMi Global Software Reset Register ($00107C_H$+i*800_H) Application Reset Value: $0000\ 0000_H$



Field	Bits	Type	Description
RST_CHx (x=0-7)	x	rw	**Software reset of channel x** This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. Please note, that the RST field width of this register depends on the number of implemented channels m within this sub-module. This register description represents a register layout for m = 8. 0_B No action 1_B Reset channel x
0	31:8	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.13Register TIM[i]_IN_SRC

TIMi AUX IN Source Selection Register

TIMi_IN_SRC (i=0-7)
TIMi AUX IN Source Selection Register **(001078_H+i*800_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE_7		**VAL_7**		**MODE_6**		**VAL_6**		**MODE_5**		**VAL_5**		**MODE_4**		**VAL_4**	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE_3		**VAL_3**		**MODE_2**		**VAL_2**		**MODE_1**		**VAL_1**		**MODE_0**		**VAL_0**	
rw		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
VAL_x (x=0-7)	4*x+1:4*x	rw	**Value to be fed to Channel x** Multicore encoding in use (**VAL_x(1)** defines the state of the signal). Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description. Any read access to a **VAL_x** bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00_B State is 0 (ignore write access) 01_B Change state to 0 10_B Change state to 1 11_B State is 1 (ignore write access)

Field	Bits	Type	Description
MODE_x (x=0-7)	4*x+3:4*x+2	rw	**Input source to Channel x** Multicore encoding in use (**MODE_x(1)** defines the state of the signal). Function table: **MODE_x(1)=0** , **VAL_x(1)**=0: The input signal defined by bit field **CICTRL** of the TIM channel is used as input source. **MODE_x(1)=0** , **VAL_x(1)**=1: The signal TIM_AUX_IN of the TIM channel is used as input source. **MODE_x(1)=1** : The state **VAL_x(1)** defines the input level for the TIM channel. Any read access to a **MODE_x** bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00_B State is 0 (ignore write access) 01_B Change state to 0 10_B Change state to 1 11_B State is 1 (ignore write access)

28.13.8.14Register TIM[i]_CH[x]_EIRQ_EN

TIMi Channel x Error Interrupt Enable Register

TIMi_CHx_EIRQ_EN (i=0-7;x=0-7)

TIMi Channel x Error Interrupt Enable Register($00103C_H$+i*800_H+x*80_H) **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										GLITCHDET_EIRQ_EN	TODET_EIRQ_EN	GPROFL_EIRQ_EN	CNTOFL_EIRQ_EN	ECNTOFL_EIRQ_EN	NEWVAL_EIRQ_EN
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
NEWVAL_EIRQ_EN	0	rw	**TIM_NEWVALx_EIRQ error interrupt enable** 0_B Disable error interrupt, error interrupt is not visible outside GTM 1_B Enable error interrupt, error interrupt is visible outside GTM
ECNTOFL_EIRQ_EN	1	rw	**TIM_ECNTOFLx_IRQ interrupt enable** Coding see bit 0.
CNTOFL_EIRQ_EN	2	rw	**TIM_CNTOFLx_IRQ interrupt enable** Coding see bit 0.
GPROFL_EIRQ_EN	3	rw	**TIM_GPROFL_IRQ interrupt enable** Coding see bit 0.

Field	Bits	Type	Description
TODET_EIRQ_EN	4	rw	**TIM_TODETx_IRQ interrupt enable** Coding see bit 0.
GLITCHDET_EIRQ_EN	5	rw	**TIM_GLITCHDETx_IRQ interrupt enable** Coding see bit 0.
0	31:6	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.15Register TIM[i]_CH[x]_TDUV

TIMi Channel x TDU Control Register

TIMi_CHx_TDUV (i=0-7;x=0-7)
TIMi Channel x TDU Control Register (001018$_H$+i*800$_H$+x*80$_H$) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		TCS		TDU_SAME_CNT_CLK	TCS_USE_SAMPLE_EVT	SLICING		TOV2							
r		rw		rw	rw	rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOV1								TOV							
rw								rw							

Field	Bits	Type	Description
TOV	7:0	rw	**Time out compare value slice0 for channel x** Compare value for TO_CNT.
TOV1	15:8	rw	**Time out compare value slice1 for channel x** Compare value for TO_CNT1.
TOV2	23:16	rw	**Time out compare value slice2 for channel x** SLICING!=0b11: Compare value for TO_CNT2. SLICING= 0b11: TOV2 operate as a shadow register for TO_CNT.
SLICING	25:24	rw	**Cascading of counter slices** If USE_LUT=0b00 / USE_LUT !=0b00 00_B Combine slice2, slice1, slice0 to 1x24-bit counter / reserved 01_B Combine slice1, slice0 to 1x16-bit counter, use slice2 as 1x8-bit counter / combine slice1, slice0 to 1x16-bit; slice2 not usable 10_B Use slice2, slice1, slice0 as 3x8-bit counter / use slice1, slice0 as 2x8-bit counter; slice2 not usable 11_B Use slice1, slice0 as 2x8-bit counter / use slice1, slice0 as 2x8-bit counter
TCS_USE_SAMPLE_EVT	26	rw	**Use tdu_sample_evt as Timeout Clock** 0_B CMU_CLK selected by TCS is in use by TO_CNT, TO_CNT2 1_B CMU_CLK selected by TCS is in use by TO_CNT2; tdu_sample_evt is in use by TO_CNT

Field	Bits	Type	Description
TDU_SAME_CNT_CLK	27	rw	**Define clocking of TO_CNT, TO_CNT1** 0_B TO_CNT clock selected by (TCS, TCS_USE_SAMPLE_EVT); TO_CNT1 clocked on tdu_word_event 1_B TO_CNT1 uses same clock as TO_CNT
TCS	30:28	rw	**Timeout Clock selection** 000_B CMU_CLK0 selected 001_B CMU_CLK1 selected 010_B CMU_CLK2 selected 011_B CMU_CLK3 selected 100_B CMU_CLK4 selected 101_B CMU_CLK5 selected 110_B CMU_CLK6 selected 111_B CMU_CLK7 selected
0	31	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.16Register TIM[i]_CH[x]_TDUC

TIMi Channel x TDU Counter Register

The register **TIM[i]_CH[x]_TDUC** is writable if Timeout unit is disabled (TOCTRL=0b00).

If USE_LUT != 0b00 (input signal generation by lookup table) the bit field TO_CNT2 is writable at any time, TO_CNT,TO_CNT1 will not be changed.

TIMi_CHx_TDUC (i=0-7;x=0-7)

TIMi Channel x TDU Counter Register (001014_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								TO_CNT2							
r								rw							

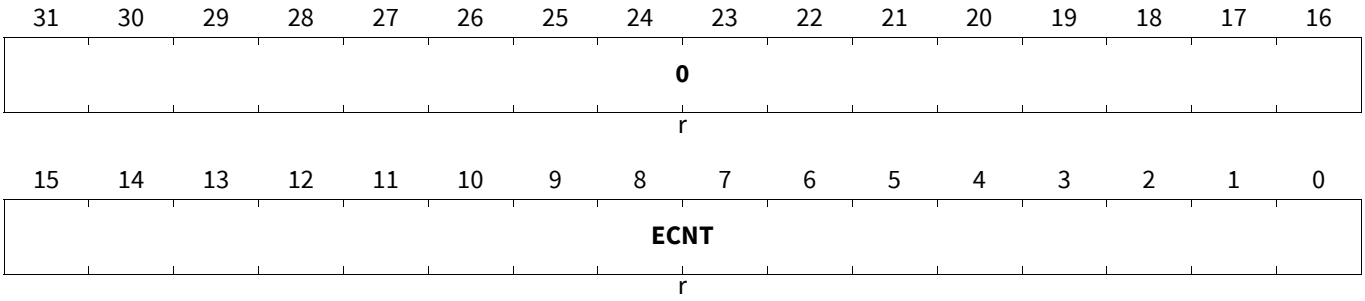
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO_CNT1								TO_CNT							
rw								rw							

Field	Bits	Type	Description
TO_CNT	7:0	rw	**Current Timeout value slice0 for channel x** SLICING != 0b11: counter will be reset to 0X0 on TDU_RESYNC condition. SLICING = 0b11 : counter will be loaded with TOV2 on TDU_RESYNC condition.
TO_CNT1	15:8	rw	**Current Timeout value slice1 for channel x** Counter will be reset to 0x0 on TDU_RESYNC condition.
TO_CNT2	23:16	rw	**Current Timeout value slice2 for channel x** Counter will be reset to 0x0 on TDU_RESYNC condition.
0	31:24	r	**Reserved** Read as zero, shall be written as zero

28.13.8.17Register TIM[i]_CH[x]_ECNT

TIMi Channel x SMU Edge Counter Register

TIMi_CHx_ECNT (i=0-7;x=0-7)
TIMi Channel x SMU Edge Counter Register($00100C_H$+i*800_H+x*80_H) Application Reset Value: 0000 0000_H

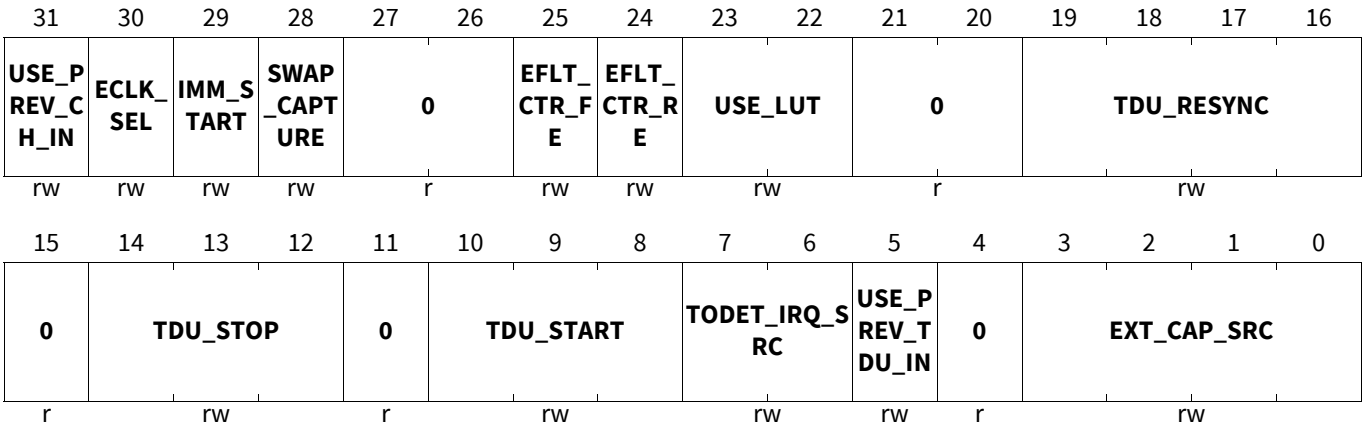


Field	Bits	Type	Description
ECNT	15:0	r	**Edge counter** If TIM channel is disabled the content of ECNT gets frozen. A read will auto clear the bits [15:1]. Further read accesses to ECNT will show on Bit 0 the actual input signal value of the channel.
0	31:16	r	**Reserved** Read as zero, shall be written as zero.

28.13.8.18Register TIM[i]_CH[x]_ECTRL

TIMi Channel x Extended Control Register

TIMi_CHx_ECTRL (i=0-7;x=0-7)
TIMi Channel x Extended Control Register(001028_H+i*800_H+x*80_H) Application Reset Value: 0000 0000_H



Field	Bits	Type	Description
EXT_CAP_SRC	3:0	rw	**Defines selected source for triggering the EXT_CAPTURE functionality** 0_H NEW_VAL_IRQ of following channel selected 1_H AUX_IN selected 2_H CNTOFL_IRQ of following channel selected 3_H CICTRL = 1: use signal TIM_IN(x) as input for channel x / CICTRL = 0: use signal TIM_IN(x-1) as input for channel x (or TIM_IN(m-1) if x is 0) 4_H ECNTOFL_IRQ of following channel selected 5_H TODET_IRQ of following channel selected 6_H GLITCHDET_IRQ of following channel selected 7_H GPROFL_IRQ of following channel selected 8_H cmu_clk selected by CLK_SEL of following channel 9_H REDGE_DET of following channel selected A_H FEDGE_DET of following channel selected B_H Logical OR of (FEDGE_DET, REDGE_DET) of following channel selected C_H tdu_sample_evt of local TDU selected D_H tdu_word_evt of local TDU selected E_H tdu_frame_evt of local TDU selected F_H Reserved
USE_PREV_TDU_IN	5	rw	**Select input data source for TDU** 0_B Use input data of local filter for TDU 1_B Use input data of previous channel (after filter unit) for TDU
TODET_IRQ_SRC	7:6	rw	**selection of source for TODET_IRQ** With TODET_IRQ_SRC=0b00 the ACB bit 2 will be driven by signal tdu_timeout_evt, if TODET_IRQ_SRC!=0b00 ACB2 will be 0. 00_B Use tdu_timeout_evt 01_B Use tdu_word_evt 10_B Use tdu_frame_evt 11_B Use tdu_sample_evt

Field	Bits	Type	Description
TDU_START	10:8	rw	**Defines condition which will start the TDU unit** In mode SLICING=0b11 every start/restart will load the TO_CNT with value TOV2. Note: tdu_start_000_event is defined as: Each writing of TOCTRL != 0 (independent of current TOCTRL) while TDU_START=0b000 and TDU is stopped (initially or stopped by TDU_STOP event). This event will last 1 system clock cycle. 000_B Start once immediate on tdu_start_000_event 001_B Start once with occurrence of first cmu_clk selected by CLK_SEL when measure unit is enabled by TIM_EN=1 010_B Start once with occurrence of first active edge selected by TOCTRL; restart on tdu_frame_evt if TDU is stopped 011_B Start once with occurrence of first active edge selected by TOCTRL 100_B Start/restart with occurrence of external capture event (if TDU is stopped, restart again) 101_B Start/restart with occurrence of first cmu_clk selected by CLK_SEL when measure unit is enabled by TIM_EN=1 (if TDU is stopped, restart again) 110_B Start once with occurrence of external capture event; restart on tdu_frame_evt if TDU is stopped 111_B Start/restart with occurrence of first active edge selected by TOCTRL (if TDU is stopped, restart again)
TDU_STOP	14:12	rw	**Defines condition which will stop the TDU unit** Note: tdu_toctrl_0_event is defined as: Each writing of TOCTRL = 0 (independent of current TOCTRL) while TDU is started. This event will last 1 system clock cycle. 000_B Immediate stop counting of TDU on tdu_toctrl_0_event (see Note) 001_B Stop counting of TDU on tdu_word_evt or on tdu_toctrl_0_event (see Note) 010_B Stop counting of TDU on tdu_frame_evt or on tdu_toctrl_0_event (see Note) 011_B Stop counting of TDU on tdu_timeout_evt or on tdu_toctrl_0_event (see Note) 100_B Stop counting of TDU on external capture event or on tdu_toctrl_0_event (see Note)
TDU_RESYNC	19:16	rw	**Defines condition which will resynchronize the TDU unit** Encoding see **Table 46** and **Table 47**.
USE_LUT	23:22	rw	**Generate Filter input by lookup table** 00_B Lookup table not in use, lut_in0(x) used as filter input 01_B Use 3-bit lookup table with index = ext_capture(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index]. 10_B Use 3-bit lookup table with index = fout_prev(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index]. 11_B Use 3-bit lookup table with index = tssm_out(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index].
EFLT_CTR_RE	24	rw	**Extension of bit field FLT_CTR_RE** Details described in FLT_CTR_RE bit field of register TIM[i]_CH[x]_CTRL.

Field	Bits	Type	Description
EFLT_CTR_FE	25	rw	**Extension of bit field FLT_CTR_FE** Details described in FLT_CTR_FE bit field of register TIM[i]_CH[x]_CTRL.
SWAP_CAPTURE	28	rw	**Swap point of time of capturing CNTS and GPR1** This bit is only applicable in TPWM and TPIM mode. Set to 0 in all other modes. 0_B Inactive edge will capture data in CNTS; NEWVAL_IRQ event will capture data in GPR1 1_B Swap time of capture: inactive edge will capture data in GPR1; NEWVAL_IRQ event will capture data in CNTS
IMM_START	29	rw	**Start immediately the measurement** This bit is only applicable in TPWM and TPIM mode. Set to 0 in all other modes. 0_B Start with first active edge the measurement 1_B Start immediately after enable (TIM_EN=1) the measurement
ECLK_SEL	30	rw	**Extension of bit field CLK_SEL** Details described in CLK_SEL bit field of register TIM[i]_CH[x]_CTRL.
USE_PREV_CH_IN	31	rw	**Select input data source for TIM channel** 0_B Use input data of local filter unit for channel measurements 1_B Use input data of previous channel (after filter unit) for channel measurements
0	4, 11, 15, 21:20, 27:26	r	**Reserved** Read as zero, shall be written as zero.

Table 46 Behavior of TDU_RESYNC with SLICING != 0b11

TDU_RESYNC	Behavior
0000_B	reset counter TO_CNT2 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event [1]; reset counters TO_CNT,TO_CNT1 on tdu_timeout_evt or on tdu_start_000_event [1]; if SLICING=0b10 and TO_CTRL=0b-1 then reset TO_CNT on rising input edge; if SLICING=0b10 and TO_CTRL=0b1- then reset TO_CNT1 on falling input edge; if SLICING!=0b10 then reset counters TO_CNT,TO_CNT1 on each active edge selected by TOCTRL
$0\text{--}1_B$	if SLICING=0b10 and TO_CTRL=0bx1 then reset TO_CNT on rising input edge; if SLICING=0b10 and TO_CTRL=0b1- then reset TO_CNT1 on falling input edge; if SLICING!=0b10 then reset counters TO_CNT,TO_CNT1 on each active edge selected by TOCTRL; if SLICING=0b00 then reset TO_CNT2 on each active edge selected by TOCTRL
$0x1\text{-}_B$	reset counters TO_CNT on tdu_word_evt;
01--_B	reset counter TO_CNT1 on tdu_frame_evt; if SLICING=0b01 then reset TO_CNT on tdu_frame_evt
1000_B	reset counters TO_CNT,TO_CNT1,TO_CNT2 on event selected by EXT_CAP_SRC
1---_B	if SLICING!=0b00 then reset counter TO_CNT2 on tdu_sample_evt

Table 46 Behavior of TDU_RESYNC with SLICING != 0b11 (cont'd)

TDU_RESYNC	Behavior
1--1 $_B$	reset counter TO_CNT2 on each active edge selected by TOCTRL; if SLICING=0b10 and TO_CTRL=0b-1 then reset TO_CNT on rising input edge; if SLICING=0b10 and TO_CTRL=0b1- then reset TO_CNT1 on falling input edge; if SLICING!=0b10 then reset counters TO_CNT,TO_CNT1 on each active edge selected by TOCTRL
1-1-$_B$	reset counters TO_CNT on tdu_word_evt
11-- $_B$	reset counter TO_CNT1 on tdu_frame_evt; if SLICING=0b01 then reset TO_CNT on tdu_frame_evt

1) tdu_start_000_event is defined as: Each writing of TOCTRL != 0 (independent of current TOCTRL) while TDU_START=0b000 and TDU is stopped (initially or stopped by TDU_STOP event). This event will last 1 system clock cycle.

Table 47 Behavior of TDU_RESYNC with SLICING = 0b11

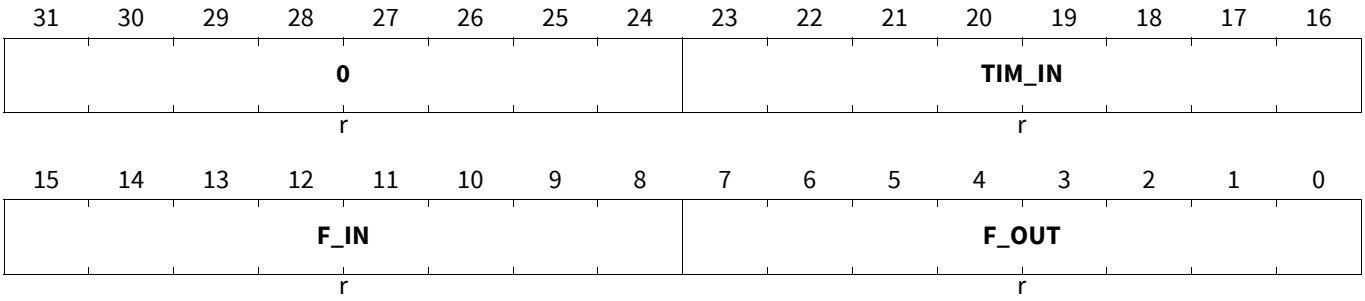
TDU_RESYNC	Behavior
0000_B	load counter TO_CNT with TOV2 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event [1]; reset counter TO_CNT1 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event [1]
---1 $_B$	load counter TO_CNT with TOV2 on each active edge selected by TOCTRL; reset counter TO_CNT1 on each active edge selected by TOCTRL
0-1- $_B$	load counter TO_CNT with TOV2 on tdu_word_evt
1-1-$_B$	reset counter TO_CNT on tdu_word_evt
-1--$_B$	reset counter TO_CNT1 on tdu_frame_evt
1000_B	load counter TO_CNT with TOV2; reset counter TO_CNT1 on event selected by EXT_CAP_SRC

28.13.8.19Register TIM[i]_INP_VAL

TIMi Input Value Observation Register

TIMi_INP_VAL (i=0-7)

TIMi Input Value Observation Register **(001074_H+i*800_H)** **Application Reset Value: 0000 0000_H**



Field	Bits	Type	Description
F_OUT	7:0	r	**Signals after TIM FLT unit**
F_IN	15:8	r	**Signals after INPSRC selection, before TIM FLT unit**

Field	Bits	Type	Description
TIM_IN	23:16	r	**Signals after TIM input signal synchronization**
0	31:24	r	**Reserved** Read as zero, shall be written as zero.