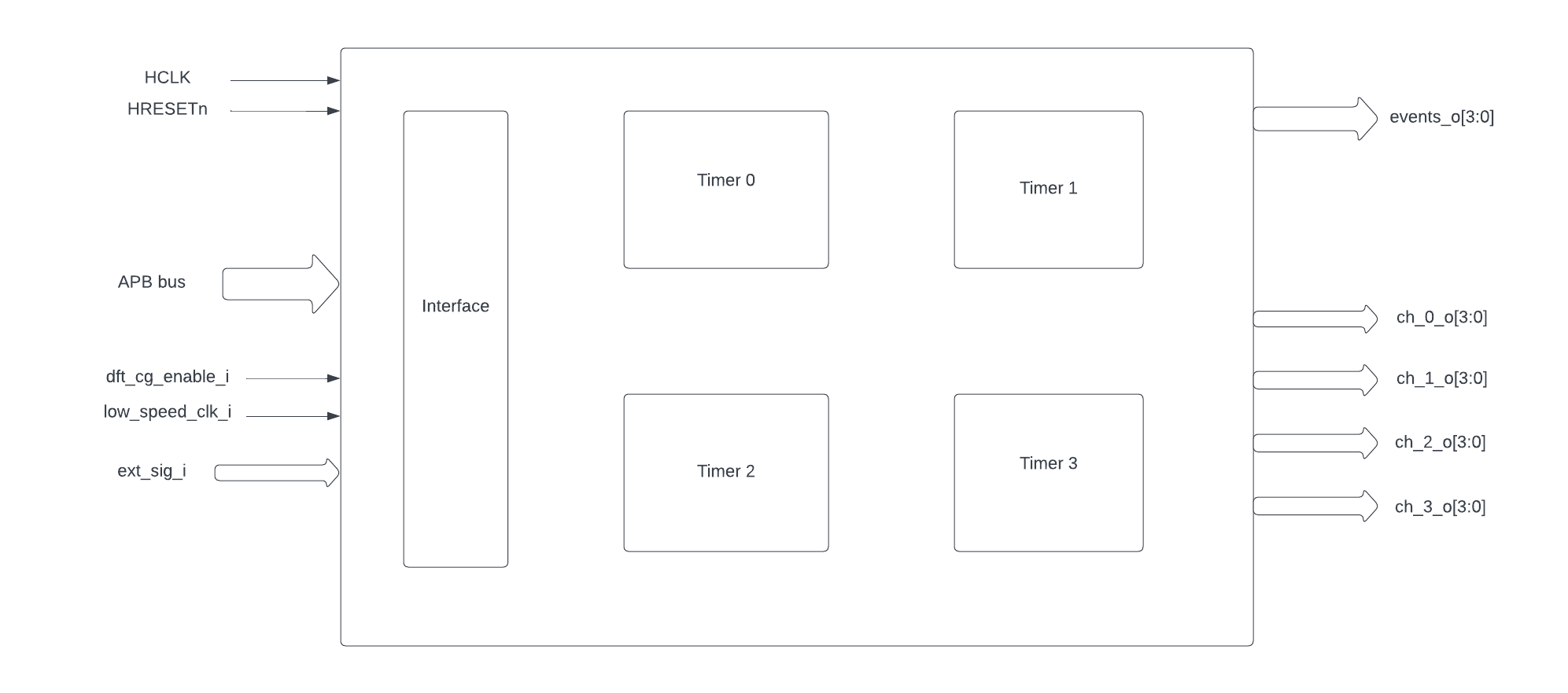
# **APB Advanced Timer**

**The Advanced Timer supports four programmable timers called “channels”. A typical use of the Advanced Timer is PWM generation.**

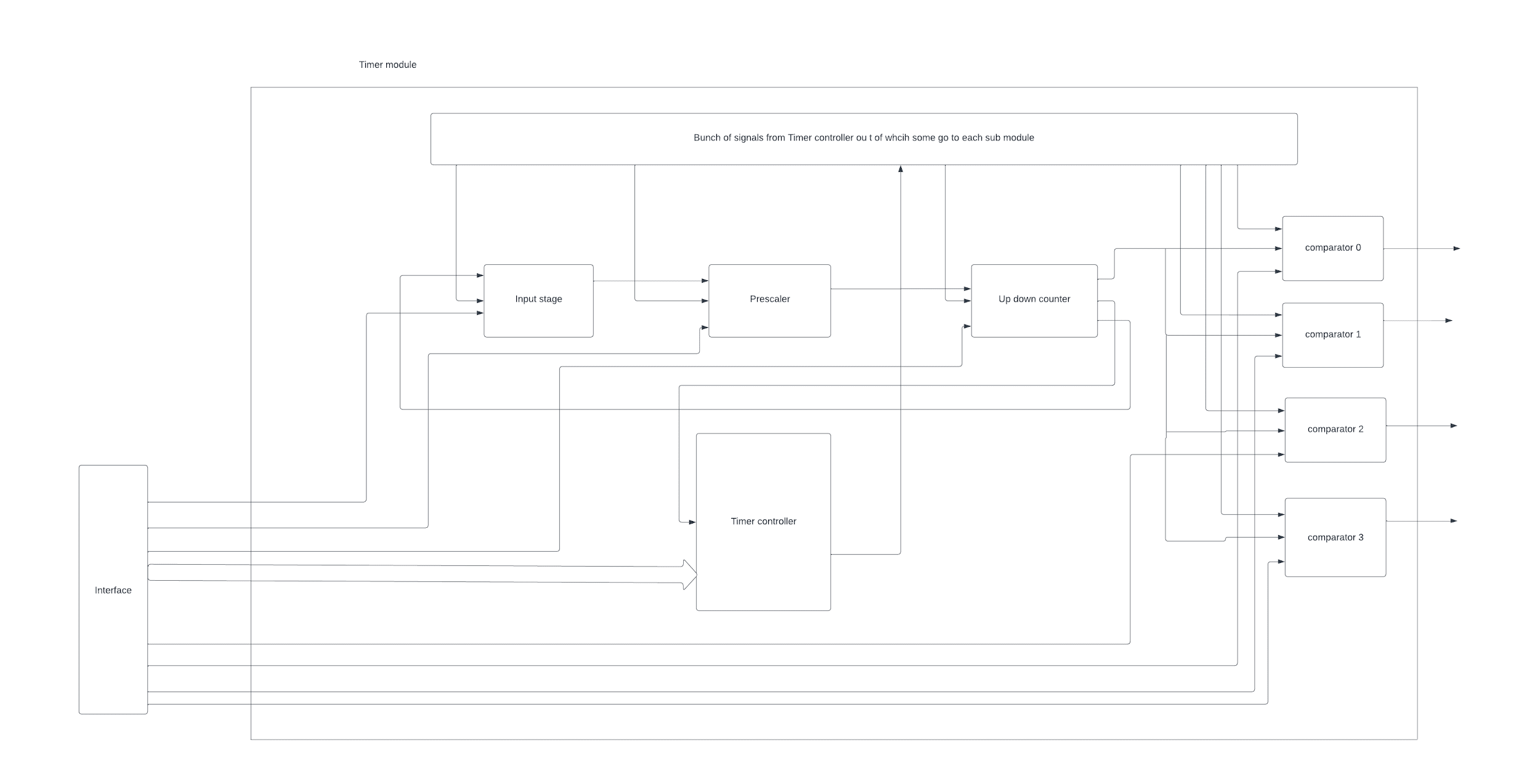
## **Features**

* **multiple trigger input sources:**
  + **output signal channels of all timers**
  + **32 GPIOs**
  + **reference clock at 32kHz**
  + **FLL clock**
* **configurable input trigger modes**
* **configurable prescaler for each timer**
* **configurable counting mode for each timer**
* **configurable channel threshold action for each timer**
* **four configurable output events**
* **configurable clock gating of each timer**

**Block Diagram:-**

****

**Timer module:-**

****

**Each submodule gets inputs from the timer controller,interface and from other modules as well.**

**Theory of Operation:-**

Input ports:-

1)HCLK :- External clock for synchronization

2)HRESETn :- reset pin to reset the timer

3)APB bus pins

4)dft\_cg\_enable\_i

5)low\_speed\_clk\_i

6)ext\_sig\_i

Output ports:

1)events\_o[3:0]

2)ch0\_o[3:0]

3)ch1\_o[3:0]

4)ch2\_o[3:0]

5)ch3\_o[3:0]

**HIGH LEVEL SPEC**

**APB advanced timer:-**

* When the HRESETn signal is low,then registers are default 0 and outputs are low.
* Four timer modules have four clock gates which will be enabled(meaning passes the ref clock to respective timer module) only when either dft\_cg\_enable\_i is high or the bit in respective position of REG\_CH\_EN register is high(0th bit for timer\_0,1st bit for timer\_1,etc).
* At every positive edge of the clock the CSR registers are updated based on APB signals.
* When the command register START is high and the timer is not active yet (which means the timer is started for the first time) then all the config values of all modules are commanded to be updated to default .They are,
  + The start value of the up down counter(TH\_LO)
  + The end value of the up down counter(TH\_HI)
  + The direction of the up down counter(default is 0)
  + The sawtooth mode of the up down counter(UPDOWNSEL)
  + The counter value of the up down counter (TH\_LO)
  + The prescaler value(PRESC),The MODE and INSEL register values.
  + For each channel the MODE and TH values
  + Here,The general update of all the config happens in sync with the positive edge of the clock but the config of the up down counter (TH\_LO,TH\_HI,direct and UPDOWNSEL are updated immediately).
* After the start of the model,The update to the submodules will happen based on register value UPDATE,RESET,and count update signal from the up down counter.
* At every clock positive edge,based on the command register START or STOP,the state of the timer is set as active or not .Once the timer is active then all the updates of the submodules depend on the corresponding registers.
  + The RESET command register
  + The UPDATE command register
* Based on the change in the config register CLKSEL ,it is decided whether the input selected from the set of inputs in ext\_sig\_i will be in sync with the rising edge of the low\_speed\_clk\_i in sync with the ref clock.
* At every positive edge of the clock,the input signal is selected from a set of signals in ext\_sig\_i using the config register INSEL value and how the events are generated from the signal is decided by the config register MODE.
  + If MODE is 3’b000
    - The event is always high
  + If MODE is 3’b001
    - The event is sensitive to the negation of the signal selected
  + If MODE is 3’b010
    - The output event is sensitive to the input signal selected
  + If MODE is 3’b011
    - The output event is sensitive to the rising edge of the selected signal in sync with the clock.
  + If MODE is 3’b100
    - The output event is sensitive to the rising edge of the selected signal in sync with the clock.
  + If MODE is 3’b101
    - The output event is sensitive to both rising edge and falling edge of the selected signal in sync with the clock.
  + If MODE is 3’b110
    - If the timer is armed ,i,e,the register ARM is high then the event is made high for the rising edge of the selected signal and remains the same until the next rising edge of the signal.If ARM register is low,then the output event is low forever.
  + If MODE is 3’b111
    - If the timer is armed ,i,e,the register ARM is high then the event is made high for the falling edge of the selected signal and remains the same until the next falling edge of the signal.If ARM register is low,then the output event is low forever.
* The Event signal generated from the selected input based on the MODE config register in the previous step is scaled based on the prescaler value(PRESC register value).At every positive edge of the clock the register PRESC is updated.The scaling happens in a way that after every time the number of events in sync with the external clock generated is equal to the PRESC register value then counter is made to 0 and an event is generated.Like this whenever the lock synced events generated is equal to PRESC value then one output event is generated at positive edge of the clock(the frequency is scaled according to the PRESC register value).
* The above output scaled events generated go to the up down counter.For every event the counter is incremented starting from the start value(TH\_LO register) .Based on the register UPDOWNCLK representing the sawtooth mode,it is decided whether the counter should reset after reaching end of the counting range (TH\_HI) or it should reverse the direction and go counting down to start value(TH\_LO) after which it resets to the default values of start,stop,direction,etc .
* At every input event in sync with the clock an **output event** is generated and also the counter is incremented .Whenever the counter reaches the end of a counting range an event is generated representing the end of the counter and reset happens.The output port representing the counter is updated at every clock positive edge.
* Here, the counter value,event representing the end of the timer,the **output event** generated above ,and the UPDOWNSEL register value are used by the comparator below.

**Comparator:-**

* At every positive edge of the clock,When the timer is started the first time or explicitly updated through the update command register named UPDATE ,the module is updated then , the register values TH and MODE of the register are read in which TH value is the comparator threshold value and MODE is the operation that should be done when counter of the up down counter reaches the comparator threshold value.
* At every positive edge of the clock when the event coming out of the up down counter is high ,based on the register MODE value ,output is generated accordingly.
* There are two events that can happen in the comparator,
  + When timer counter value reaches the comparator offset **(match event)**
  + When the UPDOWNSEL register is high and the timer reaches its end or when UPDOWNSEL is low and the timer counter value reaches the comparator threshold offset.**(event\_2)**
* define OP\_SET 3'b000
* define OP\_TOGRST 3'b001
* define OP\_SETRST 3'b010
* define OP\_TOG 3'b011
* define OP\_RST 3'b100
* define OP\_TOGSET 3'b101
* define OP\_RSTSET 3'b110
* If MODE value is OP\_SET
  + Then the output event is high when there is a match otherwise remains the same .
* If MODE value is OP\_TOGRST
  + Then if sawtooth mode is on ,then if a match happens then the output event is toggled else if event\_2 happens then output event is low.
  + If sawtooth mode is off,then if match event happens and event\_2 doesn't happen then output event is toggle and event\_2 is made high ,else if match event happens and event\_2 also happens then output event is made low and event\_2 is also made low.
* If MODE value is OP\_SETRST
  + Then if sawtooth mode is on ,then if a match happens then the output event is high else if event\_2 happens then output event is low.
  + If sawtooth mode is off,then if match event happens and event\_2 doesn't happen then output event is made high and event\_2 is made high.,else if match event happens and event\_2 also happens then output event is made low and event\_2 is also made low.
* If MODE value is OP\_TOG
  + Then the output event is toggled when the match event occurs else remains the same.
* If MODE value is OP\_RST
  + Then the output event is made low when the match event occurs else remains the same.
* If MODE value is OP\_TOGSET
  + Then if sawtooth mode is on ,then if a match happens then the output event is toggled else if event\_2 happens then output event is high.
  + If sawtooth mode is off,then if match event happens and event\_2 doesn't happen then output event is toggle and event\_2 is made high ,else if match event happens and event\_2 also happens then output event is made high and event\_2 is also made low.
* If MODE value is OP\_RSTSET
  + Then if sawtooth mode is on ,then if a match happens then the output event is low else if event\_2 happens then output event is high.
  + If sawtooth mode is off,then if match event happens and event\_2 doesn't happen then output event is made low and event\_2 is made high.,else if match event happens and event\_2 also happens then the output event is made high and event\_2 is also made low.
* By default the output event remains the same (state remains same until further change in input) and event\_2 is kept low.

**CSR register for the timer module 0:-**

**REG\_TIM0\_CMD** offset=0x000

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:5 |  |  |  |
| ARM | 4:4 | Config | R/W | arm command bitfield |
| RESET | 3:3 | Config | R/W | reset command bitfield |
| UPDATE | 2:2 | Config | R/W | update command bitfield |
| STOP | 1:1 | Config | R/W | Stop command field |
| START | 0:0 | Config | R/W | Start command field |

**REG\_TIM0\_CFG** offset=0x004

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:24 |  |  |  |
| PRESC | 23:16 | Config | R/W | prescaler value configuration bitfield |
| Reserved | 15:13 |  |  |  |
| UPDOWNSEL | 12:12 | Config | R/W | center-aligned mode configuration bitfield |
|  |  |  |  | 1’b0: The counter counts up and down alternatively |
|  |  |  |  | 1’b1: The counter counts up and resets to 0 when it reaches the threshold. |
| CLKSEL | 11:11 | Config | R/W | clock source configuration bitfield |
|  |  |  |  | 1’b0: FLL |
|  |  |  |  | 1’b1: reference clock at 32kHz |
| MODE | 10:8 | Config | R/W | trigger mode configuration bitfield |
|  |  |  |  | 3’h0: trigger event at each clock cycle |
|  |  |  |  | 3’h1: trigger event if input source is 0 |
|  |  |  |  | 3’h2: trigger event if input source is 1 |
|  |  |  |  | 3’h3: trigger event on input source rising edge |
|  |  |  |  | 3’h4: trigger event on input source falling edge |
|  |  |  |  | 3’h5: trigger event on input source falling or rising edge |
|  |  |  |  | 3’h6: trigger event on input source rising edge when armed |
|  |  |  |  | 3’h7: trigger event on input source falling edge when armed |
| INSEL | 7:0 | Config | R/W | input source configuration bitfield |
|  |  |  |  | 0-31: GPIO[0] to GPIO[31] |
|  |  |  |  | 32-35: Channel 0 to 3 of ADV\_TIMER0 |
|  |  |  |  | 36-39: Channel 0 to 3 of ADV\_TIMER1 |
|  |  |  |  | 40-43: Channel 0 to 3 of ADV\_TIMER2 |
|  |  |  |  | 44-47: Channel 0 to 3 of ADV\_TIMER3 |

**REG\_TIM0\_TH** offset =0x008

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| TH\_HI | 31:16 | Config | R/W | threshold high part configuration bitfield |
| TH\_LO | 15:0 | Config | R/W | threshold low part configuration bitfield |

**REG\_TIM0\_CH0\_TH** offset=0x00C

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:19 |  |  |  |
| MODE | 18:16 | Config | R/W | channel 0 threshold match action on channel output signal  configuration bitfield |
|  |  |  |  | 3’h0: set |
|  |  |  |  | 3’h1: toggle then next threshold match action is clear. |
|  |  |  |  | 3’h2: set then next threshold match action is clear |
|  |  |  |  | 3’h3: toggle |
|  |  |  |  | 3’h4: clear |
|  |  |  |  | 3’h5: toggle then next threshold match action is set |
|  |  |  |  | 3’h6: clear then next threshold match action is set |
| TH | 15:0 | Config | R/W | channel 0 threshold configuration bitfield |

**REG\_TIM0\_CH1\_TH** offset=0x010

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:19 |  |  |  |
| MODE | 18:16 | Config | R/W | channel 1 threshold match action on channel output signal  configuration bitfield |
|  |  |  |  | 3’h0: set |
|  |  |  |  | 3’h1: toggle then next threshold match action is clear. |
|  |  |  |  | 3’h2: set then next threshold match action is clear |
|  |  |  |  | 3’h3: toggle |
|  |  |  |  | 3’h4: clear |
|  |  |  |  | 3’h5: toggle then next threshold match action is set |
|  |  |  |  | 3’h6: clear then next threshold match action is set |
| TH | 15:0 | Config | R/W | channel 1 threshold configuration bitfield |

**REG\_TIM0\_CH2\_TH** offset=0x014

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:19 |  |  |  |
| MODE | 18:16 | Config | R/W | channel 2 threshold match action on channel output signal  configuration bitfield |
|  |  |  |  | 3’h0: set |
|  |  |  |  | 3’h1: toggle then next threshold match action is clear. |
|  |  |  |  | 3’h2: set then next threshold match action is clear |
|  |  |  |  | 3’h3: toggle |
|  |  |  |  | 3’h4: clear |
|  |  |  |  | 3’h5: toggle then next threshold match action is set |
|  |  |  |  | 3’h6: clear then next threshold match action is set |
| TH | 15:0 | Config | R/W | channel 2 threshold configuration bitfield |

**REG\_TIM0\_CH3\_TH** offset=0x018

| Field | Bits | Type | Access | Description |
| --- | --- | --- | --- | --- |
| Reserved | 31:19 |  |  |  |
| MODE | 18:16 | Config | R/W | channel 3 threshold match action on channel output signal  configuration bitfield |
|  |  |  |  | 3’h0: set |
|  |  |  |  | 3’h1: toggle then next threshold match action is clear. |
|  |  |  |  | 3’h2: set then next threshold match action is clear |
|  |  |  |  | 3’h3: toggle |
|  |  |  |  | 3’h4: clear |
|  |  |  |  | 3’h5: toggle then next threshold match action is set |
|  |  |  |  | 3’h6: clear then next threshold match action is set |
| TH | 15:0 | Config | R/W | channel 3 threshold configuration bitfield |

**REG\_TIM0\_COUNTER** offset=0x02C

| Field | bits | Type | access | description |
| --- | --- | --- | --- | --- |
| T0\_COUNTER | 31:0 | Status | R | ADV\_TIMER0  counter register |