Version 1.5 – June 2019

Abstract

This application note describes the capabilities and configuration of the syn1588® event I/Os namely the TRIGGER output, the PERIOD output, and the EVENT input.

syn1588® Clock Structure

The generation (TRIGGER/PERIOD output) and capturing (EVENT input) of syn1588® I/Os is directly controlled by the highly accurate, synchronized syn1588® hardware clock. The syn1588® hardware clock is an adder based clock, whose increment is depending on the configured STEP value; basically the period of the syn1588® hardware clock. Please note that the nominal STEP value may not be chosen freely but has to defined with respect to capabilities of the hardware. E.g. the syn1588® PCIe NIC Revision 2 relies on a syn1588® clock frequency of 125 MHz (8 ns period). The clock frequency used in a syn1588® PCIe NIC can be verified by reading register 0x204 via the syn1588 utility as shown below; it returns the clock frequency in Hz.

```
>0x204
0x07735940
```

In the example above, the register read out value corresponds to 125 MHz. This value may change with the firmware of the card.

Figure 1 shows the graphical illustration of the syn1588® clock value being incremented as a function of the STEP value at every rising edge of the system clock.

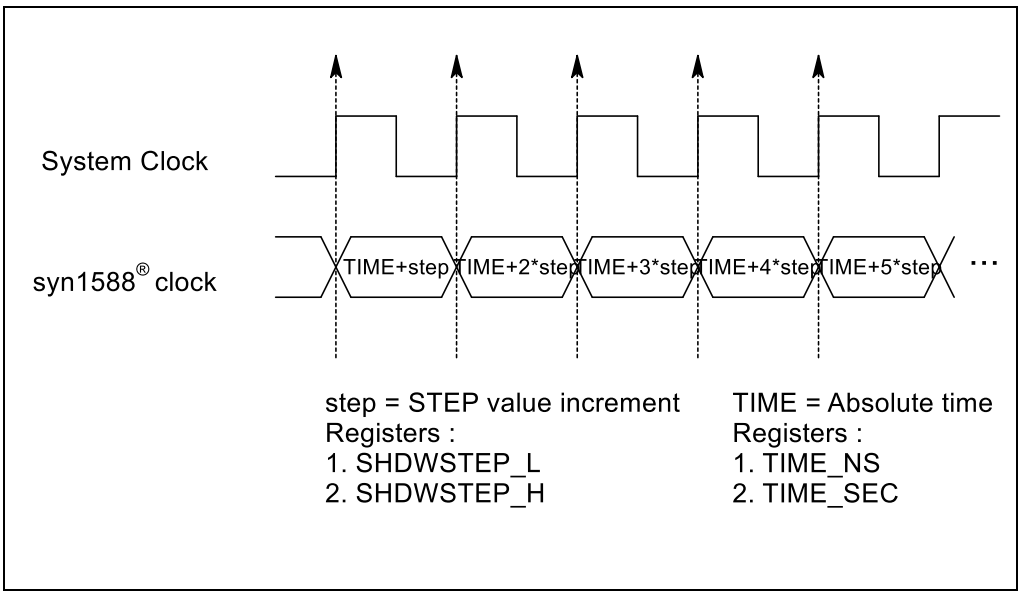

Figure 1 At every rising edge of the system clock, syn1588®clock increments with STEP value.

Example: Configuring the syn1588® clock manually

All values of the Adder Based Clock are configured automatically using their respective start-up values whenever the syn1588® PTP stack is invoked. However, both the step size and the start-up value of the clock can be configured manually as well as part of a custom initialization procedure. For example, to start the syn1588® clock with an absolute time of 12 sec and 10 ns together with a STEP value of 8 ns, the following registers need to be written. Please note that the appropriate clock period has to be used to configure the STEP size.

Please note that these commands should not be executed when the syn1588® PTP Stack is already running.

Please note, that after changing the STEP value for some temporary experiments, the syn1588® PTP Stack has to be started with the additional option "-G" (clean-start), to restore the correct default STEP-value.

- Write SHDWSTEP_L[31:0] = 0 sub-ns (Addr. 0x50)
- Write SHDWSTEP_H[31:0] = 8 ns (Addr. 0x54)
- Write SHDWTIME_NS[31:0] = 10 ns (Addr. 0x84)
- Write SHDWTIME_SEC[31:0] = 12 sec (Addr. 0x88)
- After writing the desired values to the SHDWSTEP and SHDWTIME registers they have to be loaded into the syn1588® clock synchronously. This is done by setting bit 0 of the TIMECTRL register which initiates the transfer of all values from the shadow registers to the actual clock registers.
- Write TIMECTRL = 0x00000001. (Addr. 0x48).

This will enable loading of the shadow registers above configured to the actual registers. As soon as the clock increment is set to a value greater than zero the syn1588® clock will start functioning.

The syn1588 utility commands for this command script sequence is shown below.

```
>0x050 0x0
>0x054 0x00000800
>0x084 0x0000000A
>0x088 0x0000000C
>0x048 0x00000001
```

Please refer to the syn1588® Register Map Application Note for a detailed description of the IEEE 1588 register layout and for details of all the registers.

syn1588® Event I/O s

This section presents the example scenarios of generation and capturing of the syn1588® events. Please note, that for all examples it is assumed: if the syn1588® PCIe NIC with the syn1588® PTP Stack running is in slave mode it is already synchronized to the master.

Example: Generate a TRIGGER Output

In this example we will generate a TRIGGER, i.e. a single event at a given precise time in the future. Typically there are two TRIGGER outputs available that may be independently controlled: TRIGGER0 and TRIGGER1. TRIGGER0 is a FIFO based implementation that allows generation of densely packed events. TRIGGER0 allows generation of up to 16 TRIGGER events. TRIGGER1 is a register based implementation that allows generation of a single TRIGGER event.

The following steps describe the actions needed for the generation of a single TRIGGER event on TRIGGER0 at an absolute time of 11 sec, 10 ns with an output state ‘1’.

- The syn1588® hardware clock needs to be initialized with the required STEP value as described in the above example. Alternatively one may synchronize the card to an external Grandmaster using the syn1588® PTP Stack. In this case the syn1588® PTP Stack will manage STEP and TIME registers.
- The desired time in future where the TRIGGER event shall be generated and also the desired output state of the TRIGGER event has to be written to the corresponding TRIGTIME0_L/TRIGTIME0_H registers.
- Write EVENTCTRL = 0x00000004, which enables the TRIGGER0 function.
- Write TRIGTIME0_L[31:0] = 10 ns
- Write TRIGTIME0_H[19:0] = 11 sec and TRIGTIME0_H[21:20] = “01” to select the TRIGGER output state ‘1’
- Always write to the lower register first.

The syn1588 utility commands for this command sequence is shown below.

```
>0x084 0x0000000A
>0x088 0x0000000A
>0x048 0x80000000
>0x04C 0x00000004
>0x0D0 0x0000000A
>0x0D4 0x0010000B
```

Please note that the appropriate clock period has to be used to configure the STEP size.

Please further note, that this example also contains the commands to set the time, which is not recommended when the syn1588® PTP Stack is running.

After writing to the above said registers, the TRIGGER changes its output state to '1' precisely at the point in time when the value of the clock reached the value configured in TRIGTIME. Firing of the TRIGGER output will happen only when the local TIME of the syn1588® hardware clock is greater than or equal to the configured TRIGTIME.

Due to the architecture of the syn1588® hardware clock as seen from Figure 1, the next local TIME of the syn1588® hardware clock will be incremented by the STEP value. Hence, the TRIGGER event may not exactly happen at the configured TRIGTIME but it may take an additional clock cycle and therefore happen in the time interval "TRIGTIME – 0" and "TRIGTIME + STEP". Figure 2 illustrates that if T0 is the configured TRIGTIME, T1 is the actual time when the TRIGGER is fired.

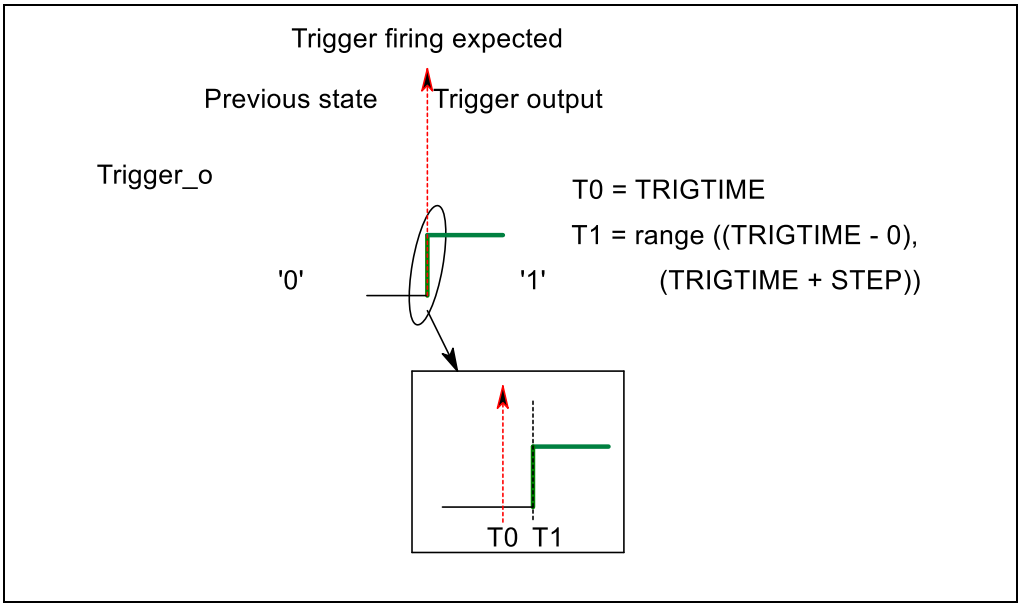

Figure 2 TRIGGER output after the TRIGTIME elapsed. The TRIGGER output is set to '1' in this case. T0 is the configured TRIGTIME and T1 is the actual time when the TRIGGER is fired.

Please note that the example given above assumes that the previous state of the TRIGGER output had been '0'. If it would have been already '1' no state transition will occur on the TRIGGER output signal. One may choose to set the TRIGGER output signal on a TRIGGER event to '0', to '1' or to toggle it's state.

Example: Generate a PERIOD Output

In this example we will generate a PERIOD signal with a configured PERIODTIME.

Typically there are two PERIOD outputs available that may be independently controlled: PERIOD0 and PERIOD1. It is possible to generate a purely static value on the PERIOD signal as well as a standard (i.e. 50% duty cycle) PERIOD signal. This functionality is identical for both PERIOD0 and PERIOD1.

In addition to this, it is possible to generate a PERIOD0 signal with a programmable duty cycle. This functionality is available only for the PERIOD0 signal. In this case, PERIOD1 signal is disabled and cannot be used, because both the PERIODTIMEx_L/PERIODTIMEx_H registers of PERIOD0 and PERIOD1 are used to configure the frequency as well as the duty cycle for PERIOD0 signal.

An alternative method to configure the period outputs, is to use the syn1588 utility (for the syn1588® PCIe NIC) with the "frequency" command.

Note, that the maximum allowed frequency of the period output is dependent on the clock frequency. The minimal half-period time is 4.5 clock periods. This results in a maximal frequency of the period output of clock_frequency/9.

The following steps describe the actions needed for the generation of a PERIOD0 signal with a period of 1000 us and with an initial value '1', starting immediately.

- The syn1588® hardware clock is assumed to be already initialized with the required STEP value as shown in the first example. Again the syn1588® PTP Stack may be used to control the operation of the syn1588® hardware clock.
- The desired half-period time has to be written into the PERIODTIME0_L/PERIODTIME0_H registers. Thus the generated PERIOD0 signal contains a time period equal to 2*PERIODTIME configured.
- Write PERIODTIME0_L[31:0] = 0xA1200000
- Write PERIODTIME0_H[31:0] = 0x00000007
- Always write to the lower register first.
- Write EVENTCTRL = 0x00000150, which enables the PERIOD0 function, enables the PERIOD0 output and also sets the initial value to '1'.

The syn1588 utility commands for this command sequence is shown below.

If the PERIOD0 is already running, it has to be stopped first by first clearing bit 4 of the EVENTCTRL register. Afterwards the period0 output has to be set to zero by clearing bit 9, so there can be seen a transition at the start-time.

```
>0x0F0 0xA1200000
>0x0F4 0x00000007
>0x04C 0x00000150
```

After writing to the above said registers, the PERIOD0 signal starts with ON state. After the configured PERIODTIME elapses, the PERIOD0 signal switches to the OFF state. Due to the architecture of the syn1588® hardware clock, the actual switching of the PERIOD output signal will be in the time interval "PERIODTIME – 0 " and "PERIODTIME + STEP", similar to that of the TRIGGER function explained in the example above. Please note that this uncertainty is limited to a single edge on the output signal. The next occurrence of the next edge is computed using the accurate time. This uncertainty will not (!) be accumulated thus is will not affect the precise value of the frequency. Figure 3 illustrates this example.

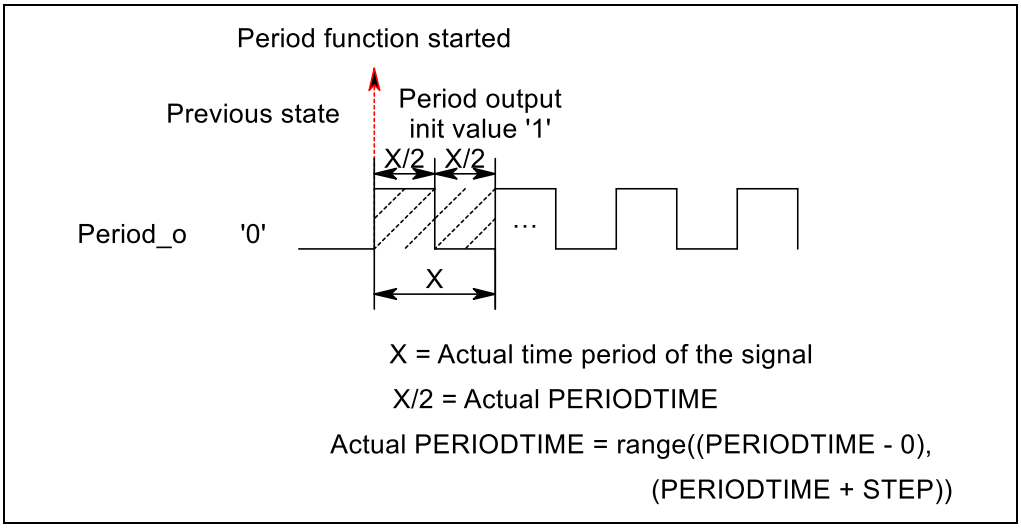

Figure 3 The relation between the actual PERIODTIME and the configured PERIODTIME is depicted in this figure.

Example: Capture an EVENT Input

In this example we will capture an EVENT and read the EVENTTIME registers. Typically there are two EVENT inputs available that may be controlled independently: EVENT0 and EVENT1. EVENT0 is a FIFO based implementation that allows capturing densely packed events. It allows to capture up to 16 events at a time without the need to access the EVENT0 register. EVENT1 is a register based implementation that allows to capture a single event.

An EVENT input is captured whenever a rising edge of the EVENT input signal occurs. Also the timestamp is drawn after a three stage synchronization of the EVENT input signal.

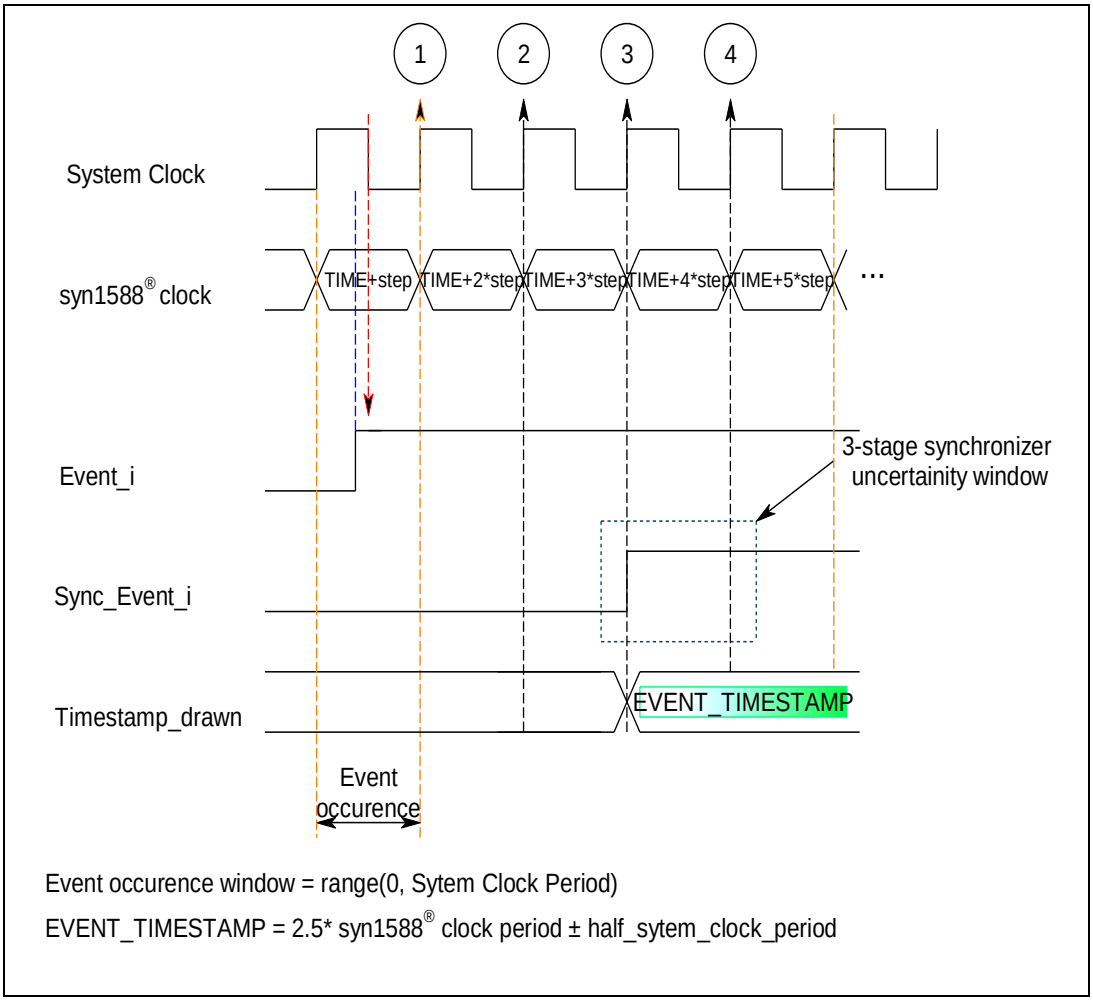

Figure 4 Capturing of the EVENT input after a three-stage synchronization and the uncertainty involved while drawing the corresponding EVENT_TIMESTAMP (EVENTTIME) is shown in this figure.

As seen from Figure 4, Event_i is the EVENT input and Sync_Event_i is the synchronized EVENT input.

EVENTIME is recorded at 3rd or 4th occurrence of the system clock, due to the uncertainty of the synchronizer stage. Therefore, on average, a delay of 2.5*(syn1588® clock period) is added to the EVENTTIME. Also, in order to effectively monitor the rising edge of the EVENT input, both edges of the system clock may be used to sample the EVENT input in the design.

Therefore, in the case where the EVENT input is detected on the negative edge of the system clock, an additional half period time of the system clock is added to the timestamp drawn. Hence,

```
EVENTTIME = Expected TIME + 2.5*(syn1588® clock period) ±
(Half of system clock period)
```

The following steps describe the actions needed to capture the EVENT0 input.

- The syn1588® hardware clock is assumed to be already initialized with the required STEP value. The syn1588® PTP Stack may be used to control the operation of the syn1588® hardware clock.
- The EVENTCTRL register needed to be configured to enable the EVENT0 function.
- Write EVENTCTRL = 0x00000001, to enable the EVENT0 function.

The syn1588 utility commands for starting the syn1588® clock and to writing the above said registers is shown below.

```
>0x04C 0x00000001
```

The syn1588 utility commands for reading the EVENTTIME0_L/H registers is shown below .

```
>0x0A4
0x00000000
>0x0A8
0x00000000
```

At the expected time of the EVENT input rising edge, start reading EVENTTIME0_L/H registers to get the first EVENTTIME. As EVENT0 is capable to store dense events, 16 entries can be stored. As explained above, the read out values of the EVENTTIME will have an additional delay.

Read out the EVENT0 FIFO until it is empty, so that next set of events can be stored. In order to determine, if the EVENT0 FIFO has become empty, read out the EVENTTIME0_L/H registers until a 0x0 value pair is read. Always access the lower register first.

Example: Enable PERIOD with a TRIGGER

In this example we will enable a PERIOD signal while firing a TRIGGER event. This allows starting the generation of a periodical signal at an exact time (simultaneously on all nodes in the network).

In addition to starting the PERIOD signal independently, one can also start it with the TRIGGER event. In case of continuous TRIGGER output changes (in case of TRIGGER 0 for example), the PERIOD function will continue to operate independently of the TRIGGER as soon as the very first TRIGGER condition is met.

The following steps describe the actions needed to enable the PERIOD0 signal with a time period of 1000 us, when the TRIGGER0 is fired at an absolute time of 11 sec and 10 ns.

- The syn1588® hardware clock is assumed to be initialized already with the required STEP value, but the absolute time is set in this example. The syn1588® PTP Stack may be used to control the operation of the syn1588® hardware clock.
- The desired half-period time has to be written into the PERIODTIME0_L/PERIODTIME0_H registers. Thus the generated PERIOD0 signal contains a time period equal to 2*PERIODTIME configured.
- Write PERIODTIME0_L[31:0] = 0xA1200000
- Write PERIODTIME0_H[31:0] = 0x00000007
- Always write to the lower register first.
- Before writing the Trigger Time, enable Trigger 0 in the EVENTCTRL register
- The desired time in the future where the TRIGGER event shall be generated as well as the desired output state of the TRIGGER event has to be written to the corresponding TRIGTIME0_L/TRIGTIME0_H registers.
- Write TRIGTIME0_L[31:0] = 10 ns
- Write TRIGTIME0_H[19:0] = 11 sec and TRIGTIME0_H[21:20] = “00” to select the TRIGGER output state ‘0’.
- Always write to the lower register first.
- Write EVENTCTRL = 0x00018054, which enables TRIGGER0, sets TRIGGER0 output state to ‘0’, enables PERIOD0 function, enables PERIOD0 output, sets initial value of PERIOD0 to ‘0’ and starts PERIOD0 function when TRIGGER0 is fired.

The syn1588 utility commands for starting the syn1588® clock and to writing the above said registers is shown below.

Please note that the appropriate clock period has to be used to configure the STEP size.

Please note, that this example also contains the commands to set the time, which is not recommended when the PTP stack is running.

```
>0x084 0x0000000A
>0x088 0x0000000A
>0x048 0x80000000
>0x0F0 0xA1200000
>0x0F4 0x00000007
>0x04C 0x00000004
>0x0D0 0x0000000A
>0x0D4 0x0000000B
>0x04C 0x00018054
```

Figure 5 shows that the PERIOD function is enabled as soon as the TRIGGER condition is met. It can be seen that if TRIGTIME is the configured time, at time instant T1, the TRIGGER is fired and the value of PERIOD output is '0' as defined by the initial value configured in EVENTCTRL register.

At time instant T2 = T1 + PT (range((PERIODTIME -0), (PERIODTIME + STEP))), the PERIOD output is changed to '1'. Similarly, if the PERIOD function is still enabled, the PERIOD output toggles at the next time instant within the interval "PERIODTIME – 0" to "PERIODTIME + STEP" as shown in the figure below. As explained before, the uncertainty will not (!) be accumulated.

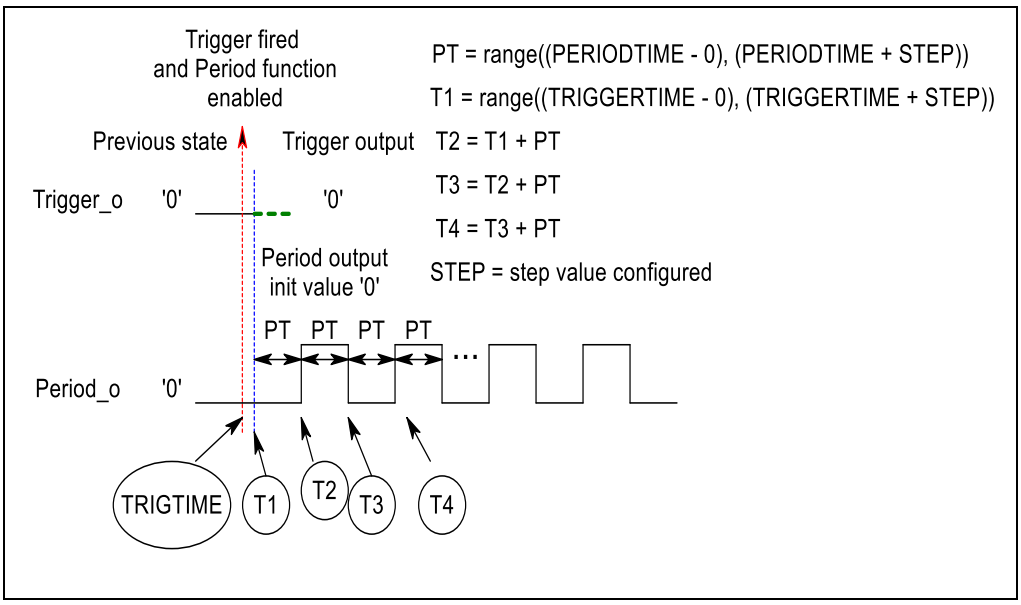

Figure 5 Firing of TRIGGER condition enables the PERIOD function. The relation of the time between the TRIGGER and PERIOD events is depicted in the figure. In this figure, the initial value of the PERIOD output is configured as '0'.

Example: Loopback TRIGGER/PERIOD to EVENT

In this example, we will enable a PERIOD with a TRIGGER and then loopback the PERIOD output to the EVENT input.

Enabling the PERIOD output with TRIGGER has already been explained in the example above. With the help of the IOMATRIX control register, one can loopback the TRIGGER or PERIOD output signals to the EVENT input so that the required EVENTTIME can be captured.

The following steps describe the actions needed to enable the PERIOD0 signal with a time period of 1000 us, when the TRIGGER0 is fired at an absolute time of 20 sec and 10 ns and also to loopback the PERIOD0 signal to the EVENT input and finally read the EVENTTIME registers.

- The syn1588® hardware clock is assumed to be initialized already with the required STEP value, but the absolute time is set in this example. The syn1588® PTP Stack may be used to control the operation of the syn1588® hardware clock.
- The desired half-period time has to be written into the PERIODTIME0_L/PERIODTIME0_H registers. Thus the generated PERIOD0 signal contains a time period equal to 2*PERIODTIME configured.
- Write PERIODTIME0_L[31:0] = 0xA1200000
- Write PERIODTIME0_H[31:0] = 0x00000007
- Always write to the lower register first.
- In order to loopback the PERIOD0 to EVENT0, one has to write to the IOMATRIX control register. The default value of the register is 0x00430059.
- Since we are interested only in the pins driving the EVENT input, we leave the remaining bits as they are and set only the required bits of the IOMATRIX register.
- Write IOMATRIX control register = 0x00450059, which loops back the PERIOD0 output to the EVENT0 input.
- Before writing the Trigger Time, enable Trigger 0 in the EVENTCTRL register
- The desired time in future where the TRIGGER event shall be generated and also the desired output state of the TRIGGER event has to be written to the corresponding TRIGTIME0_L/TRIGTIME0_H registers.
- Write TRIGTIME0_L[31:0] = 10 ns
- Write TRIGTIME0_H[19:0] = 20 sec and TRIGTIME0_H[21:20] = “00” to select the TRIGGER output state ‘0’
- Always write to lower register first.

- Write EVENTCTRL = 0x00018055, which enables EVENT0 function, enables TRIGGER0 function, sets TRIGGER0 output state to '0', enables PERIOD0 function, enables PERIOD0 output, sets initial value of PERIOD0 to '0' and starts PERIOD0 function when TRIGGER0 is fired.

The syn1588 utility commands for starting the syn1588® clock and to writing the above said registers is shown below.

Please note that the appropriate clock period has to be used to configure the STEP size.

Please further note, that this example also contains the commands to set the time, which is not recommended when the syn1588® PTP Stack is running.

```
>0x084 0x0000000A
>0x088 0x0000000A
>0x048 0x80000000
>0x0F0 0xA1200000
>0x0F4 0x00000007
>0x200 0x00450059
>0x04C 0x00000004
>0x0D0 0x0000000A
>0x0D4 0x00000014
>0x04C 0x00018054
```

The syn1588 utility commands for reading the TIME registers and EVENTTIME0_L/H registers of the syn1588® hardware clock is shown below

```
>time
9833,055784110
>0x0A4
0x00000000
>0x0A8
0x00000000
```

As shown in the script above 'time' command returns the current TIME of the syn1588® hardware clock with the notation "seconds, nanoseconds". By reading the current 'time', we can determine if the expected time has elapsed and then read the EVENTTIME0 registers 0x0A4 and 0x0A8 to get the timestamp of the captured EVENT. Always access the lower register first.

In this example, we have initialized the syn1588® hardware clock with an absolute time of 10 sec and 10 ns. And, the TRIGGER0 is configured with a TRIGTIME of 20 sec and 10 ns. This means that, the TRIGGER0 fires and starts the PERIOD0 function at 20 sec and 10 ns with an uncertainty of "0 to STEP" value. But, the initial value of the PERIOD0 is configured as '0', which means that in order to detect a rising edge on the EVENT input, it needs an additional time of PERIODTIME plus an uncertainty of "0 to STEP" value. Also the 3-stage synchronizer adds a delay of 2.5*(syn1588® clock period) ± (half of system clock period).

Therefore, in this example the first EVENTTIME captured will be range((TRIGTIME - 0), (TRIGTIME + STEP)) + range((PERIODTIME - 0), (PERIODTIME + STEP)) + 2.5*(syn1588® clock period) ± (Half of system clock period) = range([20 sec, 10 ns], [20 sec, 20 ns]) + range([500 us -0], [500 us + 10 ns]) + [2.5*(10 ns) ± (5 ns)].

Figure 6 illustrates the sequence of events in this example along with the calculation involved.

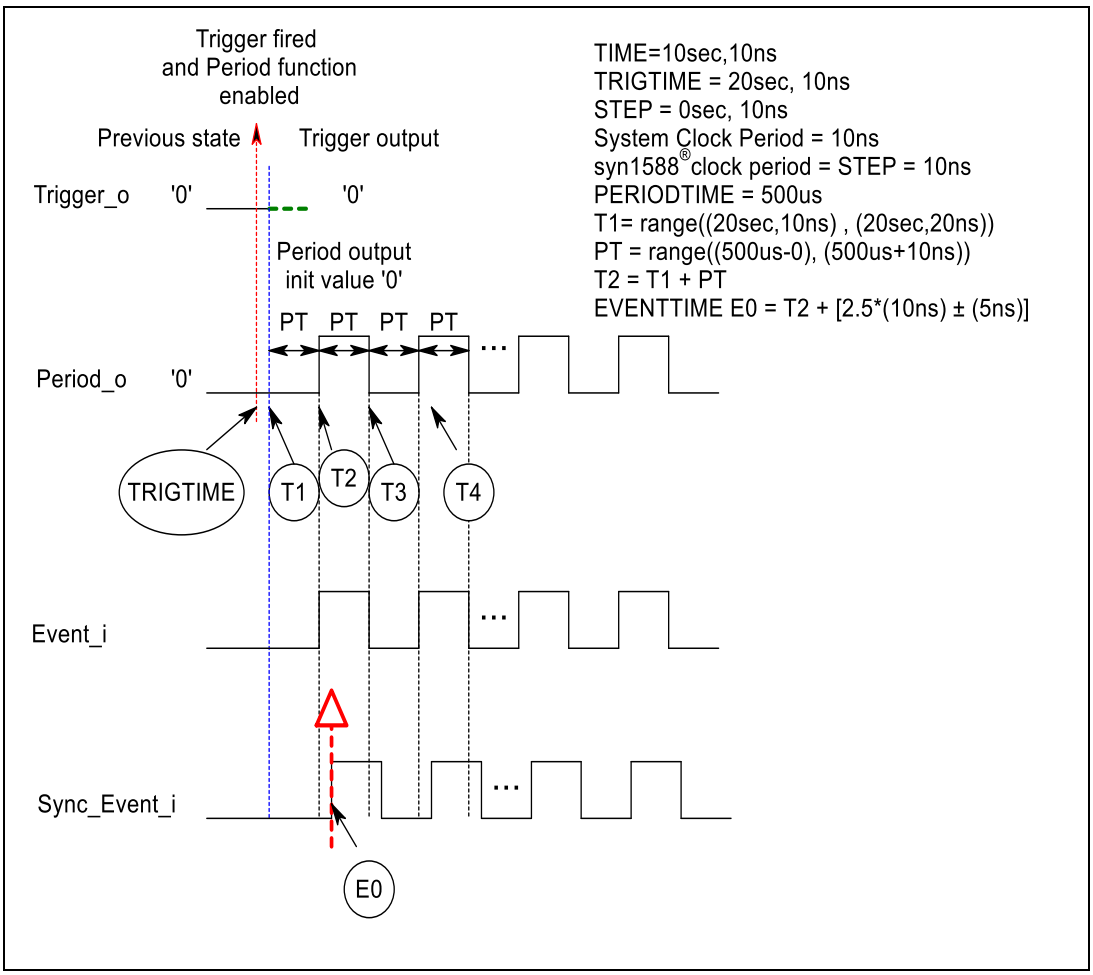

Figure 6 The figure depicts that the PERIOD signal is started with the TRIGGER and is looped back to the EVENT input. The EVENT input follows the PERIOD output, while the synchronized EVENT input has a delay equal to that of the 3-stage synchronizer.

Example: Generate PERIOD0 with programmable duty cycle

In this example, we will generate PERIOD0 signal with a programmable duty cycle.

When we use the programmable duty cycle function for PERIOD0, PERIOD1 function is not available anymore, as both the PERIODTIMEx_L/H registers of PERIOD0 and PERIOD1 are used to generate a configurable ON/OFF time for PERIOD0 signal.

Always, the first transition of the PERIOD0 signal is determined by the PERIODTIME0_L/H register pair while the next transition is determined by the PERIODTIME1_L/H register pair. This sequence is repeated.

If the initial state of the PERIOD0 signal is configured as '0', the OFF cycle starts first. Therefore in this case, the PERIODTIME0_L/H register pair define the OFF time while the PERIOTIME1_L/H register pair define the ON time.

Similarly, if the initial state of the PERIOD0 signal is configured as '1', it means that the ON cycle starts first. Therefore in this case, the PERIODTIME0_L/H register pair defines the ON time while the PERIODTIME1_L/H register pair defines the OFF time.

The following steps describe the actions needed to generate the PERIOD0 signal with an ON time of 500 us and OFF time of 600 us.

- The syn1588® hardware clock is assumed to be initialized already with the required STEP value. The syn1588® PTP Stack may be used to control the operation of the syn1588® hardware clock.
- The ON time of 500 us has to be written to the PERIODTIME0_L and PERIODTIME0_H registers.
- Write PERIODTIME0_L[31:0] = 0xA1200000
- Write PERIODTIME0_H[31:0] = 0x00000007
- Always write to the lower register first.
- Similarly the OFF time of 600 us has to be written to the PERIODTIME1_L and PERIODTIME1_H registers.
- Write PERIODTIME1_L[31:0] = 0x27C00000
- Write PERIODTIME1_H[31:0] = 0x00000009
- Always write to the lower register first.
- Write EVENTCTRL = 0x00002150, which enables the PERIOD0 function, enables the PERIOD0 output, sets the initial value to '1' and enables the duty cycle function for PERIOD0.

The syn1588 utility commands for starting the syn1588® clock and to writing the above said registers is shown below.

```
>>0x0F0 0xA1200000
>0x0F4 0x00000007
>0x100 0x27C00000
>0x104 0x00000009
>0x04C 0x00002150
```

After writing to the above said registers, the PERIOD0 function starts with an ON cycle as the initial state is configured to '1'. Hence in this case, the time defined by the PERIODTIME0_L/H register pair corresponds to the ON time. After the ON time elapses with an additional uncertainty value between "0 to STEP" value, the OFF cycle starts whose duration is equal to the value defined by PERIODTIME1_L/H register pair plus an uncertainty value between "0 to STEP" value.

Figure 7 illustrates this example. Also, the figure shows the situation when the initial state of the PERIOD0 signal is configured as '0'.

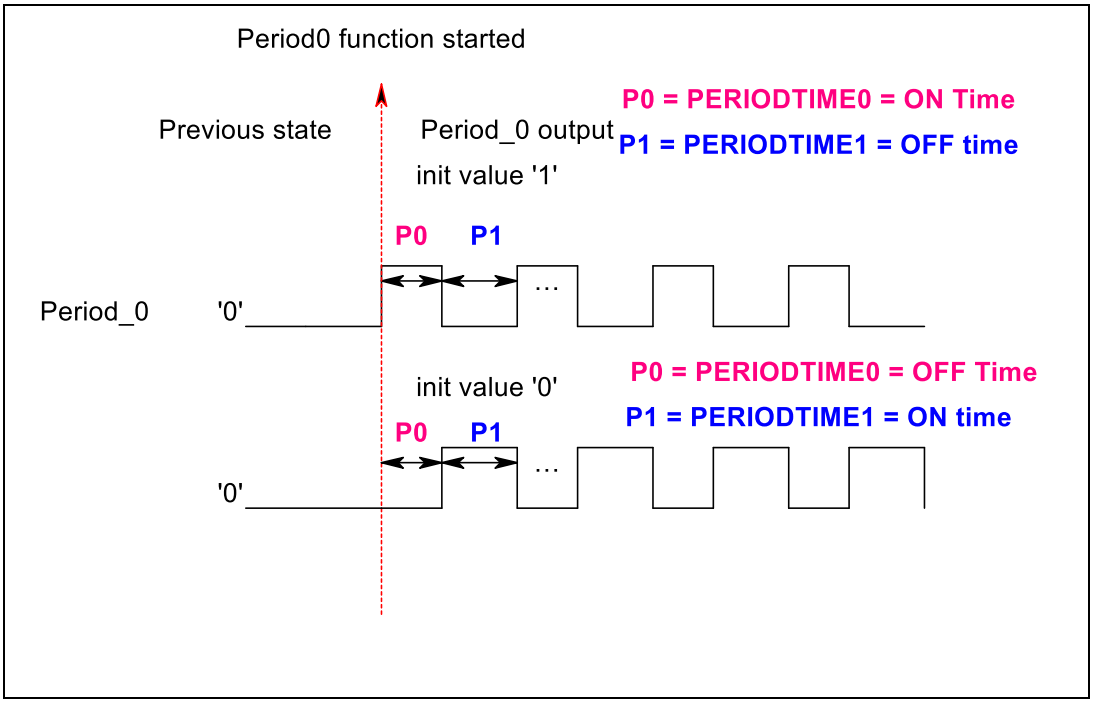

Figure 7 The figure depicts the programmable duty cycle of the PERIOD0 signal. It can be observed that, the time defined by PERIODTIME0_L/H and PERIODTIME1_L/H registers can become ON time or OFF time of the signal depending on the configured initial state of the PERIOD0 signal.

Register Overview

The following is an excerpt of the syn1588® register map described in the application note “an_register_map.pdf”, showing all registers mentioned in this application note.

Address	Name	Function	Mode
0x040	IRSRC	Interrupt source	R/W
0x044	IREN	Enables various interrupts	R/W
0x048	TIMECTRL	Controls operation of the local clock	R/W
0x04C	EVENTCTRL	Control of events	R/W
0x050	SHDWSTEP_L	Preload of new step size, lower 32 bit	R/W
0x054	SHDWSTEP_H	Preload of new step size, upper 32 bit	R/W
0x084	SHDWTIME_NS	Preload of new time of the clock, nanoseconds 32 bit	W
0x088	SHDWTIME_SEC	Preload of new time of the clock, seconds 32 bit	W
0x0A4	EVENTTIME0_L	Time of last event 0, low 32 bit	R
0x0A8	EVENTTIME0_H	Time of last event 0, upper 32 bit	R
0x0B0	EVENTTIME1_L	Time of last event 1, low 32 bit	R
0x0B4	EVENTTIME1_H	Time of last event 1, upper 32 bit	R
0x0B8	FRAC_NUM	Numerator of fractional divider for precision period	R/W
0x0BC	FRAC_DENUM	Denumerator of fractional divider for precision period	R/W
0x0D0	TRIGTIME0_L	Time to trigger event 0, first 32 bit	R/W
0x0D4	TRIGTIME0_H	Time to trigger event 0, upper 32 bit	R/W
0x0D8	TRIGTIME1_L	Time to trigger event 1, first 32 bit	R/W
0x0DC	TRIGTIME1_H	Time to trigger event 1, upper 32 bit	R/W
0x0F0	PERIODTIME0_L	Period of timer 0, lower 32 bit	R/W
0x0F4	PERIODTIME0_H	Period of timer 0, upper 32 bit	R/W
0x100	PERIODTIME1_L	Period of timer 1, lower 32 bit	R/W
0x104	PERIODTIME1_H	Period of timer 1, upper 32 bit	R/W
0x200	IOMATRIX	External IO Interconnection definition (Not located in the syn1588® Clock IP core)	R/W

Version 1.5 – June 2019

Oregano Systems
A Meinberg Company

Franzosengraben 8

A-1030 Vienna

Austria

http://oregano.at

contact@oregano.at