ANLAN203

KSZ84xx GPIO Pin Output Functionality

Introduction

Devices in Micrel's ETHERSYNCH™ family have several GPIO pins that are linked to the internal IEEE 1588 precision time protocol (PTP) clock. These pins can be individually configured for input or output. This application note describes how to use the GPIO pins in output mode. Using the GPIO pins in input mode will be described in a separate application note.

ANLAN203 applies to the following devices in the ETHERSYNCH family:

- KSZ8441
- KSZ8462
- KSZ8463

Overview of GPIO and TOU

Each device has either seven or 12 general purpose input/output (GPIO) pins. Some of these pins are dedicated GPIO pins, while others are multiplexed with other optional functions. Configuration of the multiplexed pins is done in the input and output multiplex selection register (IOMXSEL, at address 0x0D6).

The direction of each GPIO pin – either input or output – is set in the GPIO output enable register (GPIO_OEN, at address 0x682). Upon chip reset, all GPIO pins default to output mode.

Output behavior on the GPIO pins is controlled by 12 trigger output units (TOUs). Each TOU has a single output which may be programmed to drive any one GPIO pin. Each TOU is clocked by the internal 125MHz clock and is capable of generating a variety of output patterns which can be timed to the internal IEEE1588 PTP clock. Output options include individual edges and pulses, repeating pulses, and a shift register for arbitrary bit sequences. Repeating patterns may be configured to repeat a fixed number of times, or repeat infinitely (until stopped).

Cascade mode is an extended feature that allows fixed length output patterns – as described above - from one or more TOUs to be combined sequentially into a grand pattern. This grand pattern can then be repeated a fixed number of times or infinitely.

There is not a fixed relationship between TOUs and GPIO pins. Any TOU may drive any GPIO pin. When setting up each TOU, one of the setup fields defines which GPIO pin is driven by the TOU. A GPIO may be driven by multiple TOUs, but it is not possible for a TOU to drive more than one GPIO. When a GPIO is driven by multiple TOUs, the output of the GPIO pin is the logic OR of the output of the applicable TOUs.

Note that the GPIO pins are numbered from 0 (i.e. 0–6, or 0–11). TOUs are numbered from 1 (i.e. 1–12) in the datasheets, while in the driver and PTP_CLI utility they are numbered from 0 (i.e. 0–11).

Micrel Inc. • 2180 Fortune Drive • San Jose, CA 95131 • USA • tel +1 (408) 944-0800 • fax + 1 (408) 474-1000 • http://www.micrel.com

Output Signal Patterns (Modes)

Each TOU is programmable for any of seven different output signal patterns: positive and negative edge, positive and negative pulse, positive and negative cycle (i.e. repeating pulse), and bit shift register mode (see Figure 1). Edge and pulse modes produce single events that do not repeat. Cycle and register modes produce repeating events; the number of repetitions and the repetition period are programmable for each TOU.

After any TOU output pattern terminates, the output stays at the final level until a new event occurs, or until the TOU is reset. When a TOU is reset, its output goes low. When multiple TOUs drive one GPIO pin, the GPIO output is the logic OR of the applicable TOU outputs. Since a high TOU output will force the GPIO pin high, regardless of the state of the other TOUs, care should be taken to avoid unintentionally getting into this state when multiple TOUs are configured to drive the same GPIO.

Register mode is somewhat different from the other modes. It uses a 16-bit shift register (TRIGn_CFG_6) which can be programmed with an arbitrary bit pattern. This pattern is shifted out LSB first at a programmed rate (bit width) and for a programmed number of bits. If the number of output bits is specified as greater than 16, then the 16-bit pattern will repeat.

The bit width (register mode) or cycle width (cycle modes) is programmed in the TRIGn_CFG_3 & 4 registers, with a resolution of 1ns. The average bit width or cycle width will conform to the programmed value, but because the TOUs are clocked at 125MHz, the actual width of any individual bit or cycle is a multiple of 8ns rather than 1ns. Therefore, if the programmed bit/cycle width is not a multiple of 8ns, some bits/cycles will be wider than average, and others will be narrower. In contrast, the pulse width variable is defined as a multiple of 8ns, so it is not prone to this type of variation.

It should also be noted that when a TOU is enabled, after its configuration is programmed, the TOU output will immediately switch to the logic level that is appropriate for the programmed mode. Specifically, if the mode is POS_EDGE, POS_PULSE, POS_CYCLE or REG_OUTPUT, the TOU output will go low. And if the mode is NEG_EDGE, NEG_PULSE or NEG_CYCLE, the TOU output will go high. In this way, the output pattern always starts from a known state. However, it means that there may be an additional output transition (prior to the beginning of the output sequence) that the user should be aware of. This will be further illustrated in some of the examples.

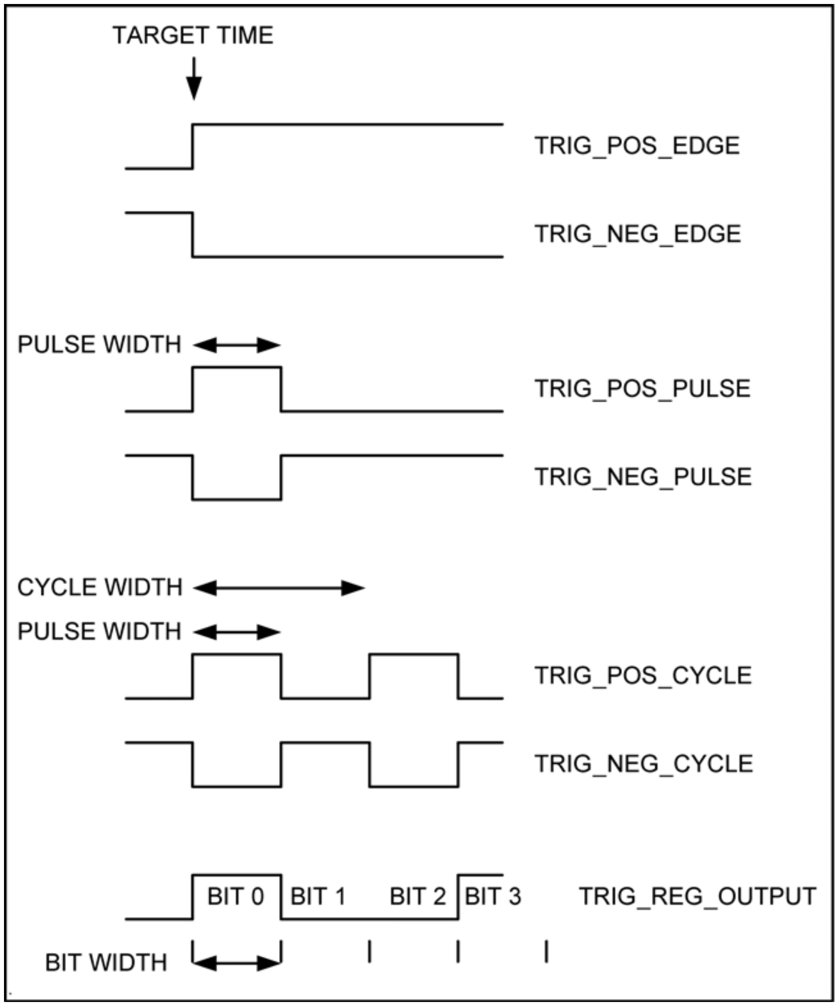

Figure 1. Output Signal Patterns (Modes)

Trigger Timing

An important aspect of TOUs is integration with the PTP 1588 clock. The PTP 1588 clock is a set of registers that hold the current time, and which are updated every 40ns by the local 25MHz clock. The PTP 1588 clock may run at exactly the frequency of the local 25MHz clock, as a free-running clock. Or the PTP 1588 clock can be locked to an external time reference, such as IEEE 1588, by adjusting the update amount to be slightly less than or slightly greater than 40ns.

Output patterns are initiated at a programmed time, an absolute time that is referenced to the PTP 1588 clock. This is known as the trigger target time, and is programmed in the TRIGn_TGT registers of each TOU. For repeating signal patterns, a cycle width or bit width must also be programmed. While the initial trigger time is defined by writing to the TRIGn_TGT registers, subsequent trigger times for repeating patterns are automatically generated by adding the cycle width (bit width) to the last trigger time. Each event will occur (i.e., it will be "triggered") when the PTP 1588 clock advances to the trigger time.

Trigger times and cycle width (bit width) times are defined to 1ns resolution, but TOUs are clocked by a 125MHz clock. Clocking the logic at 125MHz means that the output edges have a resolution of 8ns. This means that while the overall timing is calculated down to the nanosecond, and does not degrade or drift over time, individual output edges will tend to have jitter. Jitter occurs because of the resolution of the 125MHz clock, and because of jitter in the PTP 1588 clock, which occurs if any of the PTP 1588 clock adjustment features is on. The only time that the GPIO output signals will not have jitter is when the cycle width is a multiple of 8ns, and no adjustments are being made to the PTP 1588 clock.

As mentioned above, a TOU output event is triggered when the PTP 1588 clock time passes the trigger time. This works, provided that the PTP 1588 clock is running without interruption. "Without interruption" means that the clock is on, and the step adjustment and direct time setting modes are not being used (see PTP_CLK_CTL register (0x600)).

When step adjustment mode or direct time setting mode is used, the resulting discontinuities in the PTP 1588 clock time may cause the TOU trigger to be missed, which will cause the TOU to hang up and prevent further triggering. This error state may be detected by monitoring the TOU_ERR and TOU_DONE registers. Following a step adjustment or direct time setting, TOU functionality can be corrected or ensured by disabling the TOU, programming a new trigger target time, and re-enabling the TOU. The Micrel driver automatically does this for the 1 PPS output (on GPIO6) whenever the PTP software adjusts the PTP clock. In general, do not use or rely upon GPIO outputs while step adjustment or direct time setting modes are in use.

Note that this caution does not apply to continuous adjustment mode (register PTP_CLK_CTL) or to temporary adjustment mode (register PTP_SNS_RATE_H). They produce clock adjustments that are not large enough to have any impact on TOU triggering.

Setting and Executing Trigger Times

The general procedure is to setup the TOU, including a trigger time in the TRIGn_TGT registers, and then enable the TOU. The trigger time is typically calculated by first reading the current time from the PTP 1588 clock and then calculating a suitable future trigger target time. This is shown in Figure 2. The target time in the TRIGn_TGT registers is the initiation time for the first event. If the event is repeating (i.e., pulse or register mode), then the hardware will automatically calculate the trigger times for the following events by adding the cycle width or bit width value in register TRIGn_CFG_3 and 4.

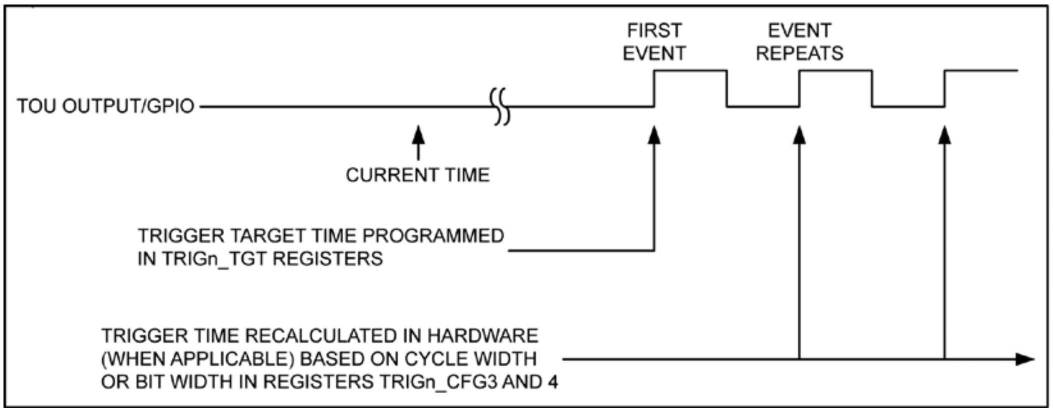

Figure 2. Standard Use of Trigger Target Time

By programming a trigger target time, the start time and the phase relationship of each TOU output can be carefully controlled. However, if this start time precision is not required, there is a simpler way to initiate a TOU output. In the TRIGn_CFG_1 register for each TOU is a TRIG_NOW bit. By enabling this bit, the TOU will be triggered (more or less) immediately once the TOU is enabled, rather than waiting for a match of the trigger target time and the PTP 1588 clock time. The TRIG_NOW feature works only if the target time in the TRIGn_TGT registers is in the past, relative to the PTP 1588 clock. If the target time is in the future, then the TRIG_NOW bit is ignored, and the event will start when the target time is reached. Trigger timing using the Trigger Now feature is shown in Figure 3.

Several of the examples later in this document utilize the TRIG_NOW feature because it makes the TOU setup simpler by eliminating the requirement to read the PTP 1588 clock and calculate and program a target time.

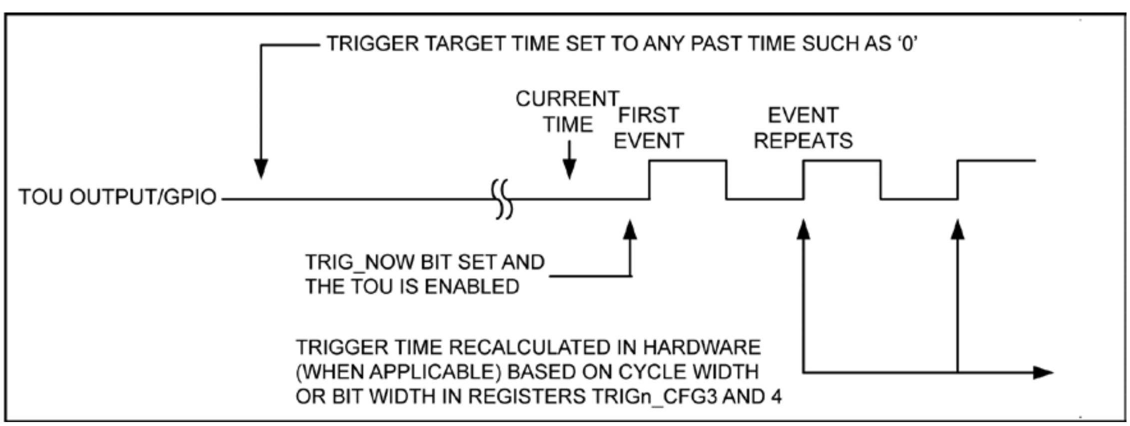

Figure 3. TRIG_NOW Event Timing

Terminating TOU Output Sequences

The repeating signal pattern modes (cycle and register modes) are programmable for repetition. If a non-zero value is programmed into register TRIG_CFG_5, then the cycle or bit sequence will repeat for that number of cycles or bits, and then stop. If the repeat value is ‘0’, then output pattern will repeat infinitely. To stop an infinitely repeating pattern, either disable the TOU (in the TRIG_EN register) or reset the TOU (in the TRIG_SW_RST register). If the TOU is reset, all CONFIG registers and the Target Time registers are reset to their default values, and the TOU output goes low. If the TOU is disabled via the TRIG_EN register, then the register settings will be maintained and the TOU output will remain in its last state. The TOU can then be restarted by programming a new (future) trigger time and then enabling the TOU.

Registers

The TOUs have a few status registers and many control registers. In particular, each TOU has 12 configuration registers, which include the trigger target time. Table 1 summarizes how the TRIGn_CFG registers are used for the various output pattern modes. Table 2 summarizes the key control bits in the TRIGn_CFG_1 register.

One additional register (TRIG12_PPS_WIDTH) is used for configurations that are unique to TOU1 and TOU12.

Table 1. Summary of TRIGn_CFG Registers

Configuration Registers	**Output Pattern Mode – Specified in TRIGn_CFG_1 Register Bits[6:4]**			
	TRIG_NEG_EDGE TRIG_POS_EDGE “Edge Mode”	**TRIG_NEG_PULSE TRIG_POS_PULSE “Pulse Mode”**	**TRIG_NEG_CYCLE TRIG_POS_CYCLE “Cycle Mode”**	**TRIG_REG_OUTPUT “Register Mode”**
TRIGn_CFG_2	N/A	Pulse Width *8ns	Pulse Width *8ns	Iteration Count (Cascade Mode Only)
TRIGn_CFG_3 & 4	N/A	N/A	Cycle Width *1ns	Bit Width *1ns
TRIGn_CFG_5	N/A	N/A	Cycle Count (0 = Infinite)	Bit Count (0 = Infinite)
TRIGn_CFG_6	Iteration Count (Cascade Mode Only)			16-Bit Pattern
TRIGn_CFG_7 & 8	Iteration Cycle Time (Cascade Mode Only)			

Table 2. Description of TRIGn_CFG_1 Register

Bit	Field Name	Usage Description
15	Enable Cascade Mode	When cascading TOUs, set this bit in each TOU. When not cascaded, clear this bit.
14	Tail Unit Indication (Cascade Mode)	Set this bit in the last (tail) TOU in the cascade sequence (for finite repetition). Do not set this bit in other cascaded TOUs, or when the TOU is not cascaded.
13:10	Upstream Trigger Unit (Cascade Mode)	When in cascade mode, specify the upstream (preceding) TOU. The first TOU in the sequence should specify the last (tail) unit. Values 0000 to 1011 correspond to TOU1 to TOU12. When not cascaded, set this field to 1111.
9	Trigger Now	The Trigger Now feature, which “triggers” the TOU immediately upon enabling it, functions only when the TOU is enabled without a future target time specified. If a future time value is written to the target time registers, the trigger will occur at the specified target time and this bit is a don’t care. If in doubt, it is safe to always set this bit.
8	Trigger Notify	Enables reporting TRIG_DONE and TRIG_ERR, as well as interrupt – if the interrupt is otherwise enabled.
7	Clock Edge Output Select (TOU2 only)	Allows TOU2 to be clocked with the falling edge of the internal 125MHz instead of the usual rising edge. This shifts the timing of the TOU2 output by 4ns, which may be useful in some cases, as demonstrated in example #6. Applies only to TOU2 in combination with GPIO1.
6:4	Output Signal Pattern / Mode	Select the TOU output pattern “mode”: edge, pulse, cycle or register. More details are given later. The mode determines which other configuration registers are used.
3:0	GPIO	Select the GPIO that is driven by this TOU. Values 0000 to 1011 correspond to GPIO0 to GPIO11.

Table 3 and Table 4 list additional registers that are used for the GPIO, but are not dedicated to individual TOUs.

Table 3. Shared Status and Control Registers (1 Bit per TOU)

Register Address	Register Name	Description
0x200 – 0x201	TRIG_ERR	Read Only. Trigger Output Unit Error.
0x202 – 0x203	TRIG_ACTIVE	Read Only. Trigger Output Unit Active.
0x204 – 0x205	TRIG_DONE	Read Only / Write Once to Clear. Trigger Output Unit Event Done.
0x206 – 0x207	TRIG_EN	Read / Write. Trigger Output Unit Enable.
0x208 – 0x209	TRIG_SW_RST	Read / Write. Trigger Output Software Reset.

Table 4. Extraneous GPIO and TOU Registers

Register Address	Register Name	Description
0x0D6 – 0x0D7	Input and Output Multiplex Selection Register (IOMXSEL)	Select Whether Certain Pins are GPIO or Another Function.
0x190 – 0x191 bit[10]	Interrupt Enable Register (IER)	Top Level IER for the Chip.
0x192 – 0x193 bit[10]	Interrupt Status Register (ISR)	Top Level ISR for the Chip.
0x682 – 0x683	GPIO Output Enable Register (GPIO_OEN)	Set Each GPIO Pin to Either Input or Output.
0x688 – 0x689	PTP Trigger Unit Interrupt Status Register (PTP_TRIG_IS)	Interrupt Status Only for TOUS.
0x68A – 0x68B	PTP Trigger Unit Interrupt Enable Register (PTP_TRIG_IE)	Interrupt Enable Only for TOUS.

Special Case: TOU12 Pulse Width (Register 0x20A, Bits [7:0])

Most of the TOUs have 16-bit Pulse Width value (register TRIGn_CFG_2). The maximum pulse width is therefore 8ns × 2^16 = 524.28µs). TOU12 is unique – it has eight additional bits (in register TRIG12_PPS_WIDTH) for a total Pulse Width value of 24-bit. This gives a maximum pulse width for TOU12 of 134.21772ms.

Special Case: Falling Edge of TOU2 / GPIO1 (Register 0x248, Bit [7])

TOU2 has a special control bit that is not present in the other TOUs. It is located in register TRIG2_CFG_1, and it determines whether TOU2 is clocked by the rising edge of the internal 125MHz clock, or the falling edge. All other TOUs are clocked by the rising clock edge. Selecting the falling clock edge effectively shifts the TOU2 by 4ns relative to the rising edge setting. This feature works only in conjunction with GPIO1.

This feature may be useful in fine tuning the output timing between GPIO1 and other GPIOs, or between TOU2 and another TOU that are both driving GPIO1. Example #6 utilizes this feature.

TOU Configuration Process

The key aspect of TOU configuration is to properly manage the trigger timing. The following high-level steps ensure correct operation of an individual TOU. Also see the specific examples, which provide full programming details for a variety of configurations.

1. Disable or reset the TOU before configuring it. Resetting the TOU (register 0x208 – 0x20A) is preferred because it resets all of the configuration registers for the TOU and all of the internal TOU logic. This is the most robust method. Alternatively, disabling the TOU (register 0x206 – 0x207) will disable the trigger, but does not reset any of the configuration register. Therefore, it is important that all configuration registers be set correctly.
2. Set the target time (TRIGn_TGT) and configuration (TRIGn_CFG) registers for the TOU. To correctly set the target time, it is usually necessary to first know the present time, by reading the PTP 1588 clock. The relevant PTP clock registers begin at address 0x600.
3. Enable the TOU by writing to the trigger enable register (register 0x206 – 0x207).

When multiple TOUs are cascaded or otherwise combined to drive a single GPIO, all of the relevant TOUs can be enabled (Step #3) together via a single write to the trigger enable register.

Configuring GPIO using the PTP_CLI Utility and Micrel Drivers

This document describes the configuration and use of the GPIO feature from a register level standpoint. Note that Micrel provides a device driver for the ETHERSYNCH family which provides convenient hooks for using the GPIOs, without the need to control everything at the individual register level. Driver details and GPIO commands are available in the document *Micrel 1588 PTP Developer Guide*.

Also, the ETHERSYNCH family of devices is available on evaluation boards, which are typically used with the KSZ9692-MII-PTP-EV SoC board. This SoC board comes programmed with Linux. In addition to the device driver mentioned above, the SoC board also includes the PTP_CLI utility. This utility provides a command line interface for users to access the device registers, and it also provides shortcuts for operating the GPIO feature. Command details are provided in the document *Micrel PTP Utilities User Guide*.

Another useful document is *Micrel 1588 PTP Application Notes*. All of these documents can be found at the Micrel web pages for the KSZ8441, KSZ8462 and KSZ8463 devices. Click on the “Eval Boards or Kit” link, and then download the software documents.

Default GPIO Outputs for Micrel Linux PTP Evaluation Software

Evaluation of ETHERSYNCH devices is normally done by mating a KSZ84xx evaluation board to the KSZ9692-MII-PTP-EV board (labeled either SoC Test BOARD or SoC 2-MII BOARD). The SoC board is pre-loaded with Linux, a PTP software stack, and the utility PTP_CLI. When booting, the SoC board configures the following two clocks on GPIO outputs of the attached KSZ84xx board. It also enables all TOU interrupts:

- GPIO6: 1 pulse per second (1 PPS), utilizing TOU12
- GPIO2: 10MHz clock, utilizing TOU2 and TOU11

Examples

These examples focus on how to correctly control and set up the TOUs in order to generate the desired output signals. For the sake of simplicity, Trigger Notify and interrupts are not utilized.

Some of the register settings must be done in a specific order, while others do not need to be. The general rule is to first reset (or disable) the TOU, then configure it, and finally enable it. The trigger target time and CFG registers can generally be configured in any order.

Note that these examples involve some simplifications, so some adaptations will be needed when applying them to real applications.

- The last step of each example is to enable the TOU(s) that are in use for that specific example. All other enable bits in register 0x206 are set to '0'. If a real application has other active GPIO output signals, then clearing the enable bits of any active TOUs would cause them to stop.
- Many of the examples set up trigger target times instead of utilizing the Trigger Now feature. The host processor will need to first read the current PTP time (registers 0x608 and 0x60A), and then calculate an appropriate trigger time – which must be in the future.
- Unless otherwise stated, the selections of TOU and GPIO are arbitrary.

Example #1

Figure 4 is a simple example of a single positive pulse. The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_POS_PULSE, the TOU output goes low when the TOU is enabled, in preparation for the positive pulse.

The target time seconds field must be set to a future time relative to the current PTP 1588 clock time. As in many of the examples, the target time nanoseconds field is arbitrarily chosen to be zero.

- Pulse width = 280ns
- Start at future target time (with target time nanoseconds = 0)
- GPIO0, TOU1

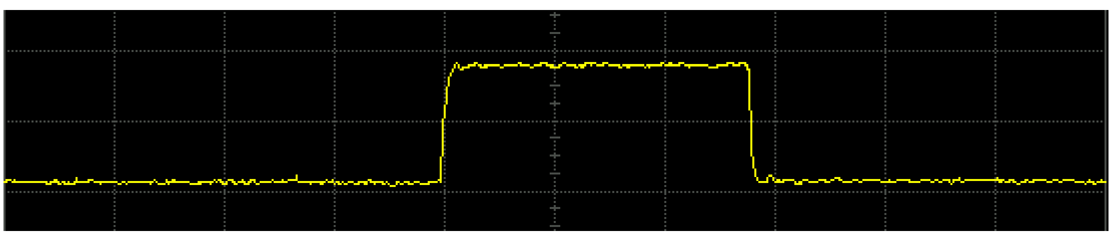

Figure 4. Single Positive Pulse (with a Target Time)

Table 5. Register Settings for Single Positive Pulse from TOU1 to GPIO0

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C30	General Configuration: No Cascading or Trigger Now, TRIG_POS_PULSE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #2

Figure 5 is a simple example of a single negative pulse. The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_NEG_PULSE, the TOU output goes high when the TOU is enabled, in preparation for the negative pulse. After a negative pulse, the TOU output stays high.

Note that this example uses the same TOU as Example #1, but drives a different GPIO pin. This is an arbitrary choice.

- Period = 280ns
- Start at future target time (with target time nanoseconds = 0)
- GPIO2, TOU1

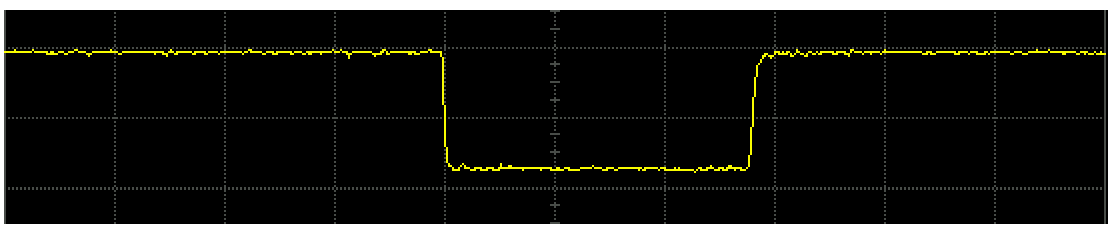

Figure 5. Single Negative Pulse (with a Target Time)

Table 6. Register Settings for Single Negative Pulse from TOU1 to GPIO2

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C22	General Configuration: No Cascading or Trigger Now, TRIG_NEG_PULSE, GPIO2
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #3

The positive cycle mode is nothing more than a repeating version of the positive pulse mode. In addition to the pulse width, a cycle width and a number of repetitions must be specified. A cycle count value of zero indicates infinite repetition.

For a 50% duty cycle, the pulse width would be 500ns. However, the pulse width cannot be configured to be 500ns because it is not an integer multiple of 8ns. The options closest to 500ns are 496ns and 504ns.

The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_POS_CYCLE, the TOU output goes low when the TOU is enabled, in preparation for the positive cycles.

- Period = 1µs
- Duty cycle = (close to) 50%
- Start at future target time (with target time nanoseconds = 0)
- Repeat infinitely
- GPIO0, TOU1

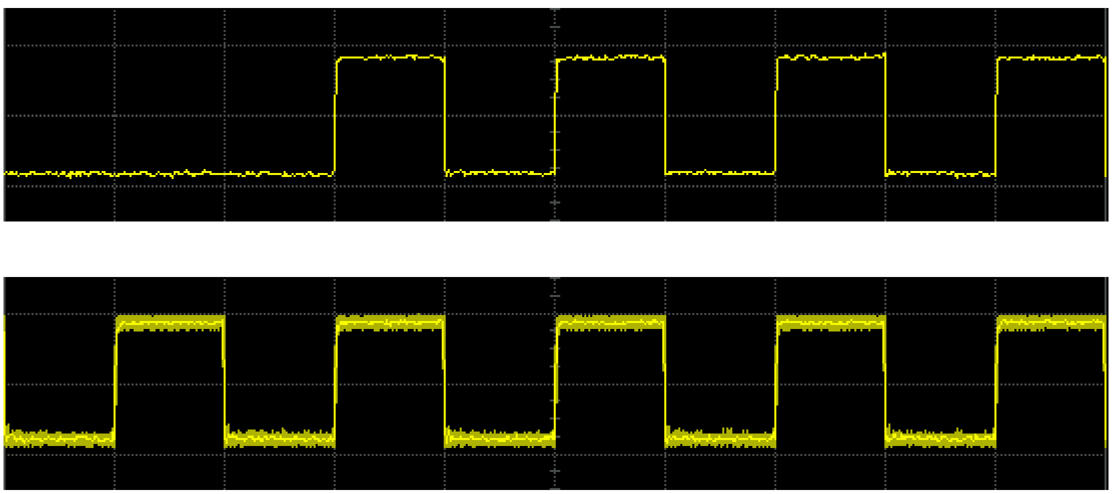

Figure 6. 1MHz Clock (with a Target Time)

Table 7. Register Settings for Positive Cycle from TOU1 to GPIO0

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C50	General Configuration: No Cascading or Trigger Now, TRIG_POS_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x003E	Pulse Width = 8ns × 0x3E = 496ns
0x22C (TRIG1_CFG_3)	0x03E8	Cycle Width (Period) = 1ns × 0x3E8 = 1000ns
0x230 (TRIG1_CFG_5)	0x0000	Cycle Count = 0 = Infinite Repetition. (Default = 0)
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #4

Example #4 is very similar to Example #3, but introduces a special register.

For the pulse and cycle modes, the maximum pulse width is 524µs, based on the 16-bit value in register TRIGn_CFG_2. However, TOU12 is unique. It has a pulse width field of 24 bits, which allows for a maximum pulse width of 134ms. The additional 8 bits are located in register TRIG12_PPS_WIDTH (0x20A). To achieve a 20ms pulse width, this example must use TOU12.

- Period = 1 second
- Positive pulse width = 20ms (duty cycle = 2%)
- Start at future target time (with target time nanoseconds = 0)
- GPIO6, TOU12

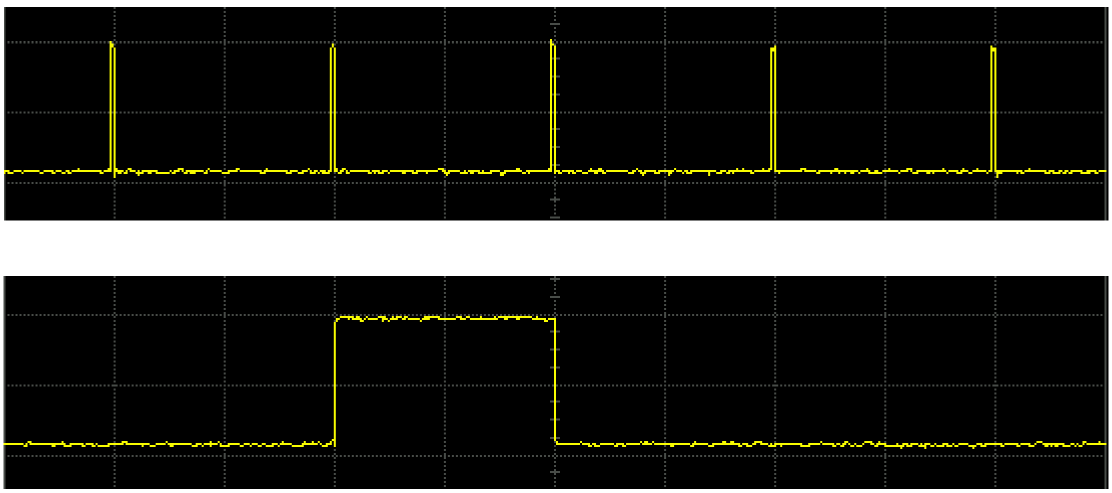

Figure 7. 1 Pulse per Second (PPS) (with a Target Time)

Table 8. Register Settings for Positive Cycle with Long Pulse Width from TOU12

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0800	Reset Trigger Output Unit 12
0x380 (TRIG12_TGT_NSL) 0x382 (TRIG12_TGT_NSH)	0x0000 0x0000	Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x384 (TRIG12_TGT_SL) 0x386 (TRIG12_TGT_SH)	0xSSSS 0xSSSS	Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x388 (TRIG12_CFG_1)	0x3C56	General Configuration: No Cascading or Trigger Now. Positive Cycle, GPIO6.
0x38A (TRIG12_CFG_2) 0x20A (TRIG12_PPS_WIDTH)	0x25A0 0x0026	Pulse Width = 8ns × 0x2625A0 = 20ms
0x38C (TRIG12_CFG_3) 0x38E (TRIG12_CFG_4)	0xCA00 0x3B9A	Cycle Width (Period) = 1ns × 0x3B9ACA00 = 1s
0x206 (TRIG_EN)	0x0800	Enable Trigger Output Unit 12

Example #5

Example #5 shows that it is relatively easy to set up a 10MHz output. However, it highlights the issue that many clock frequencies have periods that are not multiples of 8ns. In this example, the ideal clock period (cycle width) is 100ns, but the TOUs can only produce individual cycles of 96ns and 104ns. The cycle width is programmed to be 100ns, which will cause the TOU to produce alternating 96ns and 104ns cycles. In spite of the variation in cycle widths, the individual trigger times are calculated at 100ns intervals, which results in an average frequency of exactly 10MHz relative to the PTP 1588 clock. Note that the high pulse width is fixed at 48ns and does not vary; it's only the inter-pulse (low) time that varies.

For variety, this example uses a different TOU and GPIO pin.

- Period = 100ns
- Duty cycle = (close to) 50%
- Start at future target time (with target time nanoseconds = 0)
- GPIO2, TOU8

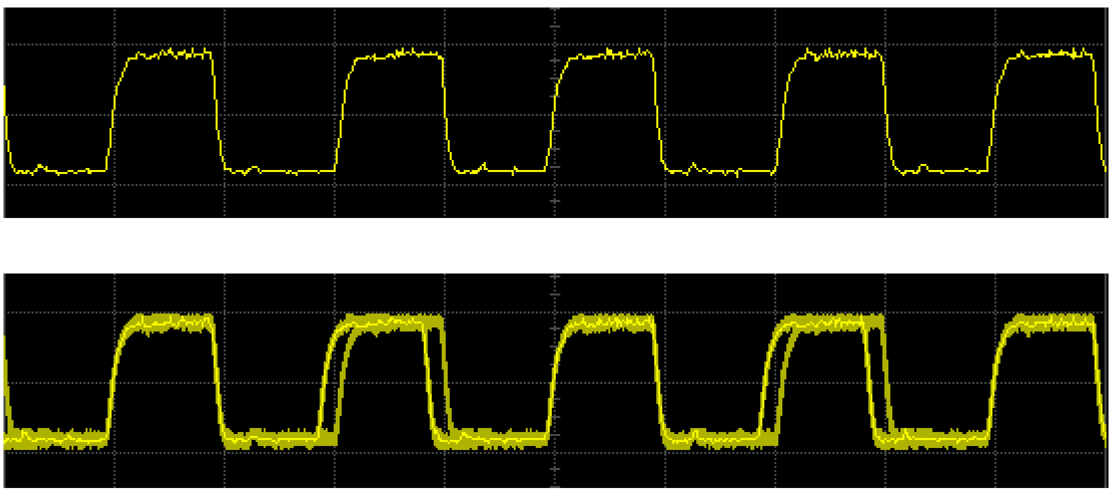

Figure 8. 10MHz Clock (with a Target Time)

Table 9. Register Settings for 10MHz Positive Cycle from TOU8 to GPIO2

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0080	Reset Trigger Output Unit 8
0x300 (TRIG8_TGT_NSL) 0x302 (TRIG8_TGT_NSH)	0x0000 0x0000	Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x304 (TRIG8_TGT_SL) 0x306 (TRIG8_TGT_SH)	0xSSSS 0xSSSS	Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x308 (TRIG8_CFG_1)	0x3C52	General Configuration: No Cascading or Trigger Now, TRIG_POS_CYCLE, GPIO2
0x30A (TRIG8_CFG_2)	0x0006	Pulse Width = 8ns × 0x6 = 48ns
0x30C (TRIG8_CFG_3)	0x0064	Cycle Width (Period) = 1ns × 0x64 = 100ns
0x206 (TRIG_EN)	0x0080	Enable Trigger Output Unit 8

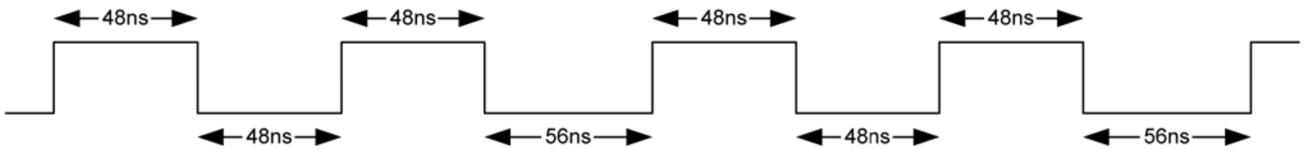

Figure 9. 10MHz: Varying Clock Periods when Cycle Width ≠ Multiple of 8ns

Example #6

Example #6 shows how the TOU2 clock edge output select feature (register TRIG2_CFG_1 (0x248) bit[7]) is used to correct the "jitter" problem seen in the previous example. Note that this configuration must use GPIO1, TOU2, and any additional TOU. The correction involves two steps:

Create a 10MHz clock by combining two 5MHz "clocks" that are offset by a specific amount. Note that each 5MHz clock uses the same 48ns high pulse width as the 10MHz example above, rather than a 50% duty cycle configuration. A specific offset between the two TOU waveforms is achieved by specifying different trigger target times for the two TOUs.

Set the TOU2 clock edge output select bit, to specify "negative edge". This feature delays the TOU2 output by 4ns. This feature is not available on any other TOU. This feature is available only for TOU2, and it works only with GPIO1. With any other GPIO, it has no effect.

- Period = 100ns
- Duty cycle = (close to) 50%
- Start at future target time (with target time nanoseconds = 0)
- GPIO1, TOU1 and TOU2

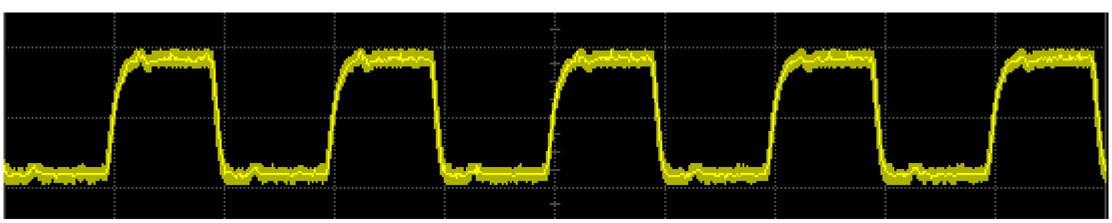

Figure 10. 10MHz Clock using TOU2 Clock Edge Output Select (with a Target Time)

Table 10. Register Settings for 10MHz Positive Cycle Using Two TOUs

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0003	Reset Trigger Output Units 1 and 2
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (The default is 0, so it's not necessary to write 0 after reset)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C51	TOU1 General Configuration: No Cascading or Trigger Now. TRIG_POS_CYCLE, GPIO1
0x22A (TRIG1_CFG_2)	0x0006	TOU1 Pulse Width = 8ns × 0x6 = 48ms
0x22C (TRIG1_CFG_3)	0x00C8	TOU1 Cycle Width (Period) = 1ns × 0x00C8 = 200ns
0x240 (TRIG2_TGT_NSL) 0x242 (TRIG2_TGT_NSH)	0x0060 0x0000	TOU2 Trigger Target Time (Nanoseconds) = 96ns
0x244 (TRIG2_TGT_SL) 0x246 (TRIG2_TGT_SH)	0xSSSS 0xSSSS	TOU2 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x248 (TRIG2_CFG_1)	0x3CD1	TOU2 General Configuration: No Cascading or Trigger Now. Negative edge of 125MHz, TRIG_POS_CYCLE, GPIO1
0x24A (TRIG2_CFG_2)	0x0006	TOU2 Pulse Width = 8ns × 0x6 = 48ms
0x24C (TRIG2_CFG_3)	0x00C8	TOU2 Cycle Width (Period) = 1ns × 0x00C8 = 200ns
0x206 (TRIG_EN)	0x0003	Enable Trigger Output Units 1 and 2

Figure 11 shows how this example utilizes two TOUs, and how the timing between them is set to 100ns by utilizing the negative edge feature of TOU2. When the two TOUs are linked to the same GPIO (specifically GPIO1), they are logically ORed together, producing a "jitter free" 10MHz clock with 48% duty cycle.

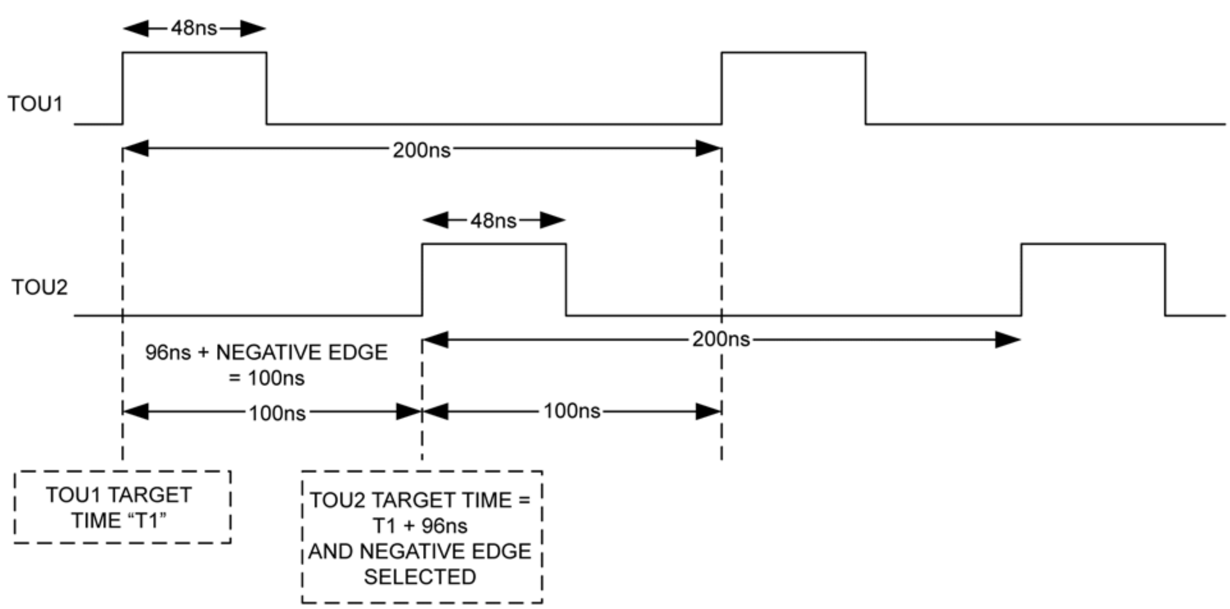

Figure 11. 10MHz: Two TOUs and TOU2 Negative Edge

Example #7

Example #7 illustrates negative cycles. The last step of this command sequence enables TOU4. Because the TOU is programmed for TRIG_NEG_CYCLE, the TOU output goes high when the TOU is enabled, in preparation for the negative pulse.

- Period = 25µs
- Negative pulse = 2µs
- Start at future target time (with target time nanoseconds = 0)
- Repeat infinitely
- GPIO0, TOU4

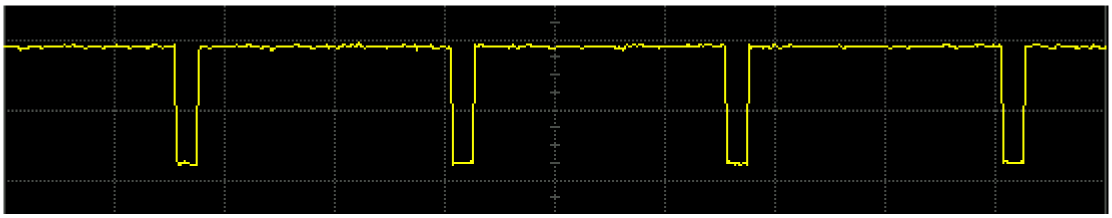

Figure 12. 40kHz Periodic Negative Cycles (with a Target Time)

Table 11. Register Settings for Negative Cycle from TOU4 to GPIO0

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0008	Reset Trigger Output Unit 4
0x280 (TRIG4_TGT_NSL) 0x282 (TRIG4_TGT_NSH)	0x0000 0x0000	Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x284 (TRIG4_TGT_SL) 0x286 (TRIG4_TGT_SH)	0xSSSS 0xSSSS	Trigger Target Time (Seconds) = A Future Time of the PTP 1588 Clock
0x288 (TRIG4_CFG_1)	0x3C40	General Configuration: No Cascading or Trigger Now, TRIG_NEG_CYCLE, GPIO0
0x28A (TRIG4_CFG_2)	0x00FA	Pulse Width = 8ns × 0xFA = 2µs
0x28C (TRIG4_CFG_3)	0x61A8	Cycle Width (Period) = 1ns × 0x61A8 = 25µs
0x290 (TRIG4_CFG_5)	0x0000	Cycle Count = 0 = Infinite Repetition. (Default = 0)
0x206 (TRIG_EN)	0x0008	Enable Trigger Output Unit 4

Example #8

Example #8 is similar to Example #7, but utilizes register TRIGn_CFG_5 to program a fixed cycle count instead of infinity. After a negative cycle, the TOU output stays high.

- Period = 25µs
- Negative pulse = 2µs
- Start at future target time, with 8ms delay after the seconds transition
- 6 cycles
- GPIO0, TOU4

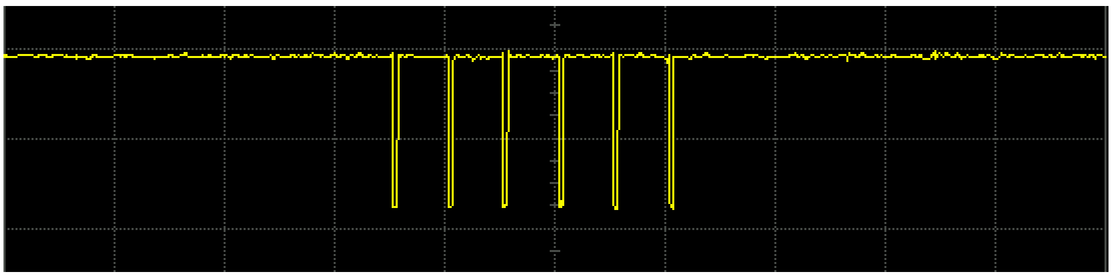

Figure 13. Fixed Number of Negative Cycles (with a Target Time)

Table 12. Register Settings for Six Negative Cycles from TOU4 to GPIO0

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0008	Reset Trigger Output Unit 4
0x280 (TRIG4_TGT_NSL) 0x282 (TRIG4_TGT_NSH)	0x1200 0x007A	Trigger Target Time (Nanoseconds) = 8ms
0x284 (TRIG4_TGT_SL) 0x286 (TRIG4_TGT_SH)	0xSSSS 0xSSSS	Trigger Target Time (Seconds) = A Future Time of the PTP 1588 Clock
0x288 (TRIG4_CFG_1)	0x3C40	General Configuration: No Cascading or Trigger Now, TRIG_NEG_CYCLE, GPIO0
0x28A (TRIG4_CFG_2)	0x00FA	Pulse Width = 8ns × 0xFA = 2µs
0x28C (TRIG4_CFG_3)	0x61A8	Cycle Width (Period) = 1ns × 0x61A8 = 25µs
0x290 (TRIG4_CFG_5)	0x0006	Cycle Count = 6
0x206 (TRIG_EN)	0x0008	Enable Trigger Output Unit 4

Example #9

Example #9 uses the TRIG_REG_OUTPUT mode and repeats the 16-bit pattern seven times. Changing the bit count value to 0 will cause it to cycle infinitely. As described earlier, the actual width of each bit will vary if the defined bit width value is not a multiple of 8ns. However, the average bit width will match the defined value. In this example, the defined bit width is a multiple of 8ns.

The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_REG_OUTPUT, the TOU output goes low when the TOU is enabled, in preparation for the output sequence. This is true, whether the leading output bit (the LSB bit of the pattern register) is a zero or a one. At the end of the output pattern, the TOU output stays at the level of the last bit. It does not automatically return to low.

- Bit pattern 0x0695
- 7 full repetitions of the 16-bit pattern
- Start at future 1588 PTP clock time, with ns field = 0
- GPIO0, TOU1

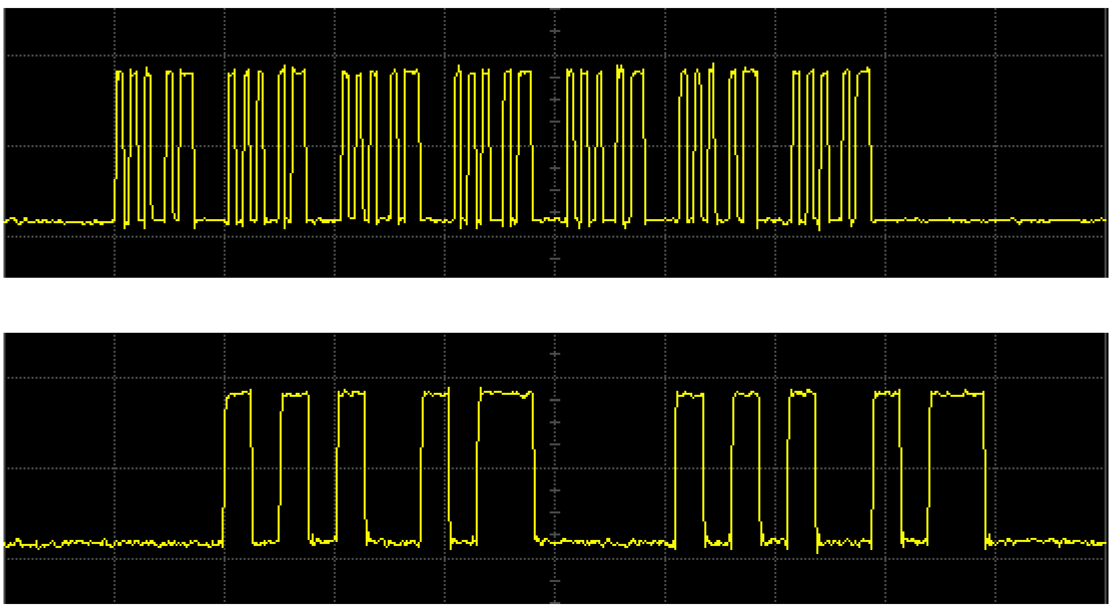

Figure 14. Register Pattern (with a Target Time)

Table 13. Register Settings for Register Pattern from TOU1 to GPIO0

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C60	General Configuration: No Cascading or Trigger Now, TRIG_REG_OUTPUT, GPIO0
0x22C (TRIG1_CFG_3)	0x0080	Bit Width = 1ns × 0x80 = 128ns
0x230 (TRIG1_CFG_5)	0x0070	Bit Count = 0x70 = 7 Loops through the 16-Bit Register
0x232 (TRIG1_CFG_6)	0x0695	The 16-Bit Pattern is 0000_0110_1001_0101
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

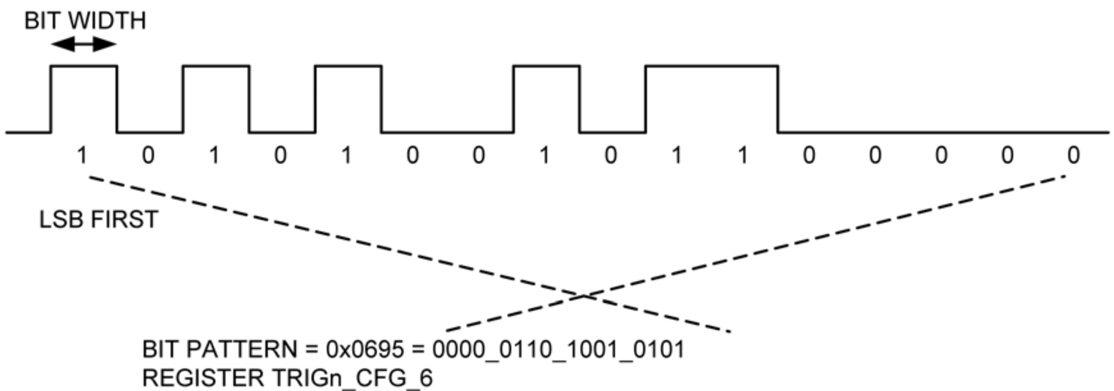

Figure 15. Register Pattern Example

Figure 16 illustrates a variation of what occurs when the bit pattern is changed so that the LSB bit is “0” and the MSB bit is “1” (note that the bit count has also been changed from 0x70 to 0x30).

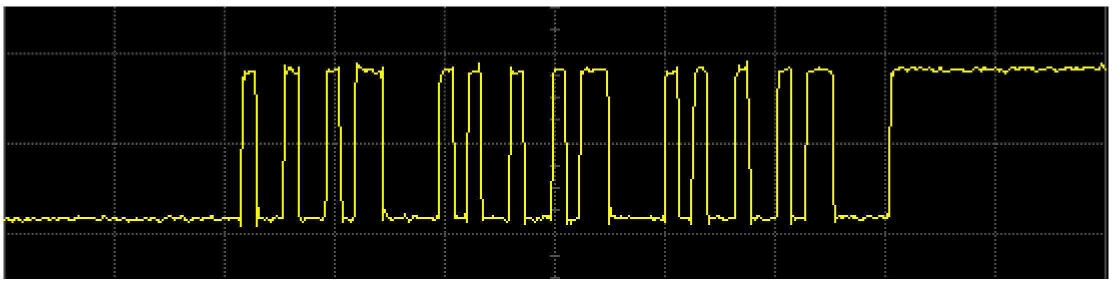

Figure 16. Register Pattern with Final (MSB) Bit = 1

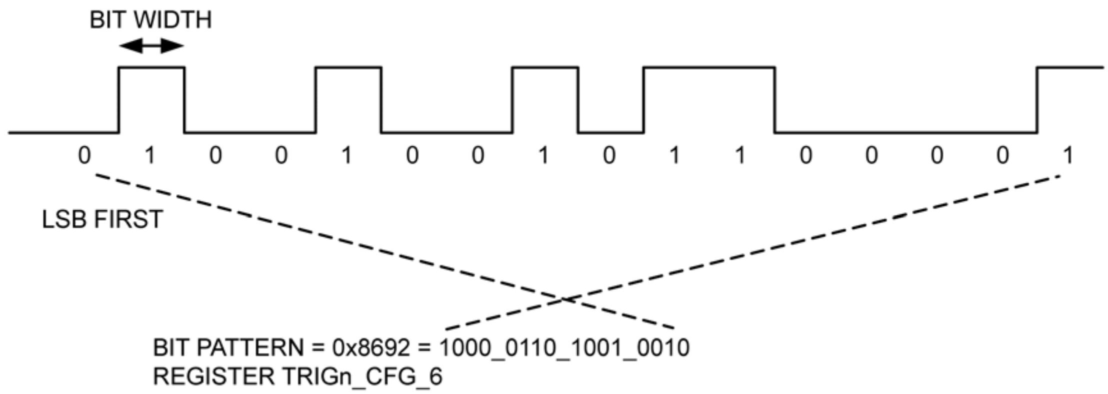

Figure 17. Register Pattern Variation

Example #10

Example #10 shows how to use positive edge mode to set an output high at a specified time. The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_POS_EDGE, the TOU output goes low when the TOU is enabled, in preparation for the positive edge.

- Start at future 1588 PTP Clock time, with ns field = 0
- GPIO1, TOU1

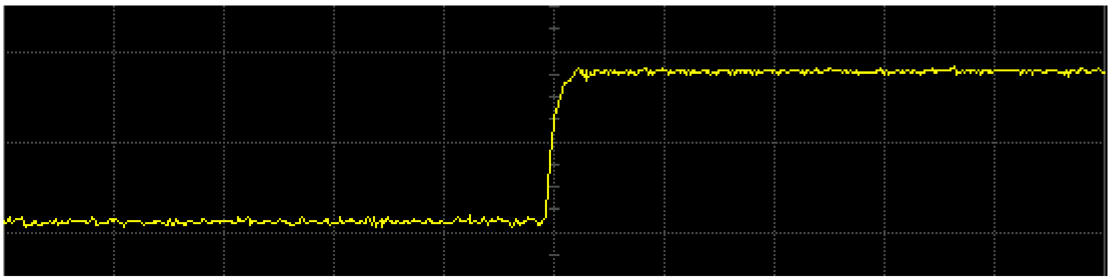

Figure 18. Setting a GPIO High (with a Target Time)

Table 14. Register Settings for Positive Edge from TOU1 to GPIO1

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C11	General Configuration: No cascading or Trigger Now, TRIG_POS_EDGE, GPIO1
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #11

Example #11 shows how to use negative edge mode to set an output low at a specified time. The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_NEG_EDGE, the TOU output goes high when the TOU is enabled, in preparation for the negative edge.

- Start at future 1588 PTP Clock time, with ns field = 0
- GPIO1, TOU1

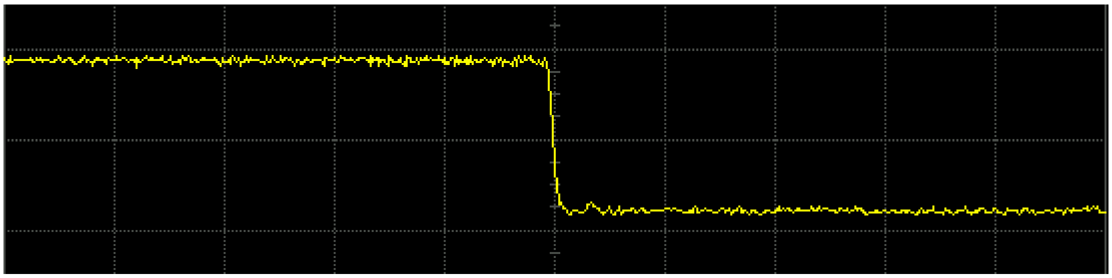

Figure 19. Setting a GPIO Low (with a Target Low)

Table 15. Register Settings for Negative Pulse from TOU1 to GPIO1

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0000 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 0. (Default = 0)
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x3C01	General Configuration: No cascading or Trigger Now, TRIG_NEG_EDGE, GPIO1
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Examples of Immediate Triggering

The following examples show how to initiate an output pattern immediately, by setting the TRIG_NOW bit instead of setting a trigger target time. With this method, the exact starting time of the output event cannot be controlled. Moreover, when using the Trigger Now feature, the TOU output may transition shortly before the output pattern starts.

For TRIG_NOW to function as intended, the Trigger Time registers must hold a past time relative to the current PTP 1588 Clock.

Example #12

This example is the Trigger Now version of Example #10, using the positive edge mode. Figure 20 shows the result when TOU1 is starting from a low state.

- Start immediately
- GPIO1, TOU1

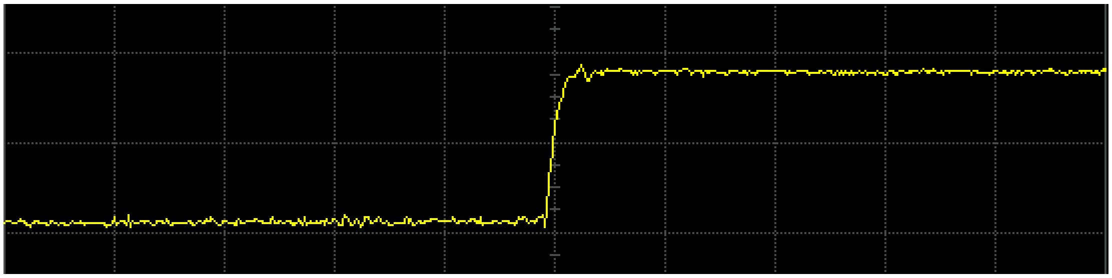

Figure 20. Positive Edge Starting from Low

Figure 21 shows the result when TOU1 is starting from a high state. This occurs because a positive edge must start from a low state, and the TOU output automatically goes low when the TOU is enabled and the mode is TRIG_POS_EDGE. This is the same behavior that occurs when a future Target Time is used as in Example #10. The difference is that Example #10 has a (relatively) long delay between enabling the TOU and the Target Time, whereas this example triggers immediately using TRIG_NOW. The delay between the two edges is 48ns.

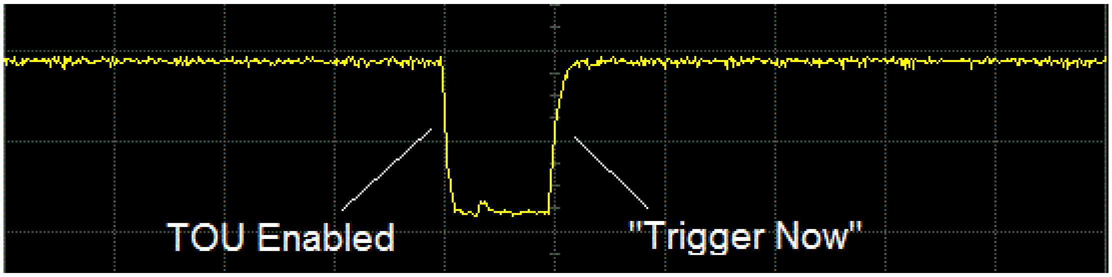

Figure 21. Positive Edge Starting from High (with Trigger Now)

Table 16. Register Settings for Positive Edge with Trigger Now

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E11	General Configuration: No cascading. Trigger Now, TRIG_POS_EDGE, GPIO1
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #13

Example #13 is the Trigger Now version of Example #11, using the negative edge mode. Figure 22 shows the result when TOU1 is starting from a high state.

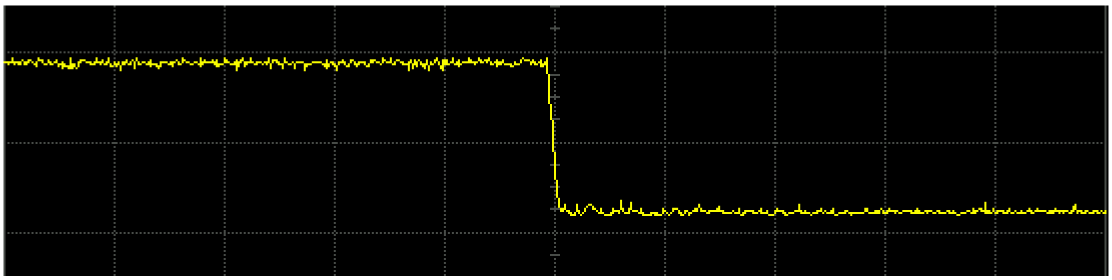

Figure 22. Negative Edge Starting from High

Figure 23 shows the result when TOU1 is starting from a low state. The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_NEG_EDGE, the TOU output goes high when the TOU is enabled, in preparation for the positive cycles. The delay between edges is 48ns.

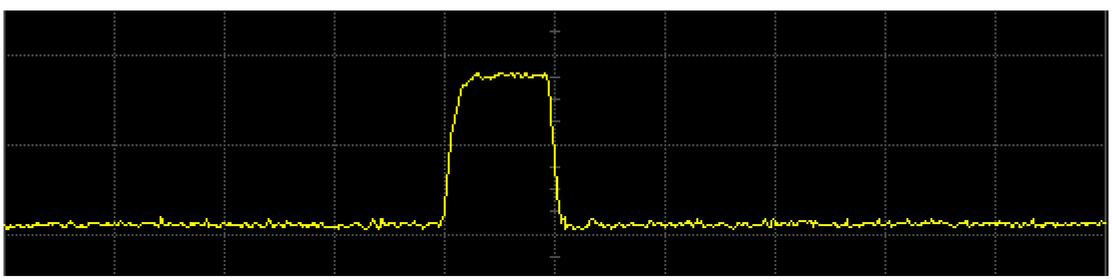

Figure 23. Negative Edge Starting from Low (with Trigger Now)

Table 17. Register Settings for Negative Edge with Trigger Now

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0x0000 0x0000	TOU1 Trigger Target Time (Seconds) = 0
0x228 (TRIG1_CFG_1)	0x3E01	General Configuration: No cascading. Trigger Now, TRIG_NEG_EDGE, GPIO1
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

An alternate method for setting a GPIO output low is to reset the TOU that is already driving the GPIO high. This has the advantage that if the TOU is already low, no pulse is generated as shown above; the output simply stays low.

Table 18. Register Setting for Resetting a TOU

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1

Example #14

A single positive pulse is generated immediately using the Trigger Now feature. The first step of this command sequence resets TOU1, causing the TOU output to go low (if it is not already low) prior to the positive pulse.

- Pulse width = 280ns
- Start immediately
- GPIO0, TOU1

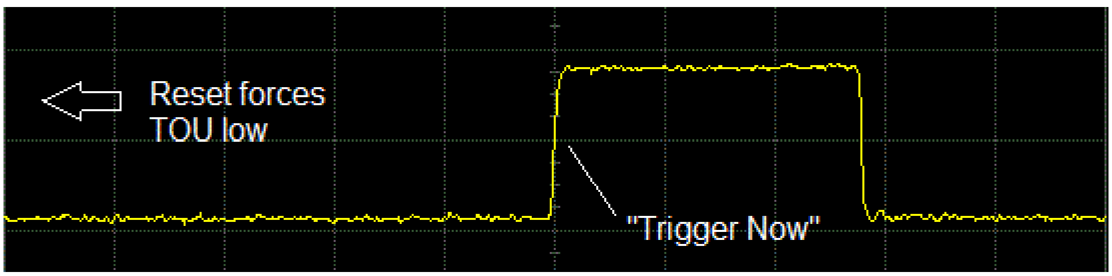

Figure 24. Positive Pulse Starting from Low (with Trigger Now)

Table 19. Register Settings for a Positive Pulse with TOU1 Reset and Trigger Now

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E30	General Configuration: No cascading. Trigger Now, TRIG_POS_PULSE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Figure 25 is a variation on this example, in which the reset step is replaced with a step that disables TOU1. Since the TOU is not reset, the values of all of the TOU registers (0x220 – 0x237) are not reset, and it is necessary to be mindful of their state. Just as importantly, the TOU output stays at its last level, rather than being forced low. This has the following effect on output timing.

The last step of this command sequence enables TOU1. Because the TOU is programmed for TRIG_POS_PULSE, the TOU output goes low when the TOU is enabled, in preparation for the positive pulse. If TOU1 was starting from a low state, the result is as shown in the picture above. If TOU1 was starting from a high state, the result is shown below. There was no reset command, so the negative edge occurs 48ns before the positive pulse.

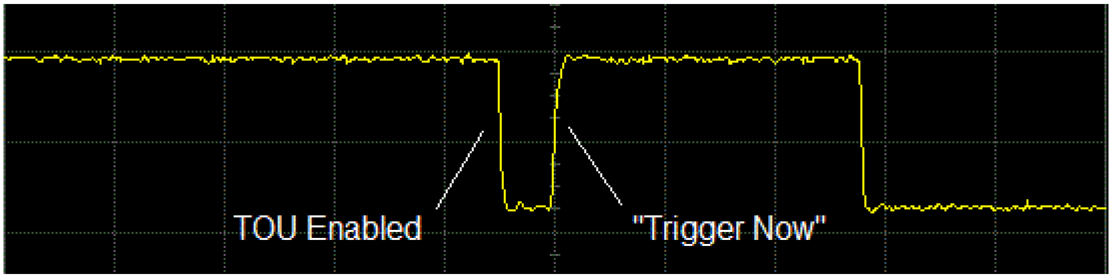

Figure 25. Positive Pulse Starting from High (TOU not Reset)

Table 20. Register Settings for Positive Pulse without TOU Reset

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E30	General Configuration: No cascading. Trigger Now, TRIG_POS_PULSE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #15

A single negative pulse is generated immediately using the Trigger Now feature. The first step of this command sequence resets the TOU, causing its output to go low. The last step enables the TOU. Because the TOU is programmed for TRIG_NEG_PULSE, the TOU output goes high when the TOU is enabled, in preparation for the negative pulse. Because the Trigger Now feature is used, there is 48ns delay between this positive edge and the negative pulse.

- Period = 280ns
- Start immediately
- GPIO0, TOU1

Figure 26. Negative Pulse Starting from Low (with Trigger Now)

Table 21. Register Settings for a Negative Pulse with TOU1 Reset and Trigger Now

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E20	General Configuration: No cascading. Trigger Now, TRIG_NEG_PULSE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Figure 27 is a variation on this example, in which the reset step is replaced with a step that disables TOU1. Since the TOU is not reset, the values of all of the TOU registers (0x220 – 0x237) are not reset, and it is necessary to be mindful of their state. Just as importantly, the TOU output stays at its last level, rather than being forced low. This has the following effect on output timing.

If the TOU output was starting from a low state, the result is the same as shown in the picture above. However, if the TOU output was starting from a high state, the extra transitions are eliminated, as shown in Figure 27. This is the effect of removing the reset command.

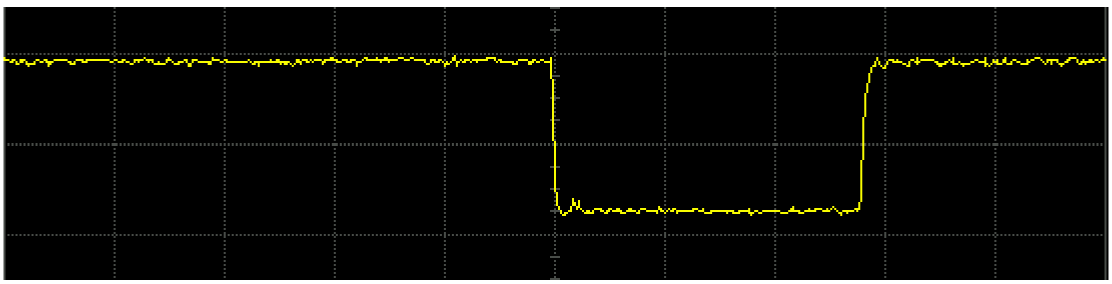

Figure 27. Negative Pulse Starting from High (TOU not Reset)

Table 22. Register Settings for Negative Pulse without TOU Reset

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E20	General Configuration: No cascading. Trigger Now, TRIG_NEG_PULSE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #16

A sequence of positive cycles is generated immediately using the Trigger Now feature. The first step resets the TOU, causing its output to go low prior to the output pattern.

- Period = 488ns
- Pulse width = 280ns
- 5 cycles
- Start immediately
- GPIO0, TOU1

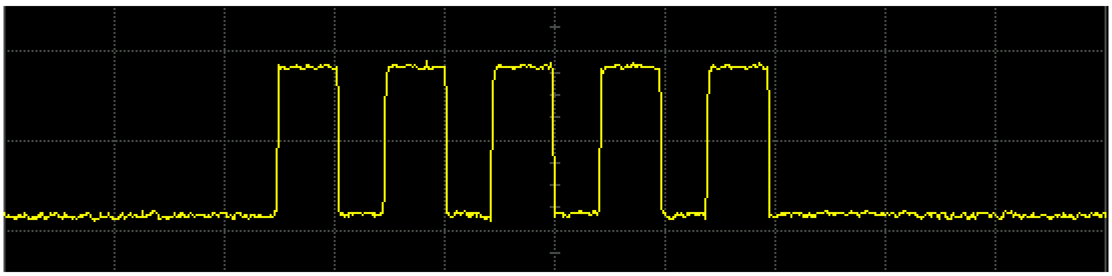

Figure 28. Fixed Number of Positive Cycles Starting from Low

Table 23. Regsister Settings for Positive Cycles with TOU Reset and Trigger Now

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E50	General Configuration: No cascading. Trigger Now, TRIG_POS_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x22C (TRIG1_CFG_3)	0x01E8	Cycle Width (Period) = 1ns × 0x1E8 = 488ns
0x230 (TRIG1_CFG_5)	0x0005	Cycle Count = 5
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

In Figure 28, the first step resets TOU1 by writing to the TRIG_SW_RST register. Amongst other things, this step forces the TOU output to a low state. Figure 29 is a variation on this example, in which the reset step is replaced with a step that disables TOU1. Since the TOU is not reset, the values of all of the TOU registers (0x220 – 0x237) are not reset, and it is necessary to be mindful of their state. The intent here is to show what happens if the TOU output is starting from a high state prior to the positive pulse.

In Figure 28, the reset operation forces the TOU output low in advance of the output pattern. In the variation below, the reset operation is removed, and replaced by a TOU disable operation. Because the TOU is programmed for TRIG_POS_CYCLE, the TOU output goes low when the TOU is enabled in the last step, in preparation for the positive cycles. If the TOU output is starting from a low state, the output appears as shown in the picture above. However, if the TOU output is starting from a high state, it will first transition low 48ns prior to the start of the output pattern, as shown in Figure 29.

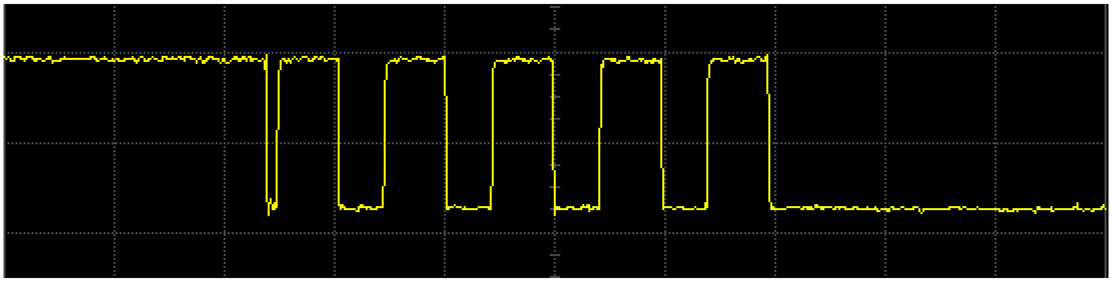

Figure 29. Fixed Number of Positive Cycles Starting from High (with Trigger Now)

Table 24. Register Settings for Positive Cycles without TOU Reset

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E50	General Configuration: No cascading. Trigger Now, TRIG_POS_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x22C (TRIG1_CFG_3)	0x01E8	Cycle Width (Period) = 1ns × 0x1E8 = 488ns
0x230 (TRIG1_CFG_5)	0x0005	Cycle Count = 5
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Example #17

A sequence of negative cycles is generated immediately using the Trigger Now feature. The first step of this command sequence resets the TOU, causing its output to go low prior to the output pattern. The last step enables the TOU. Because the TOU is programmed for TRIG_NEG_CYCLE, the TOU output goes high when the TOU is enabled, in preparation for the negative cycle. Because the Trigger Now feature is used, there is 48ns delay between this positive edge and the negative pulse.

- Period = 488ns
- Pulse width = 280ns
- 5 cycles
- Start immediately
- GPIO0, TOU1

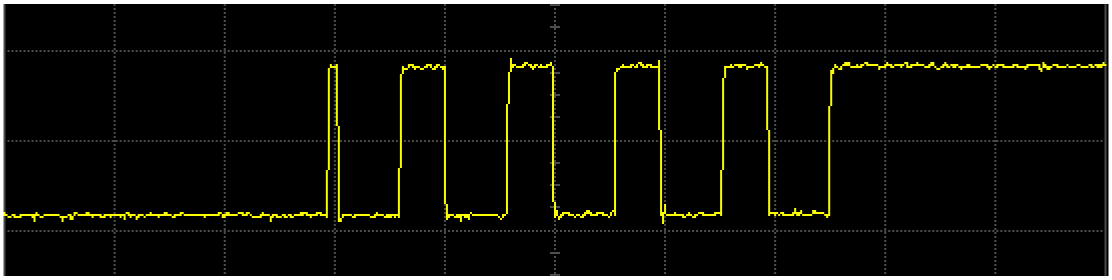

Figure 30. Fixed Number of Negative Cycles Starting from Low (with Trigger Now)

Table 25. Register Settings for Negative Cycles with TOU Reset and Trigger Now

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0001	Reset Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E40	General Configuration: No cascading. Trigger Now, TRIG_NEG_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x22C (TRIG1_CFG_3)	0x01E8	Cycle Width (Period) = 1ns × 0x1E8 = 488ns
0x230 (TRIG1_CFG_5)	0x0005	Cycle Count = 5
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

In Figure 31, the reset command is removed and replaced with a TOU disable command and the TOU output is not forced low. If the TOU output is starting from a low state, it will produce the same result as shown in Figure 30. However, if the TOU output is starting from a high state, the result is shown in Figure 31.

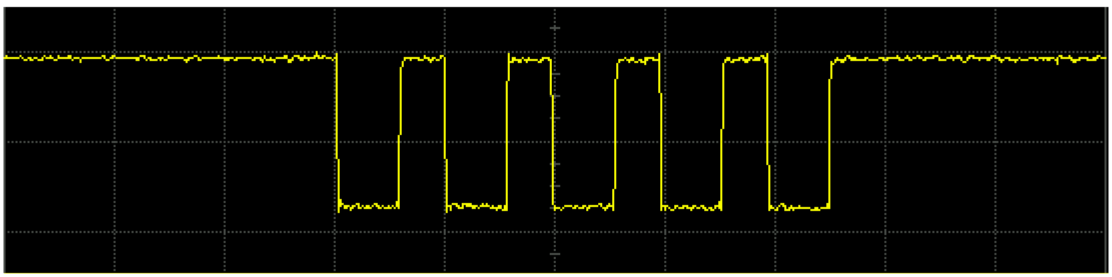

Figure 31. Fixed Number of Negative Cycles Starting from High

Table 26. Register Settings for Negative Cycles without TOU Reset

Register	Value	Description
0x206 (TRIG_EN)	0x0000	Disable Trigger Output Unit 1
0x228 (TRIG1_CFG_1)	0x3E40	General Configuration: No cascading. Trigger Now, TRIG_NEG_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x0023	Pulse Width = 8ns × 0x23 = 280ns
0x22C (TRIG1_CFG_3)	0x01E8	Cycle Width (Period) = 1ns × 0x1E8 = 488ns
0x230 (TRIG1_CFG_5)	0x0005	Cycle Count = 5
0x206 (TRIG_EN)	0x0001	Enable Trigger Output Unit 1

Cascade Mode

Cascading is a feature for synchronizing two or more TOUs so that their outputs are combined in a sequential manner as shown in Figure 32. Typically they are configured to drive the same GPIO pin, but this is not a requirement. Any combination of TOUs may be used, and they may be linked in any order. To avoid confusion with particular TOUs, the description below uses letters ‘A’, ‘B’ and ‘C’ instead of specific numbers to designate different TOUs. Three TOUs are shown cascaded, but the concept is the same for any number of TOUs, from 2 to 12.

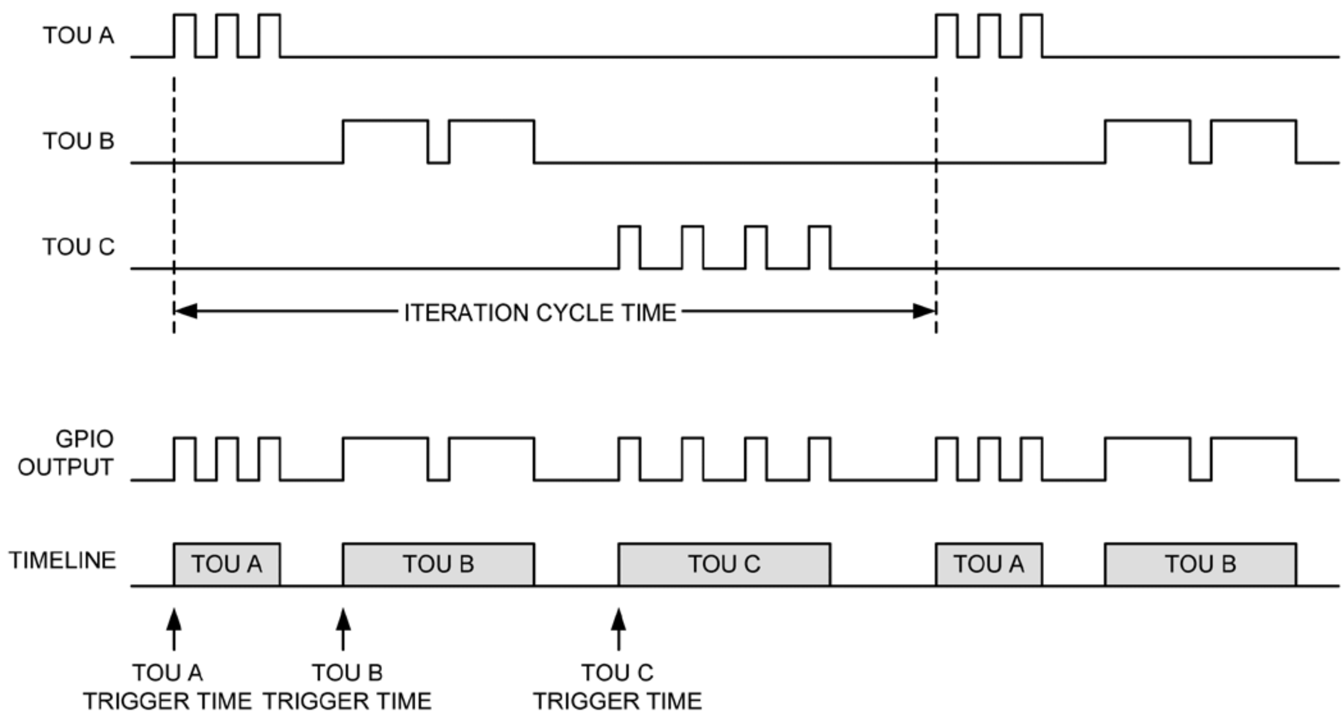

Figure 32. Cascade Mode Timing Overview

The first TOU of the cascade group is also called the “head” unit. It generates its output pattern when the PTP clock time equals the TOU’s trigger time (the trigger time for the initial event is programmed in the TRIGn_TGT target time registers, and subsequent trigger times are automatically calculated by adding the iteration cycle time from the TRIGn_CFG_7 and 8 registers). Likewise, the other TOUs initiate their output patterns when their individual trigger times are reached. After the last (tail) TOU in the cascade group has been triggered, the cascade group may repeat. It will repeat infinitely if no TOU is identified as the tail unit by the setting of the tail unit indication bit in the TRIGn_CFG_1 register. It will repeat a finite number of times by setting the tail unit indication bit for the last TOU, and specifying a value for the iteration count in either register TRIGn_CFG_2 or TRIGn_CFG_6 (depending on the signal pattern mode). The number of loops is the iteration count value plus one, so a value of zero gives a single pass through the cascade sequence.

All TOUs in a cascade group must be programmed with the same iteration cycle time value, but the Iteration Count value only needs to be programmed for the last (tail) unit in the sequence.

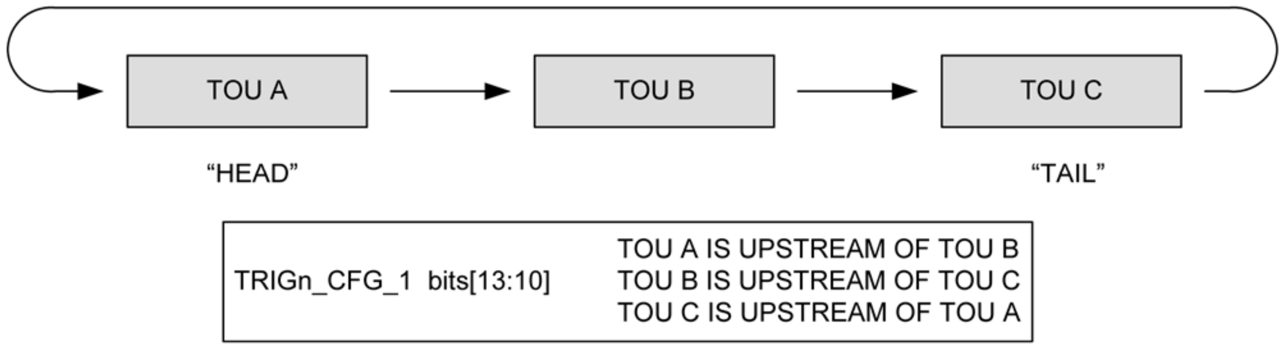

Figure 33. Sequence of Cascaded TOUs

Recall that when multiple TOUs are configured to drive the same GPIO, their outputs are combined with a logic OR function. It is therefore standard practice to configure all of the cascaded TOUs so that they start and end their output patterns in a logic low state. Each cascaded TOU may be configured for any of the signal pattern modes (edge, pulse, etc.), although the edge modes are typically not of practical use in a cascaded configuration.

The iteration cycle time and the individual TOU trigger target times (TRIGn_TGT registers) should be calculated carefully so that the output patterns of sequential TOUs do not overlap. For example, if one TOU generates an output pattern of four pulses with a cycle width of 1µs, then the total time for that pattern is 4µs, and the trigger target time for the next TOU in the sequence should be at least 4µs greater than the trigger target time for this TOU. Likewise, the iteration cycle time, which is the interval at which the entire cascade sequence repeats, must be great enough so that the last (tail) TOU has time to complete its output pattern before the first (head) TOU begins again.

The TRIGn_CFG_1 register has important bit fields that must be set properly for cascade mode, as described in Table 27.

Table 27. Description of TRIGn_CFG_1 Register

Bit	Field Name	Usage Description for Cascade Mode
15	Enable Cascade Mode	Set this bit for every cascaded TOU.
14	Tail Unit Indication (Cascade Mode)	Setting this bit indicates the last (tail) TOU in a cascade sequence that is to repeat a finite number of times. To set up an infinitely repeating cascade sequence, set this bit to 0 in all the tail TOU in the cascade sequence. Do not set this bit in non-tail cascaded TOUs, or when the TOU is not cascaded.
13:10	Upstream Trigger Unit (Cascade Mode)	When in cascade mode, specify the upstream (preceding) TOU. The first TOU in the sequence should specify the last (tail) unit. Values 0000 to 1011 correspond to TOU1 to TOU12. When a TOU is not cascaded, set this field to 1111.
9	Trigger Now	When in cascade mode, valid (future) trigger times must be specified in the target time registers (do not rely on Trigger Now). It is therefore appropriate to set this bit to 0.

Table 28 shows how these bits, and the cascade-specific registers, would be set in the case of a cascade consisting of three TOUs.

Table 28. Register Settings for Three Cascaded TOUs

Bit Field	Field Name	Value for TOU A	Value for TOU B	Value for TOU C
TRIGn_CFG_1 bit[15]	Enable Cascade Mode	1	1	1
TRIGn_CFG_1 bit[14]	Tail Unit Indication	0	0	0 = Repeat infinitely 1 = Repeat count is indicated in Iteration Count register
TRIGn_CFG_1 bits[13:10]	Upstream Trigger Unit	TOU C (0000 = TOU1 and 1011 = TOU12)	TOU A (0000 = TOU1 and 1011 = TOU12)	TOU B (0000 = TOU1 and 1011 = TOU12)
TRIGn_CFG_2 (register mode) or TRIGn_CFG_6 (not register mode)	Iteration Count	Not used	Not used	Iteration Count (if Tail Unit Indication bit is set). (0x0000 = 1 count and 0x000F = 16 counts)
TRIGn_CFG_7 and 8	Iteration Cycle Time	The same value must be entered for all trigger units in the cascade.		

Cascading Configuration Summary

This list summarizes the extra settings that are required for cascade mode:

- TRIGn_CFG_1: Set bit 15 (Cascade Enable) for every cascaded TOU.
- TRIGn_CFG_1: Set bit 14 (Tail Unit Indication) for the last (tail) TOU in the cascaded sequence if the sequence is to repeat a fixed number of times. If it is to repeat infinitely, then clear this bit in the tail TOU. For all other cascaded TOUs, clear this bit.
- TRIGn_CFG_1: Specify the upstream (preceding) TOU in bits [13:10]. The value will be different for each TOU. For the first (head unit) TOU, the value should point to the last (tail) TOU.
- In the last (tail) TOU, specify the Iteration Count. This is in either register TRIGn_CFG_2 or register TRIGn_CFG_6, depending on the signal patter mode.
- TRIGn_CFG_7 and 8: For all of the TOUs in a particular cascade, program the same Iteration Cycle Time.
- TRIGn_TGT registers: Program a different trigger time for each TOU.
- TRIG_EN: Enable only the first TOU, not all of them.

Cascading Example

Example #18 uses two TOUs, both of which generate a small number of pulses. The trigger target time registers are used to specify distinct starting times for each TOU (the Trigger Now feature cannot be used). In this case, there is a 1500ns gap between the start of TOU1 and the start of TOU2. The TOU1 output sequence is able to complete within this time. The same iteration cycle time is programmed for both TOUs.

The time scales for Figure 34 are 5µs/div and 1µs/div.

- GPIO0. TOU1 -> TOU2
- TOU1: Positive Cycle, pulse width = 48ns, cycle width = 200ns, cycle count = 3
- TOU2: Positive Cycle, pulse width = 120ns, cycle width = 188ns, cycle count = 4
- Cascade repetition = 6
- Cascade Cycle Time = 5µs
- Start at future target time (with nanoseconds = 256) (This value is arbitrary)

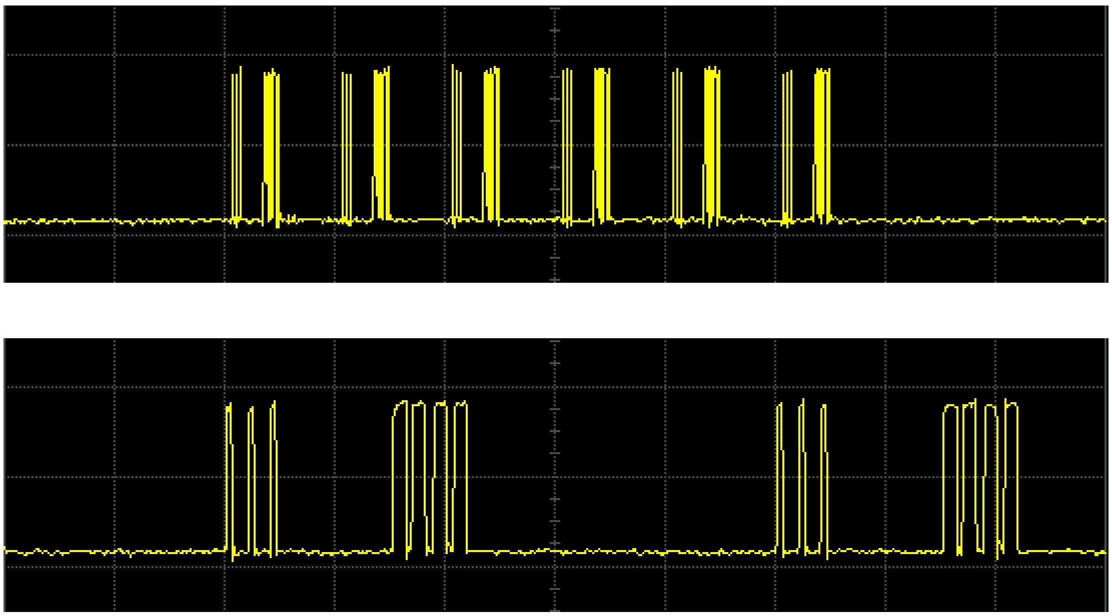

Figure 34. Cascade Mode with Two Output Units

Table 29. Register Settings for Cascading Example

Register	Value	Description
0x208 (TRIG_SW_RST)	0x0003	Reset Trigger Output Units 1 and 2
0x220 (TRIG1_TGT_NSL) 0x222 (TRIG1_TGT_NSH)	0x0100 0x0000	TOU1 Trigger Target Time (Nanoseconds) = 256ns
0x224 (TRIG1_TGT_SL) 0x226 (TRIG1_TGT_SH)	0xSSSS 0xSSSS	TOU1 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x228 (TRIG1_CFG_1)	0x8450	1000_0100_0101_0000 General Configuration: Cascade enabled, not a tail unit, upstream unit = 1 = TOU2, don't Trigger Now, TRIG_POS_CYCLE, GPIO0
0x22A (TRIG1_CFG_2)	0x0006	Pulse Width = 8ns × 0x06 = 48ns
0x22C (TRIG1_CFG_3)	0x00C8	Cycle Width = 1ns × 0xC8 = 200ns
0x230 (TRIG1_CFG_5)	0x0003	Cycle Count = 3
0x234 (TRIG1_CFG_7)	0x1388	Cascade Iteration Cycle Time = 1ns × 0x1388 = 5000ns
0x240 (TRIG2_TGT_NSL) 0x242 (TRIG2_TGT_NSH)	0x06DC 0x0000	TOU2 Trigger Target Time (Nanoseconds) = 1756ns (= 1500ns after TOU1)
0x244 (TRIG2_TGT_SL) 0x246 (TRIG2_TGT_SH)	0xSSSS 0xSSSS	TOU2 Trigger Target Time (Seconds) = A future time of the PTP 1588 Clock
0x248 (TRIG2_CFG_1)	0xC050	1100_0000_0101_0000 General Configuration: Cascade enabled, tail unit yes, upstream unit = 0 = TOU1, don't Trigger Now, TRIG_POS_CYCLE, GPIO0
0x24A (TRIG2_CFG_2)	0x000F	Pulse Width = 8ns × 0x0F = 120ns
0x24C (TRIG2_CFG_3)	0x00BC	Cycle Width = 1ns × 0xBC = 188ns
0x250 (TRIG2_CFG_5)	0x0004	Cycle Count = 4
0x252 (TRIG2_CFG_6)	0x0005	Cascade Iteration Count = 5 + 1 = 6
0x254 (TRIG2_CFG_7)	0x1388	Cascade Iteration Cycle Time = 1ns × 0x1388 = 5000ns
0x206 (TRIG_EN)	0x0001	Enable TOU1 only, not TOU2

Appendix 1

Trigger Output Unit Registers

	TOU1	TOU2	TOU3	TOU4	TOU5	TOU6	TOU7	TOU8	TOU9	TOU10	TOU11	TOU12
Trigger Error Register (TRIG_ERR)	0x200											
Trigger Active Register (TRIG_ACTIVE)	0x202											
Trigger Done Register (TRIG_DONE)	0x204											
Trigger Enable Register (TRIG_EN)	0x206											
Trigger Software Reset Register (TRIG_SW_RST)	0x208											
Trigger Output Unit 12 Output PPS Pulse Width Register (TRIG_PPS_WIDTH)	0x20A [10-8]											0x20A [7-0]
Trigger Output Unit Target Time in ns Low Word (TRIGn_TGT_NSL)	0x220	0x240	0x260	0x280	0x2A0	0x2C0	0x2E0	0x300	0x320	0x340	0x360	0x380
Trigger Output Unit Target Time in ns High Word (TRIGn_TGT_NSH)	0x222	0x242	0x262	0x282	0x2A2	0x2C2	0x2E2	0x302	0x322	0x342	0x362	0x382
Trigger Output Unit Target Time in Seconds High Word (TRIGn_TGT_SH)	0x224	0x244	0x264	0x284	0x2A4	0x2C4	0x2E4	0x304	0x324	0x344	0x364	0x384
Trigger Output Unit Target Time in Seconds Low Word (TRIGn_TGT_SL)	0x226	0x246	0x266	0x286	0x2A6	0x2C6	0x2E6	0x306	0x326	0x346	0x366	0x386
Trigger Output Unit Configuration and Control Register 1 (TRIGn_CFG_1)	0x228	0x248*	0x268	0x288	0x2A8	0x2C8	0x2E8	0x308	0x328	0x348	0x368	0x388
Trigger Output Unit Configuration and Control Register 2 (TRIGn_CFG_2)	0x22A	0x24A	0x26A	0x28A	0x2AA	0x2CA	0x2EA	0x30A	0x32A	0x34A	0x36A	0x38A
Trigger Output Unit Configuration and Control Register 3 (TRIGn_CFG_3)	0x22C	0x24C	0x26C	0x28C	0x2AC	0x2CC	0x2EC	0x30C	0x32C	0x34C	0x36C	0x38C
Trigger Output Unit Configuration and Control Register 4 (TRIGn_CFG_4)	0x22E	0x24E	0x26E	0x28E	0x2AE	0x2CE	0x2EE	0x30E	0x32E	0x34E	0x36E	0x38E
Trigger Output Unit Configuration and Control Register 5 (TRIGn_CFG_5)	0x230	0x250	0x270	0x290	0x2B0	0x2D0	0x2F0	0x310	0x330	0x350	0x370	0x390
Trigger Output Unit Configuration and Control Register 6 (TRIGn_CFG_6)	0x232	0x252	0x272	0x292	0x2B2	0x2D2	0x2F2	0x312	0x332	0x352	0x372	0x392
Trigger Output Unit Configuration and Control Register 7 (TRIGn_CFG_7)	0x234	0x254	0x274	0x294	0x2B4	0x2D4	0x2F4	0x314	0x334	0x354	0x374	0x394

Note:

Register 0x248 has an extra field (bit 7) that is not present in the other TRIGn_CFG_1 registers.

MICREL, INC. 2180 FORTUNE DRIVE SAN JOSE, CA 95131 USA
TEL +1 (408) 944-0800 FAX +1 (408) 474-1000 WEB http://www.micrel.com