Micrel 1588 PTP Application Notes

Rev 1.1

August 24, 2011

Table of Contents

1 Revision History....3
2 Introduction....4
3 Trigger Output Operation....4
3.1 Register Description....4
3.2 Single Command Execution....10
3.2.1 Before Command Completion....10
3.2.2 After Command Completion....11
3.2.3 Notes....11
3.3 Cascade Command Execution....11
3.3.1 Before Command Completion....12
3.3.2 After Command Completion....13
3.3.3 Notes....13
4 Timestamp Input Detection....14
4.1 Register Description....14
4.2 Single Command Execution....17
4.2.1 Before Command Completion....17
4.2.2 After Command Completion....18
4.2.3 Notes....18
4.3 Cascade Command Execution....18
4.3.1 Before Command Completion....18
4.3.2 After Command Completion....19
4.3.3 Notes....19
5 Clock Adjustment....19
5.1 Register Description....19
5.2 Notes....22
6 Ingress Timestamp....22
7 Egress Timestamp....22
7.1 Register Description....22
8 PTP Clock Modes and Filtering....24
8.1 Register Description....24
9 Clock Delays....26
9.1 Register Description....26

1 Revision History

Revision	Date	Summary of Changes
1.0	06/06/11	Initial revision.
1.1	08/24/11	Added **PTP Clock Modes and Filtering** section.

2 Introduction

This document describes how to use Micrel KSZ846x 1588 PTP devices in software. It provides instructions to execute Trigger Output Operation and Timestamp Event Detection. It describes how the PTP clock can be adjusted to provide synchronization.

Most of the software implementation is done in driver level. They can also be done in application level, with the exception of interrupt processing. Refer to the ***Micrel 1588 PTP Developer Guide*** for how the driver provides functions that can be accessed by applications.

3 Trigger Output Operation

3.1 Register Description

Trigger Output Configuration and Control Register 1 (0x228 – 0x229)

Bit	Default	R/W	Description
15	0	RW	**Enable Trigger Output Unit in Cascade mode**
14	0	RW	**Indicate Tail Unit in Cascade Mode**
13:10	0xF	RW	**Select Upstream Trigger Done Unit in Cascade Mode**
9	0	RW	**Trigger Now**
8	0	RW	**Trigger Notify**
7	0	RO	**Reserved**
6:4	0x0	RW	**Trigger Output Signal Pattern**
3:0	0x0	RW	**Trigger Output to GPIO**

This register is the main control of trigger output. Inside the GPIO and output signal pattern can be specified.

Output Event	Value	Output Signal
TRIG_NEG_EDGE	0	Negative edge—a falling edge from high to low
TRIG_POS_EDGE	1	Positive edge—a rising edge from low to high
TRIG_NEG_PULSE	2	Negative pulse—falling edge then rising edge after pulse time
TRIG_POS_PULSE	3	Positive pulse—rising edge then falling edge after

		pulse time
TRIG_NEG_CYCLE	4	Negative cycle—falling edge then rising edge after pulse time and stay high through cycle time
TRIG_POS_CYCLE	5	Positive cycle—rising edge then falling edge after pulse time and stay low through cycle time
TRIG_REG_OUTPUT	6	Register bit pattern output—0 to indicate low and 1 to indicate high

The TRIG_NOW bit is used in case when the target time is already past. Then the hardware immediately executes the command. Otherwise the operation results in error. Normally it is turned off.

The TRIG_NOTIFY bit is used to let hardware notifies the host by interrupt to indicate command completion or error. Normally it should be turned on.

The TRIG_CASCADE_EN bit is used when cascade mode is used. Then the Upstream Trigger Done Unit is used to point to the previous output unit. Suppose units 1, 2, and 3 are used in cascade mode. The Upstream Trigger Done Unit in unit 1 is 3; unit 2, 1; and unit 3, 2. The default value of Upstream Trigger Done Unit is TRIG_CASCADE_UPS_MASK, 0xF.

The TRIG_CASCADE_TAIL bit is used to indicate the last unit in cascade mode. If output units in cascade mode are repeated forever then last unit needs not be set.

The hardware does not always check the TRIG_CASCADE_EN bit to see if the unit is linked. Therefore it is necessary to put the Upstream Trigger Done Unit to 0xF when the unit is not used in cascade mode to guarantee proper operation.

When a previous command on a GPIO pin results in a high level, it is necessary to reset the unit first before changing to another GPIO pin. Otherwise the level of the new GPIO pin immediately goes to high.

Trigger Output Configuration and Control Register 2 (0x22A – 0x22B)

Bit	Default	R/W	Description
15-0	0x0000	RW	**Trigger Output Pulse Width** Pulse time in 8 ns unit **Trigger Output Iteration Count in Cascade Mode** Repeat count for register bit pattern output

This register programs the pulse time in 8 ns unit. It is used when the output pattern is TRIG_NEG_PULSE, TRIG_POS_PULSE, TRIG_NEG_CYCLE, or TRIG_POS_CYCLE. The maximum pulse time therefore is 524280 ns. For a longer pulse time see the Trigger Output PPS Pulse Width register. Programming 0 to this register is invalid under normal mode.

This register is also used to program the repeat count of output units under cascade mode when

the output pattern of last unit is TRIG_REG_OUTPUT, as it is the only place left available to provide that information. The value programmed is (repeat count - 1).

Trigger Output Configuration and Control Register 3, 4 (0x22C – 0x22F)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Trigger Output Cycle Width**

This register programs the cycle time in nanosecond. It is used when the output pattern is TRIG_NEG_CYCLE, TRIG_POS_CYCLE, and TRIG_REG_OUTPUT. Cycle time covers pulse time and so must be longer than that. The cycle time should be at least 58 ns longer than pulse time to guarantee proper operation. The minimum cycle time is 80 ns.

For pattern TRIG_REG_OUTPUT the cycle time indicates how long each bit maintains the output level.

This register is normally written in 32-bit.

Trigger Output Configuration and Control Register 5 (0x230 – 0x231)

Bit	Default	R/W	Description
15-0	0x0000	RW	**Trigger Output Cycle Count for Periodic Signal** **Trigger Output Bit Count for Register Output**

This register programs the repeat count of the command. Normally it will be 1. Programming 0 means the command run infinitely. This is valid only in TRIG_NEG_CYCLE, TRIG_POS_CYCLE, and TRIG_REG_OUTPUT patterns. For TRIG_REG_OUTPUT pattern it indicates how many bits are used. If it is more than 16 the bit loops back to the least significant bit.

Trigger Output Configuration and Control Register 6 (0x232 – 0x233)

Bit	Default	R/W	Description
15-0	0x0000	RW	**Trigger Output Bit Pattern** Least significant bit used first **Trigger Output Iteration Count in Cascade Mode** Repeat count for patterns other than register bit

This register programs the register bit pattern when TRIG_REG_OUTPUT pattern is used. Together with the Trigger Output Configuration and Control Register 5 it informs hardware how many bits are used.

This register is also used to program the repeat count of output units under cascade mode when

the output pattern of last unit is not TRIG_REG_OUTPUT, as it is the only place left available to provide that information. The value programmed is (repeat count - 1).

Trigger Output Configuration and Control Register 7, 8 (0x234 – 0x237)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Trigger Output Iteration Cycle Time in Cascade Mode**

This register programs the iteration time in nanosecond. It is only used when the cascade output is repeated. The iteration time is added to the target time of the unit and the operation starts at the new target time.

This register is normally written in 32-bit.

Note the iteration times of different units can vary. If the gap between units is small or the cascade repeat count is big enough, eventually the time of operation in each unit will overlap and the operation may not produce the correct result. Having the same iteration time in all units guarantee the cascade output can be repeated without any problem.

Trigger Output Target Time in Nanosecond (0x220 – 0x223)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Trigger Output Target Time in Nanosecond**

This register programs the target time in nanosecond. It is normally written in 32-bit.

Trigger Output Target Time in Second (0x224 – 0x227)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Trigger Output Target Time in Second**

This register programs the target time in second. It is normally written in 32-bit.

The target time should be in advance for the operation to start at that time. If that time is past when the command is executed, the hardware will not start the operation unless TRIG_NOW bit is set. That minimum time depends on hardware access time. For generic bus it is about 25 microseconds, but for SPI it can be longer than 8 milliseconds.

All the output units have the same set of configuration registers described above. For ease of use macros are created to return the correct register given the output unit. The configuration registers of each output unit are separated by 0x20 registers.

Trigger Output PPS Pulse Width Register (0x20A – 0x20B)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
7-0	0x00	RW	**PPS Pulse Width for Trigger Output Unit 12**

This register extends the pulse time by 8-bit value and so overcomes the limitation of the regular pulse time register. The resulting pulse width can be up to 134 milliseconds. It only applies to output unit 12 though. As a result that unit is used to generate PPS when the pulse time must be at least 20 ms.

Trigger Output Interrupt Enable Register (0x68A – 0x68B)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RW	**Trigger Output Interrupt Enable**

This register enables the interrupt notification of output unit when the operation is completed. The TRIG_NOTIFY bit needs to be turned on for the hardware to trigger interrupt.

Trigger Output Interrupt Status Register (0x688 – 0x689)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**Trigger Output Interrupt Status**

This register indicates the interrupt status of output unit when the operation is completed. It is accessed inside interrupt handling routine to find out which output unit has completed operation either successfully or in error.

Trigger Output Enable Register (0x206 – 0x207)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RW	**Trigger Output Unit Enable**

This register enables the output unit to start the operation. When the operation is completed successfully the bit is automatically self-cleared. But it is good practice to turn the unit off after completion as the result may be in error.

Trigger Output Software Reset Register (0x208 – 0x209)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RW	**Trigger Output Unit Reset**

This register resets the output unit so that it can release the GPIO pin. It is used mostly in cascade mode as the unit can be activated without being enabled directly.

Trigger Output Error Register (0x200 - 0x201)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**Trigger Output Unit Error**

This register reports the error status of output units. The reporting is turned on by not setting the TRIG_NOW bit. It is cleared by disabling the unit in Trigger Output Enable Register.

Trigger Output Active Register (0x202 – 0x203)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**Trigger Output Unit Active**

This register reports the active status of output units. The unit is active when it is enabled and ready to execute the trigger output command. It is inactive when the command is finished or there is an error.

Trigger Output Done Register (0x204 – 0x205)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**Trigger Output Unit Done**

This register reports the done status of output units. It is cleared by writing to it or when the unit becomes active again.

GPIO Monitor Register (0x680 – 0x681)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**GPIO Inputs Monitor**

This register reports the status of GPIO output pins.

3.2 Single Command Execution

Software needs to manage the output units to see which one is in use so that it is not inadvertently put to use again.

Given the output unit, GPIO pin, output mode, pulse time, cycle time, repeat count, iteration time, target time, and other parameters, software can setup the normal trigger output with the following instructions.

3.2.1 Before Command Completion

- Get the trigger output configuration register 1 using the `TRIGn_CONF_1` macro.
- Generate the register value with the GPIO pin and output mode. Make sure the Upstream Trigger Done Unit value is 0xF. Enable `TRIG_NOW` and `TRIG_NOTIFY` accordingly.
- Write the value to configuration register 1.
- If the output mode is `TRIG_REG_OUTPUT`, get the configuration register 6 using the `TRIGn_BIT_PATTERN` macro and write the bit pattern to the register. Skip programming the pulse time.
- If the output mode is greater than `TRIG_POS_EDGE`, then program the pulse time.
- Get the pulse time configuration register 2 using the `TRIGn_PULSE_WIDTH` macro.
- Convert the pulse time to 8 ns unit. If the result is 0 change it to 1. If it is more than 65535, cap it with 65535.
- Write the pulse time value to register.
- For output unit 12 the pulse time value can be larger than 65535. In this case the value is capped at 0xffffff. Program the high 8-bit value to the PPS pulse width register.
- If the output is greater than `TRIG_POS_PULSE`, then program the cycle time.
- Make sure cycle time is longer than pulse time. It should be at least 58 ns longer for proper operation. The minimum cycle time is 80 ns.
- Get the cycle time configuration register 3 using the `TRIGn_CYCLE_WIDTH_L` macro.

- Write the 32-bit cycle time value to register.
- Get the repeat count configuration register 5 using the `TRIGn_PER_OCCUR` macro.
- Make sure the count is less than 65536. Write the repeat count value to register.
- Get the target time register using the `TRIGn_TARGET_NANOSEC_L` macro.
- Write the 32-bit nanosecond target time to register.
- Get the target time register using the `TRIGn_TARGET_SEC_L` macro.
- Write the 32-bit second target time to register.
- Write to the output interrupt enable register to enable interrupt if necessary.
- Write to the output enable register to enable the unit to start the operation.

3.2.2 After Command Completion

When the operation is completed, software is notified by interrupt if that is enabled. Otherwise the output done register can be polled to get the status. Software then write to the output enable register to disable the unit if necessary. If it is required that the GPIO level also need to drop, then software can write to the output reset register to reset the unit. The unit then releases the GPIO pin.

3.2.3 Notes

1. The default operating mode is trigger output combine mode. Output units operated on the same GPIO pin will get their results ORed together. A unit keeping a GPIO level high will result in high level in that GPIO no matter what the other units do. Therefore, software needs to keep track of output units that raise GPIO level to high in case users forget which units are used to raise level high.

3.3 Cascade Command Execution

The other operation mode is trigger output cascade mode. Two or more output units are linked together to generate output sequentially. It is not required the same GPIO pin is used, but it is assumed it is.

The output units used in cascade mode need not to be consecutive, but it is normally so for ease of implementation in software.

Users need to specify how many output units are needed in cascade mode so that software can

find and allocate unused consecutive units.

Given the output units, GPIO pins, output modes, pulse times, cycle times, repeat counts, iteration times, target times, and other parameters, software can setup the cascaded trigger output with the following instructions.

3.3.1 Before Command Completion

For each output unit,

- Get the trigger output configuration register 1 using the `TRIGn_CONF_1` macro.
- Generate the register value with the GPIO pin and output mode. The Upstream Trigger Done Unit value should point to previous unit. The first unit should point to the last unit. Enable `TRIG_CASCADE_EN`. In the last unit enable `TRIG_CASCADE_TAIL` if the cascade output is not repeated infinitely. Enable `TRIG_NOW` and `TRIG_NOTIFY` accordingly.
- Write the value to configuration register 1.
- If the output mode is `TRIG_REG_OUTPUT`, get the configuration register 6 using the `TRIGn_BIT_PATTERN` macro and write the bit pattern to the register. Skip programming the pulse time.
- If the output mode is greater than `TRIG_POS_EDGE`, then program the pulse time.
- Get the pulse time configuration register 2 using the `TRIGn_PULSE_WIDTH` macro.
- Convert the pulse time to 8 ns unit. If the result is 0 change it to 1. If it is more than 65535, cap it with 65535.
- Write the pulse time value to register.
- For output unit 12 the pulse time value can be larger than 65535. In this case the value is capped at 0xffffff. Program the high 8-bit value to the PPS pulse width register.
- If the output is greater than `TRIG_POS_PULSE`, then program the cycle time.
- Make sure cycle time is longer than pulse time. It should be at least 58 ns longer for proper operation. The minimum cycle time is 80 ns.
- Get the cycle time configuration register 3 using the `TRIGn_CYCLE_WIDTH_L` macro.
- Write the 32-bit cycle time value to register.
- Get the repeat count configuration register 5 using the `TRIGn_PER_OCCUR` macro.
- Make sure the count is less than 65536. Write the repeat count value to register.
- Get the iteration time configuration 7 register using the `TRIGn_ITERATE_TIME_L` macro.

- Write the 32-bit iteration time to register.
- Get the target time register using the `TRIGn_TARGET_NANOSEC_L` macro.
- Write the 32-bit nanosecond target time to register.
- Get the target time register using the `TRIGn_TARGET_SEC_L` macro.
- Write the 32-bit second target time to register.

For the last output unit,

- Program the cascade output repeat count if not running infinitely. The count is normally 1.
- Get the pulse width configuration register 2 if output pattern is `TRIG_REG_OUTPUT` using the `TRIGn_PULSE_WIDTH` macro. Or get the register pattern configuration register 6 if output pattern is not `TRIG_REG_OUTPUT` using the `TRIGn_BIT_PATTERN` macro.
- Write the value of (repeat count - 1) to register.
- Write to the output interrupt enable register to enable interrupt if necessary.

- Write to the output enable register to enable the first unit to start the operation.

3.3.2 After Command Completion

When the operation is completed, software is notified by interrupt if that is enabled. Otherwise the output done register can be polled to get the status. Software then write to the output reset register to reset each unit except the last to make sure the GPIO pin is released by the unit. The last unit can be reset also if it is not required the GPIO level to be kept high.

3.3.3 Notes

1. Note that in normal combined mode the hardware immediately changes the GPIO level to the opposite to prepare for the output pattern. In the example of `TRIG_NEG_EDGE` the hardware raises the level to high if it is not high already. In cascade mode hardware only does this 8 ns before the target time.
2. There is a hack to start a cascade output in high level resulted from a previous cascade output without additional level change. Execute a normal trigger output in the last unit in cascade mode to raise the level using either `TRIG_POS_EDGE` or `TRIG_REG_OUTPUT` pattern. As the level is already high there is no change. Reset the last unit in previous

cascade mode to release the GPIO pin. Again there will be no change to GPIO level. Setup the cascade mode normally.

3. Although the level in TRIG_NEG_EDGE ends in low, it is recommended to reset the unit when the operation is completed in normal mode. This allows the unit running the TRIG_NEG_EDGE pattern to behave consistently in cascade mode.

4 Timestamp Input Detection

4.1 Register Description

Timestamp Configuration and Control Register (0x422 – 0x423)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11:8	0x0	RW	**Timestamp Input From GPIO**
7	0	RW	**Enable Rising Edge Detection**
6	0	RW	**Enable Falling Edge Detection**
5	0	RW	**Indicate Tail Unit in Cascade Mode**
4:1	0x0	RW	**Select Upstream Timestamp Done Unit in Cascade Mode**
0	0	RW	**Enable Timestamp Unit in Cascade Mode**

This is the main control register for timestamp input units. Inside GPIO pin and input edge detection can be specified. Both rising edge and falling edge detections can be enabled simultaneously, but in that case hardware will only generate an interrupt when both events are actually available, not only the first one.

The TS_CASCADE_EN bit is used when cascade mode is required. Then the Upstream Timestamp Done Unit is used to point to the previous input unit. Suppose units 1, 2, and 3 are used in cascade mode. The Upstream Timestamp Done Unit in unit 1 is 3; unit 2, 1; and unit 3, 2.

The TS_CASCADE_TAIL bit is used to indicate the last unit in cascade mode.

Timestamp Status Register (0x420 – 0x421)

Bit	Default	R/W	Description
15-5	0x000	RO	**Reserved**

4-1	0x0	RO	**Number of Detected Event Count**
0	0	RO	**Number of Detected Event Count Overflow**

This register reports number of events detected in the timestamp unit. The Number of Detected Event Count will increase from 0 to up to 15. Then the Number of Detected Event Count Overflow bit is set to indicate the count is more than 15. However, as the maximum number of events for most of the timestamp units is only 2, these information are meaningless. Software needs to cap the number of events to 2 for all units except unit 12 where the maximum number of events is 8.

Timestamp Event Time in Nanosecond Low-word Register (0x424 – 0x425)

Bit	Default	R/W	Description
15-0	0x0000	RO	**Timestamp Input Event Time in Nanosecond Low-word [15:0]**

This register reports the timestamp in nanosecond. It stores the low-word part of that information.

Timestamp Event Time in Nanosecond High-word Register (0x426 – 0x427)

Bit	Default	R/W	Description
15	0	RO	**Reserved**
14	0	RO	**Timestamp Input Event Edge Indication** 0 = falling edge 1 = rising edge
13-0	0x0000	RO	**Timestamp Input Event Time in Nanosecond High-word [29:16]**

This register reports the timestamp in nanosecond. It stores the high-word part of that information together with the event edge indication.

Timestamp Event Time in Second Register (0x428 – 0x42B)

Bit	Default	R/W	Description
31-0	0x00000000	RO	**Timestamp Input Event Time in Second**

This register reports the timestamp in second.

Timestamp Event Time in Sub-Nanosecond Register (0x42C – 0x42D)

Bit	Default	R/W	Description

15-3	0x0000	RO	**Reserved**
2-0	0x0	RO	**Timestamp Input Event Time in Sub 8ns** 000: 0 ns 001: 8 ns 010: 16 ns 011: 24 ns 100: 32 ns 101-111: NA

This register reports the timestamp in sub-nanosecond. The value should be added to the nanosecond value to provide more accurate time.

All timestamp input units except unit 12 have 2 sets of timestamp event registers. Unit 12 has 8 sets of timestamp event registers. The registers in each event are separated by 0x10 registers.

The configuration registers of each input unit are separated by 0x20 registers. For each of use two macros are created. They are `TSn_CONF` and `TSn_EVENT_STATUS`.

Timestamp Interrupt Enable Register (0x68E – 0x68F)

Bit	Default	R/W	Description
15-12	0x0	RW	**Egress Timestamp Interrupt Enable**
11-0	0x000	RW	**Timestamp Input Interrupt Enable**

This register enables interrupt notification of input units when event is detected.

Timestamp Enable Register (0x402 – 0x403)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RW	**Timestamp Input Unit Enable**

This register enables the input unit to detect event. After events are detected it is required to turn the unit off. Otherwise the number of detected events will increase until the overflow bit is set, although the actual events cannot be retrieved.

Timestamp Software Reset Register (0x404 – 0x405)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RW	**Timestamp Input Unit Reset**

This register resets the input unit. It is used mostly in cascade mode as unit can be activated without being enabled directly.

Timestamp Ready Register (0x400 – 0x401)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-0	0x000	RO	**Timestamp Input Unit Ready**

This register reports the ready status of input units. The unit is ready only when the first event is detected. For the rest of events software needs to poll the Timestamp Status register.

Timestamp Interrupt Status Register (0x68C – 0x68D)

Bit	Default	R/W	Description
15-12	0x0	RO	**Egress Timestamp Interrupt Status**
11-0	0x000	RO	**Timestamp Input Interrupt Status**

This register indicates interrupt status of input units when event is detected. It is a more reliable than the Timestamp Ready register so that it is used to find out which unit has events.

4.2 Single Command Execution

Software needs to manage the input units to know which one is in use so that it is not inadvertently put to use again.

Given the input unit, GPIO pin, detection mode, and other parameters, software can setup the normal timestamp input with following instructions:

4.2.1 Before Command Completion

- Get the configuration register using the `TSn_CONF` macro.
- Generate the register value with the GPIO pin and input mode.
- Write the value to configuration register.
- Write to the timestamp interrupt register to enable interrupt if necessary.
- Write to the timestamp enable register to start the detection.

4.2.2 After Command Completion

When events are detected software is notified by hardware interrupt if that is enabled. Otherwise the Timestamp Ready register can be polled to get the status. Software reads the Timestamp Status register of the input unit to get the number of events. There should be at least 1 event. Software then read the timestamp event registers to get the timestamp from each event. There is no certain way to know when the events after the first one happens unless through polling the Timestamp Status register. It is advised a timeout after the first event be implemented so that software can stop the input unit after a certain time.

4.2.3 Notes

1. An input unit can be made repeatable by just disabling the unit and then enabling it again after the first event detection. As hardware access takes time this may cause events to be missed. Therefore it is generally used on events that are predicable.
2. Hardware only generates interrupt for an input unit with both edge detections enabled when both events happen. There is no indication when the first event is available. Software needs to poll the hardware to get that information.

4.3 Cascade Command Execution

As most input units have only 2 events, they might not be enough. For more event detection cascade mode needs to be used.

The input units used in cascade mode need not to be consecutive, but it is normally so for ease of implementation in software.

Users need to specify how many input units are needed in cascade mode so that software can find and allocate unused consecutive units. The units can wrap around from unit 12 to unit 1 so that the 8 events capacity in unit 12 can be increased.

4.3.1 Before Command Completion

For each unit,

- Get the unit configuration register using the `TSn_CONF` macro.
- Generate the register value with the GPIO pin and input mode. The Upstream Timestamp Done Unit value should point to previous unit. The first unit should point to the last unit. Set `TS_CASCADE_EN` bit. For the last unit set `TS_CASCADE_TAIL` bit.

- Write the value to configuration register.
- Write to the timestamp interrupt register to enable interrupt if necessary.

- Write to the timestamp enable register to enable the first unit to start the detection.

4.3.2 After Command Completion

When events are detected software reads the first event in the first unit. It waits to be notified for events in the next input unit. When that happens it goes back to read the rest of events in the previous unit as it is guaranteed the event information is available. Software then resets and disables the previous unit as that one is completed. It reads the first event in current unit and waits for the rest. A timeout should be implemented so that software can stop waiting for more events. Software then stops the whole cascade input by resetting the rest of incomplete units.

4.3.3 Notes

1. The finished units in cascade mode can be re-used immediately even when the last unit is not completed.

5 Clock Adjustment

5.1 Register Description

PTP Clock Control Register (0x600 – 0x601)

Bit	Default	R/W	Description
15-7	0x000	RO	**Reserved**
6	0	RW	**Enable Step Adjustment to Clock**
5	0	RW	**Direction Control for Step Adjustment**
4	0	RW	**Enable Read Clock**
3	0	RW	**Enable Load Clock**
2	0	RW	**Enable Continuous Adjustment to Clock**

1	1	RW	**Enable Clock**
0	0	RW	**Reset Clock**

This register is the main control register for PTP real time clock. The hardware clock can be read or set using the register.

The Enable Step Adjustment together with the Direction Control bits are used to adjust the clock one time. Setting Direction Control to 1 means adding nanosecond to the clock, and 0 means subtracting.

The Enable Continuous Adjustment bit is used to adjust the clock continuously, which is the normal situation. It is noted that this bit needs to be turned off when adjusting the clock using the Enable Step Adjustment bit. The adjustment rate is specified in the Continuous Adjustment Rate registers.

The Enable and Reset Clock bits are normally not touched at all.

PTP Real Time Clock in Nanosecond Register (0x604 – 0x607)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Real Time Clock in Nanosecond**

This 32-bit register is used to communicate with the hardware about the clock in nanosecond. When the clock is read, it stores the time in nanosecond. When the clock is loaded, it specifies the time in nanosecond. When the clock is adjusted, it specifies how much to adjust in nanosecond.

PTP Real Time Clock in Second Register (0x608 – 0x60B)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Real Time Clock in Second**

This 32-bit register is used to communicate with the hardware about the clock in second. When the clock is read, it stores the time in second. When the clock is loaded, it specifies the time in second. It is noted that the clock cannot be adjusted in second.

PTP Real Time Clock in Sub-Nanosecond Phase Register (0x60C – 0x60D)

Bit	Default	R/W	Description
15-3	0x0000	RO	**Reserved**
2-0	0x0	RO	**Real Time Clock Time in Sub 8ns** 000: 0 ns 001: 8 ns

			010: 16 ns 011: 24 ns 100: 32 ns 101-111: NA

This register reports the real time clock in sub-nanosecond. Software needs to read it also to compensate the nanosecond time accurately.

PTP Continuous Clock Adjustment Rate Low-word Register (0x610 – 0x611)

Bit	Default	R/W	Description
15-0	0x0000	RW	**Continuous Clock Adjustment Rate Low-word [15:0]**

This register specifies the continuous adjustment rate of how much the clock is adjusted.

PTP Continuous Clock Adjustment Rate High-word Register (0x612 – 0x613)

Bit	Default	R/W	Description
15	0	RW	**Direction Control for Clock Adjustment**
14	0	RW	**Enable Temporary Clock Adjustment**
13-0	0x0000	RW	**Continuous Clock Adjustment Rate High-word [29:16]**

This register specifies how to adjust the clock. In addition to the continuous adjustment rate, there are a direction control and a feature to adjust the clock temporarily.

The continuous adjustment rate unit is 2^{-32} ns. The mathematical formula is `(2^32 * adjustment * 1000000000 / interval / 25000000)` = adjustment unit. If the interval is normalized at 1 second the formula is reduced to `(2^32 * adjustment / 25000000)`. The range of adjustment is under 6250 microseconds. Setting Direction Control to 1 makes the clock runs faster, while setting 0 makes it runs slower.

PTP Temporary Clock Adjustment Duration Register (0x614 – 0x617)

Bit	Default	R/W	Description
31-0	0x00000000	RW	**Temporary Adjustment Duration**

This 32-bit register is used to specify the duration when temporary clock adjustment is used. The value of duration is in 40 ns unit, so a time of 1 microsecond is calculated as (1 * 25) units. Currently there is no application to use this feature.

5.2 Notes

1. Continuous running trigger outputs like PPS may need to be restarted after clock is changed significantly. As a result it is preferred to adjust the clock using the continuous adjustment method.

6 Ingress Timestamp

The hardware embeds the ingress timestamp of incoming PTP message inside the 32-bit reserved field of PTP header of the payload. The format of the timestamp is `(((second & 3) << 30) | nanosecond)`. As the second is truncated, software needs to construct the actual second from the current time. The normal way is to read the hardware clock. But as hardware access may be very slow, it is recommended that software maintains its own system time to quickly come up with the second field.

7 Egress Timestamp

When PTP event messages are sent, hardware needs to report the egress timestamps of those messages so that software can calculate the time offset and path delay. The 4 event messages are Sync, Delay_Req, Pdelay_Req, and Pdelay_Resp. Hardware stores those timestamps in registers after sending the messages so software can retrieve them. There is one register for Sync and one for Pdelay_Resp. As E2E and P2P clock modes are not used simultaneously in normal situation, Delay_Req and Pdelay_Req messages shares another register.

7.1 Register Description

PTP Port Egress Timestamp for Pdelay_Req and Delay_Req Register (0x648 – 0x64B)

Bit	Default	R/W	Description
31-0	0x00000000	RO	**Egress Timestamp for Pdelay_Req and Delay_Req**

This register stores the egress timestamp of the Pdelay_Req or Delay_Req message sent. The format of the 32-bit timestamp is described above.

PTP Port Egress Timestamp for Sync Register (0x64C – 0x64F)

Bit	Default	R/W	Description
31-0	0x00000000	RO	**Egress Timestamp for Sync**

This register stores the egress timestamp of the Sync message sent. The format of the 32-bit timestamp is described above.

PTP Port Egress Timestamp for Pdelay_Resp Register (0x650 – 0x653)

Bit	Default	R/W	Description
31-0	0x0000	RO	**Egress Timestamp for Pdelay_Resp**

This register stores the egress timestamp of the Pdelay_Resp message sent. The format of the 32-bit timestamp is described above.

The egress timestamp registers of each port are separated by 0x20 registers.

As the registers will be wiped out when another PTP message with same type is sent, software needs to take care not to send the same type of PTP messages to the same port quickly before retrieving the egress timestamp.

Timestamp Interrupt Enable Register (0x68E – 0x68F)

Bit	Default	R/W	Description
15	0	RW	**Port 2 Egress Timestamp for Pdelay_Req/Resp and Delay_Req Interrupt Enable**
14	0	RW	**Port 2 Egress Timestamp for Sync Interrupt Enable**
13	0	RW	**Port 1 Egress Timestamp for Pdelay_Req/Resp and Delay_Req Interrupt Enable**
12	0	RW	**Port 1 Egress Timestamp for Sync Interrupt Enable**
11-0	0x000	RW	**Timestamp Input Interrupt Enable**

This register enables interrupt notification of egress timestamp when PTP event message is sent.

Timestamp Interrupt Status Register (0x68C – 0x68D)

Bit	Default	R/W	Description
15	0	RO	**Port 2 Egress Timestamp for Pdelay_Req/Resp and Delay_Req Interrupt Status**

14	0	RO	**Port 2 Egress Timestamp for Sync Interrupt Status**
13	0	RO	**Port 1 Egress Timestamp for Pdelay_Req/Resp and Delay_Req Interrupt Status**
12	0	RO	**Port 1 Egress Timestamp for Sync Interrupt Status**
11-0	0x000	RO	**Timestamp Input Interrupt Status**

This register indicates the interrupt status of available egress timestamp. With the status software can read the corresponding egress timestamp register to retrieve the latest timestamp. Note that Pdelay_Req and Pdelay_Resp timestamps share one status due to insufficient resource. Software needs to read both registers to find out which one has new timestamp.

8 PTP Clock Modes and Filtering

Hardware can filter PTP messages depending on different clock modes to help software not to process those messages. Read the **PTP Message Filtering** section in the ***Micrel 1588 PTP Developer Guide*** to see how messages are forwarded to different ports.

8.1 Register Description

PTP Message Configuration 1 Register (0x620 – 0x621)

Bit	Default	R/W	Description
15-8	0x00	RO	**Reserved**
7	0	RW	**Enable IEEE 802.3as**
6	1	RW	**Enable IEEE 1588 PTP**
5	0	RW	**Enable 802.3 Ethernet PTP Message Detection**
4	1	RW	**Enable IPv6/UDP PTP Message Detection**
3	1	RW	**Enable IPv6/UDP PTP Message Detection**
2	0	RW	**Select E2E or P2P Clock Mode**
1	0	RW	**Select Slave or Master Clock Mode**
0	1	RW	**Select 2-step or 1-step Clock Mode**

This register enables and selects certain PTP clock mode to influence how PTP messages are filtered.

Setting 2-step clock will cause hardware not to forward PTP messages to the other port as the

host is responsible for forwarding those messages. Make sure it is set to 1-step clock when the software is not running.

Setting master clock will normally cause hardware not to forward PTP messages to the other port as the master clock is like a termination unit that only sends Sync/Follow_Up and responds to Delay_Req messages. Make sure it is set to slave clock when the software is not running.

Setting P2P clock will cause hardware to add the link delay to Sync messages when forwarded to the other port, so it should be set to E2E when software is not running.

All three PTP message detections are normally enabled.

IEEE 1588 PTP is always enabled for normal operation.

IEEE 802.3as can be enabled for debug purpose. All PTP messages are forwarded to port 3 and none to port 2.

PTP Message Configuration 2 Register (0x622 – 0x623)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11	0	RW	**Enable Alternate Master**
10	0	RW	**Select Highest TX Priority for All PTP Messages**
9	0	RW	**Enable Sync/Follow_Up Association**
8	0	RW	**Enable Delay_Req/Delay_Resp Association**
7	0	RW	**Enable Pdelay_Req/Pdelay_Resp Association**
6	0	RO	**Reserved**
5	0	RW	**Drop Sync/Follow_Up and Delay_Req**
4	0	RW	**Enable Domain Check**
3	0	RO	**Reserved**
2	1	RW	**Enable IPv4/UDP Checksum**
1	0	RW	**Announce Message From Port 1**
0	0	RW	**Announce Message From Port 2**

This register enables more features for PTP message filtering.

Alternate Master allows PTP messages normally not forwarded to port 2 in master clock to be forwarded, and in 2-step clock those messages are also forwarded to port 3 for the host to forward those messages.

Domain Check allows only PTP messages with same domain to pass to the host and most important forwards messages with different domain to port 2. As a result it should be enabled all the time. The domain number is set in the PTP Domain and Version register.

Enable IPv4/UDP Checksum causes hardware to re-calculate UDP checksum after message

contents are changed. Otherwise, zero is used to indicate no checksum. When the message is received with no checksum hardware does not try to change it. It is recommended to enable checksum always. However, due to some hardware implementation issues it may be needed to disable it for proper operation. IPv6 does not allow zero checksum.

Select Highest TX Priority for All PTP Messages can be used instead of just PTP event messages.

Enable Delay_Req/Delay_Resp Association allows hardware not to forward Delay_Resp message to the other port if it reaches the requesting clock.

Enable Pdelay_Req/Pdelay_Resp Association allows hardware not to forward Pdelay_Resp and Pdelay_Resp_Follow_Up_follow_up messages to the host if they do not match the Pdelay_request message.

Enable Sync/Follow_Up Association allows hardware not to forward Follow_Up message to the host if it does not match the Sync message. However, currently there is a bug in this feature so it needs to be disabled.

Drop Sync/Follow_Up and Delay_Req causes hardware to drop those messages. It is not quite useful.

Announce Message From Port is not used.

PTP Domain and Version Register (0x624 – 0x625)

Bit	Default	R/W	Description
15-12	0x0	RO	**Reserved**
11-8	0x2	RW	**PTP Version**
7-0	0x00	RW	**PTP Domain**

This register programs the domain number for use with Domain Check feature in PTP Message Configuration 2 register.

9 Clock Delays

There are some clock delay values that need to be set outside the hardware through precise measurement so that clock synchronization can be more accurate.

9.1 Register Description

PTP Port Receive Latency Register (0x640 – 0x641)

Bit	Default	R/W	Description
15-0	0x019F	RW	**Receive Latency in Nanosecond**

This register stores the receive latency of the port. Software will normally override it.

PTP Port Transmit Latency Register (0x642 – 0x643)

Bit	Default	R/W	Description
15-0	0x002D	RW	**Transmit Latency in Nanosecond**

This register stores the transmit latency of the port. Software will normally override it.

PTP Port Asymmetric Delay Register (0x644 – 0x645)

Bit	Default	R/W	Description
15	0	RW	**Sign Bit**
14-0	0x0000	RW	**Asymmetric Delay in Nanosecond**

This register stores the asymmetric delay of the port. As specified in the PTP standard, this signed value is added to the correction field of ingress Sync and Pdelay_Resp messages or subtracted from the correction field of egress Delay_Req and Pdelay_Req messages when the messages are forwarded to other port.

PTP Port Link Delay Register (0x646 – 0x647)

Bit	Default	R/W	Description
15-0	0x0000	RW	**Link Delay in Nanosecond**

This register stores the link delay of the port. Software updates this register constantly in P2P clock mode so that hardware can compensate the correction field of Sync messages forwarded.