

ENT-AN1238
Application Note
Timestamp Out-of-Sync (OOS) Summary
January 2019



Contents

1	Revision History 1.0	1
1.1	Revision 1.0	1
2	1588 Timestamp Out-of-Sync (OOS)	2
2.1	Loss of Timestamp Coherency	2
2.2	Impact of Packet Size	3
2.3	Impact of Local Time Counter Clock	3
3	Software Recovery	4
3.1	Recovery Procedure	4
3.2	Resource Requirements	5
3.3	API Call	5
3.4	Design Considerations	6
4	Frequently Asked Questions	7

1 **Revision History 1.0**

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 **Revision 1.0**

Revision 1.0 of this document was published in August 2017. This was the first publication.

2 1588 Timestamp Out-of-Sync (OOS)

This document describes the susceptibility of the VSC8574 family of devices (revision B, C, and D die) to incorrectly process timestamps within the IEEE 1588 block. When IEEE 1588 functionality is enabled, the device should process precision time protocol (PTP) frames by inserting or modifying timestamps within these packets. Certain link conditions, signal impairments, and protocol events can lead to a loss of timestamp coherency, commonly termed out-of-sync (OOS). The methods and impact of entering OOS as well as the software mitigation and recovery routine are covered.

The VSC8574 device family contains several part numbers, of which the VSC8574 and VSC8572 base part numbers have an IEEE 1588 block, and are therefore relevant to this document. Different revisions and grade devices exist within each base part number and are indicated by different dash extensions (for example, VSC8574XKS-04). A device's full part number is visibly marked on the top of each package. The die revision can be determined by reading the last letter of the device's datecode, also printed on the top of the package. The die revisions B, C, and D are impacted by the loss of timestamp coherency issue and are addressed within this document. Revision E, devices, available in the second half of 2017, do not exhibit this susceptibility and so will not require any software routine workaround.

2.1 Loss of Timestamp Coherency

The VSC8574 and VSC8572 devices support the IEEE 1588 functionality by timestamping classified frames with a timestamp based on a calibrated local time counter (LTC) located within the device. The classification of the frame can be based on various types of supported PTP encapsulations, such as PTPoE, PTP-over-IP, and so on.

Timestamp coherency is impacted by several factors. Link conditions such as cable cuts, SFP pulls, auto-negotiations (ANEG), link flaps due to link reprovisioning, toggling the power-down management bit in MII register 0, or impairments may truncate packets. Signal impairments, especially to the reference clock for the 1588 engine, will impact the engine's behavior and accuracy.

Under normal operating conditions, the device will generate a timestamp upon reception of any packet, either at the MAC or media interface. Although every packet creates a timestamp, only PTP-classified packets will require its timestamp for further processing. Timestamps of non-classified packets will traverse the processor, but not be used. Based on the 1588 operating mode, PTP-related timestamps may be directly inserted into PTP frames at a specific field location, and/or added to the correction field of the packet. The process for ingress PTP frames is the same as egress PTP frames. The scope of each timestamp is strictly local to the frame for which it is generated.

For each frame reception, regardless of the classification, the following operations are expected:

- Push timestamp entry into the TSP_FIFO corresponding to frame reception on the media or MAC interface (depending on the direction: ingress or egress).
- Optional write of the timestamp from the head of the TSP_FIFO into the frame upon its departure.
- Pop the timestamp entry from the TSP_FIFO when the frame completes.

Every time a start of frame is detected at the interface, a timestamp is generated and pushed into the TSP_FIFO, which will hold this entry until a pop operation is issued to pull the FIFO's next entry before the frame is written with this timestamp upon its departure from the 1588 processor. However, if for some reason every frame received does not have a corresponding pop operation, the FIFO will accumulate an extra timestamp, causing subsequent packets to be loaded with the incorrect timestamp.

It is possible that link events such as link breaks, cable pulls, link flaps, or certain traffic interruptions may meet the defined criteria and cause the TSP_FIFO to go out-of-sync. Frames following an OOS condition will be timestamped with the time from past frames. This continues for every classified frame thereafter, until the OOS condition is cleared.

The following illustration shows the sequence of events relating to timestamps. It demonstrates two OOS conditions as a result of short or runt frames that fail to induce a corresponding TSP_FIFO pop operation and the events that follow.

Figure 1 • Timestamp Sequence

Arriving Frame	F1	F2	R3	F4	F5	F6	F7	R8	F9	F10	F11											
			Runt					Runt														
Timestamps in FIFO	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11											
PTP Classified Frame	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes											
Departing Frame	F1	F2		F4	F5	F6	F7		F9	F10	F11											
	T1	T2		T4	T5	T6	T7		T9	T10	T11											
OOS Event Sequence	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	A, B: Rx PTP frame F1, T1 into FIFO, T1 out of FIFO and into outgoing F1 (Normal) C, D: Rx PTP frame F2, T2 into FIFO, T2 out of FIFO and into outgoing F2 (Normal) E, F: Rx runt frame R3, T3 into FIFO, nothing outgoing so T3 remains in FIFO G, H: Rx std. frame F4, T4 into FIFO, T3 out of FIFO but NOT placed in outgoing F4 I, J: Rx PTP frame F5, T5 into FIFO, T4 out of FIFO and into outgoing F5 (Wrong) K, L: Rx PTP frame F6, T6 into FIFO, T5 out of FIFO and into outgoing F6 (Wrong)											M, N: Rx PTP frame F7, T7 into FIFO, T6 out of FIFO and into outgoing F7 (Wrong) O, P: Rx runt frame R8, T8 into FIFO, nothing outgoing so T7 and T8 remain in FIFO Q, R: Rx PTP frame F9, T9 into FIFO, T9 out of FIFO and into outgoing F9 (Wrong) S, T: Rx std. frame F10, T10 into FIFO, T8 out of FIFO but NOT placed in outgoing F10 U, V: Rx PTP frame F11, T11 into FIFO, T9 out of FIFO and into outgoing F11 (Wrong)										

2.2 Impact of Packet Size

As previously discussed, the positioning and handling of faults are a significant contributor to the initiation of OOS. During a link transition from up to down or down to up, external factors (such as RJ45 plug tension, optical module stabilization, SerDes lock and stabilization, and PCS synchronization) all result in some amount of PCS-codeword chatter that may impact the MAC as well as assertion of link up or down by the PHY. During this chatter, fault codes may appear that affect the internal SOF/EOF signal triggering inside the 1588 processor.

As a result of such non-determinism, a link transition must be assumed fatal to the timestamp coherency of the 1588 processor, requiring a recovery routine to assure timestamp coherence upon link re-establishment.

2.3 Impact of Local Time Counter Clock

The local time clock is the primary clock to the 1588 processor logic. Signals are transferred from the PCS block into the 1588 processor, which operates in different clock domains. Synchronization logic is used to ensure proper signal transfer between domains. The SOF signal from the PCS domain is stretched to three words (24 ns) as it crosses to the 1588 processor domain. Assuming the lowest allowed 1588 processor clock frequency of 125 MHz (or an 8 ns period), if one edge is missed due to a glitch or intended clock transition, the second edge will still arrive within the 24 ns SOF pulse width, thus allowing the signal to correctly cross domains. If no edges occur during the 24 ns SOF pulse, the 1588 processor domain will lose synchronization with the data path, resulting in a command FIFO error known as TSP_CMD_FIFO_OOS.

A local time clock error greater than two clock cycles must be assumed fatal to the timestamp coherency of the 1588 processor, requiring a recovery routine to assure timestamp coherence upon local time clock recovery.

Note

Because the local time counter (LTC) is based off the 1588 processor clock, any missing edges will result in a LTCError (for instance, the time of day value). This error will continue until the counter is reloaded or updated.

3 Software Recovery

The software routine to recover from the OOS condition within the 1588 timestamping engine uses the on-chip Ethernet pattern generator (EPG) along with on-chip loopback modes to perform a full system detect and recovery. A high-level description of the recovery algorithm is provided, along with resource reservations required of the host system in order to execute this recovery algorithm. Any register settings modified by the routine are saved at entry and restored prior to exit. The routine executes as a blocking process, preventing interruption by any other calls to the PHY API.

The following recovery sequence should be executed after any potential OOS-inducing event.

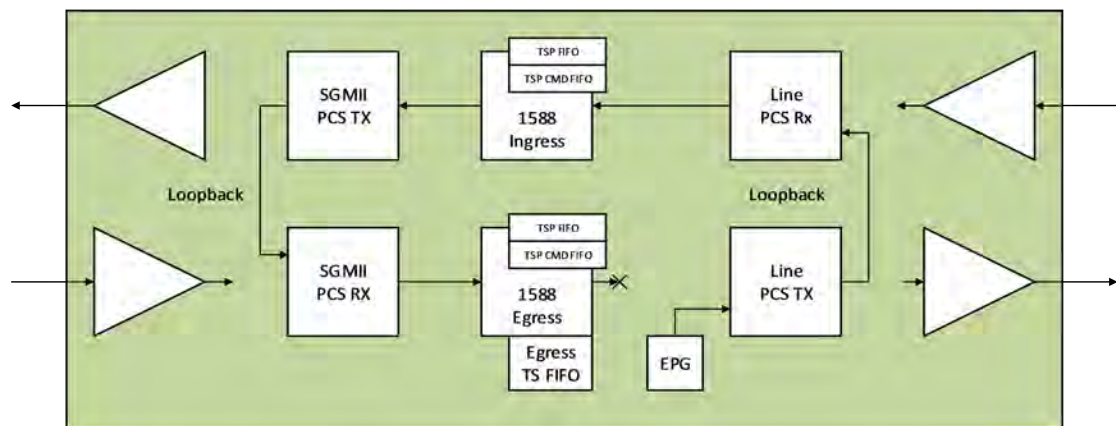
The following events should be used to trigger the recovery API:

- A link-down state (on the MAC or media interface) while traffic is flowing
- Toggling of the 1588 IP block bypass bit while traffic is flowing
- Toggling of test loopback bits (for instance, digital loopback MII register 0, bit 14) while traffic is flowing
- Toggling the power down management bit in MII register 0, bit 11 while traffic is flowing
- Toggling the restart auto-negotiation management bit in MII register 0, bit 9 while traffic is flowing
- Interruption of the local time clock to the 1588 engine while traffic is flowing

3.1 Recovery Procedure

The following illustration shows a channel in the VSC8574 device. The initial steps of the software patch will isolate the targeted recovery port in the PHY by setting near-end and media loopbacks (so that traffic ends in the 1588 egress block) and by enabling MII isolate.

Figure 2 • VSC8574 Targeted Recovery Port



The following transactions occur within the routine in order to recover the OOS condition within the 1588 timestamping engine:

- EPG generates frames using test mode configuration.
- Frames are not transmitted on the MAC side (through the MII Isolate bit) or media side because the PHY datapath is isolated early in the routine.
- Ingress 1588 block is configured to insert 30-bit timestamps in PTP packet.
- Egress 1588 block is configured to extract ingress timestamps from the frames and save both 10-byte egress timestamps and 30-bit ingress timestamps into the egress TS FIFO.
- A TSP_FIFO recovery routine will execute on the 1588 engine (Eng 0 or Eng1) configured for PTP_ETH_IPv4 encapsulation.
- A TSP_CMD_FIFO recovery routine will execute on one PTP_OAM engine (Eng2A or 2B).

Once complete, the EPG is disabled and the 1588 engine configuration is restored to the original configuration programmed by the user PTP application and external frame traffic is re-enabled in the PHY datapath.

3.2 Resource Requirements

The following constraints are placed on the 1588 timestamping engine to properly execute the OOS recovery routine.

- One of the 1588 engines (Eng 0 or Eng1) must be configured for PTP_ETH_IPv4 encapsulation; its lowest priority 1588 flow (flow 7) will always be reserved for this algorithm.
- One PTP_OAM engine (Eng2A or 2B) will also be reserved; its lowest priority Ethernet flow will always be reserved for this algorithm.
- The application PTP signature mask must be configured to include at least VTSS_PHY_TS_FIFO_SIG_SEQ_ID | VTSS_PHY_TS_FIFO_SIG_SOURCE_PORT_ID.

3.3 API Call

The software recovery API referenced in this application note involves minor differences between API versions in terms of compilation defines as well as an execution method. The API and corresponding compilation flags referenced in the following section represents the general-release method to initiate OOS recovery compatible with the latest Unified API.

Note

The software recovery API requires downloading a new micro-code to the 8051 microcontroller embedded in the VSC8574. Thus, service-interrupted (cold-boot) upgrade of the PHY API is required to include this software recovery API.

In conjunction with using the VSC8574 software recovery within the 1588 API, the following 1G compilation defines for a system (that vary by application) are available.

- -DVTSS_OPSYS_LINUX=1 : at compilation, configure for Linux OS
- -DVTSS_OPT_PORT_COUNT=4 : at compilation, set the maximum number of ports this API instance supports
- -DVTSS_CHIP_CU_PHY : compile in 1G copper PHY API
- -DVTSS_FEATURE_PHY_TIMESTAMP : compile in 1588 PTP API code
- -DVTSS_PHY_TS_SPI_CLK_THRU_PPS0 : compile in new SPI mode

The added compilation defines for the VSC8574 software recovery within the 1588 API are:

- DVTSS_FEATURE_SERDES_MACRO_SETTINGS—compile in VSC8574 family SerDes 6G patch config
- DTESLA_ING_TS_ERRFIX—compile in VSC8574 OOS recovery code enable (in conjunction with SKU-specific micro-patch download for VSC8574 revision B/C and VSC8574 revision D. Must also define VTSS_FEATURE_SERDES_MACRO_SETTINGS)

To include VSC8574 OOS recovery, add these compilation defines:

- VSC8574 rev. B/C with TESLA_ING_TS_ERRFIX defined and VTSS_FEATURE_SERDES_MACRO_SETTINGS defined, micro-code CRC is: 0xA44A for rev. B/C
- VSC8574 rev. D with TESLA_ING_TS_ERRFIX defined and VTSS_FEATURE_SERDES_MACRO_SETTINGS defined, micro-code CRC is: 0x042C for rev. D

These two definitions must be defined for the VSC8574 SerDes configuration to be performed by the API, as well as compiled in the micro-patch for VSC8574 OOS recovery micro-patch.

Otherwise, to exclude VSC8574 OOS recovery code:

- Do not define TESLA_ING_TS_ERRFIX, the micro-patch CRC is: 0x29E8 for all silicon revisions. VSC8574 SerDes configuration will be performed by the internal micro-patch and not in the API.

Assuming VSC8574 OOS recovery is included using the compilation options given, then the call within the 1588 API to perform the recovery is:

```
vtss_rc vtss_phy_ts_tesla_tsp_fifo_sync(const vtss_inst_t inst,
                                       const vtss_port_no_t port_no,
                                       const vtss_debug_printf_t pr,
                                       const vtss_phy_ts_fifo_conf_t *fifo_conf, BOOL *OOS);
```

This function will in turn call other functions to perform the steps outlined in Recovery Procedure, page 5, depending upon the intended response. See the 1588 API header file for a strict definition of the function

arguments to `vtss_phy_ts_tesla_tsp_fifo_sync()`.

In addition to making a direct call from the user application, there is an alternative to run the OOS recovery from a private call within the PHY API. To enable this alternative of calling the recovery from within the API, the following compilation defines must be used: `TESLA_ING_TS_ERRFIX`, `VTSS_FEATURE_SERDES_MACRO_SETTINGS`, `VTSS_TS_FIFO_SYNC`, and `VTSS_TS_FIFO_MEDIA_SWAP_SYNC`.

Note

Including the OOS recovery routine within the PHY API using the described compilation option does not exclude/prevent the application from calling OOS recovery directly. However, the API will not return an error if the application calls OOS directly when configuring the PHY API this way. In this situation, the OOS recovery routine may execute twice-in-a-row whenever it is called directly from the user application as a consequence of the link drop during OOS recovery execution, which consequently triggers `vtss_phy_status_get_private()` within the PHY API to execute the recovery routine again.

Configuring the API to automatically call OOS from within the API for either link down or from media change events are currently default configured as:

```
vtss_phy_ts_fifo_conf_t fifo_conf = {.detect_only = FALSE,
                                     .eng_recov = VTSS_PHY_TS_PTP_ENGINE_ID_0,
                                     .eng_minE = VTSS_PHY_TS_OAM_ENGINE_ID_2B
                                    };
```

There is no method to configure the automatic OOS recovery where the application would pass in the structure `vtss_phy_ts_fifo_conf_t` for the automated environment. Instead, the structure currently has to be set inside the private API functions: `vtss_phy_status_get_private()` and `vtss_phy_reset_private()`. Using it this way requires user modification of the PHY API in order to correctly configure the automatic OOS recovery for the intended parameterization: detection versus recovery, the TSP_FIFO recovery engine (Eng 0 or Eng1), and the TSP_CMD_FIFO recovery engine (Eng 2A or Eng2B). Also, if the API is configured to automatically call OOS, then any direct application call to the OOS recovery routine must define its `vtss_phy_ts_fifo_conf_t` structure identically to the existing `fifo_conf` within the private API functions `vtss_phy_status_get_private()` and `vtss_phy_reset_private()`.

3.4 Design Considerations

The software workaround for VSC8574 will drop both the MAC and media interface links during execution. No frame traffic will traverse the device while the workaround is executing. Execution time will vary based on MDIO utilization for a customer application but typically will complete in less than 1.5 seconds per call to `vtss_phy_ts_tesla_tsp_fifo_sync()`.

After execution, all PHY configuration is restored to its original state programmed by the user PTP application and external frame flow resumes.

4 Frequently Asked Questions

1. Is the out-of-sync timestamp FIFO the one accessed by MDIO or read out by the SPI interface?
No. The FIFO is internal to the 1588 engine datapath and is not directly accessible through the PHY management interface.
2. Can we write our own recovery method for the out-of-sync timestamp FIFO?
Microsemi will only support the recovery API referenced in this application note and provided by Microsemi engineering.
3. Can we avoid changing the Microsemi PHY API software?
Adding the software patch will require a software upgrade to the system; however, the new silicon revision is targeted to be hardware compatible and software compatible for systems using the current active production silicon parts. The mapping of revisions is listed in the following table.

Table 1 • VSC8574 Family Part Numbers

Part Number	Software-Compatible New Revision (with fix)
VSC8572XKS-01	VSC8572XKS-02
VSC8572XKS-04	VSC8572XKS-05
VSC8574XKS-01	VSC8574XKS-02
VSC8574XKS-04	VSC8574XKS-05

**Microsemi Headquarters**

One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

© 2017 Microsemi. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions; setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

VPPD-04468