Application Note AN013

syn1588® Timestamping of User Defined Packets

Version 1.17.2 - November 2023

Abstract

With the next release of the syn1588® Software Suite a completely new API will be available. This new API encapsulates all register access which will dramatically simplify using the MII timestamping functions. Thus, we strongly recommend new customers not to use the direct register access but wait for the new API. Please contact Oregano Systems' support for further information.

There are two timestamping units within every syn1588® PCIe NIC which allows detecting two independent classes of Ethernet packets in both receive and transmit direction respectively. One set of timestamping units is used for detecting PTP packets and thus solely used by the syn1588® PTP Stack. The second timestamping units may be used by the application software to detect an arbitrary class of Ethernet packets in each direction and draw a timestamp upon transmission or reception of a packet meeting the user defined filter condition.

This application note briefly outlines the required actions in the user software. A simple example is presented that may be extended to meet specific application requirements.

Introduction

The timestamping unit available in the syn1588® Clock_M IP core which is used in the syn1588® PCIe NIC as well as the syn1588® Dual NIC allows detecting of two independent classes of Ethernet packets both for receive and transmit direction. The following figures show the basic principle of these two units both being attached to the (G)MII interface as pure listening devices. Consequently, these timestamping units act as passive observers on the (G)MII interface without altering the data transferred in either direction.

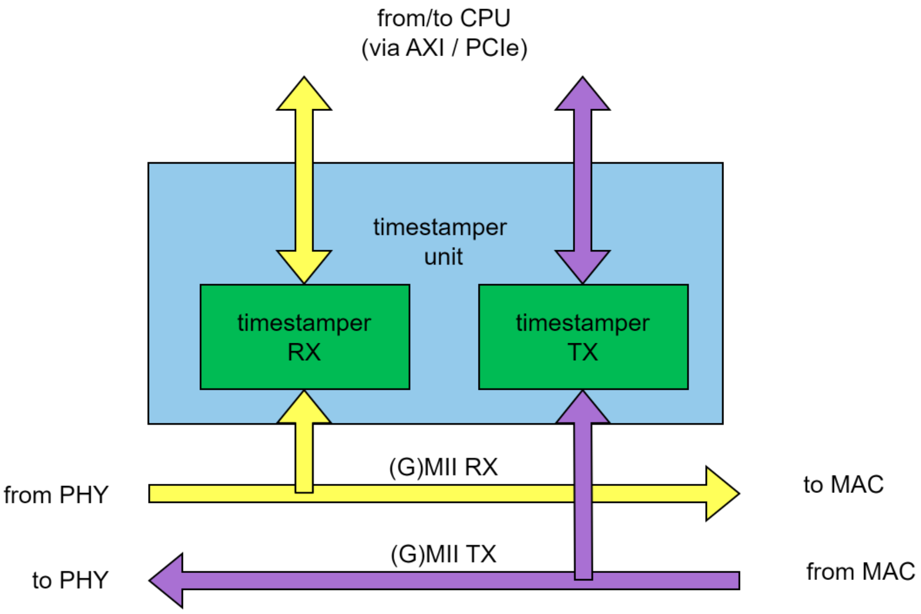

Figure 1. (G)MII timestamping units in the syn1588® PCIe NICs

For 2-step operation both timestamp units passively attach to the Media Dependent Bus interface of the MAC-to-PHY connection. For 1-step operation the transmit bus is fed through the timestamping unit since the timestamp has to be inserted on-the-fly.

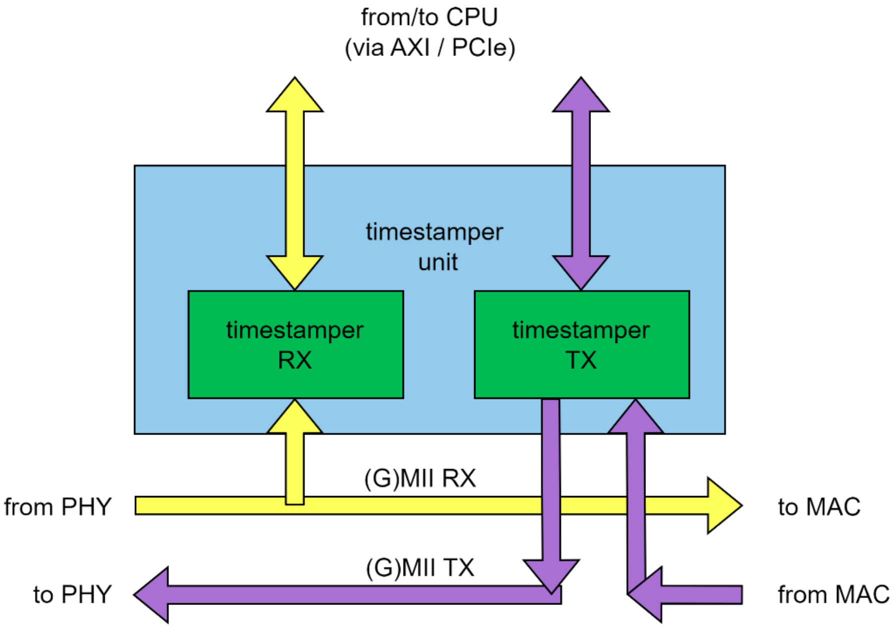

Figure 2. (G)MII timestamping units in 1-step mode

The following figure shows the basic element of the timestamping unit: the time-stamping comparator. Two of these comparators are contained in one timestamping unit. Two timestamping units comprising four timestamping comparators are available in the syn1588® PCIe/Dual NIC; one RX/TX timestamping pair is available in the Ethernet MAC (which is reserved for usage by the syn1588® PTP Stack) while a second pair of timestamping units is available in the syn1588®Clock_M IP core. Latter one is available for the user. Note, this timestamping unit does not offer 1-step support.

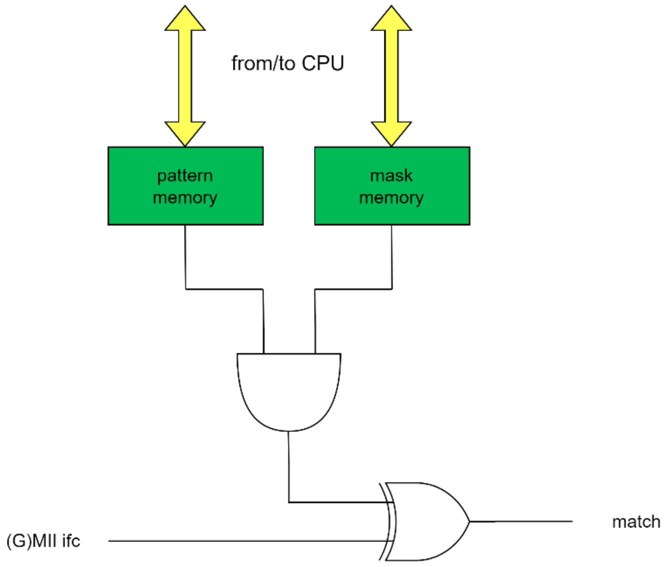

Figure 3. Basic timestamping comparator

The timestamping comparator will check up to 128 bytes of every Ethernet packet. The compare process starts with the first byte following the SFD (start-of-frame-delimiter). For every bit of the packet the user may define an expected value and a mask value. For every bit that is set to 1 in the mask memory the value of the bit in the packet is compared against the value of the respective bit in the pattern memory.

The timestamping comparator produces a match, if all active data bits in the pattern memory (i.e the bits with the corresponding mask bits set to 1) match the respective bits in the packet. In this case the timestamp of the respective packet which has been drawn previously (i.e. at SFD) is copied into a FIFO buffer memory. To allow the user application to distinguish these packets from other Ethernet traffic additional packet data can be copied into the buffer memory together with the time stamp.

A control register is attached to every timestamping comparator. providing the means to enable the timestamping comparator as well as to define the overall length of the filter (i.e. the number of bits to be analyzed starting from the SFD) and two separate data sections to be copied into the buffer memory The latter are defined via two independent offset and length values.

Timestamp Unit Control Register

Four distinct Timestamp Unit control registers are available; one for each timestamping comparator. The following table shows the contents of these Timestamp Unit control register.

Bit(s)	Description
NBYTES[7:0]	Number of bytes in the packet to compare against the pattern & mask memory. If NBYTES is set to 0 the time stamping unit is disabled. If NBYTES is larger than the total packet length no match is found. If the first NBYTES of the packet match against the pattern & mask RAM the packet is matched and the timestamp as well as the extracted data is written to the FIFO.
LEN1[11:8]	Length in number of bytes to extract from the packet starting at OFF1 to be stored into the time stamping unit FIFO. Note: (LEN1 + LEN2) mod 4 = 0
LEN2[15:12]	Length in number of bytes to extract from the packet starting at OFF2 to be stored into the time stamping unit FIFO. Note: (LEN1 + LEN2) mod 4 = 0
OFF1[23:16]	Offset value for the first data section to extract from the packet if a match has been found. The values is counted in multiples of bytes starting with the first byte after the SOF octet
OFF2[31:24]	Offset value for the second data section to extract from the packet if a match has been found. The values is counted in multiples of bytes starting with the first byte after the SOF octet

Table 1 Timestamp Unit Control Register – register bits

The following table lists the address of these control registers. Note that these are the local address of the syn1588® PCIe NIC.

Address	Description
0x1214	MII Timestamper #0 Receive – control register[1]
0x1208	MII Timestamper #0 Transmit – control register[1]
0x120c	MII Timestamper #1 Receive – control register
0x1210	MII Timestamper #1 Transmit – control register

Table 2 Address map of the Timestamp Unit Control Registers

The timestamper unit allows to specify how many packet bytes shall be compared (field NBYTES). Note, the packet data and the timestamp are routed

[1] Note, the MII Timestamper #0 is not implemented in the syn1588® PCIe NIC; it is mentioned here for completeness only.

in completely different ways through your software layers of your computer. To allow the application software to match a timestamp with the right packet characteristic data can be (shall be) extracted out of the packet to allow the identification of the suitable packet on the application level. For PTP packets e.g. the clock ID, the port ID and the sequence number are extracted. One can extract up to two sequences of data out of the packet. Every extracted data sequence is specified by an offset OFF (byte number within the packet) and a length LEN (in bytes). The total length of extracted data bytes has to be at least 4. The total number of extracted bytes has to be a multiple of 4 bytes while length of every of the two subsections has to be a multiple of 2 bytes. One shall use at least the first packet data extraction sequence using OFF1 and LEN1. If the second packet extraction sequence is not used set LEN2 to 0 and OFF2 to OFF1+LEN1.

For every timestamping unit a separate pattern and mask memory defining the comparator condition is available. To simplify the implementation the pattern and mask data are combined into a single physical memory with 128 entries; every entry containing one mask and one pattern byte. The even byte of the pattern and mask memory hold the pattern data while the mask value is stored at the odd byte location.

The following table lists the base addresses of these pattern & mask memories.

Address	Description
0x1400	MII Timestamper #0 Receive – pattern & mask memory[2]
0x1600	MII Timestamper #0 Transmit – pattern & mask memory[2]
0x1800	MII Timestamper #1 Receive – pattern & mask memory
0x1a00	MII Timestamper #1 Transmit – pattern & mask memory

Table 3 Addresses map of the Timestamp Unit pattern and mask memories

[2] Note, the MII Timestamper #0 is not implemented in the syn1588® PCIe NIC; it is mentioned here for completeness only

For every timestamping comparator a separate timestamping FIFO is available. This timestamping FIFO holds the extracted packet data as well as the timestamp drawn when the packet is sent (TX) or received (RX). Whenever the respective comparator detects a packet matching its preset condition the packet's timestamp value as well as optional data extracted from the packet are stored in this timestamping FIFO. Additionally, an interrupt can be generated informing the user application or the syn1588® PTP Stack that there is new data available.

Address	Description
0x1010	MII Timestamper #0 Receive FIFO[2]
0x1018	MII Timestamper #0 Transmit FIFO[2]
0x101C	MII Timestamper #1 Receive FIFO
0x1020	MII Timestamper #1 Transmit FIFO

Table 4 Addresses of the Timestamp Unit FIFOs

Finally there is a status register for every RX/TX timestamp FIFO pair available that holds full and empty flags as well as the number of words held by the FIFO.

Bit(s)	Description
RX FIFO EMPTY[0]	a '1' flags that the RX timestamp FIFO is empty
RX FIFO FULL[1]	a '1' flags that the RX timestamp FIFO is full
RX FIFO CNT[9:2]	number of words available in the RX timestamp FIFO
RESERVED[15:10]	reserved
TX FIFO EMPTY[16]	a '1' flags that the TX timestamp FIFO is empty
TX FIFO FULL[17]	a '1' flags that the TX timestamp FIFO is full
TX FIFO CNT[25:18]	number of words available in the TX timestamp FIFO
RESERVED[31:26]	Reserved

Table 5 Timestamp Unit FIFO Status Register – register bits

Address	Description
0x1030	MII Timestamper #0 FIFO Status
0x1034	MII Timestamper #1 FIFO Status

Table 6 Address map of the Timestamp Unit FIFO Status Register

User Timestamping Demo Application

There is a simple example code available, which shows how to setup the timestamping unit for a given class of Ethernet traffic using the syn1588® API. The example source-code may be found on the wooden Oregano Systems' USB stick containing the syn1588® Live System in the folder /sw/api/.

This folder contains a .zip file which in order contains the syn1588® API example source code and build environment.

The source code is written for Linux and for Windows. For the compilation on Linux OS, GNU-make is used together with the g++ compiler and for Windows OS the Microsoft Visual Studio compiler is used. If you require further information, please contact the Oregano Systems support.

Version 1.17.2 - November 2023

Oregano Systems
A Meinberg Company

Franzosengraben 8
A-1030 Vienna
Austria
http://oregano.at
contact@oregano.at

Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.

Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.

Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.