PtpOrdinaryClock

Reference Manual

Product Info	
Product Manager	Sven Meier
Author(s)	Sven Meier
Reviewer(s)	-
Version	2.9
Date	01.04.2025

Copyright Notice

Disclaimer

The information available to you in this document/code may contain errors and is subject to periods of interruption. While NetTimeLogic GmbH does its best to maintain the information it offers in the document/code, it cannot be held responsible for any errors, defects, lost profits, or other consequential damages arising from the use of this document/code.

NETTIMELOGIC GMBH PROVIDES THE INFORMATION, SERVICES AND PRODUCTS AVAILABLE IN THIS DOCUMENT/CODE "AS IS," WITH NO WARRANTIES WHATSOEVER. ALL EXPRESS WARRANTIES AND ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY RIGHTS ARE HEREBY DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. IN NO EVENT SHALL NETTIMELOGIC GMBH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL AND EXEMPLARY DAMAGES, OR ANY DAMAGES WHATSOEVER, ARISING FROM THE USE OR PERFORMANCE OF THIS DOCUMENT/CODE OR FROM ANY INFORMATION, SERVICES OR PRODUCTS PROVIDED THROUGH THIS DOCUMENT/CODE, EVEN IF NETTIMELOGIC GMBH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IF YOU ARE DISSATISFIED WITH THIS DOCUMENT/CODE, OR ANY PORTION THEREOF, YOUR EXCLUSIVE REMEDY SHALL BE TO CEASE USING THE DOCUMENT/CODE.

Overview

NetTimeLogic's PTP Ordinary Clock is a full hardware (FPGA) only implementation of an Ordinary Clock according to IEEE1588-2019 (PTP). The whole protocol handling, algorithms and calculations are implemented in the core, no CPU is required. This allows running PTP synchronization completely independent and standalone from the user application. The Ordinary Clock can act as Master and Slave based on the Best Master Clock (BMC) algorithm. All datasets according to IEEE1588-2019 can be configured either by signals or by an AXI4Lite-Slave Register interface.

Key Features:

- PTP Ordinary Clock according to IEEE1588-2019
- Intercepts path between MAC and PHY
- Single Port
- Synchronization accuracy: +/- 25ns
- Support for Default Profile: Layer 2 (Ethernet) and Layer 3 (Ipv4/v6)
- Support for Power Profile: C37.238-2011 including VLAN support
- Support for Utility Profile: including HSR and PRP tag handling
- Support for TSN Profile: including HSR, PRP and TSN tag handling
- Support for ITU Profiles : ITUG82651, ITUG82751, ITUG82752, up to 4096 Unicast Slaves at 128 frames/s
- One Step and Two Step support
- Peer to Peer (P2P) and End to End (E2E) delay measurement
- Optional unicast E2E handling
- Master and Slave support
- Full line speed
- AXI4Lite register set or static configuration
- Datasets according to IEEE1588
- MII/GMII/RGMII Interface support (optional AXI4 stream for interconnection to 3rd party cores)
- Optional Management Message support
- Optional Signaling Message support
- Optional TwoStep Message sending support
- Optional Lucky Packet Filter
- Timestamp resolution with 50 MHz system clock: 10ns
- Hardware PI Servo
- Asymmetry Correction

Revision History

This table shows the revision history of this document.

Version	Date	Revision
0.1	28.12.2015	First draft
1.0	07.10.2016	First release
1.1	29.06.2017	E2E added
1.2	11.08.2017	Message Intervals added
1.3	14.11.2017	Delay Naming and Link Speed added
1.4	20.12.2017	Status interface added
1.5	30.08.2018	TwoStep and Signaling and TSN added
1.6	29.09.2018	Added Asymmetry
1.7	09.10.2018	Added Variance as configurable parameter and status
1.8	16.01.2019	OC Static Status type changesd
1.9	25.04.2019	Added Default Dataset Values to Status Record
2.0	14.06.2019	Added MaxDelay
2.1	25.01.2021	Added IPv6
2.2	11.06.2021	Register set fixes
2.3	26.07.2022	E2E Unicast added
2.4	23.08.2022	E2E Unicast reworked
2.5	03.01.2023	Added Vivado upgrade version description
2.6	18.07.2023	Added Unicast modes
2.7	13.11.2023	Added DSCP
2.8	25.10.2024	IEEE1588-2008 => IEEE1588-2019/2008
2.9	01.04.2025	Added Lucky Packet Filter

Table 1: Revision History

Content

1 INTRODUCTION 10

1.1 Context Overview 10

1.2 Function 10

1.3 Architecture 11

1.4 Deviations from the Standards or Limitations 14

2 PTP BASICS 15

2.1 Protocol 15

2.2 Principles 15
2.2.1 PTP Nodes 18
2.2.2 Delay Mechanisms 22
2.2.3 Unicast vs. Multicast 25
2.2.4 Profiles 26

2.3 Accuracy 27
2.3.1 Timestamp accuracy 28

3 REGISTER SET 30

3.1 Register Overview 30

3.2 Register Descriptions 33
3.2.1 General 33
3.2.2 Default Dataset 46
3.2.3 Port Dataset 56
3.2.4 Current Dataset 67
3.2.5 Parent Dataset 73
3.2.6 Time Properties Dataset 84
3.2.7 Unicast Dataset 96

4 DESIGN DESCRIPTION 111

4.1 Top Level – PTP Ordinary Clock 111

4.2 Design Parts 137
4.2.1 RX Processor 137
4.2.2 TX Processor 143
4.2.3 Announce Processor 149
4.2.4 Delay Processor 155
4.2.5 Sync Processor 162
4.2.6 Management Processor 169
4.2.7 Signaling Processor 175
4.2.8 MAC Processor 180
4.2.9 Best Master Clock Algorithm 183
4.2.10 Datasets 189
4.2.11 Clock Counter 196
4.2.12 Ethernet Interface Adapter 198
4.2.13 Registerset 201

4.3 Configuration example 206
4.3.1 Static Configuration 206
4.3.2 AXI Configuration 207

4.4 Clocking and Reset Concept 209
4.4.1 Clocking 209
4.4.2 Reset 209

5 RESOURCE USAGE 211

5.1 Intel/Altera (Cyclone V) 211

5.2 AMD/Xilinx (Artix 7) 211

6 DELIVERY STRUCTURE 212

7 TESTBENCH 213

7.1 Run Testbench 214

8 REFERENCE DESIGNS 215

8.1 Intel/Altera: Terasic SocKit 215

8.2 AMD/Xilinx: Digilent Arty 217

8.3 AMD/Xilinx: Vivado version 218

Definitions

Definitions	
Ordinary Clock	A synchronization end node according to IEEE1588 that can take a Master and Slave role
Transparent Clock	A network node (Switch) that is IEEE1588 aware and compensates network jitter
Default Profile	PTP Profile according to IEEE1588
Power Profile	PTP Profile according to C37.238-2011
Utility Profile	PTP Profile according to IEC 61850 9-3
TSN Profile	PTP Profile according to IEEE802.1AS

Table 2: Definitions

Abbreviations

Abbreviations	
AXI	AMBA4 Specification (Stream and Memory Mapped)
IRQ	Interrupt, Signaling to e.g. a CPU
PRP	Parallel Redundancy Protocol (IEC 62439-3)
HSR	High-availability Seamless Redundancy (IEC 62439-3)
PTP	Precision Time Protocol (See also IEEE1588)
MAC	Media Access Controller
PHY	Physical Media Access Controller
OC	Ordinary Clock
TC	Transparent Clock
TS	Timestamp
ETH	Ethernet
TB	Testbench
LUT	Look Up Table
FF	Flip Flop

RAM	Random Access Memory
ROM	Read Only Memory
FPGA	Field Programmable Gate Array
VHDL	Hardware description Language for FPGA's

Table 3: Abbreviations

1 Introduction

1.1 Context Overview

The PTP Ordinary Clock is meant as a co-processor handling PTP. It intercepts the Media Independent Interface (MII) on the Ethernet path between the MAC and PHY where it handles all PTP traffic. This means it generates and processes PTP frames directly in hardware, using the same data path as the normal traffic coming from or going to the Ethernet MAC. This also means that it uses a small amount (around 4 frames per second) of the bandwidth on the MII so if 100% Network traffic shall be constantly sent by the Ethernet MAC it would eventually drop some frames to still handle PTP. The PTP Ordinary Clock is designed to work in cooperation with the Counter Clock core from NetTimeLogic (not a requirement). It contains an AXI4Lite slave for configuration from a CPU, this is however not required since the PTP Ordinary Clock can also be configured statically via signals/constants directly from within the FPGA.

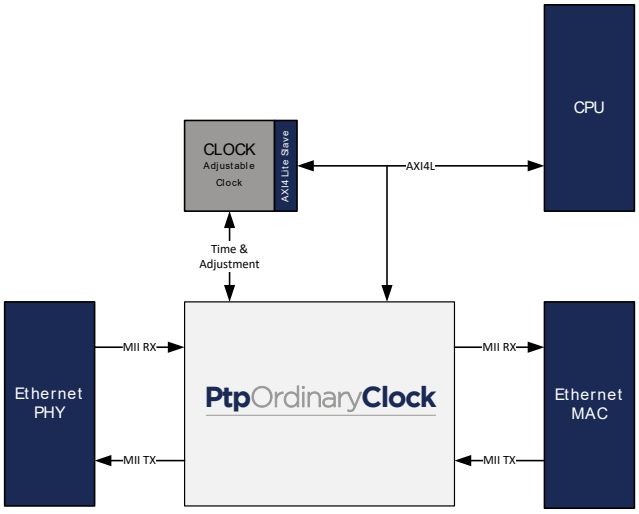

Figure 1: Context Block Diagram

1.2 Function

The PTP Ordinary Clock is a PTP OC according to IEEE1588-2019. It is an end node that can synchronize itself to another PTP Master as a PTP Slave or it can also act as PTP Master and synchronize other clocks. It measures the path delays either via the P2P or E2E delay mechanism. It contains all Datasets defined in IEEE1588.

It uses Announce messages to determine whether it is a PTP Master or Slave based on the Best Master Algorithm (BMC). It calculates the offset and drift towards its Master with Sync messages and corrects it Offset and drift corrections are feed to the PI servo loops of the Adjustable Counter Clock Core and the output of the Servo loop used for the next calculations. In case of an error the correction is stopped until two Sync message measurements are correct again after the error flag was asserted.

1.3 Architecture

The core is split up into different functional blocks for reduction of the complexity, modularity and maximum reuse of blocks. The interfaces between the functional blocks are kept as small as possible for easier understanding of the core.

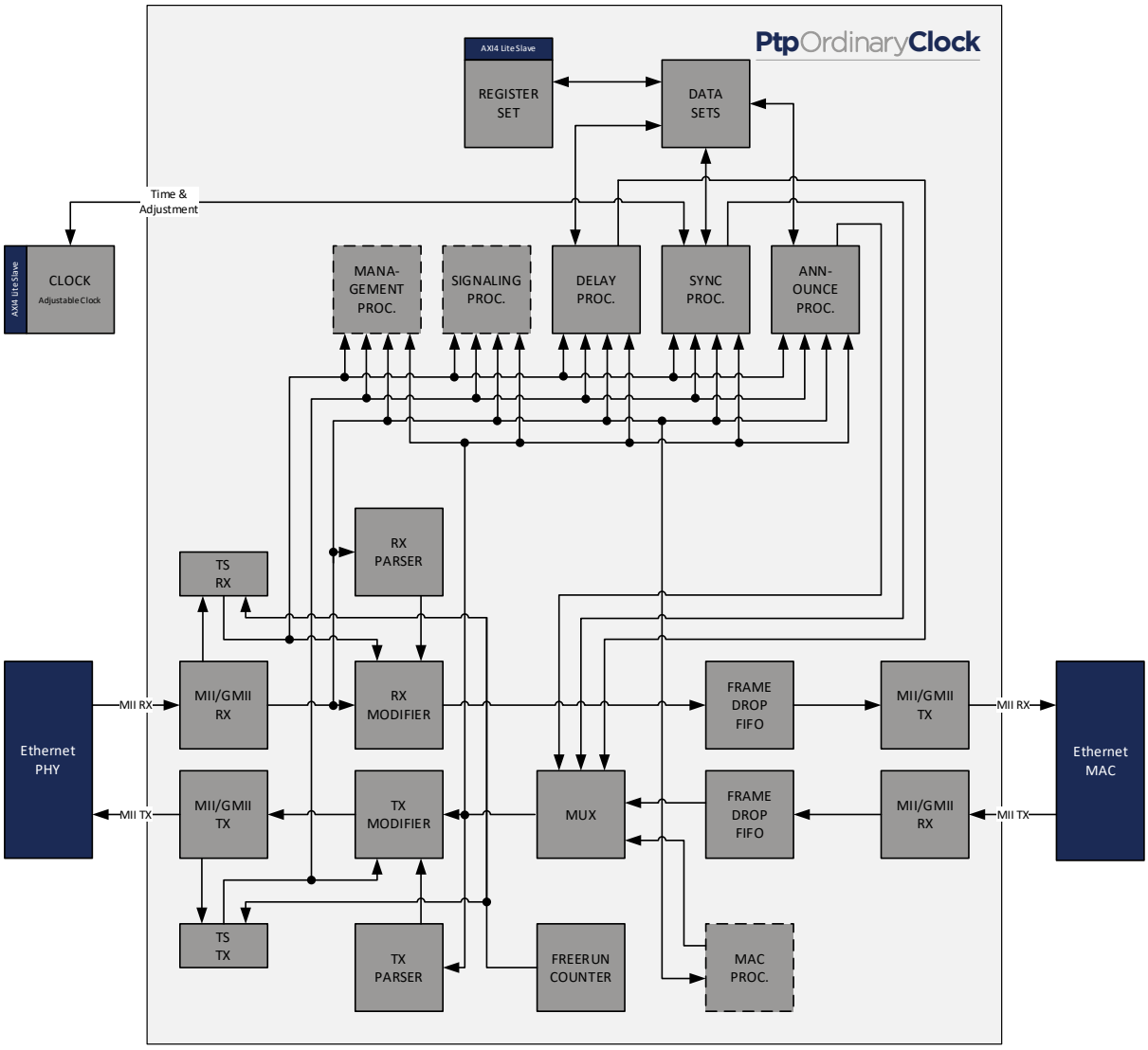

Figure 2: Architecture Block Diagram

Register Set

This block allows reading status values and writing configuration.

Data Sets

This block represents the data storage for the datasets in the data format according to IEEE1588. Multiple sources can adapt these values.

Management Processor

This block handles the PTP Management messages as a control-node, which means it is only responding to PTP management messages but no initiating PTP management messages. This block is optional for security reasons.

Signaling Processor

This block handles the PTP Signaling messages. This block is optional since only in some profiles used.

MAC Processor

This block handles the MAC Address resolution via ARP or ICMP.

Delay Processor

This block handles the PTP Delay messages. In P2P mode, it will respond to PTP PDelayReq messages with PTP PDelayResp messages in one-step mode with residence time in the correction field. It also sends PTP PDelayReq and waits for PTP PDelayResp and calculates the delay for this port and averages over the last 4 measurements. In E2E mode, when master, it will respond to PTP DelayReq messages with PTP DelayResp messages. When slave, it will send DelayReq and and calculate the delay the same way as for P2P.

Sync Processor

This block handles the PTP Sync messages. In Master state it generates PTP Sync messages in one-step mode. In Slave mode it calculates the Offset and Drift to the current master and adjusts it via the counter clock core.

Announce Processor

This block handles the PTP Announce messages and all event and state decisions based on the Best Master Clock (BMC) algorithm. It analyses incoming PTP An-

nounce messages and runs the BMC and decides in which state to core shall be. In Master state it generates Announce messages.

Receive/Transmit Timestamper

These blocks generate a snapshot of the free-running and counter clock time when the Start of Frame Delimiter (SFD) was detected by the Interface Adapters

Receive/Transmit Frame Parser

These blocks analyze the frame and extract frame information like the PTP correction field, the frame length or if the CRC ok.

Receive/Transmit Frame Modifier

These blocks modify PTP event messages (Sync and PDelay) on the fly by inserting newly calculated correction fields, timestamps and inaccuracy fields (power profile) and recreating a new CRC. It also decides whether a frame shall be dropped or not (all PTP frames are dropped in the MAC direction).

Frame Multiplexer

This block multiplexes the frames coming from all the frame processors and the forwarding path from the MAC to the PHY.

Frame Drop Fifo

This FIFO is a store and forward FIFO with drop functionality. During frame reception a drop signal can be asserted which will cause that the frame is dropped. If it will run into an overflow condition it drops the incoming frame.

Free-running Counter

This block generates a free-running counter which is used for the path delay measurement. It counts in the same format as the counter clock but with no adjustments.

(R)(G)MII Receive/Transmit Interface Adapter

These blocks convert the data stream from the (R)(G)MII to a 32bit AXI stream and back form 32bit AXI stream to (R)(G)MII.

1.4 Deviations from the Standards or Limitations

Deviations and limitations:

- Sync Intervals slowest 1/2s (which is OK for Default, Utility, Power, ITU and TSN profiles (all default within range))
- Preemption not supported for PTP frames
- 2 Step can be disabled also for IEEE802.1 AS
- No Neighbor rates calculations and announcing
- ITU8265.1 can handle multiple masters, handled as one node with multiple possible masters, switchover between masters is done based on the adapted BMCA, no hot standby.
- Unicast Delay measurement only to the current master, for others just request sending.

2 PTP Basics

2.1 Protocol

PTP means Precision Time Protocol and is standardized in IEEE1588-2019 (or also IEC61588-Ed.2). It describes the mechanisms how to distribute time (phase and frequency) precisely (sub-microsecond accuracy) over an Ethernet based, packet based network and determines the best clock for time distribution automatically. The principal of the protocol is based on frames that are exchanged periodically between nodes containing timestamps of when the exchanged frames where sent and received along with information of the clock quality of the nodes.

2.2 Principles

PTP defines a Master Slave system. In a PTP network there is only one active Master and multiple Slaves. As already mentioned there are messages periodically exchanged (and timestamped) between the Master and Slave to determine and correct the offset and drift of the slave against the master and to measure the network delay between the Slave and the Master to correct this also in the offset. For measuring the delay between the Master and Slave two mechanisms are defined: Peer To Peer (P2P) and End To End (E2E). As the names say P2P is measuring the delay only to the next neighbor and E2E is measuring from the Slave to the Master. We will see the advantages and disadvantages of the two mechanisms later, for now we assume a simple setup of a Slave directly connected to a Master with nothing then a cable in between:

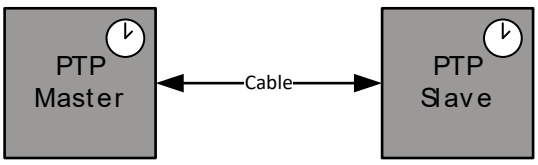

Figure 3: Simple setup

Now we look at the messages and calculations done for the two mechanisms: In both cases the Master is sending a so called Announce message and so called Sync messages to the Slave. The Master takes a timestamp T1 when it starts to send the Sync message and depending on its capabilities puts the timestamp T1 on the fly into the Sync message (one-step handling) or sends a second message called Sync FollowUp which contains T1 (two-step). On the Slave side it takes a timestamp T2

when the Sync message is coming in. With these two timestamps (T2 and T1) the slave can calculate an offset but the propagation delay between the master and slave is still missing so the Slave would have a constant offset of the delay to the Master. For calculating the delay now the two mechanisms differ:
For E2E the Slave sends a so called Delay Req message to the Master and stores the send timestamp T3. The Master takes a timestamp T4 when it receives the Delay Req message and sends this timestamp T4 via a so called Delay Resp to the Slave. Now the Slave has all four timestamps (T1-T4) to calculate the Delay according to the calculations below.
For P2P, things work a bit different. The Slave sends a so called PDelay Req message to its neighbor (in this case the Master) and stores the send timestamp T3. The neighbor takes a timestamp T4 when it receives the PDelay Req message. Then is sends a so called PDelay Resp containing T4 but and in parallel timestamps the sending moment of the PDelay Resp with T5. Again depending on the capabilities of the node it inserts the timestamp T5 on the fly into the PDelay Resp message (one-step) or sends a second message called PDelay Resp FollowUp containing T5 (two-step). The Slave takes a timestamp T6 when it receives the PDelay Resp message. Now the slave also has the four timestamps (T3-T6) to calculate the Delay according to the calculations below. In contrary to the E2E mechanism also the Master (respectively the neighbor) is also calculating the Delay the same way as the Slave.
So for E2E the Delay calculation is based on Sync messages sent by the Master where for P2P the Delay calculation is completely independent of Sync messages. This means for a high accuracy delay measurement the frequency of the two clocks have to be as close to each other as possible, where for E2E this is more important as for the P2P case. Both delay mechanisms assume a symmetrical delay which is normally the case for Ethernet.
Once the Delay is calculated the real offset can be calculated with the two timestamps (T1 and T2) from the Sync and the propagation delay calculated via one of the mechanisms. With Offset correction the phase is corrected to the one from the Master. This is done with every Sync message.
The last value that is needed to get high accuracy synchronization is the so called Drift which is the frequency difference between the Master and Slave. Since the oscillators of the Master and Slave are never 100% identical the Slave will drift away from the master during two Sync messages. To adjust the frequency the timestamps from two Sync messages are needed (T1, T1' and T2 and T2'). With these four timestamps the frequency difference can be calculated and adjusted at

the Slave. After this both frequency and phase are adjusted and the Slave is synchronized to the Master.

PTP Nodes have to be able to handle both types of messages: one-step and two-step, but they don't need to generate two-step frames if they are one-step capable and vice versa.

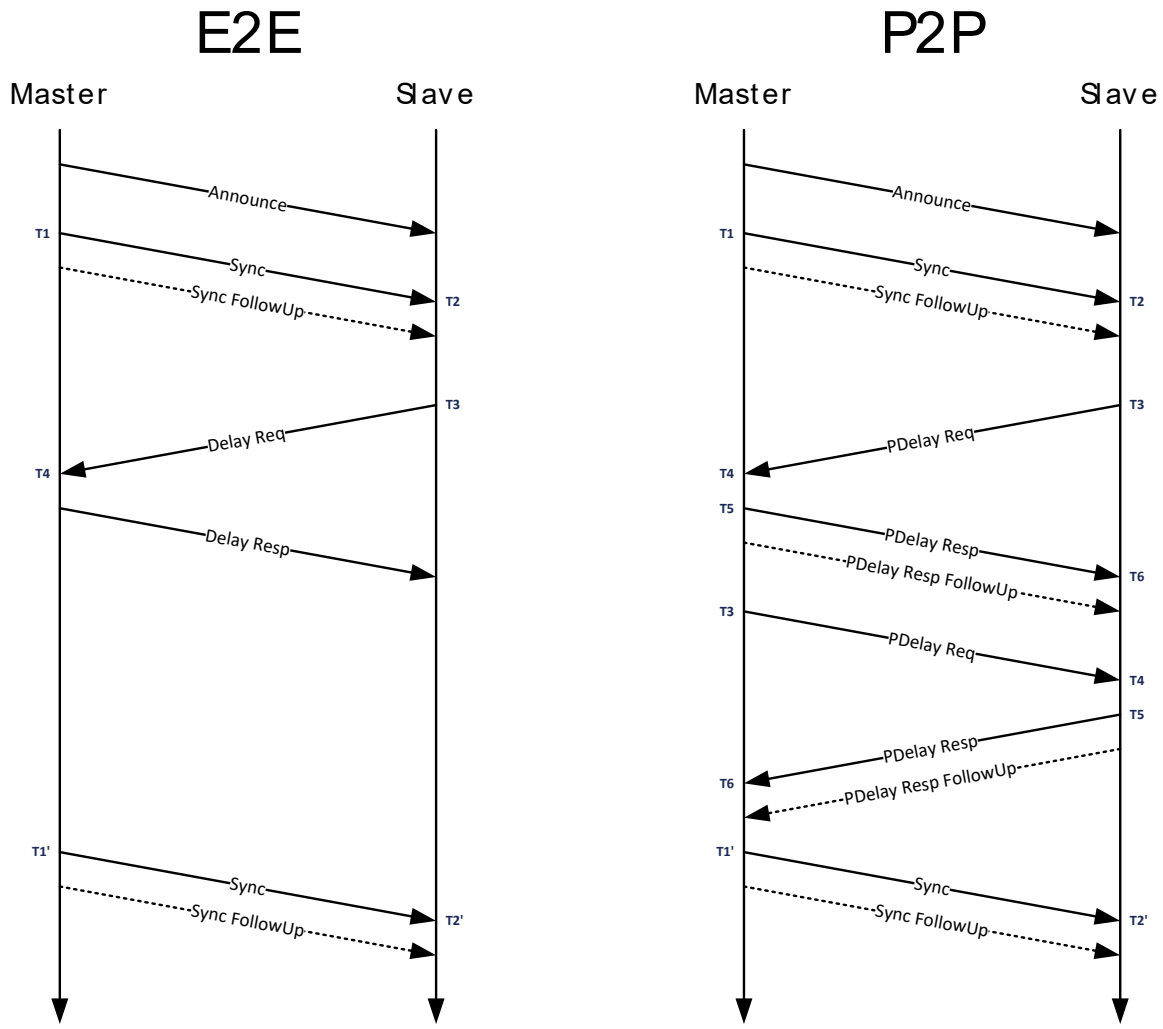

Figure 4: Message exchange simple setup

$$Delay = \frac{(T4 - T1) - (T3 - T2)}{2} \qquad Delay = \frac{(T6 - T3) - (T5 - T4)}{2}$$

$$Offset = (T2 - T1) - Delay$$

$$Drift = \frac{(T2' - T2) - (T1' - T1)}{(T1' - T1)}$$

In this example one Master was connected to exactly one Slave. In a normal setup there are many Slaves and one Master. PTP is self-organizing, which means it chooses the best available Master in a Network and all Slaves are then synchronizing to this Master. In a PTP network there are normally multiple Master capable nodes, therefore the Announce messages exist. With the Announce messages the Master capable device announces its clock quality in the network as long as no Announce message from a better node is received or a timeout occurred. This way in a steady state only one node is sending Announce messages and therefor is the Master in the network. Also the node which is sending Announce messages has to send Sync messages since it is the Master in the network. The comparison of the clock quality parameters and the state machine is defined in the Best Master Clock (BMC) algorithm.

2.2.1 PTP Nodes

IEEE1588 defines seven types of PTP nodes which all have different functions in a PTP network

2.2.1.1 Ordinary Clock (OC)

An Ordinary Clock (OC) is defined as a PTP clock with a single PTP port. It can operate either as a Master or Slave in the PTP network. The mode is selected via the BMC algorithm. Ordinary Clocks are the most common node type in a PTP network as they are generally used as end-nodes within a network requiring synchronization between each other. One of the OCs will act as a Master and all other ones will stay in Slave mode. If the current Master goes away one of the OCs will take over the Master role and synchronize the other nodes.

2.2.1.2 Grandmaster Clock (GM)

A Grandmaster Clock (GM) is defined as a PTP Ordinary Clock with either an external time source (GPS, IRIG) or a very high accuracy time (ATOM). It can only act as a Master in the PTP network and will win the Master role according the BMC. In the case that more than one Grandmaster is connected to the same PTP network the one which is worse according to the BMC will go in a Passive state where it remains as long as the Master is active. This is used in the case of backup Grandmasters.

2.2.1.3 Slave Only Clock (SC)

A Slave Only Clock (SC) is defined as a PTP Ordinary Clock which can only act as a Slave in the PTP network and will never win the Master role according the BMC; it

will therefor also never send Announce Messages. Slave Only Clocks are always end nodes, so if no Master is available in the network they will be unsynchronized. Since they don't really participate in the BMC selection a very lightweight implementation of slave only clocks is possible.

2.2.1.4 Boundary Clock (BC)

A Boundary Clock (BC) is a network element with PTP functionality. It has in contrary to the OC more than one port. A Boundary Clock is normally an Ethernet Switch or Router. The Problem with normal Switches and Routers is that the forwarding delay between a frame coming in and going out of the device is not deterministic. Therefore the concept of Boundary Clocks was introduced, where all PTP messages end and are sourced by this node rather than forwarding the PTP messages. A Boundary Clock synchronizes itself on one of the ports to the Master, so it is Slave on that port and acts on all other ports as Master synchronizing the other nodes. The state decision on the Ports is again based on the BMC. If no better Master is available it can also take the role of the Grandmaster in the network, in that case it is Master on all ports. A BC can also act as a bridge between different PTP network configurations.

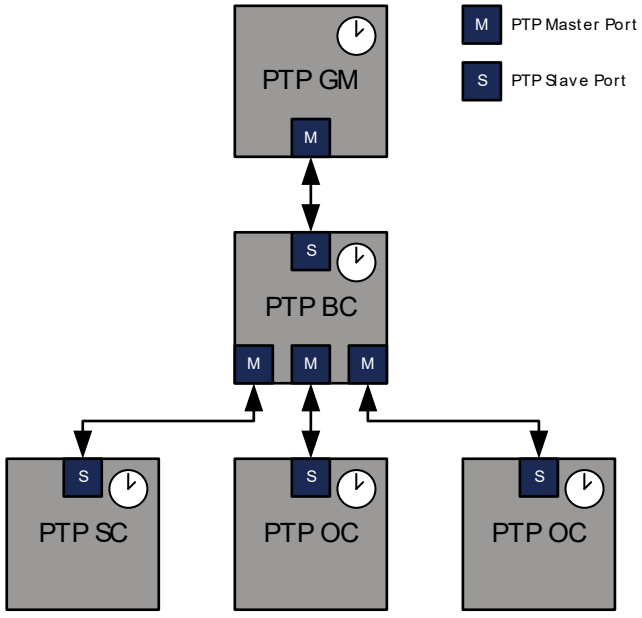

Figure 5: PTP network with Boundary Clock

2.2.1.5 Transparent Clock (TC)

A Transparent Clock (TC) is as the Boundary Clock a network element with PTP functionality. A Transparent Clock is normally an Ethernet Switch or another Network element with more than one port. In contrary to the Boundary Clock it is stateless, so no port is in a Master or Slave state. To overcome the mentioned problem of non-deterministic forwarding delays in the Switch it measure the residence time of a PTP message in the Switch and adds this value to a so called Correction Field within the PTP messages. So for the Slave a Transparent Clock is not visible, it just gets the correction values which it has to take into account in the Delay, Offset and Drift calculations. The Transparent clock comes in different flavors: E2E one-step or two-step TC and P2P one-step or two-step TC. To simplify the implementations of TCs only one-step TCs are considered in this description. A one-step TC can put the residence time of a PTP message on the fly into the message. This residence time has then be taken into account when calculating Delays and Offset so the residence time is falling out of the calculation and only the cable delays are remaining. An advantage of the TC over the BC is that no cascaded servo loops are introduced which makes deeper hierarchies possible without making the chain unstable Also reaction time of the system significantly increases since for each hierarchy you don't have to wait until the higher hierarchy has been synchronized.

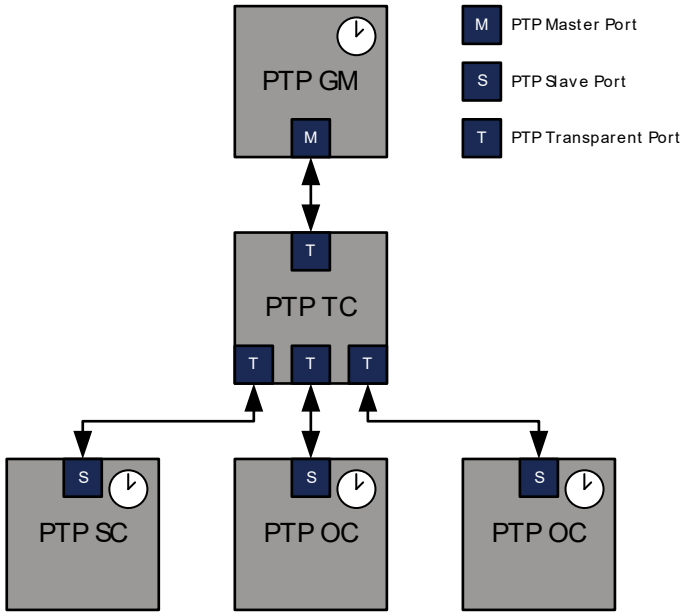

Figure 6: PTP network with Transparent Clock

2.2.1.6 Hybrid Clock (HC)

A Hybrid Clock (HC) is a combination of a Transparent and Ordinary Clock. Hybrid Clocks are often used in Daisy-Chains. In a Daisy-Chain a three port TC is used, two ports are used for the forwarding path on the Daisy-Chain and one is used as the uplink to the CPU and OC.

2.2.1.7 Management Node (MN)

A Management Node (MN) does not take part in the synchronization and BMC of the PTP network. It sends PTP Management messages to the nodes to supervise the state of the PTP network. All PTP nodes have to response to PTP Management messages according to the standard, however a lot of the PTP nodes don't support PTP Management anymore because of security reasons, therefore PTP Management nodes are not widely used.

2.2.2 Delay Mechanisms

Measuring the delay is one of the important mechanisms in PTP. In general all network nodes (Switches/Routers) shall be PTP aware (BC or TC) because of the mentioned non-determinism of message forwarding in Switches and Routers. However for E2E Delay measurements also standard Switches could be in the network, but this requires a lot of statistics and high message rates to achieve sub-microsecond accuracy (and is not always possible).
For the P2P Delay mechanism only PTP aware Switches/Routers are allowed, breaking this rule will break the synchronization! For the next chapters only PTP aware nodes are considered in the network. Only one delay mechanism per PTP segment is allowed and cannot be mixed. Special Boundary Clocks exist which can run different delay mechanisms per PTP segment (ports belonging to one network, in best case per port).

2.2.2.1 E2E

In End to End (E2E) Delay measurement the Slave measures the whole path delay to the next Master port.

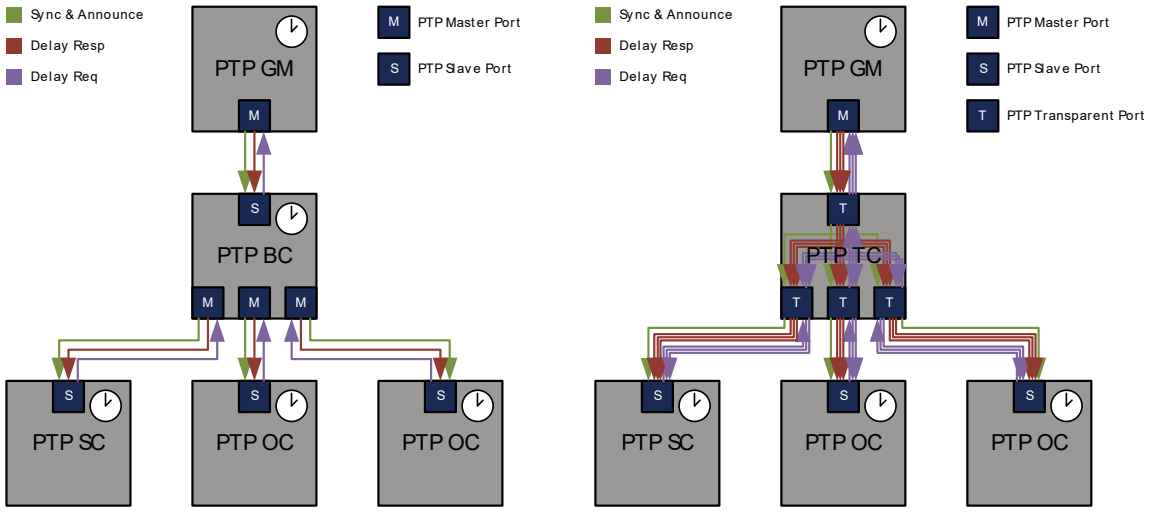

Figure 7: E2E Delay measurement with BC and TC

In the case of a network with a BC this means the Slaves measure the path directly to the BC. The BC itself measures the Delay to the Grandmaster and synchronizes to it. Since the Sync, Announce and Delay messages always end and are sourced at the BC, there is no correction needed in the Messages. Each Master port only receives the Delay Req messages from the Slave directly connected to it and each Slave port only receives Delay Resp for his Delay Req (this is not the case if non PTP aware Switches/Routers are used) from the BC.

In the case of a network with an E2E TC, things look differently. Since the TC is stateless it just forwards all PTP messages according to switching rules (all PTP messages in this case are L2 multicast messages) to all other ports except the port the message came from. This means that the Grandmaster receives the Delay Req messages from all three Slaves and has to answer all of the with a Delay Resp message. Also a Slave receives all Delay Req messages from all the other Slaves as well as all Delay Resp messages from the Master, means the Delay Resp message for his Delay Req message but also the Delay Resp messages as response to the other Slaves Delay Req messages. In a large-scale network this produces quite some unnecessary network load because of the other Slaves Delay messages at the Slaves and quite some CPU and network load on the Master because it has to answer all Slaves Delay Req message. The TC corrects the residence time of the Sync and Delay Req messages directly in the Correction Field of these messages. A Slave can subtract this correction value from its Delay so the only things that remain are the cable delays.

There is also a mixed multicast/unicast mode where PTP Sync and Announce messages are still L2 multicast messages, but PTP Delay Req and Delay Resp messages are sent as L2 unicast messages. This will allow to reduce the bandwidth usage in the network with E2E TCs so other PTP nodes don't receive any unnecessary PTP frames. Otherwise, everything is working the same way. The only requirement is that all PTP nodes also support this scheme.

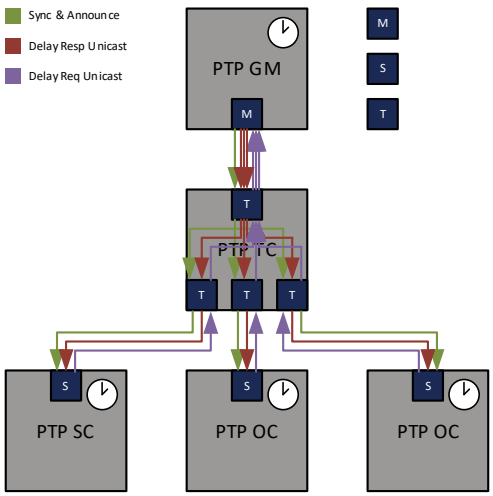

Figure 8: E2E Delay unicast measurement with TC

2.2.2.2 P2P

In Peer to Peer (P2P) Delay measurement each PTP node measures the delay only to its direct neighbor independent of it's the node type and port state. So the Slave never knows the whole delay to the master.

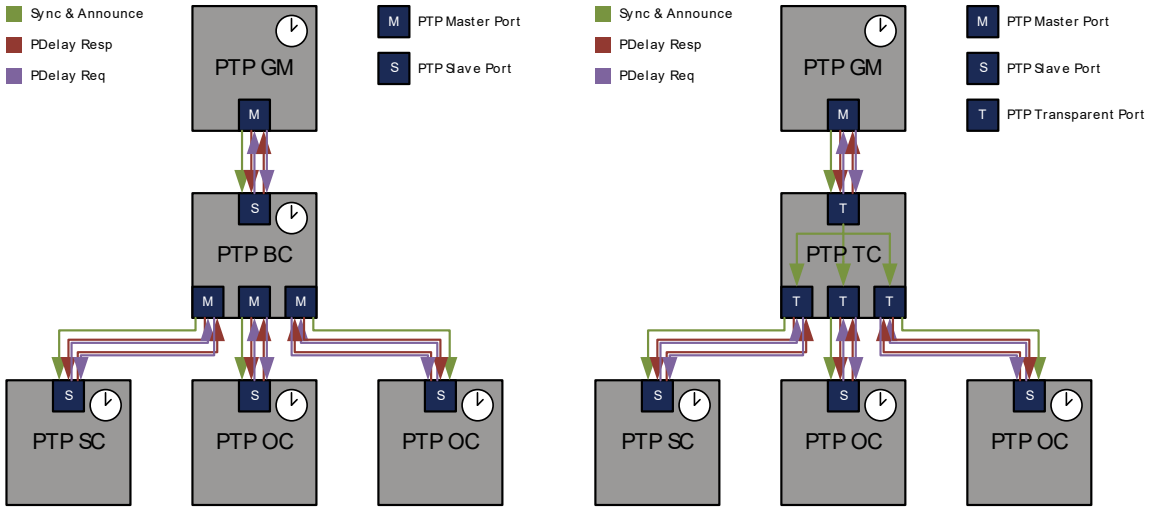

Figure 9: P2P Delay measurement with BC and TC

In the case of a network with a BC this means not only the Slaves but also the Grandmaster measure the path directly to the BC. The BC itself measures the Delay to the Grandmaster and synchronizes to it, and it also measures the Delay to all other nodes connected to it. Since the Sync, Announce and Delay messages always end and are sourced at the BC, there is no correction needed in the Messages. Each port only receives the PDelay Req and PDelay Resp messages from the node directly connected to it but this also means that PDelay Req messages have to be answered with PDelay Resp messages by each port in the network participating in PTP.

In the case of a network with a P2P TC, things work a bit differently. Unlike in the E2E case the TC measures in P2P like the BC the Delay to all other nodes (Slaves and Master) and answers all PDelay Req messages on each port. Since the PDelay messages have a Link-Local destination MAC address they will not be passed through a Switch or Router. Therefore similar as with a BC the PDelay messages end at and are sourced by the TC. The difference to E2E delay message handling gets clear when it comes to Sync messages. When a Sync message is received at the TC the Delay measured on this specific port is added to the Correction Field of this Sync message. When the frame is leaving the TC it adds the residence also to the Correction Field and forwards it to the Slave. So for a Slave the whole delay of

the Frame up to its last neighbor is summed up in the Correction field. Together with its own measured delay it gets the whole Delay that this frame has faced in the transmission from the Master to the Slave port.
A Slave can subtract this Correction together with its Delay value from its T2 timestamp and gets from there the Offset from the master.

2.2.2.3 E2E vs. P2P

They main advantage of E2E is that it works with non-PTP aware Switches (legacy or not feasible), which is often the case in Telecom or Office environments and that an E2E TC can be implemented very easily. Other than that E2E delay measurements has only drawbacks compared to P2P: E2E creates more network load and CPU load on the Master which means it doesn't scale well. E2E cannot react fast on Master switches, since it first has to measure the whole Delay chain again, where with P2P an immediate switchover can happen because of pre-measured delays and summing up of Delay and Residence values in the case of TCs. Also E2E measurement is not the preferred mechanism for Redundancy protocols like HSR and PRP and Ring topologies because this would require that Sync and Delay messages take the same way which is not always given.
So in general, whenever possible P2P delay measurement is the preferred mechanism.

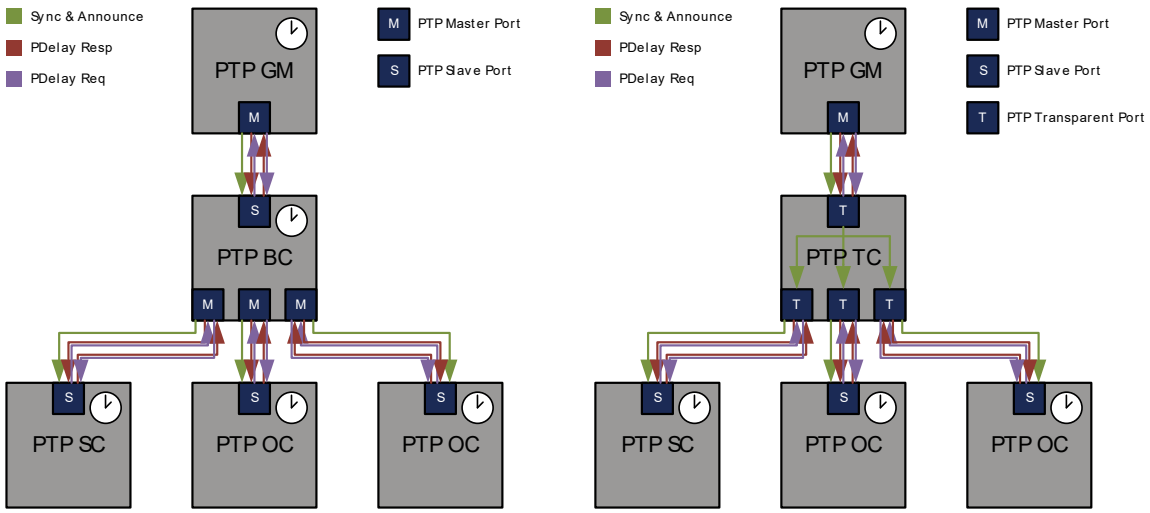

Figure 10: Unicast Scheme

2.2.3 Unicast vs. Multicast

Most PTP implementations work in a Multicast scheme. The advantage is, that no configuration is required and reconfiguration runs without interaction.

Especialy some ITU profiles however use the Unicast scheme, where fixed roles are defined. Unicast requires the configuration of the Slave to know the Master. Also Unicast requires a registration scheme so the Master knows who to serve (this is done via PTP Signaling Messages). Also Unicast potentially generates much more traffic if the network does not support PTP or Transparent clocks are used. Also Unicast generates a significant load on the Grandmaster which means only a limited number of Slaves can be served.
The reasons why ITU decided to use Unicast are mainly that they wanted to use legacy Network infastructure and Multicast is often not supported in Telecom networks.

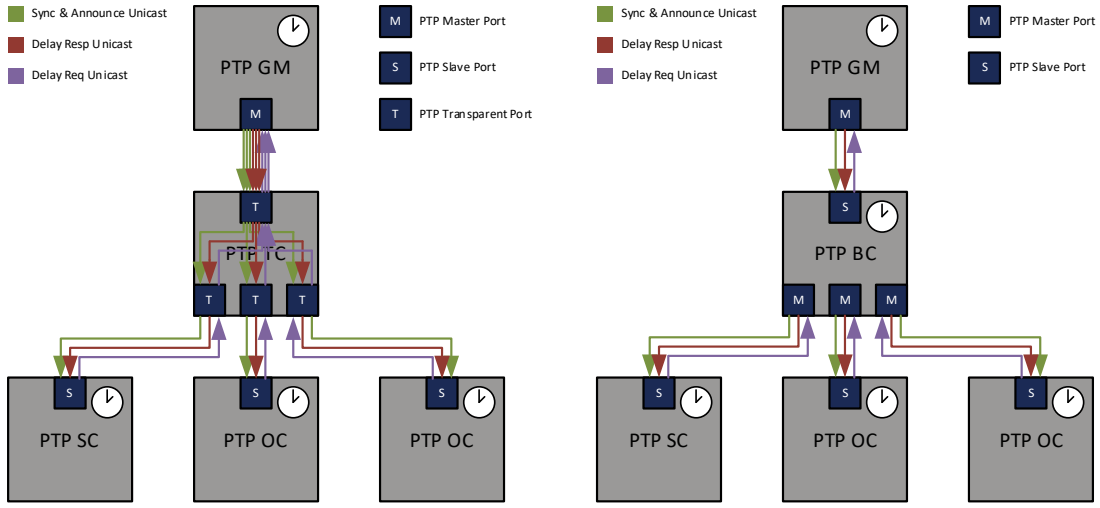

Figure 11: P2P Delay measurement with BC and TC

2.2.4 Profiles

PTP comes in different flavors (Profiles), depending on the environment it shall be used in. Profiles define communication medium mappings (Ethernet, Profinet, and IR etc.), message rates, the delay mechanisms, default values of datasets and sometimes much more:

- Default Profile uses either Layer 2 or 3 with Multicast and either E2E or P2P Delay mechanism
- Power Profile uses Layer 2 with Multicast and P2P Delay mechanism and additional TLV
- Utility Profile uses also Layer 2 with Multicast and P2P Delay mechanism in combination with Redundancy Protocols like HSR or PRP

- TSN Profile uses Layer 2 with Multicast (with all PDelay MAC addresses) and P2P Delay mechanism, high Sync message rates and signaling messages and TC like BCs with syntonization
- ITU Profile 8265.1 uses Layer 3 with Unicast and E2E Delay mechanism (optional) with potentially no PTP support of the Network and a limited BMC, normaly only frequency adjustments are done
- ITU Profile 8275.1 uses Layer 2 with Multicast and E2E Delay mechanism with full PTP support and SyncE support of the Network and an adapted BMC
- ITU Profile 8275.2 uses Layer 3 with Unicast and E2E Delay mechanism with partial PTP support of the Network and an adapted BMC
- ...

There are other Profiles with other feature sets and mappings. Some Profiles are subsets of the Default Profile and compatible, some are supersets and therefor incompatible with other Profiles. This makes interoperability difficult.
So if you want to use PTP, first it is important to choose the right Profile for your application and second to make sure that all devices in the network support the chosen Profile

2.3 Accuracy

The accuracy of the synchronization depends highly on the precision of the timestamps.
They should reflect the send and receive time as precise as possible. The slave's Offset and Delay calculations are based on the difference of timestamps taken at two different places. Therefore, the two clocks should use the same scale, i.e. the same frequency.
This is achieved by Drift compensation: the Slave’s clock rate is accelerated or slowed down by a control loop. A slightly different frequency will degrade the result.
It is assumed that the propagation Delay is the same for both directions. At a first glance, this is the case with an Ethernet link.
In the long run, conditions may change due to reconfiguration or environmental conditions (temperature).
How fast the clocks can react depends on the frequency of sync and delay measurement and the dynamic behavior of the servos controlling the Slave clock.

To sum things up, the achievable accuracy depends on:

- Timestamp accuracy
- Clock stability
- Sync interval
- Clock control loop characteristics
- Drift compensated clocks (i.e. adjusted time base in Master and Slave clocks)
- The communication channel symmetry (i.e. same delay in both directions and constant over a longer period of time)

2.3.1 Timestamp accuracy

As just stated timestamp accuracy is the key to high accuracy. PTP timestamp support can be implemented at different layers with decrease in accuracy in the higher layers. For this solution a timestamp point between MAC and PHY (on MII) was chosen to get the best possible accuracy without implementing PHY functionality This interface is perfect for the use of FPGAs since it is a strictly digital interface, standardized and has only a low frequency requirement. This interface can either be intercept if one-step support is desired or passively listened if two-step support is sufficient. For this implementation the FPGA is intercepting the Path between MAC and PHY to provide one-step support.

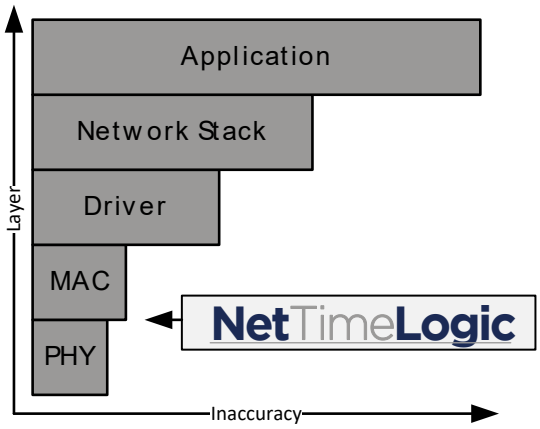

Figure 12: Timestamp Incauracy in the different Layers

Not only PTP timestamping but also frame generation, handling and servo loops can be done in different stages. Often a solution is to have the timestamping at a very low layer and all other things in the application layer which implies that a MAC, CPU and Operating System with Drivers a Network Stack and a PTP application is in place. NetTimeLogic's approach is different: NetTimeLogic's PTP cores are completely implemented in an FPGA means that all the upper layers after the PHY

are not required. This decreases the complexity and dependencies between the different layers and allows running the system without CPU.

3 Register Set

This is the register set of the PTP Ordinary Clock. It is accessible via AXI4Lite Memory Mapped. All registers are 32bit wide, no burst access, no unaligned access, no byte enables, no timeouts are supported. Register address space is not contiguous. Register addresses are only offsets in the memory area where the core is mapped in the AXI inter connects. Non existing register access in the mapped memory area is answered with a slave decoding error.

3.1 Register Overview

Registerset Overview			
Name	Description	Offset	Access
Ptp OcControl Reg	OC Enable Control Register	0x00000000	RW
Ptp OcStatus Reg	OC Error Status Register	0x00000004	WC
Ptp OcVersion Reg	OC Version Register	0x0000000C	RO
Ptp OcNrOfUnicastEntries Reg	OC Number of Unicast Entries Register	0x00000010	RO
Ptp OcConfigControl Reg	OC Configuration Control Register	0x00000080	RW
Ptp OcConfigProfile Reg	OC Configuration Profile Register	0x00000084	RW
Ptp OcConfigVlanDscp Reg	OC Configuration VLAN Register	0x00000088	RW
Ptp OcConfigIp Reg	OC Configuration IPv6 0 and Ipv4 Register	0x0000008C	RW
Ptp OcConfigIpv61 Reg	OC Configuration IPv6 1 Register	0x00000090	RW
Ptp OcConfigIpv62 Reg	OC Configuration IPv6 2 Register	0x00000094	RW
Ptp OcConfigIpv63 Reg	OC Configuration IPv6 3 Register	0x00000098	RW
Ptp OcConfigLpFilter Reg	OC Configuration Lucky Packet Filter Register	0x0000009C	RW
Ptp OcDefaultDsControl Reg	OC Default Dataset Control Register	0x00000100	RW
Ptp OcDefaultDs1 Reg	OC Default Dataset 1 Register	0x00000104	RW
Ptp OcDefaultDs2 Reg	OC Default Dataset 2 Register	0x00000108	RW
Ptp OcDefaultDs3 Reg	OC Default Dataset 3 Register	0x0000010C	RW

Ptp OcDefaultDs4 Reg	OC Default Dataset 4 Register	0x00000110	RW
Ptp OcDefaultDs5 Reg	OC Default Dataset 5 Register	0x00000114	RW
Ptp OcDefaultDs6 Reg	OC Default Dataset 6 Register	0x00000118	RW
Ptp OcDefaultDs7 Reg	OC Default Dataset 7 Register	0x0000011C	RO
Ptp OcDefaultDs8 Reg	OC Default Dataset 8 Register	0x00000120	RW
Ptp OcPortDsControl Reg	OC Port Dataset Control Register	0x00000200	RW
Ptp OcPortDs1 Reg	OC Port Dataset 1 Register	0x00000204	RO
Ptp OcPortDs2 Reg	OC Port Dataset 2 Register	0x00000208	RO
Ptp OcPortDs3 Reg	OC Port Dataset 3 Register	0x0000020C	RO
Ptp OcPortDs4 Reg	OC Port Dataset 4 Register	0x00000210	RW
Ptp OcPortDs5 Reg	OC Port Dataset 5 Register	0x00000214	RW
Ptp OcPortDs6 Reg	OC Port Dataset 6 Register	0x00000218	RW
Ptp OcPortDs7 Reg	OC Port Dataset 7 Register	0x0000021C	RW
Ptp OcPortDs8 Reg	OC Port Dataset 8 Register	0x00000220	RW
Ptp OcPortDs9 Reg	OC Port Dataset 9 Register	0x00000224	RW
Ptp OcCurrentDsControl Reg	OC Current Dataset Control Register	0x00000300	RW
Ptp OcCurrentDs1 Reg	OC Current Dataset 1 Register	0x00000304	RO
Ptp OcCurrentDs2 Reg	OC Current Dataset 2 Register	0x00000308	RO
Ptp OcCurrentDs3 Reg	OC Current Dataset 3 Register	0x0000030C	RO
Ptp OcCurrentDs4 Reg	OC Current Dataset 4 Register	0x00000310	RO
Ptp OcCurrentDs5 Reg	OC Current Dataset 5 Register	0x00000314	RO
Ptp OcParentDsControl Reg	OC Parent Dataset Control Register	0x00000400	RW
Ptp OcParentDs1 Reg	OC Parent Dataset 1 Register	0x00000404	RO
Ptp OcParentDs2 Reg	OC Parent Dataset 2 Register	0x00000408	RO
Ptp OcParentDs3 Reg	OC Parent Dataset 3 Register	0x0000040C	RO
Ptp OcParentDs4 Reg	OC Parent Dataset 4 Register	0x00000410	RO
Ptp OcParentDs5 Reg	OC Parent Dataset 5 Register	0x00000414	RO
Ptp OcParentDs6 Reg	OC Parent Dataset 6 Register	0x00000418	RO
Ptp OcParentDs7 Reg	OC Parent Dataset 7 Register	0x0000041C	RO
Ptp OcParentDs8 Reg	OC Parent Dataset 8 Register	0x00000420	RO

Ptp OcParentDs9 Reg	OC Parent Dataset 9 Register	0x00000424	RO
Ptp OcTimePropertiesDsControl Reg	OC Time Properties Dataset Control Register	0x00000500	RW
Ptp OcTimePropertiesDs1 Reg	OC Time Properties Datasett 1 Register	0x00000504	RW
Ptp OcTimePropertiesDs2 Reg	OC Time Properties Datasett 2 Register	0x00000508	RW
Ptp OcTimePropertiesDs3 Reg	OC Time Properties Datasett 3 Register	0x0000050C	RW
Ptp OcTimePropertiesDs4 Reg	OC Time Properties Datasett 4 Register	0x00000510	RW
Ptp OcTimePropertiesDs5 Reg	OC Time Properties Datasett 5 Register	0x00000514	RW
Ptp OcTimePropertiesDs6 Reg	OC Time Properties Datasett 6 Register	0x00000518	RW
Ptp OcTimePropertiesDs7 Reg	OC Time Properties Datasett 7 Register	0x0000051C	RW
Ptp OcTimePropertiesDs8 Reg	OC Time Properties Datasett 8 Register	0x00000520	RW
Ptp OcTimePropertiesDs9 Reg	OC Time Properties Datasett 9 Register	0x00000524	RW
Ptp OcUnicastDsControl Reg	OC Unicast Dataset Control Register	0x00000600	RW
Ptp OcUnicastDs1 Reg	OC Unicast Dataset 1 Register	0x00000604	RW
Ptp OcUnicastDs2 Reg	OC Unicast Dataset 2 Register	0x00000608	RW
Ptp OcUnicastDs3 Reg	OC Unicast Dataset 3 Register	0x0000060C	RW
Ptp OcUnicastDs4 Reg	OC UnicastDataset 4 Register	0x00000610	RW
Ptp OcUnicastDs5 Reg	OC Unicast Dataset 5 Register	0x00000614	RW
Ptp OcUnicastDs6 Reg	OC Unicast Dataset 6 Register	0x00000618	RW
Ptp OcUnicastDs7 Reg	OC Unicast Dataset 7 Register	0x0000061C	RW
Ptp OcUnicastDs8 Reg	OC Unicast Dataset 8 Register	0x00000620	RW
Ptp OcUnicastDs9 Reg	OC Unicast Dataset 9 Register	0x00000624	RW
Ptp OcUnicastDs10 Reg	OC Unicast Dataset 10 Register	0x00000628	RW
Ptp OcUnicastDs11 Reg	OC Unicast Dataset 11 Register	0x0000062C	RW
Ptp OcUnicastDs12 Reg	OC Unicast Dataset 12 Register	0x00000630	RW

Table 4: Register Set Overview

3.2 Register Descriptions

3.2.1 General

3.2.1.1 PTP OC Control Register

Used for general control over the PTP Ordinary Clock, all configurations on the core shall only be done when disabled.

Ptp OcControl Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															ENABLE
RO																															RW
Reset: 0x00000000																															
Offset: 0x0000																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
ENABLE	Enable	Bit: 0	RW

3.2.1.2 PTP OC Status Register

Shows the current status of the PTP Ordinary Clock.

Ptp OcStatus Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															ERROR
RO																															WC
Reset: 0x00000000																															
Offset: 0x0004																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
ENABLE	Error (sticky)	Bit: 0	WC

3.2.1.3 PTP OC Version Register

Version of the IP core, even though is seen as a 32bit value, bits 31 down to 24 represent the major, bits 23 down to 16 the minor and bits 15 down to 0 the build numbers.

Ptp OcVersion Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																															
RO																															
Reset: 0xXXXXXXXX																															
Offset: 0x000C																															

Name	Description	Bits	Access
VERSION	Version of the core	Bit: 31:0	RO

3.2.1.1 PTP OC Number of Unicast Entries

The maximum number of Unicast Entries.

Ptp OcVersion Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																NR_OF_ENTRIES															
RO																															
Reset: 0x0000XXXX																															
Offset: 0x0010																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
NR_OF_ENTRIES	Max Number of Unicast Entries in the Table 0-4096	Bit: 15:0	RO

3.2.1.2 PTP OC Config Control Register

Configuration valid bits, used to mark the corresponding fields as valid.

Ptp OcConfigControl Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											LPFILTER_VAL	DSCP_VAL	IP_VAL	VLAN_VAL	PROFILE_VAL
RO																											RW	RW	RW	RW	RW
Reset: 0x00000000																															
Offset: 0x0080																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:5	RO
LPFILTER_VAL	Lucky Packet Filter valid (autocleared)	Bit: 4	RW
DSCP_VAL	DSCP valid (autocleared)	Bit: 3	RW
IP_VAL	IP valid (autocleared)	Bit: 2	RW
VLAN_VAL	VLAN valid (autocleared)	Bit: 1	RW
PROFILE_VAL	Profile valid (autocleared)	Bit: 0	RW

3.2.1.3 PTP OC Config Profile Register

PTP profile to run, changing this will automatically change clock parameters to the default values of the corresponding profile, these parameters can be overwritten afterwards. For the Default Profile also the layer mapping and delay mechanism can be chosen.

Ptp OcConfigProfile Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						DELAY_E2E_UNICAST	DELAY_MECHANISM	AUTH_UNICAST_SLAVE	DISABLE_OFFSET_COR	MASTER_SLAVE_ONLY		-		LAYER		-						SIGNALING	TWO_STEP	-								PROFILE	
RO						RW	RW	RW	RW	RW		RO		RW		RO						RW	RW	RO						RW	
Reset: 0x00000000																															
Offset: 0x0084																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:26	RO
DELAY_E2E_UNICAST	Delay E2E Unicast (0 Multicast, 1 Unicast)	Bit:25	RW
DELAY_MECHANISM	Delay Mechanism (0 P2P, 1 E2E)	Bit:24	RW
AUTH_UNICAST_SLAVE	Only allow Unicast Slaves previously added to the Unicast Dataset (IP required only and valid)	Bit:23	RW
DISABLE_OFFSET_COR	Disable Offset Correction	Bit:22	RW

MASTER_SLAVE_ONLY	Master or Slave Only, 0 = both, 1 = Slave Only, 2 = Master Only	Bit:21:20	RW
-	Reserved, read 0	Bit:19:18	RO
LAYER	Layer (0 802.3, 1 IPv4, 2 IPv6)	Bit:17:16	RW
-	Reserved, read 0	Bit:15:10	RO
SIGNALING	Signaling	Bit:9	RW
TWO_STEP	TwoStep for Sync and Pdelay	Bit:8	RW
-	Reserved, read 0	Bit:7:3	RO
PROFILE	Profile (0 Default Profile, 1 Power Profile, 2 Utility Profile, 3 TSN Profile, 4 ITU G8265.1 Profile, 5 ITU G8275.1 Profile, 6 ITU G8275.2 Profile)	Bit:2:0	RW

3.2.1.4 PTP OC Config Vlan and DSCP Register

VLAN for 802.3q priority tagging or virtual networks. VLAN can be enabled or disabled for Power, Utility and TSN Profile. In Default Profile this is ignored.
Additionaly for Ipv4 the DSCP can be defined.

Ptp OcConfigVlanDscp Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-						DSCP						-			VLAN_EN	VLAN															
RO						RW						RO			RW	RW															
Reset: 0x00000000																															
Offset: 0x0088																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit: 31:26	RO
DSCP	DSCP Classification	Bit: 25:20	RW
-	Reserved, read 0	Bit: 19:17	RO
VLAN_EN	VLAN enable (0 disabled, 1 enabled)	Bit: 16	RW
VLAN	VLAN (as in the TAG, combined Prio and VID)	Bit: 15:0	RW

3.2.1.5 PTP OC Config IP Register

IP address of the node. Used as source IP if in Default Profile and Layer 3 mappings, otherwise ignored. LSB is transferred first on the network. This is the full IP address for IPv4 and the highest part of IPv6

Ptp OcConfigIp Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(3)								IP(2)								IP(1)								IP(0)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x008C																															

Name	Description	Bits	Access
IP(3)	IP Byte 3	Bit:31:24	RW
IP(2)	IP Byte 2	Bit:23:16	RW
IP(1)	IP Byte 1	Bit:15:8	RW
IP(0)	IP Byte 0	Bit:7:0	RW

3.2.1.6 PTP OC Config IP V6 1 Register

IPv6 address of the node. Used as source IP if in Default Profile and Layer 3 IPv6 mapping, otherwise ignored. LSB is transferred first on the network.

Ptp OcConfigIpv61 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(7)								IP(6)								IP(5)								IP(4)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0090																															

Name	Description	Bits	Access
IP(7)	IP Byte 7 (Ipv6 only)	Bit:31:24	RW
IP(6)	IP Byte 6 (Ipv6 only)	Bit:23:16	RW
IP(5)	IP Byte 5 (Ipv6 only)	Bit:15:8	RW
IP(4)	IP Byte 4 (Ipv6 only)	Bit:7:0	RW

3.2.1.7 PTP OC Config IP V6 2 Register

IPv6 address of the node. Used as source IP if in Default Profile and Layer 3 IPv6 mapping, otherwise ignored. LSB is transferred first on the network.

Ptp OcConfigIpv62 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(11)								IP(10)								IP(9)								IP(8)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0094																															

Name	Description	Bits	Access
IP(11)	IP Byte 11 (Ipv6 only)	Bit:31:24	RW
IP(10)	IP Byte 10 (Ipv6 only)	Bit:23:16	RW
IP(9)	IP Byte 9 (Ipv6 only)	Bit:15:8	RW
IP(8)	IP Byte 8 (Ipv6 only)	Bit:7:0	RW

3.2.1.8 PTP OC Config IP V6 3 Register

IPv6 address of the node. Used as source IP if in Default Profile and Layer 3 IPv6 mapping, otherwise ignored. LSB is transferred first on the network.

Ptp OcConfigIpv63 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(15)								IP(14)								IP(13)								IP(12)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0098																															

Name	Description	Bits	Access
IP(15)	IP Byte 15 (Ipv6 only)	Bit:31:24	RW
IP(14)	IP Byte 14 (Ipv6 only)	Bit:23:16	RW
IP(13)	IP Byte 13 (Ipv6 only)	Bit:15:8	RW
IP(12)	IP Byte 12 (Ipv6 only)	Bit:7:0	RW

3.2.1.9 PTP OC Config Lucky Packet Filter Register

Configuration of the Lucky Packet Filter. The most important configuration is the Window size. It applies to Sync and E2E Delay Request. The Window defines to find the “fastest” Packet within the window. The value shall be choosen that approximately 1/s an adjustment shall be made.

Ptp OcConfigProfile Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			LP_WINDOW													-														LP_ONE_SAMPLE	LP_ENABLE
RO			RW													RO														RW	RW
Reset: 0x0XXX0000																															
Offset: 0x009C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:29	RO
LP_WINDOW	Lucky Packet Filter Window in number of Samples (initial and max size limited by generic LuckyPacketFilter-Samples_Gen)	Bit:28:16	RW
-	Reserved, read 0	Bit:15:2	RO
LP_ONE_SAMPLE	One Sample Per Window only enforced	Bit:1	RW

LP_ENABLE	Enable Lucky Packet Filter	Bit:0	RW

3.2.2 Default Dataset

Parameters from the PTP Default Dataset according to IEEE1588-2019 Clause 8.2.1.

3.2.2.1 PTP OC Default Dataset Control Register

Configuration valid bits, used to mark the corresponding fields as valid. Additional flags to snapshot the current Default Dataset: set READ flag and check for READ_DONE flag set.

Ptp OcDefaultDsControl Reg

Reg Description

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ																						LOCAL_PRIO_VAL	MAX_STEPS_REM_VAL	GM_TIME_INAC_VAL	GM_ID_VAL	PRIORITY2_VAL	PRIORITY1_VAL	CLOCK_QUALITY_VAL	DOMAIN_NR_VAL	CLOCK_ID_VAL
RO	RW	RO																					RW	RW	RW	RW	RW	RW	RW	RW	RW

Reset: 0x00000000

Offset: 0x0100

Name	Description	Bits	Access
READ_DONE	Default Dataset was read	Bit: 31	RO

READ	Read Default Dataset (autocleared)	Bit: 30	RW
-	Reserved, read 0	Bit 29:9	RO
LOCAL_PRIO_VAL	Local Priority valid (autocleared)	Bit: 8	RW
MAX_STEPS_REM_VAL	Max Steps Removed valid (autocleared)	Bit: 7	RW
GM_TIME_INAC_VAL	Grandmaster Time Inacuracy valid (autocleared)	Bit: 6	RW
GM_ID_VAL	Grandmaster Short Identity valid (autocleared)	Bit: 5	RW
PRIORITY2_VAL	Priority2 valid (autocleared)	Bit: 4	RW
PRIORITY1_VAL	Priority1 valid (autocleared)	Bit: 3	RW
CLOCK_QUALITY_VAL	Clock Quality valid (autocleared)	Bit: 2	RW
DOMAIN_NR_VAL	Domain Number valid (autocleared)	Bit: 1	RW
CLOCK_ID_VAL	Clock Identity valid (autocleared)	Bit: 0	RW

3.2.2.2PTP OC Default Dataset 1 Register

First 4 bytes of the Clock Identity of the node. CLOCK_ID(2..0) are also the first bytes of the MAC address of the node, LSB first. E.g. 0x01234567 => MAC: 67:45:32:XX:XX:XX. MAC48 to UID64 CLOCK_ID(3) should be set to FF.

Ptp OcDefaultDs1 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLOCK_ID(3)								CLOCK_ID(2)								CLOCK_ID(1)								CLOCK_ID(0)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0104																															

Name	Description	Bits	Access
CLOCK_ID(3)	Clock ID Byte 3	Bit:31:24	RW
CLOCK_ID(2)	Clock ID Byte 2	Bit:23:16	RW
CLOCK_ID(1)	Clock ID Byte 1	Bit:15:8	RW
CLOCK_ID(0)	Clock ID Byte 0	Bit:7:0	RW

3.2.2.3PTP OC Default Dataset 2 Register

Second 4 bytes of the Clock Identity of the node. CLOCK_ID(7..5) are also the last bytes of the MAC address of the node, LSB first. E.g. 0x01234567 => MAC: XX:XX:XX:45:23:01. MAC48 to UID64 CLOCK_ID(4) should be set to FE.

Ptp OcDefaultDs2 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLOCK_ID(7)								CLOCK_ID(6)								CLOCK_ID(5)								CLOCK_ID(4)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0108																															

Name	Description	Bits	Access
CLOCK_ID(7)	Clock ID Byte 7	Bit:31:24	RW
CLOCK_ID(6)	Clock ID Byte 6	Bit:23:16	RW
CLOCK_ID(5)	Clock ID Byte 5	Bit:15:8	RW
CLOCK_ID(4)	Clock ID Byte 4	Bit:7:0	RW

3.2.2.4 PTP OC Default Dataset 3 Register

Domain to run on, only one is supported and Priorities.

Ptp OcDefaultDs3 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIORITY1								PRIORITY2								MAX_STEPS_REM								DOMAIN_NR							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x010C																															

Name	Description	Bits	Access
PRIORITY1	Priority 1	Bit:31:24	RW
PRIORITY2	Priority 2	Bit:23:16	RW
MAX_STEPS_REM	Maximum Steps Removed	Bit:15:8	RW
DOMAIN_NR	Domain Number	Bit: 7:0	RW

3.2.2.5PTP OC Default Dataset 4 Register

PTP Clock Parameters, these are set to the corresponding default parameters according to the used Profile.

Ptp OcDefaultDs4 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLOCK_CLASS								CLOCK_ACCURACY								CLOCK_VARIANCE															
RW								RW								RW															
Reset: 0x00000000																															
Offset: 0x0110																															

Name	Description	Bits	Access
CLOCK_CLASS	Clock Quality, Clock Class	Bit:31:24	RW
CLOCK_ACCURACY	Clock Quality, Clock Accuracy	Bit:23:16	RW
CLOCK_VARIANCE	Clock Quality, Clock Variance	Bit:15:0	RW

3.2.2.6PTP OC Default Dataset 5Register

Grandmaster Short Identity of the Node. Only used for Power Profile, else ignored.

Ptp OcDefaultDs5 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-																GM_ID															
RO																RW															
Reset: 0x00000000																															
Offset: 0x0114																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
GM_ID	Grandmaster Short ID according to PowerProfile	Bit: 15:0	RW

3.2.2.7PTP OC Default Dataset 6 Register

Grandmaster Time Inaccuracy, known inaccuracy to the norm second. Only used for Power Profile, else ignored.

Ptp OcDefaultDs6 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_TIME_INAC																														
RW																														
Reset: 0x00000000																														
Offset: 0x0118																														

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
GM_TIME_INAC	Grandmaster Time Inaccuracy according to PowerProfile	Bit: 31:0	RW

3.2.2.8PTP OC Default Dataset 7 Register

Number of ports, for an OC this is always 1.

Ptp OcDefaultDs7 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																NUMBER_OF_PORTS															
RO																RO															
Reset: 0x00000001																															
Offset: 0x011C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
NUMBER_OF_PORTS	Number of ports of the OC (always 1)	Bit: 15:0	RO

3.2.2.9PTP OC Default Dataset 8 Register

Local Priority for ITU Profiles.

Ptp OcDefaultDs7 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								LOCAL_PRIORITY							
RO																								RW							
Reset: 0x00000080																															
Offset: 0x0120																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:8	RO
LOCAL_PRIORITY	Local Priority (for ITU Profiles)	Bit: 7:0	RW

3.2.3 Port Dataset

Parameters from the PTP Port Dataset according to IEEE1588-2019 Clause 8.2.5.

3.2.3.1 PTP OC Port Dataset Control Register

Flags to snapshot the current Port Dataset: set READ flag and check for READ_DONE flag set.

Ptp OcPortDsControl Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ																										LOCAL_PRIO_VAL	MAX_PDELAY_VAL	ASYMMETRY_VAL	MSG_INTERVAL_VAL	DELAY_MECHANISM VAL
RO	RW	RO																									RW	RW	RW	RW	RW
Reset: 0x00000000																															
Offset: 0x0200																															

Name	Description	Bits	Access
READ_DONE	Port Dataset was read	Bit: 31	RO
READ	Read Port Dataset (autocleared)	Bit: 30	RW
-	Reserved, read 0	Bit 29:5	RO
LOCAL_PRIO_VAL	Local Priority valid (autocleared)	Bit: 4	RW

MAX_PDELAY_VAL	Maximum Peer Delay valid (autocleared)	Bit: 3	RW
ASYMMETRY_VAL	Asymmetry valid (autocleared)	Bit: 2	RW
MSG_INTERVAL_VAL	Message Intervals valid (autocleared)	Bit: 1	RW
DELAY_MECHANISM_VAL	Delay Mechanism valid (autocleared)	Bit: 0	RW

3.2.3.2PTP OC Port Dataset 1 Register

Higher 32 bits of the current measured Peer Delay on this port. This is in scaled nanoseconds format.

Ptp OcPortDs1 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEER_DELAY(63:32)																														
RO																														
Reset: 0x00000000																														
Offset: 0x0204																														

Name	Description	Bits	Access
PEER_DELAY(63:32)	Peer Mean Path Delay higher 32bits	Bit: 31:0	RO

3.2.3.3PTP OC Port Dataset 2 Register

Lower 32 bits of the current measured Peer Delay on this port. This is in scaled nanoseconds format.

Ptp OcPortDs2 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEER_DELAY(31:0)																														
RO																														
Reset: 0x00000000																														
Offset: 0x0208																														

Name	Description	Bits	Access
PEER_DELAY(31:0)	Peer Mean Path Delay lower 32bits	Bit: 31:0	RO

3.2.3.4 PTP OC Port Dataset 3 Register

Port State as defined in IEEE1588

Ptp OcPortDs3 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								PORT STATE							
RO																								RO							
Reset: 0x00000000																															
Offset: 0x020C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:8	RO
PORT_STATE	Port state as defined in IEEE1588	Bit: 7:0	RO

3.2.3.5PTP OC Port Dataset 4 Register

Signed Peer Delay Request and Delay Request interval (only available when DynamicMessageRatesSupport_Gen is true)

Ptp OcPortDs4 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								DELAY_TIMEOUT								DELAYREQ_INTERVAL								PDELAYREQ_INTERVAL							
RO								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0210																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit 31:24	RO
DELAY_TIMEOUT	Delay timeout	Bit 23:16	RW
DELAYREQ_INTERVAL	Signed Delay Request Log interval	Bit 15:8	RW
PDELAYREQ_INTERVAL	Signed Peer Delay Request Log interval	Bit 7:0	RW

3.2.3.6PTP OC Port Dataset 5 Register

Signed Announce interval and Announce timeout (only available when DynamicMessageRatesSupport_Gen is true)

Ptp OcPortDs5 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-																ANNOUNCE_TIMEOUT								ANNOUNCE_INTERVAL							
RO																RW								RW							
Reset: 0x00000000																															
Offset: 0x0214																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
ANNOUNCE TIMEOUT	Announce timeout	Bit: 15:8	RW
ANNOUNCE_INTERVAL	Signed Announce Log interval	Bit: 7:0	RW

3.2.3.7PTP OC Port Dataset 6 Register

Signed Sync interval (only available when DynamicMessageRatesSupport_Gen is true)

Ptp OcPortDs6 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-																SYNC_TIMEOUT								SYNC_INTERVAL							
RO																RW								RW							
Reset: 0x00000000																															
Offset: 0x0218																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit 31:16	RO
SYNC_TIMEOUT	Announce timeout	Bit 15:8	RW
SYNC_INTERVAL	Signed Sync Log interval	Bit 7:0	RW

3.2.3.8PTP OC Port Dataset 7 Register

Asymmetry as signed nanoseconds.

Ptp OcPortDs7 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYMMETRY																														
RW																														
Reset: 0x00000000																														
Offset: 0x021C																														

Name	Description	Bits	Access
ASYMMETRY	Signed Asymmetry in Nanoseconds	Bit: 31:0	RW

3.2.3.9PTP OC Port Dataset 8 Register

Maximum Peer Delay which is still considered valid as nanoseconds. This is only used for IEEE802.1AS (TSN Profile)

Ptp OcPortDs7 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_PDELAY																														
RW																														
Reset: 0x00000000																														
Offset: 0x0220																														

Name	Description	Bits	Access
MAX_PDELAY	Maximum Pdelay which is still valid (only for TSN)	Bit: 31:0	RW

3.2.3.10 PTP OC Port Dataset 9 Register

Local Priority for ITU Profiles.

Ptp OcPortDs3 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								LOCAL_PRIORITY							
RO																								RW							
Reset: 0x00000000																															
Offset: 0x020C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:8	RO
LOCAL_PRIORITY	Local Priority (for ITU Profiles)	Bit: 7:0	RW

3.2.4 Current Dataset

Parameters from the PTP Current Dataset according to IEEE1588-2019 Clause 8.2.2.

3.2.4.1 PTP OC Current Dataset Control Register

Flags to snapshot the current Current Dataset: set READ flag and check for READ_DONE flag set.

Ptp OcCurrentDsControl Reg

Reg Description

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ																														
RO	RW	RO																													
Reset: 0x00000000																															
Offset: 0x0300																															

Name	Description	Bits	Access
READ_DONE	Current Dataset was read	Bit: 31	RO
READ	Read Current Dataset (autocleared)	Bit: 30	RW
-	Reserved, read 0	Bit 29:0	RO

3.2.4.2 PTP OC Current Dataset 1 Register

Number of network hops (BCs) between the Master and the Slave. If OC is Master this value is 0.

Ptp OcCurrentDs1 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-			PTSF_UNCERTAIN	PTSF_UNUSABLE	PTSF_DELAY_LOSS	PTSF_SYNC_LOSS	PTSF_ANNOUNCE_LOSS	-								STEPS_REMOVED																
RO			RO	RO	RO	RO	RO	RO								RO															
Reset: 0x00000000																															
Offset: 0x0304																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:29	RO
PTSF_UNCERTAIN	ITU PTSF Sync Uncertain	Bit: 28	RO
PTSF_UNUSABLE	ITU PTSF Unusable (not used, criteria not defined by ITU)	Bit: 27	RO
PTSF_DELAY_LOSS	ITU PTSF Delay Loss	Bit: 26	RO
PTSF_SYNC_LOSS	ITU PTSF Sync Loss	Bit: 25	RO
PTSF_ANNOUNCE_LOSS	ITU PTSF Announce Loss	Bit: 24	RO
-	Reserved, read 0	Bit:23:16	RO
STEPS_REMOVED	How many steps were removed between master and slave	Bit: 15:0	RO

3.2.4.3 PTP OC Current Dataset 2 Register

Higher 32 bits of the current measured/calculated Offset to the Master. This is in scaled nanoseconds format.

Ptp OcCurrentDs2 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET(63:32)																														
RO																														
Reset: 0x00000000																														
Offset: 0x0308																														

Name	Description	Bits	Access
OFFSET(63:32)	Offset from Master higher 32bits	Bit: 31:0	RO

3.2.4.4 PTP OC Current Dataset 3 Register

Lower 32 bits of the current measured/calculated Offset to the Master. This is in scaled nanoseconds format.

Ptp OcCurrentDs3 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET(31:0)																													
RO																													
Reset: 0x00000000																													
Offset: 0x030C																													

Name	Description	Bits	Access
OFFSET(31:0)	Offset from Master lower 32bits	Bit: 31:0	RO

3.2.4.5 PTP OC Current Dataset 4 Register

Higher 32 bits of the current measured E2E Delay. This is in scaled nanoseconds format.

Ptp OcCurrentDs4 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY(63:32)																														
RO																														
Reset: 0x00000000																														
Offset: 0x0310																														

Name	Description	Bits	Access
DELAY(63:32)	Mean Path Delay higher 32bits	Bit: 31:0	RO

3.2.4.6 PTP OC Current Dataset 5 Register

Lower 32 bits of the current measured E2E Delay. This is in scaled nanoseconds format.

Ptp OcCurrentDs5 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY(31:0)																														
RO																														
Reset: 0x00000000																														
Offset: 0x0314																														

Name	Description	Bits	Access
DELAY(31:0)	Mean Path Delay lower 32bits	Bit: 31:0	RO

3.2.5 Parent Dataset

Parameters from the PTP Parent Dataset according to IEEE1588-2019 Clause 8.2.3.

3.2.5.1 PTP OC Parent Dataset Control Register

Flags to snapshot the current Parent Dataset: set READ flag and check for READ_DONE flag set.

Ptp OcParentDsControl Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ																														
RO	R W	RO																													
Reset: 0x00000000																														
Offset: 0x0400																														

Name	Description	Bits	Access
READ_DONE	Parent Dataset was read	Bit: 31	RO
READ	Read Parent Dataset (autocleared)	Bit: 30	RW

-	Reserved, read 0	Bit 29:0	RO

3.2.5.2PTP OC Parent Dataset 1 Register

First 4 bytes of the Clock Identity of the current Master. If the OC is Master this is the same as the CLOCK_ID from the Default Dataset.

Ptp OcParentDs1 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARENT_CLOCK_ID(3)								PARENT_CLOCK_ID(2)								PARENT_CLOCK_ID(1)								PARENT_CLOCK_ID(0)							
RO								RO								RO								RO							
Reset: 0x00000000																															
Offset: 0x0404																															

Name	Description	Bits	Access
PARENT_CLOCK_ID(3)	Parent Clock ID Byte 3	Bit:31:24	RO
PARENT_CLOCK_ID(2)	Parent Clock ID Byte 2	Bit:23:16	RO
PARENT_CLOCK_ID(1)	Parent Clock ID Byte 1	Bit:15:8	RO
PARENT_CLOCK_ID(0)	Parent Clock ID Byte 0	Bit:7:0	RO

3.2.5.3PTP OC Parent Dataset 2 Register

Second 4 bytes of the Clock Identity of the current Master. If the OC is Master this is the same as the CLOCK_ID from the Default Dataset.

Ptp OcParentDs2 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARENT_CLOCK_ID(7)								PARENT_CLOCK_ID(6)								PARENT_CLOCK_ID(5)								PARENT_CLOCK_ID(4)							
RO								RO								RO								RO							
Reset: 0x00000000																															
Offset: 0x0408																															

Name	Description	Bits	Access
PARENT_CLOCK_ID(7)	Parent Clock ID Byte 7	Bit:31:24	RO
PARENT_CLOCK_ID(6)	Parent Clock ID Byte 6	Bit:23:16	RO
PARENT_CLOCK_ID(5)	Parent Clock ID Byte 5	Bit:15:8	RO
PARENT_CLOCK_ID(4)	Parent Clock ID Byte 4	Bit:7:0	RO

3.2.5.4 PTP OC Parent Dataset 3 Register

Port Number part of the Clock Identity of the current Master. If the OC is master this is 0. Priorities of the GM.

Ptp OcParentDs3 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_PRIORITY1								GM_PRIORITY2								PARENT_PORT_NR															
RO								RO								RO															
Reset: 0x00000000																															
Offset: 0x040C																															

Name	Description	Bits	Access
GM_PRIORITY1	Grandmaster Priority 1	Bit:31:24	RO
GM_PRIORITY2	Grandmaster Priority 2	Bit:23:16	RO
PORT_NR	Parent Port Number	Bit: 15:0	RO

3.2.5.5PTP OC Parent Dataset 4 Register

First 4 bytes of the Grandmaster Clock Identity of the current Grandmaster. If the OC is Master this is the same as the CLOCK_ID from the Default Dataset.

Ptp OcParentDs4 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_CLOCK_ID(3)								GM_CLOCK_ID(2)								GM_CLOCK_ID(1)								GM_CLOCK_ID(0)							
RO								RO								RO								RO							
Reset: 0x00000000																															
Offset: 0x0410																															

Name	Description	Bits	Access
GM_CLOCK_ID(3)	Grandmaster Clock ID Byte 3	Bit:31:24	RO
GM_CLOCK_ID(2)	Grandmaster Clock ID Byte 2	Bit:23:16	RO
GM_CLOCK_ID(1)	Grandmaster Clock ID Byte 1	Bit:15:8	RO
GM_CLOCK_ID(0)	Grandmaster Clock ID Byte 0	Bit:7:0	RO

3.2.5.6PTP OC Parent Dataset 5 Register

Second 4 bytes of the Grandmaster Clock Identity of the current Grandmaster. If the OC is Master this is the same as the CLOCK_ID from the Default Dataset.

Ptp OcParentDs5 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_CLOCK_ID(7)								GM_CLOCK_ID(6)								GM_CLOCK_ID(5)								GM_CLOCK_ID(4)							
RO								RO								RO								RO							
Reset: 0x00000000																															
Offset: 0x0414																															

Name	Description	Bits	Access
GM_CLOCK_ID(7)	Grandmaster Clock ID Byte 7	Bit:31:24	RO
GM_CLOCK_ID(6)	Grandmaster Clock ID Byte 6	Bit:23:16	RO
GM_CLOCK_ID(5)	Grandmaster Clock ID Byte 5	Bit:15:8	RO
GM_CLOCK_ID(4)	Grandmaster Clock ID Byte 4	Bit:7:0	RO

3.2.5.7PTP OC Parent Dataset 6 Register

Clock Parameters of the current Grandmaster. If the OC is Master this is the same as the Clock Parameters from the Default Dataset.

Ptp OcParentDs6 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_CLOCK_CLASS								GM_CLOCK_ACCURACY								GM_CLOCK_VARIANCE															
RO								RO								RO															
Reset: 0x00000000																															
Offset: 0x0418																															

Name	Description	Bits	Access
GM_CLOCK_CLASS	Grandmaster Clock Quality, Clock Class	Bit:31:24	RO
GM_CLOCK_ACCURACY	Grandmaster Clock Quality, Clock Accuracy	Bit:23:16	RO
GM_CLOCK_VARIANCE	Grandmaster Clock Quality, Clock Variance	Bit:15:0	RO

3.2.5.8PTP OC Parent Dataset 7 Register

Grandmaster Short Identity of the current Grandmaster. If the OC is Master this is the same as the GM_ID from the Default Dataset.

Ptp OcParentDs7 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																GM_ID															
RO																RO															
Reset: 0x00000000																															
Offset: 0x041C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
GM_ID	Grandmaster Short ID according to PowerProfile	Bit: 15:0	RO

3.2.5.9PTP OC Parent Dataset 8 Register

Grandmaster Time Inaccuracy of the current Grandmaster. If the OC is Master this is the TIME_INAC from the Default Dataset.

Ptp OcParentDs8 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM_TIME_INAC																															
RO																															
Reset: 0x00000000																															
Offset: 0x0420																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
GM_TIME_INAC	Grandmaster Time Inacuracy according to PowerProfile	Bit: 31:0	RO

3.2.5.10 PTP OC Parent Dataset 9 Register

Accumulated received Network Time Inaccuracy between the Slave and the Grandmaster. If the OC is Master this is the 0.

Ptp OcParentDs9 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NETWORK_TIME_INAC																														
RO																														
Reset: 0x00000000																														
Offset: 0x0424																														

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
NETWORK_TIME_INAC	Network Time Inaccuracy according to PowerProfile	Bit: 31:0	RO

3.2.6 Time Properties Dataset

Parameters from the PTP Time Properties Dataset according to IEEE1588-2019 Clause 8.2.4

3.2.6.1 PTP OC Time Properties Dataset Control Register

Configuration valid bits, used to mark the corresponding fields as valid. Additional flags to snapshot the current Time Properties Dataset: set READ flag and check for READ_DONE flag set.

Ptp OcTimePropertiesDsControl Reg																												
Reg Description																												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ																		DISP_NAME_VAL	DISP_NAME_LENG_VAL	TIME_OF_NEXT_JMP_VAL	JUMP_SECOND_VAL	CURRENT_OFFSET_VAL	TIME_SOURCE_VAL	PTP_TIMESCALE_VAL	FREQ_TRACEABLE_VAL	TIME_TRACEABLE_VAL	LEAP61_VAL	LEAP59_VAL	UTC_OFFSET_VAL_VAL	UTC_OFFSET_VAL
RO	RW	RO																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset: 0x00000000																															
Offset: 0x0500																															

Name	Description	Bits	Access
READ_DONE	Time Properties Dataset was read	Bit: 31	RO
READ	Read Time Properties Dataset (autocleared)	Bit: 30	RW

-	Reserved, read 0	Bit 29:13	RO
DISP_NAME_VAL	Display Name Valid (autocleared) (power profile)	Bit:12	RW
DISP_NAME_LENG_VAL	Diplay Name Length Valid (autocleared) (power profile)	Bit:11	RW
TIME_OF_NEXT_JMP_VAL	Time of Next Jump Valid (autocleared) (power profile)	Bit:10	RW
JUMP_SECOND_VAL	Jump in Seconds Valid (autocleared) (power profile)	Bit:9	RW
CURRENT_OFFSET_VAL	Current Offset Valid (autocleared) (power profile)	Bit:8	RW
TIME_SOURCE_VAL	Time Source Valid (autocleared)	Bit:7	RW
PTP_TIMESCALE_VAL	PTP Timescale Valid (autocleared)	Bit:6	RW
FREQ_TRACEABLE_VAL	Frequency Traceable Valid (autocleared)	Bit:5	RW
TIME_TRACEABLE_VAL	Time Traceable Valid (autocleared)	Bit:4	RW
LEAP61_VAL	Leap Second 61 Valid (autocleared)	Bit:3	RW
LEAP59_VAL	Leap Second 59 Valid (autocleared)	Bit:2	RW
UTC_OFFSET_VAL_VAL	UTC Offset Valid Valid (autocleared)	Bit:1	RW
UTC_OFFSET_VAL	UTC Offset Valid (autocleared)	Bit:0	RW

3.2.6.2PTP OC Time Properties Dataset 1 Register

Quality bits of the current Grandmaster. If the OC is Master and external synchronization is disabled, the values are set to the unknown default values according to IEEE1588-2019 Clause 9.4 else to the values provided by the external synchronization source.

Ptp OcTimePropertiesDs1 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UTC_OFFSET																-		UTC_OFFSET_VAL	LEAP59	LEAP61	TIME_TRACEABLE	FREQ_TRACEABLE	PTP_TIMESCALE	TIME_SOURCE							
RW																RO		RW	RW	RW	RW	RW	RW	RW							
Reset: 0x00000000																															
Offset: 0x0504																															

Name	Description	Bits	Access
UTC_OFFSET	Current UTC offset in seconds	Bit:31:16	RW
-	Reserved, read 0	Bit:15:14	RO
UTC_OFFSET_VAL	Current UTC offset is valid	Bit: 13	RW
LEAP59	Leap second 59 and next midnight change	Bit: 12	RW
LEAP61	Leap second 61 and next midnight change	Bit: 11	RW
TIME_TRACEABLE	Time is also phase traceable	Bit: 10	RW
FREQ_TRACEABLE	Time is Frequency traceable	Bit: 9	RW

PTP_TIMESCALE	Time is in PTP Timescale	Bit: 8	RW
TIME_SOURCE	Time Source	Bit: 7:0	RW

3.2.6.3PTP OC Time Properties Dataset 2 Register

Alternate Timescale Offset of the current Grandmaster. Only used for Power Profile. If the OC gets Master this is reset to 0, it can be changed afterwards. This is not directly used by the PTP core. Software has to calculate its Alternate Timescale from the current time.

Ptp OcTimePropertiesDs2 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT_OFFSET																														
RW																														
Reset: 0x00000000																														
Offset: 0x0508																														

Name	Description	Bits	Access
CURRENT_OFFSET	Current Offset in Seconds	Bit: 31:0	RO

3.2.6.4 PTP OC Time Properties Dataset 3 Register

Alternate Timescale Increment/Decrement of the next Jump in Seconds. Only used for Power Profile. If the OC gets Master this is reset to 0, it can be changed afterwards. This is not directly used by the PTP core. Software has to calculate its Alternate Timescale from the current time.

Ptp OcTimePropertiesDs3 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JUMP_SECONDS																														
RW																														
Reset: 0x00000000																														
Offset: 0x050C																														

Name	Description	Bits	Access
JUMP_SECONDS	Jump in Seconds	Bit: 31:0	RW

3.2.6.5 PTP OC Time Properties Dataset 4 Register

Higher 2 bytes of the Alternate Timescale Time of the next Jump in Seconds. Only used for Power Profile. If the OC gets Master this is reset to 0, it can be changed afterwards. This is not directly used by the PTP core. Software has to calculate its Alternate Timescale from the current time.

Ptp OcTimePropertiesDs4 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TIME_OF_NEXT_JUMP(47:32)															
RO																RW															
Reset: 0x00000000																															
Offset: 0x0510																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:1	RO
TIME_OF_NEXT_JUMP(47:32)	Time of next Jump higher Second part	Bit: 31:0	RW

3.2.6.6 PTP OC Time Properties Dataset 5 Register

Lower 4 bytes of the Alternate Timescale Time of the next Jump in Seconds. Only used for Power Profile. If the OC gets Master this is reset to 0, it can be changed afterwards. This is not directly used by the PTP core. Software has to calculate its Alternate Timescale from the current time.

Ptp OcTimePropertiesDs5 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME_OF_NEXT_JUMP(31:0)																														
RW																														
Reset: 0x00000000																														
Offset: 0x0514																														

Name	Description	Bits	Access
TIME_OF_NEXT_JUMP(31:0)	Time of next Jump lower Second part	Bit: 31:0	RW

3.2.6.7PTP OC Time Properties Dataset 6 Register

Display Name Length in bytes,without \0 termination of the Alternate Timescale. Only used for Power Profile. If the OC gets Master this is reset to 3 (“PTP”), it can be changed afterwards.

Ptp OcTimePropertiesDs6 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								DISPLAY_NAME_LENG							
RO																								RW							
Reset: 0x00000000																															
Offset: 0x0518																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:8	RO
DISPLAY_NAME_LENG	Diplay Name Lenght (max 12)	Bit: 7:0	RO

3.2.6.8 PTP OC Time Properties Dataset 7 Register

First 4 characters of Display Name in ASCII text of the Alternate Timescale. Only used for Power Profile. If the OC gets Master this is reset to “PTP”, it can be changed afterwards.

Ptp OcTimePropertiesDs7 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPLAY_NAME(3)								DISPLAY_NAME(2)								DISPLAY_NAME(1)								DISPLAY_NAME(0)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x051C																															

Name	Description	Bits	Access
DISPLAY_NAME(3:0)	Display Name Byte 0 to 3	Bit: 31:0	RW

3.2.6.9 PTP OC Time Properties Dataset 8 Register

Second 4 characters of Display Name in ASCII text of the Alternate Timescale. Only used for Power Profile. If the OC gets Master this is reset to \0, it can be changed afterwards.

Ptp OcTimePropertiesDs8 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPLAY_NAME(7)								DISPLAY_NAME(6)								DISPLAY_NAME(5)								DISPLAY_NAME(4)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0520																															

Name	Description	Bits	Access
DISPLAY_NAME(3:0)	Display Name Byte 4 to 7	Bit: 31:0	RW

3.2.6.10 PTP OC Time Properties Dataset 9 Register

Third 4 characters of Display Name in ASCII text of the Alternate Timescale. Only used for Power Profile. If the OC gets Master this is reset to \0, it can be changed afterwards.

Ptp OcTimePropertiesDs9 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPLAY_NAME(11)								DISPLAY_NAME(10)								DISPLAY_NAME(9)								DISPLAY_NAME(8)							
RW								RW								RW								RW							
Reset: 0x00000000																													
Offset: 0x0524																													

Name	Description	Bits	Access
DISPLAY_NAME(3:0)	Display Name Byte 8 to 11	Bit: 31:0	RW

3.2.7 Unicast Dataset

Unicast Dataset containing potential Masters and registered Slaves in ITU Profiles.

3.2.7.1 PTP OC Unicast Dataset Control Register

Configuration valid bits, used to mark the corresponding fields as valid. Additional flags to read and write specific entries in the table (either addressed by Entry Nr or IP).

Ptp OcUnicastDsControl Reg																												
Reg Description																												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
READ_DONE	READ	WRITE_DONE	WRITE	ENTRY_NR												CLEAR	ACCESS_MODE	ACCESS_OK	REFRESH_CLEAR_OR_DUR	REFRESH_VAL	DELAY_DURATION_VAL	DELAY_INTERVAL_VAL	SYNC_DURATION_VAL	SYNC_INTERVAL_VAL	AN-NOUNCE_DURATION_VAL	ANNOUNCE_INTERVAL_VAL	PRIORITY_VAL	IP_VAL	MAC_VAL	FLAGS_VAL	ENTRY_VAL
RW	RW	RW	RW	RO												RW	RW	RO	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset: 0x00000000																															
Offset: 0x0600																															

Name	Description	Bits	Access
READ_DONE	Unicast Dataset Entry was read	Bit: 31	RO

READ	Read Unicast Dataset Entry (autocleared)	Bit: 30	RW
WRITE_DONE	Unicast Dataset Entry was written	Bit: 29	RO
WRITE	Write Unicast Dataset Entry (autocleared)	Bit: 28	RW
ENTRY_NR	Entry in the Unicast Dataset to read or write	Bit: 27:16	RW
CLEAR	Clear the Unicast Dataset	Bit: 15	RW
ACCESS_MODE	Access mode on the table 0 = Search and Read or Search and Insert, 1 = Read or Write	Bit: 14	RW
ACCESS_OK	If the access was ok	Bit: 13	RO
REFRESH_CLEAR_OR_DUR	0 = Set the Refresh values to the same as the durations 1 = Clear to 0	Bit: 12	RW
REFRESH_VAL	Update Refresh Rates valid	Bit: 11	RW
DELAY_DURATION_VAL	Delay Duration field valid	Bit: 10	RW
DELAY_INTERVAL_VAL	Delay Interval field valid	Bit: 9	RW
SYNC_DURATION_VAL	Sync Duration field valid	Bit: 8	RW
SYNC_INTERVAL_VAL	Sync Interval field valid	Bit: 7	RW
ANNOUNCE_DURATION_VAL	Anounce Duration field valid	Bit: 6	RW
ANNOUNCE_INTERVAL_VAL	Announce Interval field valid	Bit: 5	RW
PRIORITY_VAL	Priority field valid	Bit: 4	RW
IP_VAL	Ip field valid	Bit: 3	RW
MAC_VAL	Mac and Mac Refresh field valid	Bit: 2	RW
FLAGS_VAL	Flags field valid	Bit: 1	RW
ENTRY_VAL	Entry Valid field valid	Bit: 0	RW

3.2.7.2PTP OC Unicast Dataset 1 Register

Flags of a Unicast Node, Grant bits can be overwritten (but generaly shouldn't). If in Master Mode this is also used to have only listed Slaves or Static Slavesin the table. In Slave mode we need to enter all potential Masters.

Ptp OcUnicastDs1 Reg																															
Reg Description																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIORITY								-	MAC_VALID_VAL	DELAY_GRANT_VAL	SYNC_GRANT_VAL	ANNOUNCE_GRANT_VAL	DELAY_CONF_VAL	SYNC_CONF_VAL	ANNOUNCE_CONF_VAL	-	MAC_VALID	DELAY_GRANT	SYNC_GRANT	ANNOUNCE_GRANT	DELAY_CONF	SYNC_CONF	ANNOUNCE_CONF	-							ENTRY_VALID
RW								RO	RW	RW	RW	RW	RW	RW	RW	RO	RW	RW	RW	RW	RW	RW	RW	RO							RW
Reset: 0x00000000																															
Offset: 0x0604																															

Name	Description	Bits	Access
PRIORITY	Priority of this Entry	Bit: 31:24	RW
-	Reserved, read 0	Bit: 23	RO
MAC_VALID_VAL	MAC Address Valid Flag Valid	Bit: 22	RW
DELAY_GRANT_VAL	Delay Messages Grant Flag Valid	Bit: 21	RW
SYNC_GRANT_VAL	Sync Messages Grant Flag Valid	Bit: 20	RW

ANNOUNCE_GRANT_VAL	Announce Messages Grant Flag Valid	Bit: 19	RW
DELAY_CONF_VAL	Delay Messages Flag Valid	Bit: 18	RW
SYNC_CONF_VAL	Sync Messages Flag Valid	Bit: 17	RW
ANNOUNCE_CONF_VAL	Announce Messages Flag Valid	Bit: 16	RW
-	Reserved, read 0	Bit: 15	RO
MAC_VALID	MAC Address Valid Flag	Bit: 14	RW
DELAY_GRANT	Delay Messages Grant Flag	Bit: 13	RW
SYNC_GRANT	Sync Messages Grant Flag	Bit: 12	RW
ANNOUNCE_GRANT	Announce Messages Grant Flag	Bit: 11	RW
DELAY_CONF	Delay Messages Configured Flag	Bit: 10	RW
SYNC_CONF	Sync Messages Configured Flag	Bit: 9	RW
ANNOUNCE_CONF	Announce Messages Configured Flag	Bit: 8	RW
-	Reserved, read 0	Bit: 7:1	RO
ENTRY_VALID	If the Entry is valid	Bit: 0	RW

3.2.7.3PTP OC Unicast Dataset 2 Register

MAC address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs2 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC(3)								MAC(2)								MAC(1)								MAC(0)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0608																															

Name	Description	Bits	Access
MAC(3)	MAC Byte 3	Bit:31:24	RW
MAC(2)	MAC Byte 2	Bit:23:16	RW
MAC(1)	MAC Byte 1	Bit:15:8	RW
MAC(0)	MAC Byte 0	Bit:7:0	RW

3.2.7.4 PTP OC Unicast Dataset 3 Register

MAC address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs3 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-																MAC(5)								MAC(4)							
RO																RW								RW							
Reset: 0x00000000																															
Offset: 0x060C																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:16	RO
MAC(5)	MAC Byte 5	Bit:15:8	RW
MAC(4)	MAC Byte 4	Bit:7:0	RW

3.2.7.5PTP OC Unicast Dataset 4 Register

Ipv4/Ipv6 address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs4 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(3)								IP(2)								IP(1)								IP(0)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0610																															

Name	Description	Bits	Access
IP(3)	IP Byte 3 (ipv4 and Ipv6)	Bit:31:24	RW
IP(2)	IP Byte 2 (ipv4 and Ipv6)	Bit:23:16	RW
IP(1)	IP Byte 1 (ipv4 and Ipv6)	Bit:15:8	RW
IP(0)	IP Byte 0 (ipv4 and Ipv6)	Bit:7:0	RW

3.2.7.6PTP OC Unicast Dataset 5 Register

IPv6 address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs5 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(7)								IP(6)								IP(5)								IP(4)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0614																															

Name	Description	Bits	Access
IP(7)	IP Byte 7 (Ipv6 only)	Bit:31:24	RW
IP(6)	IP Byte 6 (Ipv6 only)	Bit:23:16	RW
IP(5)	IP Byte 5 (Ipv6 only)	Bit:15:8	RW
IP(4)	IP Byte 4 (Ipv6 only)	Bit:7:0	RW

3.2.7.7PTP OC Unicast Dataset 6 Register

IPv6 address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs6 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(11)								IP(10)								IP(9)								IP(8)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0618																															

Name	Description	Bits	Access
IP(11)	IP Byte 11 (Ipv6 only)	Bit:31:24	RW
IP(10)	IP Byte 10 (Ipv6 only)	Bit:23:16	RW
IP(9)	IP Byte 9 (Ipv6 only)	Bit:15:8	RW
IP(8)	IP Byte 8 (Ipv6 only)	Bit:7:0	RW

3.2.7.8PTP OC Unicast Dataset 7 Register

IPv6 address of the Master in Slave Mode, of the Slave in Master Mode.

Ptp OcUnicastDs7 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP(15)								IP(14)								IP(13)								IP(12)							
RW								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x061C																															

Name	Description	Bits	Access
IP(15)	IP Byte 15 (Ipv6 only)	Bit:31:24	RW
IP(14)	IP Byte 14 (Ipv6 only)	Bit:23:16	RW
IP(13)	IP Byte 13 (Ipv6 only)	Bit:15:8	RW
IP(12)	IP Byte 12 (Ipv6 only)	Bit:7:0	RW

3.2.7.9PTP OC Unicast Dataset 8 Register

Frame intervals of the Slaves in Master mode. In Slave mode this has no effect, intervals come from the Port Dataset and if granted the rate we requested.

Ptp OcUnicastDs8 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-								DELAY_INTERVAL								SYNC_INTERVAL								ANNOUNCE_INTERVAL							
RO								RW								RW								RW							
Reset: 0x00000000																															
Offset: 0x0620																															

Name	Description	Bits	Access
-	Reserved, read 0	Bit:31:24	RO
DELAY_INTERVAL	Signed Log Delay Interval	Bit:23:16	RW
SYNC_INTERVAL	Signed Log Sync Interval	Bit:15:8	RW
ANNOUNCE_INTERVAL	Signed Log Announce Interval	Bit:7:0	RW

3.2.7.10 PTP OC Unicast Dataset 9 Register

Announce Duration to be configured for Signaling messages as Slave (maybe also the same value if REFRESH_CLEAR_OR_DUR=0).
When the Unicast Entry is read it is the remaining until the next Refresh as a Slave and as Master until Grant is dropped.

Ptp OcUnicastDs9 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANNOUNCE_DURATION																														
RW																														
Reset: 0x00000000																														
Offset: 0x0624																														

Name	Description	Bits	Access
ANNOUNCE_DURATION	Lease time for Announce Messages in Seconds	Bit: 31:0	RW

3.2.7.11PTP OC Unicast Dataset 10 Register

Sync Duration to be configured for Signaling messages as Slave (maybe also the same value if REFRESH_CLEAR_OR_DUR=0).
When the Unicast Entry is read it is the remaining until the next Refresh as a Slave and as Master until Grant is dropped.

Ptp OcUnicastDs10 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC_DURATION																													
RW																													
Reset: 0x00000000																													
Offset: 0x0628																													

Name	Description	Bits	Access
SYNC_DURATION	Lease time for Sync Messages in Seconds	Bit: 31:0	RW

3.2.7.12 PTP OC Unicast Dataset 11 Register

Delay Duration to be configured for Signaling messages as Slave (maybe also the same value if REFRESH_CLEAR_OR_DUR=0).
When the Unicast Entry is read it is the remaining until the next Refresh as a Slave and as Master until Grant is dropped.

Ptp OcUnicastDs11 Reg																														
Reg Description																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY_DURATION																														
RW																														
Reset: 0x00000000																														
Offset: 0x062C																														

Name	Description	Bits	Access
DELAY_DURATION	Lease time for Delay Messages in Seconds	Bit: 31:0	RW

3.2.7.13 PTP OC Unicast Dataset 12 Register

ARP (IPv4), ICMP (IPv6) refresh rate in Seconds to obtain MAC address from Remote Node.

Ptp OcUnicastDs12 Reg																													
Reg Description																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_REFRESH																													
RW																													
Reset: 0x00000000																													
Offset: 0x0630																													

Name	Description	Bits	Access
MAC_REFRESH	Until another ARP Request Shall be sent in Seconds, 0xFFFFFFFF means never	Bit: 31:0	RW

4 Design Description

The following chapters describe the internals of the PTP Ordinary Clock: starting with the Top Level, which is a collection of subcores, followed by the description of all subcores.

4.1 Top Level – PTP Ordinary Clock

4.1.1.1 Parameters

The core must be parametrized at synthesis time. There are a couple of parameters which define the final behavior and resource usage of the core.

Name	Type	Size	Description
DefaultProfileSupport_Gen	boolean	1	If the core shall support the PTP Default Profile
PowerProfileSupport_Gen	boolean	1	If the core shall support the PTP Power Profile
UtilityProfileSupport_Gen	boolean	1	If the core shall support the PTP Utility Profile
TsnProfileSupport_Gen	boolean	1	If the core shall support the PTP TSN Profile
ItuG82651ProfileSupport_Gen	boolean	1	Support for ITU G8265.1 Profile
ItuG82751ProfileSupport_Gen	boolean	1	Support for ITU G8275.1 Profile
ItuG82752ProfileSupport_Gen	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicastEntries_Gen	natural	1	Number of Unicast Entries
TwoStepSupport_Gen	boolean	1	If the core shall also support to generate TwoStep Sync and PDelay
AsymmetrySupport_Gen	boolean	1	If the core shall also support asymmetry corrections
DynamicMessageRatesSupport_Gen	boolean	1	Support for dynmic message rates
E2eSupport_Gen	boolean	1	If the core shall support E2E

			delay mechanism (only valid with default profile support)
E2eUnicastSupport_Gen	boolean	1	If the core shall support E2E unicast delay mechanism (only valid with default profile support and E2E support)
P2pSupport_Gen	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
Layer2Support_Gen	boolean	1	If in Default Profile if Layer 2 shall be supported (Power and Utility Profile always use Layer 2)
Layer3v4Support_Gen	boolean	1	If in Default Profile if Layer 3 Ipv4 shall be supported (Power and Utility Profile always use Layer 2)
Layer3v6Support_Gen	boolean	1	If in Default Profile if Layer 3 Ipv6 shall be supported (Power and Utility Profile always use Layer 2)
PassThrough_Gen	boolean	1	If frames after the OC shall be passed further
StaticConfig_Gen	boolean	1	If Static Configuration or AXI is used: true = Static, false = AXI
ExtSync_Gen	boolean	1	If the core shall run in Grandmaster mode where the timing quality parameters are configured from outside
SlaveOnly_Gen	boolean	1	If the clock shall be a Slave-Only clock
MasterOnly_Gen	boolean	1	If the clock shall be a Master-Only clock
MacProcessor Support_Gen	boolean	1	If the MAC discovery for Unicast nodes shall be done
ClockClkPeriod Nanosecond_Gen	natural	1	Clock Period in Nanosecond: Default for 50 MHz = 20 ns

ClockClkPeriod FractNum_Gen	natural	1	Fractional Clock Period Numerator (0 if integer)
ClockClkPeriod FractDeNum_Gen	natural	1	Fractional Clock Period Denumerator (0 if integer)
DriftCorrection_Gen	boolean	1	If the clock shall be syntonized for delay mesaurements, this requires that the clock gives the frequency adjustement values to the OC
CounterClock_Gen	boolean	1	If an internal free running counter clock shall be instantiated which is used for P2P measurements. This shall only be set to false if GM only or E2E measurement is used only.
AdvancedFilter_Gen	boolean	1	Averaging of Delay with an advanced filter (over more values)
AverageWindow Nanosecond_Gen	natural	1	Averaging window for the delay measurement. If a new measurement is out of that range averaging is skipped
LuckyPacketFilter Support_Gen	boolean	1	If the Lucky Packet Filter shall be supported
LuckyPacketFilter Samples_Gen	natural	1	Maximum number of Samples in Lucky Packet Filter Window
RxDelayNanosecond 10_Gen	integer	1	PHY receive delay (10Mbit)
RxDelayNanosecond 100_Gen	integer	1	PHY receive delay (100Mbit)
RxDelayNanosecond 1000_Gen	integer	1	PHY receive delay (1000Mbit)
TxDelayNanosecond 10_Gen	integer	1	PHY transmit delay (10Mbit)
TxDelayNanosecond 100_Gen	integer	1	PHY transmit delay (100Mbit)
TxDelayNanosecond	integer	1	PHY transmit delay

1000_Gen			(1000Mbit)
HighResSupport_Gen	boolean	1	If a high-resolution clock SysClkNx with alignment to SysClk is used
HighResFreq Multiply_Gen	natural	1	Multiplication factor of the high-resolution clock compared to SysClk
MaxOffset_Gen	natural	1	Max Offset for changing from the Uncalibrated to Slave state
IoFf_Gen	boolean	1	If the interface adapter shall contain an IO flip flop
P2pNoDelay_Gen	boolean	1	If no peer delay shall be measured and just set to 0
AxiAddressRang Low_Gen	std_logic_vector	32	AXI Base Address
AxiAddressRange High_Gen	std_logic_vector	32	AXI Base Address plus Registerset Size Default plus 0xFFFF
ManagementSupport_Gen	boolean	1	If the core shall handle PTP Management messages
SignalingSupport_Gen	boolean	1	If the core shall support Signaling
ManufacturerIdentity_Gen	Common_Byte_Type	3	3 byte array for the Manufacturer Identity => NTL in hex 0x4E, 0x54, 0x4C
Product Description_Gen	string	32	String of the Products Description
RevisionData_Gen	string	32	String of the Products Revision
UserDescription_Gen	string	32	String of the Products User Description
TimeInaccuracy Nanosecond_Gen	natural	1	The Inaccuracy that shall be added to the PTP frames when forwarded
Sim_Gen	boolean	1	If in Testbench simulation mode: true = Simulation, false =

			Synthesis

Table 5: Parameters

One of the four parameters DefaultSupport_Gen, PowerProfileSupport_Gen, UtilityProfileSupport_Gen and TsnProfileSupport_Gen has to be true.

4.1.1.2 Structured Types

4.1.1.2.1 Clk_Time_Type

Defined in Clk_Package.vhd of library ClkLib
Type represents the time used everywhere. For this type overloaded operators + and – with different parameters exist.

Field Name	Type	Size	Description
Second	std_logic_vector	32	Seconds of time
Nanosecond	std_logic_vector	32	Nanoseconds of time
Fraction	std_logic_vector	2	Fraction numerator (mostly not used)
Sign	std_logic	1	Positive or negative time, 1 = negative, 0 = positive.
TimeJump	std_logic	1	Marks when the clock makes a time jump (mostly not used)

Table 6: Clk_Time_Type

4.1.1.2.2 Clk_TimeAdjustment_Type

Defined in Clk_Package.vhd of library ClkLib
Type represents the time used everywhere. For this type overloaded operators + and – with different parameters exist.

Field Name	Type	Size	Description
TimeAdjustment	Clk_Time_Type	1	Time to adjust
Interval	std_logic_vector	32	Adjustment interval, for the drift correction this is the denumerator of the rate in nanoseconds (TimeAdjustment every Interval = drift

			rate), for offset correction this is the period in which the time shall be correct-ed(TimeAdjustment in Inter-val), for setting the time this has no mining.
Valid	std_logic	1	Whether the Adjustment is valid or not

Table 7: Clk_TimeAdjustment_Type

4.1.1.2.3 Ptp_OrdinaryClockStaticConfig_Type

Defined in Ptp_OrdinaryClockAddrPackage.vhd of library PtpLib
This is the type used for static configuration.

Field Name	Type	Size	Description
Profile	Ptp_Profile_Type	1	Which Profile the core shall be run in: DefaultProfile_E PowerProfile_E UtilityProfile_E TsnProfile_E
Layer	Ptp_Layer_Type	1	Which layer shall be used in Default Profile: Layer2_E Layer3v4_E Layer3v6_E
DelayMechanism	Ptp_Delay Mechanism_Type	1	Which layer shall be used in Default Profile: P2p_E E2e_E
DelayE2eUnicast	std_logic	1	If E2E measurements shall be done with unicast frames and answered unicast if requested unicast
TwoStep	std_logic	1	If TwoStep shall be used
Signaling	std_logic	1	If Signaling shall be used

Vlan	Ptp_Vlan_Type	1	The Pcp,Dei and Vid of the VLAN
VlanEnable	std_logic	1	If VLAN shall be used
Ip	Com-mon_Byte_Type	16	The source IP to be used if in Layer3 mode, 4 bytes Ipv4, 16 bytes Ipv6, index 0 = MSB
DefaultDataset_ ClockIdentity	Ptp_ClockIdentit y_Type	1	The Clock Identity of the clock, also used for the MAC (with-out bytes 3&4)
DefaultDataset_ DomainNumber	std_logic_vector	8	Which PTP Domain the core shall run on
DefaultDataset_ ClockQuality	Ptp_ClockQuality _Type;		ClockClass, ClockAccuracy and OffsetScaledLogVariance of the clock
DefaultDataset_ Priority1	std_logic_vector	8	Priority 1 of the clock
DefaultDataset_ Priority2	std_logic_vector	8	Priority 2 of the clock
DefaultDataset_ GrandmasterId	std_logic_vector	16	Grandmaster ID for the Power Profile to be announced
DefaultDataset_ Grandmaster TimeInaccuracy	std_logic_vector	32	Grandmaster Inaccuracy to be Announce
TimeProperties Dataset_ CurrentUtcOffset	std_logic_vector	16	UTC offset from TAI to UTC
TimeProperties Dataset_CurrentUtc OffsetValid	std_logic	1	If the UTC offset is valid
TimeProperties Dataset_Leap59	std_logic	1	If a leap second 59 shall be announced
TimeProperties Dataset_Leap61	std_logic	1	If a leap second 61 shall be announced
TimeProperties Dataset_ TimeTraceable	std_logic	1	If the time is traceable to a primary source
TimeProperties Dataset_	std_logic	1	If the frequency is traceable to a primary source

FrequencyTraceable			
TimeProperties Dataset_PtpTimescale	std_logic	1	If the clock runs in PTP time-scale (TAI)
TimePropertiesDa-taset_TimeSource	std_logic_vector	8	What the clock source for the time to distribute is
TimeProperties Dataset_CurrentOffset	std_logic_vector	32	The current Offset of the PTP time in an alternative timescale in Power Profile
TimePropertiesDa-taset_JumpSeconds	std_logic_vector	32	The offset in seconds to jump when announce in power Profile (not used internally, only for distribution)
TimeProperties Dataset_Time OfNextJumpSeconds	std_logic_vector	48	When the next timejump will happen
TimeProperties Dataset_ DisplayNameLength	std_logic_vector	8	Display name length shall be 3
TimeProperties Dataset_DisplayName	Com-mon_Byte_Type	12	Display name shall be “PTP” in hex

Table 8: Ptp_OrdinaryClockStaticConfig_Type

4.1.1.2.4 Ptp_OrdinaryClockStaticConfigVal_Type

Defined in Ptp_OrdinaryClockAddrPackage.vhd of library PtpLib
This is the type used for valid flags of the static configuration.

Field Name	Type	Size	Description
Enable_Val	std_logic	1	Enables the PTP Ordinary Clock
Profile_Val	std_logic	1	If the Profile shall be set
Vlan_Val	std_logic	1	If the VLAN shall be set
Ip_Val	std_logic	1	If the IP shall be set
DefaultDataset_ ClockIdentity_Val	std_logic	1	If the ClockIdentity shall be set
DefaultDataset_ DomainNumber_Val	std_logic	1	If the DomainNumber shall be set

DefaultDataset_ ClockQuality_Val	std_logic	1	If the ClockQuality shall be set
DefaultDataset_ Priority1_Val	std_logic	1	If the Priority1 shall be set
DefaultDataset_ Priority2_Val	std_logic	1	If the Priority2 shall be set
DefaultDataset_ GrandmasterId_Val	std_logic	1	If the GrandmasterId shall be set
DefaultDataset_ Grandmaster TimeInaccuracy_Val	std_logic	1	If the GrandmasterTimeInaccu-racy shall be set
TimeProperties Dataset_ CurrentUtcOffset_Val	std_logic	1	If the CurrentUtcOffset shall be set
TimeProperties Dataset_ CurrentUtcOffset Valid_Val	std_logic	1	If the CurrentUtcOffsetValid shall be set
TimeProperties Dataset_Leap59_Val	std_logic	1	If the Leap59 shall be set
TimeProperties Dataset_Leap61_Val	std_logic	1	If the Leap61 shall be set
TimeProperties Dataset_ TimeTraceable_Val	std_logic	1	If the TimeTraceable shall be set
TimePropertiesDa-ta-set_FrequencyTraceable_Val	std_logic	1	If the FrequencyTraceable shall be set
TimeProperties Dataset_ PtpTimescale_Val	std_logic	1	If the PtpTimescale shall be set
TimeProperties Dataset_ TimeSource_Val	std_logic	1	If the TimeSource shall be set
TimeProperties Dataset_ CurrentOffset_Val	std_logic	1	If the CurrentOffset shall be set

TimeProperties Dataset_ JumpSeconds_Val	std_logic	1	If the JumpSeconds shall be set
TimeProperties Dataset_Time OfNextJumpSeconds _Val	std_logic	1	If the TimeOfNextJumpSec-onds shall be set
TimeProperties Dataset_DisplayName Length_Val	std_logic	1	If the DisplayNameLength shall be set
TimeProperties Dataset_ DisplayName_Val	std_logic	1	If the DisplayName shall be set

Table 9: Ptp_OrdinaryClockStaticConfigVal_Type

4.1.1.2.5 Ptp_OrdinaryClockStaticStatus_Type

Defined in Ptp_OrdinaryClockAddrPackage.vhd of library PtpLib
This is the type used for static status supervision.

Field Name	Type	Size	Description
CoreInfo	Clk_CoreInfo_ Type	1	Info about the Cores state
DefaultDataset_Profile	Ptp_Profile_Type	1	Which Profile the Core runs in
DefaultDataset_Layer	Ptp_Layer_type	1	Which Transport Layer it uses
DefaultDataset_Vlan	Ptp_Vlan_Type	1	VLAN Tag that the Core uses
DefaultDataset_ VlanEnable	std_logic	1	If the Core uses VLAN Tags
DefaultDataset_Ip	Common_ Byte_Type	4	Which Source IP the Core uses
DefaultDataset_ Signaling	std_logic	1	If the Core uses Signaling
DefaultDataset_ TwoStepFlag	std_logic	1	If the Core runs in Twostep mode
DefaultDataset_ ClockIdentity	Ptp_Clock Identity_Type	1	Clockidentity of the Core
DefaultDataset_	std_logic_vector	16	Number of Ports (1 for OC)

NumberPorts			
DefaultDataset_ ClockQuality	Ptp_ClockQuality _Type	1	Clock Quality of the Core
DefaultDataset_ Priority1	std_logic_vector	8	Priority 1 of the Core
DefaultDataset_ Priority2	std_logic_vector	8	Priority 2 of the Core
DefaultDataset_ DomainNumber	std_logic_vector	8	Domain Number the Core runs on
DefaultDataset_ SlaveOnly	std_logic	1	If it is running in slave only mode
DefaultDataset_ GrandmasterId	std_logic_vector	16	Power Profile Grandmaster Id
DefaultDataset_ Grandmaster TimeInaccuracy	std_logic_vector	32	Power Profile Grandmaster Inaccuracy
TimeProperties Dataset_ CurrentUtcOffset	std_logic_vector	16	UTC offset from TAI to UTC
TimeProperties Dataset_CurrentUtc OffsetValid	std_logic	1	If the UTC offset is valid
TimeProperties Dataset_Leap59	std_logic	1	If a leap second 59 shall be announced
TimeProperties Dataset_Leap61	std_logic	1	If a leap second 61 shall be announced
TimeProperties Dataset_ TimeTraceable	std_logic	1	If the time is traceable to a primary source
TimeProperties Dataset_ FrequencyTraceable	std_logic	1	If the frequency is traceable to a primary source
TimeProperties Dataset_PtpTimescale	std_logic	1	If the clock runs in PTP time-scale (TAI)
TimePropertiesDa-taset_TimeSource	std_logic_vector	8	What the clock source for the time to distribute is
TimeProperties Dataset_CurrentOffset	std_logic_vector	32	The current Offset of the PTP time in an alternative timescale

			in Power Profile
TimePropertiesDa-taset_JumpSeconds	std_logic_vector	32	The offset in seconds to jump when announce in power Profile (not used internally, only for distribution)
TimeProperties Dataset_Time OfNextJumpSeconds	std_logic_vector	48	When the next timejump will happen
TimeProperties Dataset_ DisplayNameLength	std_logic_vector	8	Display name length shall be 3
TimeProperties Dataset_DisplayName	Common_ Byte_Type	12	Display name shall be “PTP” in hex
CurrentDataset_ StepsRemoved	std_logic_vector	16	Number of Steps between Grandmaster and Slave
CurrentDataset_ OffsetFromMaster	std_logic_vector	64	Offset from Master
CurrentDataset_ MeanPathDelay	std_logic_vector	64	E2E Delay
ParentDataset_ ParentPortIdentity	Ptp_Port Identity_Type	1	Parent Port Indentity
ParentDataset_ ParentStats	std_logic	1	Parent Statistics (always 0)
ParentDataset_ ObsParentOffset ScaledLogVaria	std_logic_vector	16	Observed Parent Offser Vari-ance
ParentDataset_ ObsParentClock PhaseChangeRate	std_logic_vector	32	Parent Phase Change Rate
ParentDataset_ GrandmasterIdentity	Ptp_Clock Identity_Type	1	Granmaster Clock Identity
ParentDataset_ Grandmaster ClockQuality	Ptp_Clock Quality_Type	1	Grandmaster Clock Quality
ParentDataset_ GrandmasterPriority1	std_logic_vector	8	Grandmaster Priority 1
ParentDataset_ GrandmasterPriority2	std_logic_vector	8	Grandmaster Priority 2

ParentDataset_ GrandmasterId	std_logic_vector	16	Grandmaster Short ID
ParentDataset_ Grandmaster TimeInaccuracy	std_logic_vector	32	Granmaster Inaccuracy
ParentDataset_ Network TimeInaccuracy	std_logic_vector	32	Network Time Inacucuracy to Grandmaster
PortDataset_ PortIdentity	Ptp_Port Identity_Type	1	Port Idenity
PortDataset_ PortState	std_logic_vector	8	Port State
PortDataset_ LogMinDelayReq Interval	std_logic_vector	8	Delay Request Message Interval
PortDataset_ PeerMeanPathDelay Valid	std_logic	1	P2P Delay valid
PortDataset_ PeerMeanPathDelay	std_logic_vector	64	P2P Delay
PortDataset_ LogAnnounceInterval	std_logic_vector	8	Announce Message Interval
PortDataset_ AnnounceReceipt Timeout	std_logic_vector	8	Announce Receipt Timeout
PortDataset_ LogSyncInterval	std_logic_vector	8	Sync Message Interval
PortDataset_ LogPtpCapable SignalingInterval	std_logic_vector	8	802.1 Signaling Message Interval
PortDataset_ PtpCapableTimeout	std_logic_vector	8	Timeout for 802.1 PTP Capable signaling messages
PortDataset_ DelayMechanism	std_logic_vector	8	Delay Mechanism
PortDataset_ DelayE2eUnicast	std_logic	1	E2E Unicast handling
PortDataset_ LogMinPdelayReq	std_logic_vector	8	Peer Delay Request Message Interval

Interval			
PortDataset_ DelayTimeout	std_logic_vector	8	Timeout for Peer Dealy Answers
PortDataset_ MaxPdelay	std_logic_vector	32	Maximum allowed Peer Delay in TSN mode
PortDataset_ VersionNumber	std_logic_vector	4	PTP Version
PortDataset_ Asymmetry	std_logic_vector	32	Port Asymmetry

Table 10: Ptp_OrdinaryClockStaticStatus_Type

4.1.1.2.6 Ptp_OrdinaryClockStaticStatusVal_Type

Defined in Ptp_OrdinaryClockAddrPackage.vhd of library PtpLib
This is the type used for valid flags of the static status supervision.

Field Name	Type	Size	Description
CoreInfo_Val	std_logic	1	Core Info valid

Table 11: Ptp_OrdinaryClockStaticStatusVal_Type

4.1.1.3 Entity Block Diagram

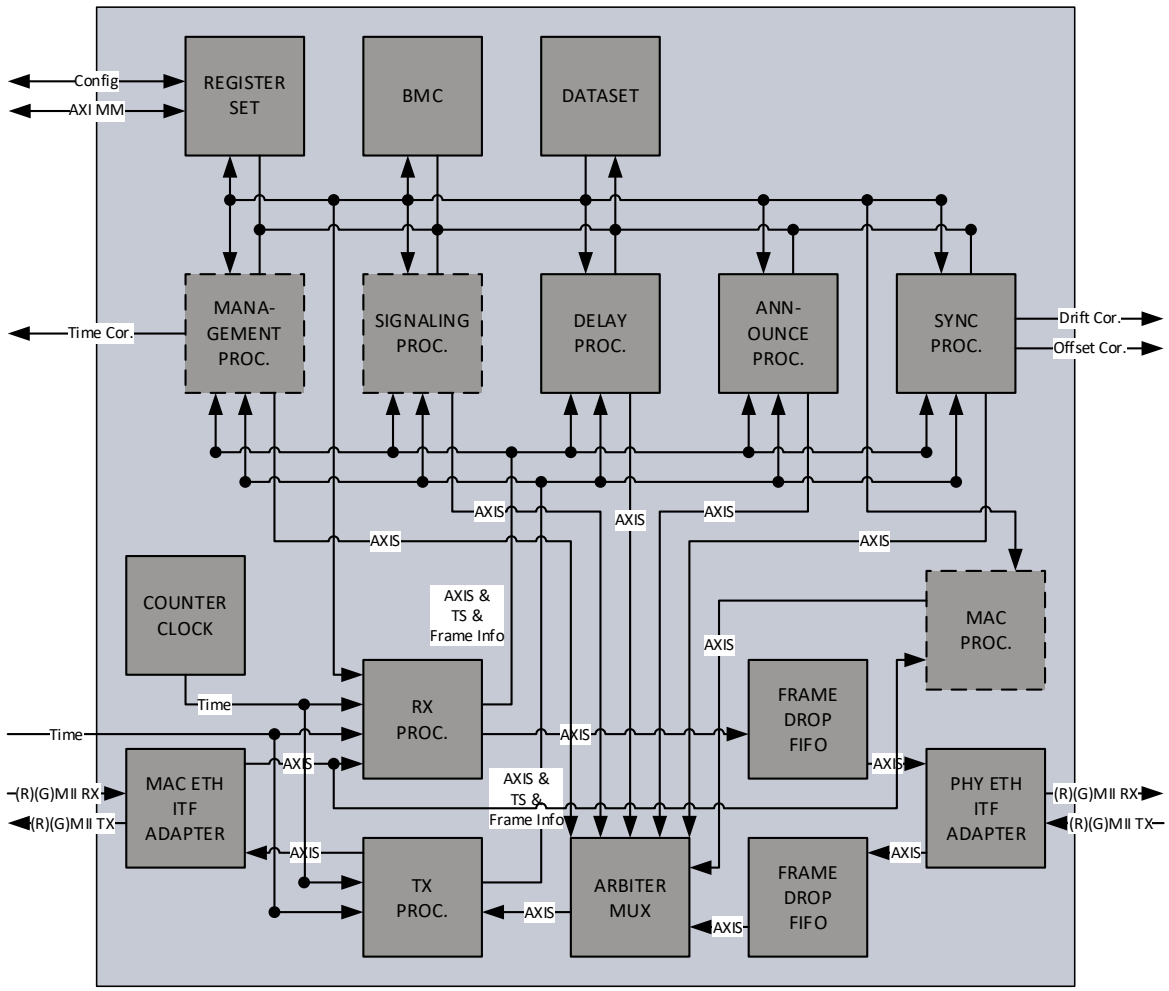

Figure 13: PTP Ordinary Clock

4.1.1.4 Entity Description

Frame Arbiter & Mux

This module multiplexes multiple AXI stream inputs into one AXI stream output. Multiplexing is done on stream base, which means that once a stream has been started it will be the only stream forwarded until a TLAST is asserted, and then arbitration starts again. It handles two different priority levels. Each priority level is arbitrated in a round robin manner. Low priority streams can starve; high priority streams never starve.

Frame Drop Fifo

This module is AXI stream frame FIFO. It is always ready to receive data, so no throttling of the stream is necessary. This on the other hand allows overflow condi-

tions. The FIFO is large enough that it can contain at least two maximum sized frames. This is important, since when it runs into an overflow condition the current input frame shall be dropped. It can be configured either as cut-through (frame is forwarded immediately) or store-and-forward (whole frame is stored first before forwarding) FIFO. When run in store-and-forward mode it handles additionally a user drop-flag which allows explicitly dropping the current frame. In this design it is only used as store-and-forward FIFO, but with the drop-flag. The FIFO on the RX path has a couple of functionalities, it allows to drop PTP and corrupted frames (they shall not be forwarded), it allows a deterministic behavior of the RX Processor and it compensates the speed differences between incoming and outgoing data streams. The FIFO on the TX path has also similar functionalities, it buffers the incoming data stream for the case when an internal frame is processed (send DelayReq, DelayResp, Sync, Announce etc..), it allows a deterministic behavior (no starving) of the TX Processor and it compensates the speed differences between incoming and outgoing data streams.

Rx Processor

This module handles all incoming frames. It does the RX timestamping, frame parsing, on-the-fly CorrectionField processing (adding PortDelay and/or subtracting RX timestamps) and is the source of the drop-flag going to the RX Frame Drop Fifo for PTP frame filtering. In addition it realigns the 32bit stream so the PTP frames are always aligned the same way in case an IP header or HSR tag is inserted.
See 4.2.1 for more details.

Tx Processor

This module handles all outgoing frames. It does the TX timestamping; frame parsing, on-the-fly CorrectionField and TimestampField processing (adding or inserting TX timestamps). In addition it realigns the 32bit stream so the PTP frames are always aligned the same way in case an IP header or HSR tag is inserted.
See 4.2.2 for more details.

MAC Processor

This block handles the MAC Address resolution via ARP or ICMP.

Announce Processor

This module handles all PTP Announce frames. In Master mode it periodically generates Announce frames based on the parameters of the Datasets. In Slave

mode it extracts all relevant information from the Announce frame and updates the Foreign Master Dataset for Master qualification.
See 4.2.3 for more details.

Delay Processor

This module handles all PTP Delay (P2P only, no E2E support) frames. It runs independent of the current state of the PTP Ordinary Clock. It has a Client and Server functionality. As Client it periodically generates PeerDelayRequest frames based on the parameters of the Datasets and waits for the other peer to respond with a PeerDelayResponse. Based on the information and timestamps of the frames the Client calculates the PeerDelay averages it and stores it in a Dataset.
As Server it waits for incoming PeerDelayRequests and answers them with corresponding PeerDelayResponses.
See 4.2.4 for more details.

Sync Processor

This module handles all PTP Sync frames. In Master mode it periodically generates Sync frames based on the parameters of the Datasets. In Slave mode it extracts all relevant information from the Sync frame and based on this information, timestamps and the delay to the peer it calculates the offset and drift of the local clock against the Master as reference and corrects it.
See 4.2.5 for more details.

Management Processor

This module handles all PTP Management frames. It is optional. It runs independent of the current state of the PTP Ordinary Clock. It acts as a Server, so it waits for incoming Management frames and answers them with corresponding Management frames. This allows remote configuration and supervision based on PTP.
See 4.2.6 for more details.

Signaling Processor

This module handles all PTP Signaling frames. It is optional. If Master it acts as a Server, so it waits for incoming Signaling frames. If Slave it sends period Signaling frames
See 4.2.7 for more details.

Best Master Clock Algorithm

This module implements the Best Master Clock Algorithm and Statemachine according to IEEE1588-2019 clause 9.2 and 9.3. It runs periodically with information from the Datasets and incoming frames and updates the Datasets accordingly. All PTP Events used by the Statemachine are also generated in this module.
See 4.2.7.2 for more details.

Datasets

This module contains the storage of all PTP Datasets: Default-, Port-, Parent-, Current-, TimeProperties- and ForeignMaster-Dataset. All values of the Datasets are writable.
See 4.2.10 for more details.

Clock Counter

This is a free-running counter with nanosecond resolution in a 32 bit second and 32 bit nanosecond format. It is used by the Delay and Residence time measurements.
See 4.2.11 for more details.

MAC & PHY Ethernet Interface Adapter

This module converts the Media Independent Interface (MII) to AXI stream and vice versa. In parallel it has a SFD detector for each path which generates an event when a SFD is detected; this is used for timestamping in the RX/TX Processors. It is also in charge of generating correct Interframe Gaps and a Preamble with SFD. See 4.2.12 for more details.

Registerset

This module is an AXI4Lite Memory Mapped Slave. It provides access to all Datasets and allows to configure the PTP Ordinary Clock. It can be configured to either run in AXI or StaticConfig mode. If in StaticConfig mode, the configuration of the Datasets is done via signals and can be easily done from within the FPGA without CPU. If in AXI mode, a AXI Master has to configure the Datasets with AXI writes to the registers, which is typically done by a CPU
See 4.2.13 for more details.

4.1.1.5 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				

DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
E2eUnicastSupport_Gen	-	boolean	1	E2E Unicast handling support
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
Asymmetry Support_Gen	-	boolean	1	Support for Asymmetry corrections

DynamicMessage RatesSupport_Gen	-	boolean	1	Support for dynmic message rates
PassThrough_Gen	-	boolean	1	If frames after the OC shall be passed further
StaticConfig_Gen	-	boolean	1	If Static Configuration or AXI is used
ExtSync_Gen	-	boolean	1	If the PTP Clock is synchronized externally when in Master mode
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
MacProcessorSupport_Gen		boolean	1	Support for MAC discovery in Unicast mode
ClockClkPeriodNanosecond_Gen	-	natural	1	Integer Clock Period
ClockClkPeriodFractNum_Gen	-	natural	1	Fractional Clock Period Numerator
ClockClkPeriodFractDeNum_Gen	-	natural	1	Fractional Clock Period Denumerator
DriftCorrection_Gen	-	boolean	1	If the clock shall be syntonized for delay mesaurements, this requires that the clock gives the frequency adjustement values to the OC
CounterClock_Gen	-	boolean	1	If an internal free running counter clock shall be instantiated which is used for P2P measurements. This shall only be set to false if GM only or E2E

				measurement is used only.
AdvancedFilter_Gen	-	boolean	1	Averaging of Delay with an advanced filter (over more values)
AverageWindow Nanosecond_Gen	-	natural	1	Delay Averaging window, out of this window, delay is set hard
LuckyPacketFilter Support_Gen	-	boolean	1	If the Lucky Packet Filter shall be supported
LuckyPacketFilter Samples_Gen	-	natural	1	Maximum number of Samples in Lucky Packet Filter Window
RxDelayNanosecond 10_Gen	-	integer	1	RX Delay of the PHY in Nanosecond
RxDelayNanosecond 100_Gen	-	integer	1	RX Delay of the PHY in Nanosecond
RxDelayNanosecond 1000_Gen	-	integer	1	RX Delay of the PHY in Nanosecond
TxDelayNanosecond 10_Gen	-	integer	1	TX Delay of the PHY in Nanosecond
TxDelayNanosecond 100_Gen	-	integer	1	TX Delay of the PHY in Nanosecond
TxDelayNanosecond 1000_Gen	-	integer	1	TX Delay of the PHY in Nanosecond
HighResSupport_Gen	-	boolean	1	If a high-resolution clock SysClkNx with alignment to SysClk is used
HighResFreq Multiply_Gen	-	natural	1	Multiplication factor of the high-resolution clock compared to SysClk
MaxOffset_Gen	-	natural	1	If Offset is larger

				than this change into Uncalibrated state if Slave
P2pNoDelay_Gen	-	boolean	1	If no peer delay shall be measured and just set to 0
IoFf_Gen	-	boolean	1	Shall IO flip flops be instantiated
AxiAddressRangeLow_Gen	-	std_logic_vector	32	AXI Base Address
AxiAddressRangeHigh_Gen	-	std_logic_vector	32	AXI Base Address plus Registerset Size
ManagementSup-port_Gen	-	boolean	1	Should Management Messages be han-dled
Signaling Support_Gen	-	boolean	1	Support for Signal-ing frames
ManufacturerIdenti-ty_Gen	-	Common_Byte_Type	3	Manufacturing Identity string
ProductDescrip-tion_Gen	-	string	32	Product Description string
RevisionData_Gen	-	string	32	Product Revision string
UserDescription_Gen	-	string	32	User Description string
TimeInaccuracyNanosecond_Gen	-	natural	1	What shall be an-nounced when master and in Power Profile
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysClkNx_ClkIn	in	std_logic	1	High-resolution clock (multiple of Sys Clock)

SysRstN_RstIn	in	std_logic	1	System Reset
Config				
StaticConfig_DatIn	in	Ptp_OrdinaryClock StaticConfig_Type	1	Static Configuration
StaticConfig_ValIn	in	Ptp_OrdinaryClock StaticConfigVal _Type	1	Static Configuration valid
Status				
StaticStatus_DatOut	out	Ptp_OrdinaryClock StaticStatus_Type	1	Static Status
StaticStatus_ValOut	out	Ptp_OrdinaryClock StaticStatusVal _Type	1	Static Status valid
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Drift Input				
DriftCount Adjustment_DatIn	in	Clk_DriftCountAdjust ment_Type	1	Adjustment of the frequency so the clock is syntonized as well for delay measurements
®(G)Mii RX Clk/Rst Input				
®(G)MiiRxClk_ClkIn	in	std_logic	1	RX Clock
®(G)MiiRxRstN_RstIn	in	std_logic	1	Reset aligned with RX Clock
®(G)Mii TX Clk/Rst Input				
®(G)MiiTxClk_ClkIn	in	std_logic	1	TX Clock
®(G)MiiTxRstN_RstIn	in	std_logic	1	Reset aligned with TX Clock
®(G)Mii RX Data Input/Output				
®(G)MiiRxDv_Ena	in/ out	std_logic	1	RX Data valid
®(G)MiiRxErr_Ena	in/	std_logic	1	RX Error

	out			
®(G)MiiRxData_Dat	in/ out	std_logic_vector	2-8	RX Data MII:4, RMII:2, GMII:8, RGMII:4
®(G)MiiCol_Dat	in/ out	std_logic	1	Collision
®(G)MiiCrs_Dat	in/ out	std_logic	1	Carrier Sense
®(G)Mii TX Data Input/Output				
®(G)MiiTxEn_Ena	in/ out	std_logic	1	TX Data valid
®(G)MiiTxErr_Ena	in/ out	std_logic	1	TX Error
®(G)MiiTxData_Dat	in/ out	std_logic_vector	2-8	TX Data MII:4, RMII:2, GMII:8, RGMII:4
External Sync Dataset Input				
TimeProperties ExtSync_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties input from external 134oolean134ization for this clock
AXI4 Lite Slave				
AxiWriteAddrValid _ValIn	in	std_logic	1	Write Address Valid
AxiWriteAddrReady _RdyOut	out	std_logic	1	Write Address Ready
AxiWriteAddrAddress _AdrIn	in	std_logic_vector	32	Write Address
AxiWriteAddrProt _DatIn	in	std_logic_vector	3	Write Address Protocol
AxiWriteDataValid _ValIn	in	std_logic	1	Write Data Valid
AxiWriteDataReady _RdyOut	out	std_logic	1	Write Data Ready
AxiWriteDataData _DatIn	in	std_logic_vector	32	Write Data
AxiWriteDataStrobe _DatIn	in	std_logic_vector	4	Write Data Strobe
AxiWriteRespValid _ValOut	out	std_logic	1	Write Response Valid
AxiWriteRespReady _RdyIn	in	std_logic	1	Write Response Ready

AxiWriteResp Response_DatOut	out	std_logic_vector	2	Write Response
AxiReadAddrValid _ValIn	in	std_logic	1	Read Address Valid
AxiReadAddrReady _RdyOut	out	std_logic	1	Read Address Ready
AxiReadAddrAddress _AdrIn	in	std_logic_vector	32	Read Address
AxiReadAddrProt _DatIn	in	std_logic_vector	3	Read Address Protocol
AxiReadDataValid _ValOut	out	std_logic	1	Read Data Valid
AxiReadDataReady _RdyIn	in	std_logic	1	Read Data Ready
AxiReadData Response_DatOut	out	std_logic_vector	2	Read Data
AxiReadDataData _DatOut	out	std_logic_vector	32	Read Data Re-sponse
Time Adjustment Output				
TimeAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Time to set hard
TimeAdjustment _ValOut	out	std_logic	1	Time valid
Offset Adjustment Output				
OffsetAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Calculated new Offset between Master and Slave
OffsetAdjustment _ValOut	out	std_logic;	1	Calculated new Offset valid
Drift Adjustment Output				
DriftAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Calculated new Drift between Master and Slave
DriftAdjustment _ValOut	out	std_logic;	1	Calculated new Drift valid
Offset Adjustment Input				
OffsetAdjustment _DatIn	in	Clk_TimeAdjustment _Type	1	Calculated new Offset after the PI Servo loop
OffsetAdjustment _ValIn	in	std_logic;	1	Calculated new Offset after the PI Servo loop valid
Drift Adjustment Input				
DriftAdjustment	in	Clk_TimeAdjustment	1	Calculated new Drift

_DatIn		_Type		after the PI Servo loop
DriftAdjustment _ValIn	in	std_logic	1	Calculated new Drift after the PI Servo loop valid

Table 12: PTP Ordinary Clock

4.2 Design Parts

The PTP Ordinary Clock core consists of a couple of subcores. Each of the subcores itself consist again of smaller function block. The following chapters describe these subcores and their functionality.

4.2.1 RX Processor

4.2.1.1 Entity Block Diagram

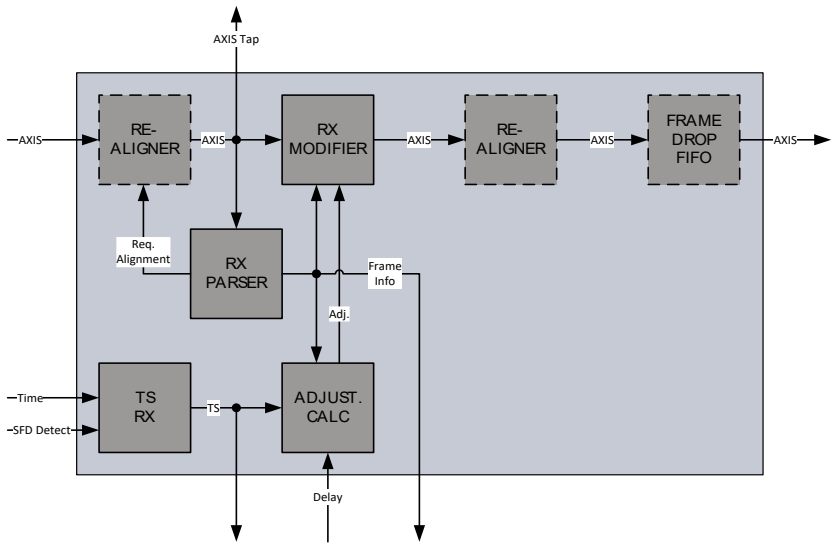

Figure 14: RX Processor

4.2.1.2 Entity Description

RX Timestamper

This module takes a snapshot of the free-running Counter Clock and of the PTP Adjusted Clock when the SFD detected event was asserted by the Interface Adapter. The timestamp is buffered and aligned with the incoming frame, so the Adjustment Calculator can calculate the correct adjustments.

RX Parser

This module parses all incoming Ethernet frames. It extracts the frame length, in the case of PTP frames also the CorrectionField and NetworkTimeInaccuracy and it checks the CRC. If Layer 3 mapping or the Utility or TSN Profile is used it request a specific alignment from the Realigner so the PTP frame is starting 32 bit aligned again.

RX Modifier

This module is the heart of the RX processor. It also parses the frame, waits for the CorrectionField in the frame. It then replaces the CorrectionField with the one from the Adjustment Calculation. In the case of Utility Profile it also modifies the Source Port Identity of PTP frames according to the Stateless Redbox defined in IEC62439. Since the frame is modified on-the-fly the CRC and UDP checksum (Ipv6) is recalculated and overwritten if the incoming CRC was correct otherwise the frames are dropped. PTP frames shall not be forwarded to the MAC, therefore it drops the frames. This is the same Modifier as in the Transparent Clock but with a different configuration. If no PeerDelay was calculated yet, all PTP frames are dropped.

Adjustment Calculation

This module calculates the new CorrectionField out of the incoming Correction-Field, the receive timestamp (subtracted) and the PeerDelay (added). For the OrdinaryClock this is not used, but as with the Modifier this is the same module for the Transparent Clock.

Re-Aligner

This module allows aligning the AXI Data stream as required. It is only instantiated if Layer 3 or Utility or TSN mapping is used. The input can be 32/24/16/8 bit aligned and the output can also be 32/24/16/8 bit aligned. From an external input, the output alignment can be requested. This is used for alignment of the PTP start to a 32 bit boundary (Some mappings cause that the header is not 32 bit aligned anymore) again, and for realignment to a constant 32bit wide stream again.

Frame Drop Fifo

This is an optional Frame Drop Fifo to overcome additional data speed differences. It is not used in the Ordinary Clock.

4.2.1.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile	-	boolean	1	Support for Power

Support_Gen				Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
UnicastSupport_Gen	-	boolean	1	Support for Unicast Messages
Asymmetry Support_Gen	-	boolean	1	Support for Asym-metry corrections
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (manda-tory for Power and Utility profile)
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
HighResSupport_Gen	-	boolean	1	If a high-resolution clock SysClkNx with alignment to SysClk

				is used
HighResFreq Multiply_Gen	-	natural	1	Multiplication factor of the high-resolution clock compared to SysClk
ClockClkPeriod Nanosecond_Gen	-	natural	1	Clock Period in Nanosecond
RX Processor				
DelayNanosecond10 _Gen	-	integer	1	Input Delay of the PHY in Nanosecond
DelayNanosecond100 _Gen	-	integer	1	Input Delay of the PHY in Nanosecond
DelayNano second1000_Gen	-	integer	1	Input Delay of the PHY in Nanosecond
PortIdModify_Gen	-	boolean	1	NA
PortId_Gen	-	std_logic_vector	4	NA
Buffer_Gen	-	boolean	1	If a frame drop fifo shall be instantiated
TimeInaccuracy Nanosecond_Gen	-	natural	1	NA
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysClkNx_ClkIn	in	std_logic	1	High-resolution clock (multiple of Sys Clock)
SysRstN_RstIn	in	std_logic	1	System Reset
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Counter Input				
CounterTime_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Time
CounterTime_ValIn	in	std_logic	1	Freerun Counter Clock Time valid
Link Speed Input				
LinkSpeed_DatIn	in	Common_	1	Link Speed of the

		LinkSpeed_Type		interface
Frame Input				
SfdDetected_EvtIn	in	std_logic	1	Start of Frame Delimiter detected
Timestamp Output				
ClockTimestamp_DatOut	out	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestamp_ValOut	out	std_logic	1	Adjusted PTP Clock Timestamp valid
Timestamp Output				
CounterTimestamp_DatOut	out	Clk_Time_Type	1	Freerun Counter Clock Timestamp
CounterTimestamp_ValOut	out	std_logic	1	Freerun Counter Clock Timestamp valid
Dataset Input				
PortDataset_DatIn	in	Ptp_PortDataset_Type	1	PTP Port Dataset input for this port, contains Delay
DefaultDataset_DatIn	in	Ptp_DefaultDataset_Type	1	PTP Default Dataset input for this clock
Drop Input				
Drop_ValIn	in	std_logic	1	If input frame shall be dropped
Axi Input				
AxisValid_ValIn	in	std_logic	1	AXI Stream frame input
AxisReady_ValOut	out	std_logic	1	
AxisData_DatIn	in	std_logic_vector	32	
AxisStrobe_ValIn	in	std_logic_vector	4	
AxisKeep_ValIn	in	std_logic_vector	4	
AxisLast_ValIn	in	std_logic	1	
AxisUser_DatIn	in	std_logic_vector	3	
Axi Tap Output				
AxisTapValid_ValOut	out	std_logic	1	AXI Stream frame tap output
AxisTapReady_ValOut	out	std_logic	1	
AxisTapData_DatOut	out	std_logic_vector	32	
AxisTapStrobe_ValOut	out	std_logic_vector	4	
AxisTapKeep_ValOut	out	std_logic_vector	4	
AxisTapLast_ValOut	out	std_logic	1	

AxisTapUser_DatOut	out	std_logic_vector	3	
Drop Output				
Drop_ValOut	out	std_logic	1	If output frame shall be dropped
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
Frame Info Output				
FrameInfo_DatOut	out	Ptp_FrameInfo _Type	1	Frame Information
FrameInfo_ValOut	out	Ptp_FrameInfoVal_T ype	1	Frame Information valid

Table 13: RX Processor

4.2.2 TX Processor

4.2.2.1 Entity Block Diagram

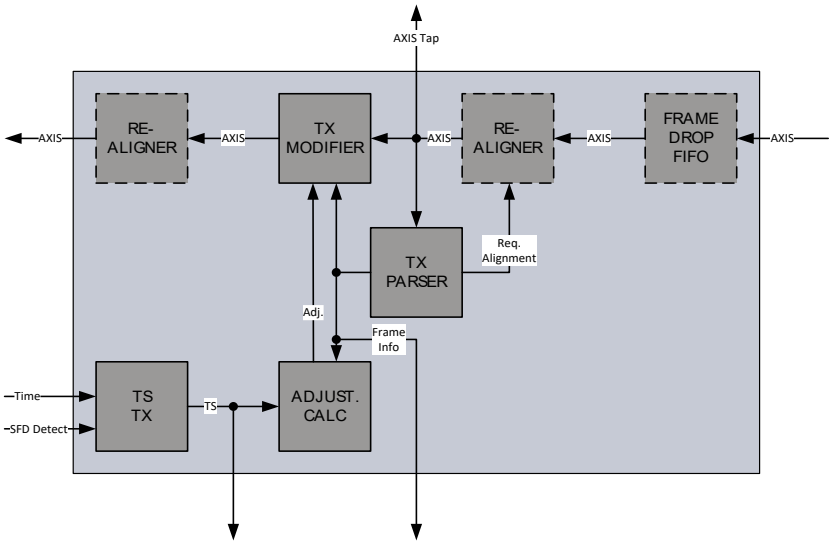

Figure 15: TX Processor

4.2.2.2 Entity Description

TX Timestamper

This module takes a snapshot of the free-running Counter Clock and of the PTP Adjusted Clock when the SFD detected event was asserted by the Interface Adapter. The timestamp is buffered and aligned with the incoming frame, so the Adjustment Calculator can calculate the correct adjustments.

TX Parser

This module parses all outgoing Ethernet frames. It extracts the frame length, in the case of PTP frames also the CorrectionField and NetworkTimeInaccuracy and it checks the CRC. If Layer 3 mapping or the Utility or TSN Profile is used it request a specific alignment from the Realigner so the PTP frame is starting 32 bit aligned again.

TX Modifier

This module is the heart of the TX processor. It also parses the frame, waits for the CorrectionField in the frame. It then replaces the CorrectionField with the one from the Adjustment Calculation. For local generated Sync messages it does the same thing with the TimestampField. In the case of Utility Profile it also modifies (zeros

out) the Source Port Identity of PTP frames according to the Stateless Redbox defined in IEC62439. Since the frame is modified on-the-fly the CRC and UDP checksum (IPv6) is recalculated and overwritten if the incoming CRC was correct or the frame was locally generated, otherwise the frames are marked with a wrong CRC. This is the same Modifier as in the Transparent Clock but with a different configuration.

Adjustment Calculation

This module calculates the new CorrectionField out of the outgoing Correction-Field and the transmit timestamp (added). It also provides the TimestampField for local generated Sync frames.

Re-Aligner

This module allows aligning the AXI Data stream as required. It is only instantiated if Layer 3 or Utility or TSN mapping is used. The input can be 32/24/16/8 bit aligned and the output can also be 32/24/16/8 bit aligned. From an external input, the output alignment can be requested. This is used for alignment of the PTP start to a 32 bit boundary (Some mappings cause that the header is not 32 bit aligned anymore) again, and for realignment to a constant 32bit wide stream again.

Frame Drop Fifo

This is an optional Frame Drop Fifo to overcome additional data speed differences. It is not used in the Ordinary Clock.

4.2.2.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile

ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
UnicastSupport_Gen	-	boolean	1	Support for Unicast Messages
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
Asymmetry Support_Gen	-	boolean	1	Support for Asymmetry corrections
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
HighResSupport_Gen	-	boolean	1	If a high-resolution clock SysClkNx with alignment to SysClk is used
HighResFreq Multiply_Gen	-	natural	1	Multiplication factor of the high-resolution clock compared to SysClk

ClockClkPeriod Nanosecond_Gen	-	natural	1	Clock Period in Nanosecond
TX Processor				
DelayNanosecond10 _Gen	-	integer	1	Input Delay of the PHY in Nanosecond
DelayNanosecond100 _Gen	-	integer	1	Input Delay of the PHY in Nanosecond
DelayNano second1000_Gen	-	integer	1	Input Delay of the PHY in Nanosecond
PortIdModify_Gen	-	boolean	1	NA
PortId_Gen	-	std_logic_vector	4	NA
Buffer_Gen	-	boolean	1	If a frame drop fifo shall be instantiated
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysClkNx_ClkIn	in	std_logic	1	High-resolution clock (multiple of Sys Clock)
SysRstN_RstIn	in	std_logic	1	System Reset
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Counter Input				
CounterTime_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Time
CounterTime_ValIn	in	std_logic	1	Freerun Counter Clock Time valid
Link Speed Input				
LinkSpeed_DatIn	in	Common_ LinkSpeed_Type	1	Link Speed of the interface
Frame Input				
SfdDetected_EvtIn	in	std_logic	1	Start of Frame Delimiter detected
Timestamp Output				
ClockTimestamp _DatOut	out	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestamp	out	std_logic	1	Adjusted PTP Clock

_ValOut				Timestamp valid
Timestamp Output				
CounterTimestamp _DatOut	out	Clk_Time_Type	1	Freerun Counter Clock Timestamp
CounterTimestamp _ValOut	out	std_logic	1	Freerun Counter Clock Timestamp valid
Dataset Input				
PortDataset_DatIn	in	Ptp_PortDataset _Type	1	PTP Port Dataset input for this port
DefaultDataset_DatIn	in	Ptp_DefaultDataset _Type	1	PTP Default Dataset input for this clock
Drop Input				
Drop_ValIn	in	std_logic	1	If input frame shall be dropped
Axi Input				
AxisValid_ValIn	in	std_logic	1	AXI Stream frame input
AxisReady_ValOut	out	std_logic	1	
AxisData_DatIn	in	std_logic_vector	32	
AxisStrobe_ValIn	in	std_logic_vector	4	
AxisKeep_ValIn	in	std_logic_vector	4	
AxisLast_ValIn	in	std_logic	1	
AxisUser_DatIn	in	std_logic_vector	3	
Axi Tap Output				
AxisTapValid_ValOut	out	std_logic	1	AXI Stream frame tap output
AxisTapReady _ValOut	out	std_logic	1	
AxisTapData_DatOut	out	std_logic_vector	32	
AxisTapStrobe _ValOut	out	std_logic_vector	4	
AxisTapKeep_ValOut	out	std_logic_vector	4	
AxisTapLast_ValOut	out	std_logic	1	
AxisTapUser_DatOut	out	std_logic_vector	3	
Drop Output				
Drop_ValOut	out	std_logic	1	If output frame shall be dropped
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	

AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
Frame Info Output				
FrameInfo_DatOut	out	Ptp_FrameInfo _Type	1	Frame Information
FrameInfo_ValOut	out	Ptp_FrameInfoVal_T ype	1	Frame Information valid

Table 14: TX Processor

4.2.3 Announce Processor

This is one of the four (Announce, Delay, Management and Sync) core parts, handling all Announce message related things.

4.2.3.1 Entity Block Diagram

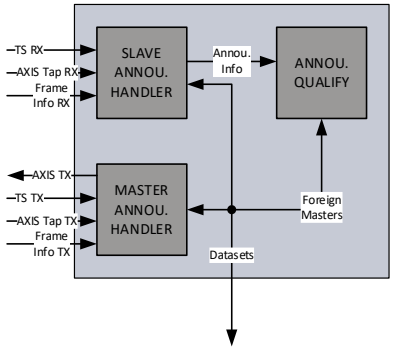

Figure 16: Announce Processor

4.2.3.2 Entity Description

Slave Announce Handler

This module detects and parses incoming Announce messages. It extracts information which is later used by the Announce Qualifier and BMC for state calculations. It accepts Announce messages from all sources and also makes no sequence check since the Announce messages from different nodes can come interleaved. For Unicast it checks if the message is the Unicast Table otherwise ignores it.

Master Announce Handler

This module periodically sends Announce messages when in the Master state. It takes information from the Default-, Port- and TimeProperties-Datasets to build an Announce message with the correct content. No timestamp and CRC is inserted and calculated, since this is done in the TX Processor for internally generated frames. It shares it's transmit path with the other Processors (Delay, Management and Sync). For Unicast it constantly fetches the whole Unicast Table and checks if an Announce message needs to be sent.

Announce Qualifier

This module takes the information from the Slave Announce Handler module and stores it in a buffer called Foreign Master Table. This Table contains the information

about up to 5 Masters and counts the number of received Announce messages per Master which is the requirement for a Master to be qualified for the BMC. This Table contains all information required for the BMC.

4.2.3.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)

Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX frame input
AxisReadyRx_ValIn	in	std_logic	1	
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi TX Input				
AxisValidTx_ValIn	in	std_logic	1	AXI Stream TX frame input
AxisReadyTx_ValIn	in	std_logic	1	
AxisDataTx_DatIn	in	std_logic_vector	32	
AxisStrobeTx_ValIn	in	std_logic_vector	4	
AxisKeepTx_ValIn	in	std_logic_vector	4	

AxisLastTx_ValIn	in	std_logic	1	
AxisUserTx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
Announce Info Output				
AnnounceReceived_ValOut	out	std_logic	1	Announce Message was received, all other values in this section are valid if set
PortIdentity_DatOut	out	Ptp_PortIdentity_Type	1	Port Identity of the sender of the received Announce Message
TimeSource_DatOut	out	std_logic_vector	8	Time Source of the sender of the received Announce Message
StepsRemoved_DatOut	out	std_logic_vector	16	Steps between the OC and the sender of the received Announce Message
FlagField_DatOut	out	std_logic_vector	16	Flag field of the received Announce Message
UtcOffset_DatOut	out	std_logic_vector	16	UTC Offset of the received Announce Message
GrandmasterIdentity_DatOut	out	Ptp_ClockIdentity_Type	1	Grandmaster Identity of the received Announce Message
GrandmasterPriority1_DatOut	out	std_logic_vector	8	Grandmaster Priori-

				ty1 of the received Announce Message
GrandmasterPriori-ty2_DatOut	out	std_logic_vector	8	Grandmaster Priori-ty2 of the received Announce Message
Grandmaster-ClockQuality_DatOut	out	Ptp_ClockQuality _Type	1	Grandmaster Clock Quality of the re-ceived Announce Message
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the OC
Frame Info RX Input				
FrameInfoRx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoRx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Frame Info TX Input				
FrameInfoTx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoTx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Timestamp RX Input				
ClockTimestampRx _DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestampRx _ValIn	in	std_logic	1	Adjusted PTP Clock Timestamp valid
Timestamp TX Input				
ClockTimestampTx _DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestampTx _ValIn	in	std_logic	1	Adjusted PTP Clock Timestamp valid
Dataset Input				
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset _DatIn	in	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatIn	in	Ptp_TimeProperties	1	PTP Time Properties

		Dataset_Type		Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset RespArray_Type	2	PTP Unicast Dataset response for this clock
UnicastDataset Resp_ValIn	in	std_logic_vector	2	PTP Unicast Dataset response for this clock valid
Dataset Output				
ParentDataset _DatOut	out	Ptp_Parent Dataset_Type	1	PTP Parent Dataset data output
ParentDataset _ValOut	out	Ptp_Parent DatasetVal_Type	1	PTP Parent Dataset valid output
ForeignMasterDataset _DatOut	out	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset data output
ForeignMasterDataset _ValOut	out	Ptp_ForeignMaster DatasetVal_Type	1	PTP Foreign Master Dataset valid output
TimeProperties Dataset_DatOut	out	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset data output
TimeProperties Dataset_ValOut	out	Ptp_TimeProperties DatasetVal_Type	1	PTP Time Properties Dataset valid output
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset ReqArray_Type	2	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic_vector	2	PTP Unicast Dataset request valid output

Table 15: Announce Processor

4.2.4 Delay Processor

This is one of the four (Announce, Delay, Management and Sync) core parts, handling all Delay message related things.

4.2.4.1 Entity Block Diagram

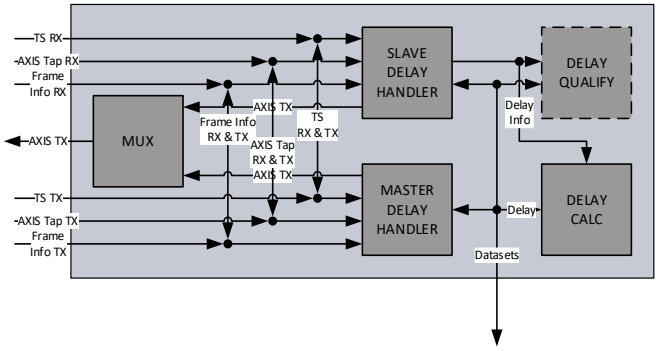

Figure 17: Delay Processor

4.2.4.2 Entity Description

Slave Delay Handler

This module periodically sends PDelay Requests messages stores the transmit timestamp and waits for the PDelay response (and PDelay Response FollowUp) messages from the other nodes. When a PDelay Response message is received it extracts and merges the frame information needed for the delay calculation from the Delay Response and Delay Response FollowUp and passes the information to the Delay Calculator block. No CRC is inserted and calculated, since this is done in the TX Processor for internally generated frames. It shares it's transmit path with the other Processors (Announce, Master Delay, Management and Sync). For Unicast it constantly fetches the whole Unicast Table and checks if an Delay message needs to be sent and for the current master it also waits for the Delay response. In E2E mode, optionally it runs the received timestamps through a Lucky Packet Filter. The Lucky Packet filter takes the fastest Delay Request messages within a configurable window, it also already received the Lucky Packet filtered Sync timestamps.

Master Delay Handler

This module detects and parses incoming PDelay Request messages and sends PDelay Response messages. It extracts information from the PDelay Requests which is used for generating the PDelay Responses. It acts in one-step mode,

means the residence time will be stored in the correction field and no PDelay Response FollowUp will be sent. For that it subtracts the RX timestamp from the correction field. The TX Processor will add the TX timestamp to the correction field in a later stage and calculates the CRC. No CRC is inserted and calculated, since this is done in the TX Processor for internally generated frames. It shares it's transmit path with the other Processors (Announce, Slave Delay, Management and Sync). For Unicast it checks if the Requestor is in the Unicast Table and if Delay response is allowed to be sent.

Delay Qualifier

This module takes the information from the Slave Delay Handler module and stores it in a buffer called Foreign Master Table. This Table contains the information about up to 5 Masters and counts the number of received Delay messages per Master which is the requirement for a Master to be qualified for the BMC.

Delay Calculator

This module calculates the delay based on the RX and TX timestamps, correction fields and information extracted from the frame by the Slave Delay Handler. It averages the Delay over the last N messages if they are all within a certain range or hard sets the delay if they are out of the window. This allows smooth measurements for the case no topology change has happened. Finally it stores the calculated delay in the Port Dataset

Multiplexer

This module multiplexes the two transmit paths from the Slave Delay Handler and Master Delay handler. Arbitration is round robin.

4.2.4.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile

UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
E2eUnicastSupport_Gen	-	boolean	1	E2E Unicast handling support
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
LuckyPacketFilter Support_Gen	-	boolean	1	If the Lucky Packet Filter shall be supported
LuckyPacketFilter Samples_Gen	-	natural	1	Maximum number of Samples in Lucky Packet Filter Win-

				dow
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Delay Processor				
AverageWindow Nanosecond_Gen	-	natural	1	Delay Averaging window, out of this window, delay is set hard
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX frame input
AxisReadyRx_ValIn	in	std_logic	1	
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi TX Input				
AxisValidTx_ValIn	in	std_logic	1	AXI Stream TX frame input
AxisReadyTx_ValIn	in	std_logic	1	
AxisDataTx_DatIn	in	std_logic_vector	32	
AxisStrobeTx_ValIn	in	std_logic_vector	4	
AxisKeepTx_ValIn	in	std_logic_vector	4	

AxisLastTx_ValIn	in	std_logic	1	
AxisUserTx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the OC
Frame Info RX Input				
FrameInfoRx_DatIn	in	Ptp_FrameInfo_Type	1	Frame Information
FrameInfoRx_ValIn	in	Ptp_FrameInfoVal_Type	1	Frame Information valid
Frame Info TX Input				
FrameInfoTx_DatIn	in	Ptp_FrameInfo_Type	1	Frame Information
FrameInfoTx_ValIn	in	Ptp_FrameInfoVal_Type	1	Frame Information valid
Timestamp RX Input				
CounterTimestampRx_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Timestamp
CounterTimestampRx_ValIn	in	std_logic	1	Freerun Counter Clock Timestamp valid
ClockTimestampRx_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Timestamp
ClockTimestampRx_ValIn	in	std_logic	1	Freerun Counter Clock Timestamp valid
Timestamp TX Input				
CounterTimestampTx_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Timestamp
CounterTimestampTx_ValIn	in	std_logic	1	Freerun Counter Clock Timestamp

				valid
ClockTimestampTx_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Timestamp
ClockTimestampTx_ValIn	in	std_logic	1	Freerun Counter Clock Timestamp valid
Redundancy Input				
RedMode_DatIn	in	Ptp_RedMode_Type	1	Redundancy Mode: Hsr_E Prp_E No_E
RedSeqResp_DatIn	in	Ptp_RedSeqResp Array_Type	2	Sequence Number Response
RedSeqResp_ValIn	in	Ptp_RedSeqRespVal Array_Type	2	Sequence Number Response valid
Redundancy Output				
RedSeqReq_DatOut	out	Ptp_RedSeqReq Array_Type	2	Sequence Number Request
RedSeqReq_ValOut	out	Ptp_RedSeqReqVal Array_Type	2	Sequence Number Request valid
Sync Info Input				
SyncReceived_ValIn	in	std_logic	1	Sync Message was received
CorrectionField SyncRx_DatIn	in	std_logic_vector	64	Correction field of received Sync
Timestamp SyncRx_DatIn	in	std_logic_vector	80	Timestamp when Sync was received
Timestamp SyncTx_DatIn	in	std_logic_vector	80	Timestamp when Sync was sent
Dataset Input				
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset RespArray_Type	2	PTP Unicast Dataset response for this clock

UnicastDataset Resp_ValIn	in	std_logic_vector	2	PTP Unicast Dataset response for this clock valid
Dataset Output				
PortDataset_DatOut	out	Ptp_Port Dataset_Type	1	PTP Port Dataset data output
PortDataset_ValOut	out	Ptp_Port DatasetVal_Type	1	PTP Port Dataset valid output valid output valid output
Current Dataset_DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset data output
Current Dataset_ValOut	out	Ptp_Current DatasetVal_Type	1	PTP Current Dataset valid output valid output valid output
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset ReqArray_Type	2	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic_vector	2	PTP Unicast Dataset request valid output

Table 16: Delay Processor

4.2.5 Sync Processor

This is one of the four (Announce, Delay, Management and Sync) core parts, handling all Sync message related things.

4.2.5.1 Entity Block Diagram

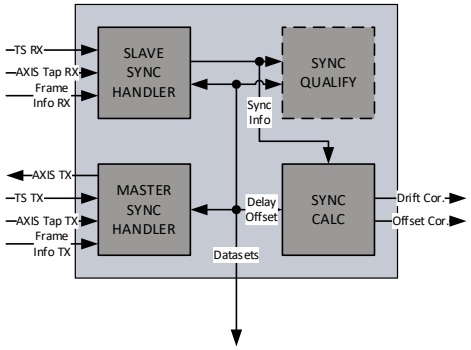

Figure 18: Sync Processor

4.2.5.2 Entity Description

Slave Sync Handler

This module detects and parses incoming Sync (and Sync FollowUp) messages when in the Slave or Uncalibrated state. It extracts information which is later used by the Sync Calculator for Offset and Drift calculation. It only accepts Sync messages from the current Master and checks sequence numbers and merges the information of Sync and Sync FollowUp and passes it to the Sync Calculator. Optionally it runs the received timestamps through a Lucky Packet Filter. The Lucky Packet filter takes the fastest Sync messages within a configurable window.

Master Sync Handler

This module periodically sends Sync messages when in the Master state. It takes information from the Default- and Port-Datasets to build a Sync message with the correct content. This module is one-step capable, means the timestamp will be inserted on the fly and no Sync FollowUp will be sent. No timestamp and CRC is inserted and calculated, since this is done in the TX Processor for internally generated frames. It shares it's transmit path with the other Processors (Announce, Delay and Management). For Unicast it constantly fetches the whole Unicast Table and checks if an Sync message needs to be sent.

Sync Qualifier

This module takes the information from the Slave Sync Handler module and stores it in a buffer called Foreign Master Table. This Table contains the information about up to 5 Masters and counts the number of received Sync messages per Master which is the requirement for a Master to be qualified for the BMC.

Sync Calculator

This module calculates with the information from the Sync and the measured delay the Drift and Offset against the Master clock. First the Drift is calculated with the information of the last calculated Offset. Then the Offset is calculated with information of the newly calculated Drift. This way the two correction values are not influencing each other. Once the clock is synchronized quite good most corrections are only done with Drift corrections anymore.

4.2.5.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall

				support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
E2eUnicastSupport_Gen	-	boolean	1	E2E Unicast handling support
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
LuckyPacketFilter Support_Gen	-	boolean	1	If the Lucky Packet Filter shall be supported
LuckyPacketFilter Samples_Gen	-	natural	1	Maximum number of Samples in Lucky Packet Filter Window
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Sync Processor				
ClockClkPeriodNanosecond_Gen	-	natural	1	Clock period of the PTP clock, used for claculation
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset

Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX frame input
AxisReadyRx_ValIn	in	std_logic	1	
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi TX Input				
AxisValidTx_ValIn	in	std_logic	1	AXI Stream TX frame input
AxisReadyTx_ValIn	In	std_logic	1	
AxisDataTx_DatIn	in	std_logic_vector	32	
AxisStrobeTx_ValIn	in	std_logic_vector	4	
AxisKeepTx_ValIn	in	std_logic_vector	4	
AxisLastTx_ValIn	in	std_logic	1	
AxisUserTx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
Sync Info Output				
SyncReceived_ValOut	out	std_logic	1	Sync Message was received
CorrectionField SyncRx_DatOut	out	std_logic_vector	64	Correction field of

				received Sync
Timestamp SyncRx_DatOut	out	std_logic_vector	80	Timestamp when Sync was received
Timestamp SyncTx_DatOut	out	std_logic_vector	80	Timestamp when Sync was sent
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the OC
Frame Info RX Input				
FrameInfoRx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoRx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Frame Info TX Input				
FrameInfoTx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoTx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Timestamp RX Input				
ClockTimestampRx _DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestampRx _ValIn	in	std_logic	1	Adjusted PTP Clock Timestamp valid
Timestamp TX Input				
ClockTimestampTx _DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Timestamp
ClockTimestampTx _ValIn	in	std_logic	1	Adjusted PTP Clock Timestamp valid
Dataset Input				
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
CurrentDataset _DatOut	in	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset RespArray_Type	2	PTP Unicast Dataset response for this

				clock
UnicastDataset Resp_ValIn	in	std_logic_vector	2	PTP Unicast Dataset response for this clock valid
Dataset Output				
CurrentDataset _DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset data output
CurrentDataset _ValOut	out	Ptp_Current DatasetVal_Type	1	PTP Current Dataset valid output
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset ReqArray_Type	2	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic_vector	2	PTP Unicast Dataset request valid output
Offset Adjustment Output				
OffsetAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Calculated new Offset between Master and Slave
OffsetAdjustment _ValOut	out	std_logic;	1	Calculated new Offset valid
Drift Adjustment Output				
DriftAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Calculated new Drift between Master and Slave
DriftAdjustment _ValOut	out	std_logic;	1	Calculated new Drift valid
Offset Adjustment Input				
OffsetAdjustment _DatIn	in	Clk_TimeAdjustment _Type	1	Calculated new Offset after the PI Servo loop
OffsetAdjustment _ValIn	in	std_logic;	1	Calculated new Offset after the PI Servo loop valid
Drift Adjustment Input				
DriftAdjustment _DatIn	in	Clk_TimeAdjustment _Type	1	Calculated new Drift after the PI Servo loop
DriftAdjustment _ValIn	in	std_logic	1	Calculated new Drift after the PI Servo loop valid

Table 17: Sync Processor

4.2.6 Management Processor

This is one of the four (Announce, Delay, Management and Sync) core parts, handling all Management message related things. This module is optional since a lot of the PTP Profiles removed PTP Management from the scope by purpose for security reasons.

4.2.6.1 Entity Block Diagram

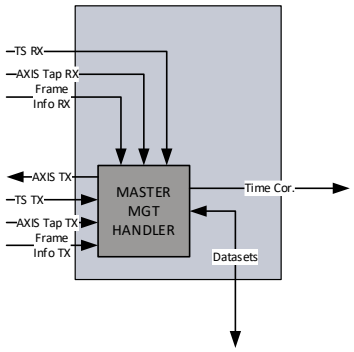

Figure 19: Management Processor

4.2.6.2 Entity Description

Master Management Handler

This module detects and parses incoming Management messages and sends Management Messages (Responses). It supports most of the PTP Management messages except some which were disabled because of security reasons or because they are PTP Profile specific. It extracts the command and does the corresponding action: For a GET it gets the requested value from the corresponding Datasets and sends the value with a Response Management message; For a SET it takes the value from the Management message, sets the requested value in the corresponding Dataset and sends an Acknowledge Management message. Also the time can be set, this via the time adjustment interface similar to the Drift and Offset.

4.2.6.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock

DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Management Processor				
NumberOfPorts_Gen	-	natural	1	Number of Ports of

				that Clock
ManufacturerIdenti-ty_Gen	-	Common_Byte_Type	3	Manufacturing Identity string
ProductDescrip-tion_Gen	-	string	32	Product Description string
RevisionData_Gen	-	string	32	Product Revision string
UserDescription_Gen	-	string	32	User Description string
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX frame input
AxisReadyRx_ValIn	in	std_logic	1	
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi TX Input				
AxisValidTx_ValIn	in	std_logic	1	AXI Stream TX frame input
AxisReadyTx_ValIn	in	std_logic	1	
AxisDataTx_DatIn	in	std_logic_vector	32	
AxisStrobeTx_ValIn	in	std_logic_vector	4	
AxisKeepTx_ValIn	in	std_logic_vector	4	
AxisLastTx_ValIn	in	std_logic	1	

AxisUserTx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the OC
Frame Info RX Input				
FrameInfoRx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoRx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Frame Info TX Input				
FrameInfoTx_DatIn	in	Ptp_FrameInfo _Type	1	Frame Information
FrameInfoTx_ValIn	in	Ptp_FrameInfoVal_T ype	1	Frame Information valid
Dataset Input				
CurrentDataset_DatIn	in	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset _DatIn	in	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset Resp_Type	1	PTP Unicast Dataset response for this

				clock
UnicastDataset Resp_ValIn	in	std_logic	1	PTP Unicast Dataset response for this clock valid
Dataset Output				
PortDataset _DatOut	out	Ptp_Port Dataset_Type	1	PTP Port Dataset data output
PortDataset _ValOut	out	Ptp_Port DatasetVal_Type	1	PTP Port Dataset valid output
CurrentDataset _DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset data output
CurrentDataset _ValOut	out	Ptp_Current DatasetVal_Type	1	PTP Current Dataset valid output
DefaultDataset _DatOut	out	Ptp_Default Dataset_Type	1	PTP Default Dataset data output
DefaultDataset _ValOut	out	Ptp_Default DatasetVal_Type	1	PTP Default Dataset valid output
ParentDataset _DatOut	out	Ptp_Parent Dataset_Type	1	PTP Parent Dataset data output
ParentDataset _ValOut	out	Ptp_Parent DatasetVal_Type	1	PTP Parent Dataset valid output
ForeignMasterDataset _DatOut	out	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset data output
ForeignMasterDataset _ValOut	out	Ptp_ForeignMaster DatasetVal_Type	1	PTP Foreign Master Dataset valid output
TimeProperties Dataset_DatOut	out	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset data output
TimeProperties Dataset_ValOut	out	Ptp_TimeProperties DatasetVal_Type	1	PTP Time Properties Dataset valid output
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset Req_Type	1	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic	1	PTP Unicast Dataset request valid output
Time Adjustment Output				
TimeAdjustment _DatOut	out	Clk_TimeAdjustment _Type	1	Time to set hard
TimeAdjustment _ValOut	out	std_logic	1	Time valid

Table 18: Management Processor

4.2.7 Signaling Processor

This module is handling all Signaling message related things. This module is optional since a lot of the PTP Profiles don't require Signaling.

4.2.7.1 Entity Block Diagram

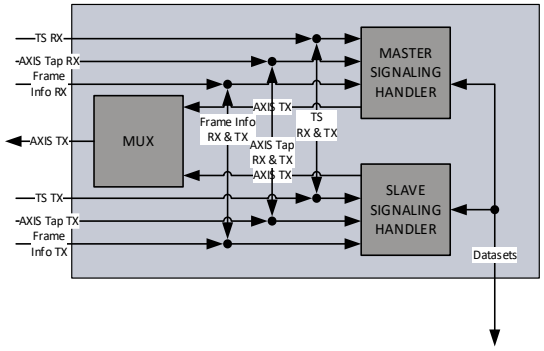

Figure 20: Signaling Processor

4.2.7.2 Entity Description

Master Signaling Handler

This module detects and parses incoming Signaling messages and acts accordingly; e.g. setting frame intervals etc. It is also in charge of granting Unicast transmission

Slave Signaling Handler

This module sends periodic Signaling messages e.g. requesting intervals or send an announcement that it supports a specific profile etc. It is also in charge of requesting Unicast transmission. For Unicast it constantly fetches the whole Unicast Table and checks if an Signaling refresh message needs to be sent.

4.2.7.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile

PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Signaling Processor				
NumberOfPorts_Gen	-	natural	1	Number of Ports of that Clock
Ports				

System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX frame input
AxisReadyRx_ValIn	in	std_logic	1	
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi TX Input				
AxisValidTx_ValIn	in	std_logic	1	AXI Stream TX frame input
AxisReadyTx_ValIn	in	std_logic	1	
AxisDataTx_DatIn	in	std_logic_vector	32	
AxisStrobeTx_ValIn	in	std_logic_vector	4	
AxisKeepTx_ValIn	in	std_logic_vector	4	
AxisLastTx_ValIn	in	std_logic	1	
AxisUserTx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the

				OC
Frame Info RX Input				
FrameInfoRx_DatIn	in	Ptp_FrameInfo_Type	1	Frame Information
FrameInfoRx_ValIn	in	Ptp_FrameInfoVal_Type	1	Frame Information valid
Frame Info TX Input				
FrameInfoTx_DatIn	in	Ptp_FrameInfo_Type	1	Frame Information
FrameInfoTx_ValIn	in	Ptp_FrameInfoVal_Type	1	Frame Information valid
Dataset Input				
CurrentDataset_DatIn	in	Ptp_CurrentDataset_Type	1	PTP Current Dataset input for this clock
ParentDataset_DatIn	in	Ptp_ParentDataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset_DatIn	in	Ptp_ForeignMasterDataset_Type	1	PTP Foreign Master Dataset input for this clock
TimePropertiesDataset_DatIn	in	Ptp_TimePropertiesDataset_Type	1	PTP Time Properties Dataset input for this clock
PortDataset_DatIn	in	Ptp_PortDataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_DefaultDataset_Type	1	PTP Default Dataset input for this clock
UnicastDatasetResp_DatIn	in	Ptp_UnicastDatasetRespArray_Type	2	PTP Unicast Dataset response for this clock
UnicastDatasetResp_ValIn	in	std_logic_vector	2	PTP Unicast Dataset response for this clock valid
Dataset Output				
PortDataset_DatOut	out	Ptp_PortDataset_Type	1	PTP Port Dataset data output
PortDataset_ValOut	out	Ptp_PortDatasetVal_Type	1	PTP Port Dataset valid output
CurrentDataset_DatOut	out	Ptp_CurrentDataset_Type	1	PTP Current Dataset data output

CurrentDataset_ValOut	out	Ptp_CurrentDatasetVal_Type	1	PTP Current Dataset valid output
DefaultDataset_DatOut	out	Ptp_DefaultDataset_Type	1	PTP Default Dataset data output
DefaultDataset_ValOut	out	Ptp_DefaultDatasetVal_Type	1	PTP Default Dataset valid output
ParentDataset_DatOut	out	Ptp_ParentDataset_Type	1	PTP Parent Dataset data output
ParentDataset_ValOut	out	Ptp_ParentDatasetVal_Type	1	PTP Parent Dataset valid output
ForeignMasterDataset_DatOut	out	Ptp_ForeignMasterDataset_Type	1	PTP Foreign Master Dataset data output
ForeignMasterDataset_ValOut	out	Ptp_ForeignMasterDatasetVal_Type	1	PTP Foreign Master Dataset valid output
TimePropertiesDataset_DatOut	out	Ptp_TimePropertiesDataset_Type	1	PTP Time Properties Dataset data output
TimePropertiesDataset_ValOut	out	Ptp_TimePropertiesDatasetVal_Type	1	PTP Time Properties Dataset valid output
UnicastDatasetReq_DatOut	out	Ptp_UnicastDatasetReqArray_Type	2	PTP Unicast Dataset request data output
UnicastDatasetReq_ValOut	out	std_logic_vector	2	PTP Unicast Dataset request valid output

Table 19: Signaling Processor

4.2.8 MAC Processor

This module is handling all MAC Address Resolution related protocols like ARP and ICMP. It is only required for Unicast.

4.2.8.1 Entity Block Diagram

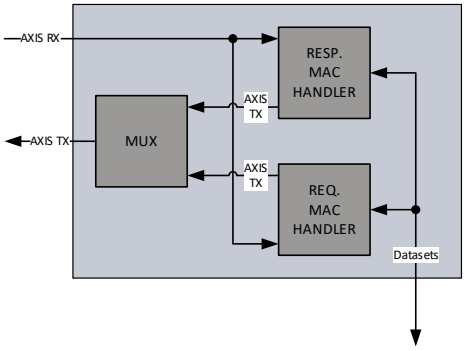

Figure 21: Mac Processor

4.2.8.2 Entity Description

Responder MAC Handler

This module responses to ARP Requests and ICMP Neighbor Solicitation Requests.

Requester MAC Handler

This module requests ARP Requests and ICMP Neighbor Solicitation Requests for Unicast Nodes.

4.2.8.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile

TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
RedTagSupport_Gen	-	boolean	1	If HSR/PRP shall be supported
Sim_Gen	-	boolean	1	If in Testbench simulation mode
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
Axi RX Input				
AxisValidRx_ValIn	in	std_logic	1	AXI Stream RX

AxisReadyRx_ValIn	in	std_logic	1	frame input
AxisDataRx_DatIn	in	std_logic_vector	32	
AxisStrobeRx_ValIn	in	std_logic_vector	4	
AxisKeepRx_ValIn	in	std_logic_vector	4	
AxisLastRx_ValIn	in	std_logic	1	
AxisUserRx_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	
PTP Statemachine Input				
Ptp_State_StaIn	in	Ptp_State_Type	1	Current State of the OC
Dataset Input				
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset RespArray_Type	2	PTP Unicast Dataset response for this clock
UnicastDataset Resp_ValIn	in	std_logic_vector	2	PTP Unicast Dataset response for this clock valid
Dataset Output				
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset ReqArray_Type	2	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic_vector	2	PTP Unicast Dataset request valid output

Table 20: MAC Processor

4.2.9 Best Master Clock Algorithm

This is another core module of this implementation. It defines the State of the clock

4.2.9.1 Entity Block Diagram

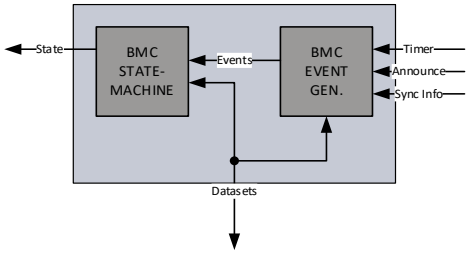

Figure 22: BMC Algorithm

4.2.9.2 Entity Description

BMC Statemachine

This module implements the state machine according to IEEE1588-2019/2008 clause 9.2.

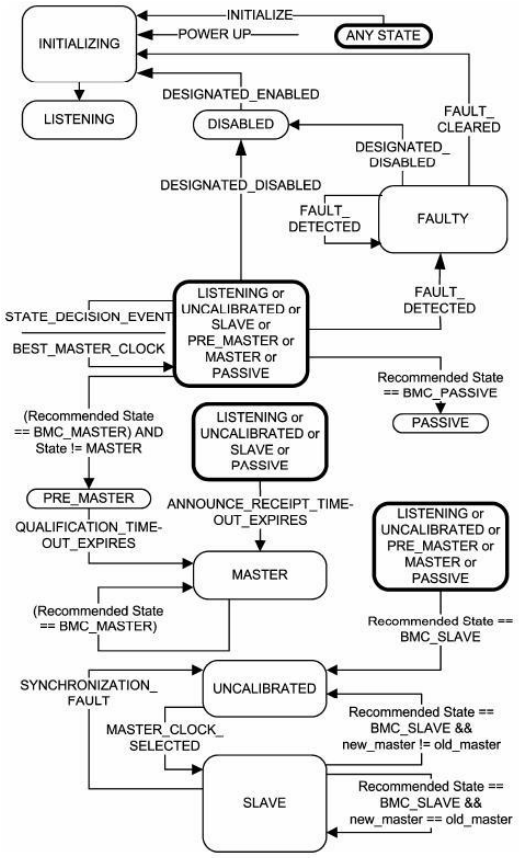

Figure 23: Statemachine (source IEEE1588-2019/2008 Std)

According to the SlaveOnly flag the state machine changes its behavior. The state machine can handle one event per clock cycle. If multiple events occur simultaneously, only the event with the highest priority (according to the implementer) is handled. In addition for ITU Profiles the adapted BMCA is implemented.

BMC Event Generator

All PTP events according to IEEE1588-2019/2008 clause 9.2.6 are generated by this module. The events are generated based on received messages, calculated Offset values and Dataset values. The timer event signal is sequentialized (by one clock cycle each) so timing driven events do not happen simultaneously so no event is lost.
The following events are defined by IEEE1588-2019/2008 and the corresponding State transitions can be seen in Figure 23:

- POWERUP (implicit)
- INITIALIZE (implicit by a reset)
- FAULT_DETECTED (never happens)
- FAULT_CLEARED (because fault never happens this never happens)
- STATE_DECISION_EVENT (timed)
- ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES (timed)
- QUALIFICATION_TIMEOUT_EXPIRES (timed)
- DESIGNATED_ENABLED (timed, when enabling the clock)
- DESIGNATED_DISABLED (timed, when disabling the clock)
- MASTER_CLOCK_SELECTED (timed)
- SYNCHRONIZATION_FAULT (timed)

4.2.9.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile

UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
ExtSync_Gen	-	boolean	1	If in Master mode, synchronized from externally
ClockType_Gen	-	Ptp_ClockType_Typ	1	What Kind of Clock this is (OC)
Sim_Gen	-	boolean	1	If in Testbench simulation mode
BMC				
MaxOffset_Gen	-	natural	1	If Offset is larger than this change into Uncalibrated state if Slave
UnicastSupport_Gen	-	boolean	1	Support for Unicast Messages
Ports				
System				

SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Enable Input				
Enable_EnaIn	in	std_logic	1	Clock Enabled
Time Input				
ClockTime_DatIn	in	Clk_Time_Type	1	Adjusted PTP Clock Time
ClockTime_ValIn	in	std_logic	1	Adjusted PTP Clock Time valid
Timer				
Timer1ms_EvtIn	in	std_logic	1	Adjusted PTP Clock aligned 1 millisecond Timer event
PTP Statemachine Output				
Ptp_State_StaOut	out	Ptp_State_Type	1	Current State of the OC
Dataset Input				
CurrentDataset_DatIn	in	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset _DatIn	in	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
TimeProperties ExtSync_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties input from external synchronizationm for this clock
Dataset Output				
PortDataset _DatOut	out	Ptp_Port Dataset_Type	1	PTP Port Dataset data output
PortDataset _ValOut	out	Ptp_Port	1	PTP Port Dataset

		DatasetVal_Type		valid output
CurrentDataset _DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset data output
CurrentDataset _ValOut	out	Ptp_Current DatasetVal_Type	1	PTP Current Dataset valid output
ParentDataset _DatOut	out	Ptp_Parent Dataset_Type	1	PTP Parent Dataset data output
ParentDataset _ValOut	out	Ptp_Parent DatasetVal_Type	1	PTP Parent Dataset valid output
TimeProperties Dataset_DatOut	out	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset data output
TimeProperties Dataset_ValOut	out	Ptp_TimeProperties DatasetVal_Type	1	PTP Time Properties Dataset valid output
Announce Info Input				
AnnounceRe-ceived_ValIn	in	std_logic	1	Announce Message was received, all other values in this section are valid if set
PortIdentity_DatIn	in	Ptp_PortIdentity _Type	1	Port Identity of the sender of the re-ceived Announce Message
TimeSource_DatIn	in	std_logic_vector	8	Time Source of the sender of the re-ceived Announce Message
StepsRemoved_DatIn	in	std_logic_vector	16	Steps between the OC and the sender of the received Announce Message
FlagField_DatIn	in	std_logic_vector	16	Flag field of the received Announce Message
UtcOffset_DatIn	in	std_logic_vector	16	UTC Offset of the received Announce Message
GrandmasterIdenti-ty_DatIn	in	Ptp_ClockIdentity	1	Grandmaster Identi-

		_Type		ty of the received Announce Message
GrandmasterPriority1_DatIn	in	std_logic_vector	8	Grandmaster Priority1 of the received Announce Message
GrandmasterPriority2_DatIn	in	std_logic_vector	8	Grandmaster Priority2 of the received Announce Message
GrandmasterClockQuality_DatIn	in	Ptp_ClockQuality_Type	1	Grandmaster Clock Quality of the received Announce Message
Sync Info Input				
SyncReceived_ValIn	in	std_logic	1	Sync Message was received

Table 21: BMC Algorithm

4.2.10 Datasets

4.2.10.1 Entity Block Diagram

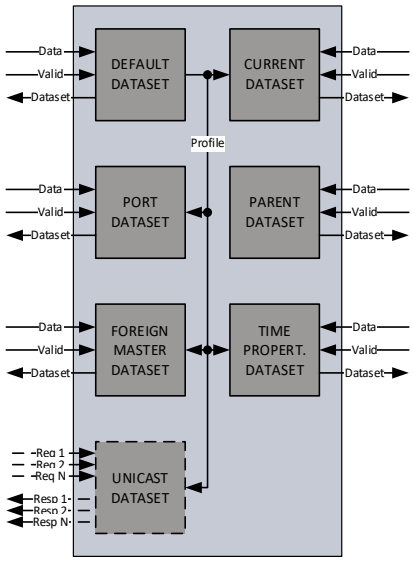

Figure 24: Datasets

4.2.10.2 Entity Description

Default Dataset

This module contains the storage for the Dataset members of the Default Dataset according to IEEE1588-2019 clause 8.2.1:

```
Profile                     : Ptp_Profile_Type;
Layer                       : Ptp_Layer_type;
Vlan                        : Ptp_Vlan_Type;
VlanEnable                  : std_logic;
Dscp                        : std_logic_vector(5 downto 0);
Ip                          : Common_Byte_Type(0 to 15);
TwoStepFlag                 : std_logic;
ClockIdentity               : Ptp_ClockIdentity_Type;
NumberPorts                 : std_logic_vector(15 downto 0);
ClockQuality                : Ptp_ClockQuality_Type;
Priority1                   : std_logic_vector(7 downto 0);
Priority2                   : std_logic_vector(7 downto 0);
DomainNumber                : std_logic_vector(7 downto 0);
SlaveOnly                   : std_logic;
GrandmasterId               : std_logic_vector(15 downto 0);
GrandmasterTimeInaccuracy   : std_logic_vector(31 downto 0);
```

Port Dataset

This module contains the storage for the Dataset members of the Port Dataset according to IEEE1588-2019 clause 8.2.5:

```
PortIdentity                  : Ptp_PortIdentity_Type;
PortState                     : std_logic_vector(7 downto 0);
LogMinDelayReqInterval        : std_logic_vector(7 downto 0);
PeerMeanPathDelayValid        : std_logic;
PeerMeanPathDelay             : std_logic_vector(63 downto 0);
DelayReceiptTimeout           : std_logic_vector(7 downto 0);
LogAnnounceInterval           : std_logic_vector(7 downto 0);
AnnounceReceiptTimeout        : std_logic_vector(7 downto 0);
LogSyncInterval               : std_logic_vector(7 downto 0);
SyncReceiptTimeout            : std_logic_vector(7 downto 0);
DelayMechanism                : std_logic_vector(7 downto 0);
LogMinPdelayReqInterval       : std_logic_vector(7 downto 0);
VersionNumber                 : std_logic_vector(3 downto 0);
```

Foreign Master Dataset

This module contains the storage for the Dataset members of the Foreign Master Dataset according to IEEE1588-2019 clause 9.3.2.4:

```
EntryValid                    : std_logic;
SequenceIdentity              : std_logic_vector(15 downto 0);
CurrentUtcOffset              : std_logic_vector(15 downto 0);
CurrentUtcOffsetValid         : std_logic;
Leap59                        : std_logic;
Leap61                        : std_logic;
TimeTraceable                 : std_logic;
FrequencyTraceable            : std_logic;
PtpTimescale                  : std_logic;
TimeSource                    : std_logic_vector(7 downto 0);
ForeignMasterPortIdentity     : Ptp_PortIdentity_Type;
ForeignMasterAnnounceMessages : Ptp_AnnounceMessages_Type(4 downto 0);
StepsRemoved                  : std_logic_vector(15 downto 0);
GrandmasterIdentity           : Ptp_ClockIdentity_Type;
GrandmasterClockQuality       : Ptp_ClockQuality_Type;
GrandmasterPriority1          : std_logic_vector(7 downto 0);
GrandmasterPriority2          : std_logic_vector(7 downto 0);
```

Current Dataset

This module contains the storage for the Dataset members of the Current Dataset according to IEEE1588-2019 clause 8.2.2:

```
StepsRemoved                  : std_logic_vector(15 downto 0);
OffsetFromMaster              : std_logic_vector(63 downto 0);
MeanPathDelay                 : std_logic_vector(63 downto 0);
```

Parent Dataset

This module contains the storage for the Dataset members of the Parent Dataset according to IEEE1588-2019 clause 8.2.3:

```
ParentPortIdentity             : Ptp_PortIdentity_Type;
ParentStats                    : std_logic;
ObsParentOffsetScaledLogVaria  : std_logic_vector(15 downto 0);
ObsParentClockPhaseChangeRate  : std_logic_vector(31 downto 0);
GrandmasterIdentity            : Ptp_ClockIdentity_Type;
GrandmasterClockQuality        : Ptp_ClockQuality_Type;
GrandmasterPriority1           : std_logic_vector(7 downto 0);
GrandmasterPriority2           : std_logic_vector(7 downto 0);
GrandmasterId                  : std_logic_vector(15 downto 0);
GrandmasterTimeInaccuracy      : std_logic_vector(31 downto 0);
NetworkTimeInaccuracy          : std_logic_vector(31 downto 0);
```

Time Properties Dataset

This module contains the storage for the Dataset members of the TimeProperties Dataset according to IEEE1588-2019 clause 8.2.4:

```
CurrentUtcOffset               : std_logic_vector(15 downto 0);
CurrentUtcOffsetValid          : std_logic;
Leap59                         : std_logic;
Leap61                         : std_logic;
TimeTraceable                  : std_logic;
FrequencyTraceable             : std_logic;
PtpTimescale                   : std_logic;
TimeSource                     : std_logic_vector(7 downto 0);
CurrentOffset                  : std_logic_vector(31 downto 0);
JumpSeconds                    : std_logic_vector(31 downto 0);
TimeOfNextJumpSeconds          : std_logic_vector(47 downto 0);
DisplayNameLength              : std_logic_vector(7 downto 0);
DisplayName                    : Common_Byte_Type(11 downto 0);
```

Unicast Dataset

The Unicast Dataset is a Table of Nodes with the following information:

```
EntryValid                     : std_logic;
Flags                          : std_logic_vector(6 downto 0);
Mac                            : Common_Byte_Type(0 to 5);
Ip                             : Common_Byte_Type(0 to 15);
Priority                       : std_logic_vector(7 downto 0);
LogAnnounceInterval            : std_logic_vector(7 downto 0);
AnnounceDuration               : std_logic_vector(31 downto 0);
AnnounceRefresh                : std_logic_vector(31 downto 0);
AnnounceTimer                  : std_logic_vector(31 downto 0); -- in us
LogSyncInterval                : std_logic_vector(7 downto 0);
SyncDuration                   : std_logic_vector(31 downto 0);
SyncRefresh                    : std_logic_vector(31 downto 0);
SyncTimer                      : std_logic_vector(31 downto 0); -- in us
LogMinDelayInterval            : std_logic_vector(7 downto 0);
```

```
DelayDuration                  : std_logic_vector(31 downto 0);
DelayRefresh                   : std_logic_vector(31 downto 0);
DelayTimer                     : std_logic_vector(31 downto 0); -- in us
AnnounceSequenceIdentity       : std_logic_vector(15 downto 0);
SyncSequenceIdentity           : std_logic_vector(15 downto 0);
DelaySequenceIdentity          : std_logic_vector(15 downto 0);
SignalingSequenceIdentity      : std_logic_vector(15 downto 0);
SignalingAnnounceRetry         : std_logic_vector(3 downto 0);
SignalingSyncRetry             : std_logic_vector(3 downto 0);
SignalingDelayRetry            : std_logic_vector(3 downto 0);
MacRefresh                     : std_logic_vector(31 downto 0);
```

4.2.10.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile
TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
NrOfUnicast Ports_Gen	-	natural	1	Number of Ports that want to access the Unicast Table
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay

				mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
E2eUnicastSupport_Gen	-	boolean	1	E2E Unicast handling support
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
Signaling Support_Gen	-	boolean	1	Support for Signaling frames
Asymmetry Support_Gen	-	boolean	1	Support for Asymmetry corrections
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Dataset Input				
CurrentDataset_DatIn	in	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
CurrentDataset_ValIn	in	Ptp_Current DatasetVal_Type	1	PTP Current Dataset input for this clock
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ParentDataset_ValIn	in	Ptp_Parent DatasetVal_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset_DatIn	in	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock

ForeignMasterDataset_ValIn	in	Ptp_ForeignMaster DatasetVal_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset input for this clock
TimeProperties Dataset_ValIn	in	Ptp_TimeProperties DatasetVal_Type	1	PTP Time Properties Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
PortDataset_ValIn	in	Ptp_Port DatasetVal_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
DefaultDataset_ValIn	in	Ptp_Default DatasetVal_Type	1	PTP Default Dataset input for this clock
UnicastDataset Req_DatIn	in	Ptp_UnicastDataset ReqArray_Type	NrOf Unicast Ports_Gen	PTP Unicast Dataset request data output
UnicastDataset Req_ValIn	in	std_logic_vector	NrOf Unicast Ports_Gen	PTP Unicast Dataset request valid output
Dataset Output				
CurrentDataset _DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
ParentDataset _DatOut	out	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset _DatOut	out	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatOut	out	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset input for this clock
PortDataset _DatOut	out	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset _DatOut	out	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatOut	out	Ptp_UnicastDataset RespArray_Type	NrOf Unicast Ports_Gen	PTP Unicast Dataset response for this

				clock
UnicastDataset Resp_ValOut	out	std_logic_vector	NrOf Unicast Ports_Gen	PTP Unicast Dataset response for this clock valid

Table 22: Datasets

4.2.11 Clock Counter

4.2.11.1Entity Block Diagram

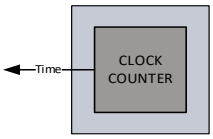

Figure 25: Clock Counter

4.2.11.2 Entity Description

Clock Counter

This is a free-running counter with nanosecond resolution in a 32 bit second and 32 bit nanosecond format. It is used by the Delay and Residence time measurements. The reason for this is mainly based on the fact that this is also used for the PTP Transparent clock and no cascading PI servo loops shall be introduced. It can take any input frequency, also non integer values with fractions. For the PTP Ordinary Clock this is run on the system clock. The time domain is TAI, so after every reset the clock starts from 1.1.1970 at midnight.

4.2.11.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
Counter Clock				
ClockClkPeriodNano-second_Gen	-	natural	1	Integer Clock Period
ClockClkPeriodFract-Num_Gen	-	natural	1	Fractional Clock Period Numerator
ClockClkPeriod-FractDeNum_Gen	-	natural	1	Fractional Clock Period Denumerator
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Time Output				
ClockTime_DatIn	in	Clk_Time_Type	1	Freerun Counter Clock Time

ClockTime_ValIn	in	std_logic	1	Freerun Counter Clock Time valid

Table 23: Clock Counter

4.2.12 Ethernet Interface Adapter

4.2.12.1 Entity Block Diagram

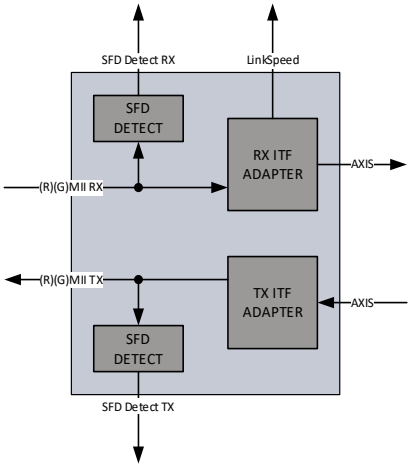

Figure 26: Ethernet Interface Adapter

4.2.12.2 Entity Description

SFD Detector

This module detects the Start of Frame Delimiter (SFD) on the (R)(G)MII stream. It runs directly on the (R)(G)MII clock domain for minimal jitter on the timestamp point detection. Once the SFD is detected, an event is signaled which is used by the timestamper.

RX Interface Adapter

This module convert the Media Independent Interface (R)(G)MII data stream (2/4/8bit) into a 32bit AXI stream. First bytes on the cable are mapped to the AXI MSB of the data array. It contains an asynchronous Fifo to on one hand do clock domain crossing from the external clock to the system clock and on the other hand also to minimal buffer data for speed differences. The Fifo size is kept quite small to assure correct timestamp alignment with the frame. It converts the different data widths into a 32bit block AXI stream. The Preamble and SFD are removed on reception. Also, it detects the link speed based on the interface clock.

TX Interface Adapter

This module convert the 32bit AXI stream into a Media Independent Interface (R)(G)MII data stream (2/4/8bit) which is continuous. The MSB of the AXI data array is mapped to the first byte on the cable. It contains an asynchronous Fifo to

on one hand do clock domain crossing from the system clock to the external clock and on the other hand also to minimal buffer data for speed differences. The Fifo size is kept quite small to assure correct timestamp alignment with the frame. It converts the 32bit block AXI stream into the different data widths. The Preamble and SFD are added before transmission. It also assures the correct interframe gap between frames.

4.2.12.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
Interface Adapter				
ClockClkPeriodNano-second_Gen	-	natural	1	Integer Clock Period
IoFf_Gen	-	boolean	1	Shall IO flip flops be instantiated
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
(R)(G)Mii RX Clk/Rst Input				
(R)(G)MiiRxClk_ClkIn	in	std_logic	1	RX Clock
(R)(G)MiiRxRstN_RstIn	in	std_logic	1	Reset aligned with RX Clock
(R)(G)Mii TX Clk/Rst Input				
(R)(G)MiiTxClk_ClkIn	in	std_logic	1	TX Clock
(R)(G)MiiTxRstN_RstIn	in	std_logic	1	Reset aligned with TX Clock
(R)(G)Mii RX Data Input/Output				
(R)(G)MiiRxDv_Ena	In/ out	std_logic	1	RX Data valid
(R)(G)MiiRxErr_Ena	In/ out	std_logic	1	RX Error
(R)(G)MiiRxData_Dat	In/ out	std_logic_vector	2-8	RX Data MII:4, RMII:2, GMII:8, RGMII:4
(R)(G)MiiCol_Dat	In/ out	std_logic	1	Collision

(R)(G)MiiCrs_Dat	In/ out	std_logic	1	Carrier Sense
(R)(G)Mii TX Data Input				
(R)(G)MiiTxEn_Ena	In/ out	std_logic	1	TX Data valid
(R)(G)MiiTxErr_Ena	In/ out	std_logic	1	TX Error
(R)(G)MiiTxData_Dat	In/ out	std_logic_vector	2-8	TX Data MII:4, RMII:2, GMII:8, RGMII:4
Link Speed Output				
LinkSpeed_DatOut	out	Common_ LinkSpeed_Type	1	Link Speed of the interface
SfdDetected Output				
(R)(G)MiiInSfdDetected_EvtOut	out	std_logic	1	Start of Frame Delimiter detected
(R)(G)MiiOutSfdDetected_EvtOut	out	std_logic	1	Start of Frame Delimiter detected
Axi Input				
AxisValid_ValIn	in	std_logic	1	AXI Stream frame input
AxisReady_ValOut	out	std_logic	1	
AxisData_DatIn	in	std_logic_vector	32	
AxisStrobe_ValIn	in	std_logic_vector	4	
AxisKeep_ValIn	in	std_logic_vector	4	
AxisLast_ValIn	in	std_logic	1	
AxisUser_DatIn	in	std_logic_vector	3	
Axi Output				
AxisValid_ValOut	out	std_logic	1	AXI Stream frame output
AxisReady_ValIn	in	std_logic	1	
AxisData_DatOut	out	std_logic_vector	32	
AxisStrobe_ValOut	out	std_logic_vector	4	
AxisKeep_ValOut	out	std_logic_vector	4	
AxisLast_ValOut	out	std_logic	1	
AxisUser_DatOut	out	std_logic_vector	3	

Table 24: Ethernet Interface Adapter

4.2.13 Registerset

4.2.13.1 Entity Block Diagram

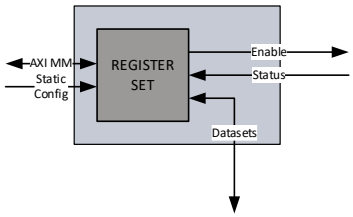

Figure 27: Registerset

4.2.13.2 Entity Description

Register Set

This module is an AXI4Lite Memory Mapped Slave. It provides access to all Datasets and allows configuring the PTP Ordinary Clock. AXI4Lite only supports 32 bit wide data access, no byte enables, no burst, no simultaneous read and writes and no unaligned access. It can be configured to either run in AXI or StaticConfig mode. If in StaticConfig mode, the configuration of the Datasets is done via signals and can be easily done from within the FPGA without CPU. For each parameter a valid signal is available, the enable signal shall be set last (or simultaneously). To change parameters the clock has to be disabled and enabled again. If in AXI mode, a AXI Master has to configure the Datasets with AXI writes to the registers, which is typically done by a CPU. Parameters can in this case also be changed at runtime.

4.2.13.3 Entity Declaration

Name	Dir	Type	Size	Description
Generics				
General				
SlaveOnly_Gen	-	boolean	1	Slave Only Clock
MasterOnly_Gen	-	boolean	1	Master Only Clock
DefaultProfile Support_Gen	-	boolean	1	Support for Default Profile
PowerProfile Support_Gen	-	boolean	1	Support for Power Profile
UtilityProfile Support_Gen	-	boolean	1	Support for Utility Profile

TsnProfile Support_Gen	-	boolean	1	Support for TSN Profile
ItuG82651Profile Support_Gen	-	boolean	1	Support for ITU G8265.1 Profile
ItuG82751Profile Support_Gen	-	boolean	1	Support for ITU G8275.1 Profile
ItuG82752Profile Support_Gen	-	boolean	1	Support for ITU G8275.2 Profile
NrOfUnicast Entries_Gen	-	natural	1	Number of Unicast Entries
E2eSupport_Gen	-	boolean	1	If the core shall support E2E delay mechanism (only valid with default profile support)
P2pSupport_Gen	-	boolean	1	If the core shall support P2P delay mechanism (mandatory for Power and Utility profile)
E2eUnicastSupport_Gen	-	boolean	1	E2E Unicast handling support
Layer2Support_Gen	-	boolean	1	Support for Layer 2 Mapping
Layer3v4 Support_Gen	-	boolean	1	Support for Layer 3 Ipv4 Mapping
Layer3v6 Support_Gen	-	boolean	1	Support for Layer 3 Ipv6 Mapping
TwoStepSupport_Gen	-	boolean	1	Support for TwoStep frames
Signaling Support_Gen	-	boolean	1	Support for Signaling frames
LuckyPacketFilter Support_Gen	-	boolean	1	If the Lucky Packet Filter shall be supported
LuckyPacketFilter Samples_Gen	-	natural	1	Maximum number of Samples in Lucky Packet Filter Win-

				dow
Asymmetry Support_Gen	-	boolean	1	Support for Asymmetry corrections
Register Set				
StaticConfig_Gen	-	boolean	1	If Static Configuration or AXI is used
AxiAddressRangeLow_Gen	-	std_logic_vector	32	AXI Base Address
AxiAddressRangeHigh_Gen	-	std_logic_vector	32	AXI Base Address plus Registerset Size
Ports				
System				
SysClk_ClkIn	in	std_logic	1	System Clock
SysRstN_RstIn	in	std_logic	1	System Reset
Config				
StaticConfig_DatIn	in	Ptp_OrdinaryClockStaticConfig_Type	1	Static Configuration
StaticConfig_ValIn	in	Ptp_OrdinaryClockStaticConfigVal_Type	1	Static Configuration valid
Status				
StaticStatus_DatOut	out	Ptp_OrdinaryClockStaticStatus_Type	1	Static Status
StaticStatus_ValOut	out	Ptp_OrdinaryClockStaticStatusVal_Type	1	Static Status valid
Error Input				
Ptp_ErrIn	in	std_logic	1	An error happened
AXI4 Lite Slave				
AxiWriteAddrValid_ValIn	in	std_logic	1	Write Address Valid
AxiWriteAddrReady_RdyOut	out	std_logic	1	Write Address Ready
AxiWriteAddrAddress_AdrIn	in	std_logic_vector	32	Write Address
AxiWriteAddrProt_DatIn	in	std_logic_vector	3	Write Address Protocol
AxiWriteDataValid_ValIn	in	std_logic	1	Write Data Valid
AxiWriteDataReady_RdyOut	out	std_logic	1	Write Data Ready

AxiWriteDataData_DatIn	in	std_logic_vector	32	Write Data
AxiWriteDataStrobe_DatIn	in	std_logic_vector	4	Write Data Strobe
AxiWriteRespValid_ValOut	out	std_logic	1	Write Response Valid
AxiWriteRespReady_RdyIn	in	std_logic	1	Write Response Ready
AxiWriteResp Response_DatOut	out	std_logic_vector	2	Write Response
AxiReadAddrValid_ValIn	in	std_logic	1	Read Address Valid
AxiReadAddrReady_RdyOut	out	std_logic	1	Read Address Ready
AxiReadAddrAddress_AdrIn	in	std_logic_vector	32	Read Address
AxiReadAddrProt_DatIn	in	std_logic_vector	3	Read Address Protocol
AxiReadDataValid_ValOut	out	std_logic	1	Read Data Valid
AxiReadDataReady_RdyIn	in	std_logic	1	Read Data Ready
AxiReadData Response_DatOut	out	std_logic_vector	2	Read Data
AxiReadDataData_DatOut	out	std_logic_vector	32	Read Data Re-sponse
Dataset Input				
CurrentDataset_DatIn	in	Ptp_Current Dataset_Type	1	PTP Current Dataset input for this clock
ParentDataset_DatIn	in	Ptp_Parent Dataset_Type	1	PTP Parent Dataset input for this clock
ForeignMasterDataset_DatIn	in	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset input for this clock
TimeProperties Dataset_DatIn	in	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset input for this clock
PortDataset_DatIn	in	Ptp_Port Dataset_Type	1	PTP Port Dataset input for this clock
DefaultDataset_DatIn	in	Ptp_Default Dataset_Type	1	PTP Default Dataset input for this clock
UnicastDataset Resp_DatIn	in	Ptp_UnicastDataset Resp_Type	1	PTP Unicast Dataset response for this

				clock
UnicastDataset Resp_ValIn	in	std_logic	1	PTP Unicast Dataset response for this clock valid
Dataset Output				
PortDataset _DatOut	out	Ptp_Port Dataset_Type	1	PTP Port Dataset data output
PortDataset _ValOut	out	Ptp_Port DatasetVal_Type	1	PTP Port Dataset valid output
CurrentDataset _DatOut	out	Ptp_Current Dataset_Type	1	PTP Current Dataset data output
CurrentDataset _ValOut	out	Ptp_Current DatasetVal_Type	1	PTP Current Dataset valid output
DefaultDataset _DatOut	out	Ptp_Default Dataset_Type	1	PTP Default Dataset data output
DefaultDataset _ValOut	out	Ptp_Default DatasetVal_Type	1	PTP Default Dataset valid output
ParentDataset _DatOut	out	Ptp_Parent Dataset_Type	1	PTP Parent Dataset data output
ParentDataset _ValOut	out	Ptp_Parent DatasetVal_Type	1	PTP Parent Dataset valid output
ForeignMasterDataset _DatOut	out	Ptp_ForeignMaster Dataset_Type	1	PTP Foreign Master Dataset data output
ForeignMasterDataset _ValOut	out	Ptp_ForeignMaster DatasetVal_Type	1	PTP Foreign Master Dataset valid output
TimeProperties Dataset_DatOut	out	Ptp_TimeProperties Dataset_Type	1	PTP Time Properties Dataset data output
TimeProperties Dataset_ValOut	out	Ptp_TimeProperties DatasetVal_Type	1	PTP Time Properties Dataset valid output
UnicastDataset Req_DatOut	out	Ptp_UnicastDataset Req_Type	1	PTP Unicast Dataset request data output
UnicastDataset Req_ValOut	out	std_logic	1	PTP Unicast Dataset request valid output
Enable Output				
PtpOrdinaryClock Enable_DatOut	out	std_logic	1	Enable PTP Ordinary Clock

Table 25: Registerset

4.3 Configuration example

4.3.1 Static Configuration

```
constant PtpStaticConfigOc_Con : Ptp_OrdinaryClockStaticConfig_Type := (
    Profile                                         => DefaultProfile_E,
    Layer                                           => Layer2_E,
    DelayMechanism                                  => P2p_E,
    DelayE2eUnicast                                 => '0',
    TwoStep                                         => '0',
    Signaling                                       => '0',
    Vlan                                            => Ptp_Vlan_Type_Rst_Con,
    VlanEnable                                      => '1',
    Ip                                              => (
        0                                           => x"C0",
        1                                           => x"A8",
        2                                           => x"00",
        3                                           => x"10",
        others                                      => x"00"),
    DefaultDataset_ClockIdentity                    => (
        0                                           => x"4E",
        1                                           => x"54",
        2                                           => x"4C",
        3                                           => x"FF",
        4                                           => x"FE",
        5                                           => x"00",
        6                                           => x"00",
        7                                           => x"10"),
    DefaultDataset_DomainNumber                     => x"00",
    DefaultDataset_ClockQuality                     => Ptp_ClockQuality_Type_Rst_Con,
    DefaultDataset_Priority1                        => x"80",
    DefaultDataset_Priority2                        => x"80",
    DefaultDataset_GrandmasterId                    => x"0003",
    DefaultDataset_GrandmasterTimeInaccuracy        => x"00000032",
    TimePropertiesDataset_CurrentUtcOffset          => x"0019",
    TimePropertiesDataset_CurrentUtcOffsetValid     => '0',
    TimePropertiesDataset_Leap59                    => '0',
    TimePropertiesDataset_Leap61                    => '0',
    TimePropertiesDataset_TimeTraceable             => '0',
    TimePropertiesDataset_FrequencyTraceable        => '0',
    TimePropertiesDataset_PtpTimescale              => '1',
    TimePropertiesDataset_TimeSource                => x"A0",
    TimePropertiesDataset_CurrentOffset             => (others => '0'),
    TimePropertiesDataset_JumpSeconds               => (others => '0'),
    TimePropertiesDataset_TimeOfNextJumpSeconds     => (others => '0'),
    TimePropertiesDataset_DisplayNameLength         => x"03",
    TimePropertiesDataset_DisplayName => (
        0                               => Common_CharacterToStdLogic_Func('P'),
        1                               => Common_CharacterToStdLogic_Func('T'),
```

```
        2                               => Common_CharacterToStdLogic_Func('P'),
        others                          => (others => '0'))
    );

    constant PtpStaticConfigValOc_Con : Ptp_OrdinaryClockStaticConfigVal_Type := (
        Enable_Val                                  => '1',
        Profile_Val                                 => '1',
        Vlan_Val                                    => '1',
        Ip_Val                                      => '1',
        DefaultDataset_ClockIdentity_Val            => '1',
        DefaultDataset_DomainNumber_Val             => '1',
        DefaultDataset_ClockQuality_Val             => '1',
        DefaultDataset_Priority1_Val                => '1',
        DefaultDataset_Priority2_Val                => '1',
        DefaultDataset_GrandmasterId_Val            => '1',
        DefaultDataset_GrandmasterTimeInaccuracy_Val    => '1',
        TimePropertiesDataset_CurrentUtcOffset_Val      => '1',
        TimePropertiesDataset_CurrentUtcOffsetValid_Val => '1',
        TimePropertiesDataset_Leap59_Val            => '1',
        TimePropertiesDataset_Leap61_Val            => '1',
        TimePropertiesDataset_TimeTraceable_Val         => '1',
        TimePropertiesDataset_FrequencyTraceable_Val    => '1',
        TimePropertiesDataset_PtpTimescale_Val          => '1',
        TimePropertiesDataset_TimeSource_Val            => '1',
        TimePropertiesDataset_CurrentOffset_Val         => '1',
        TimePropertiesDataset_JumpSeconds_Val           => '1',
        TimePropertiesDataset_TimeOfNextJumpSeconds_Val => '1',
        TimePropertiesDataset_DisplayNameLength_Val     => '1',
        TimePropertiesDataset_DisplayName_Val           => '1'
    );
```

Figure 28: Static Configuration

4.3.2 AXI Configuration

The following code is a simplified pseudocode from the testbench: The base address of the Ordinary Clock is 0x10000000.

```
-- PTP OC
-- Config
-- profile: 0 default profile
AXI WRITE 10000084 00000000
-- VLAN 0x4000 (unused)
AXI WRITE 10000088 00014000
-- set config valid bits
AXI WRITE 10000080 00000003

-- Default Dataset
-- clock id: 00:01:02:FF:FE:03:04:05
AXI WRITE 10000104 FF020100
```

```
AXI WRITE 10000108 050403FE
-- domain: 0
AXI WRITE 1000010C 00000000
-- clock class: BB, clock accuracy FE, priority1 80, Priority 80
AXI WRITE 10000110 BBFE8080
-- grandmaster id: 0003 (unused)
AXI WRITE 10000114 00000003
-- grandmaster inaccuracy: 00000028 (unused)
-- AXI WRITE 10000118 00000028
-- set default dataset valid bits
AXI WRITE 10000100 0000007F

-- TimeProperties Dataset
-- current utc offset & "00" & current utc offset valid &
-- leap59 & leap61 & time traceable & frequency traceable & ptp timescale &
-- time source
AXI WRITE 10000504 001901A0
-- current offset: 0
AXI WRITE 10000508 00000000
-- jumpseconds: 0
AXI WRITE 1000050C 00000000
-- time of next jump seconds(31 downto 0): 0
AXI WRITE 10000510 00000000
-- x"0000" & time of next jump seconds(47 downto 32): 0
AXI WRITE 10000514 00000000
-- x"000000" & display name length
AXI WRITE 10000518 00000003
-- display name(3 downto 0): NTL
AXI WRITE 1000051C 004C544E
-- display name(7 downto 4): -
AXI WRITE 10000520 00000000
-- display name(11 downto 8): -
AXI WRITE 10000524 00000000
-- set timeproperties dataset valid bits
AXI WRITE 10000500 00001FFF

-- enable PTP OC
AXI WRITE 10000000 00000001
```

Figure 29: AXI Configuration

4.4 Clocking and Reset Concept

4.4.1 Clocking

To keep the design as robust and simple as possible, the whole Ordinary Clock, including the Counter Clock and all other cores from NetTimeLogic are run in one clock domain. This is considered to be the system clock. Per Default this clock is 50MHz. Where possible also the interfaces are run synchronous to this clock. For clock domain crossing asynchronous Fifos with gray counters or message patterns with meta-stability flip-flops are used. Clock domain crossings for the AXI interface is moved from the AXI slave to the AXI interconnect.

Clock	Frequency	Description
System		
System Clock	50MHz (Default)	System clock where the OC runs on as well as the counter clock etc.
(R)(G)MII Interface		
PHY (R)(G)MII RX Clock	2.5/25/125MHz	Asynchronous, external receive clock from the PHY also used for the MAC. Depending on the interface not all frequencies apply.
PHY (R)(G)MII TX Clock	2.5/25/125MHz	Asynchronous, external transmit clock to/from the PHY also used for the MAC. Depending on the interface not all frequencies apply.
AXI Interface		
AXI Clock	50MHz (Default)	Internal AXI bus clock, same as the system clock

Table 26: Clocks

4.4.2 Reset

In connection with the clocks, there is a reset signal for each clock domain. All resets are active low. All resets can be asynchronously set and shall be synchronously released with the corresponding clock domain. All resets shall be asserted for the first couple (around 8) clock cycles. All resets shall be set simultaneously and released simultaneously to avoid overflow conditions in the core. See the reference designs top file for an example of how the reset shall be handled.

Reset	Polarity	Description
System		
System Reset	Active low	Asynchronous set, synchronous release with the system clock
(R)(G)MII Interface		
PHY (R)(G)MII RX Reset	Active low	Asynchronous set, synchronous release with the (R)(G)MII RX clock
PHY (R)(G)MII TX Reset	Active low	Asynchronous set, synchronous release with the (R)(G)MII TX clock
AXI Interface		
AXI Reset	Active low	Asynchronous set, synchronous release with the AXI clock, which is the same as the system clock

Table 27: Resets

5 Resource Usage

Since the FPGA Architecture between vendors and FPGA families differ there is a split up into the two major FPGA vendors.

5.1 Intel/Altera (Cyclone V)

Configuration	FFs	LUTs	BRAMs	DSPs
Minimal (SlaveOnly, Only Default Profile, L2, No Management)	9336	20026	18	0
Maximal (All Profiles, L2 & L4, Management)	14292	26769	18	0

Table 28: Resource Usage Intel/Altera

5.2 AMD/Xilinx (Artix 7)

Configuration	FFs	LUTs	BRAMs	DSPs
Minimal (SlaveOnly, Only Default Profile, L2, No Management)	8131	16041	5	0
Maximal (All Profiles, L2 & L4, Management)	14208	28552	5	0

Table 29: Resource Usage AMD/Xilinx

6 Delivery Structure

```
AXI                                 -- AXI library folder
 |-Library                          -- AXI library component sources
 |-Package                          -- AXI library package sources

CLK                                 -- CLK library folder
 |-Library                          -- CLK library component sources
 |-Package                          -- CLK library package sources

COMMON                              -- COMMON library folder
 |-Library                          -- COMMON library component sources
 |-Package                          -- COMMON library package sources

PPS                                 -- PPS library folder
|-Package                           -- PPS library package sources

PTP                                 -- PTP library folder
|-Core                              -- PTP library cores
 |-Doc                              -- PTP library cores documentations
 |-Library                          -- PTP library component sources
 |-Package                          -- PTP library package sources
 |-Refdesign                        -- PTP library cores reference designs
 |-Testbench                        -- PTP library cores testbench sources and sim/log

SIM                                 -- SIM library folder
 |-Doc                              -- SIM library command documentation
 |-Package                          -- SIM library package sources
 |-Testbench                        -- SIM library testbench template sources
 |-Tools                            -- SIM simulation tools
```

7 Testbench

The Ptp Ordinary Clock testbench consist of 4 parse/port types: AXI, CLK, ETH and PTP. Multiple instances exist. PTP0 CLK, PTP1 CLK, PTP0 PTP, PTP1 PTP and MAC0 ETH ports are all multiplexed with the MAC0 ETH MUX to one Ethernet channel connected to the port going to the PHY from the DUT (which acts like a MAC). PHY0 is connected to the port going to the MAC from the DUT (which acts like a PHY)

The PTP Master ports take the CLK ports times as reference and sets the timestamps aligned with the time from the CLK ports. In Master mode the send Announce and Sync messages in one and two-step mode. The PTP Slave ports take the time of the Clock instance as reference and checks it with the times in the frames from the DUT. Once the clock is synchronized (when the TB is acting as Master) the CLK port and Clock generated time should be phase and frequency aligned to one of the PTP ports. The PTP ports are answering PDelay Request messages and also measure the Delay themselves with their own PDelay Request messages. In addition for configuration and result checks an AXI read and write port is used in the testbench and for accessing more than one AXI slave also an AXI interconnect is required.

With this Setup Master and Slave scenarios as well as multiple PTP nodes can be simulated with different PTP Profiles.

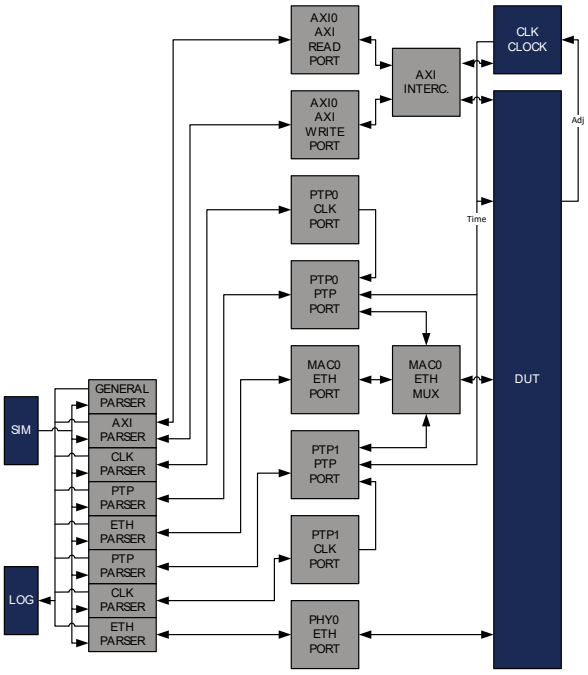

Figure 30: Testbench Framework

For more information on the testbench framework check the Sim_ReferenceManual documentation.

With the Sim parameter set the time base for timeouts are divided by 1000 to 100000 to speed up simulation time.

7.1 Run Testbench

1. Run the general script first

```
source XXX/SIM/Tools/source_with_args.tcl
```

2. Start the testbench with all test cases

```
src XXX/PTP/Testbench/Core/PtpOrdinaryClock/Script/run_Ptp_OrdinaryClock_Tb.tcl
```

3. Check the log files LogFileX.txt in the XXX/PTP/Testbench/Core/PtpOrdinaryClock/Log/ folder for simulation results.

8 Reference Designs

The PTP Ordinary Clock reference design contains a PLL to generate all necessary clocks (cores are run at 50 MHz), an instance of the PTP Ordinary Clock IP core and an instance of the Adjustable Counter Clock IP core (needs to be purchased separately). The Reference Design is intended to be connected to any PTP Master or Slave device (The AMD/Xilinx SlaveOnly Clock only supports PTP Masters) which can run in Layer 2, P2P mode either in Default or Power Profile. The Reference Design is using MII in 100Mbit full duplex as Ethernet link.
All generics can be adapted to the specific needs.

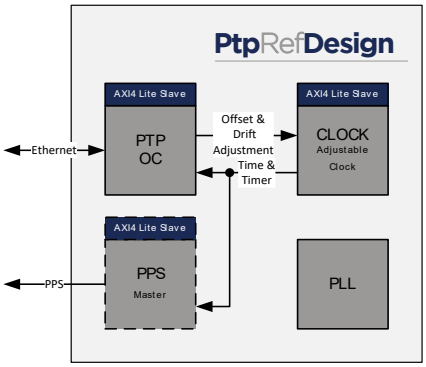

Figure 31: Reference Design

8.1 Intel/Altera: Terasic SocKit

The SocKit board is an FPGA board from Terasic Inc. with a Cyclone V SoC FPGA from Intel/Altera. (http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=816) and an Ethernet adapter in HSMC form factor HSMC-NET also from Terasic: (http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=71&No=355)

1. Open Quartus 16.x
2. Open Project /PTP/Refdesign/Altera/SocKit/PtpOrdinaryClockMii/PtpOrdinaryClock.qpf
3. If the optional core PPS Master Clock is available add the files from the corresponding folders (PPS/Core, PPS/Library and PPS/Package)
4. Change the generics (PpsMasterAvailable_Gen) in Quartus (in the settings menu, not in VHDL) to true for the optional cores that are available.
5. Rerun implementation

6. Download to FPGA via JTAG

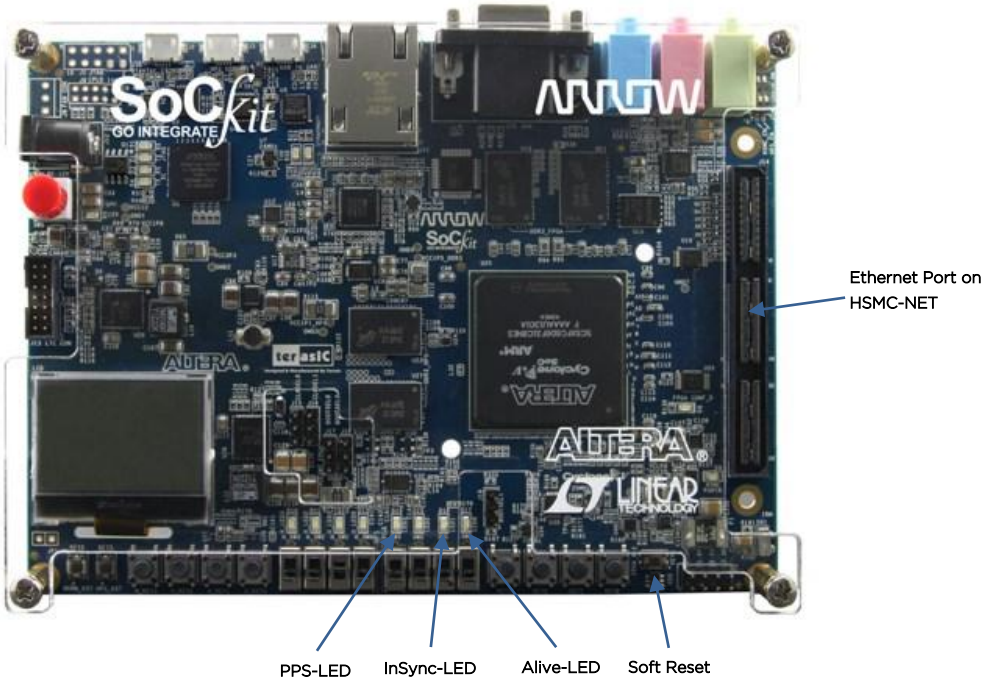

Figure 32: SocKit (source Terasic Inc)

For the ports on the HSMC connector the Ethernet to HSMC adapter from Terasic Inc. was used.

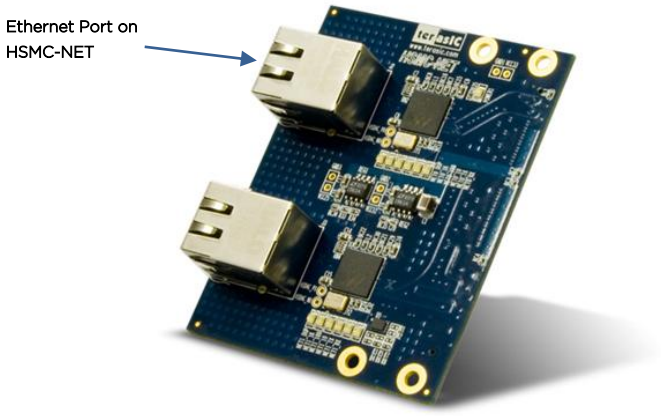

Figure 33: HSMC-NET (source Terasic Inc)

8.2 AMD/Xilinx: Digilent Arty

The Arty board is an FPGA board from Digilent Inc. with an Artix7 FPGA from AMD/Xilinx. (http://store.digilentinc.com/arty-board-artix-7-fpga-development-board-for-makers-and-hobbyists/)

1. Open Vivado 2019.1.
   Note: If a different Vivado version is used, see chapter 8.3.
2. Run TCL script
   /PTP/Refdesign/Xilinx/Arty/PtpSlaveOnlyClockMii/PtpSlaveOnlyClock.tcl
   a. This has to be run only the first time and will create a new Vivado Project
3. If the project has been created before open the project and do not rerun the project TCL
4. If the optional core PPS Master Clock is available add the files from the corresponding folders (PPS/Core, PPS/Library and PPS/Package) to the corresponding Library (PpsLib).
5. Change the generics (PpsMasterAvailable_Gen) in Vivado (in the settings menu, not in VHDL) to true for the optional cores that are available.
6. Rerun implementation
7. Download to FPGA via JTAG

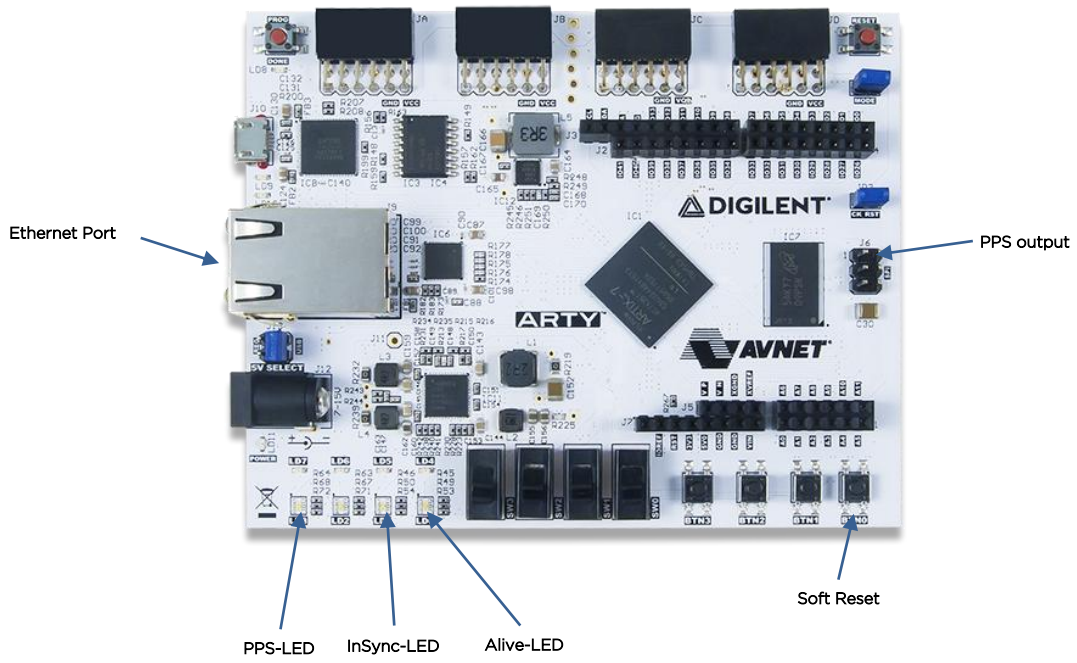

Figure 34: Arty (source Digilent Inc)

8.3 AMD/Xilinx: Vivado version

The provided TCL script for creation of the reference-design project is targeting AMD/Xilinx Vivado 2019.1.
If a lower Vivado version is used, it is recommended to upgrade to Vivado 2019.1 or higher.
If a higher Vivado version is used, the following steps are recommended:

- Before executing the project creation TCL script, the script's references of Vivado 2019 should be manually replaced to the current Vivado version. For example, if version Vivado 2022 is used, then:
  - The statement occurrences:
    ```
    set_property flow "Vivado Synthesis 2019" $obj
    ```
    shall be replaced by:
    ```
    set_property flow "Vivado Synthesis 2022 $obj
    ```
  - The statement occurrences:
    ```
    set_property flow "Vivado Implementation 2019" $obj
    ```
    shall be replaced by:
    ```
    set_property flow "Vivado Implementation 2022" $obj
    ```
- After executing the project creation TCL script, the AMD/Xilinx IP cores, such as the Clocking Wizard core, might be locked and a version upgrade might be required. To do so:
  1. At "Reports" menu, select "Report IP Status".
  2. At the opened "IP Status" window, select "Upgrade Selected". The tool will upgrade the version of the selected IP cores.

A List of tables

Table 1: Revision History 4
Table 2: Definitions 8
Table 3: Abbreviations 9
Table 4: Register Set Overview 32
Table 5: Parameters 115
Table 6: Clk_Time_Type 115
Table 7: Clk_TimeAdjustment_Type 116
Table 8: Ptp_OrdinaryClockStaticConfig_Type 118
Table 9: Ptp_OrdinaryClockStaticConfigVal_Type 120
Table 10: Ptp_OrdinaryClockStaticStatus_Type 124
Table 11: Ptp_OrdinaryClockStaticStatusVal_Type 124
Table 12: PTP Ordinary Clock 136
Table 13: RX Processor 142
Table 14: TX Processor 148
Table 15: Announce Processor 154
Table 16: Delay Processor 161
Table 17: Sync Processor 168
Table 18: Management Processor 173
Table 19: Signaling Processor 179
Table 20: MAC Processor 182
Table 21: BMC Algorithm 188
Table 22: Datasets 195
Table 23: Clock Counter 197
Table 24: Ethernet Interface Adapter 200
Table 25: Registerset 205
Table 26: Clocks 209
Table 27: Resets 210
Table 28: Resource Usage Intel/Altera 211
Table 29: Resource Usage AMD/Xilinx 211

B List of figures

Figure 1: Context Block Diagram 10
Figure 2: Architecture Block Diagram 11
Figure 3: Simple setup 15
Figure 4: Message exchange simple setup 17

Figure 5: PTP network with Boundary Clock 19
Figure 6: PTP network with Transparent Clock 20
Figure 7: E2E Delay measurement with BC and TC 22
Figure 8: E2E Delay unicast measurement with TC 23
Figure 9: P2P Delay measurement with BC and TC 24
Figure 10: Unicast Scheme 25
Figure 11: P2P Delay measurement with BC and TC 26
Figure 12: Timestamp Incauracy in the different Layers 28
Figure 13: PTP Ordinary Clock 125
Figure 14: RX Processor 137
Figure 15: TX Processor 143
Figure 16: Announce Processor 149
Figure 17: Delay Processor 155
Figure 18: Sync Processor 162
Figure 19: Management Processor 169
Figure 20: Signaling Processor 175
Figure 21: Mac Processor 180
Figure 22: BMC Algorithm 183
Figure 23: Statemachine (source IEEE1588-2019/2008 Std) 184
Figure 24: Datasets 189
Figure 25: Clock Counter 196
Figure 26: Ethernet Interface Adapter 198
Figure 27: Registerset 201
Figure 28: Static Configuration 207
Figure 29: AXI Configuration 208
Figure 30: Testbench Framework 213
Figure 31: Reference Design 215
Figure 32: SocKit (source Terasic Inc) 216
Figure 33: HSMC-NET (source Terasic Inc) 216
Figure 34: Arty (source Digilent Inc) 218