RapidIO Trade Association

Suite 325, 3925 W. Braker Lane

Austin, TX 78759

512-305-0070 Tel.

512-305-0009 FAX.

TWG Second Showing

Item 11-05-00000.006

Subject: Time Distribution Specification

Background: Other packet based interconnects support a low jitter method for distribution of a real time clock within the network. The ability to synchronize geographically distributed elements is important for many applications which use RapidIO. To remain competitive, RapidIO fabrics and endpoints must support the ability to be tightly synchronized in time.

Contributor: Barry Wood, IDT

Comment Expiration Date:

Distribution: RapidIO TA Technical Working Group members

1.0 Introduction

Each endpoint of a RapidIO network which supports real time behavior must have some sense of time. In this discussion, the "sense of time" is known as a timestamp generator (TSG).

The ability to synchronize timestamp generators located across a RapidIO network requires two capabilities:

1. The ability to synchronize the absolute time on each endpoint connected to the fabric.
2. The ability to ensure that the timestamp generator on each endpoint can remain in sync with the other timestamp generators.

The "most challenging" requirements for distribution of time within a RapidIO network are taken to be similar to that of IEEE 1588 in networking applications, where customers want to keep nodes synchronized within 1 microsecond across 30 network hops. This leads to extremely stringent requirements:

1. Nanosecond granularity for time distribution
2. Uncertainty for time updates of 25 nanoseconds or less per hop

Given the advantages of RapidIO technology, companies should be able to deliver devices which meet these requirements.

The intent of this showing is to define the mechanisms which will enable maximal accuracy, while providing implementers multiple options to meet the needs of their customers and remain interoperable with other devices.

1.1 Common Fixed Frequency Distribution

One way to ensure that TSGs on endpoints remain synchronized to each other is to distribute a common clock signal. RapidIO currently supports distribution of a common clock signal using the Multicast Event Control Symbol (MECS). A MECS can be transmitted periodically to indicate that an amount of time has passed. This is simple, but has some limitations if extreme accuracy is required. MECS is subject to jitter induced by protocol, as well as at each clock crossing boundary in a design. While it is possible to minimize the jitter induced by protocol, it is difficult to remove all of the possible jitter. It is therefore extremely difficult to meet the proposed requirements through the use of MECS.

In many other protocols (i.e. T1/E1, CPRI, OBSAI), a common reference frequency is distributed as the framing reference for data transmission. In others, such as Synchronous Ethernet, a common reference frequency is distributed as the transmission clock for the protocol. The received frequency

is recovered from the received transmission and passed to a component which attenuates jitter. The resulting clock signal can then be used as the transmit reference clock for the next hop. Both are solutions for distributing a highly accurate network clock source as the basis for clocking throughout the network. Usually, there is a redundancy scheme which allows the network clock source to fail over to a different network clock source.

It is expected that the easiest method for distributing a common reference frequency is as the clock signal embedded within a RapidIO link. When link partners have a common reference frequency for the timestamp generators, the timestamp generators are expected to advance at the same rate. It is possible that the timestamp generators would advance with different granularity. For example, assume a common frequency is distributed over a 3.125 Gbaud link. One TSG could advance every 3.2 nsec, while the other could advance every 6.4 nsec.

The advantages of using a common reference frequency may include removal of clock boundary crossings between receiver and transmitter, consistent timing measurements in both receive and transmit directions, ability to "set and forget" the link partners TSG, and the ability to simplify the link protocol by removing the clock compensation sequence. These advantages are only realized when the reference frequency is used for the link receiver and transmitter, as well as the source for the TSG.

The disadvantages of distributing a common reference clock frequency may include additional pins for clock signals, and more complexity in clocking design for components and boards.

RapidIO currently does not standardize a scheme for the distribution of a common reference frequency. According to the RapidIO protocol, the clock frequency of each RapidIO processing element is allowed to vary by +/- 100 ppm from the specified frequency. The frequency can wander, as long as it does not exceed +/- 100 ppm from the specified base frequency.

Distribution of a fixed frequency within a RapidIO network is further complicated by the different lane speeds allowed by RapidIO. Allowed lane speeds of 1.25, 2.5, 3.125, 5.0, 6.25 and 10.3125 will be supported by the Gen3 protocol. A common base frequency of 156.25 MHz can be used to generate these bit rates. However, the framing frequencies are 125, 250, 312.5, 500, 625 and 153.9179 MHz. It is easy to come to the conclusion that it is not possible to distribute a common clock frequency for use as a transmission clock when a network must support the combination of Gen3 and either Gen 1 or Gen 2 lane speeds. However, "universal" clocking technology exists which allows translation from any input frequency to any output frequency. It should be possible to distribute a common clock

frequency throughout a RapidIO network which uses Gen1, Gen2 and Gen3 lane speeds.

The details of the distribution of a common reference clock frequency are not relevant to the interoperability of time synchronization schemes. The time distribution proposal allows for, but does not require, the distribution of a common reference frequency throughout the RapidIO network.

In the absence of the distribution of a common clock frequency, it is possible to calibrate the difference in frequency between the transmit clocks of each end of a link. This information can be used to determine a correction factor to synchronize absolute time between endpoints. This alternative is presented as a potential lower cost alternative for systems which do not require highly accurate clocking.

1.2 Synchronization of Absolute Time

Synchronizing two endpoints X and Y to within nanoseconds of each other requires that the transmission delay between X and Y be calibrated. This is typically performed by having X send a packet to Y, and noting the time the packet was sent. Y returns the packet to X. X knows the time the packet was sent, and the time the response was received, and so can determine the latency of the transmission delay to some level of accuracy based on X's clock. The fewer factors which can affect the latency and jitter of timestamping, the more closely the two endpoints can be synchronized.

For example, IEEE 1588 Precision Time Protocol (PTP) uses packets to compute the transmission delay on a lane. The timestamp is placed in the packet as close to the SerDes pins as possible in order to limit variability in transmission delay. This timestamping capability is used to determine the transmission loop delay on a link, which is used in turn to provide a a time offset for synchronization purposes. IEEE 1588 supports timestamps which are 80 bits in size, composed of a 32 bit nanosecond timestamp and a 48 bit seconds timestamp.

IEEE 1588 supports many different approaches to synchronizing nodes in a network. This proposal includes support for synchronization across a network through the use of maintenance packets, and synchronization on a hop-by-hop basis using a combination of maintenance packets and control symbols.

RapidIO has some advantages when implementing a time synchronization protocol, in that timestamps could be exchanged using control symbols. Control symbols are simpler to compose and send at a predictable time than packets.

One issue which all synchronization mechanisms must take into account is the precision of the timestamp which can be tracked. The precision of a timestamp is the clock period used to increment the time. Typically, the higher the reference clock frequency, the more precisely a timestamp can be tracked. In theory, timestamps could be tracked at the bit frequency of the link. Practically speaking, for Gen 1 and Gen 2 links the rate of 10 bit codes would determine the most precise timestamps. Similarly, on Gen 3 links most precise timestamps occur at the 67 bit code group rate.

The remainder of this discussion assumes that the maximum clocking rate for the real time clock associated with a link is the 10 bit or 67 bit code group rate. The implication is that there is at least one clock period of uncertainty when exchanging timestamps on the link, because it is unlikely that the amount of delay on the link precisely matches the clock period with which timestamps are tracked. The amount of uncertainty may exceed one clock period, as the clock period is not always a whole number of nanoseconds.

1.3 Jitter Induced by Protocol

Several operations are required to synchronize and maintain synchronization of timestamp generators. These operations can be implemented using either packets or control symbols. Either approach requires that protocol induced jitter must be reduced in order to improve the accuracy of the synchronization. The sources of jitter, and a discussion on how to deal with this jitter, are captured in the table below. This table is intended to justify the design choices made for the proposed implementation.

Table 1-1. Protocol Induced Sources of Jitter

Source	Description	Discussion
Clock Compensation Sequence	Clock compensation sequence (CCS) must be sent every 5000 code groups on each lane. A CCS cannot be interrupted once transmission has started. A CCS will induce jitter for both packets and control symbols. Clock compensation sequences also induce jitter because characters can be dropped, which means that the delay between transmission and reception of a character can vary.	The reason a CCS induces jitter is that it is not possible to interrupt it with a packet or control symbol. A CCS therefore can result in a multiple code group delay when propagating a control symbol or packet. To avoid CCS jitter, it is possible to delay propagation of a control symbol or packet by a fixed amount of time to ensure that a CCS completes transmission. In the event that link partners have a common reference clock, it may be better to disable transmission of CCS to eliminate jitter.
Status Control Symbols	A control symbol which communicates the buffer status of a port must be sent once every 1024 code groups on each lane. If such a control symbol has not been transmitted in 1024 code groups, then a status control symbol must be transmitted, inducing jitter on the link for both packets and control symbols.	Status control symbols should not induce jitter as they can be combined with the control symbol used to delimit a packet, or combined with the control symbols used to implement time synchronization functions. Again, it should be possible to avoid jitter induced on control symbol transmission by delaying transmission of a control symbol by a fixed amount of time to allow a Status Control Symbol to be sent if necessary.
VC Status Control Symbols	A control symbol which communicates the VC buffer status of a port must be sent within a configurable interval between 1024 code groups and 256K code groups on each lane.	VC Status control symbols should not induce jitter as they can be combined with the control symbol used to delimit a packet, or combined with the control symbols used to implement time synchronization functions. In the event that VC Status control symbols cannot be combined, jitter can be avoided by implementing a constant delay to ensure that VC status control symbols complete transmission before the time synchronization function packet/control symbol is transmitted.
CS Transmission Alignment Within Packets on a 1x Link	When a control symbol is embedded within a packet, it must be embedded at a 4 byte boundary. When a control symbols starts within an IDLE sequence, it can start immediately. Jitter is induced by the difference in timing between control symbol transmission in an IDLE sequence and embedding that control symbol within a packet. The jitter is only induced for control symbols.	Practically speaking, it should be possible to interrupt IDLE sequences at a multiple of four bytes, consistent with embedding within a packet.

Table 1-1. Protocol Induced Sources of Jitter

Source	Description	Discussion
CS Transmission Alignment For Multi-Lane Ports vs Single Lane Ports	When a control symbol or packet is transmitted on a multilane port, it is striped across multiple lanes. Single lane ports do not require striping. When a packet or control symbol (i.e. MECS) must be propagated from a multi-lane port to a single lane port, retransmission may need to wait for a 4 byte boundary on the single lane port. This behavior induces jitter on packets and control symbols.	Time synchronization can be performed on a hop-by-hop basis using control symbol based mechanisms to minimize jitter. This approach avoids the jitter induced by multi-lane to single-lane propagation.
Gen1/Gen2 to Gen3 Alignment	Gen1 and Gen2 links stripe packets and control symbols across lanes one byte at a time. Gen3 links stripe packets and control symbols across lanes in chunks of 8 bytes. Transfers from a Gen1 or Gen2 link must therefore wait for 8 bytes to accumulate for each lane on the Gen3 port. The amount of delay varies based on the number of bytes which must be accumulated. This will induce jitter on both control symbols and packets.	Time synchronization can be performed on a hop-by-hop basis using control symbol based mechanisms to minimize jitter. This approach avoids the jitter induced by Gen1/Gen2 to Gen3 propagation.
Gen3 Dynamic Lane Width Management	Transitions from multi-lane to single lane operation and back have windows of time where no data may be exchanged on the link.	It may be necessary to disable dynamic lane width management for links which must support time synchronization functions. Since time synchronization functions occur periodically, it should be possible to prevent lane widths from changing when time synchronization functions are active.
Impact of Input-Error Stopped/Output-Error Stopped States	Bit errors can be detected at any time. Recovery from bit errors induce jitter in time synchronization functions. Input-error stopped/output-error stopped states prevent the operation of time synchronization functions implemented using packets. Control symbols may continue to be exchanged when a link is in input-error stopped or output-error stopped state.	Bit errors can induce jitter at any time, whatever mechanism is used. Packet based mechanisms have higher jitter due to bit errors, due to packet retransmission delays in the input-error stopped/output-error stopped states.
Impact of Link Reinitialization	Link reinitialization can occur at any time, if the bit error rate exceeds a threshold.	Link reinitialization prevents execution of all time synchronization functions. It is important that the delay for link reinitialization not result in corruption of the real time counter of the link partner.

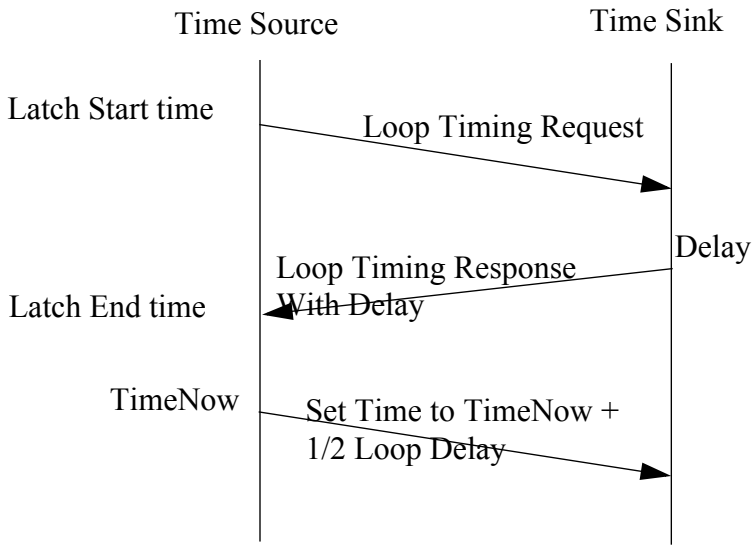
1.4 Time Synchronization Using a Common

Reference Clock

Figure 1-1 describes the algorithm, but not the specific mechanisms, for synchronizing time between two nodes. It makes use of the following terms:

- Loop Timing Request: A request which causes the time source to latch a “Start time” timestamp when the request is transmitted. The time sink sends a Loop Timing Response when it receives a Loop Timing Request.
- Loop Timing Response: The response to a loop timing request. It contains the amount of Delay. Reception of a loop timing response packet causes the time source to latch the “End Time” for when the response has been received
- Delay: Time within the time sink from the time a loop timing request has been received, to the time a loop timing response is transmitted.
- Set Time: A request for setting the time on a link partner.
- TimeNow, TimeNow1, TimeNow2, TimeNow3: The time at this point in the message sequence chart.
- Loop Delay: Time between sending a Loop Timing Request and receiving a Loop Timing Response, less the Delay in the time sink receiving the Loop Timing Request. Computed as (End Time - StartTime - Delay)

Figure 1-1. Time Synchronization with Synchronous Link Partners



The algorithm first determines the loop delay between the time source and time sink. The time source then sets the time sink's time to be offset from the time sources time by half of the loop delay.

The loop timing request can be implemented using maintenance requests across a RapidIO fabric, or using control symbols between link partners. This proposal supports both approaches. The use of control symbols is necessary to maximize timing accuracy. Note that the "Latch Start time" and "Latch End time" operations need not require any special circuitry, as software could just read the Time Source TSG.

An alternative approach could cause timestamps to be latched on the Time Sink when a Loop Timing Request is received and again when a Loop Timing Response is transmitted. This alternative approach is not presented in this proposal, as it requires more circuitry in the Time Sink compared to the approach chosen.

Note that the measurement of loop delay may require multiple trials to arrive at an average value, in order to account for jitter due to clock crossing boundaries in both the time source and time sink, and varying network delay.

Further note that the mechanism assumes that the delay from Time Source to Time Sink is the same as that from Time Sink to Time Source. This is unlikely to be the case. It is certain that there will be different delays in each direction, described hereafter as "asymmetry".

It is not possible for the Time Source to measure the amount of asymmetry on a link.

Board trace length is unlikely to be a measurable cause of asymmetry. The highest framing frequency for RapidIO is a Gen2 6.25 Gbaud link, which is 625 MHz. This translates to a clock period of 1.6 nsec, or a board asymmetry of approximately 9.6 inches. While it is possible to have one direction of a link be 9.6 inches longer than the reverse path, it is unlikely. It is reasonable for RapidIO silicon suppliers to specify the impact of board trace differences on timing accuracy, and for customers to optimize their design to the degree necessary to achieve their required synchronization accuracy.

Based on current designs, most link asymmetry will result from differences between the receive path and transmit path within a device. The difference are likely to be a constant multiple of the framing frequency for the link. This proposal therefore takes the approach of specifying the asymmetry through a register value, to allow the Time Source to account for the constant asymmetry.

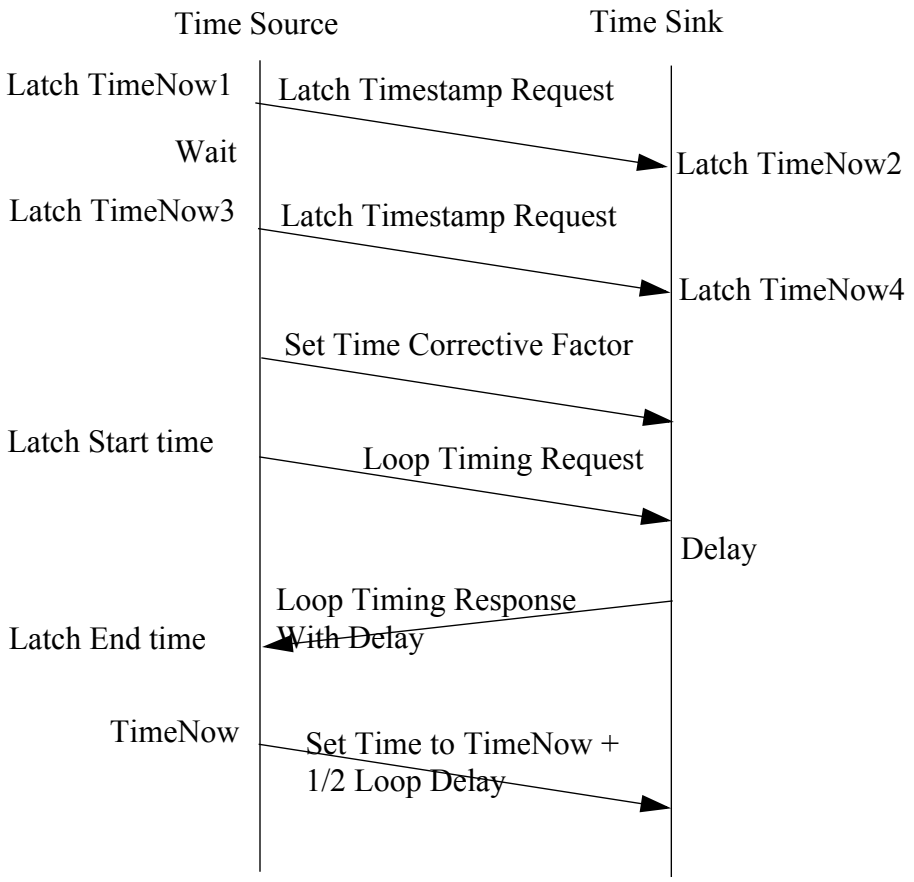
1.5 Link Partners With RapidIO Standard Clocking

Currently, RapidIO link partners are allowed to use reference clock frequencies which differ by +/- 100 PPM. To synchronize both the advancement of time as well as delay computations, operations are required to calibrate the difference in clock rates between the two link partners and then correct for these differences.

Figure 1-2 on page -13 describes the algorithm, but not the specific mechanisms, for synchronizing time between two nodes. It makes use of the following additional terms:

- ATimeNow1-4: Timestamps latched before the Time Corrective Factor is applied.
- Time Corrective Factor: Mechanism for adjusting the Time Sink's time stamp progression based on the measured difference between the Time Source and Time Sink's reference clock rate. An example of a Time Corrective Factor is to specify that every X ticks, add another Y ticks or skip Z ticks, where X, Y and Z are programmable.
- TimeNow, TimeNow1, TimeNow2, TimeNow3: The time at this point in the message sequence chart.

Figure 1-2. Time Synchronization with Asynchronous Link Partners



The algorithm is identical to that found in "Section 1.4, Time Synchronization Using a Common Reference Clock" on page 9, with the addition of computing and setting the Time Corrective Factor. To compute the Time Corrective Factor, the Time Source sends two timestamp requests separated by a fixed time. The fixed time is computed based on the Time Source clock. The Time Source then reads the time stamps captured in the Time Sink, and computes the Time Corrective Factor. The Time Corrective Factor is applied to the reference clock rate used by the Time Sink, to ensure that time advances at the same rate on the Time Sink and the Time Source. Once the Time Corrective Factor has been applied, the remainder of the time synchronization algorithm is identical to that found in "Section 1.4, Time Synchronization Using a Common Reference Clock" on page 9.

Measurement of a Time Corrective Factor needs to be done over a significant period of time (millions of clock cycles) in order to determine the deviation of the Sink clock rate from the Source clock rate. However, these

measurements can still take place faster than it is possible to read a timestamp latched in the link partner. For this reason, separate timestamps for the start and end are required.

Note that measurement of time corrective factor and loop delay may require multiple trials to come up with an average value, in order to account for jitter due to clock crossing boundaries in both the time source and time sink.

Also note that time corrective factors and loop delay may vary as temperature and other environmental factors affect propagation rates within the link media.

Note that equivalent accuracy can be achieved by updating the link partners timestamp generator more often.

1.6 Discussion of Implementation Options

This section discusses implementation options for the mechanisms used in "Section 1.4, Time Synchronization Using a Common Reference Clock" on page 9. Mechanisms required in "Section 1.5, Link Partners With RapidIO Standard Clocking" on page 12 (latch timestamp request) are not included in this proposal.

1.6.1 Loop Timing Request

The loop timing request must have low jitter and latency. For this reason, the Loop Timing Request should be a control symbol.

1.6.2 Loop Timing Response

The loop timing response must have low jitter and latency. For this reason, the Loop Timing Response should be a control symbol. The control symbol must be able to communicate an amount of Delay.

1.6.3 Set Time

The Set Time operation must have low jitter and latency. For this reason, Set Time should be implemented using control symbols.

Ideally, it should be possible to implement time synchronization for Gen1, Gen2 and Gen3 links. The size of control symbols for Gen1 links (4 bytes) is much less than that for Gen2 (8 bytes) or Gen3 (8 bytes). The approach chosen makes use of fields found in 4 byte control symbols to ensure that the same mechanism can be used for 8 byte control symbols.

Note that it is also possible to take a similar approach to IEEE 1588 by specifying either a new packet FType or a new maintenance packet Transaction Type code to be used specifically for setting timestamps. This alternative method is not part of the proposal, because it is difficult to meet the accuracy required for synchronizing timestamps in this manner.

The proposal does allow the Set Time operation to be performed using Maintenance Packets, rather than Control Symbols, to minimize the resources required to implement Time Synchronization.

1.7 Proposal for Changes to Part 6

This is a proposal for adding support for timestamp generators, and optional capabilities for timestamp generator synchronization, to the RapidIO Gen3 specification Physical Layer. The proposal consists of:

a) Several different profiles for devices
   - Devices with TSGs and no control symbol support for minimal implementation cost. Synchronization is accomplished using Maintenance Packets only.
   - TSG Slave devices accept control symbols which support loop delay calculations and TSG setting using control symbols. This design supports higher accuracy for time synchronization. It is capable of high accuracy if the TSG of the Slave uses the same clock source/frequency as the TSG Master.
   - TSG Master devices can transmit all time synchronization related control symbols. TSG Masters can also automatically/periodically update the link partner TSG using control symbols.

b) A new register extension block for Timestamp Generation and synchronization

— Only 5 registers are required to support Timestamp Generation with maintenance packet synchronization
   - No TSG Master or TSG Slave support with registers or control symbols

— TSG Masters require per-port timestamp and control registers for higher accuracy TSG synchronization functions, and the ability to generate and receive time synchronization control symbols
   - loop-delay request
   - loop response for loop-delay request
   - timestamp

— TSG Slaves require 5 header registers, one per-port register, and the ability to act as a target for TSG synchronization functions
   - Support link-request/loop-delay and timestamp control symbols

1.7.1 Stype0 Control Symbol Format Description Modification

Change Part 6 Section 3.4 “SType0 Control Symbols”, Table 3-2. “Stype0 Control Symbol Encoding” from:

Table 1-2. Stype0 Control Symbol Encoding

stype0	Function	Contents of		Reference
		Parameter0	Parameter1	
0b000	packet-accepted	packet_ackID	buf_status	Section 3.4.1
0b001	packet-retry	packet_ackID	buf_status	Section 3.4.2
0b010	packet-not-accepted	arbitrary	cause	Section 3.4.3
0b011	reserved	—	—	—
0b100	status	ackID_status	buf_status	Section 3.4.4
0b101	VC_status	VCID	buf_status	Section 3.4.5
0b110	link-response	ackID_status	port_status	Section 3.4.6
0b111	implementation-defined *	implementation-defined	implementation-defined	—

to:

Table 1-3. SType0 Control Symbol Encoding

stype0	Function	Contents of		Reference
		Parameter0	Parameter1	
0b000	packet-accepted	packet_ackID	buf_status	3.4.1
0b001	packet-retry	packet_ackID	buf_status	3.4.2
0b010	packet-not-accepted	arbitrary	cause	3.4.3
0b011	timestamp	timestamp Bits 0-4	timestamp Bits 5-9	3.4.4
0b100	status	ackID_status	buf_status	3.4.5
0b101	VC_status	VCID	buf_status	3.4.6
0b110	link-response	ackID_status	buf_status	3.4.7
0b111	implementation-defined	implementation-defined	implementation- defined	----

Add the following row to Table 3-3 Stype0 Parameter Definitions:

Table 1-4. Stype0 Parameter Definitions

Parameter	Definition
Timestamp	An amount of time, sent as a loop-response or as part of a timestamp update.

Add the following as a new Part 6 Section 3.4.4:

"3.4.4 Timestamp Control Symbol

Timestamp control symbols are used to set the timestamp generator value of the link partner with a high degree of accuracy, and as a response to a loop timing request (loop-response). Timestamp control symbols contains 10 bits of a time value which is spread across the parameter0 and parameter1 fields as shown in Figure 1-3.

Figure 1-3. Timestamp Control Symbol Formats

To set a timestamp generator value, a sequence of 8 timestamp control symbols is sent. Each timestamp control symbol in the sequence contains two flags and eight bits of the 64 bit timestamp generator value, as shown in Table 1-5.

Table 1-5. Sequence of Timestamp Control Symbols

Control Symbol Sequence	Parameter 0 Bit 0 "Start Flag"	Parameter 0 Bit 1 "End Flag"	Parameter 0 Bits 2-4	Parameter 1 Bits 0-4
0	1	0	Timestamp [0:2]	Timestamp[3:7]
1	0	0	Timestamp [8:10]	Timestamp[11:15]
2	0	0	Timestamp [16:18]	Timestamp[19:23]
3	0	0	Timestamp [24:26]	Timestamp[27:31]
4	0	0	Timestamp [32:34]	Timestamp[35:39]
5	0	0	Timestamp [40:42]	Timestamp[43:47]
6	0	0	Timestamp [48:50]	Timestamp[51:55]
7	0	1	Timestamp [56:58]	Timestamp[59:63]

Support for transmission and reception of a sequence of 8 timestamp control symbols is optional. A processing element shall support transmitting a sequence of 8 timestamp control symbols when the Timestamp Master Supported bit of the Timestamp CAR is 1. A processing element shall support receiving a sequence of 8 timestamp control symbols when the Timestamp Slave Supported bit of the Timestamp CAR is 1.

A loop-response shall consist of a single timestamp control symbol transmitted in response to a loop timing request. The timestamp is a 10 bit value which represents the number of nanoseconds between the time the loop-timing request was received by the link partner, and the time the loop-response was generated. A value of 0x3FF indicates that the amount of delay exceeds 1022 nanoseconds and so is too large to be encoded.

When a loop-response is received while a loop timing request is outstanding, the current value of the Timestamp Generator shall be captured in the Port n Timestamp 1 MSW CSR and Port n Timestamp 1 LSW CSR and the delay value of the loop-response control symbol shall be captured in the Port n Timestamp Synchronization Status CSR.

A processing element shall support receiving a loop timing response when the Timestamp Master Supported bit of the Timestamp CAR is 1. A processing element shall support transmitting a loop-timing response when the Timestamp Slave Supported bit of the Timestamp CAR is 1."

1.7.2 Stype1 Control Symbol Format Description Modification

Change table 3-7 in section 3.5 from:

Table 1-6. Stype1 Control Symbol Encoding

stype1	stype1 Function	cmd	cmd Function	Packet Delimiter	Reference
0b000	Start-of-packet	0b000	Start-of-packet	yes	Section 3.5.1
		0b001 - 0b111	Reserved	No	
0b001	Stomp	0b000	Stomp	yes	Section 3.5.2
		0b001 - 0b111	Reserved	No	
0b010	End-of-packet	0b000	End-of-packet	yes	Section 3.5.3
		0b001 - 0b111	Reserved	No	
0b011	Restart-from-retry	0b000	Restart-from-retry	*	Section 3.5.4
		0b001 - 0b111	Reserved	No	
0b100	Link-request	0b000 - 0b010	Reserved	*	-
		0b011	Reset-device		Section 3.5.5.1
		0b100	Input-status		Section 3.5.5.2
		0b101	Reset-port		-
		0b110- 0b111	Reserved		
0b101	Multicast-event	0b000	Multicast-event	No	Section 3.5.6
		0b001 - 0b111	Reserved	No	
0b110	Reserved	0b000 - 0b111	Reserved	No	-

Table 1-6. Stype1 Control Symbol Encoding

stype1	stype1 Function	cmd	cmd Function	Packet Delimiter	Reference
0b111	NOP (Ignore) **	0b000	NOP (Ignore) **	No	-
		0b001 - 0b111	Reserved	No	

to:

Table 1-7. Stype1 Control Symbol Encoding

stype1	stype1 Function	cmd	cmd Function	Packet Delimiter	Reference
0b000	Start-of-packet	0b000	Start-of-packet	yes	Section 3.5.1
		0b001 - 0b111	Reserved	No	
0b001	Stomp	0b000	Stomp	yes	Section 3.5.2
		0b001 - 0b111	Reserved	No	
0b010	End-of-packet	0b000	End-of-packet	yes	Section 3.5.3
		0b001 - 0b111	Reserved	No	
0b011	Restart-from-retry	0b000	Restart-from-retry	*	Section 3.5.4
		0b001 - 0b111	Reserved	No	
0b100	Link-request	0b000 - 0b010	Reserved	*	-
		0b011	Reset-device		Section 3.5.5.1
		0b100	Input-status		Section 3.5.5.2
		0b101	Reset-port		-
		0b110- 0b111	Reserved		
0b101	Timing	0b000	Multicast-event	No	Section 3.5.6.1
		0b001 - 0b010	Reserved		
		0b011	Loop-Timing Request		Section 3.5.6.2
		0b100 - 0b111	Reserved		
0b111	NOP (Ignore) **	0b000	NOP (Ignore) **	No	-
		0b001 - 0b111	Reserved	No	

Renumber existing section 3.5.6 to 3.5.6.1.

Create new section 3.5.6 as follows:

“3.5.6 Timing Control Symbols

Timing control symbols are all related to communication of events and time within a system. Unlike other control symbols, timing control symbols can trigger activity on other links of a device.”

Create new section 3.5.6.2 as follows:

"3.5.6.2 Loop Timing Request

Support for transmission and reception of the Loop Timing Request control symbol is optional. A processing element shall be capable of transmitting a Loop Timing Request control symbol if the Timestamp Master Support bit of the Timestamp CAR is 1. A processing element shall be capable of receiving a Loop Timing Request control symbol if the Timestamp Slave Support bit is 1 in the Timestamp CAR.

When a processing element transmits a Loop Timing Request control symbol, the value of it's Timestamp Generator shall be latched in the Port n Timestamp 0 MSW CSR and Port n Timestamp 0 LSW CSR. When a processing element receives a Loop Timing Request control symbol, the processing element shall transmit a loop-response control symbol."

1.7.3 Time Synchronization Algorithms Description

Create a new section 5.5.3.5 Time Synchronization as follows:

"5.5.3.5 Time Synchronization Protocol

Time synchronization is the method of synchronizing the "sense of time" between RapidIO processing elements. The "sense of time" is embodied in a Time Stamp Generator (TSG) for each node.

A TSG consists of a single 64 bit nanosecond granularity counter. The timestamp generator is divided between two 32 bit registers :

- The Timestamp Generator LSW CSR register contains the least significant 32 bits of the TSG.
- The Timestamp Generator MSW CSR register contains the most significant 32 bits of the TSG.

A timestamp is a 64 bit value, consisting of the MSW register in the most significant bits and the LSW register in the least significant bits.

The TSG counter increases in value using an implementation defined step size.

Synchronization of TSGs is supported with varying degrees of accuracy. For example:

- Synchronization of TSGs with microseconds of accuracy is required. TSGs are synchronized between link partners using specific control symbols. TSG's advance at the same frequency, +/- 100 PPM.

- Synchronization of TSGs within less than a microsecond is required. TSGs are synchronized between link partners using specific control symbols. The difference in frequency between link partners is calibrated and compensated for. Timestamp Generator master devices regularly update Timestamp Generator Slave devices.
- Synchronization of TSGs within 100 nanoseconds or less is required. All TSG's use the same clock frequency to control the rate at which time advances. The delay between link partners is calibrated using control symbols to ensure maximal accuracy. TSG's are synchronized between link partners regularly, adjusting for the delay between link partners.

The following sections discuss mechanisms which implement the above synchronization. These mechanisms may use either maintenance reads/writes or control symbols.

Table 1-8 summarizes the control symbol support required to implement the timestamp synchronization protocol based on the values of the Timestamp CAR fields. Note that for devices which can be both a TSG Master and Slave, Slave control symbol support is required in TSG Slave mode, and Master control symbol support is required when in TSG Master mode, where the TSG mode is determined by the Port Operating Mode field of the Port n Timestamp Generator Synchronization CSR. .

Table 1-8. Control Symbol Support for TSG Master and Slave Devices

Control Symbol	Timestamp Master/ Slave Supported Both = 0	Timestamp Slave Supported = 1	Timestamp Master Supported = 1
Loop-Timing Request	None	Receive	Transmit
Loop Response	None	Transmit	Receive
Timestamp Sequence	None	Receive	Transmit

5.5.3.5.1 Setting and Reading a Timestamp Generator

To set the TSG registers using Maintenance Writes, first write the TSG LSW register, then write the TSG MSW register. Software may elect to delay updating the TSG when the TSG LSW register is close to rolling over to avoid incrementing the TSG MSW register.

To read the TSG registers using Maintenance Reads, first read the TSG MSW register, then the TSG LSW register, and then the TSG MSW register again. If the value of the TSG MSW register has not changed, then the timestamp

value has been read successfully. If the value of the MSW register has changed, the TSG LSW register shall be read again to compose an accurate timestamp.

Devices may support 8 byte register reads and writes, which allow the MSW and LSW registers to be read and updated simultaneously.

The TSG of a link partner may also be set using Timestamp control symbols. A sequence of eight time stamp control symbols shall be sent to set the link partner timestamp generator value, as described in Table 1-5.

The timestamp value in the Timestamp control symbols shall be sent as if all 64 bits were captured when the first Timestamp control symbol was formulated. The timestamp value sent may have a nanoseconds offset added to it before transmission to account for transmission delay. The offset is a programmable value found in the Port n Timestamp Offset CSR.

A sequence of eight Timestamp control symbols shall not be interrupted by any other control symbols or an IDLE sequence.

If all eight control symbols are not received correctly, without interruption, the receivers TSG shall not be updated.

The receiver may adjust the timestamp value, if necessary, to reflect transmission delay due to control symbol alignment and/or the time required to receive eight Timestamp control symbols.

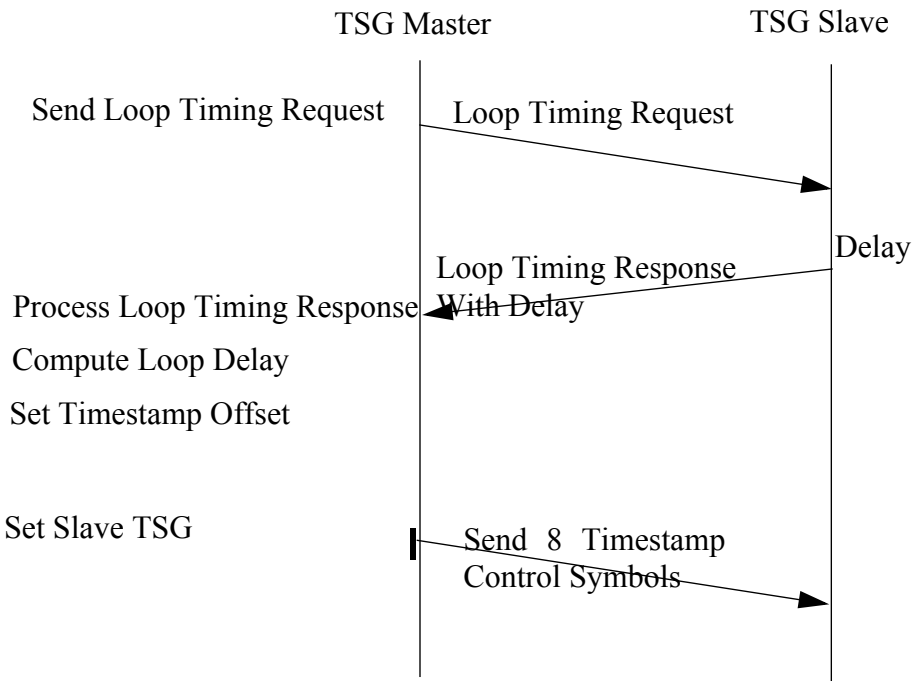
A timestamp generator shall immediately change it's value to 0 when programmed to do so. A timestamp generator shall immediately change it's value when programmed to a value larger than the current TSG value.

When either control symbols or maintenance packets are used to change a TSG to a value which is less than the current TSG value, the TSG value shall be held constant for the difference in time between the current TSG value and the time which was programmed. This has the effect of halting time until the new time value is reached. Implementations shall allow the current time value to be held constant for a period of 65535 nanoseconds. Setting a timestamp value more than 65535 nanoseconds in the past results in implementation specific behavior.

5.5.3.5.2 Calibrating Transmission Delay

The procedure for calibrating the transmission delay between the Master and Slave is shown in the Figure 1-4 message sequence chart.

Figure 1-4. Time Synchronization with Synchronous Link Partners



The steps are defined as follows:

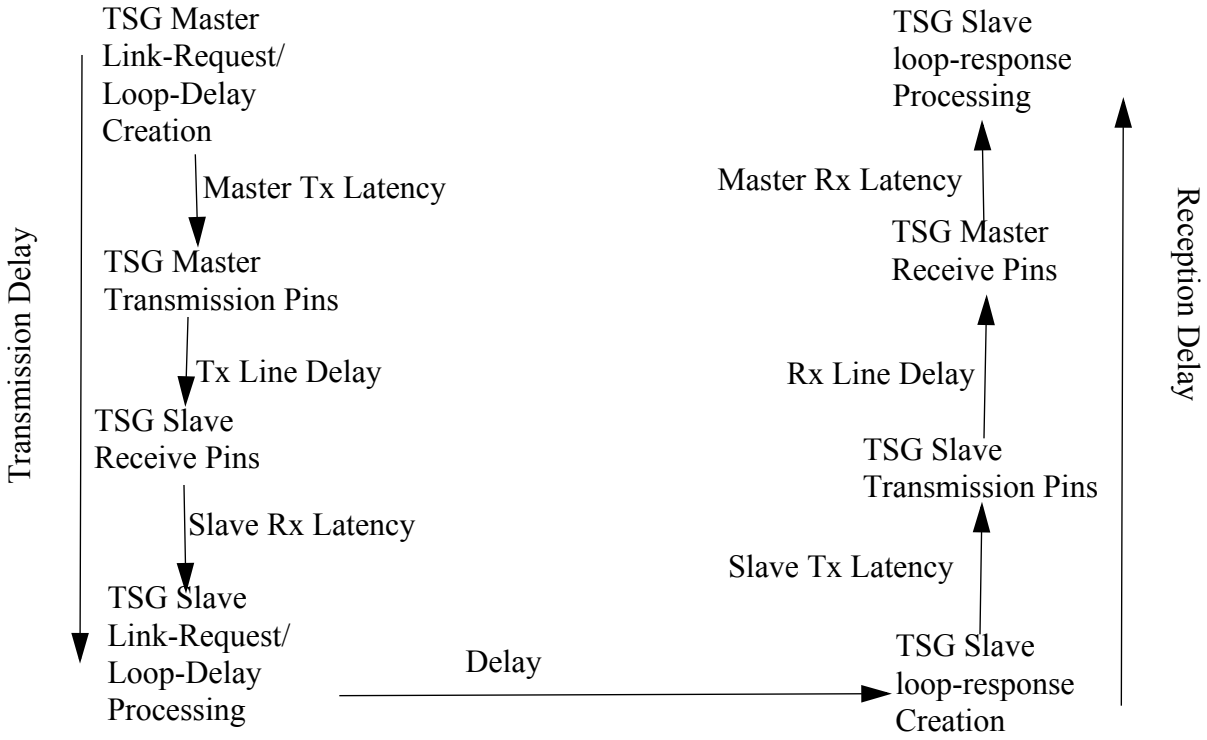
1. Send Loop Timing Request: The TSG Master sends a Loop-Timing Request control symbol to the TSG Slave by writing "0x00000003" to the Port n Timestamp Synchronization Command CSR. This latches the current TSG Master TSG value in the Port n Timestamp 0 MSW CSR and Port n Timestamp 0 LSW CSR.
2. Process Loop Timing Response: The TSG Master receives a Loop-response for the Loop-Timing Request control symbol which contains the Delay amount in the TSG Slave. This also causes the current TSG Master TSG value to be latched in the Port n Timestamp 1 MSW CSR and Port n Timestamp 1 LSW CSR.
3. Compute Loop Delay: The TSG Master computes the loop timing delay as described at the end of this section.
4. Set Timestamp Offset: The TSG Master programs it's Port n Timestamp Offset register value to the Loop Delay computed in Step 3.

5. Set Slave TSG: The TSG Master sets the TSG Slaves Timestamp Generator value by writing "0x00000010" to the Port n Timestamp Synchronization Command CSR.

A loop-response for a loop-timing request must be received within the link response timeout period. If the loop-response is not received within the link response timeout period, then the loop timing request shall be treated as completed. A loop-timing request shall not be retransmitted in the event of a timeout.

Computing loop delay is complicated by the need to account for transmit and receive asymmetries in the transmitter and receiver of the TSG Slave and TSG Master. These asymmetries are shown in Figure 1-6.

Figure 1-5. Asymmetry Computation



The following computation uses the notation "(Condition)?Val1:Val2" to describe a function which returns "Val1" if "Condition" is true, and "Val2" if "Condition" is false.

The term "Master Tx Asymmetry" below is computed using fields in the TSG Master's Port n Timestamp Synchronization Status CSR fields.

The term "Slave Tx Asymmetry" below is computed using fields in the TSG Slave's Port n Timestamp Synchronization Status CSR fields.

Total Delay = Timestamp 1 - Timestamp 0 - Delay

Master Tx Asymmetry = (Asymmetry / 2) * ((Tx is Faster Than Rx = 1)?-1:1)

Slave Rx Asymmetry = (Asymmetry / 2) * ((Tx is Faster Than Rx = 1)?1:-1);

Transmission Delay = (Total Delay / 2) + Master Tx Asymmetry + Slave Rx Asymmetry

Note that it is not possible for the TSG Master to measure the actual transmission delay. The transmission delay computed is still inaccurate by half the difference between Tx Line Delay and Rx Line Delay. Solutions which require highly accurate TSG synchronization can minimize asymmetry between Tx Line Delay and Rx Line Delay through physical design constraints.

5.5.3.5.3 Regular Timestamp Generator Re-synchronization

It may be necessary to regularly update TSG values throughout the RapidIO fabric, for example when endpoints in the system do not support Common Frequency. Two methods of automatic re synchronization are possible:

- If a device supports both a TSG Slave port which receives timestamp updates, and TSG Master ports which transmit timestamp updates, it is easiest to automatically update the TSG Master ports link partners whenever the TSG Slave port is updated.
- If a device is the TSG Master for the entire system, the device can be configured to regularly update it's link partners sense of time.

A TSG Master port can be configured to update it's link partner whenever the TSG is updated by setting the "Auto-update Link Partner Timestamp Generators" field to 1 in the Port n Timestamp Synchronization CSR.

A TSG Master port can be configured to periodically update it's link partner. The period is programmed by setting the Update Period field of the Port n Auto Update Counter CSR. Periodic updates are enabled by setting the "Periodically Update Link Partner Timestamp Generators" bit to 1 in the Port n Timestamp Synchronization CSR.

5.5.3.5.4 Timestamp Generator Synchronization Control Symbol Jitter

The accuracy of TSG synchronization is dependent upon the consistency with which frequency differences and loop delay can be measured. To ensure maximum accuracy, the following points should be considered with respect to TSG Slave and TSG Master support.

The point in the design where "Loop Timing Request", "Link Response for Loop Timing Request", and "Timestamp" control symbols are generated should have identical latency with respect to the Timestamp Generator from

the time the control symbol is formulated to the time the control symbol is transmitted.

The point in the design where “Loop Timing Request”, “Link Response for Loop Timing Request”, and “Timestamp” control symbols are processed should have identical latency with respect to the Timestamp Generator. from the time the control symbol is received by the SerDes to the time the control symbol is processed.

These two points ensure that measurements can be applied consistently to frequency offset calculations and loop delay calculations. “

1.7.4 Changes to Error Detection and Recovery Description

Add the following bullet to the list of link protocol violations:

“

- Receipt of an unsolicited Timestamp Control Symbol when the device is a timestamp Master
- Receipt of a sequence of Timestamp Control Symbols which does not conform to the sequence described in Table 1-5.

“

1.7.5 Time Synchronization Register Block

A new extension block is defined for timing synchronization as follows:

“6.9 Timestamp Generation Extension Block

The Timestamp Generation Extension Block contains different registers depending on the Timestamp CAR field values. The “General” column indicates registers which must be implemented regardless of the values of the Timestamp CAR.

The “Time Slave” column indicates which registers must be implemented when the “Timestamp Slave Supported” bit is 1. The “Time Master” column indicates which registers must be implemented when the “Timestamp Master Supported” bit is 1. In all cases, an “X” in a column means that the register shall be implemented.

If the “Timestamp Slave Supported”and “Time Master” bits are 0, then only General registers shall be implemented..

Table 1-9. Timestamp Generation Extension Block

	Block Byte Offset	Register Name	General	Time Slave	Time Master
General	0x00	Timestamp Generation Extension Block Header	X	X	X
	0x04	Timestamp CAR	X	X	X
	0x08	Timestamp Generator Status CSR	X	X	X
	0x0C-1C	Reserved	-	-	-
	0x20-2C	Implementation Specific	-	-	-
	0x30	Reserved	-	-	-
	0x34	Timestamp Generator MSW CSR	X	X	X
	0x38	Timestamp Generator LSW CSR	X	X	X
	0x3C	Reserved	-	-	-
Port 0	0x40	Reserved	-	-	-
	0x44	Port 0 Timestamp 0 MSW CSR	-	-	X
	0x48	Port 0 Timestamp 0 LSW CSR	-	-	X
	0x4C-50	Reserved	-	-	-
	0x54	Port 0 Timestamp 1 MSW CSR	-	-	X
	0x58	Port 0 Timestamp 1 LSW CSR	-	-	X
	0x5C	Reserved	-	-	-
	0x60	Port 0 Timestamp Generator Synchronization CSR	-	X	X
	0x64	Port 0 Auto Update Counter CSR	-	-	X
	0x68	Port 0 Timestamp Synchronization Command CSR	-	-	X
	0x6C	Port 0 Timestamp Synchronization Status CSR	-	-	X
	0x70	Port 0 Timestamp Offset CSR	-	-	X
	0x74-7C	Implementation Specific	-	-	-
Ports 1-14	0xC0–3FC	Assigned to Port 1-14 CSRs			

Table 1-9. Timestamp Generation Extension Block

	Block Byte Offset	Register Name	General	Time Slave	Time Master
Port 15	0x400	Reserved	-	-	-
	0x404	Port 15 Timestamp 0 MSW CSR			X
	0x408	Port 15 Timestamp 0 LSW CSR			X
	0x40C-410	Reserved	-	-	-
	0x414	Port 15 Timestamp 1 MSW CSR			X
	0x418	Port 15 Timestamp 1 LSW CSR			X
	0x41C	Reserved	-	-	-
	0x420	Port 15 Timestamp Generator Synchronization CSR	-	X	X
	0x424	Port 15 Auto Update Counter CSR	-	-	X
	0x428	Port 15 Timestamp Synchronization Command CSR	-	-	X
	0x42C	Port 15 Timestamp Synchronization Status CSR	-	-	X
	0x430	Port 15 Timestamp Offset CSR	-	-	X
	0x434-43C	Implementation Specific	-	-	-

6.9.1 Timestamp Generation Extension Block Header (Block Offset 0x0)

The Timestamp Generation Extension Block Header register contains the EF_PTR to the next EF_BLK and the EF_ID that identifies this as the Timestamp Generation Extension Block Header. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-10. The register is read-only.

Table 1-10. Bit Settings for Timestamp Generation Extension Block Header

Bit	Name	Reset Value	Description
0-15	EF_PTR		Hard wired pointer to the next block in the data structure, if one exists
16-31	EF_ID	0x000E	Hard wired Extended Features ID

6.9.2 Timestamp CAR (Block Offset 0x04)

This register indicates which Timestamp Synchronization capabilities the device supports. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-11. The bits and bit fields in this register are read only.

Table 1-11. Bit Settings for Timestamp CAR

Bit	Name	Reset Value	Description
0	Timestamp Slave Supported	See Footnote 1	Indicates whether the device supports operation as a Timestamp Slave. 0b0 - Device does not support operation as a Timestamp Slave 0b1 - Device supports operation as a Timestamp Slave.
1	Timestamp Master Supported	See Footnote 1	Indicates whether the device supports operation as a Timestamp Master. 0b0 - Device does not support operation as a Timestamp Master 0b1 - Device does support operation as a Time
2	Common Clock Frequency Supported	See Footnote 1	Indicates whether the device supports use of a common clock frequency 0b0 - Device does not support common clock frequency 0b1 - Device does support common clock frequency
3-31	---	0x00	Reserved

1 The reset value of this field is implementation specific.

6.9.3 Timestamp Generator Status CSR (Block Offset 0x08)

This register indicates the current status of the Timestamp Generator. Note that Table 1-12 contains two columns, denoted “All” and “Common Freq”. An “X” in the “All” column indicates bits which shall be implemented in this register. An “X” in the “Common Freq” column indicates bits which shall be implemented if the Timestamp CAR Common Clock Frequency Support bit field is 0b1.

The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-12. Unless otherwise noted in each field description, the bits and bit fields in this register are read only unless otherwise specified.

Table 1-12. Bit Settings for Timestamp Generator Status CSR

Bit	Name	Reset Value	Description	All	Common Freq
0	Timestamp Generator Clock Locked	See Footnote 1	Indicates whether the Timestamp Generator counter is operating from a good clock source. 0b0 - Timestamp Generator is not operating with a good clock source. 0b1 - Timestamp Generator is operating with a good clock source.	X	-
1	Timestamp Generator Common Clock	See Footnote 1	Indicates whether the Timestamp Generator counter is operating based on a clock frequency which is the same as that of the link partners. 0b0 - Timestamp Generator is not operating with a common clock frequency. 0b1 - Timestamp Generator is operating with a common clock frequency.	-	X

Table 1-12. Bit Settings for Timestamp Generator Status CSR

Bit	Name	Reset Value	Description	All	Common Freq
2	Timestamp Generator Stopped	See Footnote 1	Indicates if the Timestamp Generator counter is not advancing because it is being set to an earlier time. 0b0 - Timestamp Generator is advancing 0b1 - Timestamp Generator is temporarily not advancing	X	-
3	Timestamp Generator Was Stopped	See Footnote 1	Indicates if the Timestamp Generator counter has not advanced because it has been set to an earlier time. 0b0 - Timestamp Generator has advanced continuously since this bit was last cleared 0b1 - Timestamp Generator had temporarily stopped advancing at least once since this bit was last cleared. This bit may be cleared by writing "1" to it.	X	-
4-31	---	0x00	Reserved	-	-

1 The reset value of this field is implementation specific.

6.9.4 Timestamp Generator MSW CSR (Block Offset 0x034)

This register indicates the most significant 32 bits of the timestamp generator. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-15. The bits and bit fields in this register are read/write.

Table 1-13. Bit Settings for Timestamp Generator MSW CSR

Bit	Name	Reset Value	Description
0-31	MSW Bits	0x00000000	Most significant 32 bits for the timestamp generator.

6.9.5 Timestamp Generator LSW CSR (Block Offset 0x038)

This register indicates the least significant 32 bits for the timestamp generator. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-16. The bits and bit fields in this register are read/write.

Table 1-14. Bit Settings for Timestamp Generator LSW CSR

Bit	Name	Reset Value	Description
0-31	LSW Bits	0x00000000	Least significant 32 bits for the timestamp generator.

6.9.6 Port n MSW Timestamp 0 CSR
(Block Offsets 0x44, 0x84, 0xC4, ..., 0x404)

This register contains the value of the Timestamp Generator MSW CSR when a Latch Timestamp 0 control symbol is transmitted or received. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-17. The bits and bit fields in this register are read only.

Table 1-15. Bit Settings for Port n Timestamp 0 MSW CSR

Bit	Name	Reset Value	Description
0-31	MSW Bits	0x00000000	Most significant 32 bits from the timestamp generator.

6.9.7 Port n Timestamp 0 LSW CSR
(Block Offsets 0x48, 0x88, 0xC8, ..., 0x408)

This register contains the value of the Timestamp Generator LSW CSR when a Latch Timestamp 0 control symbol is transmitted or received. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-18. The bits and bit fields in this register are read only.

Table 1-16. Bit Settings for Port n Timestamp 0 LSW CSR

Bit	Name	Reset Value	Description
0-31	LSW Bits	0x00000000	Least signficiant 32 bits from the timestamp generator.

6.9.8 Port n Timestamp 1 MSW CSR
(Block Offsets 0x54, 0x94, 0xD4, ..., 0x414)

This register contains the value of the Timestamp Generator MSW CSR when a Latch Timestamp 1 control symbol is transmitted or received. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-19. The bits and bit fields in this register are read only.

Table 1-17. Bit Settings for Port n Timestamp 1 msw CSR

Bit	Name	Reset Value	Description
0-31	MSW Bits	0x00000000	Most signficiant 32 bits from the timestamp generator.

6.9.9 Port n Timestamp 1 LSW CSR
(Block Offsets 0x58, 0x98, 0xD8, ..., 0x418)

This register contains the value of the Timestamp Generator LSW CSR when a Latch Timestamp 1 control symbol is transmitted or received. The use and

meaning of the bits and bit fields of this register shall be as specified in Table 1-20. The bits and bit fields in this register are read only.

Table 1-18. Bit Settings for Port n Timestamp 0 LSW CSR

Bit	Name	Reset Value	Description
0-31	LSW Bits	0x00000000	Least signficiant 32 bits from the timestamp generator.

6.9.10 Port n Timestamp Generator Synchronization CSR (Block Offsets 0x60, 0xA0, 0xD0, ..., 0x420)

This register controls the Timestamp Generator Synchronization capabilities which the port will accept and transmit. The columns in Table 1-21 determine which fields must be implemented, based on the bit field values of the Timestamp CAR.

The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-21. The bits and bit fields in this register are read/write.

Table 1-19. Bit Settings for Port n Timestamp Generator Synchronization CSR

Bit	Name	Reset Value	Description	Time Slave	Time Master	Com. Freq.
0	Accept Timestamps	0b0	Indicates whether the device will accept Timestamp Control Symbols from the link partner. 0b0 - Device will not accept Timestamp Control Symbols from the link partner. 0b1 - Device accepts Timestamp Control Symbols from the link partner.	X	-	-
1	Disable Clock Compensation Sequence	0b0	Controls whether the device will transmit Clock Compensation Sequences. 0b0 - Device transmits clock compensation sequences regularly as required 0b1 - Device does not transmit clock compensation sequences.	-	-	X
2	Auto-update Link Partner Timestamp Generators	0b0	Controls whether the device will automatically update the timestamp generator of the link partner connected to this port if the timestamp generator on this device is set. 0b0 - Do not automatically update the link partner timestamp generator 0b1 - Automatically update the link partner timestamp generator whenever the timestamp generator on this device is set.	-	X	-
3-5	---	0x00	Reserved	-	-	-

Table 1-19. Bit Settings for Port n Timestamp Generator Synchronization CSR

Bit	Name	Reset Value	Description	Time Slave	Time Master	Com. Freq.
6-7	Port Operating Mode	0b00	When a port supports both time slave and master capabilities, this bit is used to control the port's operating mode. 0b00 - Master and slave functionality disabled 0b01 - Time slave functionality enabled 0b10 - Time master functionality enabled 0b11 - Reserved	X	X	-
8-18	---	0x00	Reserved	-	-	-
19	Tx Has Lower Latency	See Footnote 1	Indicates whether the transmit path has lower latency than the receive path, or vice versa. This value controls how the Asymmetry field is applied to loop delay calculations. 0b0 - Tx has higher latency than Rx. 0b1 - Tx has lower latency than Rx.	X	X	-
20-31	Asymmetry	See Footnote 1	Measure of the latency difference between the receive path and transmit path of this port. The value represents the number of nanoseconds. 0x000 - No difference between transmit and receive control path latency. 0x001 - One nanosecond difference between transmit and receive control path latency ... 0xFFF - 4096 nanoseconds difference between transmit and receive control path latency.	X	X	-

1 The reset value of this field is implementation specific.

6.9.11 Port n Auto update Counter CSR (Block Offsets 0x64, 0xA4, 0xD4, ..., 0x424)

This register determines how often a timestamp generator master updates the link partners timestamp generator. This is done on a per port basis since each link partner may have different tolerances/requirements for timestamp updates. The interval allows the link partner to be updated with intervals that range from once a microsecond to once an hour.

Periodic timestamp updates shall not be sent when this register is 0.

The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-22. The bits and bit fields in this register are read/write.

Table 1-20. Bit Settings for Port n Auto Update Counter CSR

Bit	Name	Reset Value	Description
0-31	Update Period	0x00000000	Time between timestamp updates. Units are 1024 nanoseconds.

6.9.12 Port n Timestamp Synchronization Command CSR (Block Offsets 0x68, 0xA8, 0xD8, ..., 0x428)

This register enables Loop Timing Request control symbols to be sent to the link partner. This register also allows a sequence of 4 Timestamp Control Symbols to be sent to the link partner to set the link partners timestamp generator.

The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-23. The bits and bit fields in this register are read/write.

Table 1-21. Bit Settings for Port n Timestamp Synchronization Command CSR

Bit	Name	Reset Value	Description
0-22	---	0x00	Reserved
23	Send Zero Timestamp	0b0	A port shall transmit a sequence of eight Timestamp Control Symbols when this field is written with a value of 1 and the Port Operating Mode is set to 0b10 (Master Enabled). The Timestamp Control Symbols shall carry a value of zero for all timestamp generator bits.
24-26	---	0x00	Reserved
27	Send Timestamp	0b0	A port shall transmit a sequence of eight Timestamp Control Symbols when this field is written with a value of 1 and the Port Operating Mode is set to 0b10 (Master Enabled). The Timestamp Control Symbols shall carry the value of the current timestamp generator, with the addition of the Port n Timestamp Offset CSR
28	---	0x00	Reserved
29-31	Command	0b000	Contents of the “Cmd” field of a Timing control symbol to send to the link partner. Legal values are: 0b000 - Send Multicast Event Control Symbol 0b011 - Send Loop Timing Request Control Symbol

6.9.13 Port n Timestamp Synchronization Status CSR (Block Offsets 0x6C, 0xAC, 0xDC, ..., 0x42C)

This register contains the status of pending commands sent using the Port n Timestamp Synchronization Command CSR.

The use and meaning of the bits and bit fields of this register shall be as

specified in Table 1-24. The bits and bit fields in this register are read only.

Table 1-22. Bit Settings for Port n Timestamp Synchronization Status CSR

Bit	Name	Reset Value	Description
0	response_valid	0b0	If the Timestamp Synchronization Command causes a loop-response, this bit indicates that the loop-response has been received and the status fields are valid. If the Timestamp Synchronization Command does not cause a loop-response, this bit indicates that the Timestamp Synchronization Command or Timestamp request has been transmitted. This bit automatically clears on read.
28	---	0x00	Reserved
22-31	Delay	0x000	Contents of the "Delay" field of the Link Response control symbol: This field shall be valid when a loop-timing request was transmitted and the response_valid field is 1. A value of 0x3FF indicates that the delay in the link partner exceeded 1022 nsec.

6.9.14 Port n Timestamp Offset CSR
(Block Offsets 0x70, 0xB0, 0xE0, ..., 0x430)

This register contains the number of nanoseconds to add to the current Timestamp Generator value before sending a sequence of Timestamp control symbols to the link partner. The use and meaning of the bits and bit fields of this register shall be as specified in Table 1-25. The bits and bit fields in this register are read/write.

Table 1-23. Bit Settings for Port n Timestamp Offset CSR

Bit	Name	Reset Value	Description
0-15	Offset	0x0000	Count of the number of nanoseconds to add to the timestamp generator value when transmitting a sequence of Timestamp Control Symbols.
16-31	---	0x0000	Reserved