Ethernet Design Example Components User Guide

Last updated for Quartus Prime Design Suite: 16.0
UG-20019
2016.05.02

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Subscribe
Send Feedback

now part of Intel

Contents

Time-of-day Clock 1

2016.05.02

UG-20019 Subscribe Send Feedback

The ToD Clock streams 96-bit and 64-bit timestamps to one or more timestamping units in an IEEE 1588v2 solution. The timestamps consist of the following fields.

Field	96-bit Timestamp	64-bit Timestamp
Second	48 bits	–
Nanosecond	32 bits	48 bits
Fractional nanosecond	16 bits	16 bits

This component supports coarse and fine adjustments, and period correction. It also supports configurable period adjustment and offset adjustment.

Supported Devices

- Arria V GX/GT/GZ/SX/ST
- Arria 10 GX/GT/SX
- Cyclone V SE/SX/ST
- MAX 10
- Stratix V GX/GT

Resource Utilization

Table 1-1: Estimated Resource Utilization in Stratix V Devices

Configuration	ALMs	Combinational ALUTs	Logic Registers	Memory (M20K/MLAB)
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 0	727	733	1041	0
PERIOD_CLOCK_FREQUENCY = 1 OFFSET_JITTER_WANDER_EN = 0	764	736	1224	0

Configuration	ALMs	Combina-tional ALUTs	Logic Registers	Memory (M20K/MLAB)
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 1	1782	2406	2297	0

Configuring the ToD Clock

In the Quartus® Prime software, instantiate the ToD Clock by selecting it from the IP Catalog or Qsys (**Interface Protocols > Ethernet > Reference Design Components**). Specify the following parameters.

Table 1-2: Parameter Description

Name	Value	Default Value	Description
PERIOD_CLOCK_FREQUENCY	0 or 1	1	Set this parameter to 0 if the MAC connected to the ToD clock requires low period clock frequency, such as the Triple-speed Ethernet or legacy 10G Ethernet MAC. For this setting, the nanosecond field in the Period and AdjustPeriod registers is 9 bits wide. Set this parameter to 1 if the MAC connected to the ToD clock requires high period clock frequency, such as Low-latency 10G Ethernet or 40G/100G Ethernet MAC. For this setting, the nanosecond field in the Period and AdjustPeriod registers is 4 bits wide.
OFFSET_JITTER_WANDER_EN	0 or 1	0	Set this parameter to 1 to enable the offset, jitter, and wander timers; 0 otherwise. This parameter is enabled only when PERIOD_CLOCK_FREQUENCY is set to 0.
DEFAULT_NSEC_PERIOD	0 – *n*	0x0006	The reset value of the nanosecond field in the Period register. *n* is 0xF if the nanosecond field is 4 bits wide. Otherwise, *n* is 0x1FF.
DEFAULT_FNSEC_PERIOD	0 – 0xFFFF	0x6666	The reset value of the fractional nanosecond field in the Period register.
DEFAULT_NSEC_ADJPERIOD	0 – *n*	0x0006	The reset value of the nanosecond field in the AdjustPeriod register. *n* is 0xF if the nanosecond field is 4 bits wide. Otherwise, *n* is 0x1FF.
DEFAULT_FNSEC_ADJPERIOD	0 – 0xFFFF	0x6666	The reset value of the fractional nanosecond field in the AdjustPeriod register.

Using the ToD Clock

Follow these guidelines when using the ToD Clock:

- 96-bit timestamps—load the timestamp using the `time_of_day_96b_load_data[]` bus or the `SecondsH`, `SecondsL`, and `NanoSec` registers. The bus value always takes precedence over the register values. When loading the timestamp through the `time_of_day_96b_load_data[]` bus, the output is available in the `time_of_day_96[]` bus after one clock cycle. Hence, Altera recommends that you add one clock cycle to the value of the `time_of_day_96b_load_data[]` bus to accommodate the latency.
- 64-bit timestamps—load the timestamp using the `time_of_day_64b_load_data[]` bus. The output is available in the `time_of_day_64[]` bus after one clock cycle. Hence, Altera recommends that you add one clock cycle to the value of the incoming timestamp to accommodate the latency.
- The ToD Clock does not synchronize the 96-bit and 64-bit timestamps.
- The drift, jitter, and wander timers restart each time a new time of day is loaded, either through the signal or configuration registers.

Adjusting ToD Drift

You can use the `DriftAdjust` and `DriftAdjustRate` registers to correct drifts in the ToD clock due to insufficient binary representation of the 16-bit fractional nanosecond field in the `Period` register.

For example, the `Period` register is set to 6.4ns for a 10G Ethernet application. The hexadecimal representation of this value is 0x6 ns and 0x6666.4 fns. The fractional nanosecond value, 0x0000.4, cannot be represented in 16 bits thus causing the time of day to drift from the actual time by 0x0002 fns every 5 clock cycles. In other words, the time of day drifts 953.6ns every 1 second. To correct this situation, configure the registers as follow:

- `DriftAdjust` = 0x02, which sets the nanosecond field to 0x0 and the fractional nanosecond field to 0x2.
- `DriftAdjustRate` = 0x5.

Adjusting Offset, Jitter, and Wander

The ToD Clock supports several types of adjustments:

- Offset—you can use the `OffsetNS` and `OffsetFNS` registers to adjust large offsets in assisting faster system convergence. The offset can be positive or negative. The maximum correction is (10^9 - 50) ns.
- Jitter—you can use the `JitterTimer` and `JitterAdjust` registers to achieve small time scales (milliseconds or microseconds) frequency correction.

  The jitter adjustment can either be a positive or negative adjustment per unit time. This helps achieve better frequency corrections. For very low values of the jitter, such as 1 ns correction for every second, the timer must be larger and the adjustment value must be smaller.

  For example, to achieve 1 ns correction every second in a clock domain of 3.2 ns period, configure the registers as follow:

  - `JitterTimer` = 0x12A05F20, which is the hexadecimal value of (1000000000/3.2).
  - `JitterAdjust` = 0x10000, which sets the nanosecond field to 0x1 and the fractional nanosecond field to 0x0.
- Wander—you can use the `WanderTimeLSB`, `WanderTimeMSB`, and `WanderAdjust` registers to achieve large time scale correction.

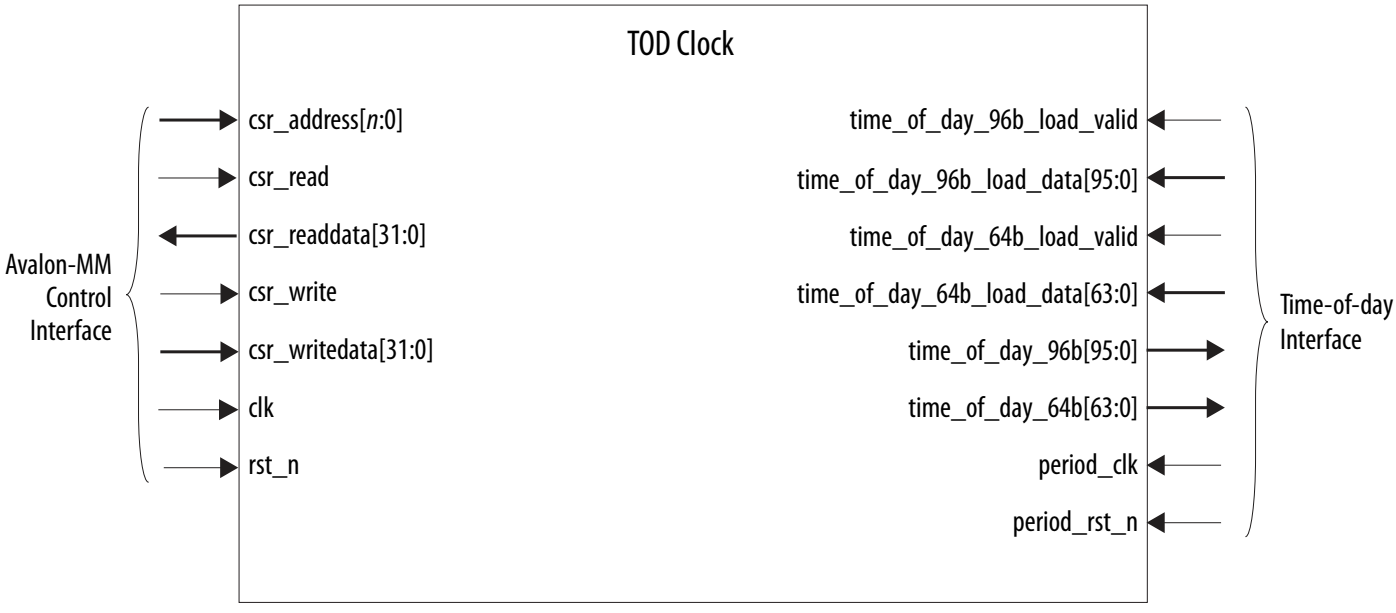
  The wander adjustment can either be a positive or negative adjustment per unit time. Wander adjustments are typically on larger time scales such as per hour. For very low values of the wander such as 1 ns per 24 hours, the timer must be larger and the adjustment value must be smaller.

  For example, to achieve 1 ns correction every 24 hours in a clock domain of 3.2 ns period, configure the registers as follow:

  - `WanderTimerLSB[29:1]` = 0x2D68_B000
  - `WanderTimerMSB[15:0]` = 0x06239

Interface Signals

Figure 1-1: Interface Signals of ToD Clock



Avalon-MM Signals

Table 1-3: Avalon-MM Signals Description

Name	Direction	Width	Description
csr_address[]	In	*n*	Use this bus to specify the register address you want to read from or write to. By default, the width of this signal is 4. When the OFFSET_JITTER_WANDER_EN parameter is set to 1, the width of this signal is 5.
csr_read	In	1	Assert this signal to request a read.
csr_readdata[]	Out	32	Data read from the specified register.
csr_write	In	1	Assert this signal to request a write.
csr_writedata[]	In	32	Data to be written to the specified register.
clk	In	1	Clock for the Avalon-MM interface, whose frequency is not more than 100 MHz.
rst_n	In	1	Active-low reset signal for the clk domain. Synchronous to clk.

Time-of-day Signals

Table 1-4: Time-of-day Signals Description

Name	Direction	Width	Description
time_of_day_96[]	Out	96	96-bit time of day streamed by the ToD clock.
time_of_day_64[]	Out	64	64-bit time of day streamed by the ToD clock.
time_of_day_96b_load_valid	In	1	Assert this signal for one clock cycle to indicate that the time_of_day_96b_load_data[] bus is valid. It indicates that the 96-bit time of day is synchronized and loaded into the ToD clock.
time_of_day_64b_load_valid	In	1	Assert this signal for one clock cycle to indicate that the time_of_day_64b_load_data[] bus is valid. It indicates that the 64-bit time of day is synchronized and loaded into the ToD clock.
time_of_day_96b_load_data[]	In	96	96-bit time of day from the master ToD clock.
time_of_day_64b_load_data[]	In	64	64-bit time of day from the master ToD clock.
period_clk	In	1	Clock for the ToD clock. Ensure that this clock is in the same clock domain as the TX and RX time of day clock signals of the MAC IP.
period_rst_n	In	1	Active-low reset signal for the period_clk domain. Synchronous to period_clk.

Configuration Registers

Table 1-5: Register Description

Byte Offset	Name	Description	Access	HW Reset Value
0x00	SecondsH	The upper 16-bit second field of the 96-bit ToD. The value occupies bits 0 to 15. Bits 16 to 31 are not used. Read the timestamp registers in this sequence: NanoSec, SecondsL, and SecondsH. Write the timestamp registers in this sequence: SecondsH, SecondsL, and NanoSec. Reading the SecondsH, SecondsL, and NanoSec registers does not necessarily return the last values written to these registers.	RW	0x0
0x04	SecondsL	The lower 32-bit second field of the 96-bit ToD. To read from or write to the timestamp registers, refer to the guidelines provided in the SecondsH register description.	RW	0x0
0x08	NanoSec	The 32-bit nanosecond field of the 96-bit ToD. Loading this register with a value equal to or larger than a billion leads to an incorrect timestamp. To read from or write to timestamp registers, refer to the guidelines provided in the SecondsH register description.	RW	0x0
0x0C	Reserved	–	–	–
0x10	Period	The period for the frequency adjustment. • Bits [24:16]: The nanosecond field if the PERIOD_CLOCK_FREQUENCY parameter is set to 0. • Bits [19:16]: The nanosecond field if the PERIOD_CLOCK_FREQUENCY parameter is set to 1. • Bits [15:0]: The fractional nanosecond field. • The remaining bits are not used. The reset value of this register, *n*, is determined by the value of the DEFAULT_NSEC_PERIOD and DEFAULT_FNSEC_PERIOD parameters.	RW	*n*

Byte Offset	Name	Description	Access	HW Reset Value
0x14	AdjustPeriod	The period for the offset adjustment. • Bits [24:16]: The nanosecond field if the PERIOD_CLOCK_FREQUENCY parameter is set to 0. • Bits [19:16]: The nanosecond field if the PERIOD_CLOCK_FREQUENCY parameter is set to 1. • Bits [15:0]: The fractional nanosecond field. • The remaining bits are not used. The reset value of this register, *n*, is determined by the value of the DEFAULT_NSEC_PERIOD and DEFAULT_FNSEC_PERIOD parameters.	RW	*n*
0x18	AdjustCount	• Bits [31:20]: Not used. • Bits [19:0]: The number of clock cycles used during offset adjustment.	RW	0x0
0x1C	DriftAdjust	The value that the ToD clock uses to periodically adjust the time of day. • Bits [31:20]: Not used. • Bits [19:16]: The nanosecond field. • Bits [15:0]: The fractional nanosecond field.	RW	0x0
0x20	DriftAdjustRate	• Bit 31: The drift direction: 0 for addition and 1 for subtraction. • Bits [30:16]: Not used. • Bits [15:0]: The interval between drift adjustments in number of clock cycles. Writing a value other than 0 to this register triggers the drift adjustment.	RW	0x0
0x24	OffsetNS	• Bit 31: Not used. • Bit 30: The offset direction: 0 for addition and 1 for subtraction. • Bits [29:0]: The nanosecond field of the offset. Writing a value other than 0 to this register triggers the offset in the time of day.	RW	0x0
0x28	OffsetFNS	• Bits [31:16]: Not used. • Bits [15:0]: The fractional nanosecond field of the offset.	RW	0x0

Byte Offset	Name	Description	Access	HW Reset Value
0x30	JitterTimer	• Bit 31: Unused • Bit 30: The direction of the jitter adjustment: 0 for addition and 1 for subtraction. • Bits [29:0]: The timer value in number of clock cycles. Periodic jitter adjustment is disabled when this register is set to 0. Writing a value other than 0 to this register enables period jitter adjustment. Hence, write to this register last.	RW	0x0
0x34	JitterAdjust	• Bits [31:16]: The nanosecond field of the jitter adjustment. • Bits [15:0]: The fractional nanosecond field of the jitter adjustment.	RW	0x0
0x40	WanderTimerLSB	• Bit 31: Unused • Bit 30: The direction of the timer adjustment: 0 for addition and 1 for subtraction. • Bits [29:0]: The least significant byte of the timer in number of clock cycles. Writing a value other than 0 to this register enables wander timer adjustment. Hence, write to the WanderTimerLSB and WanderTimerMSB registers last.	RW	0x0
0x44	WanderTimerMSB	• Bits [31:16]: Unused. • Bits [15:0]: The most significant byte of the timer in number of clock cycles.	RW	0x0
0x48	WanderAdjust	• Bits [31:16]: The nanosecond field of the wander adjustment. • Bits [15:0]: The fractional nanosecond field of the wander adjustment.	RW	0x0

Time-of-day Synchronizer 2

2016.05.02

UG-20019

The Time-of-day (ToD) Synchronizer provides an accurate synchronization between the time of day of a master ToD clock and a slave ToD clock. This component can synchronize the following combination of master and slave ToD clocks:

- Master and slave ToD clocks that operate at the same frequency, between 125 MHz and 390.625 MHz. The synchronizer also supports different clock phases and PPM.
- Master and slave ToD clocks that operate at different frequencies: 62.5 MHz, 125 MHz, 156.25 MHz, 312.5 MHz, or 390.625 MHz.

Supported Devices

- Arria V GX/GT/GZ/SX/ST
- Arria 10 GX/GT/SX
- Cyclone V SE/SX/ST
- MAX 10
- Stratix V GX/GT

Configuring the ToD Synchronizer

In the Quartus Prime software, instantiate the ToD Synchronizer by selecting it from the IP Catalog or Qsys (**Interface Protocols > Ethernet > Reference Design Components**). Specify the following parameters.

Table 2-1: ToD Synchronizer Parameters

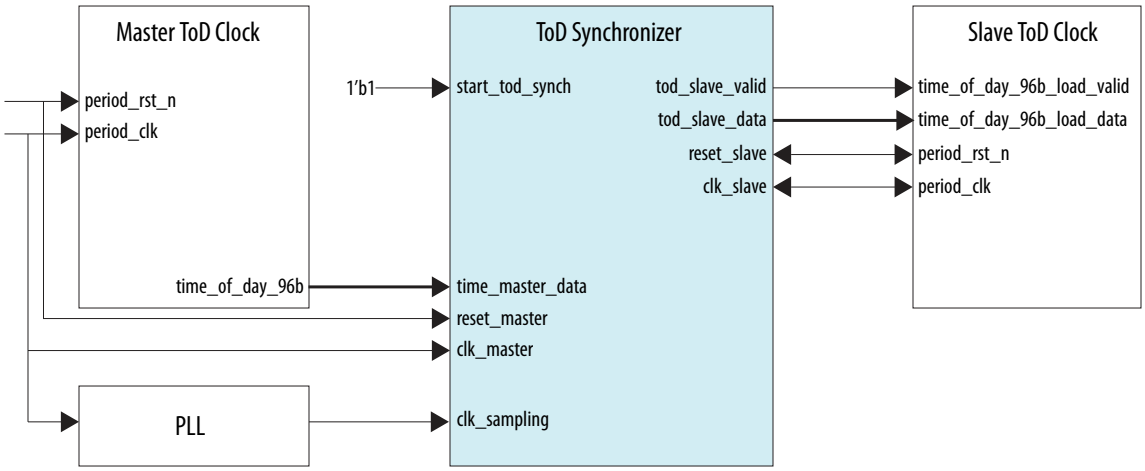
Name	Valid Values	Description
TOD_MODE	0, 1	Specifies the format of the time of day. • 0: 64 bits. 48 bits nanosecond and 16 bits fractional nanosecond. • 1: 96 bits. 48 bits seconds, 32 bits nanosecond and 16 bits fractional nanosecond. The default value is 1.

Name	Valid Values	Description
SYNC_MODE	0 – 15	Specifies the synchronization type between the master and slave ToD clocks. • 0: 125-MHz master ToD clock and 156.25 MHz slave ToD clock. • 1: 156.25-MHz master ToD clock and 125-MHz slave ToD clock. • 2: The frequencies of the master and slave ToD clocks are the same, between 125 MHz and 390.625 MHz. This synchronization type supports different phase or PPM. Ensure that you also specify the period of the master and slave clocks using the PERIOD_NSEC and PERIOD_FNSEC parameters. • 3: 156.25-MHz master ToD clock and 312.5-MHz slave ToD clock. • 4: 312.5-MHz master ToD clock and 156.25-MHz slave ToD clock. • 5: 125-MHz master ToD clock and 312.5-MHz slave ToD clock. • 6: 312.5-MHz master ToD clock and 125-MHz slave ToD clock. • 7: 125-MHz master ToD clock and 390.625-MHz slave ToD clock. • 8: 390.625-MHz master ToD clock and 125-MHz slave ToD clock. • 9: 156.25-MHz master ToD clock and 390.625-MHz slave ToD clock. • 10: 390.625-MHz master ToD clock and 156.25-MHz slave ToD clock. • 11: 312.5-MHz master ToD clock and 390.625-MHz slave ToD clock. • 12: 390.625-MHz master ToD clock and 312.5-MHz slave ToD clock. • 13: 125-MHz master ToD clock and 62.5-MHz slave ToD clock. • 14: 156.25-MHz master ToD clock and 62.5-MHz slave ToD clock. • 15: 312.5-MHz master ToD clock and 62.5-MHz slave ToD clock. The default value is 1.
PERIOD_NSEC	0 – 4'hF	Specifies the respective 4-bit nanosecond field for the reset value for the following clock frequencies: • 125 MHz: Set this parameter to 4'h8 for 8 ns. • 156.25 MHz: Set this parameter to 4'h6 for 6.4 ns. This value is the default value. • 312.5 MHz: Set this parameter to 4'h3 for 3.2 ns. • 390.625 MHz: Set this parameter to 4'h2 for 2.56 ns

Name	Valid Values	Description
PERIOD_FNSEC	0 – 16h'FFFF	Specifies the respective 16-bit fractional nanosecond field for the reset value for the following clock frequencies: • 125 MHz: Set this parameter to 16'h0 for 8 ns. • 156.25 MHz: Set this parameter to 16'h6666 for 6.4 ns. This value is the default value. • 312.5 MHz: Set this parameter to 16'h3333 for 3.2 ns. • 390.625 MHz: Set this parameter to 16'h8F5C for 2.56 ns
SAMPLE_SIZE	64, 128, or 256	Specifies the number of samples to use in calculating the FIFO buffer's fill level. More samples results in a more accurate estimation of the fill level. However, the calculation time increases with the number of samples. The default value is 64.

Using the ToD Synchronizer

Figure 2-1: ToD Synchronizer in a Design



The ToD synchronizer uses a dual-clock FIFO buffer to receive the time of day from the master ToD clock and transmits it to the slave ToD clock. To ensure that the synchronization is accurate, the transfer latency must be taken into consideration. The sampling clock (`clk_sampling`) samples the fill level of the FIFO buffer and calculates the latency. Derive this clock signal from the same source as the master ToD clock or the slave ToD clock using a PLL.

Sampling Clock Frequency

To achieve the recommended frequency for the sampling clock, follow these steps:

1. The SYNC_MODE and SAMPLE_SIZE parameters determine the sampling clock factor, which is then used to determine the required PLL settings. Use the Table 2-2 to identify the sampling clock factor for your configuration.
2. Use the sampling clock factor identified in the previous step to determine the PLL settings. Table 2-3 lists the settings for Stratix V Altera PLL.

Table 2-2: Sampling Clock Factor

SYNC_MODE	Reference Clock Frequency (MHz)	Sampling Clock Factor		
		SAMPLE_SIZE = 64	SAMPLE_SIZE = 128	SAMPLE_SIZE = 256
0, 1	125	16/63	32/33	64/63
	156.25	64/315	128/155	256/375
2	Master/slave frequency	64/63	128/153	256/375
3, 4	156.25	64/63	128/153	256/375
	312.5	32/33	64/63	128/153
5, 6	125	32/63	64/63	128/63
	312.5	64/315	128/155	256/375
7, 8	125	–	–	32/13
	390.625	–	–	256/375
9, 10	156.25	32/33	64/31	128/63
	390.625	64/155	128/185	256/375
11, 12	312.5	16/15	32/33	64/63
	390.625	64/75	128/185	256/253
13	62.5	64/63	128/153	256/253
	125	32/33	64/63	128/153
14	62.5	32/33	64/63	128/153
	156.25	64/155	128/155	256/375
15	62.5	64/63	128/153	256/253
	312.5	64/155	128/155	256/375

Table 2-3: PLL Settings for Stratix V

Sampling Clock Factor	PLL Counter		
	M	N	C
16/15	16	5	3

Sampling Clock Factor	PLL Counter		
	M	N	C
16/63	16	3	21
32/13	32	13	1
32/33	32	3	11
	32	11	3
32/63	32	3	21
64/31	64	31	1
64/63	64	9	7
	64	21	3
64/75	64	25	3
64/155	64	31	5
64/315	64	21	15
128/63	128	21	3
128/153	128	51	3
	128	17	9
	128	9	17
128/155	128	31	5
128/185	128	37	5
256/253	256	11	23
256/375	256	75	5
	256	25	15

Interface Signals

Figure 2-2: Interface Signals of ToD Synchronizer

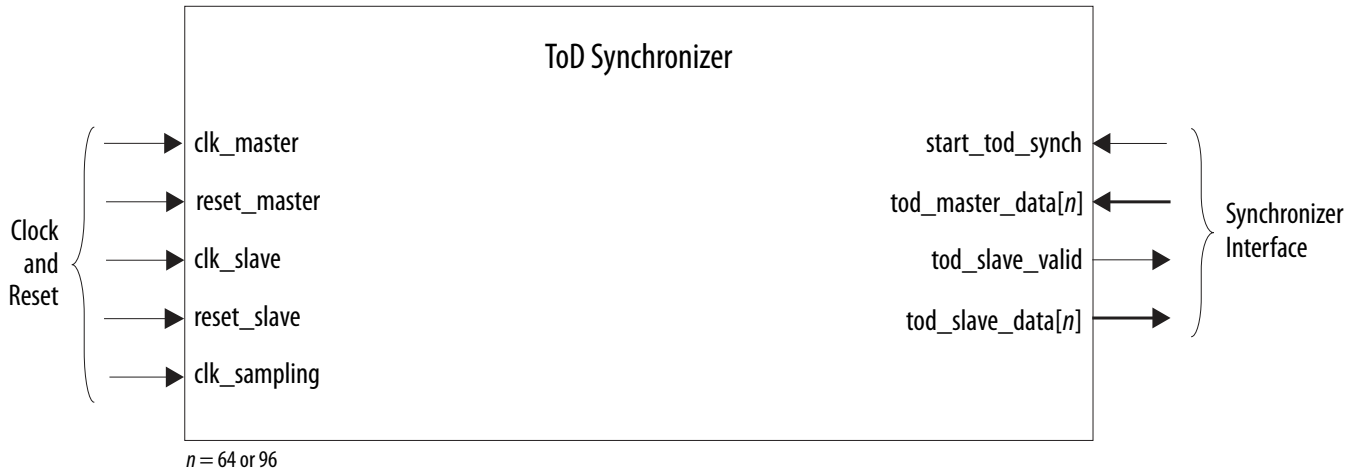


Table 2-4: Signals Description

Name	Direction	Width	Description
Clock and Reset Signals			
clk_master	In	1	Master ToD clock domain.
reset_master	In	1	Synchronous reset signal in the master ToD clock domain.
clk_slave	In	1	Slave ToD clock domain.
reset_slave	In	1	Synchronous reset signal in the slave ToD clock domain.
clk_sampling	In	1	Sampling clock to measure the transfer latency.
Interface Signals			
start_tod_sync	In	1	Assert this signal to start the synchronization process. Synchronization continues as long as this signal is asserted.
tod_master_data[]	In	64 or 96	Carries the 64-bit or 96-bit time of day from the master ToD clock. The width of this signal is determined by the TOD_MODE parameter
tod_slave_valid	Out	1	When asserted, the signal indicates that the data on the tod_data_slave bus is valid and ready for transfer in the following cycle. This signal stays asserted for only 1 clock cycle.

Name	Direction	Width	Description
tod_slave_data[]	Out	64 or 96	Carries the 64-bit or 96-bit time of day for the slave ToD clock. This time of day is synchronized to the master ToD clock with an additional one clock cycle because it takes one clock cycle to transfer the time of day to the slave ToD clock. The width of this signal is determined by the TOD_MODE parameter.

Packet Classifier 3

2016.05.02

UG-20019

The Packet Classifier decodes the packet type of incoming PTP packets, and returns the decoded information to the MAC IP core. The decoded information is aligned to the start of packet of the corresponding PTP packet.

Supported Devices

- Arria V GX/GT/GZ/SX/ST
- Arria 10 GX/GT/SX
- Cyclone V SE/SX/ST
- MAX 10
- Stratix V GX/GT

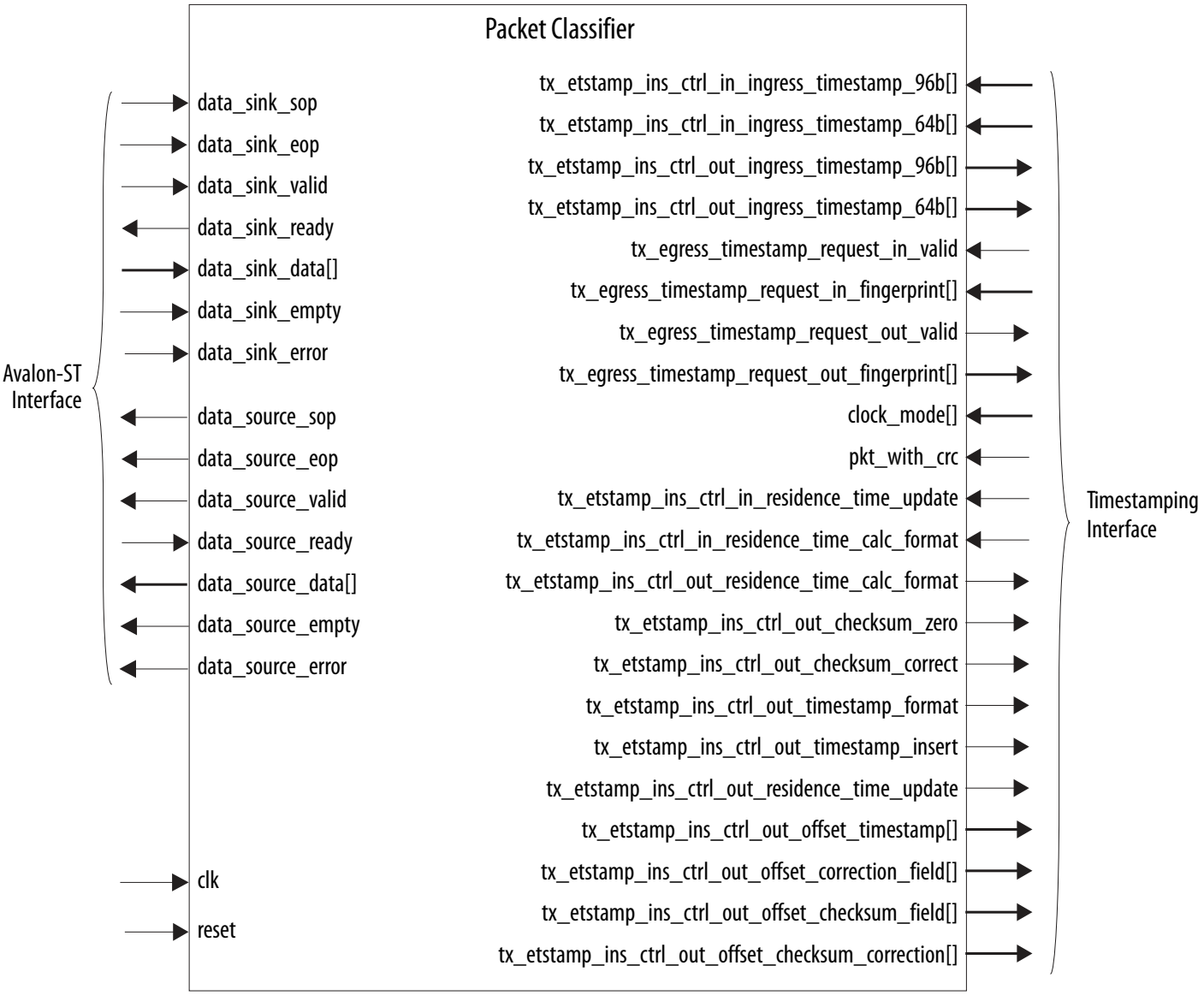
Configuring the Packet Classifier

In the Quartus Prime software, instantiate the Packet Classifier by selecting it from the IP Catalog or Qsys (**Interface Protocols > Ethernet > Reference Design Components**). Specify the parameters in the following table.

Table 3-1: Parameters Description

Name	Value	Default	Description
TSTAMP_FP_WIDTH	1 – 32	4	The width of the timestamp fingerprint.
SYMBOLSPERBEAT	1, 4, or 8	8	The number of symbols transferred in a clock cycle.
BITSPERSYMBOL	8	8	The number of bits per symbol transferred in a clock cycle.

Interface Signals



Clock and Reset Signals

Table 3-2: Clock and Reset Signals Description

Name	Direction	Width	Description
clk	In	1	Reference clock for the packet classifier. Connect this signal to the MAC TX clock.
reset	In	1	Synchronous reset signal for the packet classifier.

Avalon-ST Interface Signals

Table 3-3: Avalon-ST Signals Description

Name	Direction	Width	Description
data_sink_sop	In	1	Assert this signal to indicate the beginning of the packet.
data_sink_eop	In	1	Assert this signal to indicate the end of the packet.
data_sink_valid	In	1	Assert this signal to indicate that the data_sink_data[] signal and other signals on this interface are valid.
data_sink_ready	Out	1	When asserted, this signal indicates that the packet classifier is ready to accept data.
data_sink_data[]	In	*n*(1)	The input packet.
data_sink_empty[]	In	2, 3	Use this signal to specify the number of empty bytes in the cycle that contain the end of packet. The width of this signal is 2 when the SYMBOLSPERBEAT parameter is 4; 3 when the parameter is 8. This signal does not exist when the SYMBOLSPERBEAT is 1.
data_sink_error	In	1	Assert this signal to indicate that the current input packet contains errors.
data_src_sop	Out	1	When asserted, this signal indicates the beginning of the packet.
data_src_eop	Out	1	When asserted, this signal indicates the end of the packet.
data_src_valid	Out	1	When asserted, this signal indicates that the data_src_data[] signal and other signals on this interface are valid.
data_src_ready	In	1	Assert this signal when the receiving component is ready to accept data.
data_src_data[]	Out	*n*(1)	The output data.
data_src_empty[]	Out	2, 3	Contains the number of empty bytes in the cycle that contain the end of packet. The width of this signal is 2 when the SYMBOLSPERBEAT parameter is 4; 3 when the parameter is 8. This signal does not exist when the SYMBOLSPERBEAT is 1.
data_src_error	Out	1	When asserted, this signal indicates that the current output packet contains errors.

(1) *n*=SYMBOLSPERBEAT*BITSPERSYMBOL

Control Signals

Table 3-4: Control Signals Description

Name	Direction	Width	Description
tx_etstamp_ins_ctrl_in_ingress_timestamp_96b	In	96	The 96-bit timestamp received from the MAC RX, aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_in_ingress_timestamp_64b	In	64	The 64-bit timestamp received from the MAC RX, aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_out_ingress_timestamp_96b	Out	96	The 96-bit timestamp to the MAC TX, aligned to the corresponding output PTP packet.
tx_etstamp_ins_ctrl_out_ingress_timestamp_64b	Out	64	The 64-bit timestamp to the MAC TX, aligned to the start of packet of the corresponding output PTP packet.
tx_egress_timestamp_request_in_valid	In	1	Assert this signal to indicate that a timestamp is required for the packet. This signal must align to the start of an input packet.
tx_egress_timestamp_request_in_fingerprint	In	TSTAMP_FP_WIDTH	The timestamp's fingerprint for the input packet.
tx_egress_timestamp_request_out_valid	Out	1	Assert this signal when timestamp is required for the particular frame. This signal is aligned to the start of packet of the corresponding output PTP packet.
tx_egress_timestamp_request_out_fingerprint	Out	TSTAMP_FP_WIDTH	The timestamp's fingerprint for the output packet.
clock mode	In	2	Specify the clock mode: • 00: Ordinary clock • 01: Boundary clock • 10: End to end transparent clock • 11: Peer to peer transparent clock
pkt_with_crc	In	1	Use this signal to indicate whether or not the incoming packet contains 4-byte CRC field. • 0: the incoming packet contains the CRC field. • 1: the incoming packet does not contain the CRC field.

Name	Direction	Width	Description
tx_etstamp_ins_ctrl_in_residence_time_calc_format	In	1	Use the following values to specify the format of the timestamp to use when calculating the residence time. • 0: 96-bit timestamp • 1: 64-bit timestamp Align this signal to the start of the input packet.
tx_etstamp_ins_ctrl_out_residence_time_calc_format	Out	1	The format of the timestamp used to calculate the residence time. • 0: 96-bit timestamp • 1: 64-bit timestamp This signal is aligned to the start of the output packet.
tx_etstamp_ins_ctrl_out_checksum_zero	Out	1	When asserted, indicates that the checksum field of the PTP packet is set to zero. This signal is aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_out_checksum_correct	Out	1	When asserted, indicates that the checksum of the PTP packet is corrected by updating the checksum correction offset. This signal is aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_out_timestamp_format	Out	1	The format of the timestamp. • 0: 1588v2 format. 96-bit timestamp that consists of 48-bit second field, 32-bit nanosecond field, and 16-bit correction field for fractional nanosecond. • 1: 1588v1 format. 64-bit timestamp that consists of 32-bit second field and 32-bit nanosecond field. This signal is aligned to the start of packet of the corresponding output PTP packet.
tx_etstamp_ins_ctrl_out_timestamp_insert	Out	1	When asserted, indicates that an egress timestamp must be inserted into the corresponding PTP packet. This signal is aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_out_residence_time_update	Out	1	When asserted, indicates that the residence time is added to the correction field of the PTP packet. This signal is aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_out_offset_timestamp	Out	16	The location of the timestamp field, relative to the first byte of the packet.

Name	Direction	Width	Description
tx_etstamp_ins_ctrl_out_offset_correction_field	Out	16	The location of the correction field, relative to the first byte of the packet.
tx_etstamp_ins_ctrl_out_offset_checksum_field	Out	16	The location of the checksum field, relative to the first byte of the packet.
tx_etstamp_ins_ctrl_out_offset_checksum_correction	Out	16	The location of the checksum correction field, relative to the first byte of the packet.

Revision History

A

2016.05.02

UG-20019 Subscribe Send Feedback

Date	Version	Changes
May 2016	2016.05.02	Initial release