Ethernet Design Example Components User Guide

Updated for Intel® Quartus® Prime Design Suite: **21.1**

IP Version: **19.2.0**

Subscribe

Send Feedback

UG-20019 | 2021.03.29

Latest document on the web: **PDF** | **HTML**

Contents

1. Time-of-day Clock

The Time-of-day (TOD) Clock streams 96-bit and 64-bit time-of-day to one or more timestamping units in an IEEE 1588v2 solution. The time-of-day consist of the following fields.

Field	96-bit Timestamp Format	64-bit Timestamp Format
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.

You can instantiate the TOD clock through the **Ethernet IEEE 1588 Time of Day Clock Intel® FPGA IP** in the Intel Quartus® Prime software.

1.1. Release Information

Intel FPGA IP versions match the Intel Quartus Prime Design Suite software versions until v19.1. Starting in Intel Quartus Prime Design Suite software version 19.2, Intel FPGA IP has a new versioning scheme.

The Intel FPGA IP version (X.Y.Z) number can change with each Intel Quartus Prime software version. A change in:

- X indicates a major revision of the IP. If you update the Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 1. Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP Release Information

Item	Description
IP Version	19.2.0
Intel Quartus Prime Version	21.1
Release Date	2021.03.29
Supported Devices	• Arria® V GX/GT/GZ/SX/ST • Intel Arria 10 GX/GT/SX • Cyclone® V SE/SX/ST • Intel MAX® 10

continued...

Item	Description
	• Stratix® V GX/GT • Intel Stratix 10 • Intel Cyclone 10 GX • Intel Agilex™

Related Information

Ethernet Design Example Components Release Notes
Describes changes to the IP in a particular release.

1.2. Resource Utilization

Table 2. 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 = 0 OFFSET_JITTER_WANDER_EN = 1	1782	2406	2297	0
PERIOD_CLOCK_FREQUENCY = 1 OFFSET_JITTER_WANDER_EN = 0	764	736	1224	0

Table 3. Estimated Resource Utilization in Intel Arria 10 Devices (10AX115U2F45I2SGES)

Configuration	ALMs	Combinational ALUTs	Logic Registers	Memory (M20K Blocks)
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 0	716	741	942	0
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 1	1961	2735	2530	0
PERIOD_CLOCK_FREQUENCY = 1 OFFSET_JITTER_WANDER_EN = 0	689	734	1233	0

Table 4. Estimated Resource Utilization in Intel Stratix 10 Devices (1SG280LN2F43E1VG)

Configuration	ALMs	LUTs	Logic Registers	Memory
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 0	609	838	1312	0
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 1	1633	2553	2401	0
PERIOD_CLOCK_FREQUENCY = 1 OFFSET_JITTER_WANDER_EN = 0	897	896	2567	0

Table 5. Estimated Resource Utilization in Intel Agilex Devices (AGFA014F25AA212V)

Configuration	ALMs	LUT	Logic Registers	Memory
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 0	621	838	1312	0
PERIOD_CLOCK_FREQUENCY = 0 OFFSET_JITTER_WANDER_EN = 1	1629	2610	2532	0
PERIOD_CLOCK_FREQUENCY = 1 OFFSET_JITTER_WANDER_EN = 0	894	912	2571	0

1.3. Configuring the TOD Clock

In the Intel Quartus Prime software, instantiate the TOD clock by selecting **Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP** from the IP Catalog or Platform Designer (`Interface Protocols > Ethernet > Reference Design Components`). Specify the following parameters.

Table 6. TOD Clock Parameters Description

Name	Value	Default Value	Description
Enable high clock frequency mode (`PERIOD_CLOCK_FREQUENCY`)	On or Off	On	Turn off this parameter 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. Turn on this parameter if the MAC connected to the TOD clock requires high period clock frequency, such as Low-latency 10G Ethernet, 25G Ethernet, or 40G/100G Ethernet MAC. For this setting, the nanosecond field in the `Period` and `AdjustPeriod` registers is 4 bits wide.
Enable offset, jitter, and wander supports (`OFFSET_JITTER_WANDER_EN`)	On or Off	Off	Turn on this parameter to enable the offset, jitter, and wander timers. This parameter is available only when high clock frequency mode is disabled (`PERIOD_CLOCK_FREQUENCY= 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.

1.4. Using the TOD Clock

Follow these guidelines when using the TOD clock:

- 96-bit timestamp format—load the time-of-day 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 time-of-day 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, Intel 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 timestamp format—load the time-of-day 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, Intel recommends that you add one clock cycle to the value of the incoming time-of-day to accommodate the latency.
- The TOD clock does not synchronize the 96-bit and 64-bit timestamp format.
- The drift, jitter, and wander timers restart each time a new time-of-day is loaded, either through the signal or configuration registers.

1.4.1. 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 6.4 ns for a 10G Ethernet application. The hexadecimal representation of this value is 0x6 ns and 0x6666.4 fns.

- Fractional nanosecond field is 16 bits wide: 0.4 fns = 0.4 * 2^{16} = 26214.4 in decimal.
- Converting to hexadecimal: 26214 + 0.4 = 0x6666 + 0x0000.4 = 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 953.6 ns 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.

1.4.2. Adjusting Offset, Jitter, and Wander

The TOD clock supports several types of adjustments:

- Offset—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—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—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[29:0]` = 0x2D68_B000

  — `WanderTimerMSB[15:0]` = 0x6239

Note: The adjustments are available only when `PERIOD_CLOCK_FREQUENCY` is set to 0.

1.4.3. Correcting TOD Offset

You can use the `AdjustPeriod` and `AdjustCount` registers to correct TOD offset between master and slave TOD. `AdjustPeriod` register value is used as effective period value when `AdjustCount` register is a non-zero value. Write to these registers in sequence: AdjustPeriod register, followed by AdjustCount register. Adjustment happens whenever there is a write to the AdjustCount register.

For example, if the slave TOD is faster than master TOD by 'a' ns, to correct the TOD offset of 'a' ns by increasing the value of 'a' for a TOD clock running at 125 MHz (`Period` register value is 0x8 ns):

- set `AdjustPeriod` to '8+b' ns
- set `AdjustCount` to 'c'

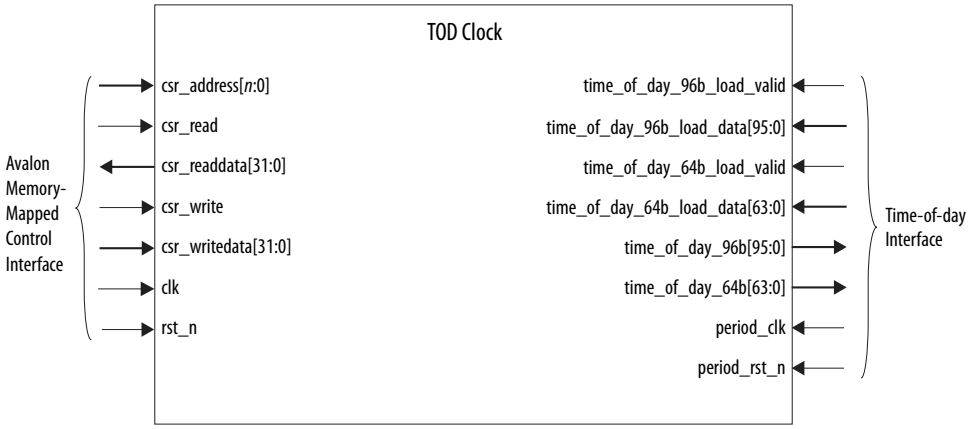
where 'a = b * c' is fulfilled.

By setting a = 16 ns, b = 2 ns, and c = 8, the logic produces TOD with effective period value of 10 ns (slower) in the next 8 clock cycles. After 8 clock cycles, the logic resumes the normal operation where effective period value = 8 ns.

Similarly, if the slave TOD is slower than master TOD, to correct negative TOD offset, set `AdjustPeriod` to '<Period register value> – b'.

1.5. Interface Signals

Figure 1. Interface Signals of TOD Clock



1.5.1. Avalon Memory-Mapped Signals

Table 7. Avalon Memory-Mapped 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® memory-mapped interface, whose frequency is not more than 100 MHz.
`rst_n`	In	1	Active-low reset signal for the `clk` domain. Synchronous to `clk`.

1.5.2. Time-of-day Signals

Table 8. 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.

continued...

Name	Direction	Width	Description
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 clock signals of the MAC IP.
period_rst_n	In	1	Active-low reset signal for the period_clk domain. Synchronous to period_clk.

1.6. Configuration Registers

Table 9. 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 TOD registers in this sequence: NanoSec, SecondsL, and SecondsH. 96-bit TOD is snapshot whenever the NanoSec register is read. Write the TOD 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
0x01	SecondsL	The lower 32-bit second field of the 96-bit TOD. To read from or write to the TOD registers, refer to the guidelines provided in the SecondsH register description.	RW	0x0
0x02	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 TOD registers, refer to the guidelines provided in the SecondsH register description.	RW	0x0
0x03	Reserved	–	–	–
0x04	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.	RW	*n*

continued...

Byte Offset	Name	Description	Access	HW Reset Value
		The reset value of this register, *n*, is determined by the value of the DEFAULT_NSEC_PERIOD and DEFAULT_FNSEC_PERIOD parameters.		
0x05	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_ADJPERIOD and DEFAULT_FNSEC_ADJPERIOD parameters. For offset adjustment, write to AdjustPeriod register followed by AdjustCount register. The TOD offset adjustment starts after the AdjustCount register is written.	RW	*n*
0x06	AdjustCount	• Bits [31:20]: Not used. • Bits [19:0]: The number of clock cycles used during offset adjustment. For offset adjustment, write to AdjustPeriod register followed by AdjustCount register. The TOD offset adjustment starts after the AdjustCount register is written.	RW	0x0
0x07	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
0x08	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
0x09	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
0x0A	OffsetFNS	• Bits [31:16]: Not used. • Bits [15:0]: The fractional nanosecond field of the offset.	RW	0x0
0x0C	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.	RW	0x0

continued...

Byte Offset	Name	Description	Access	HW Reset Value
		Writing a value other than 0 to this register enables period jitter adjustment. Hence, write to this register last.		
0x0D	`JitterAdjust`	• Bits [31:16]: The nanosecond field of the jitter adjustment. • Bits [15:0]: The fractional nanosecond field of the jitter adjustment.	RW	0x0
0x10	`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
0x11	`WanderTimerMSB`	• Bits [31:16]: Unused. • Bits [15:0]: The most significant byte of the timer in number of clock cycles.	RW	0x0
0x12	`WanderAdjust`	• Bits [31:16]: The nanosecond field of the wander adjustment. • Bits [15:0]: The fractional nanosecond field of the wander adjustment.	RW	0x0

2. Time-of-day Synchronizer

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, 390.625 MHz, or 402.83 MHz.

You can instantiate the TOD synchronizer through the **Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP** in the Intel Quartus Prime software.

2.1. Release Information

Intel FPGA IP versions match the Intel Quartus Prime Design Suite software versions until v19.1. Starting in Intel Quartus Prime Design Suite software version 19.2, Intel FPGA IP has a new versioning scheme.

The Intel FPGA IP version (X.Y.Z) number can change with each Intel Quartus Prime software version. A change in:

- X indicates a major revision of the IP. If you update the Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 10. Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP Release Information

Item	Description
IP Version	19.2.0
Intel Quartus Prime Version	21.1
Release Date	2021.03.29
Supported Devices	• Arria V GX/GT/GZ/SX/ST • Intel Arria 10 GX/GT/SX • Cyclone V SE/SX/ST • Intel MAX 10 • Stratix V GX/GT • Intel Stratix 10 • Intel Cyclone 10 GX • Intel Agilex

Related Information

Ethernet Design Example Components Release Notes
Describes changes to the IP in a particular release.

2.2. Resource Utilization

Table 11. Estimated Resource Utilization in Intel Arria 10 Devices (10AX115U2F45I2SGES)

Configuration	ALMs	Combinational ALUTs	Logic Registers	Memory (M20K Blocks)
Default Mode	563	636	1678	3

Table 12. Estimated Resource Utilization in Intel Stratix 10 Devices (1SX280LN2F43E2LG)

Configuration	ALUs	Combinational LUT	Logic Registers	Memory
Default Mode	763	957	2123	3

Table 13. Estimated Resource Utilization in Intel Agilex Devices (AGFB014F25A2E2V)

Configuration	ALUs	Combinational LUT	Logic Registers	Memory
Default Mode	672	958	2105	3

2.3. Configuring the TOD Synchronizer

In the Intel Quartus Prime software, instantiate the TOD Synchronizer by selecting **Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP** from the IP Catalog or Platform Designer (`Interface Protocols > Ethernet > Reference Design Components`). Specify the following parameters.

Table 14. TOD Synchronizer Parameters Description

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.
`SYNC_MODE`	0 – 18	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 402.83 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.

continued...

Name	Valid Values	Description
		• 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. • 16: Reserved. • 17: Reserved. • 18: 125-MHz master TOD clock and 402.83-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. • 402.83 MHz: Set this parameter to 4’h2 for 2.482 ns. This parameter is only applicable for SYNC_MODE = 2.
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. • 402.83 MHz: Set this parameter to 16’h7B80 for 2.482 ns. This parameter is only applicable for SYNC_MODE = 2.
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. This parameter is not applicable for SYNC_MODE = 18.

2.4. Using the TOD Synchronizer

Figure 2. TOD Synchronizer in a Design (SYNC_MODE 0 to 15)

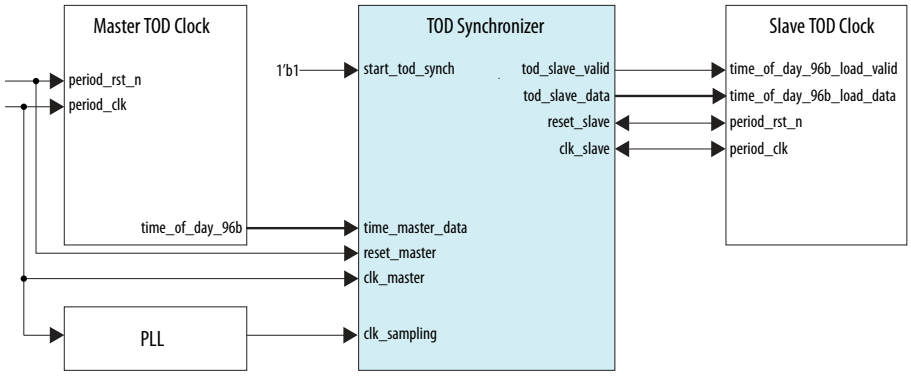
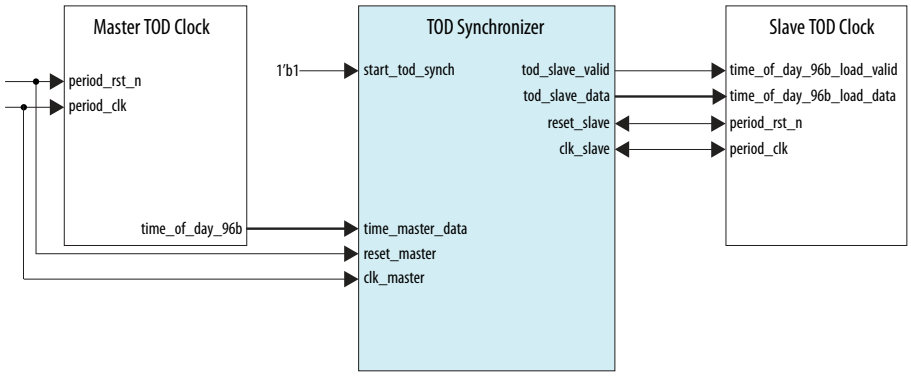


Figure 3. TOD Synchronizer in a Design (SYNC_MODE 18)



The TOD synchronizer with SYNC_MODE = 0 to 15 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.

The sampling clock (`clk_sampling`) is not required for TOD synchronizer with SYNC_MODE = 18 as it uses a different technique for synchronization.

2.4.1. 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 15 on page 16 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 18 on page 18 lists the settings for Stratix V PLL Intel FPGA IP.
3. In the case where the sampling clock factor is not achievable using the current PLL, Intel recommends that you use an alternative sampling clock frequency of 80 MHz.
4. Sampling clock granularity determines the accuracy of synchronization where smaller granularity gives better accuracy. Table 16 on page 17 and Table 17 on page 17 show the granularity for each recommended sampling clock.

Table 15. 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	N/A	N/A	32/13
	390.625	N/A	N/A	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 16. Sampling Clock Granularity for Sampling Clock Factor

The following is the sampling clock granularity for sampling clock factor in Table 15 on page 16.

SYNC_MODE	Reference Clock Frequency (MHz)	Sampling Clock Granularity (ns)		
		SAMPLE_SIZE = 64	SAMPLE_SIZE = 128	SAMPLE_SIZE = 256
0, 1	125	0.5	0.25	0.125
	156.25			
2	Master/slave frequency	Ref clock period/64	Ref clock period/128	Ref clock period/256
3, 4	156.25	0.1	0.05	0.025
	312.5			
5, 6	125	0.25	0.125	0.0625
	312.5			
7, 8	125	N/A	N/A	0.25
	390.625			
9, 10	156.25	0.2	0.1	0.05
	390.625			
11, 12	312.5	0.2	0.1	0.05
	390.625			
13	62.5	0.25	0.125	0.0625
	125			
14	62.5	0.5	0.25	0.125
	156.25			
15	62.5	0.25	0.125	0.0625
	312.5			

Table 17. Sampling Clock Granularity Using 80 MHz Sampling Clock

SYNC_MODE	Time of Day Frequency (MHz)	Sampling Clock Granularity (ns)		
		SAMPLE_SIZE = 64	SAMPLE_SIZE = 128	SAMPLE_SIZE = 256
0, 1	125	0.5	0.5	0.5
	156.25			
2	62.5	0.5	0.5	0.5
	125	0.5	0.5	0.5
	156.25	0.1	0.1	0.1
	312.5	0.1	0.1	0.1
	390.625	N/A	0.02	0.02
	402.83	N/A	N/A	N/A
3, 4	156.25	0.1	0.1	0.1
	312.5			

continued...

SYNC_MODE	Time of Day Frequency (MHz)	Sampling Clock Granularity (ns)		
		SAMPLE_SIZE = 64	SAMPLE_SIZE = 128	SAMPLE_SIZE = 256
5, 6	125	0.5	0.5	0.5
	312.5			
7, 8	125	N/A	0.5	0.5
	390.625			
9, 10	156.25	N/A	0.1	0.1
	390.625			
11, 12	312.5	N/A	0.1	0.1
	390.625			
13	62.5	0.5	0.5	0.5
	125			
14	62.5	0.5	0.5	0.5
	156.25			
15	62.5	0.5	0.5	0.5
	312.5			

Table 18. PLL Settings for Stratix V Devices

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

continued...

Sampling Clock Factor	PLL Counter		
	M	N	C
128/185	128	37	5
256/253	256	11	23
256/375	256	75	5
	256	25	15

2.5. Interface Signals

Figure 4. Interface Signals of TOD Synchronizer

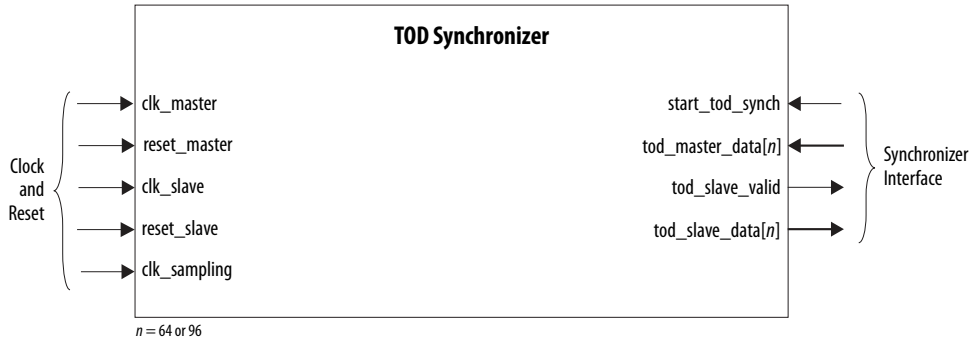


Table 19. Signals Description

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. Not available for SYNC_MODE = 18.
Interface Signals			
start_tod_sync	In	1	Assert this signal to start the synchronization process. Synchronization continues as long as this signal is asserted.

continued...

Name	Direction	Width	Description
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.
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.

3. Packet Classifier

The Packet Classifier decodes the packet type of incoming PTP packets, Avalon streaming PTP packets from client interface to the TX MAC, and returns the decoded information to the MAC IP core. The decoded information includes:

- Timestamp request and fingerprint (2-step clock)
- Timestamp insert (1-step clock)
- Timestamp format (96-bit or 64-bit)
- Residence time update, TX ingress timestamp, and format
- PTP packet header information (location of timestamp field, correction field, checksum field, and checksum correction field)

The decoded information is aligned to the start of packet of the corresponding PTP packet.

You can instantiate the packet classifier through the **Ethernet Packet Classifier Intel FPGA IP** in the Intel Quartus Prime software.

3.1. Release Information

Intel FPGA IP versions match the Intel Quartus Prime Design Suite software versions until v19.1. Starting in Intel Quartus Prime Design Suite software version 19.2, Intel FPGA IP has a new versioning scheme.

The Intel FPGA IP version (X.Y.Z) number can change with each Intel Quartus Prime software version. A change in:

- X indicates a major revision of the IP. If you update the Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 20. Ethernet Packet Classifier Intel FPGA IP Release Information

Item	Description
IP Version	19.2.0
Intel Quartus Prime Version	21.1
Release Date	2021.03.29
Supported Devices	• Arria V GX/GT/GZ/SX/ST • Intel Arria 10 GX/GT/SX • Cyclone V SE/SX/ST • Intel MAX 10 • Stratix V GX/GT • Intel Stratix 10

Related Information

Ethernet Design Example Components Release Notes
Describes changes to the IP in a particular release.

3.2. Resource Utilization

Table 21. Estimated Resource Utilization in Intel Arria 10 Devices (10AX115U2F45I2SGES)

Configuration	ALMs	Combinational ALUTs	Logic Registers	Memory (M20K Blocks)
Default Mode	212	297	179	8

Table 22. Estimated Resource Utilization in Intel Stratix 10 Devices (1SX280LN2F43E2LG)

Configuration	ALMs	Combinational ALUTs	Logic Registers	Memory (M20K Blocks)
Default Mode	292	332	533	9

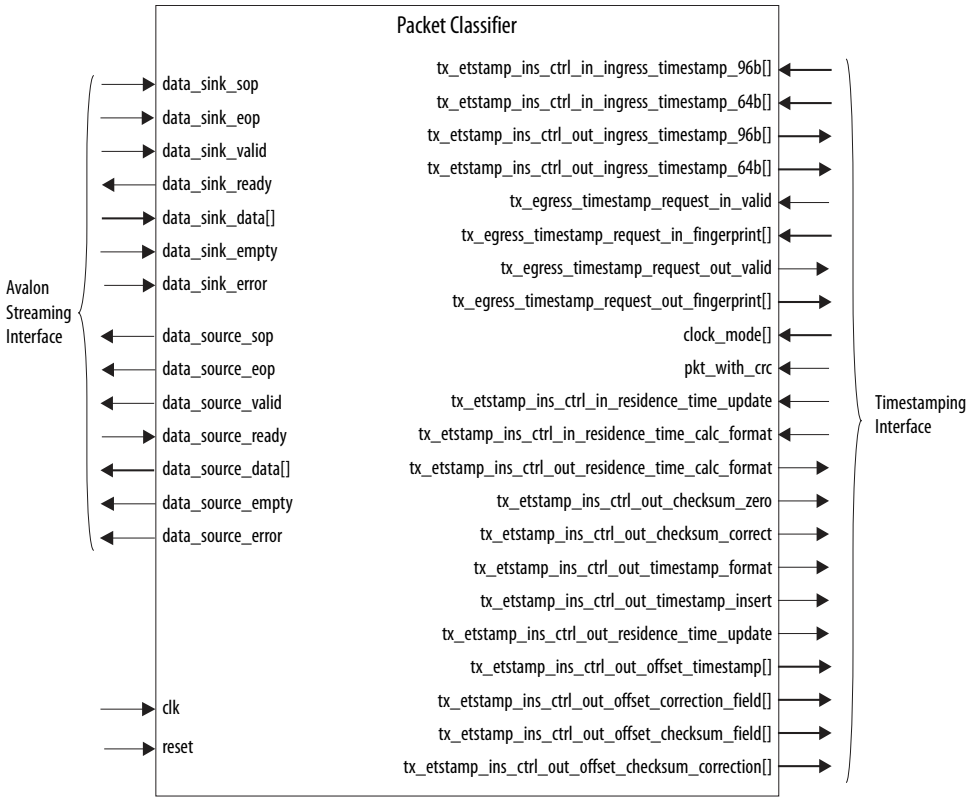
3.3. Configuring the Packet Classifier

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

Table 23. Packet Classifier 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.

3.4. Interface Signals



3.4.1. Clock and Reset Signals

Table 24. 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.

3.4.2. Avalon Streaming Interface Signals

Table 25. Avalon Streaming 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.

continued...

Name	Direction	Width	Description
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.

3.4.3. Control Signals

Table 26. Control Signals Description

Name	Direction	Width	Description
tx_etstamp_ins_ctrl_in_ingress_timestamp_96b	In	96	The 96-bit ingress timestamp of the TX Packet, aligned to the start of packet of the corresponding PTP packet.
tx_etstamp_ins_ctrl_in_ingress_timestamp_64b	In	64	The 64-bit ingress timestamp of the TX Packet, 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.

continued...

(1) n=SYMBOLSPERBEAT*BITSPERSYMBOL

Name	Direction	Width	Description
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.
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.

continued...

Name	Direction	Width	Description
			• 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.
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.

4. Ethernet Design Example Components User Guide Archives

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

If an IP core version is not listed, the user guide for the previous IP core version applies.

Intel Quartus Prime Version	IP Core Version	User Guide
20.4	19.2.0	Ethernet Design Example Components User Guide
20.2	19.2.0	Ethernet Design Example Components User Guide
19.3	19.2.0	Ethernet Design Example Components User Guide
18.0	18.0	Ethernet Design Example Components User Guide
16.0	16.0	Ethernet Design Example Components User Guide

5. Document Revision History for the Ethernet Design Example Components User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.03.29	21.1	19.2.0	• Added new tables: — *Sampling Clock Granularity for Sampling Clock Factor* — *Sampling Clock Granularity Using 80 MHz Sampling Clock*
2020.12.14	20.4	19.2.0	• Updated chapter *Time-of-day Synchronizer* and table *TOD Synchronizer Parameters Description* to include frequency 402.83 MHz and also its parameters. • Added figure *TOD Synchronizer in a Design (SYNC_MODE 18)* to section *Using the TOD Synchronizer*. • Updated the description for `clk_sampling` in table *Signals Description*. • Updated the names, values, default values, and descriptions for the following parameters in Table: *TOD Clock Parameters Description*: — Updated parameter name `PERIOD_CLOCK_FREQUENCY` to **Enable high clock frequency mode**. — Updated parameter name `OFFSET_JITTER_WANDER_EN` to **Enable offset, jitter, and wander supports**. • Updated the description for `AdjustPeriod` in Table: *Configuration Registers* for Time-of-day clock.
2020.07.14	20.2	19.2.0	Updated *Adjusting Offset, Jitter, and Wander*: • Corrected the bits for `WanderTimerLSB` from [29:1] to [29:0]. • Corrected the values for `WanderTimerMSB[15:0]` from 0x06239 to 0x6239.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.06.22	20.2	19.2.0	• Added the following tables in the *Time-of-day Clock* chapter: — *Estimated Resource Utilization in Intel Stratix 10 Devices (1SG280LN2F43E1VG)*. — *Estimated Resource Utilization in Intel Agilex Devices (AGFA014F25AA212V)*. • Added the following tables in the *Time-of-day Synchronizer* chapter: — *Estimated Resource Utilization in Intel Stratix 10 Devices (1SX280LN2F43E2LG)* — *Estimated Resource Utilization in Intel Agilex Devices (AGFB014F25A2E2V)* • Updated the description for `PERIOD_CLOCK_FREQUENCY` in Table: *TOD Clock Parameters Description*. • Updated the following topics: — *Adjusting TOD Drift* — *Adjusting Offset, Jitter, and Wander* — *Correcting TOD Offset* — *Packet Classifier* • Updated the description for the following registers in Table: *Register Description*: — `SecondsH` — `AdjustPeriod` — `AdjustCount` • Added a new Table: *Estimated Resource Utilization in Intel Stratix 10 Devices (1SX280LN2F43E2LG)*. • Updated the descriptions to the following signals in Table: *Control Signals Description*: — `tx_etstamp_ins_ctrl_in_ingress_timestamp_96b` — `tx_etstamp_ins_ctrl_in_ingress_timestamp_64b` • Updated for latest Intel branding standards.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.09.30	19.3	19.2.0	• Added support for Intel Agilex devices in the *Time-of-day Clock* and *Time-of-day Synchronizer* chapters. • Updated the following tables to include Intel Quartus Prime version: — *Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP Release Information* — *Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP Release Information* — *Ethernet Packet Classifier Intel FPGA IP Release Information*
2019.07.01	19.2	19.2.0	• Renamed the *Supported Devices* sections to *Release Information* in the respective chapters to include IP release information as well as supported devices. • Added support for Intel Stratix 10 and Intel Cyclone 10 GX devices in the *Time-of-day Clock* and *Time-of-day Synchronizer* chapters. • Added support for Intel Stratix 10 devices in the *Packet Classifier* chapter.
2018.12.03	18.0	18.0	• Rebranded as Intel. • Renamed the following Ethernet design example component names as per Intel rebranding: — "Ethernet IEEE 1588 TOD Synchronizer" to "Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP" — "Ethernet IEEE 1588 Time of Day Clock" to "Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP" — "Ethernet Packet Classifier" to "Ethernet Packet Classifier Intel FPGA IP" • Updated the following topics: — *Time-of-day Clock* — *Using the TOD Clock* • Updated Table: *Register Description* to correct the byte offsets of `SecondsL`, `NanoSec`, Reserved, `Period`, `AdjustPeriod`, `AdjustCount`, `DriftAdjust`, `DriftAdjustRate`, `OffsetNS`, `OffsetFNS`, `JitterTimer`, `JitterAdjust`, `WanderTimerLSB`, and `WanderAdjust` registers. • Made editorial updates throughout the document.

Date	Version	Changes
March 2017	2017.03.08	• Added a new topic: Correcting ToD Offset. • Updated the "Time-of-day Signals Description" table. • Added resource utilization for Arria 10 Devices for 16.1 release: — "Time-of-day Clock" section — "Time-of-day Synchronizer" section — "Packet Classifier" section
May 2016	2016.05.02	Initial release.