White Paper

Altera® SoC FPGAs
Intel® PTP Servo Solutions

altera™
An Intel Company

Intel® Precision Time Protocol Servo (Intel® PTP Servo) Solution for Time Synchronization Applications

Precision Time Protocol (PTP) is used to provide timing for an increasing number of applications and markets.

Authors

Kishan Shenoi
Principal Engineer

Hans Brandberg
SoC Design Engineer

Markos Papadonikolakis
SoC Design Engineer

Jeff Hockert
Sr. IP Technical Marketing Manager

Bret Gustafson
Strategic Business Developer
Intel Corp.

"IEEE 1588 is designed to fill a niche not well served by either of the two dominant protocols, Network Time Protocol (NTP) and Global Positioning System (GPS). IEEE 1588 is designed for local systems requiring accuracies beyond those attainable using NTP. It is also designed for applications that cannot bear the cost of a GPS receiver at each node, or for which GPS signals are inaccessible."[9]

John Eidson (one of the founders of PTP/1588)

Table of Contents

Executive summary 1
What Is Time, What Time Is It, and Why Do We Care? 2
NTP and PTP 2
O-RAN and PTP 4
The S-Plane in O-RANs 5
O-RAN and the ITU-T 7
The Intel PTP Solution 8
Intel PTP Servo implemented on Intel Agilex® 7 SoC FPGAs 9
Intel PTP Servo Performance Benchmarks 10
Intel PTP Servo Value Proposition . 10
Conclusion 10
Learn More 11
References 11

Executive Summary

To coordinate the actions of disparate electronic systems potentially dispersed around the globe, all these systems must be synchronized in time. Such systems include distributed databases, financial transactions, monitoring and control, and sensor fusion to name a few.

A predominant example is that of packet-based communication networks, such as radio access networks (RANs), which require devices to be synchronized to function correctly and to provide advertised services. Furthermore, many other applications rely on underlying communications infrastructure to satisfy their own synchronization requirements.

Establishing and measuring the level of simultaneity or the temporal ordering of events requires those events to be timestamped. In turn, these timestamps must be associated with a common timescale. The problem is ensuring that all devices forming the network are synchronized to a common "grandmaster clock" with sufficient accuracy and precision.

Early synchronization solutions employed software timestamping techniques that could synchronize clocks with millisecond-level accuracy. However, many of today's high-end applications demand higher accuracy. The Precision Time Protocol (PTP), which employs hardware timestamping of packets and can achieve sub-microsecond accuracy, was introduced to address the needs of these applications. Standardized by the IEEE, PTP is commonly known as IEEE 1588* or just "1588."

In the context of these discussions, the term "servo" refers to a software implementation of a suite of algorithms running on a processor. PTP servos are used to synchronize the clocks in devices throughout the network. Many PTP solutions are based on open-source implementations of PTP on LinuxPTP* suite, also known by the name of its PTP engine as ptp4l.

Traditional implementations utilizing LinuxPTP employ open-source servos available with ptp4l, but these servos are inadequate for many real-world network deployments. To address this issue, Intel has developed a proprietary PTP servo that can be used in all cases. The Intel PTP Servo software can be run on Intel Xeon® CPU-based motherboards, Intel® SoC FPGAs, and network interface cards (NICs) with an external digital clock synthesizer (DCS) and 1588 support.

This paper gives an introduction into time synchronization using PTP and discusses the roles and relationships between the IEEE, the Open RAN Alliance (O-RAN Alliance), and the International Telecommunication Union (ITU), including the fact that O-RAN specifications are linked to several ITU-T recommendations and IEEE specifications.

It also presents an example hardware implementation of the Intel PTP solution using an Intel SoC FPGA as part of an O-RAN application. should read: Hardware timestamping is performed by the Ethernet PHY/MAC functions in the FPGA's transceiver chiplets. Also reported is the efficacy of the Intel PTP Servo relative to an open-source servo, as determined by performance benchmarks.

What Is Time, What Time Is It, and Why Do We Care?

Time is difficult to define. As Saint Augustine of Hippo (A.D. 354-430) famously said: "What, then, is time? If no one ask of me, I know, but if wish to explain to him who asks, I know not."[1]

In the first book of The Hitchhiker's Guide to the Galaxy by Douglas Adams, one of the characters notes: "Time is an illusion. Lunchtime doubly so." Although this was intended as an "off-the-cuff" remark, many contemporary physicists agree that time may well be illusionary. Today, most physicists no longer consider time to be an independent property or quality and argue that there really isn't such a thing as space that contains things and there isn't really such a thing as time during which events occur.[2,3]

At the macro level in which humans exist, of course, time is experienced as a subjective reality. Following Einstein's publication of the general theory of relativity in 1915, it's now understood that time can speed up and slow down in the presence of cosmological bodies with different masses and velocities. This property is exemplified by global navigation satellite system (GNSS) constellations in which the clocks on satellites need to be constantly corrected to account for the relativistic effects of their orbits around the Earth.

Our need to know the time and our sociological and technological requirements have evolved. As technology grew, so did people's need to know the time more accurately. Railroads provide a classic example. In England, up until the latter part of the 18th century, time was normally determined in each town by a local sundial. As a result, the time in a town or village could differ by as much as 20 minutes from the time in London. This wasn't a problem when traveling between towns and cities on foot or by horse-drawn carriage could take days or weeks. However, it did become problematic with the introduction of railroads circa the 1820s and 1830s, when even relatively small discrepancies in time could cause confusion, disruption, and accidents.

Commensurate with advances in timekeeping technologies is the need for an agreed-upon standard. Today, one such entity is known as International Atomic Time (TAI), which is maintained by the BIPM (Bureau International des Poids et Mesures [French], a.k.a. International Bureau of Weights and Measures [English]). TAI is a continuous timescale based on a weighted average of the time kept by over 450 atomic clocks across 80 national laboratories worldwide. Although it is based on TAI, Coordinated Universal Time (UTC) is a discontinuous timescale that is adjusted by leap seconds to account for the difference between the definition of the second and the rotation of the earth around the sun.

Coordination of actions between disparate entities requires all entities involved to be synchronized in time. Even though this property is applicable in a wide range of industries and applications, it is most evident in communication networks. Such networks require devices to be synchronized to function correctly and to provide advertised services. Furthermore, many applications utilize communications networks, relying on the underlying communications fabric as their backbone to establish any required synchronization. As a result, the need for synchronization is becoming ubiquitous.

Almost every application that needs to establish or measure the level of simultaneity or the temporal ordering of events will timestamp those events. Such timestamps must be aligned for events occurring in geographically disparate locations around the globe. These applications will fail miserably if the timestamping is not associated with a common timescale. It is not uncommon for such applications to rely on their underlying communications networks for this purpose, although other approaches, such as obtaining UTC-traceable time "from the sky" using GNSS receivers, may also be employed. There are many such application examples, including distributed databases, multi-media, shared documents, stock trades, sensor fusion, multi-player gaming, and monitoring, to name a few.

NTP and PTP

The fundamental concepts underlying packet switching networks were first proposed in the 1960s and deployed in the 1970s. In the early days, various techniques were employed to synchronize clocks in the network elements. Circa 1985, the NTP, which follows the UTC time standard, was introduced to achieve clock synchronization between computer systems over packet-switched, variable-latency data networks.

NTP, which is still in use today, is usually implemented using software timestamping of packets and can synchronize clocks with millisecond-level accuracy. NTP is a robust method that can be used even over the general internet. However, millisecond accuracy is insufficient for many of today's high-end applications, including industrial measurement and control systems, financial transactions, sub-sea acoustic arrays, and 5G/6G RANs. PTP, which employs hardware timestamping of packets and can achieve accuracy in the sub-microsecond range, is a natural choice for these applications.

The first version of PTP, standardized by the IEEE and commonly known as IEEE 1588 or just "1588," was published in 2002. The second version was published in 2008, and the most recent version was published in 2019. IEEE 1588 includes a profile concept defining PTP operating parameters and options. Several profiles have been developed for various applications by the corresponding standardization bodies, including telecommunications by ITU, electric power distribution by IEEE/IEC and audiovisual implementations by SMPTE. PTP can potentially synchronize multiple clocks to nanosecond accuracy on networks that employ this

technology. PTP follows the TAI timescale while maintaining a leap-second correction to provide UTC.

The IEEE 1588 standard defines the hierarchical master-subordinate architecture requirements for clock distribution. An amendment to the current PTP standard called IEEE1588g-2022 advocates the use of "timeTransmitter" for master and "timeReceiver" for subordinate, but this has not been widely adopted at the time of this writing.

A simplified hierarchical example for telecommunication networks is shown in Figure 1. The primary clock, insofar as the network is concerned, is the telecom grandmaster clock (T-GM). The time inside the T-GM is derived from an accurate source such as a GNSS receiver. The T-GM can drive multiple telecom boundary clocks (T-BCs), and each T-BC can drive multiple telecom time subordinate clocks (T-TSCs). A T-BC may have two aspects: it is subordinate to the T-GM and a master to its T-TSCs.

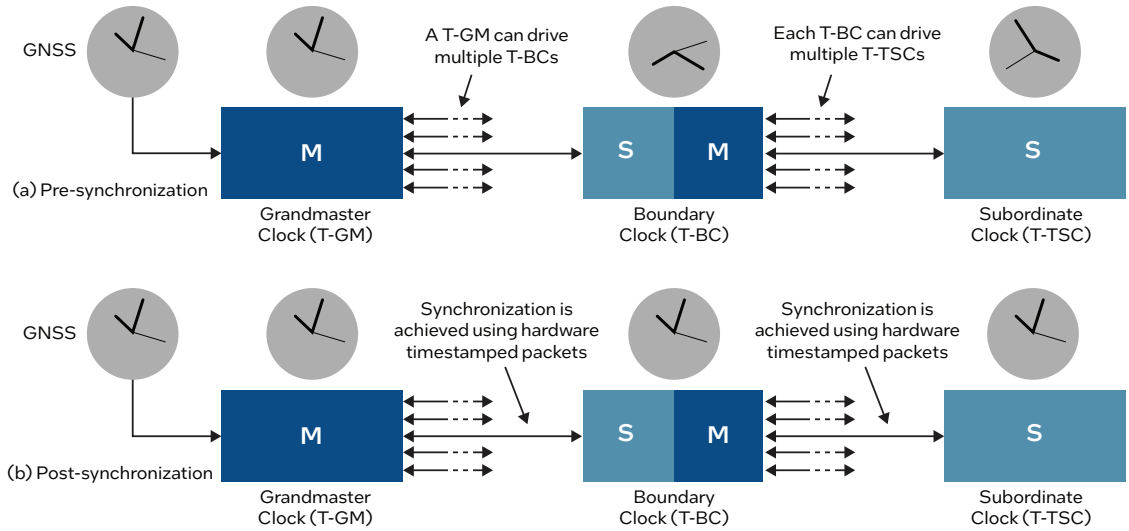

Figure 1. High-level representation of a simplified PTP hierarchy.

There may be many more layers to the hierarchy than are shown here, such as T-BCs driving T-BCs, but these will be omitted from this paper for simplicity. Furthermore, telecom transparent clocks (T-TCs) are not shown here, which are devices that don't contain clocks themselves, but instead update the timestamps on packets as they pass through the device with appropriate transport delays associated with the device.

Between any master and subordinate, a network of switches and/or routers can exist. In PTP for telecom parlance, the network may provide full-timing support (FTS) or partial-timing support (PTS). In the FTS case, whereby all intervening network elements are PTP-aware, minimal packet delay variation is experienced by the timing packets. By comparison, packet delay variation can be substantial in a PTS implementation, wherein one or more devices are PTP-unaware.

A master and its subordinate both contain digital clock synthesizer (DCS) and time-of-day (ToD) functions that they employ to keep track of the current time. The ideal goal is to have the ToD stored in a subordinate exactly match the ToD stored in its master.

Both NTP and PTP achieve synchronization through the exchange of timestamped packets. To achieve the highest possible levels of accuracy, all packets should be timestamped as physically close as possible to the interface between the network device and the network channel. The simplest exchange involving three packets is depicted using PTP terminology in Figure 2.

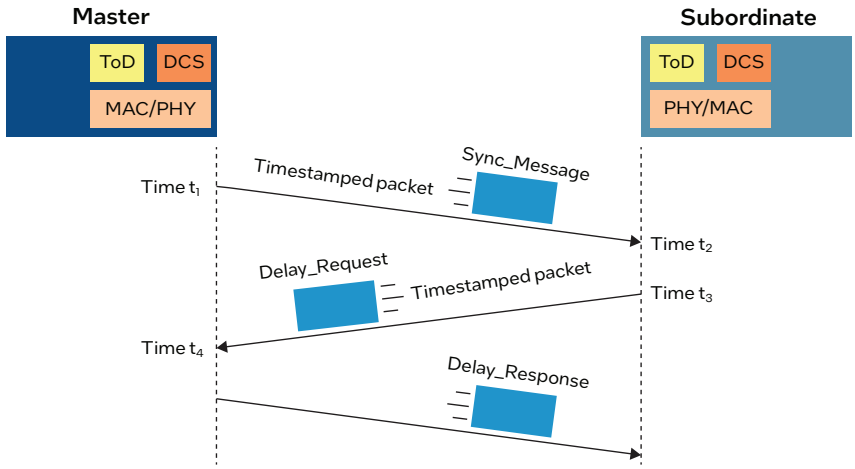

Figure 2. High-level visualization of PTP packet exchange between master and subordinate.

At time t_1, the master initiates the exchange by transmitting a Sync_Message packet to the subordinate. This packet contains a timestamp of t_1 representing the time-of-departure of the packet from the master. When the subordinate receives this packet, it augments the original timestamp with its own timestamp of t_2, representing the arrival time. The value ($t_2 – t_1$) is the sum of both the master-subordinate transmission delay (ΔMS) and the subordinate's time offset from the master (OFM). The OFM is the "error" in the subordinate's clock.

In the past, primarily due to limitations in the hardware, it was challenging to embed the t_1 timestamp in the outgoing Sync_Message packet on the fly. To overcome this problem, PTP allows for a "two-step" mode whereby the precise value of t_1 is conveyed in a Followup_Message (not shown here), transmitted after the Sync_Message. Recent advances in hardware – for example, the F-Tile transceivers in Intel SoC FPGAs – mean the t_1 timestamp can be embedded in the Sync_Message packet on the fly. In PTP parlance, this is referred to as a "one-step" mode.

The subordinate regularly sends a Delay_Request packet back to the master. In addition to the original t_1 and t_2 timestamps, the subordinate stores a timestamp t_3 representing the time-of-departure of the Delay_Request packet from the subordinate. When the master receives this packet, it records as t_4 the timestamp representing the time-of-arrival. The master responds by sending a Delay_Response packet to the subordinate. This packet contains t_4, and thus the subordinate now has all four timestamps (t_1, t_2, t_3, and t_4). The value ($t_4 – t_3$) is recognized as the difference between the subordinate-master transmission delay (ΔSM) and the OFM.

Assuming the transmission delay is symmetric (i.e., ΔSM = ΔMS), then the subordinate can compute its OFM from these four values as well as the one-way delay, or time-of-flight (ToF), of the time-stamped packets.

This is not a "one-and-done" process. For example, since this is a packet-based network, there is no fixed delay path between the master and the subordinate, which means different packets can take different routes and traverse different devices, thereby taking different durations to traverse the network. Even with a fixed route, the packets may encounter queues in routers and switches that introduce packet delay variation. There's also the drift problem over time in the subordinate's local oscillator. However, it's possible to achieve sub-microsecond synchronization by continually exchanging packets and employing sophisticated algorithms running in the subordinate.

Other aspects of the network may also change over time. Existing devices may be replaced, new devices may be added, and legacy devices may be removed. Furthermore, the current trend to virtualizing network functions adds an additional layer of complexity. For all these reasons, timing synchronization is an ongoing process.

One final point is that people often confuse synchronization with delay and latency. In many systems, such as 5G radio networks, there's a desire to minimize delays and latencies between two communication endpoints, but this has nothing to do with synchronization per se. The only goal of synchronization is to ensure that the value of the ToD clock in any subordinate matches the value of the ToD clock in that subordinate's master, and this is true from the lowest level subordinate up the hierarchy to the grandmaster clock.

O-RAN and PTP

Since the demand for mobile communications is growing dramatically concerning the number of users and the amount of data each user consumes and generates, our focus will now turn to PTP implementations in the context of RANs.

The Open RAN Alliance or O-RAN Alliance[4] is a worldwide community of mobile network operators, vendors, and research and academic institutions that is dedicated to evolving radio access networks into being smart and open.

As connectivity demands and the number of connected devices are growing, so is the necessity for a flexible RAN architecture. An Open RAN (O-RAN) is a non-proprietary implementation of a RAN that allows interoperability between cellular network equipment provided by different vendors. As the O-RAN Alliance defines the typical 5G RAN, the base station is split into three logical nodes—the O-RAN radio unit (O-RU), the O-RAN distributed unit (O-DU), and the O-RAN central unit (O-CU). This is depicted in Figure 3.

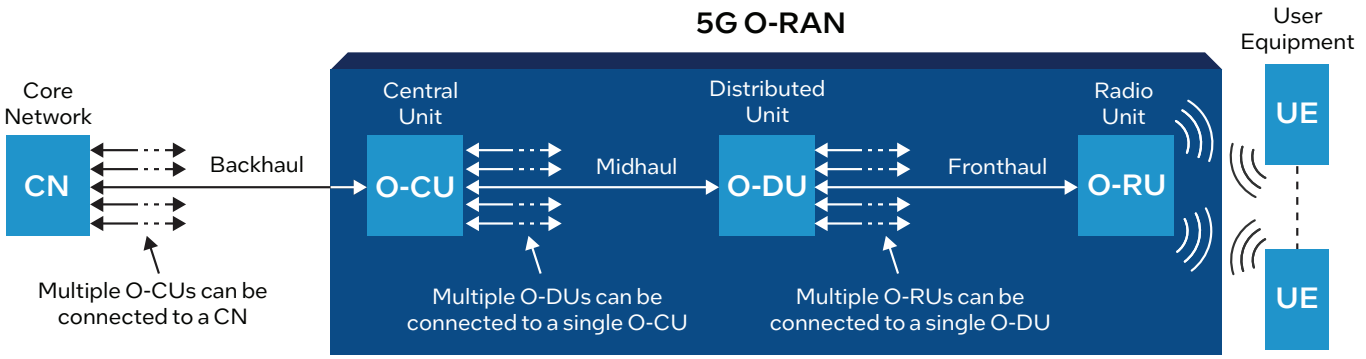

Figure 3. High-level visualization of a 5G O-RAN.

Simply put, O-RUs are where the conversion between digital and analog RF takes place; O-DUs are where much of the real-time signal processing is implemented; and O-CUs are where the timely, but not quite real-time, processing such as call-control is performed.

The fronthaul interface between an RU and a DU, typically implemented using an Ethernet network, is the most bandwidth-intensive and time-sensitive portion of a 5G O-RAN. In the case of pre-5G telecommunications standards, timing synchronization was often treated as an afterthought and effectively "shoehorned" in after the fact. By comparison, in addition to the management plane (M-plane), the control plane (C-plane), and user plane (U-plane), the O-RAN specification for the fronthaul interface also includes a synchronization plane (S-plane), which embraces the collection of requirements about timing and synchronization.

The S-Plane in O-RANs

Delivering the high data rates promised by 5G necessitates an architecture whereby multiple network stations communicate simultaneously with the user equipment (UE). To complete this complex task, sophisticated signal processing is involved. In practice, combining the signals from two radio units (RUs) is effective only if the RUs involved in the collaborative exercise are tightly synchronized.[7]

O-RAN specifications recognize that timing is a critical part of wireless systems. The S-plane addresses these requirements. Of special importance is the timing that is implicit at the very edge of the network, which is—in wireless systems—the antenna of the RU. The timing properties are measured for specificity at a hypothetical reference point corresponding to the air interface. The electrical signal in the antenna element(s) is converted to an analog radio frequency (RF) signal and vice versa. It is quite common to have an electrical test point in the RU that embodies the signal's necessary timing and synchronization properties at the air interface. The conventional timing reference is a 1 pulse-per-second (1PPS) signal, whose electrical and physical characteristics are defined in the ITU-T G.703 recommendation.

The 3rd Generation Partnership Project (3GPP)[5] is an umbrella term for a group of standards organizations developing mobile telecommunications protocols. The O-RAN Alliance has adopted several key timing requirements from the 3GPP related to time alignment error (TAE). As discussed in 3GPP TS 38.133 and 3GPP TS 38.104, these requirements state that the relative time error (i.e., the TAE) between two antennas of a cluster must be less than 130 ns; furthermore, that the error between any two antennas in general, must be less than 3 µs. It is common to consider the time at the network's core, essentially a primary reference time clock (PRTC), as the time reference for the subtending wireless network. It is also common practice to allocate one-half of the TAE allocation to each path between two endpoints and the common source.

To ensure interoperability among O-RU vendors, O-RAN characterizes the timing synchronization capabilities – in the form of maximum absolute time error (|TE|) – of an ORAN O-RU in a way that is compliant with both classes defined in the enhanced Common Public Radio Interface (eCPRI) and IEEE 802.1CM standards. Based on these standards, O-RAN specifies two O-RU timing synchronization classes: a regular O-RU with |TE| ≤ 80 ns and an enhanced O-RU with |TE| ≤ 35 ns (the O-RU time error accumulates as other contributors are added to the network). These two classes correspond to Cases 1.1 and 1.2 (Section 6.4.1 of IEEE 802.1CM), respectively, where the PTP T-TSC clock is integrated in the enhanced Radio Equipment (eRE).

With respect to the O-RAN specifications[8] (Table H.2, ORAN-WG4.CUS.0-v07.00.01), an enhanced O-RU can be deployed in networks of Categories A, B, and C as defined in Section 4.2 of Requirements for the eCPRI Transport Network V1.2. Hence, targeting and meeting the timing requirements of a Class C T-TSC, per ITU-T G.8273.2, is adequate for O-RU deployment in any Category A, B, or C network. This is illustrated in Figure 4.

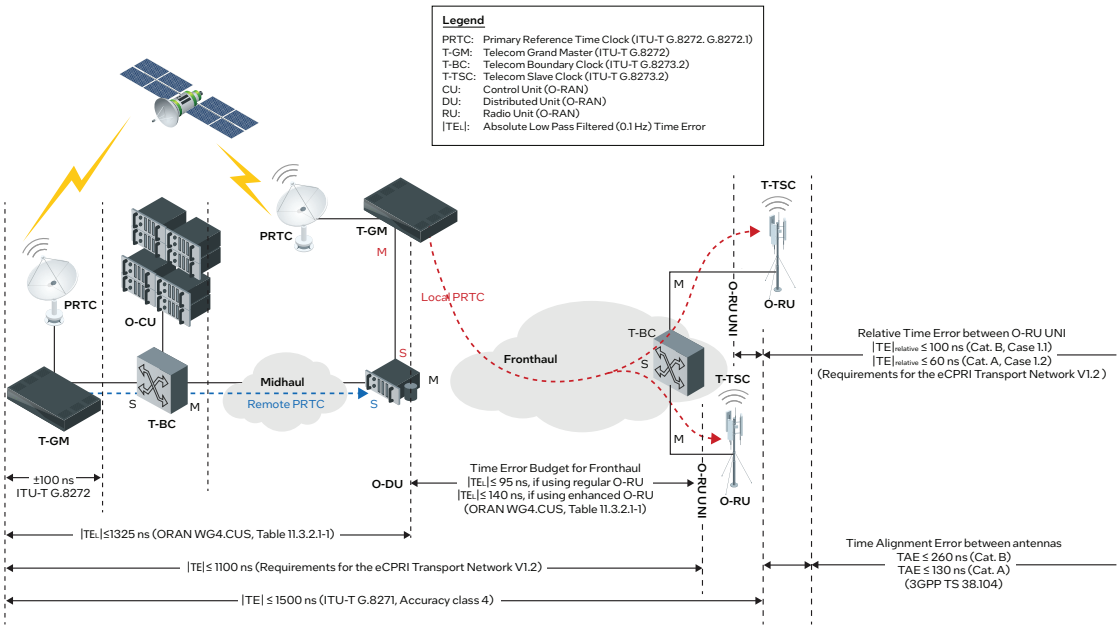

Figure 4. TAE requirements in O-RAN architectures.

In addition to adhering to 5G standards, many RANs must support legacy 4G infrastructure. For example, 4G radios typically employ the CPRI for their fronthaul transport, whereas 5G radios typically use eCPRI. In cases where the network must accommodate both 4G and 5G infrastructure, a device called a fronthaul gateway (FHGW) is often used to perform CPRI to eCPRI conversions, including the lower L1 signal processing, which include fast Fourier transform (FFT) and Physical Random Access Channel (PRACH) functions.

With respect to synchronization in the fronthaul, the O-RAN Alliance has identified four low-level split (LLS) configurations called LLS-C1, LLS-C2, LLS-C3, and LLS-C4. This is illustrated in Figure 5.

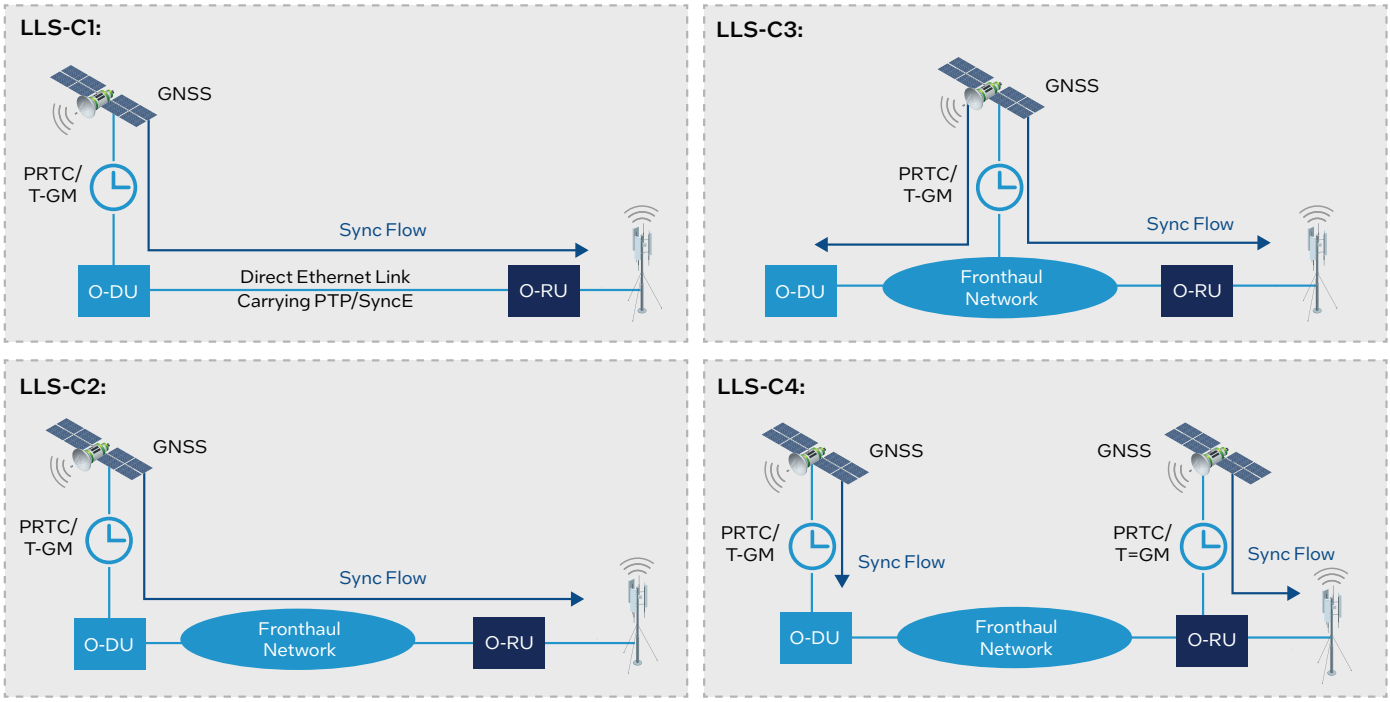

Figure 5. O-RAN LLS-C1, LLS-C2, LLS-C3, and LLS-C4 configurations.

LLS-C1 involves a point-to-point connection between the DU and the RU where the DU is the PTP synchronization source for the RU. LLS-C2 extends things further, in that Ethernet switches are included between the DU and the RU, and these switches may also act as PTP T-BCs (or T-TCs as discussed in the next section). In the case of LLS-C3, the T-GM is a distinct device and timing entity included in the fronthaul network, distributing timing to both the DU(s) and the RU(s). Finally, in the case of LLS-C4, the O-RU receives its timing and frequency references locally (e.g., by utilizing a directly connected GNSS receiver) and independently from the fronthaul and the O-DU.

In practice, all of this means that the grandmaster, master, and subordinate clock elements embodying a PTP synchronization solution must be capable of being deployed anywhere in the O-RAN infrastructure, including the O-RUs, O-DUs, and FHGWs.

O-RAN and the ITU-T

As one of the oldest international organizations still in operation, the International Telecommunication Union (ITU)[6] is a specialized agency of the United Nations responsible for information and communication technologies. The ITU comprises three sectors or branches: development (ITU-D), radio communication (ITU-R), and standardization (ITU-T). In addition to radio access networks, many other industries derive their standards based on ITU recommendations.

O-RAN specifications are linked to several ITU-T recommendations and IEEE specifications. As many as 18 ITU-T recommendations are included in O-RAN by reference. The methodology of delivering the appropriate timing to the RU is based substantially on the G.826x and G.827x families of ITU-T recommendations. A summary of the ITU-T recommendations invoked by O-RAN is provided in Table 1.

ITU Recommendations	Reason for Referencing in O-RAN Specifications
G.8275.1 and G.8275.2	Describe the PTP profiles that should be used for full timing support (FTS) networks and partial timing support (PTS) networks, respectively.
G.8271.1 and G.8271.2	Describe the network limits for FTS and PTS networks, respectively. One important piece of information found here is the definition of the reference points A, B, C, and D, of which B and C are of specific interest in the O-RAN context. For example, the O-RU UNI corresponds to reference point C in LLS-C2. The "network limits" means the "noise" an O-DU or an O-RU can expect on its input and that it needs to handle while meeting the requirements on the output.
G.8272	Describes the timing characteristics of a PRTC and the PRTC combined with a T-GM. G.8272.1 establishes more stringent requirements for an enhanced PRTC (ePRTC). This is relevant for an O-DU with local synchronization.
G.8273.2 and G.8273.3	Describe the timing characteristics of a T-BC and T-TC, respectively, that are deployed in fronthaul networks with full timing support. G.8273.2 also describes the timing characteristics of the T-TSC which is the fronthaul endpoint in the O-RU and the midhaul endpoint in the O-DU. It's worth pointing out that an O-DU is not a T-BC. G.8273.2 also defines four classes of T-BC with different requirements on timing accuracy. For the most stringent Enhanced O-RU specification the Class C (max|TE| < 30 ns) subordinate suffices.
G.8273.4	Describes the timing characteristics of a T-BC and a T-TSC in a fronthaul network with partial timing support.
G.8275	Describes a network architecture in Appendix III where the PTP profiles of the midhaul and the fronthaul are different, specifically if one is FTS and the other is PTS.
G.8260	Provides definitions of a variety of metrics. Section 3.1.20 defines some of the time error measurements used, specifically Max|TE|, cTE, and dTE.
G.8261	Describes the architecture for Synchronous Ethernet (SyncE). Appendix VI provides guidance on test conditions for networks that do not have full timing support.
G.8262 and G.8262.1	Define the jitter and wander requirements to which all network equipment in the fronthaul supporting SyncE (and enhanced SyncE) should comply.
G.8264	Defines the ESMC format that should be used for SyncE.
G.781	Defines the quality levels used in the synchronization status message (SSM) signaling over Ethernet slow message channel (ESMC).
G.810	Defines some basic measurement concepts used, specifically fractional frequency offset (FFO) and maximum time interval error (MTIE).
G.811	Defines a PRC, which is an allowed clock type in the fronthaul.

Table 1. ITU-T recommendations used in O-RAN CUS specifications.

The ITU-T considers two principal cases of network architecture referred to as FTS and PTS. These architectural concepts were briefly mentioned in the NTP and PTP discussions earlier in this paper.

In an FTS architecture, every device between a master clock and a subordinate clock must be PTP-aware and participate in the protocol. That is, each device must be either a T-BC or a T-TC. The intent is to defeat the packet delay variation introduced by queues in packet-switching schemes, but T-BCs and T-TCs achieve this differently. T-BCs act like "relays" by providing a subordinate function upstream and relaying the synchronized time downstream. T-TCs modify a correction field in the timing packet as it traverses the device by an amount equal to the residence time of the packet in the device.

As was noted earlier, IEEE 1588 provides for the tailoring of PTP to its deployment environment via the use of profiles. The ITU-T has defined a profile for FTS in recommendation G.8275.1. It should be noted that, whereas the ITU-T considers physical layer frequency synchronization (PLFS) as optional, in O-RAN it is assumed that the FTS architecture includes PLFS in the form of SyncE.

By comparison, in the case of a PTS architecture, there may be one or more PTP-unaware devices between a master clock and a subordinate clock. As a result, there could be substantive packet delay variation that must be accommodated by the subordinate. The ITU-T has defined a profile for PTS in ITU recommendation G.8275.2.

O-RAN prefers that the fronthaul and midhaul networks follow the FTS model, which however is a more costly implementation. The specification also includes a more realistic and practical approach in that the requirements include the recognition that, for various reasons, fronthaul and/or midhaul networks may, in practice, fall into the PTS category.

O-RAN borrows from ITU-T Rec. G.8273.2 for the T-TSC based on the assumption that the fronthaul network follows the FTS model. However, in O-RAN documentation, the asterisk shown in T-TSC* indicates that: "additional noise filtering to meet 3GPP may be required." This additional filtering could include:

- Any filtering must meet the air interface's ±50 ppb frequency offset requirement. Traditional wireless has always had a 50 ppb requirement, but 5G/O-RAN specifies this requirement over a short observation interval of 1 ms.
- Any additional filtering necessary to suppress wander in the physical layer to meet the Max|TE|$_{antenna}$ time-error requirement.
- Any filtering necessary to defeat packet delay variation if the fronthaul network has one or more PTP-unaware devices (i.e., the PTS model).
- Support for O-RU cascade mode operation.

The Intel PTP Solution

In the context of these discussions, the term "servo" refers to a software implementation of a suite of algorithms running on a processor. The servos running in subordinate clocks use timestamps to establish the correct time relative to the master clock, even when individual packets experience transmission delay variation.

For implementing PTP, Intel utilizes the open-source LinuxPTP suite, which includes the ptp4l protocol engine. Traditional implementations utilizing LinuxPTP employ open-source servos available with ptp4l. These servos are more than adequate for many real-world network deployments implemented using a FTS architecture. For example, the Intel O-RU Enablement Package itself employs one of these servos. Having said this, these servos don't provide any protection against the high levels of jitter experienced in PTS network architectures. For these cases, Intel has developed a proprietary PTP servo that can be run on Intel SoC FPGAs, Intel Xeon® CPU-based motherboards, and network interface cards (NICs) with an external digital clock synthesizer (DCS) and 1588 support. Furthermore, this servo also has extensions that allow the device to operate as a T-GM when provided with an external time reference, such as GNSS. It's important to note that users can take full advantage of the Intel PTP timing solution without making any changes to their LinuxPTP installation. An illustration of the PTP software architecture utilizing the Intel PTP Servo is shown in Figure 6.

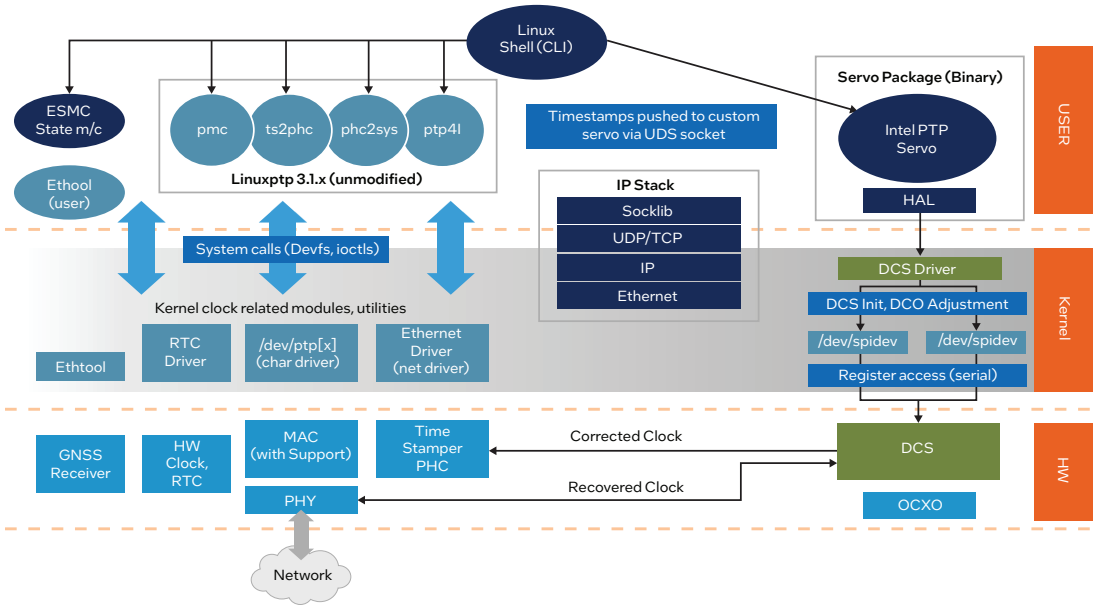

Figure 6. PTP software architecture using the Intel PTP Servo.

The Intel PTP Servo software can run on Intel Xeon CPU-based motherboards, Intel SoC FPGAs, network interface cards (NICs), and many other platforms with external DCS and 1588 support.

An example hardware implementation of the Intel PTP timing solution using an Intel SoC FPGA is illustrated in Figure 7. Hardware timestamping is performed in the Ethernet PHY/MAC functions. The packet filter, direct memory access (DMA), and ToD functions are all implemented in the FPGA's programmable fabric, while the Linux kernel and PTP software, which includes the Intel PTP Servo, run on the FPGA's hard processor subsystem (HPS). Using the timestamps encapsulated in the S-plane packets, the servo establishes the correct time-of-day and determines any phase and frequency correction factors that must be applied to the ToD function and the board-mounted DCS device. DCS devices suitable for O-RAN applications are available from multiple suppliers, and the Intel PTP Servo can work with any suitable device.

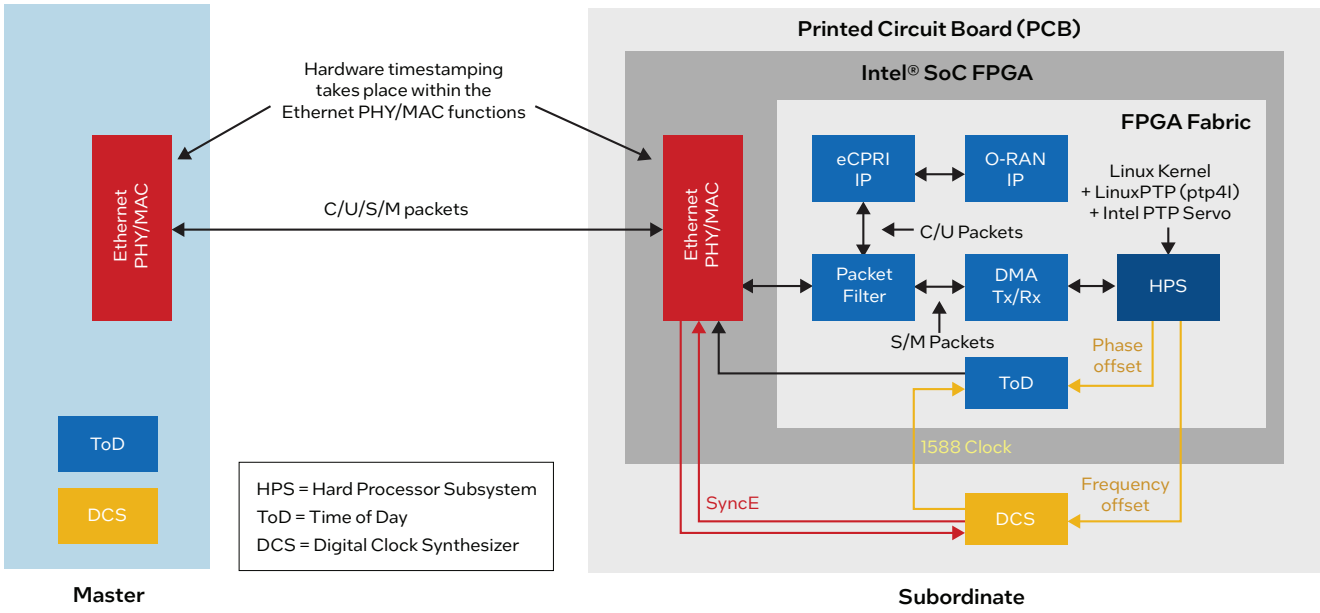

Figure 7. Example PTP hardware architecture with the Intel PTP Servo running in an Intel SoC FPGA.

This example shows eCPRI and other O-RAN intellectual property (IP) functional blocks implemented in the FPGA's programmable fabric. These functions would be replaced with other IP functions depending on the target application, such as industrial measurement and control systems, financial transactions, and sub-sea acoustic arrays.

Once again, it's important to remember that this is just one possible implementation example. The same servo software can be run on NICs, on Intel Xeon CPU-based motherboards, and many other platforms with external DCS and 1588 support.

Intel PTP Servo implemented on Intel Agilex 7 SoC FPGAs

As illustrated in the previous section, one possible Intel PTP timing solution implementation is based on Intel devices such as Intel Agilex 7 SoC FPGAs. These devices provide the extreme capacity and performance demanded by applications like O-RANs, the network core, and data centers. Intel Agilex 7 FPGA F-Series and I-Series are built on the Intel 10 nm SuperFin process technology, depicted in Figure 8. The F-Series devices are targeted at a wide range of high-end applications, while the I-Series devices are intended for bandwidth-intensive applications.

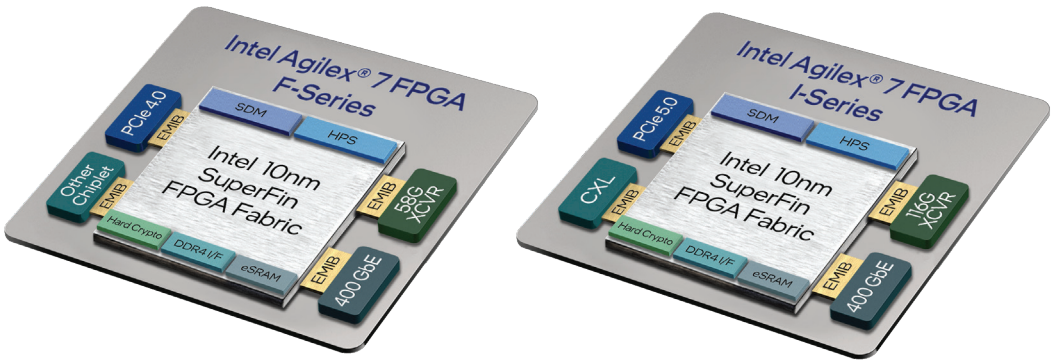

Figure 8. Example members of Intel Agilex 7 SoC FPGAs

The Intel Agilex 7 family includes FPGAs and SoC FPGAs. Both support hard interfaces in their fabric for external DDR4 memory and include a secure device manager (SDM). SoC FPGAs also include a hard processor subsystem (HPS).

A chiplet, also known as a "tile," is a small integrated circuit die containing a well-defined subset of functionality. In addition to the main FPGA die, Intel Agilex 7 devices contain two to six transceiver (XCVR) tiles. These XCVR tiles are connected to the main FPGA die using Intel embedded multi-die interconnect bridge (EMIB) technology, an elegant and cost-effective approach to the in-package high-density interconnect of heterogeneous chips. The result is that the chip and chiplets combine as a single large die.

Intel Agilex 7 SoC FPGAs support multiple types of XCVR Tiles, including E-Tiles, F-Tiles, P-Tiles, and R-Tiles. Different members of the Intel Agilex family provide different combinations of these tiles. In addition to various general-purpose input/output (GPIO) and high-speed SerDes interfaces, F-Tiles can support up to 400 Gbps Ethernet. Similarly, in addition to a range of other GPIO and high-speed SerDes interfaces, R-Tiles can support PCIe* Gen 5 and Compute Express Link (CXL) interfaces.

Of particular interest in the context of this paper is the fact that both E-Tiles and F-Tiles support PTP hardware timestamping with an accuracy of ±1.5 ns (for 10 – 100 GbE line rates) and in combination with Intel PTP Servo provide a Class-D (ITU-T G.8273.2) capable solution.

Test Case	FTS (G.8273.2, Noise Gen)		PTS (G.8271.2)		Without Support (G.8261, TC12)	
	Intel PTP Servo	ptp4l PI Servo	Intel PTP Servo	ptp4l PI Servo	Intel PTP Servo	ptp4l PI Servo
|TE|	<5 ns	<5 ns	1 µs	138 µs	4 µs	85 µs

Table 2. Comparison of 1PPS absolute time error measurements between Intel PTP Servo and ptp4l

Intel PTP Servo Performance Benchmarks

The efficacy of the Intel PTP Servo relative to an open-source servo is depicted in Table 2. The maximum absolute time error is shown for three cases: FTS, where there is essentially zero packet delay variation, and two cases of PTS for different levels/types of packet delay variation.

Concerning FTS, it's important to note that the requirement in the CPRI specification is ±8.138 ns. Observe in Table 2 that the Intel PTP Servo achieves less than 5 ns.

The most challenging situation for the O-RU is if the fronthaul is of an LLS-C2 configuration with PTS. This can be tested by applying noise to the input of the O-RU according to G.8271.2 Section 7.3.2, which describes the limits at reference point C for applications in accuracy Class 4 (i.e., ±1.5 µs) as described in G.8271 in Table 1.

The network limits provided in G.8271.2 are conservative and can be viewed as a "worst-case" for 5G network deployments. As seen in Table 2, the Intel PTP Servo achieves 1 µs (these results have been proven to be DCS independent) compared to the ptp4l proportional-integral (PI) servo solution, which can achieve only 138 µs. For a network operator, this equates to the ability to use an existing legacy network between the DU and the RU, thereby avoiding the costs associated with investing in a new FTS network.

Intel PTP Servo Value Proposition

Some of the key benefits of the Intel PTP Servo are:

- Architecture designed to support O-RAN (FTS and PTS).
- Superior performance in PTS and networks without timing-support relative to the ptp4l PI Servo.
- DCS-agnostic solution.
- Support for the open-source ptp4l solution vs. competing proprietary solutions with DCS lock-in.
- Portable timing solution scaling across current and future Intel FPGA and Intel eASIC™ products with HPS, with a strong roadmap and feature enhancements.

Please get in touch with your local Intel sales team for more information.

Conclusion

All communication networks require an appropriate level of clock synchronization across the different devices that comprise the network. The necessary level of precision and accuracy is both application and device dependent. For example, based on frequency-division-multiplexing (FDM) principles, legacy telecommunications networks required just frequency synchronization, also called syntonization. These networks further classified the synchronization requirements in terms of levels or strata. At the network core, where the requirements were most stringent, lived the Stratum 1 clocks, while Stratum 4 clocks were found at the edge where the requirements were the least stringent.

By comparison, modern networks based on packet-switching principles can perform their packet-based functions asynchronously. However, at the network edge, where the rubber meets the road, the need for precise and accurate synchronization remains because the service or application demands it. This is exemplified by mobile wireless communication systems where precise timing is required at the most extreme edge of the network, which is the antenna of a base station. To achieve this economically and reliably, high-performance clocks are placed closer to the core, and suitable distribution methods such as PTP and SyncE are used to provide accurate timing towards the edge.

Wireless usage and bandwidth expectations are growing rapidly. The RAN infrastructure to support this growth needs to be properly synchronized. The O-RAN community has recognized this phenomenon and expressed the need for stringent synchronization as detailed in the S-Plane body of requirements.

PTP is the chosen methodology for achieving this synchronization. Intel provides a significant body of technology for implementing end-to-end O-RAN solutions. In the context of this paper, Intel provides a superior PTP solution that can be applied in both FTS and PTS network architectures.

In the case of an example PTP implementation employing an Intel SoC FPGA, the Intel PTP Servo-based solution achieves a peak-to-peak time error of fewer than 5 ns in an FTS (G.8271.1) network architecture, dramatically beating the +/-8.138 ns requirement in the CPRI specification. Furthermore, in the case of a PTS (G.8271.2) network architecture, the Intel PTP Servo-based solution achieves a peak-to-peak time error of only 1 µs compared to the 138 µs result when using an open-source ptp4l servo.

It's important to remember that timing synchronization is essential for many applications. Although the body of this paper has concentrated on the demands of radio access networks, the Intel PTP Servo applies to a wide range of cutting-edge technologies, including industrial measurement and control systems, financial transactions, and sub-sea acoustic arrays.

Learn More

- Synchronization and Timing in Telecommunications, Kishan Shenoi, 2009, ISBN 978-1-4392-2632-2
- Synchronizing 5G Mobile Networks, Dennis Hagerty, Shahid Ajmeri, Anshul Tanwar, 2021, ISBN 978-0-13-683625-4.
- Measurement, Control, and Communication Using IEEE 1588, John Edison, 2006, ISBN 978-1-8462-8250-8
- WSTS: NIST-ATIS Workshop on Synchronization and Timing Systems (https://wsts.atis.org/)
- ITSF: International Timing and Synchronization Forum (http://www.telecom-sync.com/)

Intel Whitepapers

- Build Efficient and Cost-Effective mMIMO Solutions with Intel Agilex 7 FPGAs
- Build More Cost-Effective and More Efficient 5G Radios with Intel Agilex FPGAs
- Flexible Performance for 5G vRAN Deployments
- Enabling 5G Fronthaul
- Intel's Accelerated Virtual Cell Site Router Solution

Intel Devices and IP

- Intel Agilex SoC FPGAs
- Intel RAN FPGA Solutions
- Intel FPGA IP

References

1 In Search of Time: The History, Physics, and Philosophy of Time, Dan Falk, ISBN 0312603517

2 The Order of Time, Carlo Rovelli, ISBN 073521610X

3 Reality Is Not What It Seems: The Journey to Quantum Gravity, Carlo Rovelli, ISBN 0735213933

4 https://www.o-ran.org

5 https://www.3gpp.org

6 https://www.itu.int

7 Kishan Shenoi, Fundamentals of Synchronization and Timing, WSTS-2023 Tutorial Session, March 2023. WSTS 2023 Virtual Tutorial available at https://wsts.atis.org/2023-tutorial

8 O-RAN Working Group 4 (Open Fronthaul Interfaces WG), Control, User and Synchronization Plane Specification, O-RAN.WG4.CUS.0-R003-v11.00.01, Feb. 24, 2023.

9 Eidson, John C. (April 2006). Measurement, Control and Communication Using IEEE 1588. Springer. ISBN 978-1-84628-250-8.

altera™
An Intel Company