

A Living “Not-A-Draft-Of-The-Standard” Contribution from the Architecture Subgroup of the P3335 Working Group (AKA TimeCard) to the P3335 Working Group

Update 11/18/2024: This file was hosted on Google Drive in 2024 to allow for better collaborative editing. At the request of the Workgin Group, we are now putting it in iMeet. Subgroup member are encouraged to take their sections and start formalizing them to be uploaded as standards text proposals.

This is not a draft of the standard. Rather, we collectively document here our current thinking on the architecture and describe questions that need to be answered (and the proposed answers to those questions) through discussion at our meetings, such that a draft of the standard can ultimately be created more quickly.

This Table of Contents and discussion of various aspects of the architecture contains inputs from the below listed individuals, and is intended (with our collective implicit agreement) to be submitted as a contribution to the IEEE P3335 Working Group, as an effort that is (until such contribution is made) outside the working group.

Instructions to Arch Subgroup: Please be *brief*. Please do **not** paste pages of contents from elsewhere. The goal is to describe as briefly as possible those things that are likely agreeable to the team, or that need to be decided (in which case, take a position but acknowledge briefly the other alternative(s)).

Please do not make edits to sections that are not your contributions (at this time). Instead add comments (shift/alt/m)

Contributors: *[Add your name here when you make contributions below]*, Nobuyasu Shiga, Joe Gwinn, Rodney Cummings, Magnus Danielson, Zoltan Fodor, Kevin Stanton (subgroup chair and overall coordinator of this living contribution)

1. Introduction
 - a. PAR Scope
 - b. Purpose (**Ahmad** will create a paragraph describing the purpose / objectives for this standard—not how, including non-objectives)
2. Definitions
3. Normative References
4. Architecture
 - a. [**Ahmad with Magnus** via ad hoc subgroup: Block Diagram, showing Time Card as a subsystem within a System, including CPU etc]

5. Performance Metrics
 - a. [System performance]
 - b. [**Magnus** section, i.e. local clock performance]
 - c. [Timestamping resolution]
 - d. [Q: Do we care whether timestamps form a metric space? To be answered in the IRS]
 - e. Traceability
6. Interfaces (external [and internal], within, between, and without)
 - a. [Host interface **Zoli** / **Nobu**]
7. Environment
8. Applications and best practices
 - a. [**Magnus** section]
9. Annexes
 - a. Metrics (**Stefano** / **Doug** / **Magnus**)
 - b. Test procedures (**Rodney**)
 - c. Bibliography
 - d. Glossary

Introduction

PAR Scope

This standard defines the generic architecture and interfaces of a time card system, which constitutes a traceable source of time-of-day to heterogeneous systems that distribute and/or use that time. Additionally, this standard defines figures of merit that univocally characterize the relevant performance of the Time Card. The Time Card provides a traceable time-of-day for systems directly attached to it, as well as networked distributed systems. Such systems include, but are not limited to, servers hosting the Time Card, and servers synchronized with the Time Card using such protocols as Precision Time Protocol (PTP) or Network Time Protocol (RFC Request for Comments) 5905). This standard also defines the basic building blocks of the Time Card and their interfaces in order to allow modularization. The main building blocks include time source, local oscillator, and time processor. Additionally, this standard defines interfaces between the Time Card and other systems. This includes physical interfaces that allow input and output of time-related signals. This also includes logical interfaces that are compatible with Portable Operating System Interface for UNIX (POSIX) and include for example an interface to share a Physical Hardware Clock (PHC). This allows sharing the time of day with other systems, as well as providing means for diagnostic and configuration. The definition of logical interfaces allows for a variety of Time Card's form factors (e.g. Peripheral Component Interconnect Express (PCIe)) while ensuring uniform support from the operating system. Any device that complies with this standard

provides performance figures that are obtained following the specifications of this standard. As such, different implementations of the Time Card can be easily compared in terms of performance.

Purpose

This standard provides

1. Interoperability of different implementations of the Time Card with the systems, and their operating systems, that use it as the source of time-of-day; as such, to ensure a plug-and-play solution,
2. Modular implementation of the Time Card to allow better customization to industry needs, and
3. Univocal comparison of different implementations of the Time Card in terms of the relevant performance metrics.

The need for this standard (to the end-user)

[TBD]

Definitions (color words in the text that should be defined here [like this](#))

- PHC: Physical Hardware Clock or PTP Hardware Clock
- MAC: Miniaturized Atomic Clock
- RCB: Receiver Carrier Board
- PTM: Precision Time Measurement
- TDEV: Time Deviation
- MTIE: Maximum Time Interval Error
- API: Application Programming Interface

Normative references

Architecture (Magnus)

Note: The overview presentation of Architecture [is here](#)

Overview

The P3335 TimeCard is an enhancement to other systems. Thus, it is a subsystem to aid in achieving timing of that system. This system consists of other sub-systems which may or may not interact with time of the Time Card. Other sub-systems can include CPU, GPU, network interface cards and other TimeCards. The Time Card enhances these systems, as they typically do not have sufficiently good local clock, clock steering, time-measure capabilities and other time-transfer mechanisms.

Within the host system, multiple time-transfer mechanisms may exist, depending on what fits the particular needs. A total system may also have indirect transfer in relation to the Time Card, in such a way that time is compared/transferred through one or more intermediate locations. Indirect access may be sufficient for some applications, while others may require more direct access.

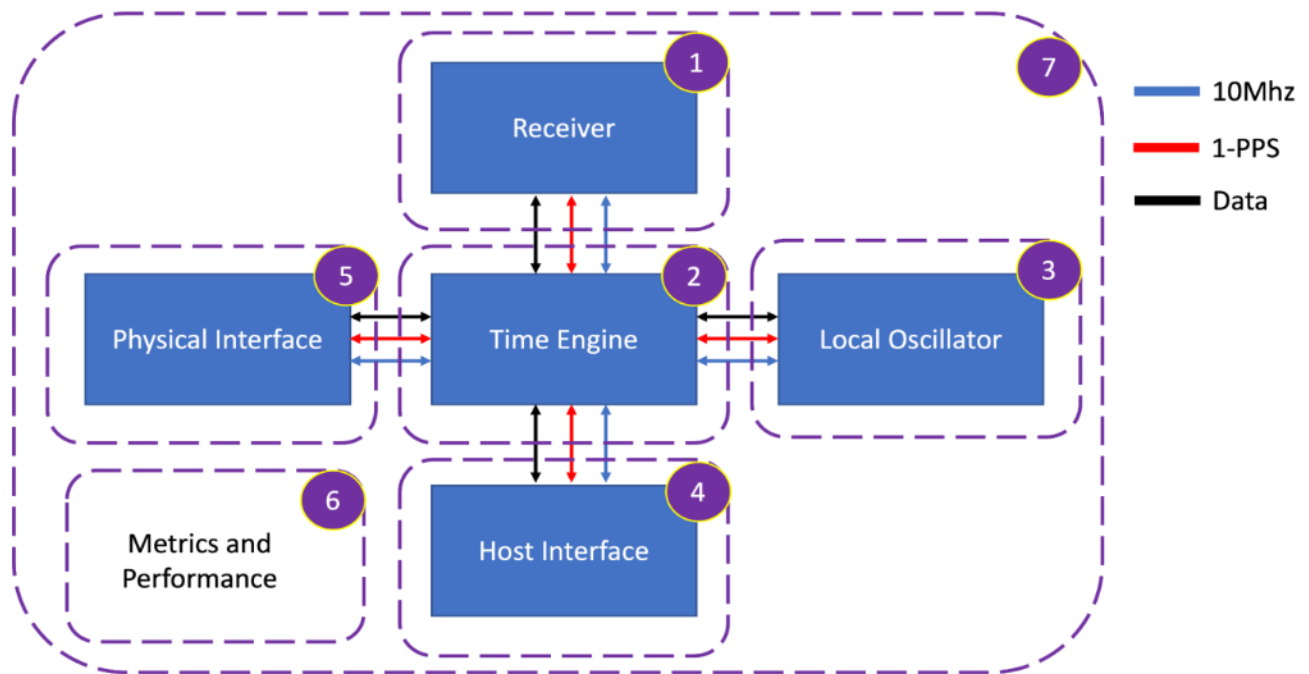
Architecture Overview

Time Card

The Time Card itself contains a set of basic features, and then various optional features and interfaces.

[While interfaces as such are needed, exactly which is required is TBD.]

[Which frequency steering mechanism (Local clock or Time generator) should be chosen or if both are needed is TBD.]



Local clock

A core feature is a local clock. The lack of a stable enough clock in the system limits what performance can be achieved for a range of applications. Having a good local clock enables the uses for many applications. The local clock may be steerable or not.

A core feature is a time generator. The time generator holds the time and is driven by the local clock. The time of the time generator may be set or adjusted. The time generator may include a frequency adjustment of the local clock.

Timestamping facility

A core feature is a timestamping facility is able to time-stamp events of various sources. It provides a timestamp in the time of the time generator, optionally with some high resolution part.

Time input interfaces

A core feature is a time input interface. This allows external signals to be time-stamped, and potentially additional time information to be associated with the time-event, such as the time as represented by the source (time of day, date etc).

Time output interfaces

A core feature is time output interfaces. This allows the time and frequency of the time generator to be delivered to other devices. It may also carry time. Examples can be PPS output, 10 MHz output.

Time transfer interfaces

An optional but important feature is time-transfer interfaces, which typically enable the comparison of time between TimeCard and some other subsystem to enable transfer of time or time measures in either direction. This enables other subsystems to convey time to the TimeCard so it can be set, measured or compared, or that the subsystem can convey the time of the TimeCard over its interfaces and other uses.

Management interface

A core feature is the management interface. The management interface enables the host to control all aspects of the TimeCard. Typically the management interface also has the support for time-transfer to the host, but it may not be the only interface for this purpose. It's unclear that P3335 should specify any management interface, instead leaving this to design and the market to decide.

Performance Specifications (Stefano)

Local clock performance

A range of possible local clock choices exists. In telecom, existing clock definitions fit specific needs and can be referenced. These may or may not be applicable to all uses.

Name	Frequency	Holdover	Standard	Location	
SEC	+/- 4.6 ppm		ITU-T G.813 Option 1	Internal	
EEC	+/- 4.6 ppm		ITU-T G.8262	Internal	
SSU	+/- 4.6 ppm		ITU-T G.812	Internal/External	
PRC	+/- 1E-11		ITU-T G.811	External	

Time Generator

It is proposed that the Time Generator supports PTP timescale as defined in IEEE 1588, can maintain time after year 2036, can hold current TAI-UTC difference. This forms a continuous technical time-scale, which can be converted to UTC as applicable.

Note: We can expect IEEE 1588 time-stamping to operate with these settings, but it also fits many other similar applications as it is easy enough to convert to and from these, but no point in inventing a new format.

It is proposed that the Time Generator support coarse time-stamps of at least 100 ns (as derived from a nominal 10 MHz oscillator), but that higher resolution is expected and encouraged. This resolution comes from the phase-accumulator core clock frequency, which may vary between implementation pending available technology and implementation decisions.

Note: The timestamp resolution is not alone limited by this, but it should be chosen to reflect a minimum expected core clock frequency.

It is proposed that the Time Generator time-stamp resolution be at least 1 ns, but higher resolution is encouraged.

Note: Timestamping often build on a coarse counting time-stamp, and a high resolution interpolation.

It is proposed that the Time Generator has a phase-accumulator synthesizer allowing for at least frequency resolution of $1\text{E-}12$ (requiring at least 40 bits), but higher resolution is recommended.

Note: Cesium and Hydrogen clocks typically have output frequency adjusted in $1\text{E-}15$ resolution. While this is relatively cheap to achieve, it may not be needed to put the additional 30 bits resolution requirement on all implementations when only a few instance benefit from it.

Interfaces [Nobu and Zoli]

Interface will provide the following;

1. Configuration and monitoring of the TimeCard
2. Passing the current time from the TimeCard to the host computer for the use of software
3. Passing the current time from one TimeCard to one or more other subsystems compliant to the p3335 interface requirements

Physical Host Interface

Physical Host Interface specifies the physical connection to the host hardware.

[Example]

The most common interface should be based on PCIe but it should not be limited to it only. Suggested to have PTM on the PCIe enabled as well.

Software Host Interface

Software Host Interface specifies the API to configure and monitor the TimeCard's states.

The underlying SW / register structure can be vendor specific, One of these implementations can be the OCP TimeCard implementation.

NOTE: [Kevin typed this 2024-03-20: A high-level, example definition of a software interface is shown as an aid (not mandatory): configure____(), get_cross_timestamp(), ... And it's OK to point at actual APIs that exist at this point as examples that we intend to 'allow', but we must not reference such as mandatory]

[Examples]

Timestamping facility interface / TimeCard PTP interface: should be based on:
Linux kernel PHC API with additional features that later on can be added to the official API.

<https://www.kernel.org/doc/html/latest/driver-api/ptp.html>

Local clock / DPLL implementation can use the Linux kernel DPLL API:

<https://www.kernel.org/doc/html/next/driver-api/dpll.html>

The GNSS API can follow:

...

[Kevin typed this 2024-03-20: PPS input? PPS output?]

Physical TimeCard measurement / communication interface

Physical Time transfer Interface specifies the physical connection from one TimeCard to one or more other TimeCards in the system

[Kevin typed this 2024-03-20: Hardware interfaces such as this would be very specific, unlike SW APIs]

Software TimeCard measurement / communication interface

Software Time Transfer Interface specifies the API to transfer the time between TimeCards

[Kevin typed this 2024-03-20: Opher suggested referencing IEEE P1952, Resilient Positioning, Navigation and Timing (PNT), including its terminology and the architecture (which is quite stable). Also PTP: Note, Doug A is editor of this standard]

Environment

The operational environment of a TimeCard can vary with implementation. An assumed operational environment is that of a Central Office / Computer center type of environment. Such is regulated in ETSI ETS 300 019-1-3 and equivalent exist in Bellcore GR-63. For other applications, extended temperature, humidity, pressure and altitude limits may apply. There

exist some standard environmental tests for temperature cycling. See also IEEE 1193 for environmental effects on oscillators, as well as the ITU G.811, G.812, G.813 standards for oscillator performance and associated test conditions. While environmental conditions may be chosen not to be normative, it is good to consider the range of environmental conditions that can exist.

Applications & Best Practices / Use Conditions (Joe / Magnus)

In order to facilitate an understanding of various operational context, groups of application scenarios is presented. These are non-normative, but documented to facilitate understanding of the different uses. It also represents a non-exhaustive list, and more cases can be added as it is seen that distinct environments need to be considered for some case.

More extensive use case descriptions are documented [here](#).

Server case

Server cases include small and large server farms, where machines need access to improved timing. The server platform typically do not have good clock for timing, but sufficient for keeping the hardware operational. Environmentally, these use Central Office type of conditions, and can at times operate fairly hot and with variations in temperature due to AC cycling. The power density requires forced convection cooling, and it can be quite challenging from that aspect. Exchange of time using PTP and other protocols can be expected. For larger server setups, it is unfeasible for all machines to have GNSS reception, at which case network needs to be used. While not all server locations meet particular environment specs, existing telecom “central office” specs can probably be used for most of these.

Base station case

In mobile networks, IT-styled hardware is used in cooperation with dedicated hardware to build the base station. The physical location of this can vary greatly, including uncontrolled huts being exposed to cold as well as very hot conditions. A cold start can require extra time just to get up to temperature. A hot situation can exist when there is no cooling on a hot day. Exchange of time using PTP and other protocols can be expected. GNSS availability can not always be assumed or generally allowed. Some operational conditions bans all but approved GNSS systems. For this scenario will a range of pre-existing performance standards exist, and also environmental specs.

Annex A - Metrics ()

Note: These will be driven by Performance Requirements

For oscillator performance (and overall system performance as measured on the output interfaces), the long term stability metrics are TDEV and MTIE. TDEV is a giving a time deviation, similar to standard deviation, for the effect of random noise effect on phase stability. MTIE is giving the maximum amplitude deviation for systematic disturbances. Both these give measure over some observation time τ . These metrics can also be used for noise tolerance specifications at the input interfaces) For short-term stability, jitter, jitter tolerance limits are given, for output and tolerances at input. These metrics assume oscillators (and system) operating stable within the temperature range after any startup behaviors have stabilized and without environmental disturbances.

In addition the holdover metric specifies for how long an oscillator (and overall system) can keep within some defined limits. The specification is typically in terms of accumulated phase error over time after external reference is lost and the system enters holdover

In telecom standards exists for a number of clock-types, giving limits and test-conditions for these measures.

In addition, standards also exist for total systems, where said clocks is integrated with some form of interface system. (comment: I think the main target is specification for the system; that is why I suggest to add this clarifications earlier on in the text)

Time error accuracy can be specified in terms of maximum absolute Time error, as verified against a primary time standard (typically UTC).

Base definitions are found in ITU-T G.810, G.8260, IEEE 1139 and IEEE 1193.

Annex B - Test Procedures (Rodney)

Note: No need to push these forward early. May or may not be necessary / included in the standard.

x< Note from Rodney (delete later): I took most of the organizational concepts from the O-RAN Conformance Test Specification. To see a relevant example, go to

<https://orandownloadsweb.azurewebsites.net/specifications>

Scroll down to WG4 Open Fronthaul Interfaces Workgroup, click Download for O-RAN Conformance Test Specification, open the Word document and scroll way down to 3.3 S-Plane Conformance Tests, and read a few pages from there. "S-Plane" means synchronization, like PTP.

ITU-T has relevant test specs as well. Specifically, G.8272 and G.8272.1 specify performance testing for a "time card" that outputs PTP and SyncE.

Of course we can copy practices from other organizations as well.>

Overview

< Note from Rodney (delete later): Here we explain P3335's overall approach to testing.- At a minimum, we should describe the types of testing that can be performed on a Time Card, and clarify which type is specified here in the P3335 document itself.

- Conformance: Testing performed on a single Time Card product, to validate that it conforms to the P3335 requirements (in previous clauses). There are two sub-types:
 - Functional: Does the Time Card do what is required?
 - Performance: Does the Time Card meet the required performance metrics?For a protocol, functional testing can be very detailed (>10000 pages), often specified in a standalone document. Since P3335 is not a protocol, I'm assuming we can handle it here.
- Interoperability: Testing multiple products together. P3335 **needs to decide whether the standard will specify interoperability tests formally**, or leave it to plugfests. I'm assuming plugfests for now.
- Certification (aka badging): Will P3335 have a logo that demonstrates conformance to end-users? If so, testing organizations must be authorized to perform this testing (usually conformance). I'm assuming 'No' for now.

Test Environment

< Note from Rodney (delete later): Look at O-RAN's 3.3.1 as an example. Show a picture of how the test equipment connects to the Time Card (device under test). This is often overlooked by test specs, but is incredibly useful for understanding, and essential as a reference for the actual functional/performance specs. >

Functional Test of <something A>

< Note from Rodney (delete later): Look at O-RAN's 3.3.2 as an example. Each functional test is clear, including the steps performed by the test. The test spec should be written with automation as the goal, but not necessarily a precondition. >

Test Description and Applicability

< Note from Rodney (delete later): Does the test only apply to some Time Cards and not others? When it applies, is the test mandatory? >

Test Entrance Criteria

< Note from Rodney (delete later): What do we assume before the test executes? >

Test Methodology

< Note from Rodney (delete later): Textual description, ideally with references back to the clauses that specify what is being tested. >

Pass / Fail Criteria

< Note from Rodney (delete later): Steps of the test, with explicit pass/fail criteria for each step. >

Performance Test of <something B>

< Note from Rodney (delete later): Look at O-RAN's 3.3.3 as an example. The same four sections as Functional are used to specify the Performance test. The pass/fail criteria is typically not a table of steps, but can instead be something like a mask (e.g., MTIE). >

Annex ? : Canonical examples of possible instantiations of the architecture

Note: Subsumed in Best Practices and Use Cases

Example 1: A time card containing several PTP instances (OC or BC)

Example 2: A time card containing timescale (transformations?)

Example 3: A time card containing a large # of diverse elements

Conformance

< From Rodney (delete later): Although we don't need to decide detailed conformance at this stage, we do need to decide "What is a Time Card?", and that means deciding how interoperability happens. Some IEEE standards do this with a "Conformance" clause that describes the overall structure of how the standard works from an interoperability point of view. Here's a rough structure based on comments that I heard in the March 6 2024 System Arch call... >

Core Requirements (things every Time Card has)

- A Time Card shall plug into the backplane of a chassis, where the chassis contains a host computer.
- As specified in X.Y.Z, the quality of the Time Card's local clock shall be equal or better than the local clock of the host computer.
- The Time Card may support PTP, GNSS, <other interfaces>, ...

Host Interface

The hardware interface between the Time Card and host computer may be PCIe, <other>, ...

PCIe

If the hardware interface is PCIe, the Time Card card shall do the following:

- Requirement A (PTM?), specified in A.B.C
- Requirement B...

If the hardware interface is PCIe, the Time Card card may do the following:

- Option C

Time Interfaces

- PPS

PTP

If the Time Card supports the PTP option, the Time Card shall operate one or more IEEE 1588 PTP Instances as either OC, BC, or TC:

OC

If the PTP Instance is an OC, the Time Card card shall do the following:

- Requirement A...Change
- Requirement B...
- Option C...
- The PTP Instance may be capable of operating as a grandmaster (GM).

If the PTP Instance is GM capable, the PTP Instance shall do the following:

- Requirement D...

BC

If the PTP Instance is an BC, the Time Card card shall do the following:

- Requirement A...

GNSS

If the Time Card supports the GNSS option, the Time Card shall do the following::

- Requirement A...

Notes: 2024-06-19

- Why a Timecard Standard: ability to compare performance between timecards, and "interoperability"
- Regarding performance: perhaps some bare minimum performance, and other optional steps to improve the performance
- Kevin:

