

The Challenges of Implementing IEEE 1588 Clients

Four key decisions must be made when trying to overcome the challenges of implementing IEEE 1588 clients.

As demand grows for more bandwidth throughput, the requirement to synchronize network functions is becoming more critical. The biggest driver of synchronization is the explosion of wireless networks, whose nodes must all be in lock-step.

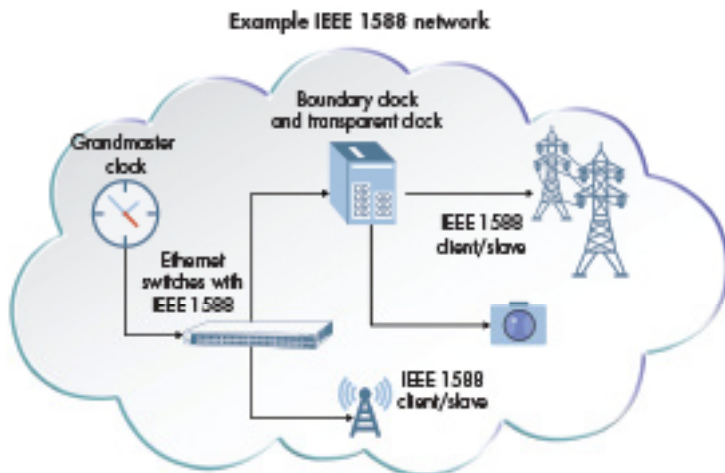
The IEEE 1588 network timing standard has been the cornerstone protocol used to implement synchronous networks in wireless backhaul, carrier Ethernet, routers, and data-center interconnects. Its popularity has led to it also being leveraged in markets outside of communications, such as industrial applications like smart grids, machine-vision cameras, and industrial automation.

While it's relatively easy to implement IEEE 1588 across most of the network, challenges arise when implementing IEEE 1588 clients. Thus, four critical decisions need to be made when creating complete client/save solutions.

Figure 1 shows some of the key points of an IEEE 1588 network.

It all starts with a grandmaster clock device, which is the keeper of accurate time. In addition, there are likely numerous Ethernet switches, gateways, and access points that need to be aware of the IEEE 1588 protocol. Finally, client or slave devices reside at the end of the network.

To ensure the most accurate timing across the network, all nodes in the network should be IEEE 1588-aware. One interesting characteristic of IEEE 1588 is that it will still work even when unaware nodes populate a network, although the accuracy of the recovered network clock will not be as precise. For the most precise synchronization, all switches and client



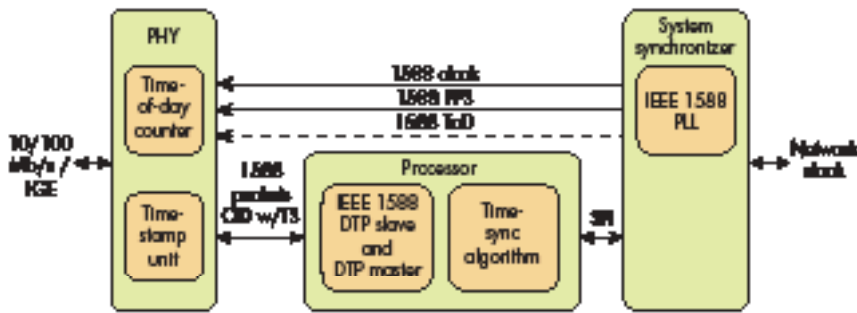
1. Here is a typical example of an IEEE 1588 network, which includes the grandmaster clock, switches/gateways, and client or slave devices.

devices should support the IEEE 1588 protocol. Knowing this background on the networking standard, what steps should be taken to implement an IEEE 1588 design?

Many vendors make grandmaster clocks that can plug into an Ethernet network. These may be packaged boxes or boards or module-type solutions. Getting a switch is also straightforward, but it has to support the IEEE 1588 protocol. This is often part of the switching support software that's purchased with an Ethernet switch.

For client devices, the answer isn't quite as simple. There's typically only one Ethernet port into the device, so buying a multi-port switch is often overkill. But, how else do you obtain the network clock? To obtain a recovered clock, a client or slave device requires four main elements:

- An Ethernet PHY to time-stamp the IEEE 1588 packets.



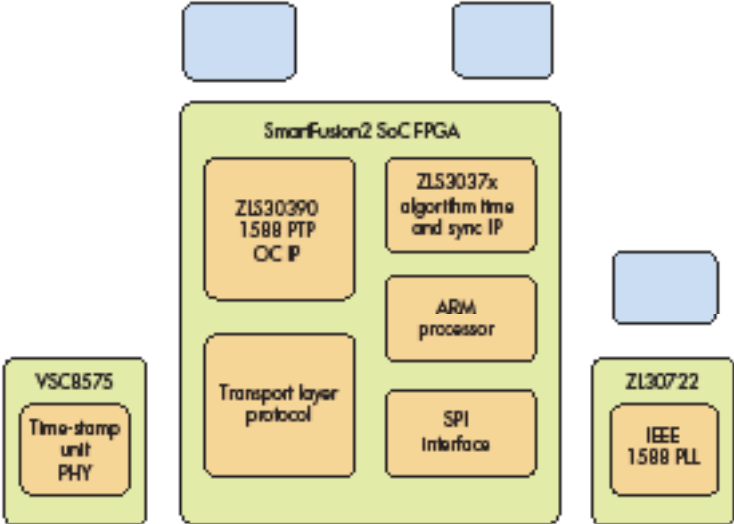
2. To achieve a complete IEEE 1588 client/slave solution such as that shown, designers must make multiple key decisions.

- A processor that can run the IEEE 1588 Precision Time Protocol (PTP).
- The processor also has to run a servo algorithm and interface to an adjustable phase-locked loop (PLL).
- A high-precision PLL that supports IEEE 1588 accuracy and then outputs the network clock.

Semiconductor vendors will often have pieces of a client/slave design, but not a complete solution. To implement this complete IEEE 1588 client/slave design (Fig. 2), architects need to make several decisions.

Decision #1: Leverage host processor or use a dedicated one?

One of the first decisions engineers must make is whether to use the host processor to run the IEEE 1588 PTP and servo algorithm, or use a dedicated processor. Of course, using the host processor doesn't add cost to the design, but it has some major drawbacks. For example, if the host processor is fairly heavily loaded and the response time from being interrupted is non-deterministic, this becomes a problem because the



3. This complete IEEE 1588 client/slave solution developed by Microsemi includes a Cortex M3-based FPGA that runs the protocol.

IEEE 1588 PTP must be monitored and reacted to in a timely manner. If this doesn't happen, the PLL will not be adjusted often enough, which will decrease the accuracy of the recovered network clock.

If a design uses a dedicated processor for the IEEE 1588 PTP and servo algorithm, then the performance would be assured and, consequently, a mid-range performance microcontroller could actually be used, such as an ARM Cortex M3. The M3 would be

dedicated to running the IEEE 1588 software timing routine and the algorithm to control the PLL. This software monitors the network clock and adjusts the PLL if necessary. It's likely that some FPGA logic will also be required to interface to the Ethernet PHY and the necessary interface buses to simplify system host integration.

Using an ARM Cortex M3-based FPGA for these functions would minimize system software integration and testing because the performance would be known. An M3-based FPGA solution also allows for quicker field updates when new IEEE 1588 profiles or enhancements are available.

Decision #2: Use IEEE 1588 PTP and servo algorithm from a third-party IP provider, or a silicon vendor?

Some PLL vendors will provide the PLL and then suggest a third-party IP provider for the IEEE 1588 PTP and servo algorithm. Purchasing from a third-party provider gives a designer the flexibility to choose any PHY and PLL, but the processor selection may be limited and support issues could be a problem.

If the IP provider's servo algorithm doesn't work, do you call the IP company or the PLL vendor? These conflicts could result in longer debug and validation cycles. Often, it's better to choose a PLL and the IEEE 1588 PTP and servo algorithm from the same company. Even better is if the vendor can demonstrate that its PLL was tested with the software. This provides confidence that the solution will operate as intended. In addition, if there are issues with the design, you can call the silicon vendor who has both components.

Decision #3: What PLL vendor capabilities are most important?

When selecting a PLL, engineers may need to consider the heritage of the silicon vendor. If the product is going to be deployed in a network with equipment from other vendors, you want to use a vendor with a sizable market share to give you

the best possibility of interoperation. If your network is closed, this isn't as important.

If you anticipate making many different types of IEEE 1588 slave/client products, you should look for a vendor who has a range of different types of PLLs. It's best to pick a vendor who can support IEEE 1588 and SyncE, can select multiple sources to switch between, and so on. Lastly, using a vendor that provides not only PLLs, but also the servo algorithm, is a wise choice in order to minimize debug issues.

Decision #4: Use a complete module or an embedded solution?

Module solutions are available for IEEE 1588 client/slave implementations that should, theoretically, be the simplest to implement. The tradeoff of a module is that they're usually a superset solution, and thus cost more. Vendors that provide modules are also smaller-size companies, and if longevity of supply is important, this may be a concern. The alternative is to use a multichip embedded solution. Figure 3 offers an example of a complete IEEE 1588 solution from Microsemi.

Although this type of implementation may require more board space, it has several positives. First, the Cortex M3-based FPGA that runs the protocol can leverage the FPGA fabric for acceleration functions if modifications are needed. Another benefit is that the PLL choice can be flexible, only needing driver changes in the software.

For simpler designs, choose a single PLL source device. If your design requires multiple clock sources, SyncE, and IEEE 1588, select a higher-end PLL. The solution will be the same with just software driver changes, making the software host

interface uniform across multiple designs. Lastly, this solution has validated performance; therefore, implementing it as shown will ensure the clock accuracy in the provided reports.

More system designers are being asked to implement IEEE 1588. The steps as outlined here are a great place to start—determine the grandmaster clock, what switches and gateways will be used, and then determine the needs for the client/slave devices. The decisions for selecting the optimal IEEE 1588 solution will depend largely on your corporation's knowledge of the protocol. Look for vendors that can provide complete solutions, as well as enable their customers to leverage this experience so they have the flexibility to implement the optimal network timing design for their specific requirements.

TED MARENA is the director of FPGA/SoC marketing at Microsemi. He has over 20 years' experience in FPGAs. Previously Marena has held roles in design engineering, technical sales/support, business development, and product & strategic marketing. In February 2014, while working for Lattice Semiconductor, he was awarded Innovator of the Year. Marena has defined, created, and executed unique marketing platform solutions for vertical markets including consumer, wireless small-cell, industrial, camera, display, and automotive applications. Marena holds a Bachelor of Science in electrical engineering Magna Cum Laude from the University of Connecticut, and a MBA from Bentley College's Elkin B. McCallum Graduate School of Business.