SoCe

System-on-Chip *engineering*

Time Sync Simplification

For Safety Critical Applications

September 2022

David Modroño

Daniel Uribe

Introduction

The objective of this presentation is to put on the discussion table the need to simplify considerably the TSN synchronization mechanism for use in airborne networks requiring Safety Critical certification.

We would like to encourage a discussion among all interested participants to come up with ideas and proposals for defining a simpler synchronization mechanism, better suited to static airborne TSN networks, with the aim of facilitating the adoption of such networks as the digital backbone for flying platforms.

Context

- Safety is an important topic in TSN for Aerospace Onboard Ethernet Communications. New customers who are considering the adoption of TSN for their new generation of airborne platforms require that TSN devices/solutions be certifiable according to DO-254 / DO-178. They are also requesting high Design Assurance Levels, up to DAL A.

- On the other side, TSN is a complex technology, comprised of many different sub-standards, capable of addressing a large amount of different use cases and network architectures.

- This adds complexity to most mechanisms and is in direct contradiction with the basic premise of safety critical systems design to make everything as simple and straightforward as possible.

Most existing TSN solutions are designed for non-safety critical environments

- The majority of TSN use cases do not have safety requirements (or not at high levels), as they do not compromise human's life. In such scenarios, TSN implementations are geared towards maximizing the set of available features (not only TSN-based, but also looking at retro-compatibility with existing Ethernet devices). This approach results in complex solutions.

- As an example, SoC-e's TSN Switch IP Core has around 80K RTL code lines and the count will increase over time as new TSN standards are approved.

- Simplicity and feasibility of "light" implementations are not usually in the list of considerations driving the development of these standards.

Most existing TSN solutions are designed for non-safety critical environments

- In the specific case of the Time Synchronization (gPTP), the use of a combined SW + HW implementation is quite common within the existing TSN solutions. This means that some components are running on the HW (for example, timestamping units in RTL) whereas the algorithm itself is implemented in software. Open-source stacks supporting gPTP with all its features are available.

- These stacks (necessary to implement all the details required by gPTP) are not simple, comprising as many as 22,645 lines of code. This is not really a good starting point for a simple mechanism that would be appropriate for Safety Critical certifiable networks.

- While one of the options could be developing an RTL-only version of a full gPTP stack, this requires a huge effort, and will also result in a large amount of RTL code.

- Another solution could be developing a certifiable gPTP stack for RTOS (VxWorks, PikeOS, etc.). There are already some solutions available, but none of them is compatible with gPTP requirements and certifiable.

Scope and Applicability

- The scope of this discussion on potential simplifications of the AS synchronization mechanism is limited to a very specific subset of use cases: static Ethernet networks which serve as a digital backbone on airborne platforms, space platforms, or other military mobile platforms.

- These networks are not only static, but also handle only pre-defined traffic. It is reasonable to expect that the TSN network configuration has been previously implemented in the lab and all the data flows have been emulated, tested and profiled using suitable development tools.

- As a result, a synchronization mechanism for these use cases would normally not be required to handle variability, extreme situations and weird borderline cases. The main concern, given the type of application, is its capability to handle correctly all potential error and failure situations.

Goal

- **The proposed goal is to define a simplified AS synchronization mechanism specific for the Synchronic Aerospace Profile (Profile B).**

- This mechanism should be simple enough to allow implementation in firmware only (VHDL code) with a reasonable investment of engineering work.

- One key point is that the simplified AS mechanism should work independently of any software and independently of any operating system.

Proposed Approach

- The first step in the process should be removing from the AS definition all the features and mechanisms that have no contribution or use when dealing with a static pre-defined TSN network. This implies removing all or most of the configuration flexibility available in gPTP.
  - Limit the number of domains
  - Use syncLocked mechanism (ignore syncInterval), forwarding instead of regeneration
    - Closer to a Transparent Clock Behavior
- The next step could be looking for additional opportunities to simplify the synchronization mechanism. For example:
  - Remove Grandmaster capability for switches
  - Remove BMCA as well as Announce messages (externalPortConfigurationEnabled)
    - https://www.ieee802.org/1/files/public/docs2022/liaison-8021to1588-announce-message-0722-v01.pdf
  - Remove Signaling messages
    - https://www.ieee802.org/1/files/public/docs2019/dg-cummings-autosar-time-sync-0519-v01.pdf
- After these two steps we would end up with the Core Synchronization Mechanism.
- The third and last step would be to analyze the functionality of the Core mechanism and propose options and ideas to define a simpler mechanism with equivalent functionality.

Ideas/Proposals to Simplify the Core Synchronization Mechanism

- Open for discussion…

CONTACT US

ADDRESS

Avenida Ribera de Axpe 50, 6

Edificio Udondo

CP: 48950

Erandio (BIZKAIA)

Spain

PHONE

+34 944 42 07 00

EMAIL

info@soc-e.com

Ips & MODULES

www.soc-e.com

END EQUIPMENT

www.relyum.com