# TDD Resync Header Format Proposal


A Leading Provider of Smart, Connected and Secure Embedded Solutions

**Scott Muma, Steve Gorshe**

7/10/2025

# Supporters

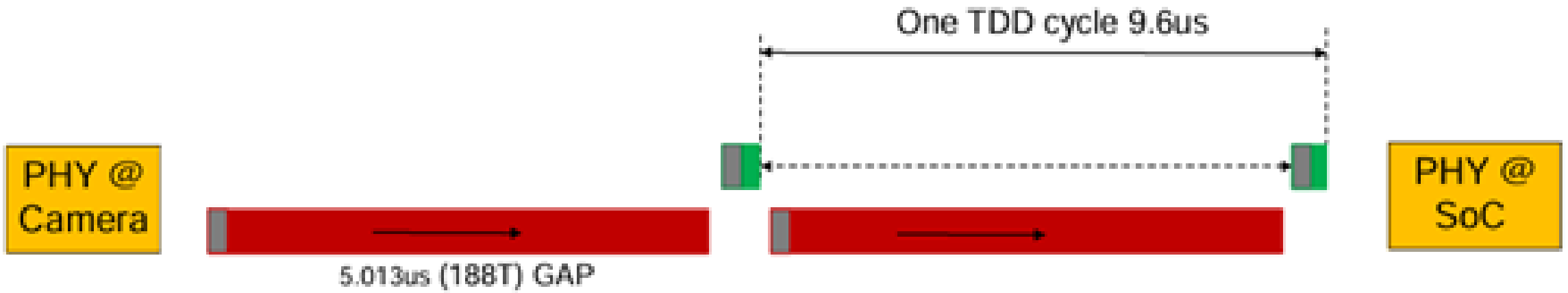
- Ahmad Chini (Broadcom)
- Kamal Dalmia (Aviva Links)
- Mehmet Tazebay (Broadcom)
- Ramanjit Ahuja (Onsemi)
- Frank Wang (Realtek)

# Introduction

The resync header (also called refresh header or burst header) is used at the beginning of TDD bursts to affirm the transmitted symbol phase, as well as the signal level, at the receiver.

- The resync header has not been specific addressed for the P802.3dm TDD proposal
  - The default assumption has been to reuse the format specified by ASA
- The fundamental characteristics required for a resync header include:
  - Adequate symbol transition density and run-length limit to simplify the RX PLL reacquisition / confirmation of the signal phase
    - The TDD format assumes frequency lock in initial training period (jitter and drift spec that determines phase drift during gap periods will be provided separately)
      - The PLL confirms and fine-tunes the phase during the resync header
  - The header symbol pattern should minimize the associated EMI
    - i.e., it should be nearly random to prevent increased PSD peaks
- The frequency lock, maintained by loop timing, allows using a much simpler header structure, such as the one proposed in this contribution

# Resync Header Illustration




The burst resync headers are shown in the grey blocks at the beginning of the bursts

# High-Level Aspects

- Resync header symbols will always be PAM2, independent of whether the data portion of the burst uses PAM2 or PAM4
- The resync header length is specified in multiples of 80 symbols:

| Data Rate and Direction | Number of Resync Header Symbols |
|---|---|
| Upstream (all rates) | 640 |
| Downstream for 2.5G | 480 |
| Downstream for 5G and 10G | 960 |

# Proposal

- Use a PRBS11 (PN11) sequence
  - The PRBS11 is sent until the last 8 bytes of the burst header. Then XOR the next 32 bits of the PRBS11 sequences with 4 bytes of 0x01 followed by 4 bytes of 0xF0.

- Notes
  - The PN11 sequence is long enough to avoid PSD peaks
  - A PN sequence is simple to implement
  - The PRBS11 pattern generator is illustrated in Figure 72-3 of 802.3-2022
  - Using the PRBS11 generator produces 2047 different burst headers before repeating
  - The PN11 is already being used in test modes
  - The inversions in the last 8 byte do not change the spectrum of the PRBS pattern significantly but provides an option to easily find bit/byte alignment and the boundaries between the burst header and the payload

# Questions?