SPI Parallel Read Operation on GBB (1722c)

Adriaan Niess, XC/EYA2, 2024-08-27

SPI Parallel Read Operation on GBB (1722c)

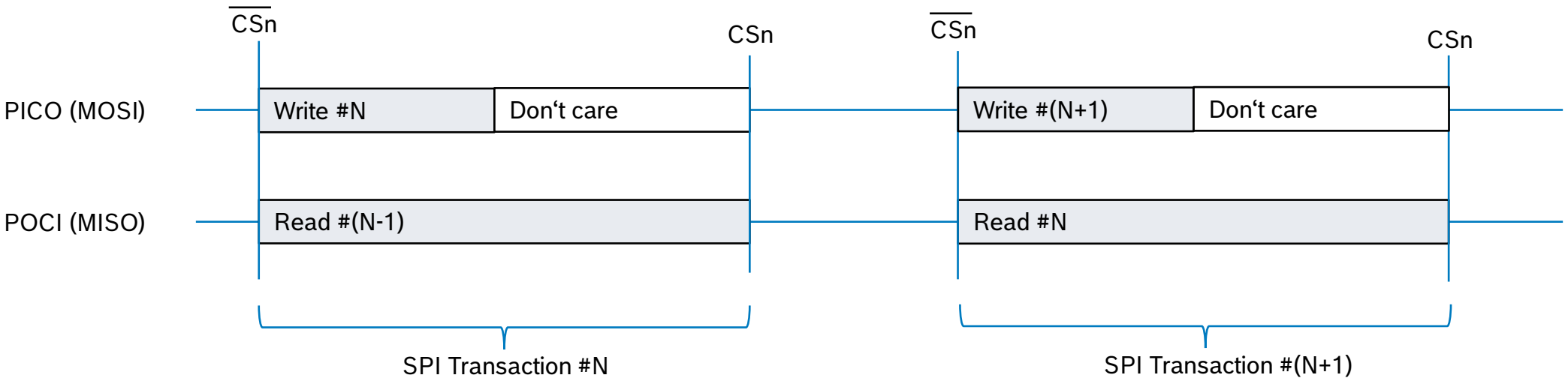
- I propose to support two versions of SPI Read in Annex O
  1. **Consecutive SPI Read**
  2. **Parallel SPI Read**

- The current Annex O draft already covers the Consecutive SPI Read (?)

- The proposal does not intend to fundamentally change the existing proposal, only to extend it with an additional Parallel SPI Read operation using GBB

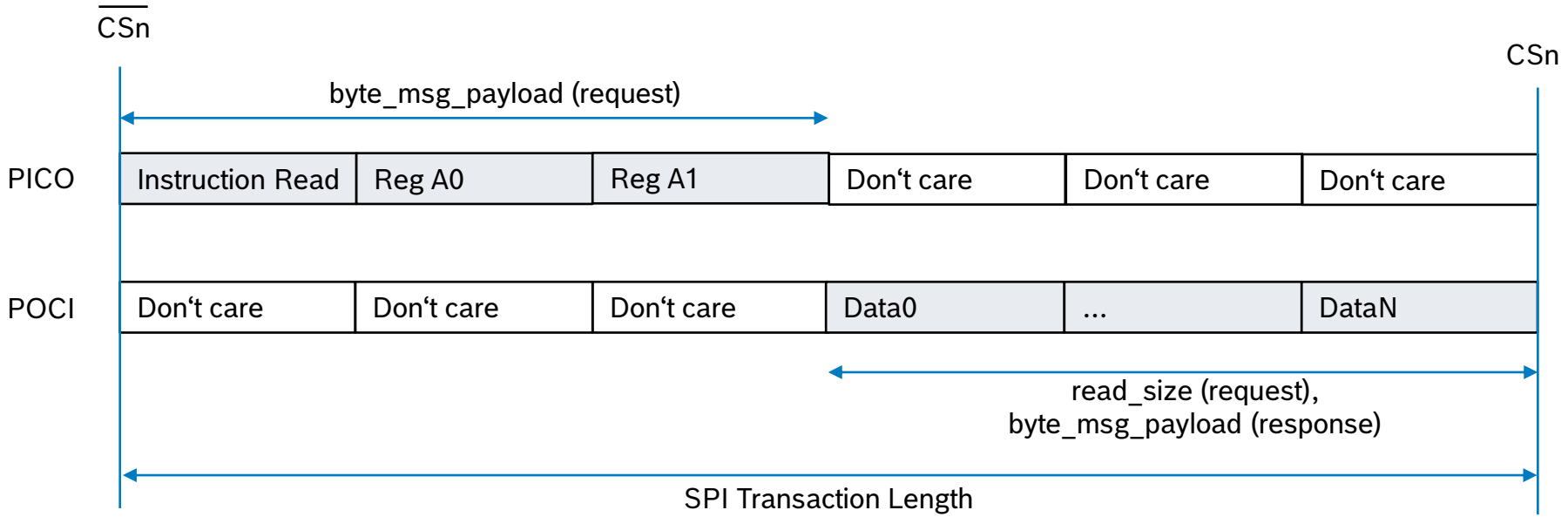
Why the need for a Parallel SPI Read?

- Certain SPI devices can operate in a pipelined mode



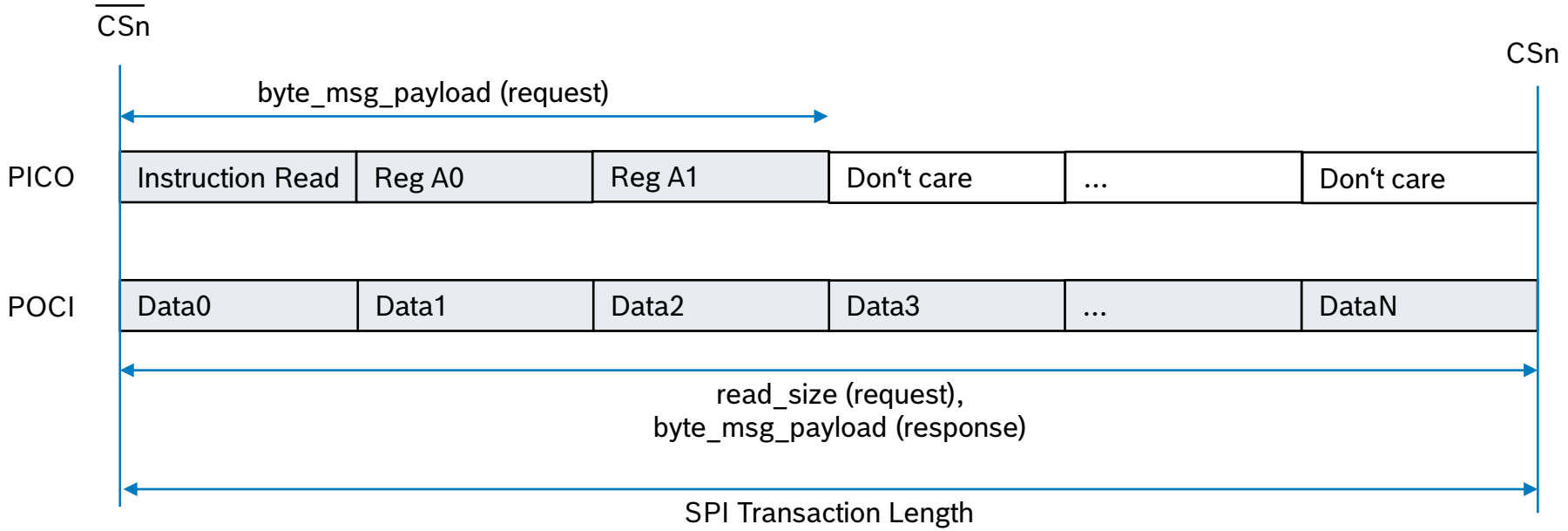
Note: The hs-bit (or any other available bit) could be used to differentiate between two kinds of Read-Operation (see following slides)

Consecutive SPI Read (1722b)



- Proposal: If op==0 and hs==0 a **Consecutive Read** operation is performed
- Total length of SPI transaction is (byte_msg_payload + read_size)
- **If hs==0, the GBB read_size is aligned to the first byte following the end of the write phase**
- To my knowledge, this is basically Hari‘s proposal(?) and should cover >90% of use cases

Proposal: Parallel SPI Read (1722c)



- Proposal: If op==1 and hs==1 a **Parallel Read** operation is performed
- Total length of SPI transaction can be calculated as MAX(request_payload_size, read_size)
- **If hs==1, the GBB read_size is aligned to the first byte of the SPI transaction**

Comparision Consecutive/Parallel Read

- **Let‘s assume we only support the (1) Consecutive SPI Read operation**
  - **Pro:** Majority of SPI devices use consecutive read phases and would be supported with good bandwidth efficiency on GBB
  - **Con**: Pipelined SPI devices or devices that read/write data in parallel wouldn‘t be supported.

- **Let‘s assume we only support the (2) Parallel SPI Read operation**
  - The Consecutive Read operation can be generalized with the Parallel Read operation.
  - **Pro**: The Parallel Read can cover all SPI devices!
  - **Con**: But this would imply very bad bandwidth efficiency for the Consecutive Read which most devices use (lot‘s of don‘t care data transferred over GBB)

- **Both cases on their own have significant drawbacks. That‘s why I would like to support both the Consecutive and Parallel Read operations in Annex O!**

BOSCH