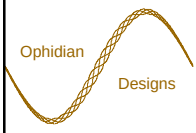
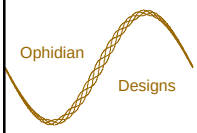


Overview

- CRC calculated on Data phase contents
- CRC returned in separate CRC phase
  - Immediately follows Data phase, CRC is not saved between phases
  - Data recipient returns CRC to sender, opposite from data flow direction
  - Variant of Data phase
- Negotiated / enabled by bit in IUTR

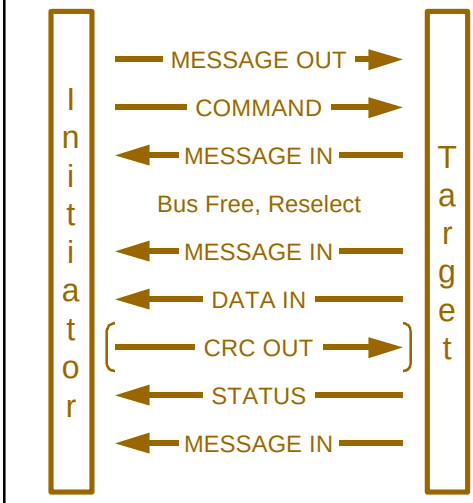


Overview

- Target controls CRC, like other phases
- Target inserts CRC whenever / wherever / whether it pleases
- No initiator context (responds to target)
- Target context: one bit per initiator
- Disadvantage: one extra phase change per command (typical)

Ophidian Designs

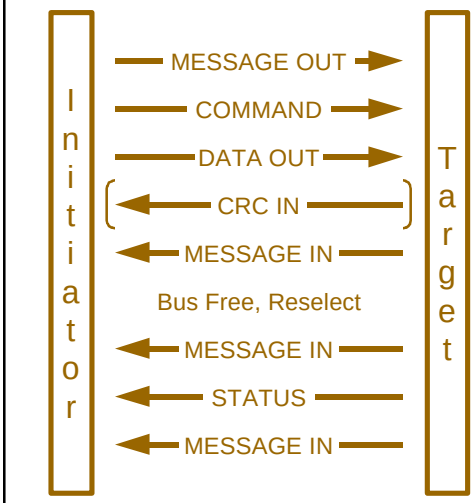
Read Example



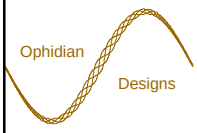
- CRC OUT phase is new
- All other phases unaltered
- DATA IN, CRC OUT pairs might be repeated for multiple CRCs

Ophidian Designs

Write Example

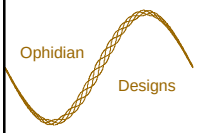


- CRC IN phase is new
- All other phases unaltered
- DATA OUT, CRC IN pairs might be repeated for multiple CRCs



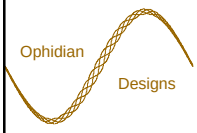
CRC Phase Signals

- CRC phase overloaded with Data phase
- C/D, MSG negated, I/O asserted:
  - CRC IN if previous phase was DATA OUT
  - DATA IN otherwise
- C/D, MSG, I/O negated:
  - CRC OUT if previous phase was DATA IN
  - DATA OUT otherwise



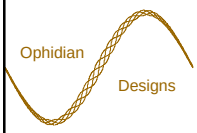
Initiator Rules, Reads

- During DATA IN, calculate CRC
- If next phase is CRC OUT (DATA OUT)
  - pad CRC data to four byte boundary
  - return CRC to target
- Discard / reset CRC



Initiator Rules, Writes

- During DATA OUT, calculate CRC
- If next phase is CRC IN (DATA IN)
  - pad CRC data to four byte boundary
  - receive CRC from target
  - if CRC doesn't match, raise ATN before asserting last ACK (same timing as parity error on last byte of CRC)
- Discard / reset CRC



Target Rules

- After each DATA IN/OUT phase, insert CRC OUT/IN phase if enabled.
- If CRC received during CRC IN doesn't match, report Check Condition with CRC error ASC/ASCQ.
- If Initiator raises ATN during CRC OUT, report Check Condition with CRC error ASC/ASCQ.