P802.3ae Serial Jitter Test Pattern Ad-Hoc Summary

Ben Brown

Ad-Hoc Chair

10-May-2001

Motivations - Why not use the $1+x^{28}+x^{31}$ PRBS?

- Use a “real life” pattern (some PLLs may expect/require sync bits)
- Use existing logic ($1+x^{39}+x^{58}$ scrambler), easy to generate
- Focus on patterns that occur once in a given period of time (week/day/hour?)
- One that is sufficiently stressful to test PLL designs

Tests to be supported

- Tests with mixed frequency pattern
  - Jitter
  - Eye Mask
  - RX Sensitivity/Saturation
  - Stressed RX Sensitivity
- Tests with square wave
  - Rise/Fall Time
  - ER/OMA
- Tests that can be done either way
  - Optical power
  - Return loss*
  - Encircled flux
  - RIN
  - Spectral Width/Center/Side Mode Suppression

*Doesn’t really require a pattern

Square Wave Pattern

- A single frequency is adequate
- Decided upon a range to suit different implementation styles
- Minimum 4 bits high/4 bits low
- Maximum 11 bits high/11 bits low
- Repeat until mode change
- Transmit tests only
- No need to capture at receiver

LAN vs. WAN

- This discussion is LAN Serial only
- Possible WAN solutions:
  - – Use LAN pattern as payload (no scramble)
    - 11 WIS frames for an integral number of 66-bit blocks may be too large for a BERT
  - – Use LAN pattern along with A1/A2 only
    - Requires significant(?) changes to Clause 50
  - – Use pattern specified in ITU-T G.957
    - Leave unspecified in standard, perhaps with a note pointing to G.957 (?)

Ideal Jitter Pattern “Must Have”s

- Short enough to fit into a BERT
- Several content types:
  - Long run length with no transitions
  - Long run lengths with few transitions
  - Rapid change in transition density
  - Extreme running disparity
  - Rapid change in running disparity
  - Polarity biased, stressing a particular data edge

“Must Have”s (cont.)

- Ability to generate and check the pattern in the same device at the same time
- Error counter attributes:
  - Sticky at max
  - Can be reset for new measurement
  - Count only once per 66-bit block
- Ability to synchronize in the presence of errors - at 10^{-5} or perhaps even 10^{-3}

“Nice to Have”s

- Ability to measure BER “down” to a particular value
  - – 8-bit counter supports 10^{-8} when read once per second
  - – Higher BERs might require a wider counter or more frequent reads
- Pattern should not be so stressful that EMI is compromised

Methodology: Common to Original Proposals

- Use the existing $1+x^{39}+x^{58}$ scrambler
- Seed the scrambler with start values
- Operate the scrambler for a finite time
- Use 66-bit blocks with fixed sync header
- Count errors at the receiver

Methodology: Agreed upon by Ad-Hoc participants

- Fixed pattern length: 8448 bits (128x66)
- 2 Seeds - “typical” & “atypical” patterns
- Always invert the patterns in the following format:
  - Seed A
  - Seed A Invert
  - Seed B
  - Seed B Invert
- Registers common between TX & RX

Methodology: Still pending

- Do we need to make the data input to the scrambler programmable? To what level of flexibility?

Continuing work

- Specify what the pattern contents actually need to be
- Search for seeds/data inputs that will generate the associated patterns

HELP

- Once we find appropriate patterns, we can make them available on the web site (hex/ascii)
- We need individuals that can actually test these patterns