Comment #45 : PRBS31 verifier count rate

Inphi

Think fast.

Andre Szczepanek : HSZ Consulting/Inphi

Supporters

- Piers Dawe
  - This comment is a re-iteration of an unresolved comment #253 from draft 2.1
    - Comment #70 (Piers Dawe) is a pile-on to that comment
- Francis Ho (Inphi)
- Chris Cole (Finisair)

Comment #45

- There is no limit to the potential increment rate of the PRBS31 checker referenced in 49.2.12.
- The checker implementation is difficult to match at high increment rates or in the presence of burst errors (the source synchronous descrambler implementation error multiplication factor depends on burst pattern).
- There will be less scope for a complex implementation in a PMA device versus a PCS.
- For most practical purposes stringent matching of the 49.2.12 implementation is not necessary. It would be sufficient to match the result of a 49.2.12 implementation only for isolated single bit errors and at errors rates less than 1 in a thousand.

The Clause 49.2.12 PRBS31 checker

The PRBS31 pattern error checker is self-synchronizing. It compares each bit received to the result of the PRBS31 generator based on the prior 31 bits received. It shall produce the same result as the implementation shown in Figure 49–11. When no errors occur, the PRBS31 pattern error signal will be zero. When an isolated bit error occurs, it will cause the PRBS31 pattern error signal to go high three times; once when it is received and once when it is at each tap. The test-pattern error counter shall increment once for each bit time that the PRBS31 pattern error signal is high.

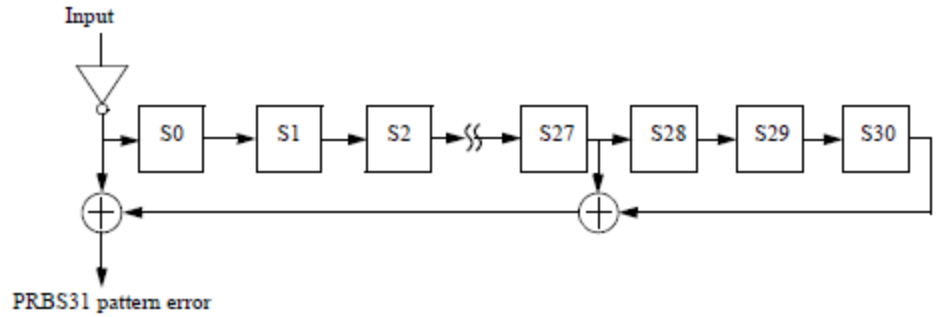


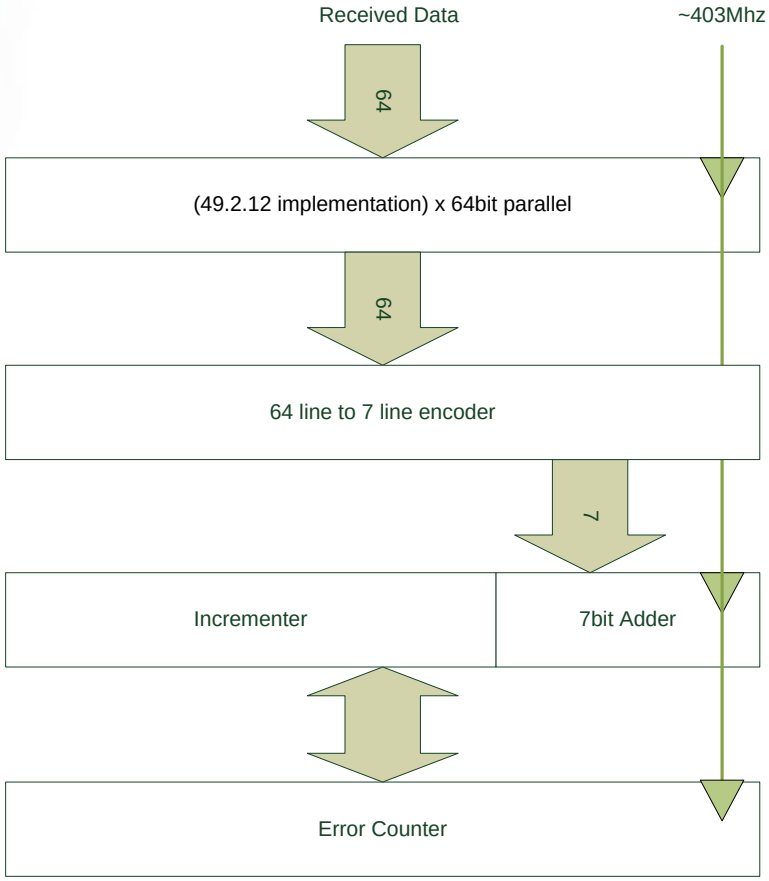
Figure 49–11—PRBS31 pattern checker

- There is no flexibility in the 49.2.12 pattern checker implementation clause
  - Compliance requires an exact match to this implementation, even at BERs worse than 1e-3
- For single bit errors, each causes 3 pattern error counts
  - This is not the way PRBS31 verifiers are normally implemented in SERDES or BERTs
  - The count from bursts depends on the position of errored bits in the burst (error cancellation can occur)

Implementation Considerations

- It is not practical to implement Figure 49-11 at 25.8Ghz. It has to be parallelized.
  - A 64bit parallel version would run at ~403Mhz
    - For low BERs between 0 and 3 bit errors may require detecting and counting per cycle
    - For high BERs or bursts, 0 to 64 bit errors may require detecting and counting per cycle
  - Up to 64 errors must be detected and counted per cycle, worst case
    - This is not impossible to do, just highly complex !
      - This logic complexity may be acceptable in a multi-million gate PCS/MAC chip, but is a huge overhead for a PMA device.
    - What system advantage is gained by requiring this complexity ?

A Fully Compliant Implementation

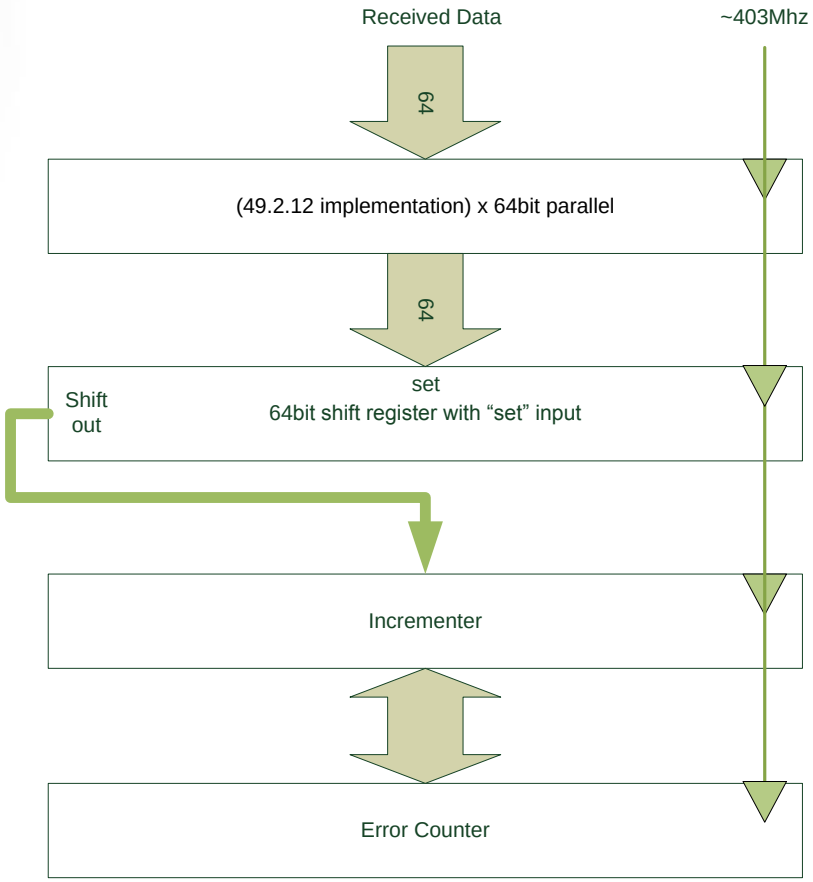


- The 64line to 7 line Encoder is extremely complex
- However it can be sub-divided eg
  - Eight 8 line to 3 line encoders followed by a hierarchy of adders to create the 7 line sum

What do we really need in a PMA ?

- Do we really want an error count that is 3x the BER ?
- Is accurate error counting at BERs worse than 1e-3 required ?
  - I believe not
- Is accurate counting of burst error bits required ?
  - Figure 49-11 does not do this anyway !
- For most practical purposes stringent matching of Figure 49-11 is not necessary.
  - It would be sufficient to match the result of Figure 49-11 only for isolated single bit errors and at errors rates less than 1 in a thousand.

A Pseudo-Compliant Implementation



- Matches the results of a compliant implementation if :
  - Single bit errors are more than 64 bits apart
  - Burst errors are less than 64bits long and more than 128bits apart
    - Or if there are no burst errors
- Considerably less complex than a fully compliant implementation
- Can be "improved" with techniques such as deference latching
- This is just an example : other pseudo-compliant implementations are possible

What can we do?

- Define our own PRBS31 verifier implementation
  - This is a lot of work, and including it before sponsor ballot would not be practical
- Retain the reference to 49.2.12, with exceptions
  - This maintains backward compatibility with 10G
  - Legitimizes real-world designs
  - Does not provide a clean solution to the 3x over-count issue
- Remove the reference to 49.2.12
  - Reference only the pattern (as defined in 49.2.8)
  - Limit compliance requirement to counting isolated bit errors at least one thousand bits apart.

Improved Comment #45 remedy

Replace:

(see 49.2.12)

With:

The checker shall increment the test pattern error counter by one for each incoming bit error in the PRBS31 pattern (see 49.2.8), for isolated single bit errors at least a thousand bits apart. Otherwise the checker should attempt to count all incoming bit errors.