An optimized self-test function for 64b/66b PCS

Pat Thaler

Agilent Technologies

pat_thaler@agilent.com

Rick Walker

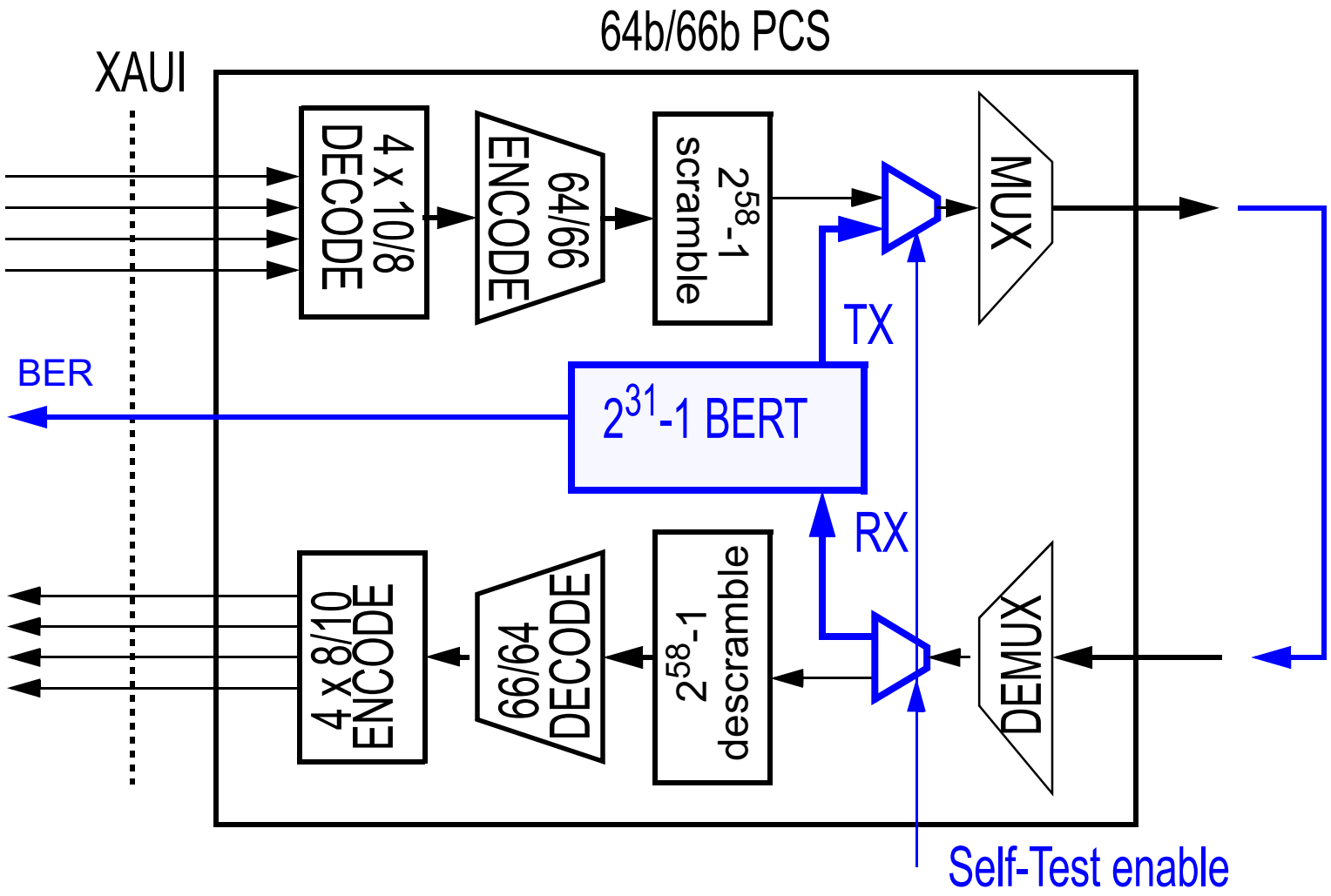
Agilent Laboratories, Palo Alto, CA

rick_walker@agilent.com

Outline

- obvious BERT implementation
- reusing existing scrambler
- implementation details
- simulation results
- summary

the obvious way to do self-test



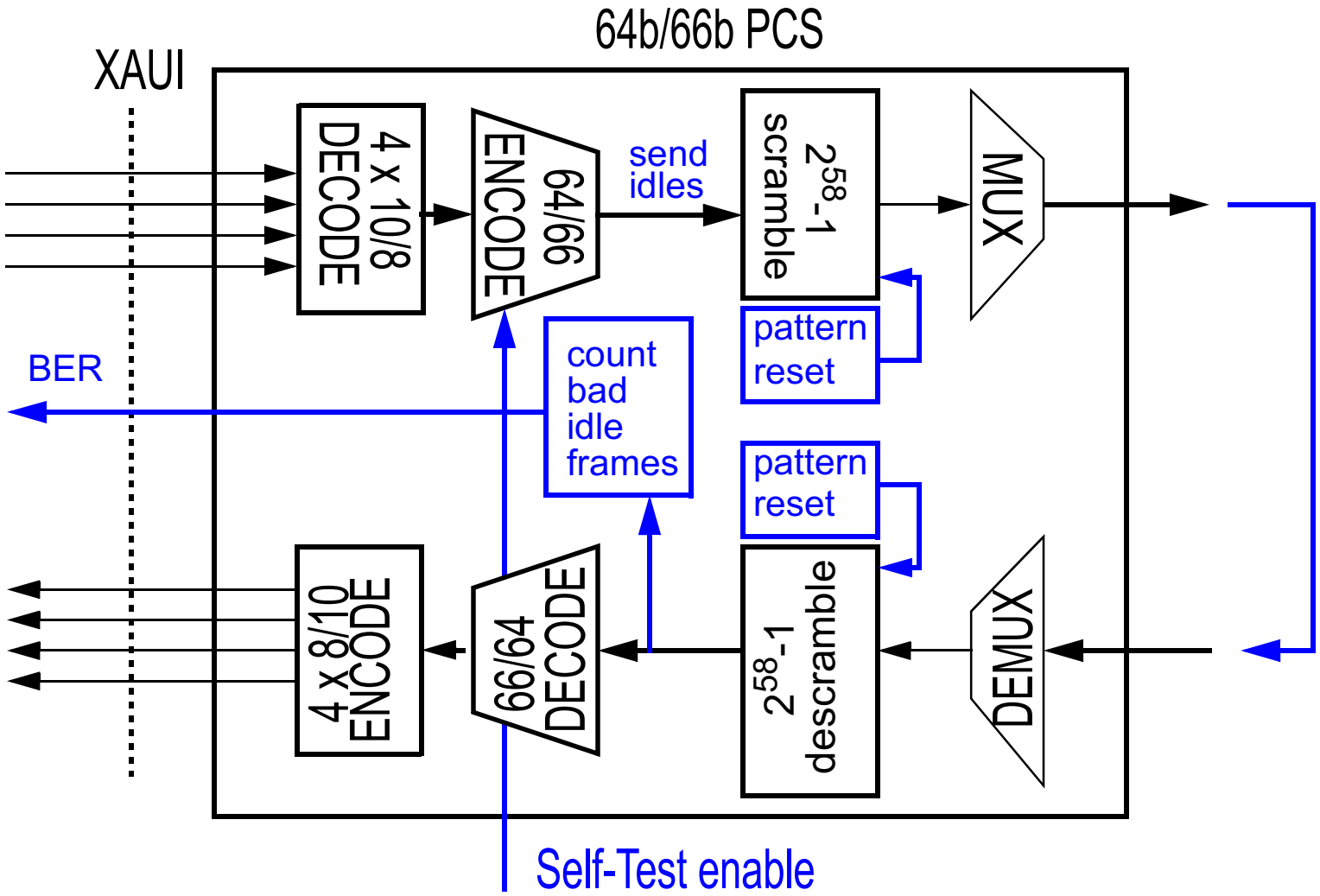
Attributes of the obvious method

+ Pattern repeats every 0.21 seconds for quick, repeatable testing
+ Pattern is compatible with standard BERT equipment
- Test sequence is *not* compatible with other 64b/66b receivers not in self-test mode
- Extra gates are needed to build a completely independent BERT test mechanism
- Limited to 31 bit run-length. This run length occurs in normal 64b/66b data more often than once in 10^{10} bits so error floor is not well tested.
- Test pattern lacks sync headers which may affect receiver performance.

Strategy

- The 58 bit scrambler *is* already 90% of a BERT implementation, so why not reuse it to save gates?
- Use a normal scrambled 64b/66b idle frames as the self-test pattern
- Reset the scrambler state at start of test to a pattern computed to produce a 46 bit run-length
- Monitor scrambler state and reset both TX and RX scramblers to ensure a 0.21 second cycle time
- Look at errors in the *descrambled* Local Fault frames to estimate error rate

An optimized self-test method



Implementation

Initialize the 58-bit scrambler registers to:

```
10101111 11111111 11100000 10111111 11111100 00000111 11111111 10
^--- oldest bit of state          youngest bit of state ---------^
```

Force coder to send Local Fault pattern into scrambler

```
0x55 0x00 0x00 0x01 0x00 0x00 0x00 0x01
```

Transmitter then produces (serialize bytes LSB first):

```
          0: ‘10’ 00 00 00 00 00 e0 ff 00  <--- 46 bit run length
         64: ‘10’ 55 00 f0 7e 00 d5 03 2d
        128: ‘10’ 6b 40 11 81 c2 3a 14 25
                          ...
2147483456: ‘10’ 13 99 44 1c 85 ca 3b 6c
2147483520: ‘10’ bf 50 94 08 9c b0 98 b6
2147483584: ‘10’ 13 1f 7a 3d 19 eb d5 f2
2147483648: ‘10’ b7 64 00 0f 55 0c f9 5e
2147483712: ‘10’ 40 2b ba 29 1e 44 6e dd  <--- reset scrambler here
(patterns given allow up to five pipeline stages for reset logic)
```

Pattern length is set to ~2^{31} - 1 to give a deterministic 0.21sec. test period

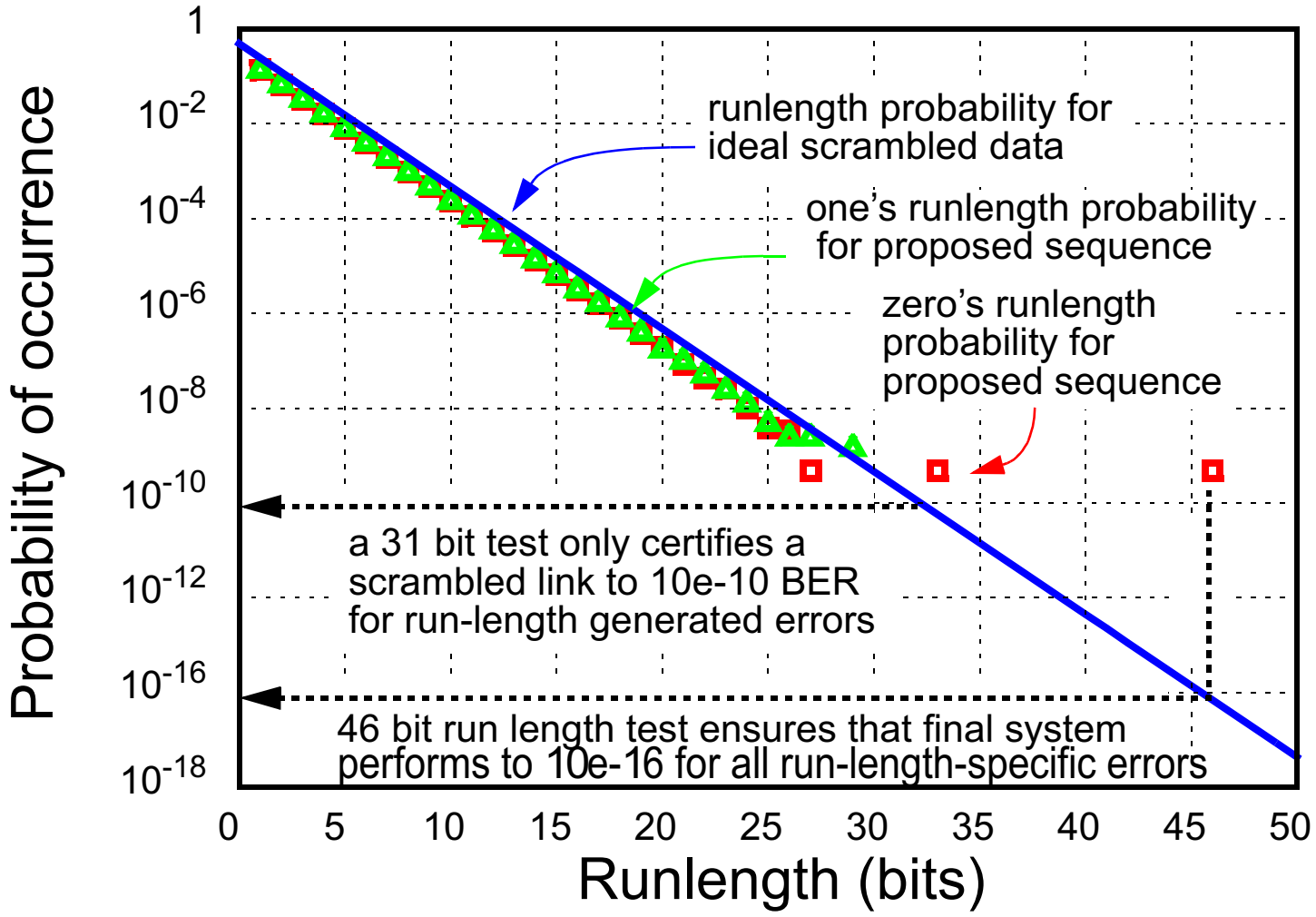
Counting errors

Under error-free conditions, the receiver descrambler will output the transmitted Local Fault pattern:

0x55 0x00 0x00 0x01 0x00 0x00 0x00 0x01

XOR-ing the actual received data bits with this expected pattern will give an error mask. The errors can be OR-ed together to compute a block error-rate, or added up and divided by 3 (the scrambler error-multiplication factor) to closely approximate the exact link BER.

Quality of test



Summary

- Proposed method optimizes gate count and complexity by reusing existing circuit
- Proposed method tests to 46-bit run length to ensure absence of run-length-induced error floor
- Proposed method holds pattern length to ~2^{31}-1 bits to ensure a rapid repeatable test result
- Proposed test pattern is a normal 64b/66b Local Fault frame, and is gracefully handled by downstream circuitry
- Algorithm can be modified to produce:
  any peak run-length up to 57 bits
  shorter pattern length
  to alternately produce runs of *both* zeros *and* ones.