Additional Text Changes for Comment #84 and #85

September 10, 2019

Mike Tu tum@broadcom.com

149.3.2.2.18 to 149.3.2.2.21

Motivation

- Multiple variables are used as indices of symbol numbers in 149.3.2.2.18 to 149.3.2.2.21
  - 149.3.2.2.18: used “n” without definition
  - 149.3.2.2.19: used “j” without definition
  - 149.3.2.2.20: used “j” without definition
  - 149.3.2.2.21: used “u” with definition
- Variable “n” also used in 149.3.4 and 149.3.5 as index of symbol numbers
- There are also redundant paragraphs in 149.3.2.2.19
- Recommend to clean up the texts and use “n” as the index of symbol numbers in 149.3.2.2.19, 149.3.2.2.20, and 149.3.2.2.21

Suggested Remedy – 149.3.2.2.18

- Page 101, line 35-37. Make the following changes:
  - "… and each pair of bits, $D_n[0]$ and $D_n[1]$, where *n* is an index indicating the symbol number, is scrambled using an additive scrambler. For each pair of interleaved bits, …"
- Page 101, line 41-42:
  - "… (MSB) to generate two scrambled data bits $\{A_n, B_n\}$ as shown in Equation (149–4)."
- Page 101, Equation 149-4:
  - $A_n = DS_n[0] \oplus D_n[0]$
  - $B_n = DS_n[1] \oplus D_n[1]$

Suggested Remedy – 149.3.2.2.19

- Page 101, delete line 49 to 51 (this paragraph is redundant)
- Page 101, line 52-54, change to:
  - "For output symbols, the PCS transmit process shall map consecutive pairs of bits, {A_n, B_n}, where A_n is the bit arriving first and *n* is an index indicating the symbol number, to Gray-coded symbols G(*n*) with one of four levels as follows:"
- Page 102, delete line 9 and 10 (this paragraph is redundant)
- Page 102, line 12 and 13:
  - "For input symbols, the PCS receive process shall map Gray-coded symbols G(*n*) with one of four levels to pairs of bits {A_n, B_n}, where A_n is considered to be the first bit, as follows:"

Suggested Remedy – 149.3.2.2.20

- Page 102, line 38-39:
  - "For each Gray-coded symbol G(*n*), a precoded symbol P(*n*) shall be determined by the following algorithm, where *n* is an index indicating the symbol number:"
- Page 102, line 41 to 47:
  - P(*n*) = G(*n*), when precoder_type = No precoder,
  - P(*n*) = (G(*n*) + P(*n*–1)) mod 4, when precoder_type = Precoder for 1 – D Channel,
  - P(*n*) = (G(*n*) – P(*n*–1)) mod 4, when precoder_type = Precoder for 1 + D Channel, and
  - P(*n*) = (G(*n*) + P(*n*–2)) mod 4, when precoder_type = Precoder for 1 – D^2 Channel.

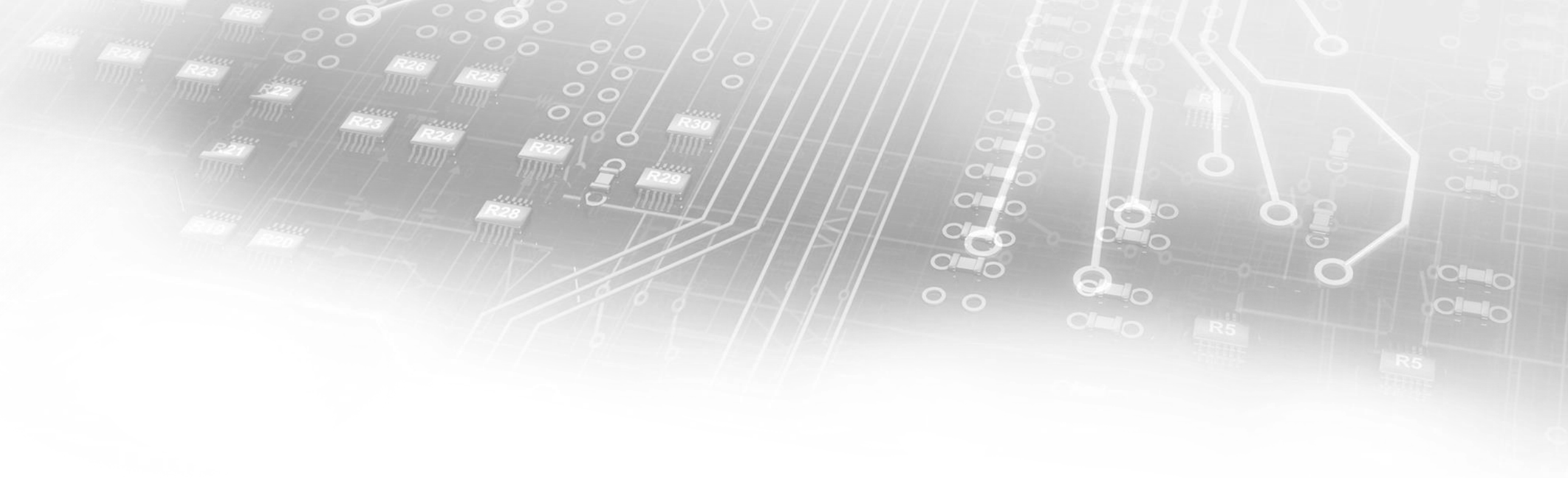
Suggested Remedy – 149.3.2.2.21

- Page 102, line 6-8:
  - "The PAM4 encoded symbols are denoted M(*n*), where *n* is an index indicating the symbol number."
- Page 102, line 10 to 11:
  - "Each consecutive precoder output symbol, P(*n*), is mapped to one of four PAM4 levels and assigned to the PAM4 encoder output M(*n*)."
- Page 102, line 13:
  - "Mapping from the precoder output symbol P(*n*) to a PAM4 encoded symbol M(*n*) is as follows:"

149.3.2.2.17

Regarding 149.3.2.2.17

- Page 99, line 40 to 45
  - Variable “n” is the length of RS-FEC codeword
    - Never referenced anywhere else as the length of RS-FEC.
    - “n” also used in other places as index indicating the symbol number.
  - Variable “k” is the number of data symbols per RS-FEC codeword
    - Never referenced anywhere else as # data symbols.
  - Variable “t” is number of correctable symbols per RS-FEC codeword
    - Never referenced anywhere else as # correctable symbols.
    - “t” also used in 149.4.3.1 to represent time (continuous).
- Recommend to delete “n”, “k”, “t” from the paragraph.
- Modify page 99, line 40 to 45:
  - The group of 3260 bits are encoded using a Reed-Solomon encoder operating over the Galois Field GF(2^{10}) where the symbol size is 10 bits. The encoder processes ~~k =~~ 326 ten-bit RS-FEC message symbols to generate ~~2t =~~ 34 ten-bit RS-FEC parity symbols, which are then appended to the message to produce a codeword of ~~n = k + 2t =~~ 360 ten-bit RS-FEC symbols. For the purposes of this clause, the particular Reed-Solomon code is denoted ~~RS-FEC(n,k), and the particular Reed-Solomon code is~~ as RS-FEC(360,326).



THANK YOU