

A large, powerful ocean wave with a surfer riding the base of the curl. The wave is a deep green color, and the surfer is silhouetted against the white foam of the wave's base. The sky is a pale, hazy blue.

512/513b

AQUANTIA[®]

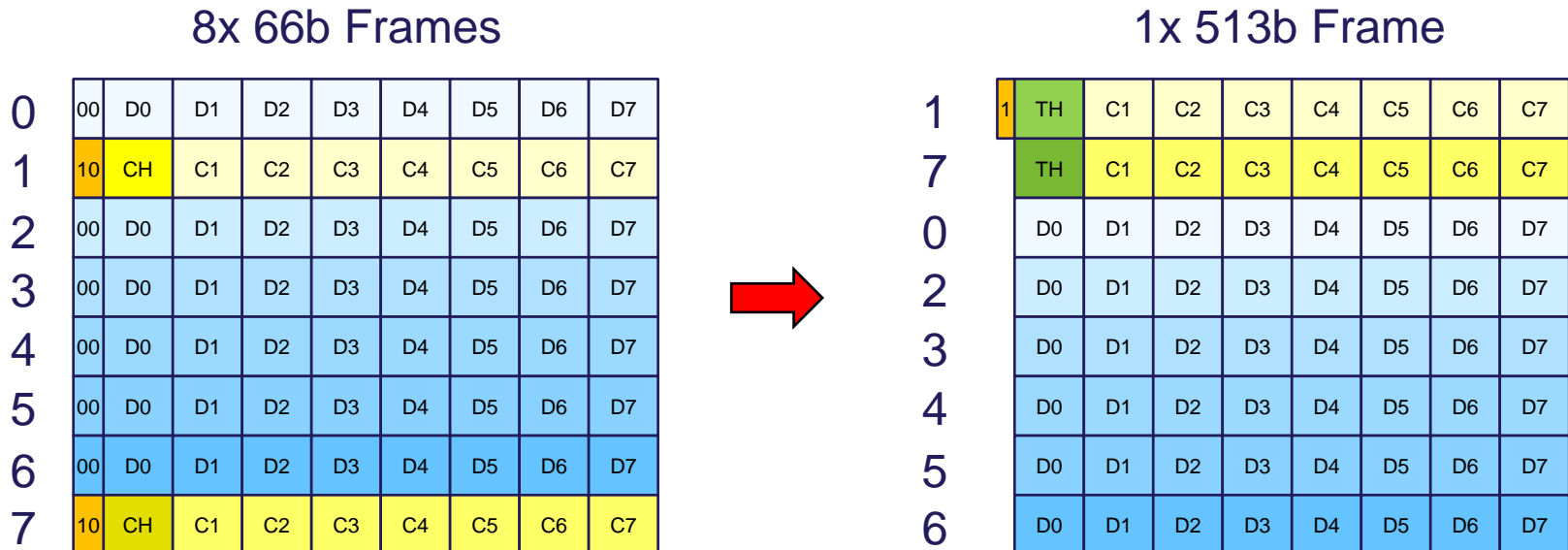
September 12, 2015

IEEE P802.3bq

Dr. Paul Langner

Overview

- 512/513b coding takes 8x 64/66b blocks (8x XLGMII transfers) and transports them with fewer total bits by virtue of combining the individual control/data frame bits
- Current baseline proposal collects the all 66b control frames within the block of 8x 66b frames and puts them at the beginning of the block, and the data frames at the end of the block



Overview (continued)

- Alternative proposal places only the control bytes for the 8 blocks at the beginning of the 513b frame, and leaves the payload for the 66b frames intact

(http://www.ieee802.org/3/bq/public/may15/wangz_3bq_01_2015_05.pdf)

8x 66b Frames

0	00	D0	D1	D2	D3	D4	D5	D6	D7
1	10	CH	C1	C2	C3	C4	C5	C6	C7
2	00	D0	D1	D2	D3	D4	D5	D6	D7
3	00	D0	D1	D2	D3	D4	D5	D6	D7
4	00	D0	D1	D2	D3	D4	D5	D6	D7
5	00	D0	D1	D2	D3	D4	D5	D6	D7
6	00	D0	D1	D2	D3	D4	D5	D6	D7
7	10	CH	C1	C2	C3	C4	C5	C6	C7



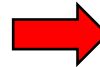
1x Alternative 513b Frame

	0	1	2	3	4	5	6	7
1	TH	TH	D0	D0	D0	D0	D0	D0
0		D1	D2	D3	D4	D5	D6	D7
1		C1	C2	C3	C4	C5	C6	C7
2		D1	D2	D3	D4	D5	D6	D7
3		D1	D2	D3	D4	D5	D6	D7
4		D1	D2	D3	D4	D5	D6	D7
5		D1	D2	D3	D4	D5	D6	D7
6		D1	D2	D3	D4	D5	D6	D7
7		C1	C2	C3	C4	C5	C6	C7

Overview (continued)

1x Alternative 513b Frame

	0	1	2	3	4	5	6	7
1	TH	TH	D0	D0	D0	D0	D0	D0
0		D1	D2	D3	D4	D5	D6	D7
1		C1	C2	C3	C4	C5	C6	C7
2		D1	D2	D3	D4	D5	D6	D7
3		D1	D2	D3	D4	D5	D6	D7
4		D1	D2	D3	D4	D5	D6	D7
5		D1	D2	D3	D4	D5	D6	D7
6		D1	D2	D3	D4	D5	D6	D7
7		C1	C2	C3	C4	C5	C6	C7



1x Alternative 513b Frame in 8x8 format

	0	1	2	3	4	5	6	7
1	TH	TH	D0	D0	D0	D0	D0	D0
	D1	D2	D3	D4	D5	D6	D7	C1
	C2	C3	C4	C5	C6	C7	D1	D2
	D3	D4	D5	D6	D7	D1	D2	D3
	D4	D5	D6	D7	D1	D2	D3	D4
	D5	D6	D7	D1	D2	D3	D4	D5
	D6	D7	D1	D2	D3	D4	D5	D6
	D7	C1	C2	C3	C4	C5	C6	C7

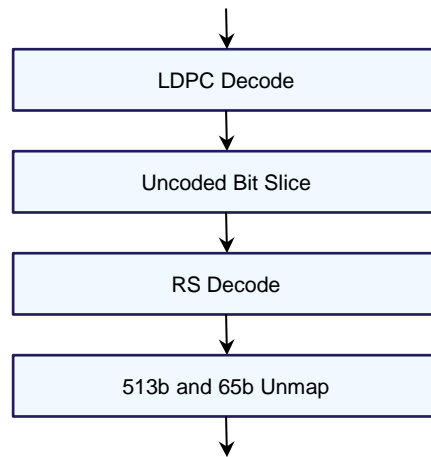
Summary of Differences

- In the Tx must wait until the last 66b frame is received before outputting the 513b frame, and both must store at least 7x 66b frames
- In the Rx, the entire frame has to be stored for block-based decode, so all bytes are simultaneously available
 - Delay is identical
- Using a 64-bit datapath, the complexity of both schemes are also identical, as both require 8x 1:8 byte MUXs in both Tx and Rx
 - Each 66b input and output byte comes from, or goes to, 1 of 8 different locations in the frame
 - The total gate count of these MUXs is roughly 4K gates (Tx + Rx) which is insignificant in a chip of this size (28nm ~1M gates/mm²)

Additional Material

Receive

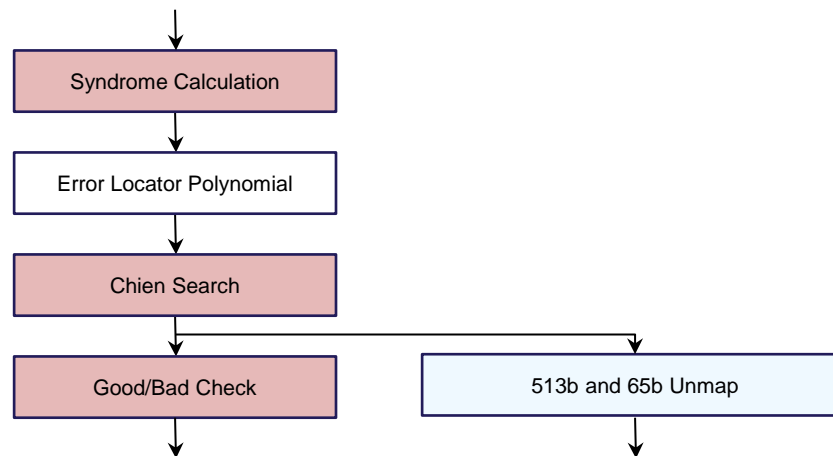
- In the receive process, the entire process (regardless of which 513b scheme is employed), must wait until LDPC decode of the 3584-bit block is done, followed by the uncoded bit slicing, RS decode, and then 513b and 65b unmapping



- Since all bytes are simultaneously available to the unmappers, there is no delay difference

Delay Depends on RS Decode

- Because of linearity properties, it is easy to introduce parallelism into most of the RS decode, but the final result needs to wait until the corrections applied are verified to zero the syndrome



- Net result is that all bits from the RS decode are not “known good” until the end of the Chien search

Complexity

- In the Tx portion, both schemes require receipt of the last 66b frame before the 513b frame can be output, requiring storage of 513 bits
- In the Rx portion, since all bytes in the RS portion of the frame need to be stored until the Chien search and syndrome check are complete, this means the entire payload must be stored as the 513b and 65b blocks are carried in both the coded and uncoded portions of the frame.
- Consequently, the only complexity difference between the current scheme and the alternative scheme is in the number of MUXs required in the Tx and Rx mappers and unmappers

Complexity (continued)

- Current scheme (assuming 64 + 2 bit datapath):
 - Payload bytes are column aligned, but require row swapping in both the Tx and Rx
 - = 2x 8x 1:8 byte MUXs
- Alternative scheme (assuming 64 + 2 bit datapath):
 - Payload bytes are neither row nor column aligned, so both row and column multiplexing is required in both the Tx and Rx
 - The first bytes of the 66B frames come and go into 8 different locations in the first row = 1:8 byte MUX
 - The payload bytes again each are spread across 8 different row and column locations (see page 4) = 7x 1:8 byte MUX
 - = 2x 8x 1:8 byte MUXs

Conclusion

- Delay through the Rx is identical, as all bytes in the frame are simultaneously available
- Complexity is identical between the schemes in both Tx and Rx for a 64-bit datapath, and is insignificant