100G backplane PAM4 PHY encoding (revised)

IEEE P802.3bj

March 2012, Hawaii

Matt Brown – AppliedMicro

Sudeep Bhoja – Broadcom

Contributors and Supporters

- Ran Adee, Intel
- Stephen Bates, PMC-Sierra
- Will Bliss, Broadcom
- David Chalupsky, Intel
- Dariush Dabiri, APM
- Dan Dove, APM
- Howard Frazier, Broadcom
- Ali Ghiasi, Broadcom
- Ziad Hatab, Vitesse
- Dimitrios Giannakopoulos, APM
- Adam Healey, LSI
- Beth Kochuparambil, Cisco
- Kent Lusted, Intel
- Richard Mellitz, Intel
- Venkatesh Nagapudi, APM
- Vasu Parthasarathy, Broadcom
- Jamal Riani, Marvell

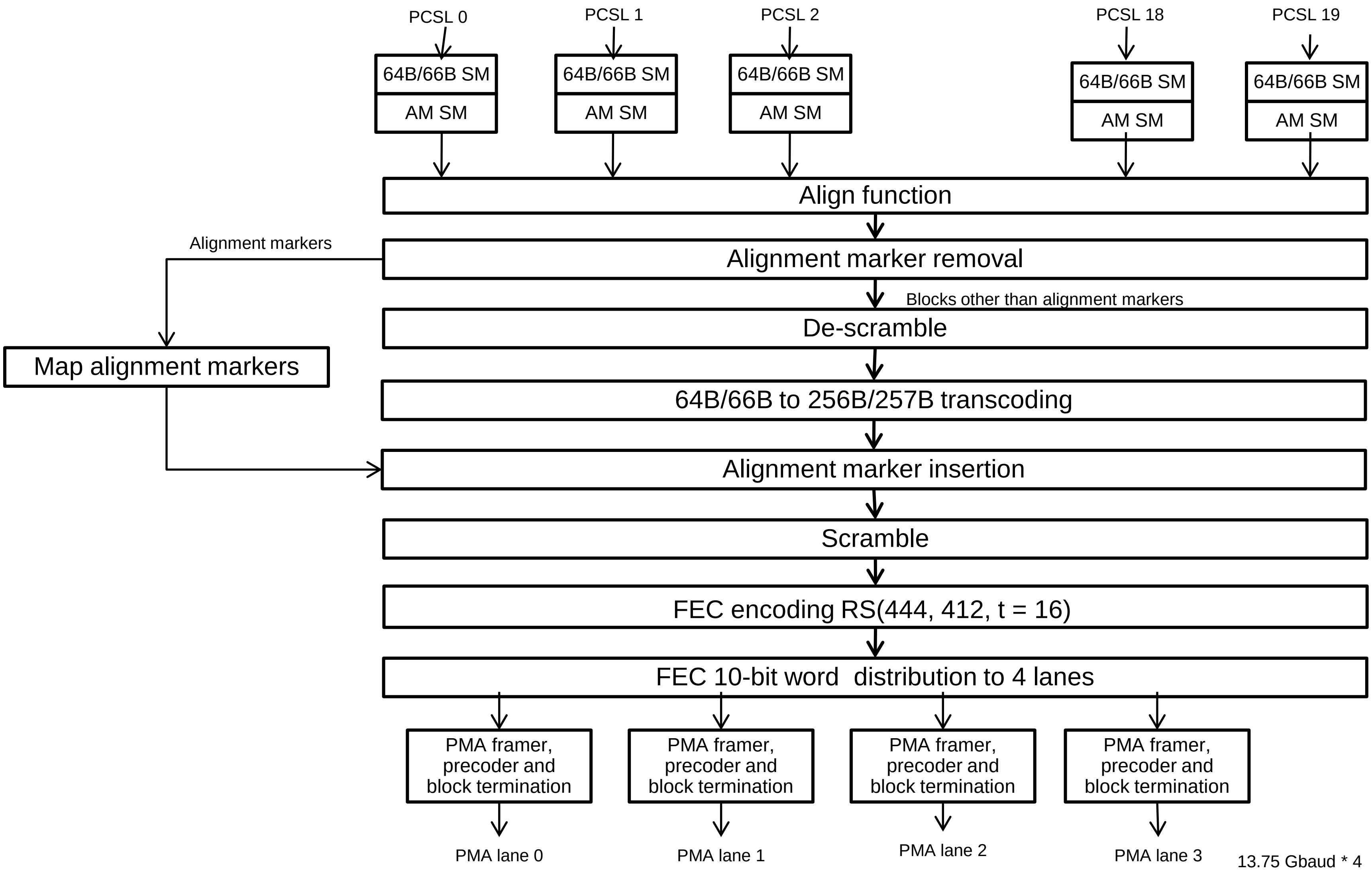
Introduction

- Provide a strawman baseline specification for the PAM4 FEC, PMA, and PMD transmitter encoding.
- Revised from January presentation to incorporate 256B/257B transcoding and alignment marker mapping.

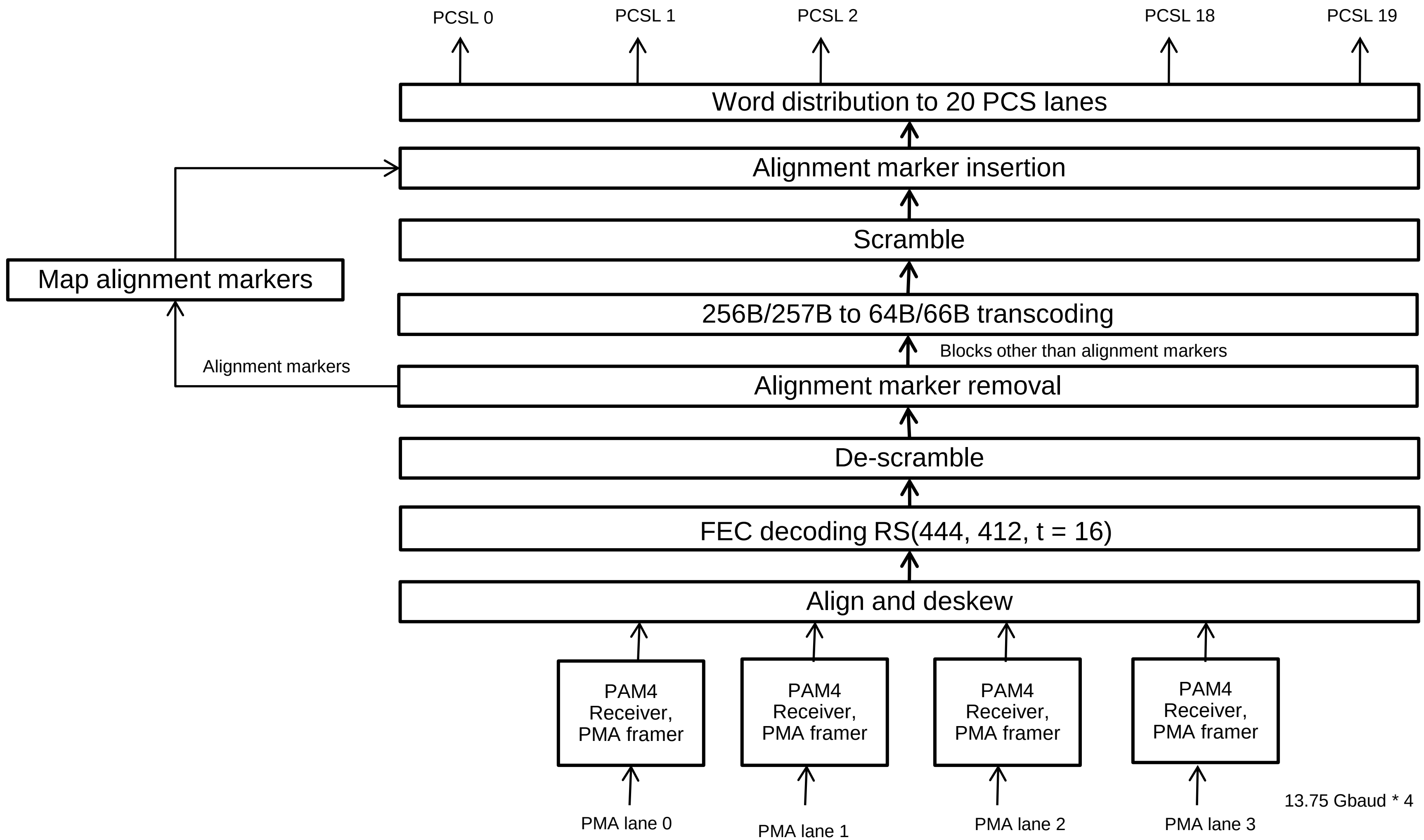
Transmitter process

- Transcoding: 256B/257B (was 512B/514B)
  - Aligns with NRZ (gustlin_01_0312)
- FEC: RS(444,412,T=16,M=10)
- PAM4 Symbols: Gray mapping,
  - {+1,+1/3,-1/3,-1} map to {10,11,01,00}
- Precoding: 1/(1+D) MOD 4
- PAM4 block termination: 1 PAM4 termination symbol per 32 PAM4 symbols
  - 63 data bits per 32 PAM4 symbols
- PAM4 symbol rate: 88 * 156.25 MHz = 13.75 Gbaud
- Tx pre-emphasis: 3 taps, one pre, one post
  - same structure as for 10GBASE-KR
- PAM4 test methodology and parameters addressed in bliss_01a_0911.

Tx encoding flow



RX decoding flow



PCS Lane Processing

- Synchronize to 64B/66B blocks on each PCS lane per 802.3ba 82.2.11.
- Synchronize to PCS alignment markers (64B/66B blocks) on each PCS lane per 802.3ba 82.2.12.
- Align (or deskew) and re-order PCS lanes based on alignment markers per 802.3ba 82.2.12.
- Descramble 64B/66B blocks per 82.2.15.
  - Required for transcoding.
- Same as for NRZ PHY.

Transcoding

- 256B/257B transcoding per cideciyan_01_0312.
- Map 64B/66B blocks to 256B/257B per gustlin_01_0312.
  - Alignment markers will not be transcoded, but instead will be re-mapped.
- Same as for NRZ.
- MTTFPA > 3.9E15 years
  - Post-FEC BER <= 1E-12, RS(444,412,16,10) FEC
  - Analysis on slide 36.
  - FYI Lifetime of universe ~= 13E9 years.

Scrambling

- Use self-synchronizing scrambler
  - Same scrambler as for PCS in 802.3ba 82.2.5.
  - All data bits including the 256B/257B header bits and alignment markers are scrambled.
- Same as for NRZ except…
  - Alignment markers are scrambled as well.
    - Need AM mapping to PAM4 to be balanced, randomized, and clock rich.
    - May be able to re-map AM's so that scrambling is not required.
      - Analysis required.
      - Ideally, re-mapping would be common to NRZ and PAM4.

FEC

- RS(444,412,T=16,M=10) code format
  - single, efficient, dual-purpose (NRZ/PAM4) FEC core is possible if FEC generator math specified similarly for both
- FEC frame content
  - correctable payload = 412*10 = 4120 bits
  - parity = 32*10 = 320 bits
  - data = 64x 64B/66B blocks transcoded to 16x 256B/257B blocks
    - total data = 4112 bits
    - 8 dummy bits (4120-4112) per FEC frame required
      - 8 zeros added (assumed) for parity calculation
      - Payload words 408-411 will contain 8 data bits and 2 dummy bits.
      - one 8-bit word will end up on each of the 4 PMA lanes
      - dummy bits not transmitted
- FEC encoding is mandatory; negotiation is not required.

13.75GBaud Precoding/FEC Summary

RS(444, 412, t = 16)	Delta (dB)	Coding Gain (dB) BER = 1E-15
Random Error		7.12
DFE Burst Error Penalty	-0.88	6.24
Extended KR channel 6.7% over clocking loss	-1.0	5.24 (<100ns total latency)

- ~6.7% over clocking (88*156.25 MHz)
- 5.24 dB Coding gain for Extended KR channel
- Overhead includes FEC parity & PAM4 block termination

11

Comparison of RS FEC candidate codes

FEC codes GF(2^10)	Total Coding Gain (dB)	Burst Coding Gain (dB)	Latency (ns)
RS(444, 412, t = 16)	5.24	6.24	82 - 123
RS(550, 520, t = 15)	5.1	5.9	102 - 154
RS(546, 520, t = 13)	4.9	5.6	102 - 154
RS(544, 520, t = 12)	5.0	5.6	102 - 154
RS(540, 520, t = 10)	4.9	5.2	102 - 154

- Codes in bhoja_01_0911 and cideciyan_01_1111 (found using computer search)
- RS(444, 412, t = 16) has best coding gain within 100ns target latency
  - Example implementation of 460K gates in 40nm CMOS has 99.9ns latency

Mapping 256B/257B blocks to FEC frame

- 256B/257B blocks are concatenated and organized into a series of 10-bit FEC words.
  - Except for last four FEC words which are 8 data bits with 2 pad bits each (see FEC slide).

FEC frame structure

first 256B/257B block starts here

40 bits

	to PMA lane 0	to PMA lane 1	to PMA lane 2	to PMA lane 3
0	tddddddddd	dddddddddd	dddddddddd	dddddddddd
1	dddddddddd	dddddddddd	dddddddddd	dddddddddd
…	…	…	…	…
11	dddddddddd	dddddddddd	dddddddddd	dddddddddd
12	dddddddddd	dddddddddd	dddddddddd	ddddtddddd
13	dddddddddd	dddddddddd	dddddddddd	dddddddddd
…	…	…	…	…
101	dddddddddd	dddddddddd	dddddddddd	dddddddddd
102	dddddddd00	dddddddd00	dddddddd00	dddddddd00
103	pppppppppp	pppppppppp	pppppppppp	pppppppppp
104	pppppppppp	pppppppppp	pppppppppp	pppppppppp
…	…	…	…	…
109	pppppppppp	pppppppppp	pppppppppp	pppppppppp
110	pppppppppp	pppppppppp	pppppppppp	pppppppppp

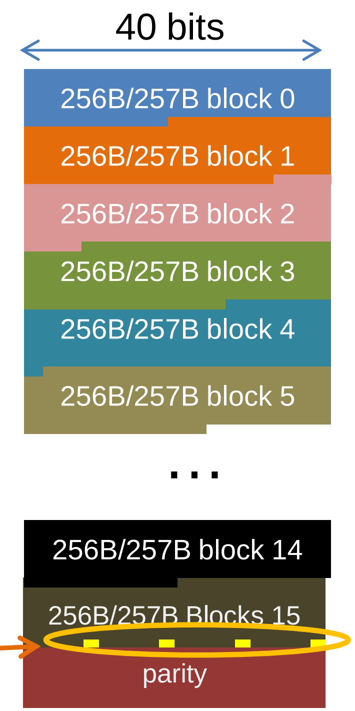
FEC payload (rows 0–102)

FEC parity (rows 103–110)

Third 256B/257B block starts here

dummy bits

drop dummy "0" bits in row 102



Legend:
"t" = 256B/257B header bit
"d" = 256B/257B data bit
"p" = FEC parity bit
"0" = temporary dummy bit

FEC frame structure

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="32">RS Symbol index</th></tr>
<tr><th colspan="2">0</th><th colspan="2">1</th><th colspan="2">2</th><th colspan="2">3</th><th colspan="2">4</th><th colspan="2">5</th><th colspan="2">6</th><th colspan="2">7</th><th colspan="2">8</th><th colspan="2">9</th><th colspan="2">10</th><th colspan="2">11</th><th colspan="2">12</th><th colspan="2">13</th><th colspan="2">14</th><th colspan="2">15</th></tr>
<tr><td rowspan="18">Row index</td><td>0</td><td colspan="32">B0</td></tr>
<tr><td>1</td><td colspan="18">B0</td><td>7b</td><td>3b</td><td colspan="12">B1</td></tr>
<tr><td>2</td><td colspan="32">B1</td></tr>
<tr><td>3</td><td colspan="6">B1</td><td>4b</td><td>6b</td><td colspan="24">B2</td></tr>
<tr><td>4</td><td colspan="26">B2</td><td>1b</td><td>9b</td><td colspan="4">B3</td></tr>
<tr><td>5</td><td colspan="32">B3</td></tr>
<tr><td>6</td><td colspan="12">B3</td><td>8b</td><td>2b</td><td colspan="18">B4</td></tr>
<tr><td>7</td><td colspan="32">B4</td></tr>
<tr><td>8</td><td>5b</td><td>5b</td><td colspan="30">B5</td></tr>
<tr><td>9</td><td colspan="20">B5</td><td>2b</td><td>8b</td><td colspan="10">B6</td></tr>
<tr><td>10</td><td colspan="32">B6</td></tr>
<tr><td>11</td><td colspan="6">B6</td><td>9b</td><td>1b</td><td colspan="24">B7</td></tr>
<tr><td>12</td><td colspan="26">B7</td><td>6b</td><td>4b</td><td colspan="4">B8</td></tr>
<tr><td>…</td><td colspan="32">…</td></tr>
<tr><td>22</td><td colspan="14">B13
70b</td><td>B13
8b</td><td>B14
2b</td><td colspan="16">B14
80b</td></tr>
<tr><td>23</td><td colspan="32">B14
160b</td></tr>
<tr><td>24</td><td colspan="2">B14
10b</td><td>B14
5b</td><td>B15
5b</td><td colspan="28">B15
140b</td></tr>
<tr><td>25</td><td colspan="16">B15
80b</td><td>B15
8b</td><td>0</td><td>B15
8b</td><td>0</td><td>B15
8b</td><td>0</td><td>B15
8b</td><td>0</td><td colspan="8">Checksum</td></tr>
<tr><td></td><td>26</td><td colspan="32">Checksum</td></tr>
<tr><td></td><td>27</td><td colspan="24">Checksum</td><td colspan="8">No data</td></tr>
</table>

Same as NRZ. (rows 0–23)

dummy bits (2 bits each)

Format from gustlin_01_0312

Alignment markers

66-bit alignment marker m, 64-bit payload denoted as AM

Bit position								
01 2								65
10	M0	M1	M2	BIP3 CD3	M4	M5	M6	BIP7 CD7

Fixed pattern unique per m (M0, M1, M2)

1's complement of first four bytes (M4, M5, M6, BIP7 CD7)

Strip sync. header and map alignment marker payloads to appear on FEC lanes as shown

	RS symbol index																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

FECL<0>	A0	A4	A8	A12	A16
FECL<1>	A1	A5	A9	A13	A17
FECL<2>	A2	A6	A10	A14	A18
FECL<3>	A3	A7	A11	A15	A19

Copied from gustlin_01_0312

FEC frame structure with AMs

<table>
<tr><th colspan="2"></th><th colspan="16">RS Symbol index</th></tr>
<tr><th colspan="2"></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td rowspan="14">Row index</td><td>0</td><td>$A0_0$</td><td>$A1_0$</td><td>$A2_0$</td><td>$A3_0$</td><td>$A0_1$</td><td>$A1_1$</td><td>$A2_1$</td><td>$A3_1$</td><td>$A0_2$</td><td>$A1_2$</td><td>$A2_2$</td><td>$A3_2$</td><td>$A0_3$</td><td>$A1_3$</td><td>$A2_3$</td><td>$A3_3$</td></tr>
<tr><td>1</td><td>$A0_4$</td><td>$A1_4$</td><td>$A2_4$</td><td>$A3_4$</td><td>$A0_5$</td><td>$A1_5$</td><td>$A2_5$</td><td>$A3_5$</td><td>$A0_6$
$A4_6$</td><td>$A1_6$
$A5_6$</td><td>$A2_6$
$A6_6$</td><td>$A3_6$
$A7_6$</td><td>$A4_7$</td><td>$A5_7$</td><td>$A6_7$</td><td>$A7_7$</td></tr>
<tr><td>2</td><td>$A4_8$</td><td>$A5_8$</td><td>$A6_8$</td><td>$A7_8$</td><td>$A4_9$</td><td>$A5_9$</td><td>$A6_9$</td><td>$A7_9$</td><td>$A4_{10}$</td><td>$A5_{10}$</td><td>$A6_{10}$</td><td>$A7_{10}$</td><td>$A4_{11}$</td><td>$A5_{11}$</td><td>$A6_{11}$</td><td>$A7_{11}$</td></tr>
<tr><td>3</td><td>$A4_{12}$
$A8_{12}$</td><td>$A5_{12}$
$A9_{12}$</td><td>$A6_{12}$
$A10_{12}$</td><td>$A7_{12}$
$A11_{12}$</td><td>$A8_{13}$</td><td>$A9_{13}$</td><td>$A10_{13}$</td><td>$A11_{13}$</td><td>$A8_{14}$</td><td>$A9_{14}$</td><td>$A10_{14}$</td><td>$A11_{14}$</td><td>$A8_{15}$</td><td>$A9_{15}$</td><td>$A10_{15}$</td><td>$A11_{15}$</td></tr>
<tr><td>4</td><td>$A8_{16}$</td><td>$A9_{16}$</td><td>$A10_{16}$</td><td>$A11_{16}$</td><td>$A8_{17}$</td><td>$A9_{17}$</td><td>$A10_{17}$</td><td>$A11_{17}$</td><td>$A8_{18}$</td><td>$A9_{18}$</td><td>$A10_{18}$</td><td>$A11_{18}$</td><td>$A8_{19}$
$A12_{19}$</td><td>$A9_{19}$
$A13_{19}$</td><td>$A10_{19}$
$A14_{19}$</td><td>$A11_{19}$
$A15_{19}$</td></tr>
<tr><td>5</td><td>$A12_{20}$</td><td>$A13_{20}$</td><td>$A14_{20}$</td><td>$A15_{20}$</td><td>$A12_{21}$</td><td>$A13_{21}$</td><td>$A14_{21}$</td><td>$A15_{21}$</td><td>$A12_{22}$</td><td>$A13_{22}$</td><td>$A14_{22}$</td><td>$A15_{22}$</td><td>$A12_{23}$</td><td>$A13_{23}$</td><td>$A14_{23}$</td><td>$A15_{23}$</td></tr>
<tr><td>6</td><td>$A12_{24}$</td><td>$A13_{24}$</td><td>$A14_{24}$</td><td>$A15_{24}$</td><td>$A12_{25}$
$A16_{25}$</td><td>$A13_{25}$
$A17_{25}$</td><td>$A14_{25}$
$A18_{25}$</td><td>$A15_{25}$
$A19_{25}$</td><td>$A16_{26}$</td><td>$A17_{26}$</td><td>$A18_{26}$</td><td>$A19_{26}$</td><td>$A16_{27}$</td><td>$A17_{27}$</td><td>$A18_{27}$</td><td>$A18_{27}$</td></tr>
<tr><td>7</td><td>$A16_{28}$</td><td>$A17_{28}$</td><td>$A18_{28}$</td><td>$A19_{28}$</td><td>$A16_{29}$</td><td>$A17_{29}$</td><td>$A18_{29}$</td><td>$A19_{29}$</td><td>$A16_{30}$</td><td>$A17_{30}$</td><td>$A18_{30}$</td><td>$A19_{30}$</td><td>$A16_{31}$</td><td>$A17_{31}$</td><td>$A18_{31}$</td><td>$A19_{31}$</td></tr>
<tr><td>8</td><td>P 5b | B5 5b</td><td colspan="15">B5
150b</td></tr>
<tr><td>9</td><td colspan="10">B5
100b</td><td>B5 2b | B6 8b</td><td colspan="5">B6
50b</td></tr>
<tr><td>…</td><td colspan="16">…</td></tr>
<tr><td>24</td><td>B14
10b</td><td>B14 5b | B15 5b</td><td colspan="14">B15
140b</td></tr>
<tr><td>25</td><td colspan="8">B15
80b</td><td>B15 8b | 0</td><td>B15 8b | 0</td><td>B15 8b | 0</td><td>B15 8b | 0</td><td colspan="4">Checksum
40b</td></tr>
<tr><td>26</td><td colspan="16">Checksum
160b</td></tr>
<tr><td></td><td>27</td><td colspan="12">Checksum
120b</td><td colspan="4"></td></tr>
</table>

Same as NRZ. (rows 0–24)

Format from gustlin_01_0312

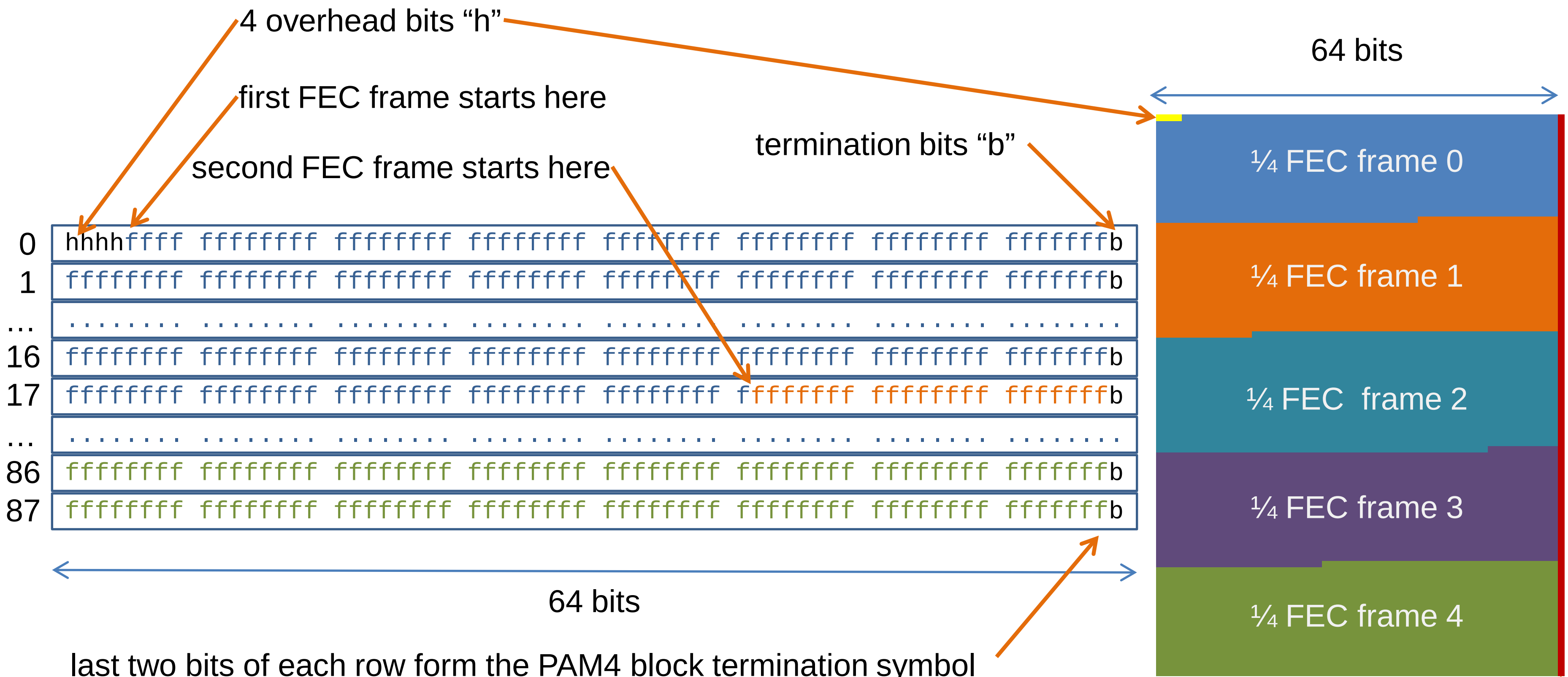
Mapping FEC to PMA lanes

- Cycle through FEC 10-bit words through each of the 4 PMA lanes.
  - The FEC frame contains 444 10-bit words
  - For each FEC frame, 111 10-bit words are destined for each of the four PMA lanes.
  - FEC words (i+j*4) go to lane i
    - i is {0,1,2,3}, where i represents the lane #
    - j is {0,1,2,…,110}, j indexes the FEC words destined for each lane
  - Note that for FEC words 408 to 411, only the 8 data bits are transferred to each lane.

PMA Frame

- PMA frame generated for each PMA lane.
- PMA frame is composed of…
  - 5 quarter FEC frames, 5*(4440-8)/4 = 5540 bits
  - 4 overhead bits
    - essential to give a resultant PAM4 symbol rate of 88 * 156.25 MHz
    - various possible applications discussed on subsequent slide
  - 88 PAM4 block termination bits
    - 1 termination bit per 63 data bits
  - 5632 bits total

PMA frame structure (one per lane)



Each pair of bits, map to one PAM4 symbol.
For the PAM4 block termination symbol, we want "b" and the preceding bit "f" to indicate +1 or -1 so …
For gray mapping, b = 0, always!
if the preceding bit is 1, then 10 maps to +1
if the preceding bit is 0, then 00 maps to -1

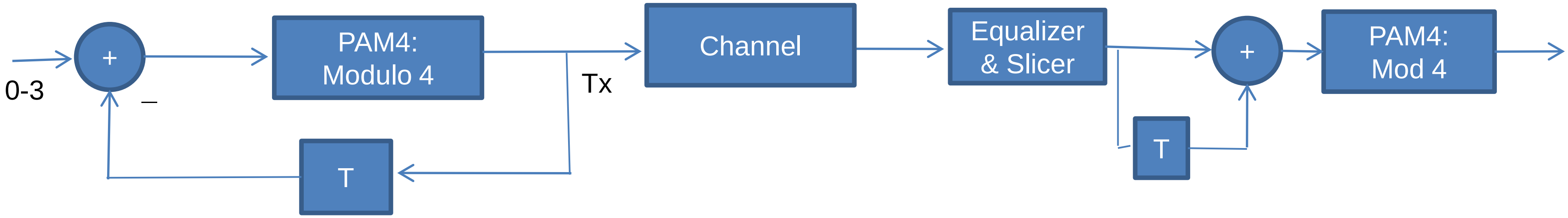
Legend:
"f" = bits from 5 FEC frames
"h" = overhead bits
"b" = block termination bits

PMA Frame Overhead Bits

- Each PMA per-lane frame has 4 overhead bits.
- Must be randomized or at least “friendly”.
- Various applications …
  - PMA frame alignment (see previous slide)
  - lane identification
  - control channel for remote transmitter control
  - vendor specific use

Pre-Coding

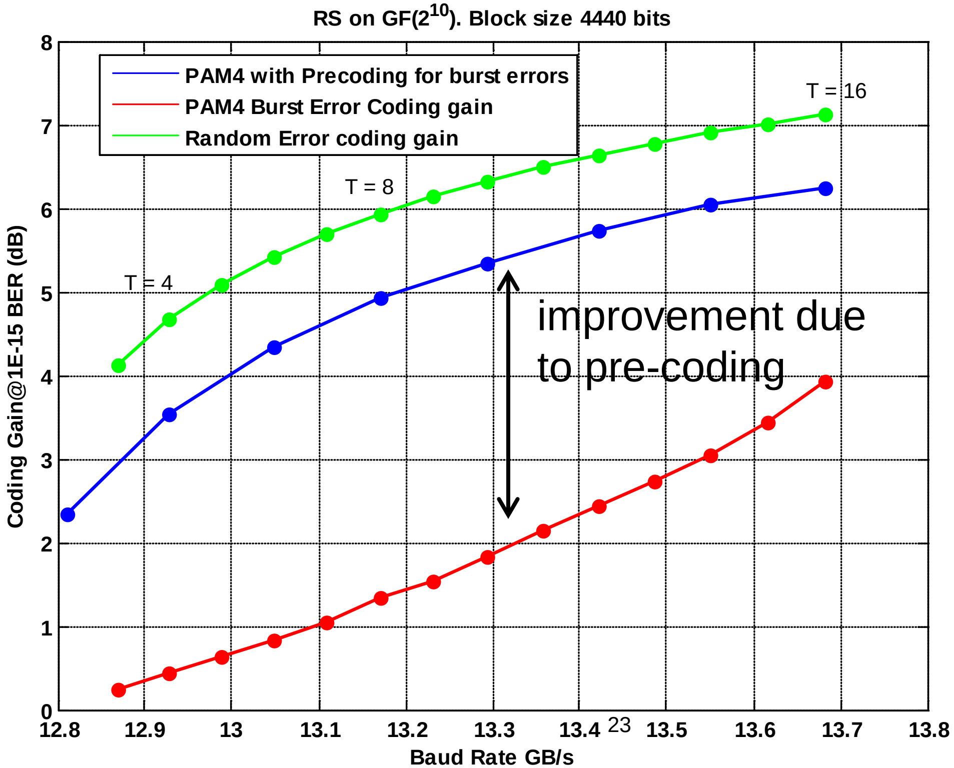
- 1/(1+D) modulus 4 pre-coding
  - See bliss_01_0311, “Signaling Terminology; PAM-M and Partial Response Precoders”
  - Rx uses a (1+D) mod 4 after slicing
- Simple to implement
- Very low Complexity; similar complexity to duo-binary precoder.
- Pre-coding is mandatory; negotiation is not required.



Motivation for pre-coding

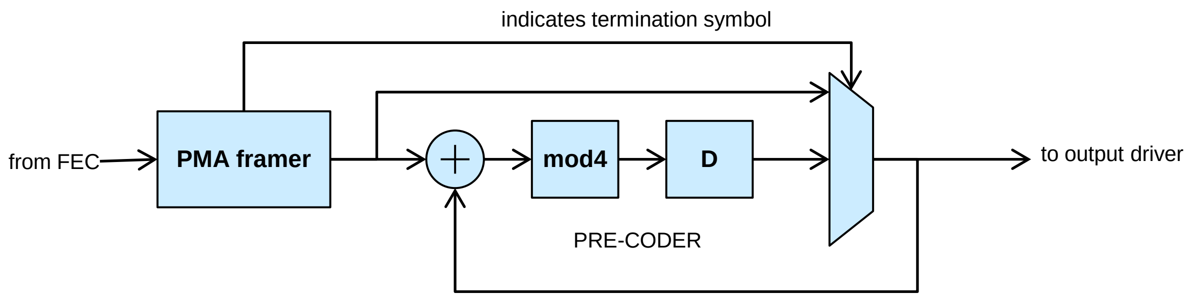
- Pre-coding mitigates error propagation in DFE and MLSD receivers.
  - Greatly reduces number of errors per burst.
  - For 1-tap DFE, reduces burst to two errors, one at beginning and one at end
  - For MLSD see dabiri_01_0911 “Enabling Improved DSP Based Receivers for 100G Backplane”

- Graph shows improved coding gain (blue) due to precoding.
- The delta between burst error and random error is ~1.0dB with 1/(1+D) mod 4 precoding



PAM4 Block Termination

- PAM4 block termination symbol every 32 PAM4 symbols
  - For efficiency, each PAM4 termination symbol transmits one data bit.
    - 63 data bits sent every 32 PAM4 symbols
    - Increases baud rate by 64/63.
  - Each PAM4 block termination symbol is mapped to either +1 or -1.
  - At the transmitter, termination added within the precoder.
  - At the detector, termination removed after the detector.
- See dabiri_01_0112.
- PAM4 block termination encoding is mandatory; negotiation is not required.



Functional representation of block termination and pre-coding

Motivation for PAM4 Block Termination

- Block termination by transmitting known PAM4 symbols on a regular cycle enables…
  - efficient and effective MLSD, maximum likelihood sequence detection (dabiri_01_0911)
  - parallel DFE implementations
    - Keshab K. Parhi, Pipelining of parallel multiplexor loop and Decision Feedback Equalizers, ICASSP, 2004

PAM4 encoding

- Gray mapping
  - pre-coder output {10, 11, 01, 00} maps to {+1,+1/3,-1/3,-1}
  - based on 2B1Q coding used in HDSL and ISDN

PMA synchonization

- Lock to PAM4 termination blocks by searching for PAM4 termination symbols
  - PAM4 termination symbols (1 in 32) are always either +1 or -1.
  - Similar to framing on 10 or 01 sequence for 64B/66B, can borrow and modify 64B/66B synchronization state machine.
- Lock to PMA frame
  - Use known content of overhead bits.
    - Once locked to the PAM4 termination blocks, look for 4 bits (2 PAM4 symbols) every 88 rows.
    - Again, similar to 64B/66B synchronization.

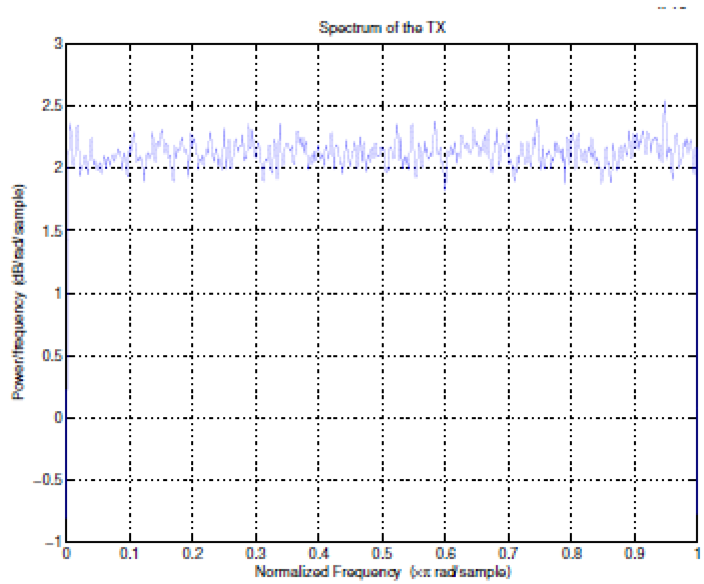
Energy Efficient Ethernet Operation

- Fast synchronization for REFRESH and WAKE.
  - Synchronize on PAM4 termination symbols.
  - Use prescribed sequence to accelerate synchronization.
- For REFRESH, PCS and FEC not required.
  - Replace with scrambled sequence.
    - Similar to EEE/LPI for 10GBASE-T.
- For WAKE, rapid alignment markers not required by the PMA/PMD receiver.
  - Will still be required at the PCS RX at the PCS end point.
- No significant impact to work being done in EEE consensus group.
  - Compatible and complementary with PCS state machine in Gustlin_02_1111.

Thanks!

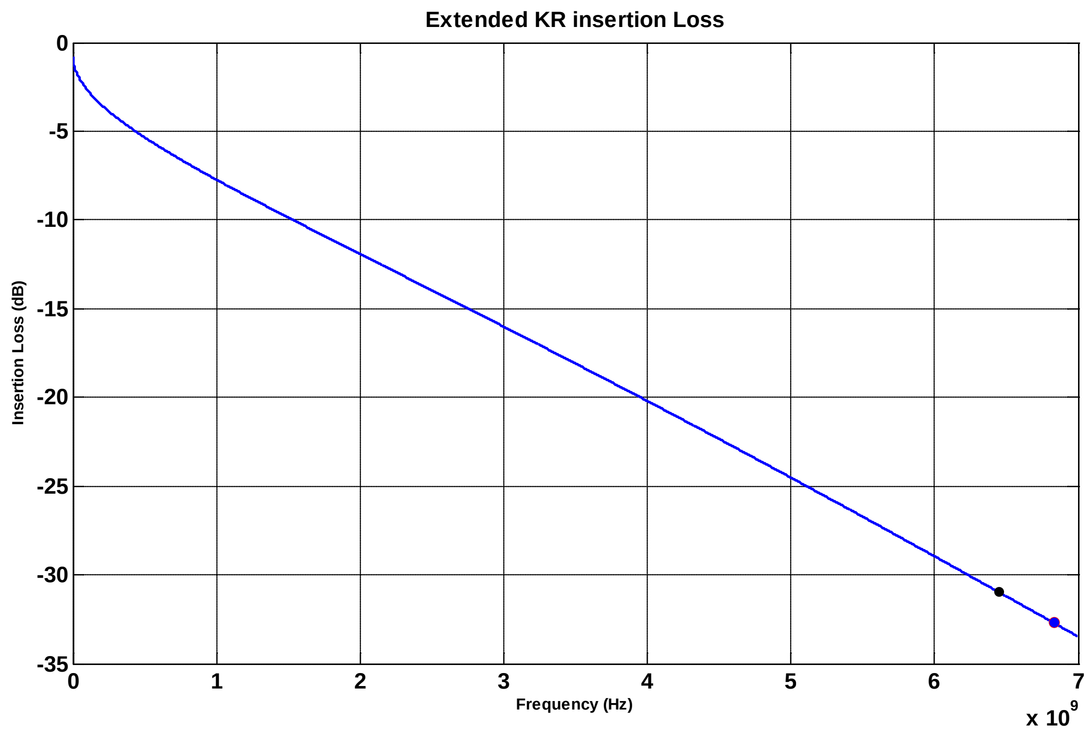
BACKUP SLIDES

Power spectrum with PAM4 block termination symbols



- The simulated spectrum above shows no spectral content due to block termination symbols.
- Pattern is repeating structure (not content) of 32 PAM4 symbols…
  - 31 random PAM4 symbols in {-1,-1/3,+1/3,-1} * 3
  - 1 random PAM4 symbol in {-1, +1} * 3

PAM4 SNR Loss due to Over clocking

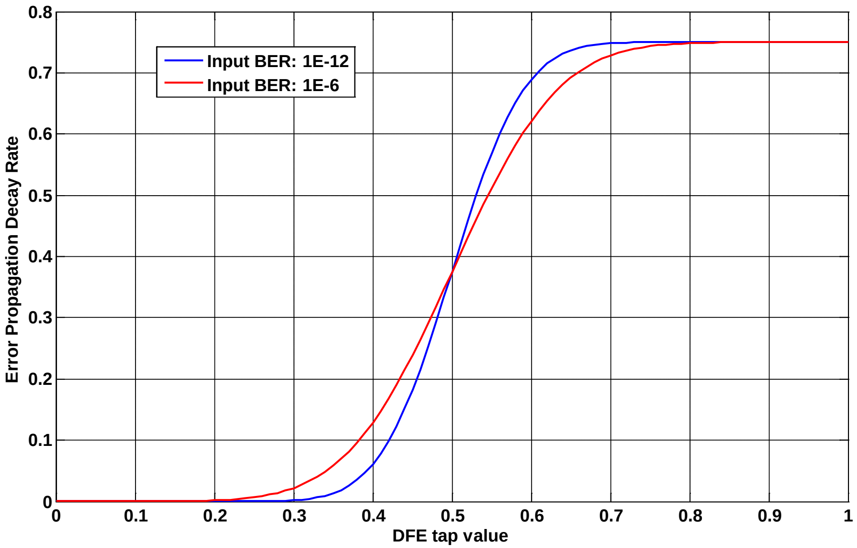


For FEC baud rate of 13.67G, the SNR loss due to over clocking

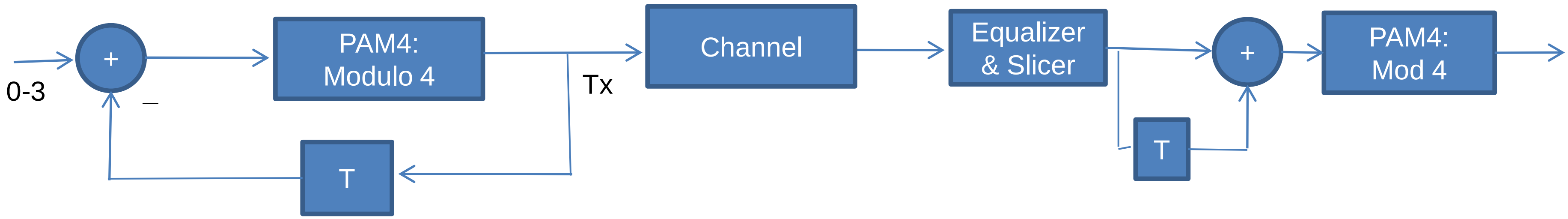
- $SNR_{delta} = (IL_{6.84GHz} - IL_{6.45\ GHz})/2 = 0.9dB$

Precoding Motivation: PAM4 DFE bursts

- DFE's are well known to multiply errors in the feedback loop
  - A single error will become a burst error
- Consider PAM4 1-tap DFE with tap coeff = 1
  - If previous decision is wrong, then there is 3/4 probability of making a successive error
  - i.e. Probability of K consecutive errors = $(3/4)^k$
- Lower 1st DFE tap between 0.6 to 1 have similar burst length as tap coefficient of 1
  - Tap of 1: 0.75^k
  - Tap of 0.7: 0.72^k
  - Tap of 0.6: 0.62^k
- A single random error may consume multiple Reed Solomon words
  - Burst error coding gain is lower than coding gain for random errors

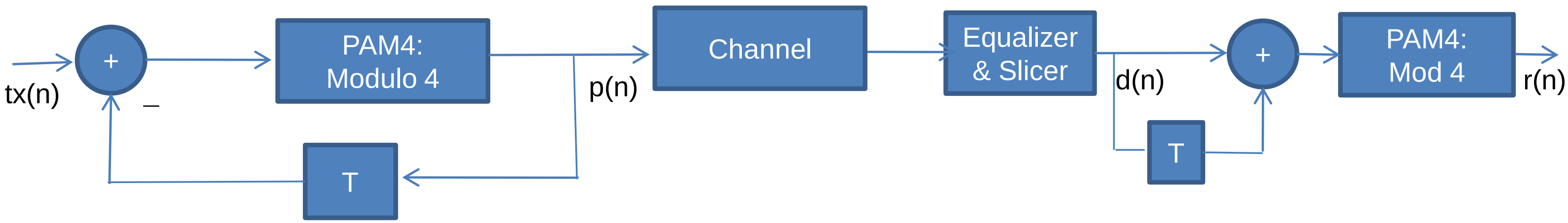


1/(1+D) Precoding for DFE burst errors



- The burst error length of the DFE error events for PAM4 can be reduced by using precoding
- PAM4 Tx precoding uses a 1/(1+D) mod 4
  - See bliss_01_0311, "**Signaling Terminology; PAM-M and Partial Response Precoders"**
  - Rx uses a (1+D) mod 4 after slicing
- Simple to implement
- Very low Complexity; similar complexity to duo-binary precoder
- Reduces 1 tap DFE burst error runs into 2 errors per error event
  - One error at the entry, one error at the exit

1/(1+D) Precoding worked example



- Precoder Input : tx(n)
  - 2 2 2 2 0 3 2 0 1 3 3 0 0 0 0 2 3 0 3
- Precoder Output : p(n)
  - 0 2 0 2 2 1 1 3 2 1 2 2 2 2 2 0 3 1 2
- DFE, Slicer Output : d(n)
  - 0 1 1 1 3 0 2 2 3 0 3 1 3 1 3 0 3 1 2
- Error Event : p(n) – d(n)
  - 0 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 0 0 0 0
- Decoder Output after 1+D at Rx : r(n)
  - 2 1 2 2 0 3 2 0 1 3 3 0 0 0 0 3 3 0 3

Entry Error

Exit Error

This example does not include the PAM4 block termination.

Mean Time To False Packet Acceptance (MTTFPA)

- Assume any FEC frame known to be in error is marked.
  - Any 64B/66B blocks within the marked FEC frame are replaced with error blocks.
  - The errored packets are then eventually discarded by the downstream MAC.
  - Only FEC frames without error detected (falsely decoded) may result in falsely accepted packets.
- Probability of a FEC false decode, P_{FFD} (i.e. outputting a false codeword)
  - P_{FFD} = 1/t!, where t is the strength of the code
  - The output codeword will generally contain 2t+1 errors
  - Ethernet CRC32 cannot guarantee detection for 2t+1 errors
  - A false CRC32 match is random with probability 2^{-32}
- Probability of false packet acceptance, P_{FPA}
  - P_{FPA} ~= P_{FFD} * BER_{OBJ} * 2^{-32} * N = 1.1E-35 * N
  - N = average number of packets affected by each FEC frame, somewhere between 0 and 7
  - BER_{OBJ} = FEC BER objective = 1E-12
- For mandatory PAM4 FEC, RS(444, 412, t = 16)
  - MTTFPA ~= 1/P_{FPA} * 1/ (13.75E9 * 2 * 4) * 1/ (60*60*24*365) years = 2.6E16 / N years
  - For N = 7, MTTFPA ~= 3.9e15 years
    - Lifetime of universe is ~13E9 years.