



Training & EEE Baseline Proposal

IEEE 802.3bp - Plenary Meeting - November 2014

William Lo, Zhenyu Liu, Marvell

Baseline Proposal

- ▶ **Adopt training and EEE framework in this presentation as baseline**
- ▶ **Based on:**
 - Lo_3bp_03_0714.pdf
 - Graba_3bp_01a_0714.pdf
 - Lo_3bp_01a_0914.pdf
 - with additional modifications
- ▶ **Consolidate everything into this presentation for easy access**

Parameters (finalized)

► Optimized to work with RS(450, 406, 2⁹), PAM3, 750MBaud/s

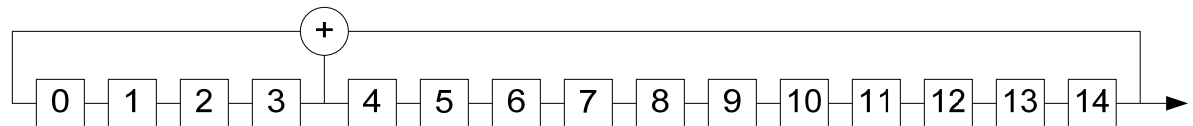
Symbol	Definition	Values
RS3	# PAM3 symbols per RS frame	2700
RST	Duration of RS frame (ns)	3600
PRS3	# PAM3 symbols per partial RS frame	180
PRST	Duration of partial RS frame (ns)	240
PF	# partial frames per RS frame	15
QRF	# RS frame per quiet refresh cycle	24
QRT	Duration of quiet refresh cycle (ns)	86400
2 x AF	# partial RS frames separating alert	30
2 x AlertGranularityT	Alert Granularity (ns)	7200
Refresh_LPI	# partial RS frames for refresh	6
Refrst_T	Duration for refresh (ns)	1440
QR Ratio	Quite/Refresh Ratio	60
Enter_LPI_RS	# RS frames with all LPI to enter LPI	1
Alert_LPI	# alert sequences to exit LPI	Not needed
Alert_T	Sense window (ns)	Implementation Dependent
Exit_LPI_RS	# RS frames with all idles upon exit LPI	1
Alert_sym	# symbols in alert sequence	Not Needed

Modifications to previous PCS Baseline (new)

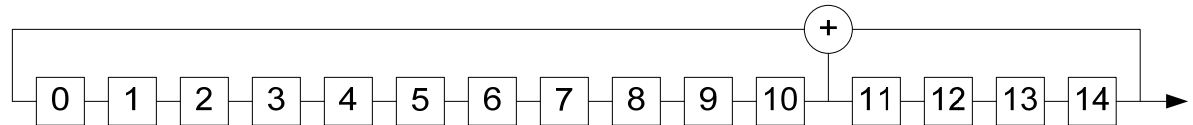
- ▶ **Make modification to datapath suggested by tu_3bp_01_1114.pdf**
 - Scrambler after RS-FEC
 - Fixed sequence scrambler instead of self sync scrambler

- ▶ **Change scrambler per chini_3bp_01a_0914.pdf**

- **Master $X^{15} + X^4 + 1$**



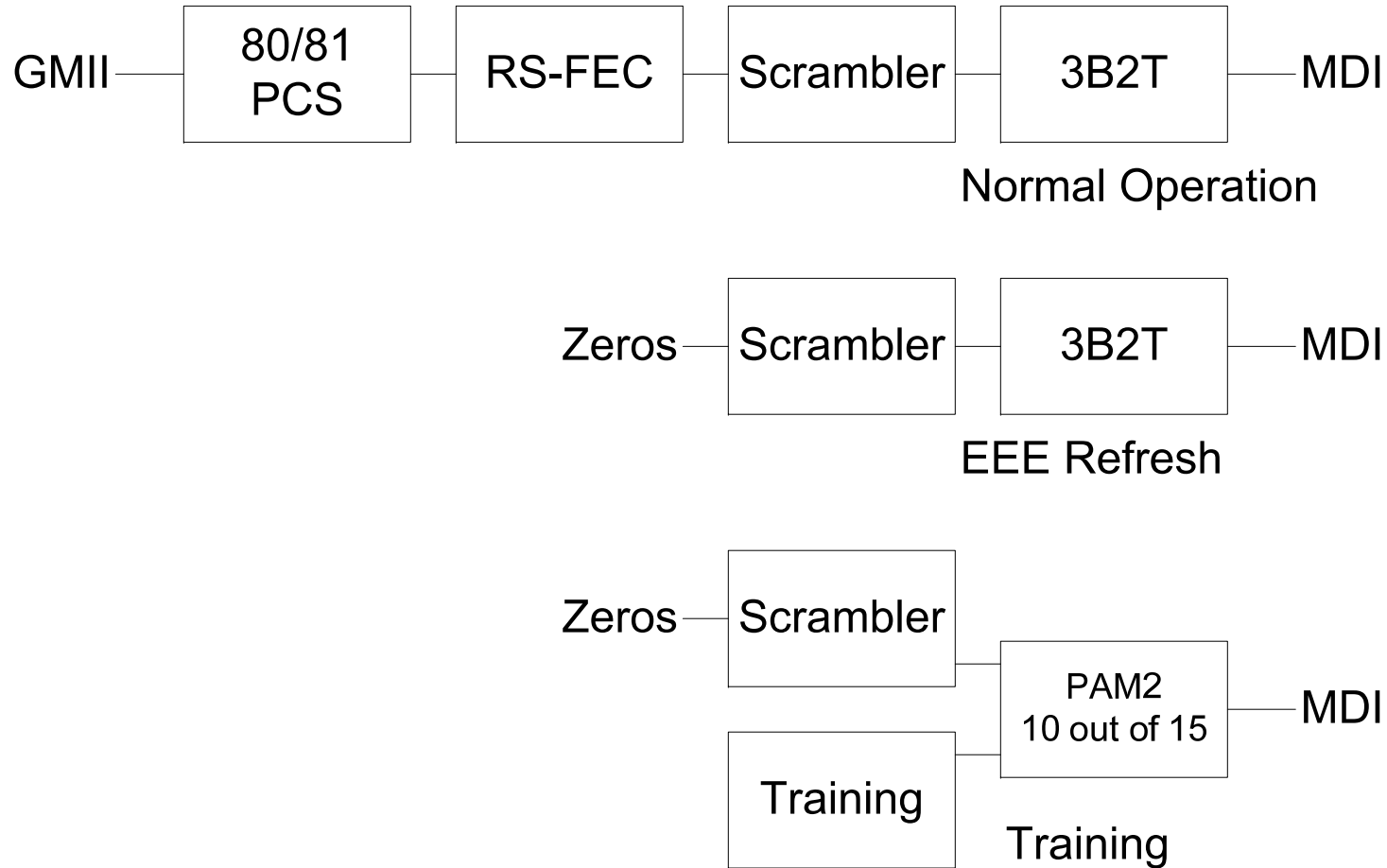
- **Slave $X^{15} + X^{11} + 1$**



- ▶ **Same scrambler used for training, normal operation, and EEE refresh**
 - Scrambler sequence is not interrupted once training starts

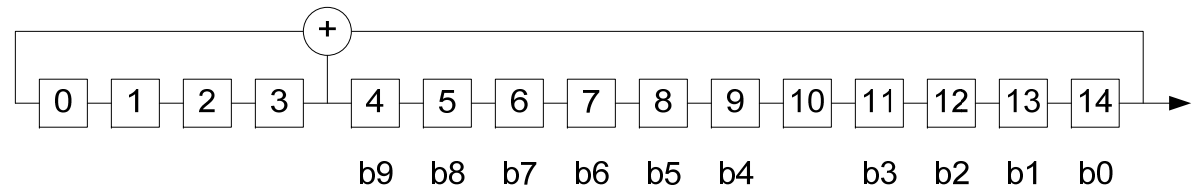
Scrambler (new)

► Scrambler sequence fixed starting from training

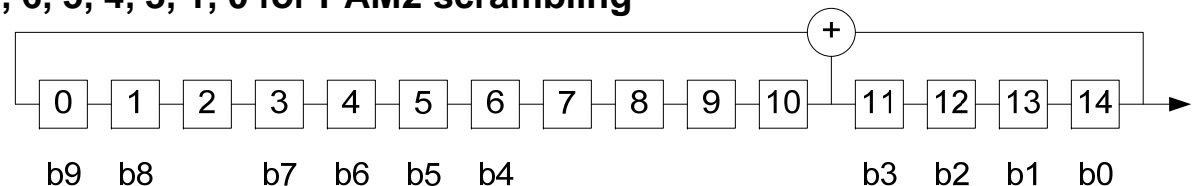


10 out of 15 bit scrambler details during training (new)

- ▶ Scrambler outputs 3 bit sequence for every 2 symbols transmitted for PAM-3 operation
- ▶ During training only 2 bits scrambler needed for 2 symbols for PAM-2 operation
- ▶ For every 15 bits scrambler output use only 10 of the bits for training
- ▶ Master $X^{15} + X^4 + 1$
 - Use bits 14, 13, 12, 11, 9, 8, 7, 6, 5, 4 for PAM2 scrambling



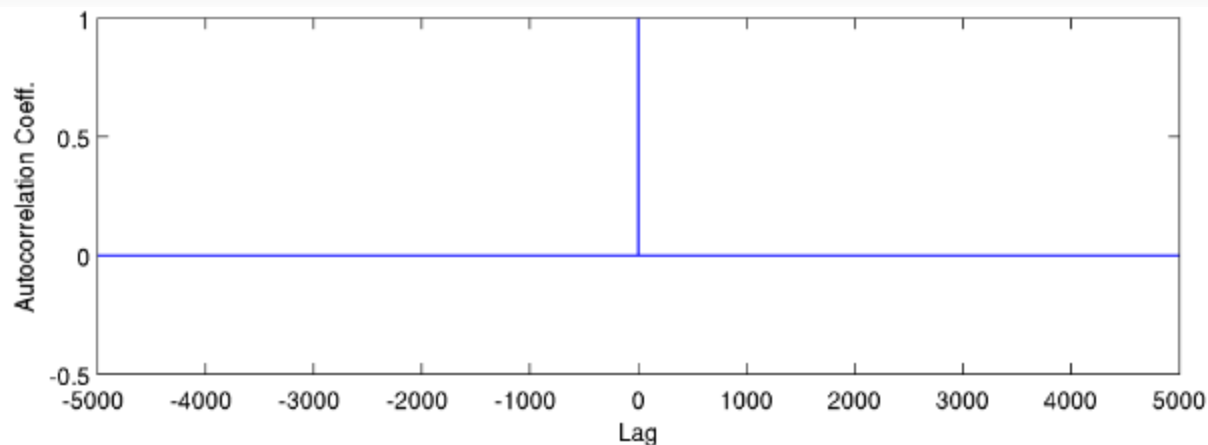
- ▶ Slave $X^{15} + X^{11} + 1$
 - Use bits 14, 13, 12, 11, 6, 5, 4, 3, 1, 0 for PAM2 scrambling



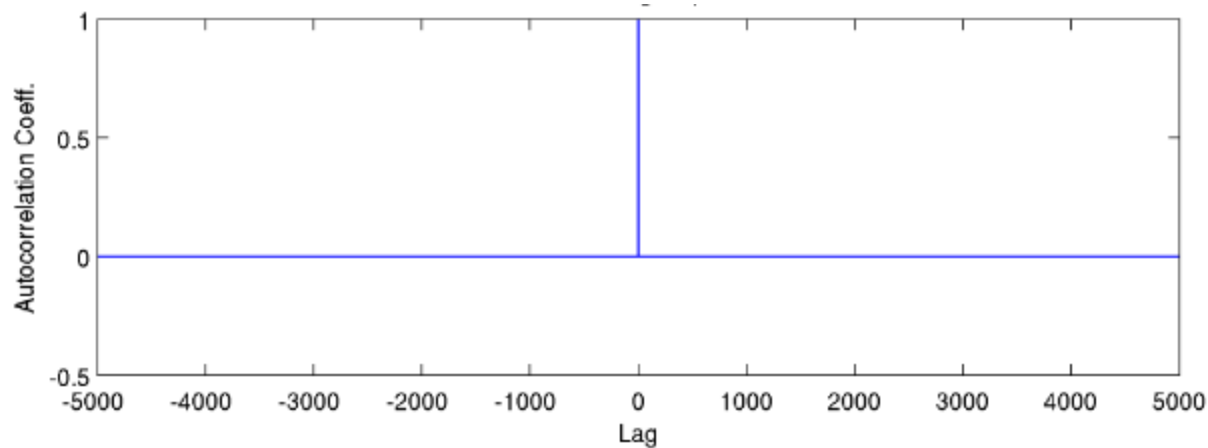
- ▶ Scrambler advances at same rate in PAM2 and PAM3 operation
 - No need to worry about scrambler when PAM2 to PAM3 switchover occurs
 - Scrambler state can be recovered at the receiver during training

Good Autocorrelation of 10/15 Training Sequence (new)

▶ Master

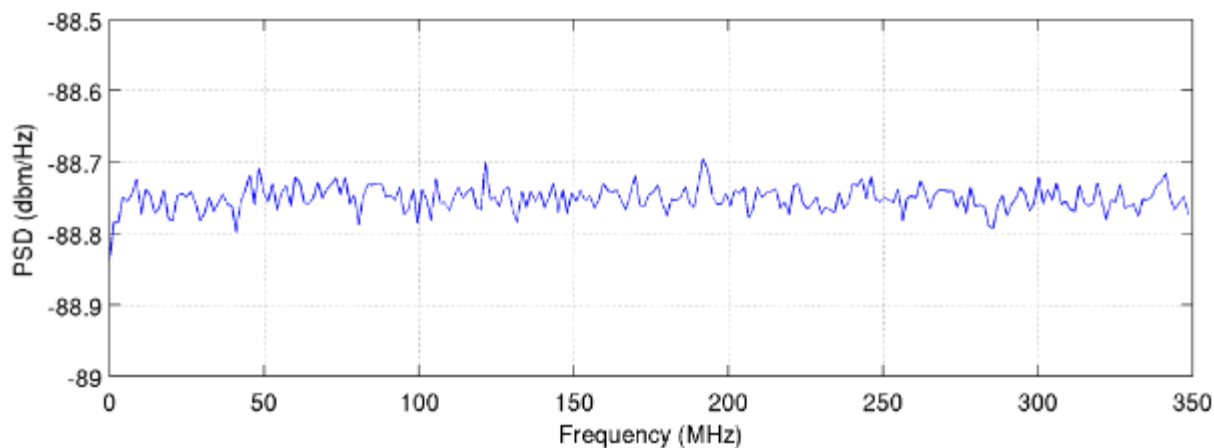


▶ Slave

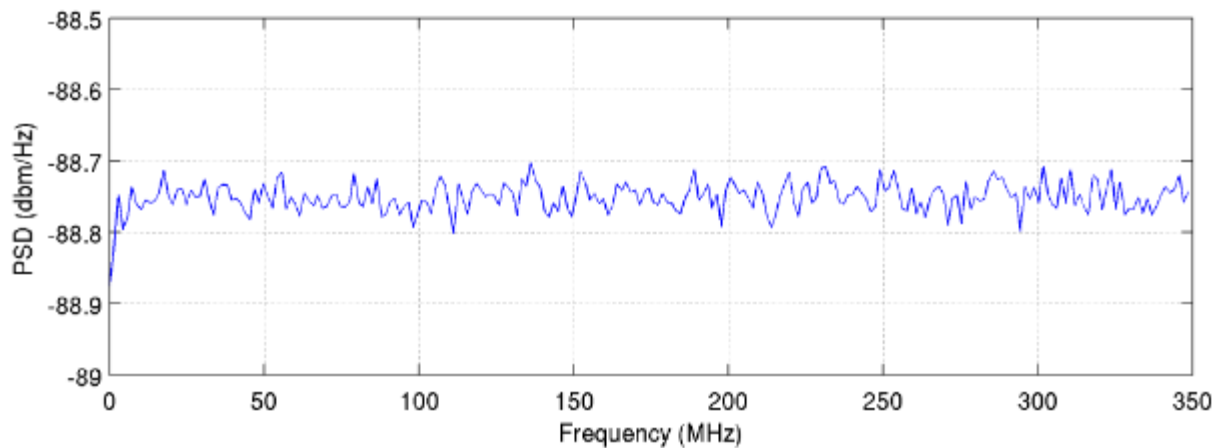


Flat PSD of 10/15 Training Sequence (new)

▶ Master



▶ Slave



Syncing the descrambler (new)

- ▶ **Actual implementation left up to the implementer – example shown here**
- ▶ **10 possible positions to align 10 bits. Assuming bits b0 to b17 are aligned correctly, the state shown in the table is the initial scrambler state when bit b0 is sent advanced by 9 states**
- ▶ **See if subsequent received pattern matches descrambler. If not then try a different alignment position**
- ▶ **Assumes training sequence is outputting all zeros**

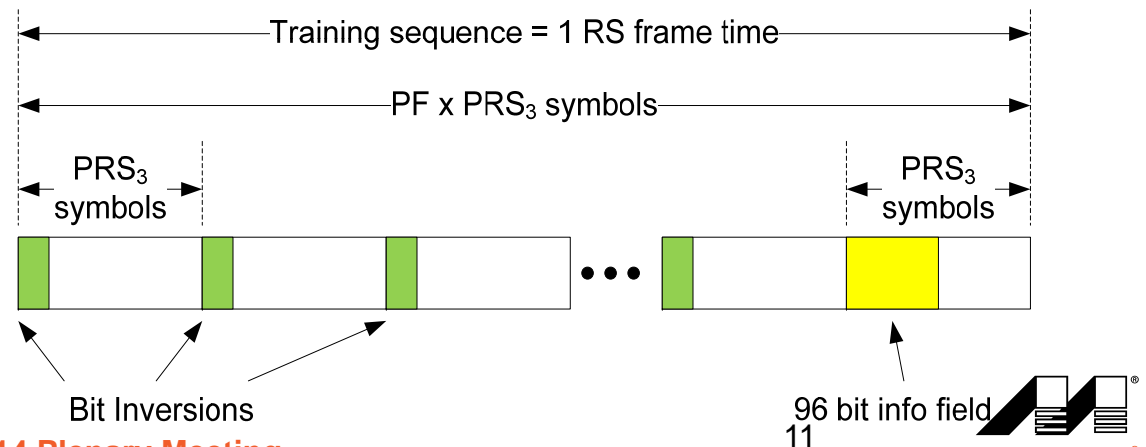
De-scrambler state bit	$X^{15} + X^4 + 1$	$X^{15} + X^{11} + 1$
14	b8	b5
13	b9	b6
12	b10 xor b0	b7
11	b11 xor b1	b14 xor b4
10	b12 xor b2	b8
9	b13 xor b3	b9
8	b10	b10
7	b11	b11
6	b12	b12
5	b13	b13
4	b17 xor b7	b10 xor b4 xor b0
3	b14	b11 xor b5 xor b1
2	b15	b12 xor b6 xor b2
1	b16	b13 xor b7 xor b3
0	b17	b14

Benefits of common fixed sequence scrambler (new)

- ▶ **No error propagation on bit errors**
- ▶ **No scrambler warm up time needed waking from EEE**
 - **Allows single RS frame to wake**
- ▶ **Descrambler locks during training without need to exchange seeds in info field for a different scrambler**
- ▶ **Switch over from PAM2 to PAM3 can be easily done during training**

1000BASE-T1 Training (same except new scrambler sequence)

- ▶ XOR training sequence with 10 out of 15 scrambler sequence
- ▶ Issue – 1 RS frame a lot longer than 1 LDPC frame
 - $RS(450, 406, 2^9) = 3600$ ns
- ▶ Want bit inversion and info field to occur more frequently given noisier environment
- ▶ Solution – Introduce partial RS frame
 - Divide RS frame time into PF number of PRS_3 symbol groups
 - Info field occurs once per RS frame time. Indicated by XORed 0xBBA7 pattern
 - Info field first 96 bits of PRS_3 symbol group to avoid offset calculations.



Info Field (same except 96 bits now)

- ▶ **Simplify to 96 bits.**
 - No need for PBO and THP
- ▶ **No transition counter needed**
 - No PBO or THP so no need to count down to readapt DSP to new TX settings
 - Significantly speeds up training
- ▶ **Partial RS Frame Count (PFC) used to establish time synchronization for EEE**
 - Simpler mechanism than using transition counter to zero LDPC frame count on entering PCS_Test training state in 10GBASE-T
 - Free running on 1000BASE-T1 master
 - Slave must match partial frame count (PFC) to within +0/-1 partial RS frame measured at the receiver input
- ▶ **Message TBD after state machine defined**

1000BASE-T1

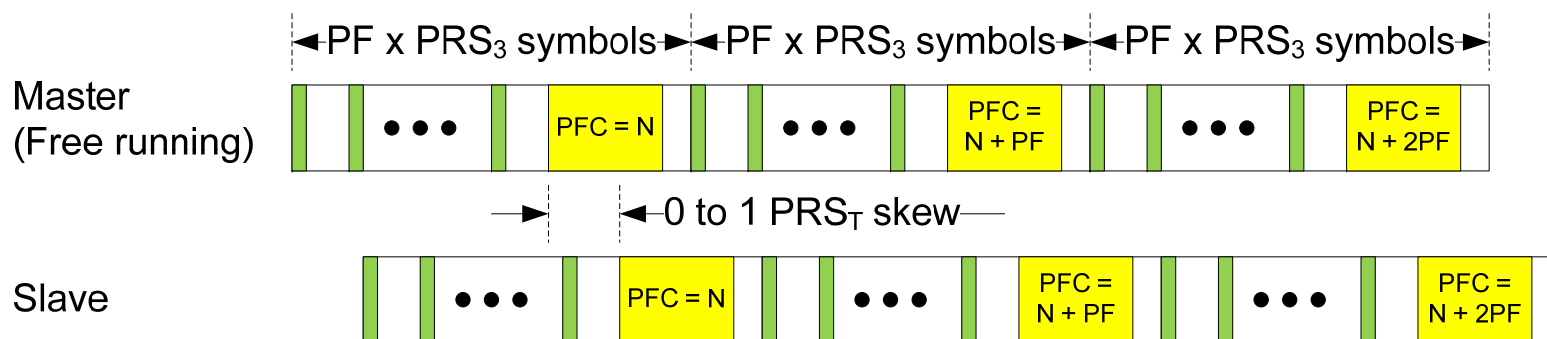
0xBB
0xA7
0x00
Message
Reserved
Reserved
Reserved
Reserved
Partial frame count mod (QF x PF)
CRC16
CRC16

10GBASE-T

0xBB
0xA7
0x00
0x00
TX Setting
TX Setting
TX Setting
Message
SNR
(format dependent) Transition counter THP Coefficient Vendor specific
CRC16
CRC16

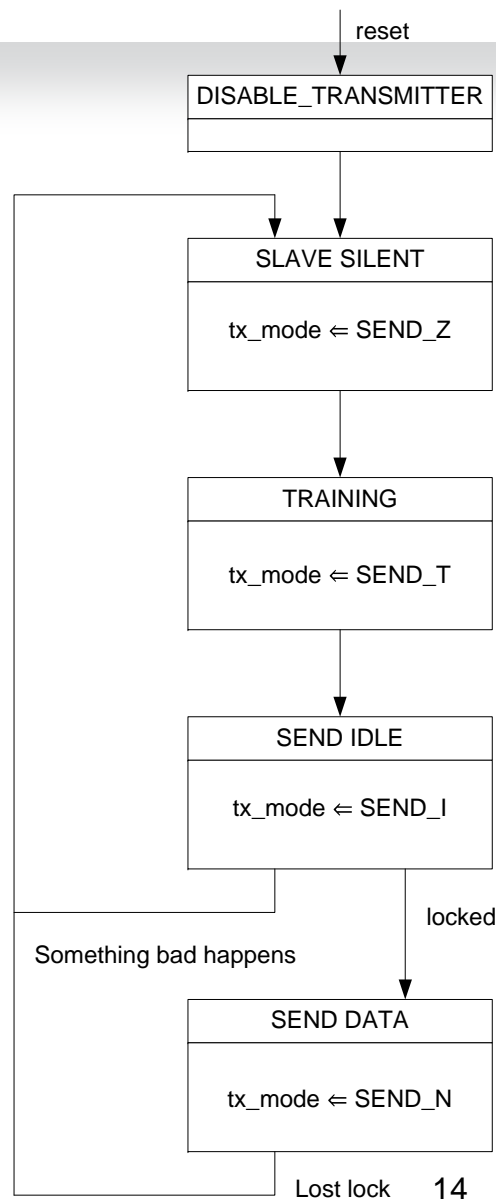
Example of slave partial RS frame count matching (same)

- ▶ PF x PRS₃ symbols per training sequence
- ▶ Master free runs and increments PFC by PF every training sequence
 - mod (QRF x PF) implied in diagram
- ▶ Slave locks to within +0/-1
 - Slave accepts master PFC only if CRC16 is good.
 - Robust to noise since not every info field needs to be processed to recover master PFC



PHY Control State Machine (same)

- ▶ **Greatly simplified since no PBO or THP coeff exchanged**
- ▶ **Sketch of state machine**
 - Master transmits PAM 2 and slave silent
 - Both transmit PAM2 in Training
 - Message exchanged in info field indicating ready to move to PAM 3
 - Send PAM3 idles for some time
 - Link up and send data
- ▶ **Details of state transition TBD**

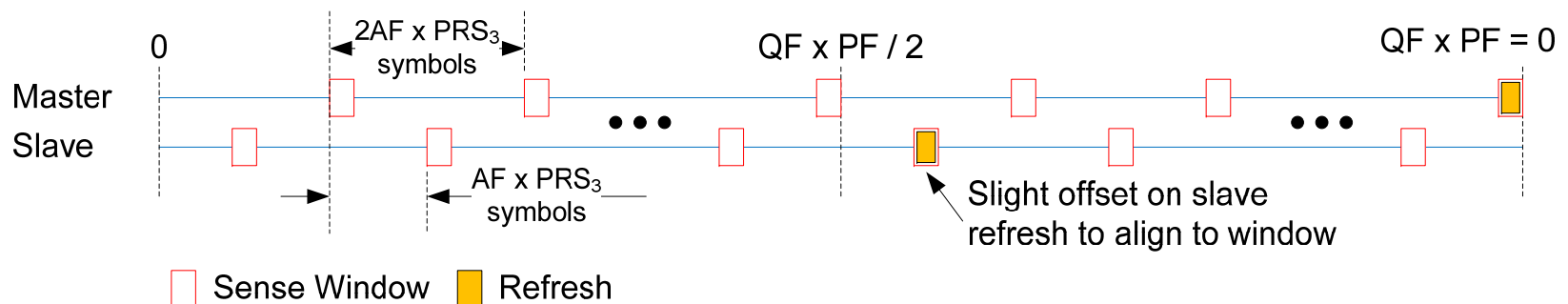


EEE - Entering LPI (same)

- ▶ If LPI seen on GMI fill remaining bytes in RS frame with LPI symbol. Then send Enter_LPI_RS number of RS frame with nothing but LPI symbols.

EEE – Quiet/Refresh and Sense (same, changed the word alert to sense to avoid confusion)

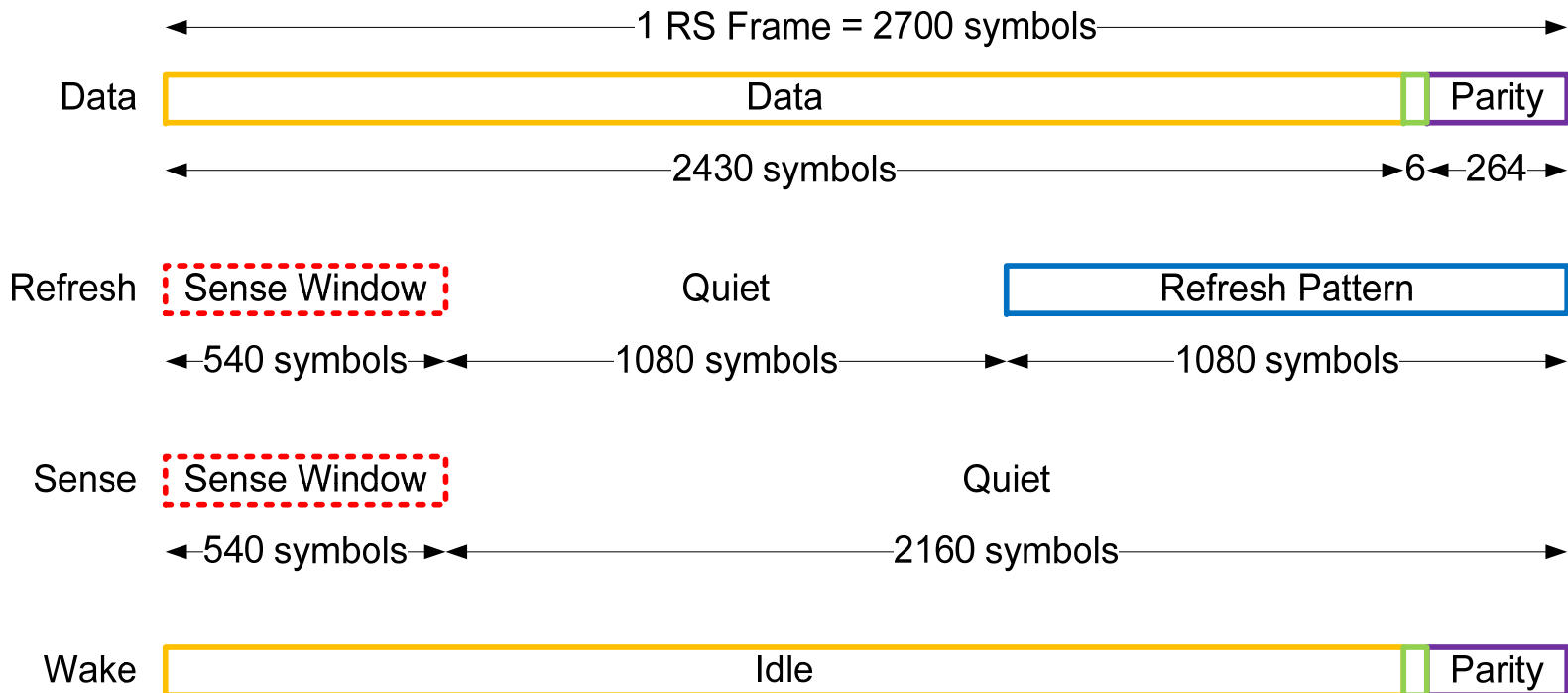
- ▶ **Master and Slave Refresh Staggered as shown**
- ▶ **Use 1000BASE-T wake time of 16.5us instead of 4.48us of 10GBASE-T to allow more power savings**
- ▶ **Allow wake frame to be sent only during certain windows**
 - Allows receiver to power down outside window
 - Stagger windows between master and slave so wake frame never overlap
 - Will increase worst case wake time waiting for window
 - Align refresh with sense window
 - Space sense windows $2 \times AF \times PRS_3$ symbols apart and stagger master and slave windows by $AF \times PRS_3$



Quiet / Refresh / Sense – With actual numbers (same, changed the word alert to sense to avoid confusion)

▶ Every 2700 symbol can have 1 of 5 activities

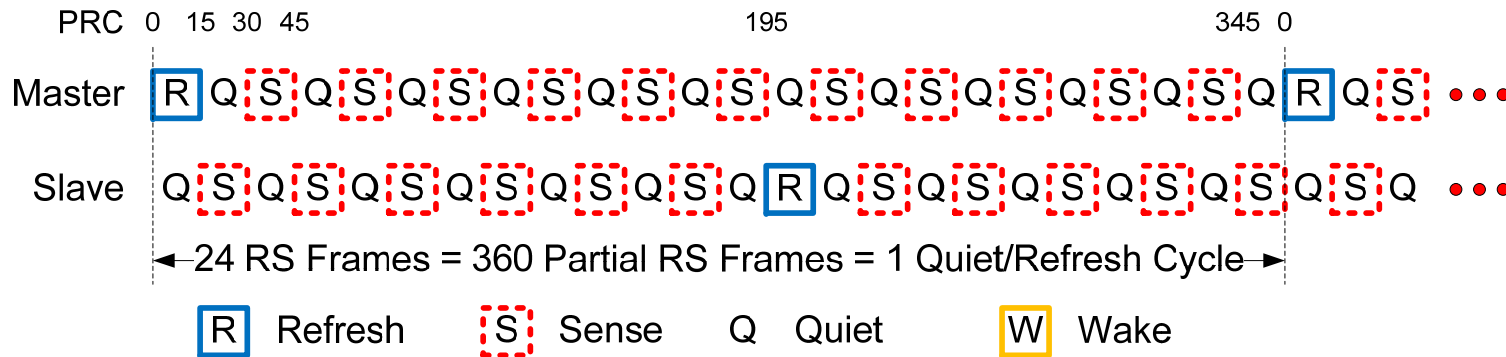
- Data – Regular RS frame
- Refresh – a sense window followed by refresh pattern
- Sense – a sense window only
- Quiet – pure quiet period.
- Wake – wake pattern – simply a data frame with all idles



*sense window shown as 540 symbols as an example. Actual duration is left to the implementer

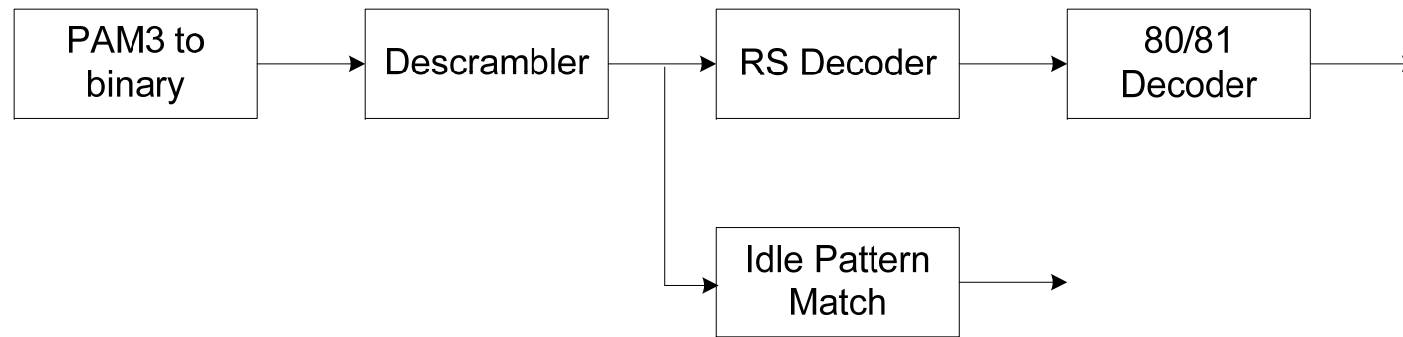
Quiet Refresh Cycle – With actual numbers (same, changed the word alert to sense to avoid confusion)

- ▶ Every quiet/refresh cycle consists of 24 RS Frame times
- ▶ Same as 360 80/81 encoder transfers
- ▶ Same as 360 Partial RS Frame times
- ▶ Quiet and Sense are offset between master and slave
- ▶ Wake can only be sent during PHYs Quiet time
 - Coincides with link partners sense window



EEE – Exit LPI Procedure (same)

- ▶ **Send 1 RS frame with all bytes idles**
 - This is the wake frame. Alert pattern not needed.
- ▶ **Lets the main data path warm up**
 - Worst case wakeup time $2 \times RS_T + \text{latency} = 2 \times 3.6\mu\text{s} + \text{approx } 5\mu\text{s} = 12.2\mu\text{s}$
- ▶ **Optional parallel path for early detection of sufficient number of idles bytes in pattern match to exit LPI**
 - Data is not corrected by RS



Refresh Pattern (new)

- ▶ **Use scrambler sequence with all zero data as shown in slide 5 during refresh window.**

THANK YOU