Marvell. Moving Forward Faster

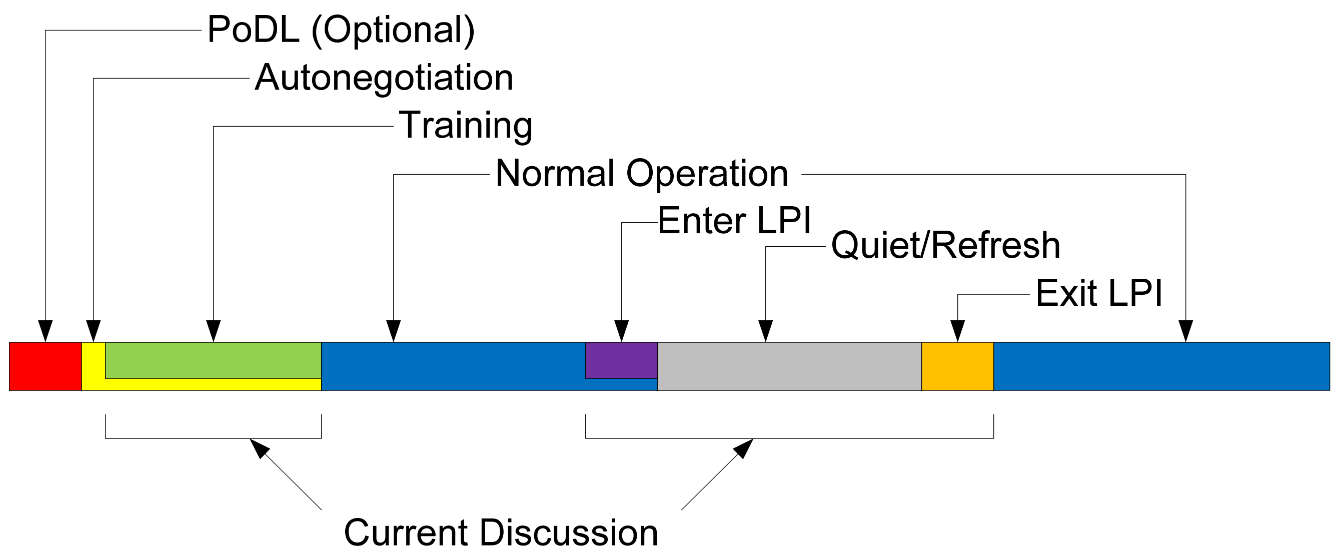
Training & EEE Proposal

IEEE 802.3bp - Plenary Meeting - July 2014

William Lo, Marvell

Agenda

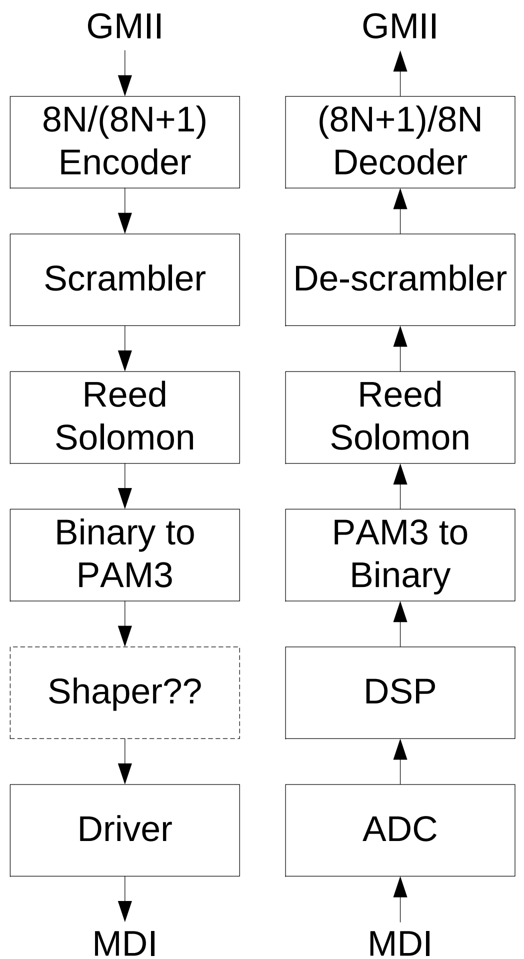
- Leverage Clause 55 10GBASE-T Training and EEE and adapt for 1000BASE-T1
- Present proposal with parameterized values
- Make tentative recommendations on the actual parameters
- Conclusions and next steps



1000BASE-T similarities to 10GBASE-T

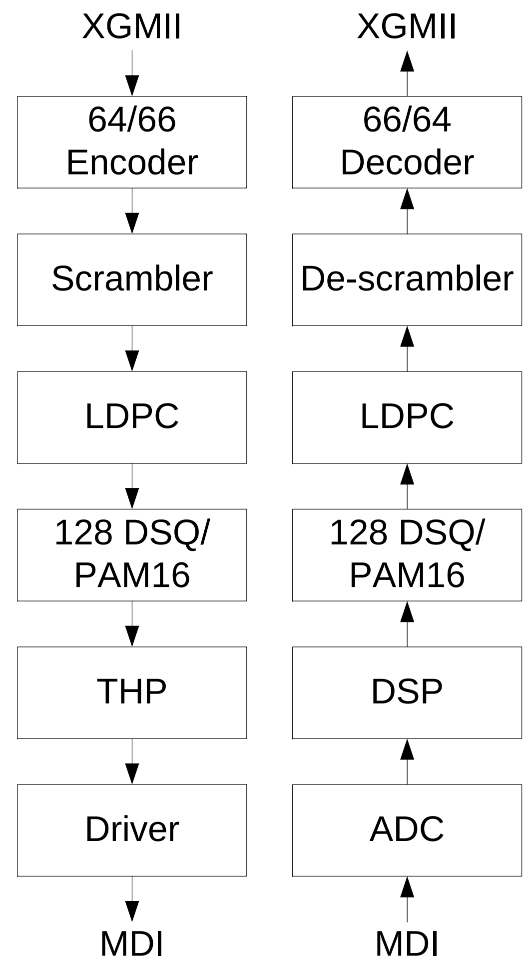
1000BASE-T1

- 750MHz, PAM 3, 1 channel



10GBASE-T

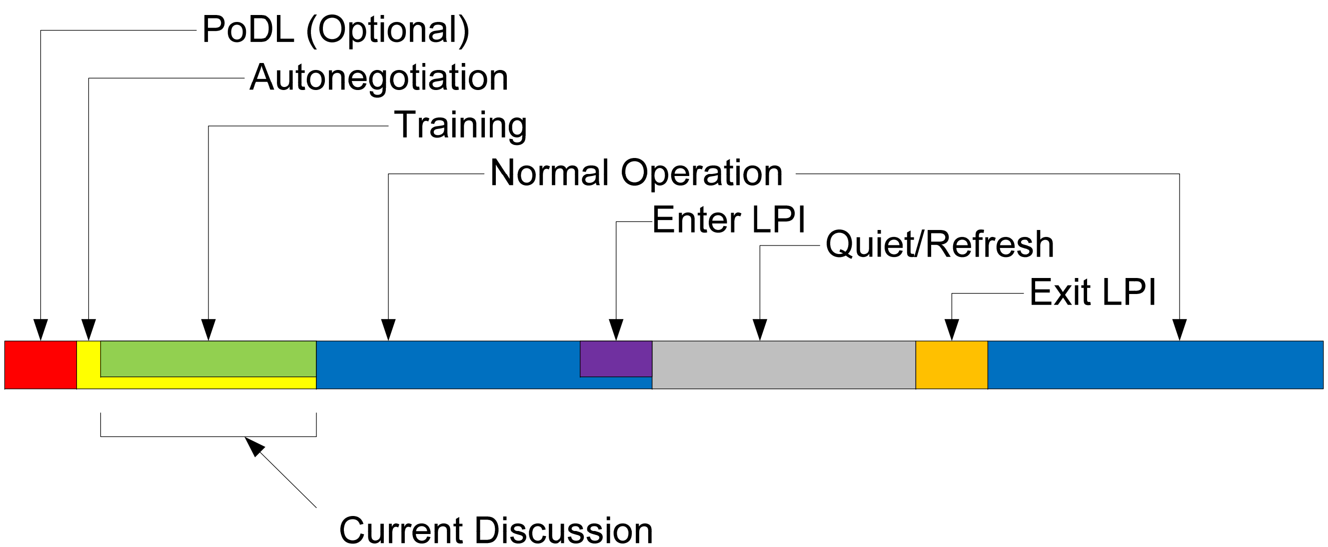
- 800MHz, PAM 16, 4 channels



Definitions (Since we parameterized everything)

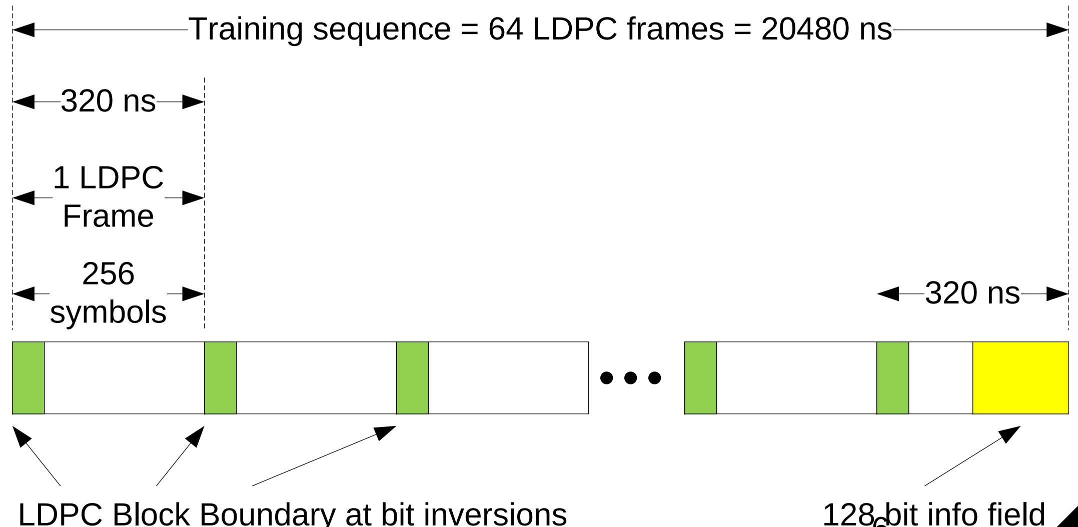
- RS_3 = Number of PAM3 symbols in Reed Solomon frame
- RS_T = Duration of Reed Solomon frame in ns
- PRS_3 = Number of PAM3 symbols in partial Reed Solomon frame
- PRS_T = Duration of partial Reed Solomon frame in ns
- PF = Number of partial RS frames per RS frame = RS_3 / PRS_3
- QRF = Number of RS_T frames time per EEE Quiet/Refresh cycle
- PFC = Partial RS frame count mod (QRF x PF)
- AF = Number of partial RS frames separating valid alert start points

PHY TRAINING



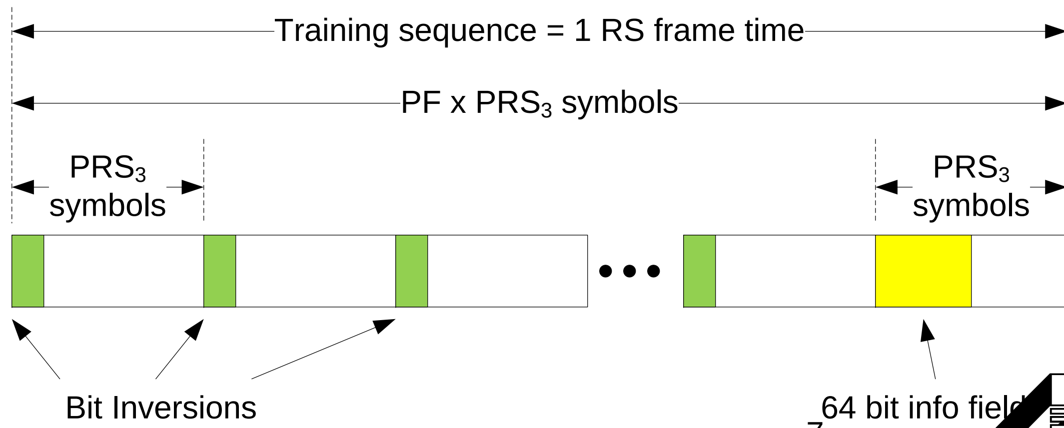
10GBASE-T Training

- PAM2 LFSR sequence for training
- First bit of 256 symbols inverted. Used for LDPC block boundary
  - LDPC frame duration is exactly 256 symbol time
- Final 128 bit of every 64th 256-symbol group XOR info field bits
  - Used to exchange training status with link partner
  - Exchange countdown timer to anticipate transition in training states
- EEE tracks LDPC frame number mod 512 for EEE timing purposes
  - EEE events transition on LDPC frame boundaries
  - Slave must track master training sequence within 1 LPDC frame



1000BASE-T1 Training

- **Use same PAM2 LFSR sequence for training**
- **Issue – 1 RS frame a lot longer than 1 LDPC frame**
  - **RS(180, 154, 2^{11}) = 1680 ns**
  - **RS(360, 308, 2^{11}) = 3360 ns**
  - **RS(630, 539, 2^{11}) = 5880 ns**
- **Want bit inversion and info field to occur more frequently given nosier 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 64 bits of PRS_3 symbol group to avoid offset calculations. Can make final instead 64 bits if we like.**



Info Field

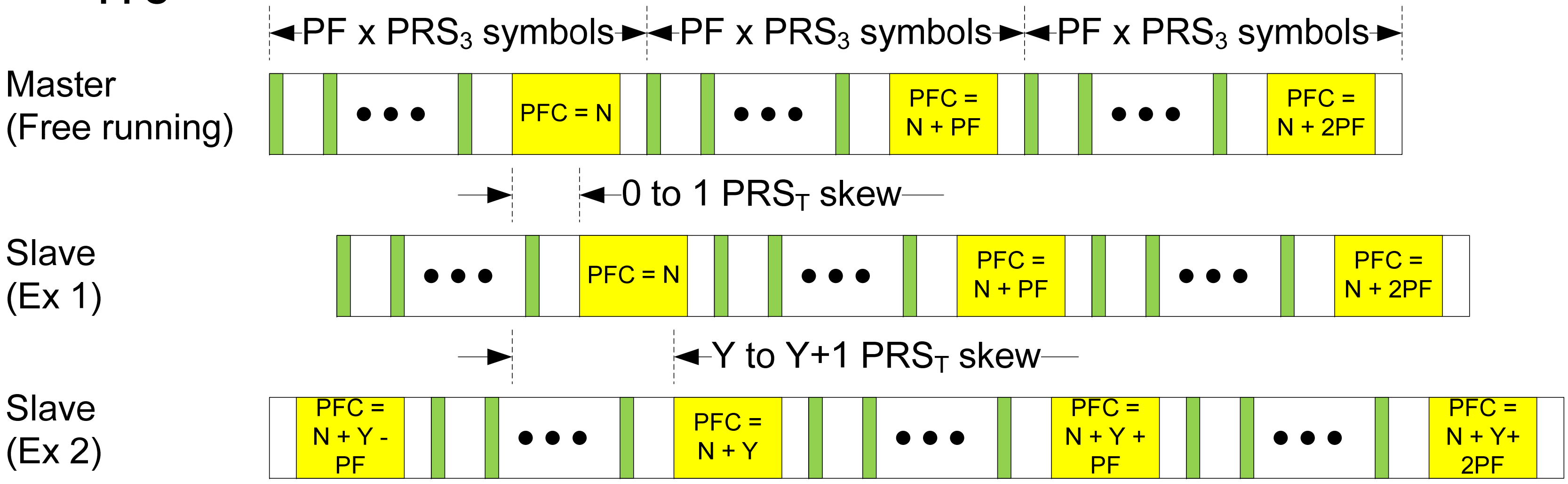
- **Simplify to 64 bits.**
  - **No need for PBO and THP**
- **No transition counter needed**
  - **No PBO or THP so no need to count down to readapt DSP to new TX settings**
  - **Significantly speeds up training**
- **Partial RS Frame Count (PFC) used to establish time synchronization for EEE**
  - **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**

1000BASE-T1
0xBB
0xA7
0x00
Message
Partial frame count mod (QRF 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

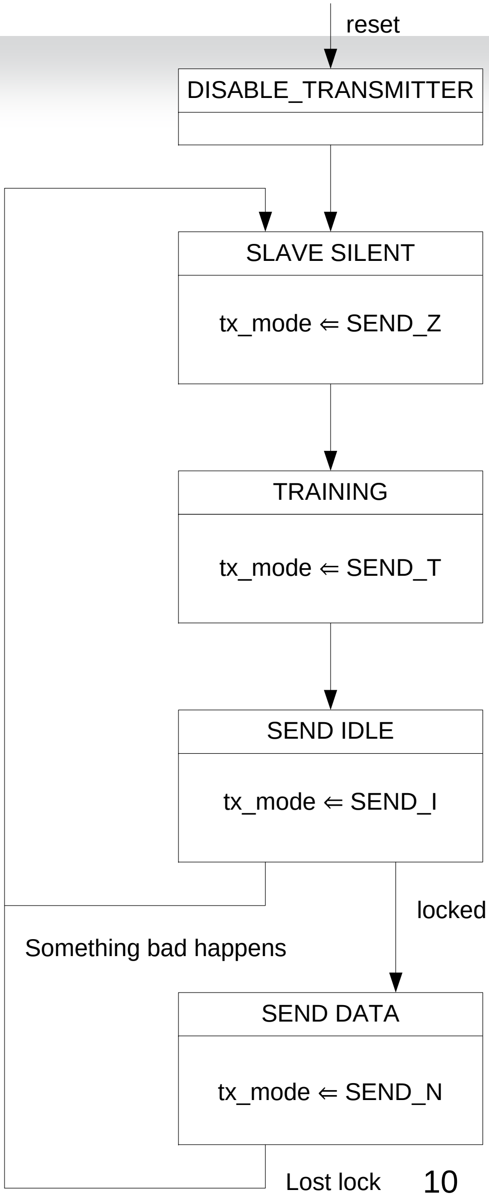
Example of slave partial RS frame count matching

- PF x PRS_3 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
  - No need to have slave info field within 1 partial RS frame of master (Ex 2)
  - Ok for slave to calculate offset
  - 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

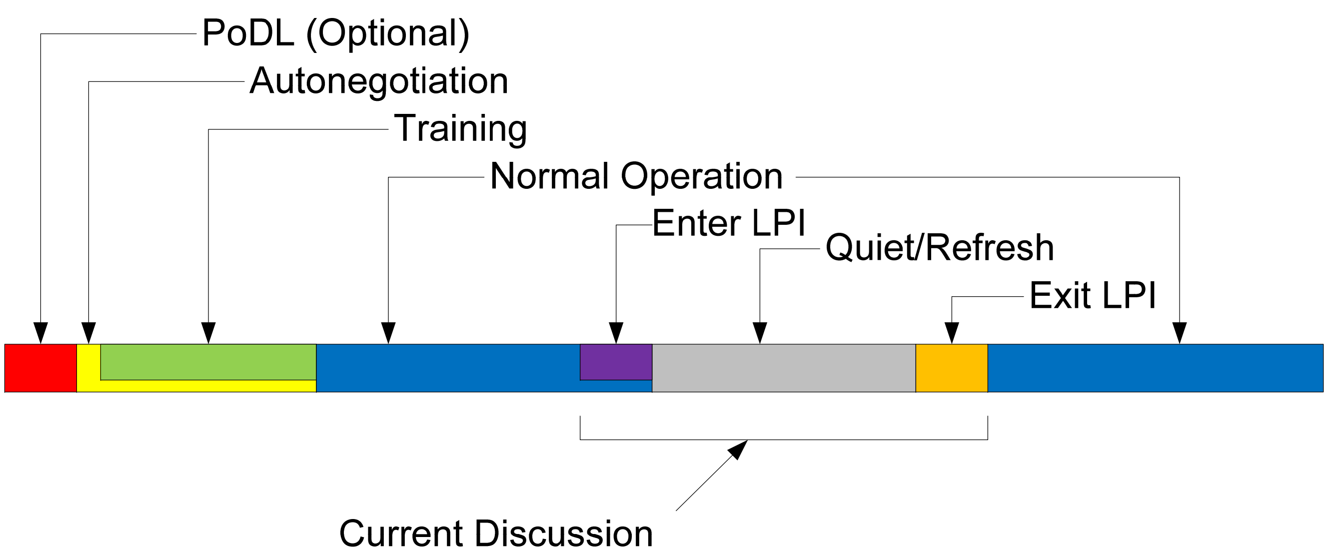


PHY Control State Machine

- 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



ENERGY EFFICIENT ETHERNET



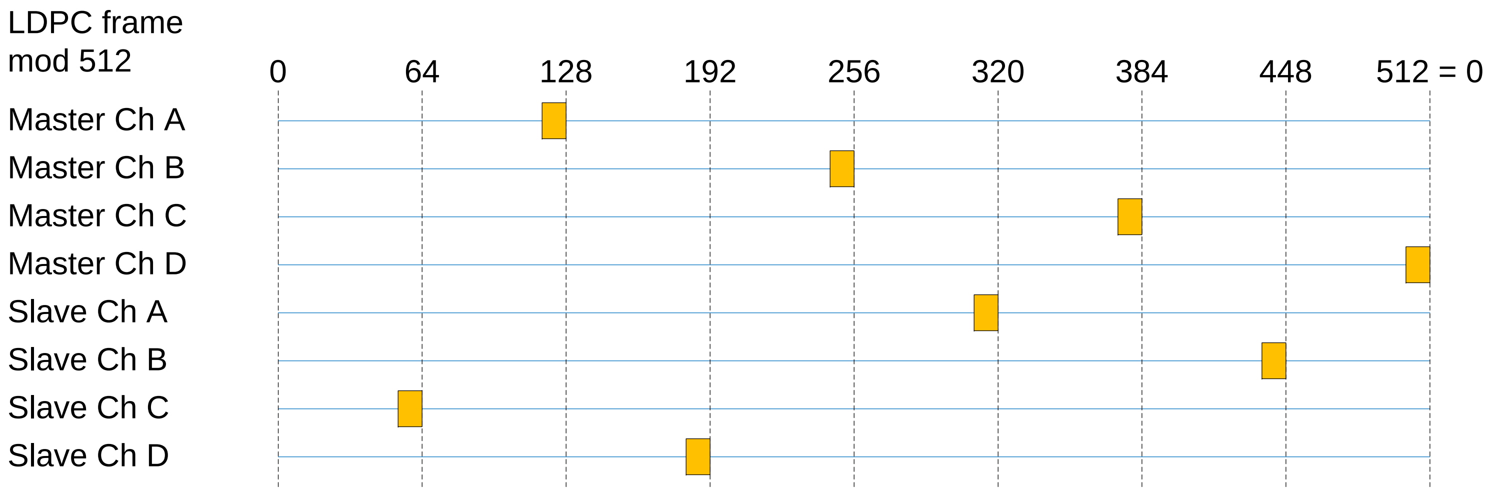
EEE – Entering LPI

- **10GBASE-T**
  - **If LPI seen on XGMII fill remaining bytes in LDPC frame with LPI symbol. Then send 9 more LDPC frames with nothing but LPI symbols.**
- **1000BASE-T1**
  - **Do the same thing by sending Enter_LPI_RS number of RS frames with nothing but LPI symbols**
  - **Don't really need to send too many RS frames.**
  - **Should be able to see lots of 8N/(8N+1) blocks with all LPI symbols in uncorrectable RS block**

EEE – Quiet / Refresh

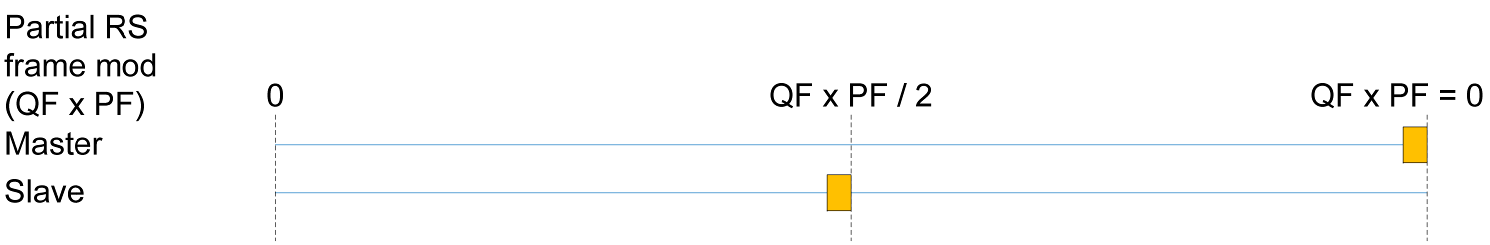
10GBASE-T

- **LDPC frame count between master and slave is within one LDPC frame**
- **Refresh time is 4 LDPC frame time**
- **Refresh spread out so only 1 PHY is sending refresh at any given time**
- **Refresh uses same PAM 2 LFSR as during training except info field is not sent**



EEE – Quiet / Refresh

- **1000BASE-T**
  - **Same concept except use partial RS frame**
  - **Refresh is Refresh_LPI x PRS_3 number of symbols**
  - **Quiet/Refresh cycle is QF x RS_T = QF x PF x PRS_T or QF x PF x PRS_3 symbols**
  - **Temporal location of partial RS frame count (PFC) is determined during training**

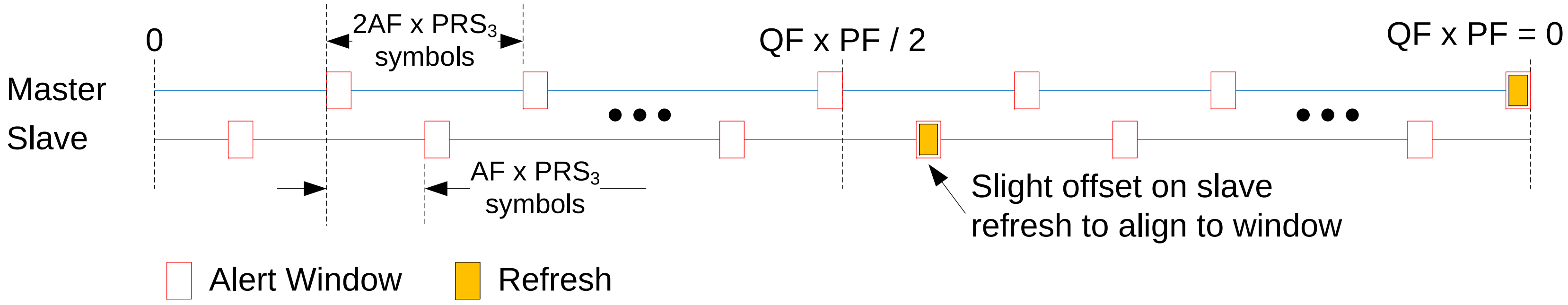


EEE – Exit LPI

- **10GBASE-T**
  - Send 128 bit PAM 2 Alert pattern 7 times, followed by 128 bit of zeros, followed by 9 LDPC frames of all idle symbols to exit LPI
  - Alert pattern can occur at any time but can only start at LDPC frame boundary
  - Alert pattern generated on channel A only for master and channel C only for slave
- **Implications**
  - Receiver attached to channel A or C cannot really shut down since alert pattern can occur any time
  - For 1000BASE-T1 with only 1 channel – neither PHY receiver can shut down with this scheme
  - Total wake latency for 10GBASE-T is 7 alert pattern + 128 zeros + 9 LDPC frames of idle + 1 LDPC worst case wait time to align to boundary = 4.48 us
  - Short wake up time limits analog power savings

EEE – Exit LPI, 1000BASE-T1 improvements

- **Use 1000BASE-T wake time of 16.5us instead of 4.48us of 10GBASE-T to allow more power savings**
- **Allow alert signal to be sent only during certain windows**
  - **Allows receiver to power down outside window**
  - **Stagger windows between master and slave so alert signal never overlap**
  - **Will increase worst case wake time waiting for window**
  - **Align refresh with alert window**
  - **Space alert windows 2 x AF x PRS_3 symbols apart and stagger master and slave windows by AF x PRS_3**



EEE – Exit LPI Procedure

- Similar to 10GBASE-T
- Send Alert_LPI number of alert sequences starting at valid alert boundary
- Send PRS_3 number of zeros
- Send Exit_LPI_RS number of RS frames with idle symbols only
- Resume normal operation

Suggested Parameter Values

- 11-bit RS symbols and 11bit – 7 PAM3 @ 750MHz

Symbol	Definition	RS(180, 154)	RS(360, 308)	RS(630, 539)
RS3	# PAM3 symbols per RS frame	1260	2520	4410
RST	Duration of RS frame (ns)	1680	3360	5880
PRS3	# PAM3 symbols per partial RS frame	90	90	90
PRST	Duration of partial RS frame (ns)	120	120	120
PF	# partial frames per RS frame	14	28	49
QRF	# RS frame per quiet refresh cycle	128	64	36
AF	# partial RS frames separating alert	28	28	28
Refresh_LPI	# partial RS frames for refresh	12	12	12
Enter_LPI_RS	# RS frames with all LPI to enter LPI	4	2	1
Alert_LPI	# alert sequences to exit LPI	11	11	11
Exit_LPI_RS	# RS frames with all idles upon exit LPI	4	2	1
Alert_sym	# symbols in alert sequence	90	90	90

Choice of partial RS frame size (PRS_3) not arbitrary

- During LPI scrambler has to keep running with all bytes of 8N/(8N+1) encoder being LPI symbols
- After exiting LPI the RS frame can align to any partial RS frame boundary to minimize wake time
- Hence PRS_T must have a duration of exactly an integer multiple of 8N ns to maintain scrambler sync
- $PF = RS_3 / PRS_3 = RS_T / PRS_T$ must be an integer

Choice of Alert sequence

- 90 bits long to match PRS_3
- Large auto correlation for reliable detection
- Master

-1	-1	1	1	-1	-1	1	1	-1	-1
1	1	1	1	-1	-1	-1	-1	1	1
1	1	-1	-1	1	1	1	1	1	1
-1	-1	1	1	1	1	-1	-1	1	1
-1	-1	-1	-1	1	1	-1	-1	-1	-1
1	1	1	1	1	1	-1	-1	-1	-1
-1	-1	1	1	-1	-1	1	1	1	1
1	1	1	1	-1	-1	-1	-1	1	1
-1	-1	1	1	-1	-1	-1	-1	-1	-1

- Slave

-1	-1	-1	-1	-1	-1	1	1	-1	-1
1	1	-1	-1	-1	-1	1	1	1	1
1	1	1	1	-1	-1	1	1	-1	-1
-1	-1	-1	-1	1	1	1	1	1	1
-1	-1	-1	-1	1	1	-1	-1	-1	-1
1	1	-1	-1	1	1	1	1	-1	-1
1	1	1	1	1	1	-1	-1	1	1
1	1	-1	-1	-1	-1	1	1	1	1
-1	-1	1	1	-1	-1	1	1	-1	-1

Next Steps

- Propose we use Clause 55 as the starting point for training and EEE
- Propose we adopt modifications shown here as 1000BASE-T1 baseline
- Need more work to fine tune the parameters

THANK YOU