EEE Support for 100 Gb/s

Draft

Mark Gustlin, Hugh Barrass

IEEE P802.3bj

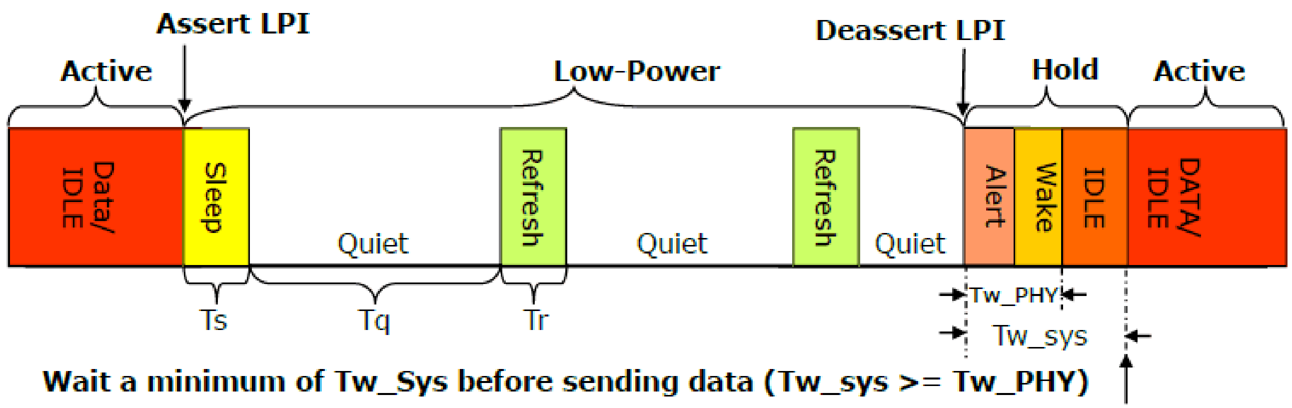
????

EEE for 100 Gb/s Overview

- This presentation will review the technical issues that need to be addressed in order to add EEE to a 100 Gb/s copper interfaces
- This will evolve into a baseline proposal over time
- One significant issue has to do with the Alignment Marker lock time, a proposal to address this concern is described
- Details and examples of other considerations for EEE at 100 Gb/s are also explored
- This presentation assumes re-using Low Power Idle, it does not investigate the complexities involved in any type of modular EEE

EEE Overview

LPI Overview



- LPI – PHY non-essential circuits shut down during idle periods
- During power-down, maintain coefficients and sync to allow rapid return to Active state
- Wake times for the respective backplane PHYs:
  - 1000BASE-KX: Tw_PHY(min) = 11.25 usec
  - 10GBASE-KX4 Tw_PHY(min) = 9.25 usec
  - 10GBASE-KR: Tw_PHY(min w/o FEC) = 12.25 usec
  - 10GBASE-KR: Tw_PHY(min w/FEC) = 14.25 usec

From: bennett_01_0311

Note that the term ‘Wake’ is overloaded in the above diagram as it is in the standard

EEE Overview

- Wake time range is 9 to 14usec for existing EEE PHYs
- Note that the wake time does not scale down with speed even though data accumulates faster at higher interface speeds
- So for 100 Gb/s should we shoot for a wakeup time of < 5usec?
  - Note that in 5 µs, 0.5Mb of data accumulates, per port

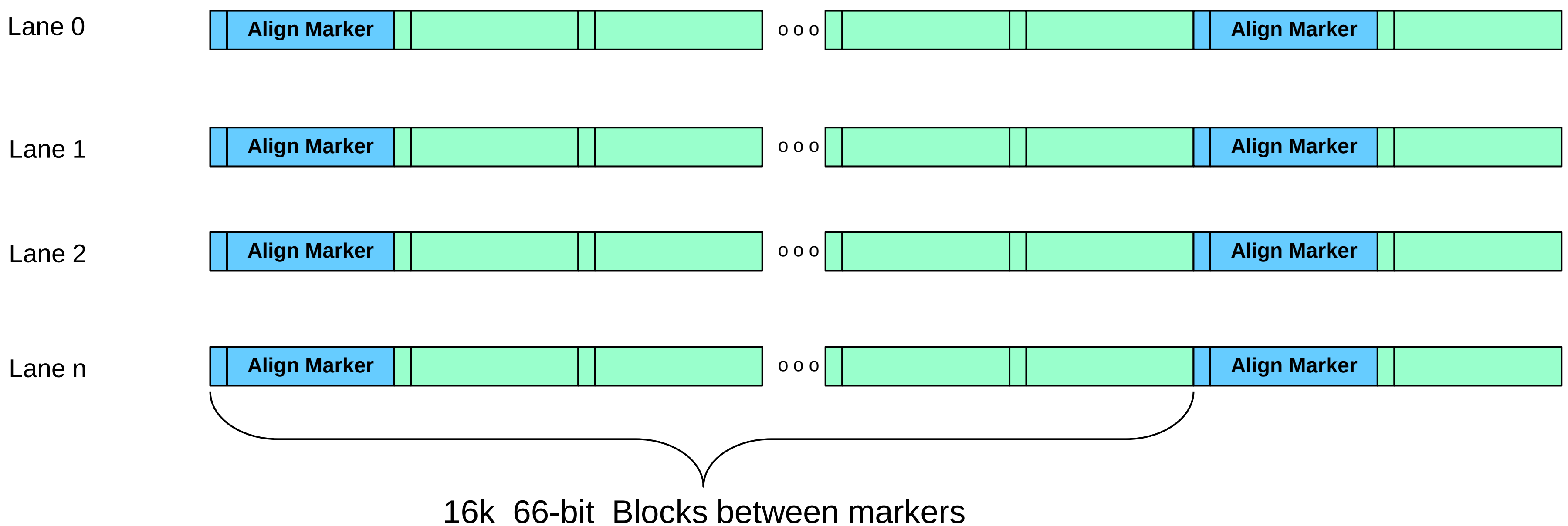
- Are there any concerns in the 100 Gb/s PCS that would prevent us from supporting a 5 usec or faster wakeup?
  - Alignment marker lock is >> 5 µs, the next few slides look at this issue

Underlying Assumptions

- Many solutions can be proposed to solve the quick link bring-up issue, depending on the assumptions that are made
- For instance, if you assume that only the PMD lanes are powered down, and that the PCS and PMA stay powered and unchanged, then you could make some simplifying assumptions
  - Skew change is very limited
- But that might limit how much power savings could be achieved
- If the PMD is powered off, then PCS lanes can move locations (due to the gearboxing), and therefore Alignment Markers are needed to find PCS locations
- In this paper is it assumed that the PCS, PMA and PMD can be powered down and therefore that:
  - PCS lanes can move locations
  - A solution should handle the maximum skew as specified in 802.3ba

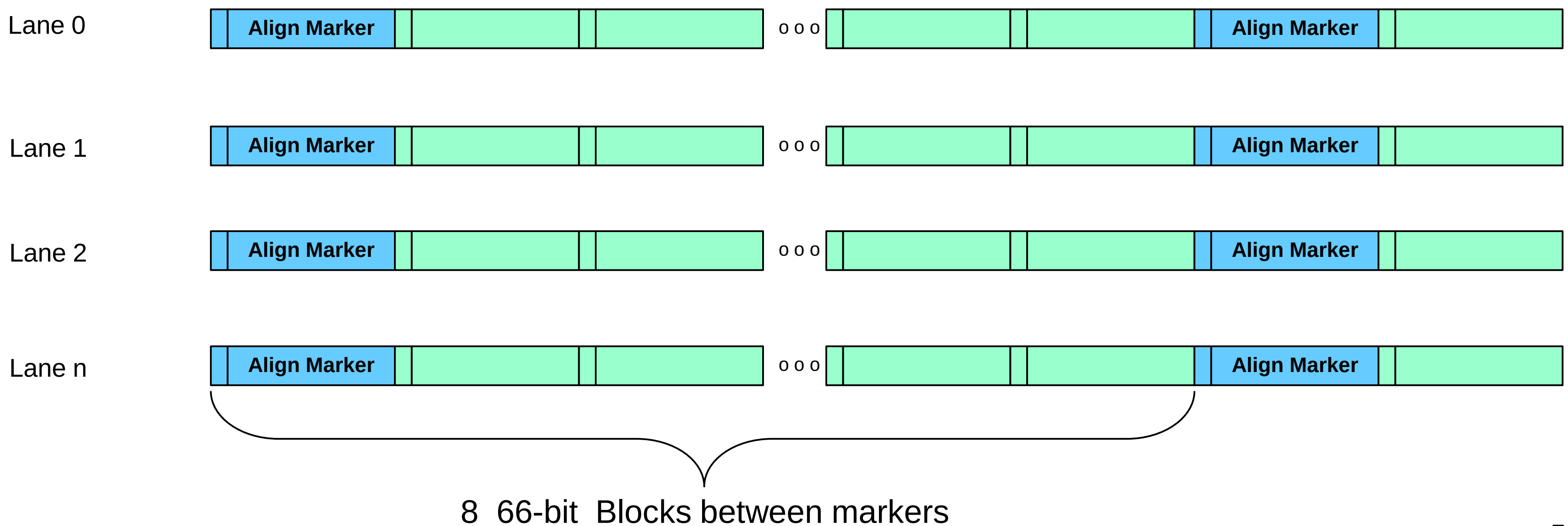
100/40GE Standard Alignment Marker Distance

- The alignment markers are widely spaced for 100 Gb/s and 40 Gb/s, 16k blocks apart on each PCS lane
- The alignment marker lock SM looks for two that match in a row before declaring lock and allowing alignment, so that is 16384*2 * 66 * 194ps = 419µs (for 100GE, ½ that for 40GE)
- This would mean that startup would take > 400µs today!
  - Ok for 802.3ba, not ok for a EEE interface



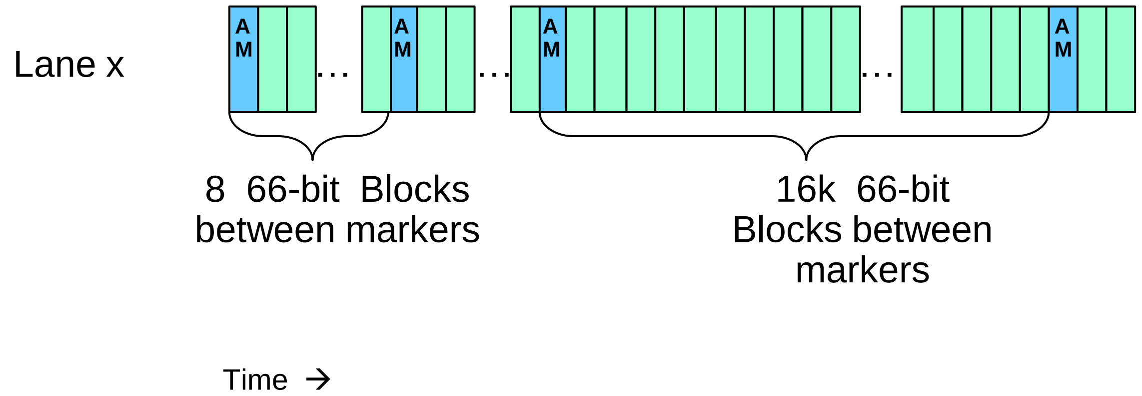
Start-up AM Distance Reduction

- When the lanes are starting up, reduce the distance between AMs temporarily
- The allowed minimum distance could be dependent on the total skew that is allowed (14 66-bit words for 100G at the RX PCS), or you could require that the receiver maintain last known alignment and only worry about the skew variation (44 bits total at the receiver, less than 1-66b word)
- The proposed option is to take advantage of the count down field which is located within the Rapid Alignment Marker (RAM) word, this allows the RAMs to be placed as close as we desire
  - Receiver can differentiate AMs on PCS lanes using the CD field
- So let's say every 8 words there is an alignment marker until startup is finished, then revert to the normal distance
- Alignment Marker lock would now take at least 8 * 2 * 66 * 194ps = 205ns (for 100GE), it can take longer with errors



Start-up Alignment Marker Distance

- When the lanes are starting back up, then Alignment Marker spacing is small, 8 for instance
- Then after a fixed time (or negotiated time), the spacing goes back to 16k
- The transmitter has to signal to the far end when this change will take place, this can be done through repeated signals such as a count down field in the Alignment Marker, so that the receiver will know when the transition will take place even in the face of errors on the link
- The receiver should lock to the sync field in the AM in order to get 66b alignment, instead of taking 64 or more words to do that



How to Signal a Change in AM Distance?

- **Standard Alignment Marker format:**

10	**Marker x**	**BIP**	**!Marker x**	**!BIP**

- **Proposed Rapid Alignment Marker format:**

10	**Marker x**	**BIP**	**!Marker x**	**CD**

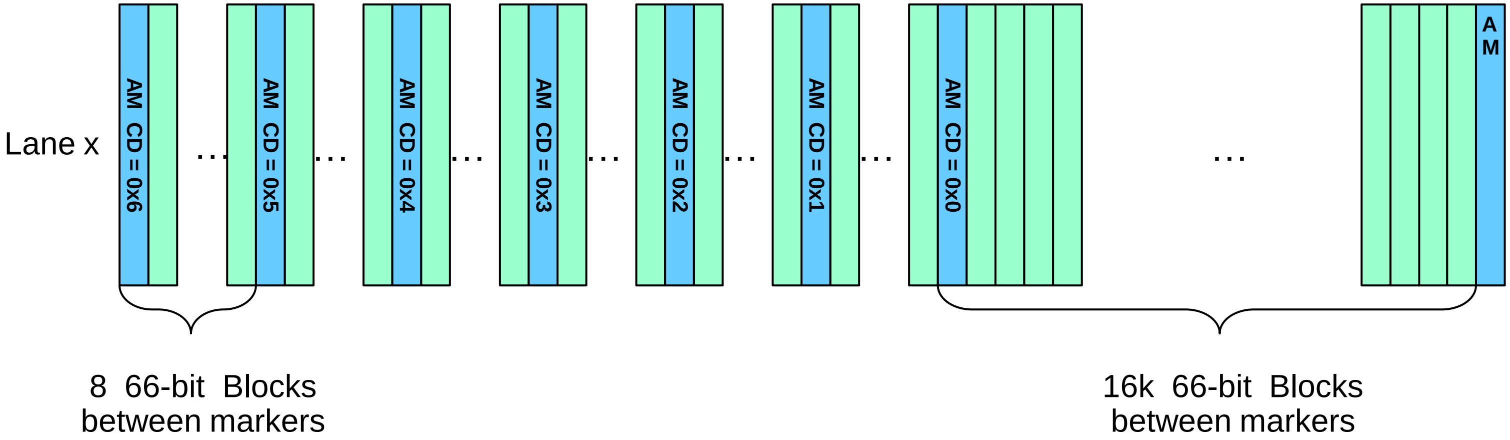
- Add a count down field to the alignment marker, the receiver can use this to predict when the transmitter will switch the distance of the alignment markers
- Note that the AM is not scrambled, so anything we add should not negatively impact the baseline wander or clock content
- Assume that we want to be resilient in the face of up to 3 individual errors
- Convert the !BIP field to a Count Down field for the rapid AMs
- Encoding of the CD field is discussed on the next slide

Count Down Field Encoding

- It was brought up during discussions that if we do use a count down field in the AM (which is not scrambled), then when we mux multiple PCS lanes together, you can adversely impact the clock content of the aggregate data-stream, how can we fix that?
- The count down field is proposed to be 8 bits, if we xor a true count down field with the first byte of the alignment marker, what does that look like?
- For instance the last 10 entries of the count down field for AM0 would be (M0 PCS lane 0 = C1: C8, C9, C6, C7, C4, C5, C2, C3, C0, C1.
- For instance the last 10 entries of the count down field for AM1 would be (M0 PCS lane 1 = 9D: 94, 95, 9a, 9b, 98, 99, 9E, 9F, 9C, 9D.
- When we mux the above it should give us very good clock content when compared to the previous proposal (0x0f countdown, the same on all lanes).
- Easy to derive the count down field since the receiver knows the M0 value for each PCS lane, simply xor the count down field with M0 to see where you are in the sequence.
- Would need to simulate for the impact to baseline wander – Pete Anslow has agreed to run the simulations
- Start the count down field at value *x* so it reaches 0 at the end of the wake time (*x* TBD depending on the wake time)

Start-up Alignment Marker Distance

- The figure shows the short spacing of the AMs on link power up
- This occurs on all PCS lanes at the same time
- The receiver will look for multiple AMs at the short spacing, and look at the count down fields, compare and lock in the count down so that it can predict when the spacing returns to 16k words
- Once AM spacing returns to 16k, the !BIP field returns to it normal .ba function
- BIP values are still calculated the same way as is standard, just over shorter distances when the link is coming up, and the !BIP7 is not populated

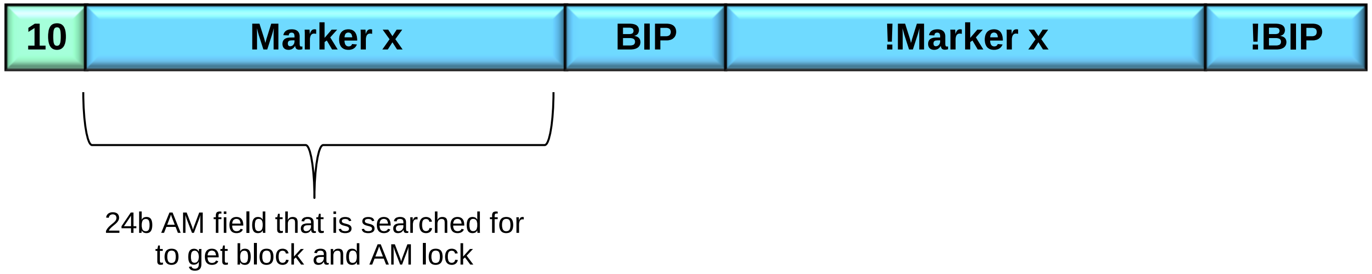


Note that the CD field is shown unencoded

Block Alignment with AMs

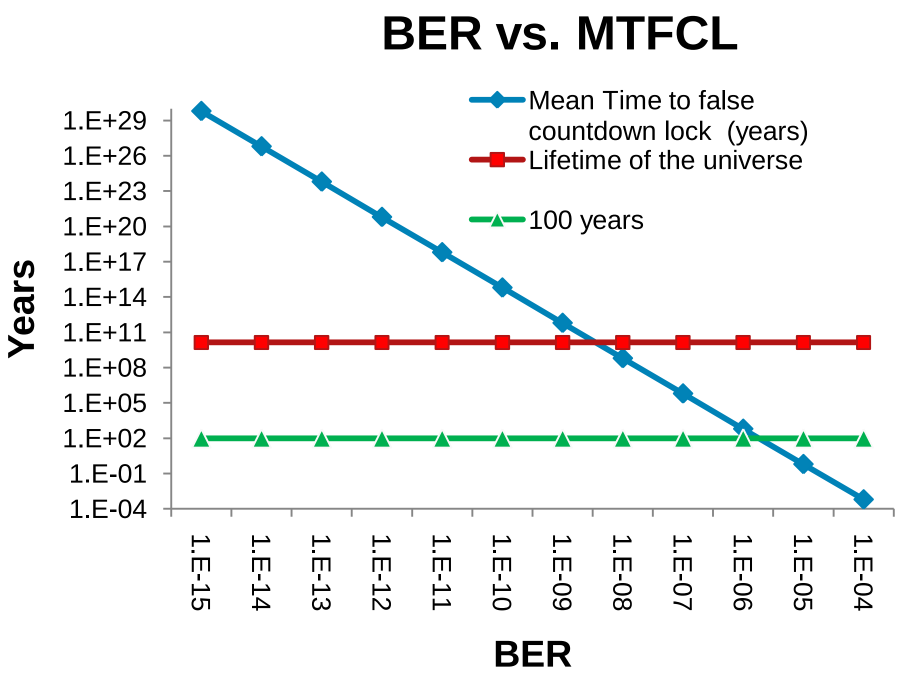
- In 802.3ba each PCS lane has two processes in order to get into lock, first block lock is run on each PCS lane, and then alignment marker lock is run on each PCS lane
- Block lock takes at least 64 blocks if there are no errors, and if sync headers are searched for in parallel. It can take much longer if there are errors or if sync headers are searched serially
  - Best case for 100G is therefore 66 * 64 * 194ps = 819ns just for block lock
- If we are trying to power up the interface in a very small number of µs then this is a significant number (a single bit error at the wrong time can double this time)
- What to do about this? If we already have the Rapid Alignment Markers being sent, then we can directly lock to the RAMs. Receiver will do a parallel search for the 24b marker field (across all 66b positions), once a match is found then look n blocks away and lock once 2 are matched, you then declare lock (both block and AM lock at the same time)

- **Standard Alignment Marker format:**



Count Down Field Sync

- What would the SM for CD sync look like?
- The CD field is 8 bits, if we look for 3 CD fields in a row that match our expectations (n, n-1, n-2), what is the probability of a false lock?
- This is per PCS lane, assuming interface cycles every 10us
- When an interface uses FEC, this lock would be pre-FEC
- Is this sufficient?



Link Start-up

- When a device first powers up, it will power up sending 802.3ba AMs, 16k apart
- What if a fault causes the receiver to miss startup, and it first sees AMs on a power bring up from LPI?
    - State Machines already handle these kind of issues for 802.3az EEE, so the same thing would apply, the SM would have timeouts to handle the error cases, no need to do anything else.

BIP Coverage

- With the very short AM distance on LPI startup, does the BIP field stay as is? And just covers a lot less data?
- Proposal is yes, that BIP values are still sent as is, but instead of covering 131k bits or more, each BIP bit covers only 64 bits or so (8*8).
- Note that the receiver should not signal BIP errors until after the rapid AMs are locked to (prevents spurious BIPs from the initial LPI startup phase)

Data ‘Randomness’

- There was a lot of work done during the 802.3ba project on baseline wander and clock content of the 100 Gb/s data stream and various sub 100 Gb/s lanes (10G CAUI lanes, 25G PMD lanes etc.) in order to ensure that the characteristics of a given serial data stream are good, see:
  http://www.ieee802.org/3/ba/public/jan08/anslow_01_0108.pdf
  http://www.ieee802.org/3/ba/public/nov08/anslow_06_1108.pdf

- When we send RAMs, does that impact the baseline wander or clock content negatively?
  - Simulations need to be run determine this
- There is some concern with when the lanes are being powered up, will the randomness of the data being sent at that time be sufficient to quickly train the receivers?
  - Simulations need to be run determine this
- The count down field is a new concept to this protocol, if you bit multiplex multiple streams together with an 802.3ba PMA, how will this impact the clock content (given that the count down field is not scrambled)?
  - A new encoding scheme has been proposed in this paper to address this concern, simulations need to be run confirm that this is not an issue
- Pete Anslow has volunteered to run some simulations in order to quantify the impact of the RAMs

Start-up timing

- How to know when to start sending and then stop sending rapid AMs?
- Proposal is to send Rapid AMs in TX_ALERT and TX_WAKE states, for 10GE that is 5usec total time
- Only Idles or LPIs are sent at this point

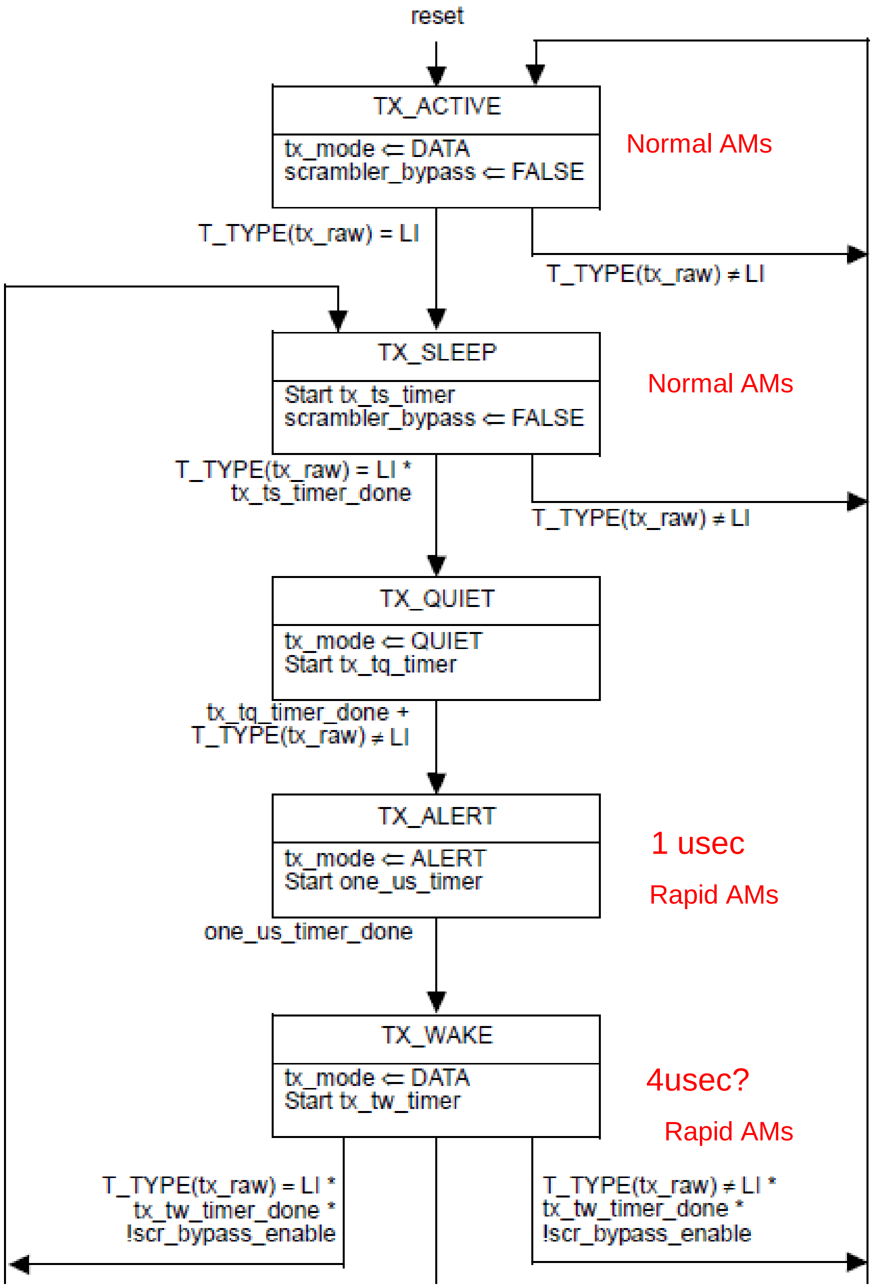
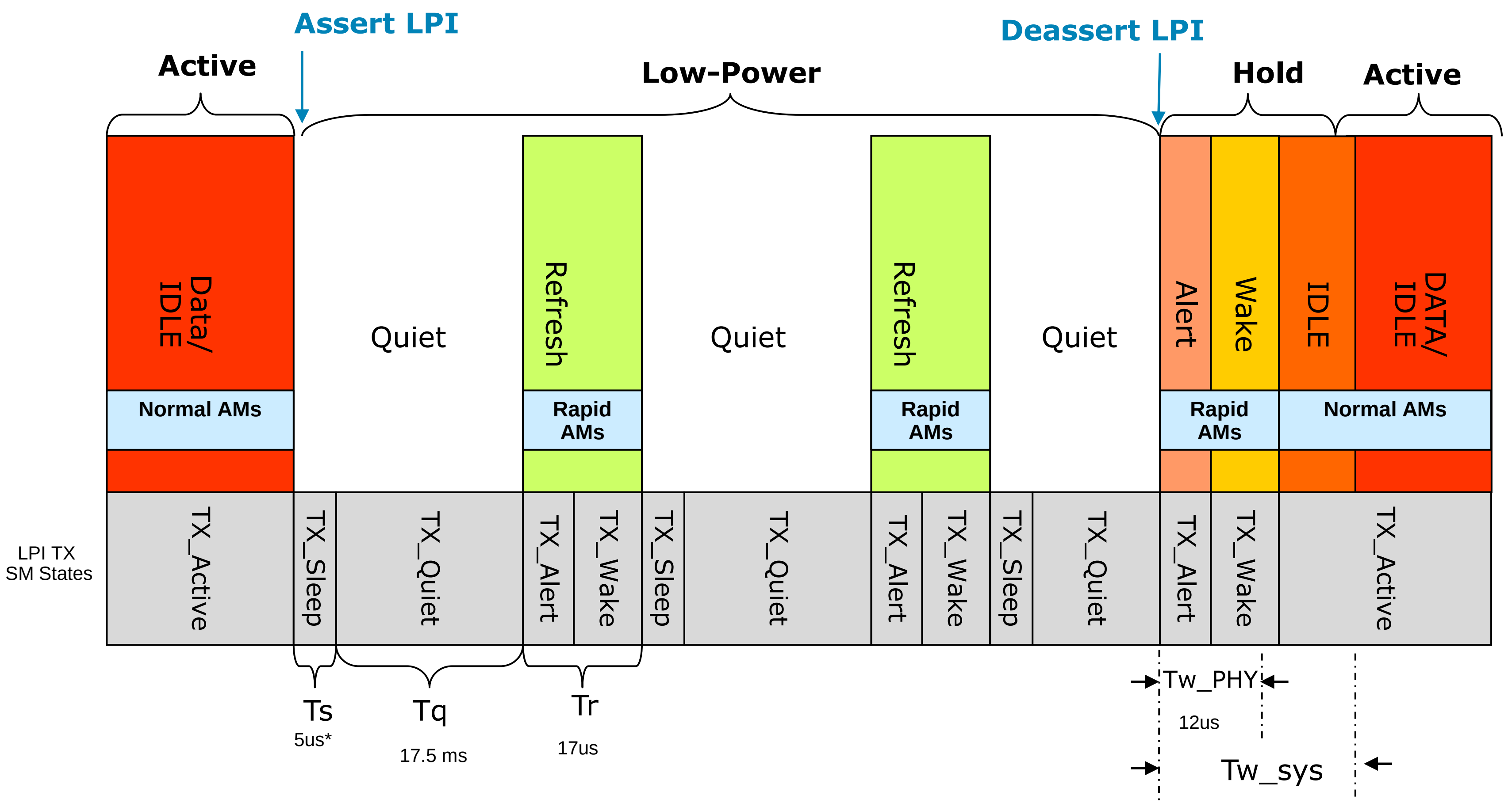


Figure 49-16– LPI Transmit state diagram

LPI with Rapid AMs



* Numbers are from 10GBSE-KR

Room for Rapid AMs?

- In 802.3ba the Alignment Markers are sent very infrequently, every 16k blocks on each PCS Lane
- This allows room for the AMs to be added into the data stream by deleting Idles periodically, just as is done for clock compensation
- If we send AMs rapidly, then we can still delete idles in order to send AMs?
  - Yes this works since rapid AMs are only sent on link re-start when transitioning out of LPI, so only LPI or Idle is being sent at that time
  - Proposal is that Either Idles or LPIs can be deleted to add in the AMs

FEC and EEE

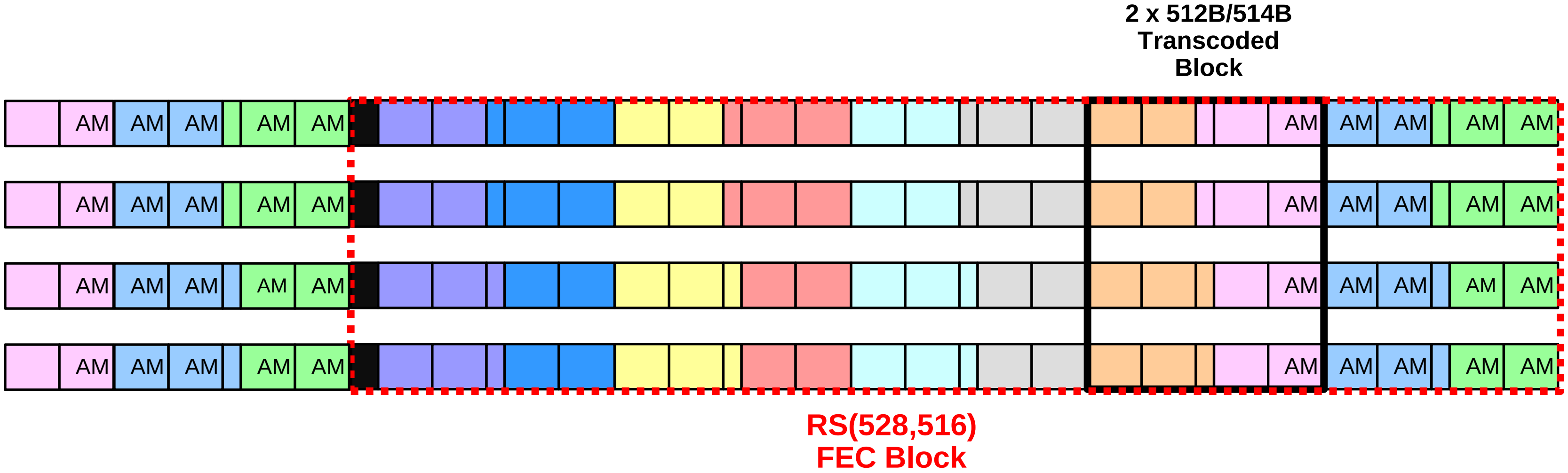
- How would 802.3bj FEC impact EEE bringup time?
- In the original EEE, they did things such as have a scrambler bypass so that the receiver can quickly lock up to a known FEC location
- For EEE and 100G, what do we need to do?
- The current direction of bj task force on FEC is some sort of transcoding and then FEC across the lanes. Also there is the plan to align the alignment markers to the beginning of the FEC block

RAMs and EEE

- Lets look at an example FEC/transcoding option, using 512B/514B transcoding + RS(528,514) m=10 from cideciyan_01a_1111
- The Alignment Markers are justified to the beginning of the FEC block, normally once every 16k/4 = 4k FEC blocks
- In EEE bring-up mode, if you send rapid AMs, just send them on some multiple of 4 (4, 8, 12 etc) in order to always justify them at the beginning of the FEC block

  Most FEC blocks being looked at within the task force can be handled in some similar manner

- This will allow us to quickly find the beginning of a FEC block for rapid alignment (by hunting for AMs), no special scrambler reset needed



Document Changes Required

If we add EEE to 100 Gb/s, what clauses have to be modified:

Clause 30: Management additions

Clause 45: MDIO register additions

Clause 74 (KR FEC): Minor changes if we include 40 Gb/s as a service to humanity

Clause 78: Add in overview of 100 Gb/s EEE, timing parameters etc

Clause 81 (RS/MII): Add in changes that are similar to those made in clause 46 for KR

Clause 82 (PCS): Add in changes that are similar to those made in clause 49 for KR, plus add in changes needed for the Rapid Alignment Marker support

Clause 83 (PMA): Add in signals that pass through the PMA (energy_detect for instance)

Clause 84 (40GBASE-KR4): EEE PMD changes if we include 40 Gb/s as a service to humanity

Clause 85 (40GBASE-CR4/100GBASE-CR10): EEE PMD changes

Any new clauses created for 802.3bj, PMD and others will require appropriate additions.

Things to do?

- PMD investigation, how fast can things power up? What tradeoffs are to be made here?
- Do we need some kind of alert signal, today that functions to ensure that energy detect is cleanly made

EEE for 100 Gb/s Summary

- The majority of the EEE protocol that was developed for 10 Gb/s can be applied directly to EEE at 100 Gb/s (Low Power Idle)
- The main protocol hurdle that 100 Gb/s has to overcome in order to support EEE is the Alignment Marker distance and startup time due to this distance
- Presented is a methodology to change to the Alignment Marker protocol which will greatly reduce the start up time and enable efficient EEE support at 100 Gb/s
- FEC also does not seem to be an issue with EEE, this will depend ultimately on what FEC mechanism is chosen

Thanks!