# LPI SIGNALING ACROSS CLAUSE 108 RS-FEC
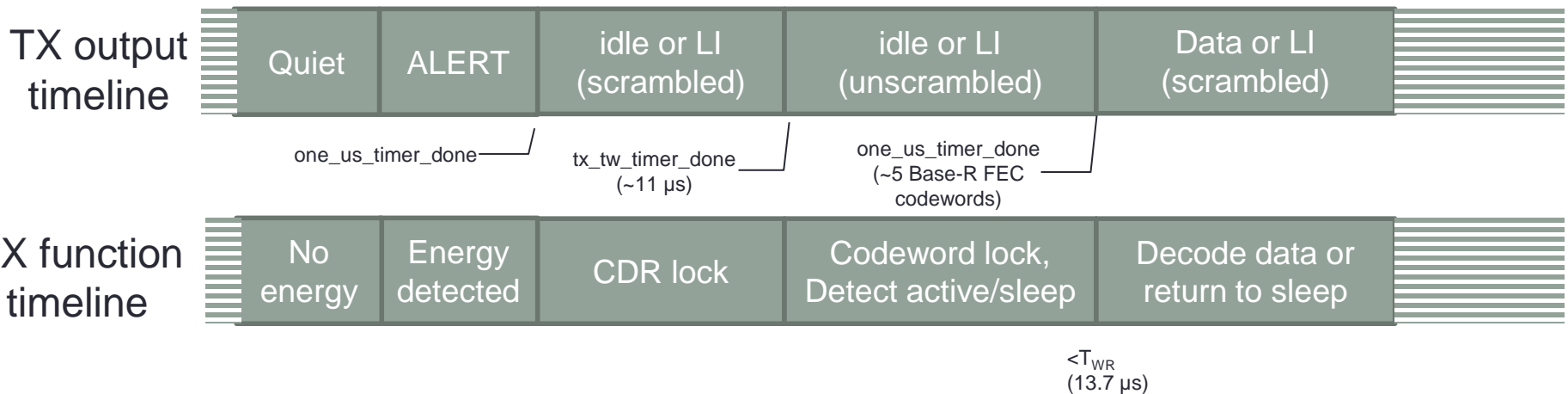
Adee Ran

Background

- LPI original functions
  - TX informs the RX that it's about to enter LPI (SLEEP).
  - TX shuts down the signal (QUIET); RX PCS generates /LI/ towards the MII during this period.
  - TX periodically turns on the signal (WAKE), and sends LI (for refresh) or idles (for exiting LPI).
  - RX is triggered by WAKE and detects the received data; determines whether to go back to SLEEP (if LI received) or to ACTIVE (if idles).

TX states: Active → Sleep → Quiet → Alert → Wake

RX states: Active → Sleep → Quiet → Wake → Fail
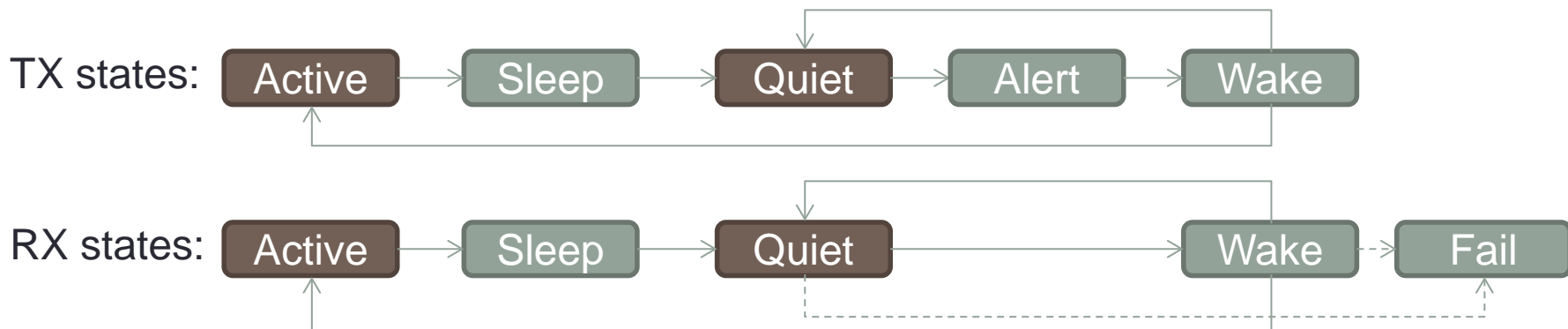
- That worked simply for 10GBASE-KR.
- For clause 74 FEC, a tweak was added: during WAKE, PCS bypasses scrambling to enable fast FEC alignment.

TX output timeline

Quiet	ALERT	idle or LI (scrambled)	idle or LI (unscrambled)	Data or LI (scrambled)

one_us_timer_done

tx_tw_timer_done (~11 μs)

one_us_timer_done (~5 Base-R FEC codewords)

RX function timeline

No energy	Energy detected	CDR lock	Codeword lock, Detect active/sleep	Decode data or return to sleep

$<T_{WR}$ (13.7 μs)

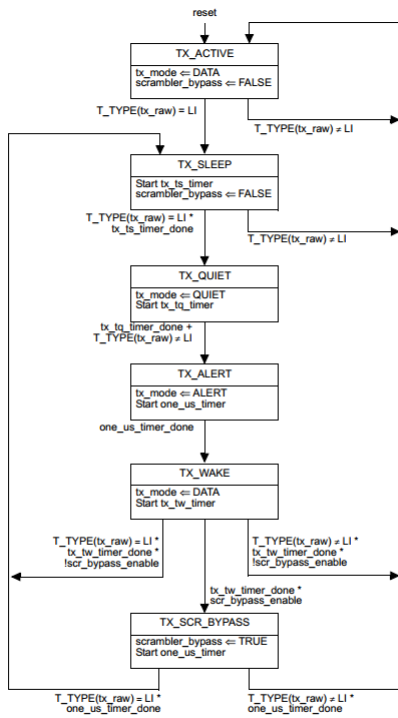# Clause 49 LPI state diagrams

- Are quite simple
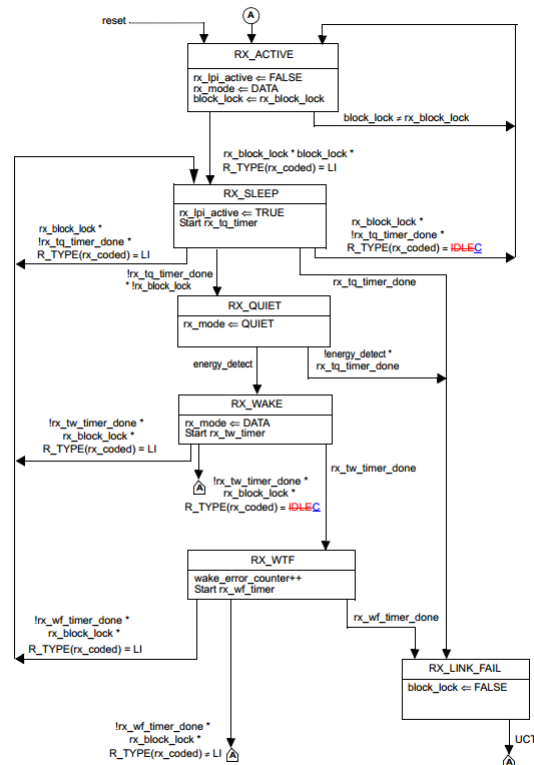
Figure 49–12—LPI Transmit state diagram

Figure 49–13—LPI Receive state diagram

Additions in 802.3bj

- 802.3bj added EEE functionality for all 40G and 100G PHYs (using clause 82 PCS) – including optics and RS-FEC support.

- Challenge: Multilane PCS and RS-FEC have to re-align using AMs, which would take too much time with normal AM spacing
  - Solution: rapid alignment markers (RAMs) used during LPI transitions. Countdown to transition from RAMs to AMs is transmitted from TX to RX.

- Challenge: Optical PMDs have very long power switching times
  - Solution: fast-wake mode – keep PCS-to-PCS signaling active for transmission of the /LI/ characters; BER may be higher (allowing some power saving). Upper layers may use LPI indication for more significant power saving.
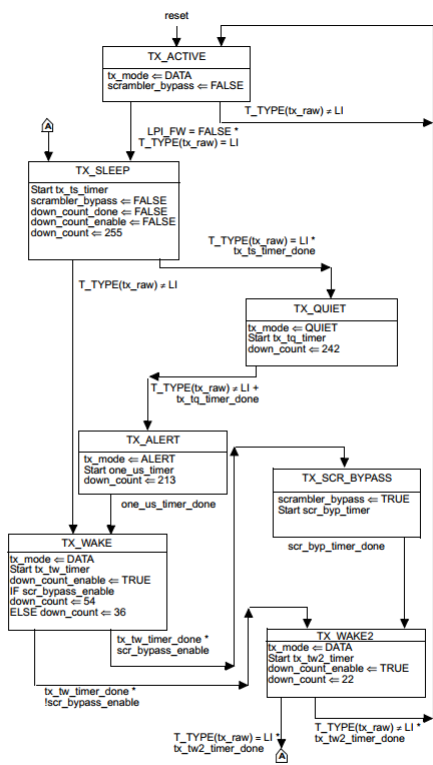
Clause 82 LPI state diagrams

- Are more complex

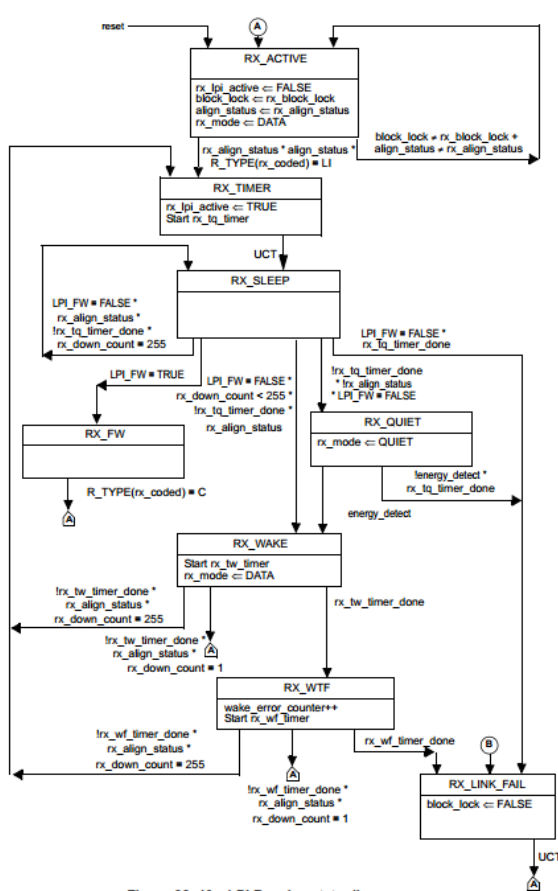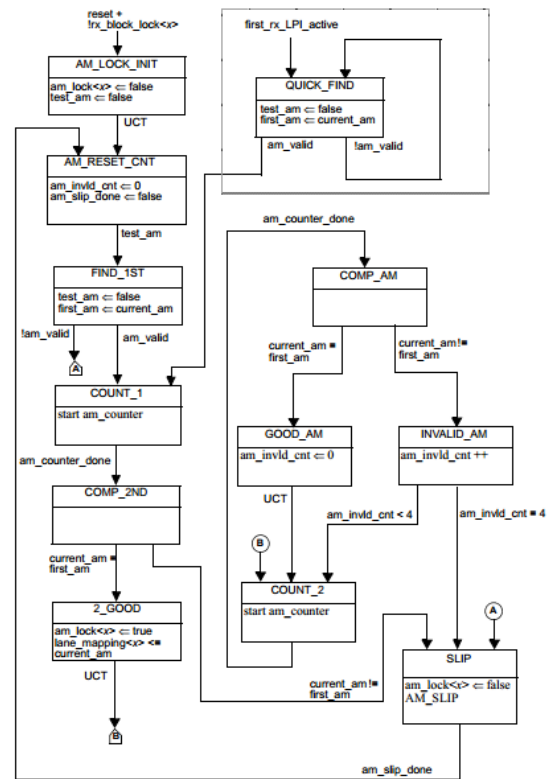Figure 82–18—LPI Transmit state diagram

Figure 82–19—LPI Receive state diagram

Figure 82–13—Alignment marker lock state diagram

LPI signaling in 802.3by

- No-FEC and Base-R FEC modes are already handled in D0.1 – using the same method of 10GBASE-KR.
- For the RS-FEC mode, we have several possible directions:
  - Allow only fast wake to be used with the RS-FEC sublayer.
  - Follow 802.3bj deep sleep – use rapid codeword markers (RCWMs).
  - Follow 10GBASE-KR – use the PCS scrambler bypass function to enable fast codeword alignment.
  - Use a new, different method (?)

Adding RCWMs to clause 108

- In the 40G/100G PHYs, AMs and RAMs are handled by the clause 82 PCS, which also has the LPI state
  - AMs/RAMs and LPI are interrelated
  - Clause 91 RS-FEC has an additional task of interpreting down_count field in the RAMs (in both TX and RX directions)
- In 802.3by, only the PCS has the LPI state, and only the RS-FEC handles CWMs…
  - The RX LPI state diagram would need down_count to know when to expect switching from rapid to normal CWMs.
  - To generate down_count, the RS-FEC should infer an LPI state from its service interface (FEC:IS_TX_MODE.request) and/or incoming data (idle or LPI)…

Issues

- Using rapid CWMs the way they are used in clause 82/91 requires adding countdown and creating new state diagrams, not parallels of clause 49 or 91, for both TX and RX LPI operation
- May need to sync these diagrams with PCS variables
- New two-sided state diagrams need careful design and review – prone to bugs

New TX and RX LPI state diagrams in clause 108

?

Using scrambler bypass with RS-FEC

- Bypass scrambling following TX_WAKE
- Raw characters (either LI or I) are transcoded to 257-bit blocks, and transmitted over the channel to the receiver
- Based on the received data, The RS-FEC RX can quickly
  - Align the incoming 257-bit blocks
  - And then align the RS-FEC codeword (parity symbols are easily discerned)
- Optimally, alignment can be achieved within two codewords ➜ less than 0.5 µs
- Clause 49 PCS bypasses scrambling for 1 µs – more than four codewords; this should be enough
  - We could extend this period if necessary

Scrambler bypass caveats and solutions

- At the RX: after codeword alignment is found, the CWM location is still unknown; can occur much later
- At the TX: the RS-FEC has to be aware of scrambler bypass to make room for CWMs (assuming it's separate from the PCS)
  - Otherwise it can't detect the idles to compensate for a CWM
  - Or may insert a CWM during scrambler bypass
- Suggested method:
  - RS-FEC in the TX direction looks for unscrambled idles/LI in the PCS stream, in parallel to the normal descramble operation.
  - Insert the first CWM at the beginning of the second codeword that contains only unscrambled data.
  - Receiver will be able to lock to both codeword and CWM location.

Timing diagrams for scrambler bypass

TX output timeline

Quiet	ALERT	idle or LI (scrambled)	idle or LI (unscrambled)	Data or LI (scrambled)

one_us_timer_done

tx_tw_timer_done (~11 µs)

one_us_timer_done (5 codewords; at least 4 fully unscrambled)

TX RS-FEC function timeline

N/A	N/A	Don't insert CWM; look for unscrambled idle/LI	Insert first CWM	Normal operation, insert CWM regularly

At the second codeword that contains only unscrambled data

RX function timeline

No energy	Energy detected	CDR lock	Codeword lock, CWM lock, detect active/sleep	Decode data or return to sleep

$<T_{WR}$ (13.7 µs)

Summary

- Using scrambler bypass seems to be a simple and sufficient solution.

- Propose using this method for EEE signaling in deep sleep.

- Fast wake suggested not to have any specified effect on the RS-FEC sublayer.

THANK YOU

Questions?