Use of RCWMs in Clause 108 RSFEC EEE

Don Cober, CoMIRA Solutions Inc.

2015.05.06

Issues with RSFEC EEE in D1P0

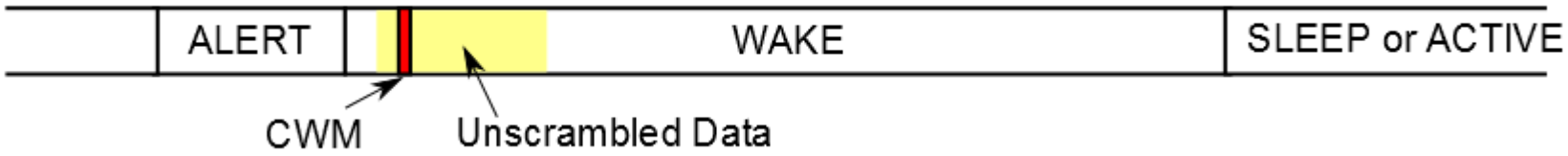
There has been a large volume of comments against the RSFEC EEE clauses in D1P0. The most common comments are citing one of two things:

1. There are concerns over DC balance and transition density when using unscrambled data that have been covered in the 4-29 AdHoc.
2. There are doubts about the mechanism used to find the CWM marker in the unscrambled data.
    1. There is only one CWM during the WAKE state that must be found on the first try.
    2. Decoding the unscrambled data will require further changes to transcoding and descrambling
    3. Detecting the transition from unscrambled to scrambled is non-trivial.

A proposed solution for these concerns is to replace the unscrambled search with a rapid codeword marker.

Proposed Solution (TX)

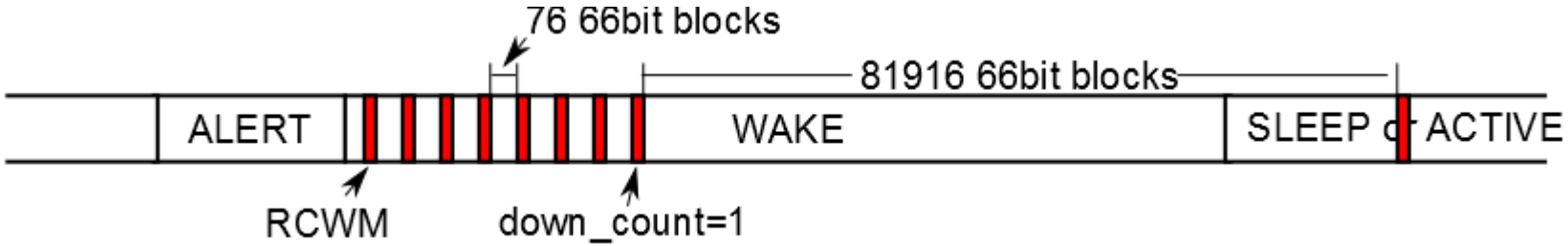
Current Specification:



On TX at ALERT->DATA transition:

1. Send unscrambled data for a period of 1.0-1.1us
2. Start sending CWM at the second full codeword.

Proposed:



On TX at ALERT->DATA transition:

1. Insert 40 RCWMs with down_count of 40…1
2. Transition back to normal CWMs by with a gap of 1023 CWs between the last RCWM and the first CWM

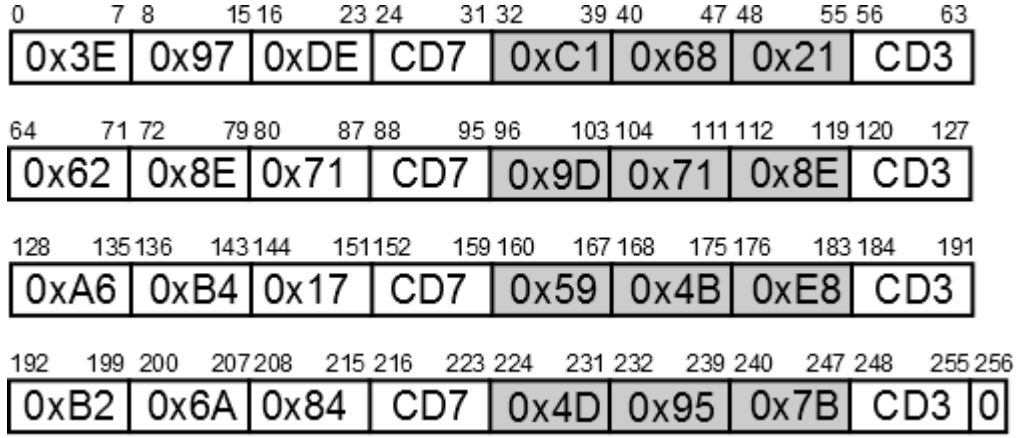
Rapid Codeword Marker Structure Details

CWM:

- Clause 108.5.2.4

RCWM:

- Re-use the clause 82 definition:
  - M0-M2 are swapped with M3-M5
  - BIP3 / BIP7 replaced with CD7/CD3
  - CD3 = down_count ^ M0
  - CD7 = ~CD3
- Distinct from CWM to allow link up during link fault conditions
- Can re-use 100G logic



Proposed Solution (RX)

Current Specification:

On RX at QUIET -> DATA transition

1. Search for codeword boundary using unscrambled data.
2. Detect first CWM position based on transition from unscrambled to scrambled data.

Proposed:

On RX at QUIET -> DATA transition

1. Search for RCWMs or CWMs
2. Wait for down count to reach 1 before expecting normal CWMs

Proposal

This change:

- Keeps the RCWMs contained to the RSFEC, so the PCS is does not need to be aware of the RSFEC during EEE.
- Is triggered by the same signal primitives as the currently defined schemed
- Reuses a well defined EEE mechanism from 802.3bj clause 82/91

Details : RCWMS

1. **Why use Clause 82 definition for RCWM?**
   - Already compatible with existing 100G logic.
   - Allows the RX to distinguish between RCWMs and CWMs incase the RX LPI FSM has become de-synchronized with the TX.

2. **Why insert RCWMS on every codeword?**
   - Increasing the number of RCWMs to the max rate will a make lock easier (the lock could search fewer simultaneous candidates)

Details : Down Count

3. What portion of the TX_WAKE period should be used for RCWMs?

- The PCS defines the TX_WAKE period to last 10.9us -11.1us
- One codeword is 0.2048 us, RCWMs can be sent on every codeword Actual TX_WAKE seen by the RSFEC may be smaller
- Once alignment lock is reached in the RSFEC, the PCS will still need to gain block lock.
  - Block lock will take a minimum 64*66 BT = 0.016 us
  - Reasonable value is: 2 blocks for each position tested * 66 positions + 64 good blocks = 2*66*66 + 64 BT = 0.340us .
- We expect some traffic at the start of TX_WAKE to be corrupted by the PMD's waking activities.

Sending 40 RCWMs on the first 40 CWs in TX_WAKE would require 8.19us. On the RX of the link partner, the PCS would still have 2.7us to achieve block lock after the RCWMs have completed.

Details: Rapid Codeword Marker Lock

4. The FEC should achieve lock by finding two consecutive valid RCWMs.

A valid RCWM will have:

1. At least 9 of the 12 nibbles of the Lane 0 marker must match
2. Sequential RCWMs must have decrementing down_count
3. down_count must not be considered valid if (CD_3 != ~CD_7)

5. How should the CD3 field be interpreted?

- down_count must be extracted by XORing CD_3 (rx_cwm[56:63]) against 0xC1(M_0) instead of XORing against rx_cwm[32:39] (in case part of rx_cwm is corrupted)
- A crisp definition is important here to maintain uniformity in all compliant products.

Details: Rapid Codeword Marker Lock

6. When should FEC_aligned be asserted?

- The PCS will be relying on the signal_OK primitive which is based on FEC_aligned. Asserting this signal will cause the LPI state machine to transition out of RX_WAKE
- It is important that we only cause this transition when we are sure that we can predict the CWM alignment, so **it is necessary to wait until down_count=1 before asserting FEC_aligned.**

7. What if RCWMS are never found?

- The search for RCWMs should expire after 11.5us to coincide with the RX_WAKE state timing out in the PCS

Error Cases

1. What happens if RCWM are not found during RX_WAKE:
   - The PCS will drop into the RX_WTF state and wait 10ms
   - The RSFEC should attempt to realign using normal CWMs
   - Alignment using normal CWMs will require ~0.629ms
   - Overall the link is not reported as down until the WTF timer expires

2. What happens if good RCWMs are seen, but then become corrupted before down_count=1 is reached.
   - The FEC has not yet asserted FEC_aligned, so the PCS is still in the RX_WAKE state
   - The PCS will drop into the RX_WTF state and wait 10ms
   - The RSFEC should attempt to realign using normal CWMs

Error Cases

3. What happens if CWM are seen when searching for RCWMs
   - The FEC should ignore normal CWMs because they would require to long (1024 cws) to validate and the RX_WAKE hold-off timer would expire
   - If the RX and TX are unsynchronized, then the FEC should wait for the WTF state to attempt to find normal CWMs

Consequences of Failure

The FEC/PCS can fail into a state that requires management intervention if the codeword boundary is aligned correctly, but the CWM is not aligned correctly.

If the CWM alignment is incorrect:

- The RSFEC will still see good/correctable codewords, since the codeword boundary is aligned.
- The RSFEC will remove the wrong blocks and allow the CWMs to pass to the PCS
- The PCS will not lose block lock because the sync headers are valid
- The PCS will not find HiBER because the sync headers are invalid
- The PCS will see 64/66 block errors but will never take the link down.

Questions?