101.3.3.1.2 LDPC decoding process within CNU (downstream)

The process of decoding FEC codewords in the {EPoC_PMD_Name} CNU receiver is illustrated in Figure 101–13.

{FEC codeword alignment needs to be tackled somewhere between the PMA and the bottom of the PCS – we had some proposals on how to find FEC codeword lock in the downstream, but I am not sure we base-lined anything with sufficient level of detail to actually put it into the draft}

Once the alignment to FEC codeword is found, the {EPoC_PMD_Name} CNU receiver aggregates the ~~total of B_Q + 1 + C_Q~~all ~~65-bit blocks received from the PMA,~~ forming the FEC payload~~, (blocks number 1 to B_Q, and bits <0> through <39> from the following 65-bit block) and~~ the FEC parity ~~(bits <40> through <64> from the 65-bit block following payload portion of the FEC codeword and followed by blocks number 1 to C_Q)~~ portions of the codeword, and the CRC40 data. Note that the ~~C_P~~ padding bits in the last parity codeword ~~(block number C_Q)~~ are locally generated within the PMA and transmitted to the PCS.

Next, ~~B_P~~ padding bits are inserted immediately after the end of the CRC40 data, and then the last 65-bit block ~~(number C_Q)~~ of the parity portion of FEC codeword is truncated, ~~removing~~ leaving the ~~last~~ parity bits in last block ~~C_{PL} bits~~, forming the input into the FEC decoder.

The FEC decoder produces the FEC payload portion of the codeword ~~with the size of F_P (in bits), where bits <F_P-B_P-1> … <F_P-1> contain~~plus any ~~-~~padding (with the binary value of "0") needed to fit into an integer number of 65-bit blocks. Next, the CRC40 is calculated over the ~~remaining 65-bit blocks 1 through B_Q~~parity portion of the codeword and then compared with the value of the CRC40 retrieved from the received FEC codeword. If both CRC40 codes match, the decoded FEC codeword is treated as error-free. Otherwise, the decoded FEC codeword is treated as errored. The behavior of the FEC decoder in the presence of CRC40 code failure depends on status of the user-configurable option to indicate an uncorrectable FEC codeword.

Finally, the FEC decoder prepends each ~~of the B_Q~~ 65-bit block~~s~~ of payload with bit <0> of the sync header containing the binary inverse of the value carried in bit <1> of the sync header, producing 66-bit blocks. This also guar~~-~~antees that properly decoded blocks meet the requirements of 49.2.4.3.

The FEC decoder in the CNU shall provide a user-configurable option to indicate an uncorrectable FEC codeword (due to an excess of symbols containing errors) to higher layers (see {insert MDIO register name and xref}). If this user-configurable option is enabled and the calculated value of CRC40 does not match the value of CRC40 retrieved from the received FEC

codeword, the FEC decoder replaces bit <0> and <1> in the sync headers in all ~~BQ~~ payload blocks with the binary value of “11”. If this user-configurable option is disabled, the FEC decoder does not make any further changes to the sync headers in ~~all BQ~~any payload blocks. Each resulting 66-bit block is then fed into the 64B/66B decoder, removing the sync header information (bit <0> and bit <1>), which is used to generate control signaling for the XGMII.

Finally, the resulting 64-bit block is then separated into two 32-bit portions, which are transmitted across the XGMII on two consecutive transfers, with the proper control signaling retrieved from the sync header information retrieved in the 64B/66B decoder.