Table 71—PHY PIB attributes

Attribute	Type	Range	Description
phyLECIMDSSSSFDEnabled	Boolean	TRUE or FALSE	A value of TRUE indicates that the SFD is present. A value of FALSE indicates that the SFD is not present. This attribute is only valid for the LECIM DSSS PHY.
phyLECIMFECTailBitingEnabled	Boolean	TRUE or FALSE	A value of TRUE indicates that tail biting is enabled. A value of FALSE indicates that it is disabled. This attribute is only valid for the LECIM DSSS PHY.
phyLECIMDSSSPSDUOVSF-SpreadingFactor	Integer	1–256	The length of the generated code in power of 2. This attribute is only valid for the LECIM DSSS PHY.
phyLECIMDSSSPSDUOVSF-CodeIndex	Integer	0, 1, ..., *N*–1	Specifies the desired code from the available set of codes. *N* is the spreading factor. This attribute is only valid for the LECIM DSSS PHY.
phyLECIMFSKPreambleLength	Integer	4–100	The number of times the preamble contains the pattern defined in 19.2.1.1. This attribute is only valid for the LECIM FSK PHY.
phyLECIMFSKPSDUPositionMod	Boolean	TRUE or FALSE	Indicates whether position-based modulation is enabled. A value of TRUE indicates that position-based modulation is enabled. A value of FALSE indicates that it is not enabled. This attribute is only valid for the LECIM FSK PHY.
phyLECIMFSKSpreading	Boolean	TRUE or FALSE	A value of TRUE indicates that spreading is enabled. A value of FALSE indicates that spreading is disabled. This attribute is only valid for the LECIM FSK PHY.
phyLECIMFSKSpreadingFactor	Enumeration	1, 2, 4, 8, 16	The spreading factor (SF) to be used when *phyLECIMFSKSpreading* is TRUE. This attribute is only valid for the LECIM FSK PHY.
phyLECIMFSKScramblePSDU	Boolean	TRUE or FALSE	A value of FALSE indicates that data whitening of the PSDU is disabled. A value of TRUE indicates that data whitening of the PSDU is enabled. This attribute is only valid for the LECIM FSK PHY.

If BPSK modulation is in use, the chip rate is equal to the modulation symbol rate. If O-QPSK modulation is in use, the chip rate is equal to twice the modulation symbol rate.

$$DataRate = ChipRate / phyLECIMDSSSPSDUSpreadingFactor \text{ kbps}$$

19.1.2.2 Reference modulator diagram

The functional block diagram in Figure 155 is provided as a reference for specifying the LECIM DSSS PHY modulation. All binary data contained in the SHR and PSDU shall be encoded using the modulation shown in Figure 155.

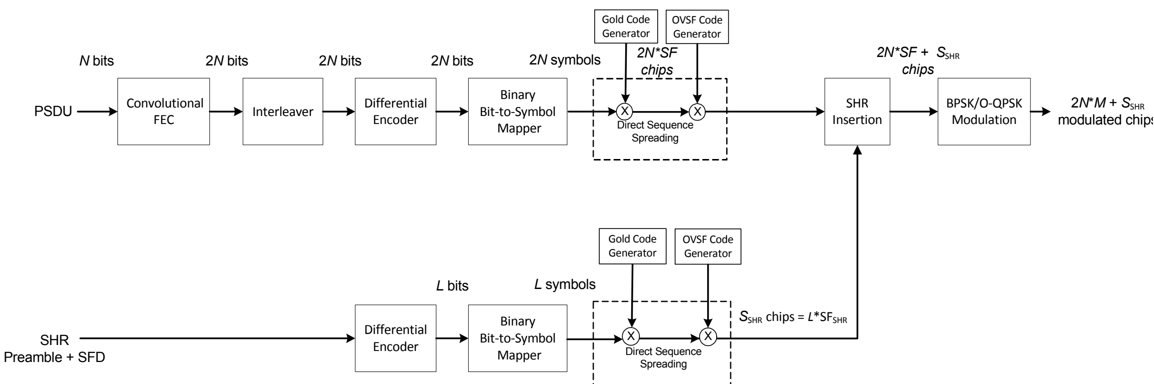

Figure 155—LECIM DSSS reference modulator diagram

19.1.2.3 Convolutional forward error correction (FEC) encoding

FEC shall employ rate 1/2 convolutional coding with constraint length $K = 7$ using the following generator polynomials:

$$G_0(x) = 1 + x^2 + x^3 + x^5 + x^6$$

$$G_1(x) = 1 + x + x^2 + x^3 + x^6$$

The encoder is shown in Figure 165, where $\oplus$ denotes modulo-2 addition.

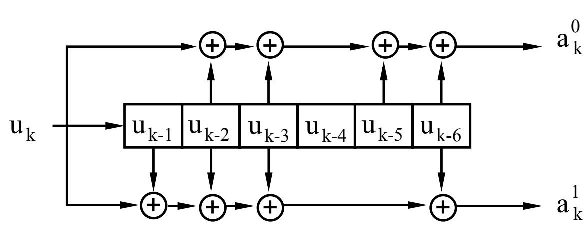

Figure 156—Convolutional encoder

Tail biting may optionally be employed. When *phyLECIMFECTailBitingEnabled* is set to TRUE, the initial encoder state at $k = 0$ shall be set to the last six bits of the PSDU.

When *phyLECIMFECTailBitingEnabled* is set to FALSE, the initial encoder state at $k = 0$ shall be set to $(u_{-1}, u_{-2}, \ldots, u_{-6}) = (0, 0, 0, 0, 0, 0)$, and the PSDU shall be extended by appending a termination sequence of eight bits, all zero.

19.1.2.4 Interleaver

The output of the convolutional coder is interleaved using a pruned bit reversal interleaving algorithm.

The text that follows contains examples of bit reverse interleavers for three fragment sizes (256, 384, 512 bits). Fragment sizes that are not powers of two (e.g., 384) employ pruning.

19.1.2.4.1 16 octet (256 bit) fragment size

If the input sequence into the interleaver is represented by

$$[S_0\ S_1 \ldots S_{255}]$$

Then the output sequence of the interleaver can be described as

$$[S_0\ S_N\ \ldots\ S_{255}]$$

The value N for the M^{th} output is determined as the bit-reversal of the value M.

Representing the value M as a binary representation

$$M = [m_7\ m_6 \ldots m_0]$$

where m_i are the binary digits, then

$$N = [m_0\ m_1 \ldots m_7]$$

where M is incremented sequentially from 0 to 255.

For example if $M = 1 = 0000\ 0001_2$, then $N = 1000\ 0000_2 = 128$

The sequence of N is shown in Table 189.

Table 189—Sequence of *N* for 256 bit fragment size

	Bit: 0	1	2	3	4	5	6	7
Octet: 0	000	128	064	192	032	160	096	224
1	016	144	080	208	048	176	112	240
2	008	136	072	200	040	168	104	232
3	024	152	088	216	056	184	120	248
4	004	132	068	196	036	164	100	228