143.3.2 Envelope Header format

Each envelope initiated by the MCRS begins with a 72-bit envelope header. The envelope header includes a Start Control Code, an EnvType flag bit, a 22-bit Envelope Length field, an Envelope Position Alignment Marker (EPAM) field, two bits (E and ~~S~~K) reserved for encryption purposes, an LLID field, and an 8-bit cyclic redundancy check (CRC8). ~~The envelope length represents the number of EQ in the envelope.~~ When the xMII is 36-bits wide the transmission envelope header, as illustrated in Figure 143–11, includes two successive transfers over the xMII. Each 36-bit transfer includes four control bits followed by 32 information bits.

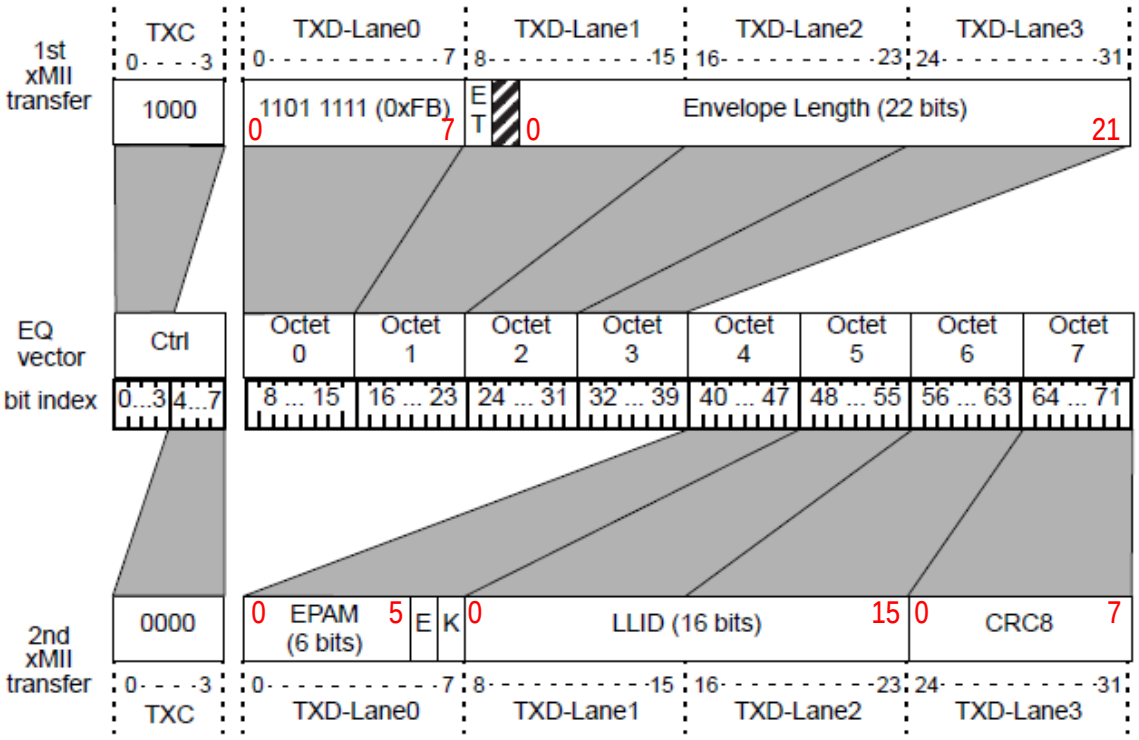

Figure 143-11 – Mapping of Envelope Header fields into two xMII transfers

Octets within each envelope header field are transmitted from least significant to most significant. Bits within each octet are transmitted from LSB to MSB. An EQ contains eight octets of information so the length of the envelope header ~~is~~ equals one EQ. ~~The EPAM is used by the receiving MCRS to remove any timing skew that may have occurred during the transmission of the envelope from the source MCRS to the destination MCRS. The LLID field is set to the value of the LLID of the MAC associated with the data in the envelope. The CRC8 field is used for error detection within the header. There is one reserved bit (EQ bit 17) and it is set to zero at the transmitter and its value is ignored at the received except for the purposes of calculating the CRC8.~~ The envelope header shall use the format shown in Table 143–3.

Table 143–3—Envelope Header EQ

EQ Bits	Value	Description
0-7	0x80	Control bits corresponding to TXC<3:0> in two successive MII transfers
8-15	0xFB	Start Control Code
16	0 for ECH 1 for ESH	EnvType flag
17	0	reserved
18-39	varies	Length of envelope (in EQ)
40-45	varies	Envelope Position Alignment Marker (Number of bits matches the size of wRow)
46-47	0x0	reserved
48-63	varies	LLID
64-71	varies	CRC8

The envelope header is identified by Start Control Code (Block type 0xFB, see Table 142-1).

The envelope start header (ESH) has the EnvType flag set to one whereas the envelope continuation header (ECH) has the EnvType flag set to zero. The ~~envelope start header~~ESH is used to indicate the beginning of a transmission from a specific LLID.

Following the EnvType flag, there is one reserved bit (EQ<17>) and it is set to zero at the transmitter and its value is ignored at the received except for the purposes of calculating the CRC8.

The Envelope Length field represents the number of EQs in the envelope, including ~~in the envelope header includes~~ the envelope header itself (1 EQ). ~~The envelope continuation header replaces any preambles encountered in the transmission and, in this case, the Envelope Length field includes the envelope continuation header.~~

The EPAM is used by the receiving MCRS to remove any timing skew that may have occurred during the transmission of the envelope from the source MCRS to the destination MCRS.

The LLID field is set to the value of the LLID ~~of the~~ (MAC instance) associated with the data in the envelope.

The CRC8 field is used for error detection within the header. ~~There is one reserved bit (EQ bit 17) and it is set to zero at the transmitter and its value is ignored at the received except for the purposes of calculating the CRC8.~~ CRC8 uses the same generating polynomial as described in 65.1.3.2.3. The CRC8 checksum is calculated over EQ bits 8 through 63. The envelope header bits are processed by the CRC8 calculating function in the same order they are transmitted, i.e., for each field the bits are processed starting with the LSB and ending with the MSB.

143.3.2.1 CRC8 calculation test sequences (informative)

The following test sequences show several example envelope header field values and the resulting CRC8 value computed by a compliant implementation.

Table 143-4 -- CRC8 computation example #1

Envelope Header Fields	**Block Type Field**	**ET**	**R**	**EnvLength**	**EPAM**	**E**	**K**	**LLID**
	0xFB	1	0	64	15	0	0	0xAB-CD
Envelope Header with CRC8 (hex)	**E5**-AB-CD-0F-00-01-01-FB (→ transmitted LSB first)							
Envelope Header with CRC8 (bin)	(last bit) (first bit) **1110-0101**-1010-1011-1100-1101-0000-1111-0000-0000-0000-0001-0000-0001-1111-1011							

Table 143-5 -- CRC8 computation example #2

Envelope Header Fields	**Block Type Field**	**ET**	**R**	**EnvLength**	**EPAM**	**E**	**K**	**LLID**
	0xFB	0	0	960	5	1	0	0x00-01
Envelope Header with CRC8 (hex)	23-00-01-45-00-0F-00-FB (→ transmitted LSB first)							
Envelope Header with CRC8 (bin)	(last bit) (first bit) **0010-0011**-0000-0000-0000-0001-0100-0101-0000-0000-0000-1111-0000-0000-1111-1011							

Table 143-6 -- CRC8 computation example #3

Envelope Header Fields	**Block Type Field**	**ET**	**R**	**EnvLength**	**EPAM**	**E**	**K**	**LLID**
	0xFB	1	0	10,000	1	1	1	0x12-34
Envelope Header with CRC8 (hex)	**B6**-12-34-C1-00-9C-41-FB (→ transmitted LSB first)							
Envelope Header with CRC8 (bin)	(last bit) (first bit) **1011-0110**-0001-0010-0011-0100-1100-0001-0000-0000-1001-1100-0100-0001-1111-1011							