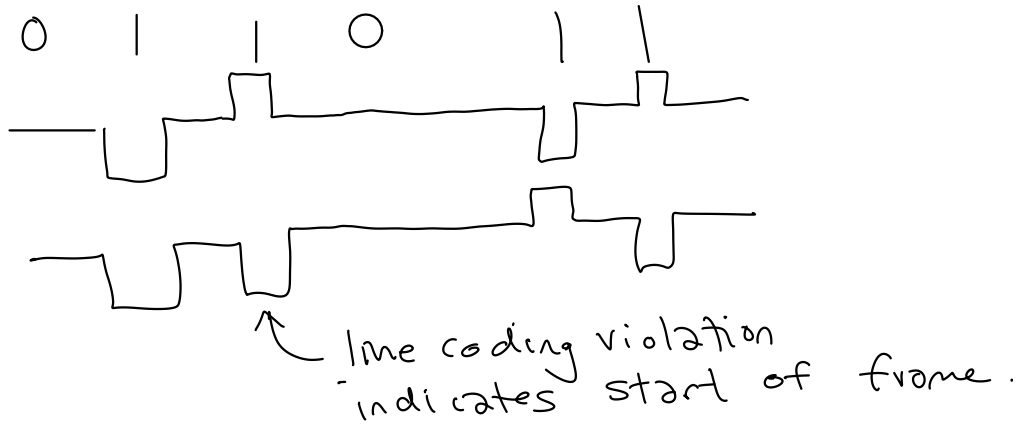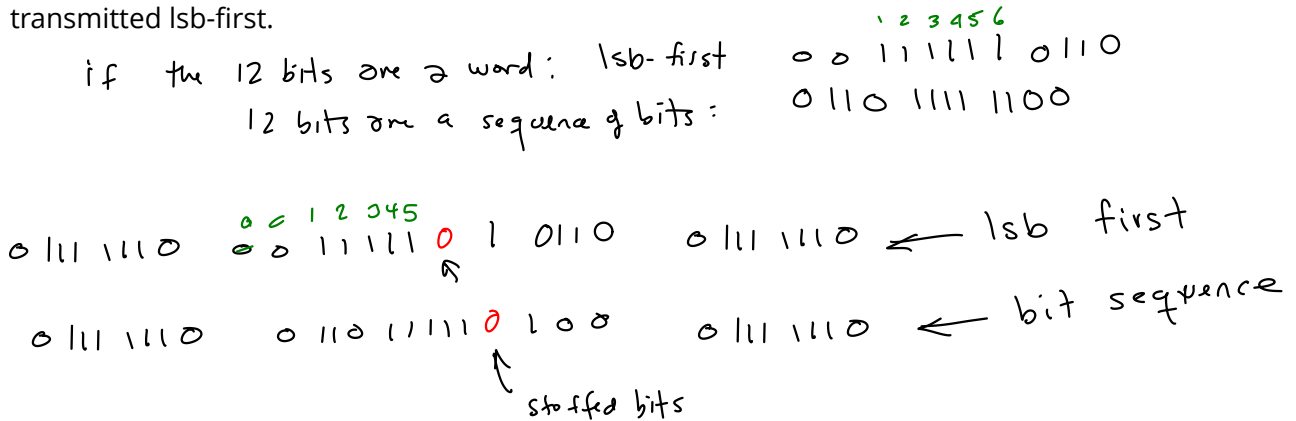# Framing

Exercise 1: Draw the waveform for an AMI-RZ encoded sequence of bits '011011' assuming the previous mark was transmitted as a positive pulse. Draw the waveform assuming the second '1' indicates the start of a frame.
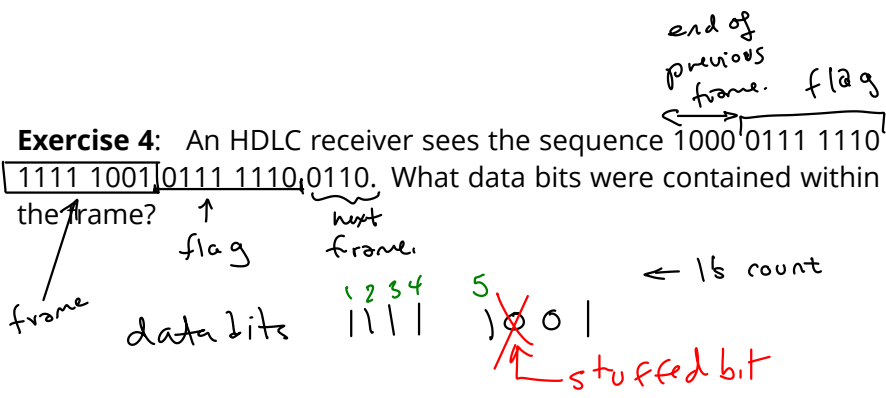
0 | | 0 | |

line coding violation indicates start of frame.

Exercise 2: You receive the sequence of bits 01110110 and are told that bit stuffing was used to limit runs of 1 to three or fewer. What was the original data sequence?

0   1 2 3   stuffed bit   1 2   0   ← count of 1's
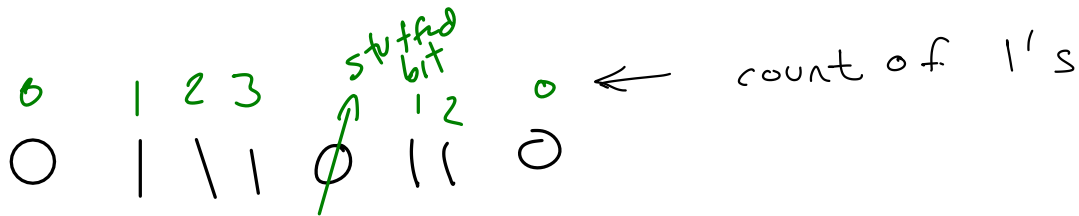
O   1 \ 1   0   1 1   0

Exercise 3: Write out the complete sequence of 1's and 0's required to transmit the 12 bits 0110 1111 1100. Include the start and end flag sequences and any necessary bit stuffing. Assume bits are transmitted lsb-first.

if the 12 bits are a word: lsb-first   1 2 3 4 5 6   o o 1 1 1 1 1 1 o 1 1 o

12 bits are a sequence of bits: O 1 1 O 1 1 1 1 1 1 0 0

0 1 1 1 1 1 1 0   0 o 1 2 3 4 5   o o 1 1 1 1 1 0 1 0 1 1 0   0 1 1 1 1 1 1 0   ← lsb first

0 1 1 1 1 1 1 0   0 1 1 0 1 1 1 1 1 0 1 0 0   0 1 1 1 1 1 1 0   ← bit sequence

stuffed bits

Exercise 4: An HDLC receiver sees the sequence 1000 0111 1110 [1111 1001] 0111 1110 0110. What data bits were contained within the frame?

end of previous frame. *flag*

↑ *flag*

frame

next frame

frame

← 1's count

data bits 1111 1001

1 2 3 4 5

— stuffed bit

Exercise 5: In this case, by how much does the use of escape characters slow down a link transmitting a continuous stream of escape characters?
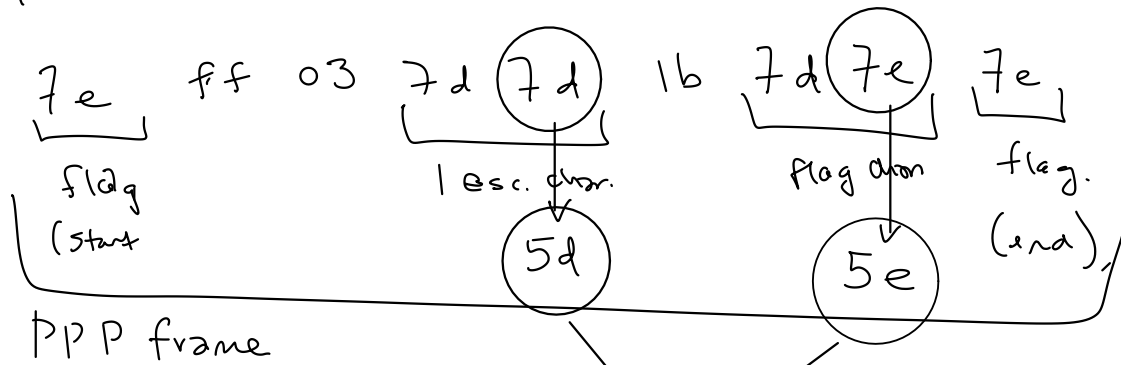
by ½ because each escape is transmitted as 2 bytes.

Exercise 6: What sequence of bytes would be sent to transmit a PPP-encapsulated frame containing the bytes `0xff` `0x03` `0x7d` `0x1b` `0x7e`?

7e = flag character
7d = esc. char.
1b = Ascii ESC characters (not special for PPP)

7e   ff   03   7d (7d)   1b   7d (7e)   7e

flag (start

1 esc. char. → 5d

Flag char → 5e

flag. (end)

PPP frame

XOR escaped byte with 0x20

0111 1101   7d
+ 0010 0000   20
0101 1101
5   d

Exercise 7: Preambles such as this allow multiple transmission formats to be used in a backwards-compatible way. What might be some disadvantages of using such a preamble? *Hint: to be decoded by old ("legacy") devices the preamble must be transmitted at the lowest possible data rate. This can be 100 times slower than the fastest devices.*

overhead of backwards compatibility.

Exercise 8: A physical layer transmits 3 bits per symbol. A frame of 128 bytes is being transmitted. How many padding bits will have to be added to the frame?

$3, 6, 9, \cdots$ bits

$128 \cdot 8 = 1024$ bits

$$\frac{1024 \text{ bits}}{3 \text{ bit/symbol}} = 341\tfrac{1}{3} \text{ symbols}$$

round up to 342 symbols

need $\left(342 - 341\tfrac{1}{3}\right) \times 3$ padding bits

or 342×3 bits $- 1024 = 2$ padding bits needed.