Framing

Sh2

Q.1

a) Frame → Count = 4+1: 000 00 101, A: 0100 0111, B: 111 00011, ESC: 111 00000, Flag: 0111 1110

(000 00 101 = 1; the rest = 4)

* overhead = 1 Byte
* if there's an error in the char count, it loses sync until reset

b) Frame → flag, MSG [A, B, \ESC, \flag], flag

↳ 0111 1110 (F), 0100 0111 (A), 111 00011 (B), 111 000 00 (\), 111 000 00 (ESC), 111 000000 (\), 0111 1110 (F), 0111 1110 (F)

* overhead$)_{max}$ = 2*8 + 2*Data → if all char are ESC or flag
* if the start flag was corrupted, we will lose this frame but will recover after sync [Next frame]. Start + Data + End + Start + --- (sync)

c) Frame → flag + MSG (Append '0' after each 5 '1') + flag

↳ 0111 1110 (F), 0100 0111 (A), 11 0 1000 11 (B), 111 0 000 00 (ESC), 0111 110 1 10 (f), 0111 1110 (f)

* overhead = 2*8 (→ start + End) + 1 * no of case '1' = 16 + 3 = 19 bits
* Recovery ⇒ same as byte stuffing

Q.2 No, we need the start flag to know if the sender started sending data or we are just reading Noise and the sender is silent.

Q.3 Yes, → if we lost a bit from the start/End flag
↳ Miss the frame
↳ if '1' was inserted after 5 '0'
↳ erroneous flag!

Error Detection & Correction

Even Parity → XOR → odd=1, even=0

Q:1 MSG ⇒ 1101 0011 0011 0101 } 16 bit

$m=16,\ m+r+1 \le 2^r \Rightarrow 17+r+1 \le 2^r,\ r=5 \Rightarrow n=m+r=21$

n →	P_1	P_2	m_3	P_4	m_5	m_6	m_7	P_8	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	P_{16}	H_{17}	H_{18}
Send →			1		1	0	1		0	0	1	1	0	0	1		1	0
Rec →																		

n →	H_{19}	H_{20}	H_{21}
Send →	1	0	1
Rec →			

$P_1 = XOR(H_3, H_5, H_7, H_9, H_{11}, H_{13}, H_{15}, H_{17}, H_{19}, H_{21}) = 0$

$P_2 = XOR(H_3, H_6, H_7, H_{10}, H_{11}, H_{14}, H_{15}, H_{18}, H_{19}) = 1$

$P_4 = XOR(m_5, m_6, m_7, m_{12}, m_{13}, m_{14}, m_{15}, m_{20}, m_{21}) = 1$

$P_8 = XOR(m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}) = 1$

$P_{16} = XOR(m_{17}, m_{18}, m_{19}, m_{20}, m_{21}) = 1$

Bit 20 was modified (0 → 1) $\xrightarrow[\text{Parity}]{\text{update}}$ $P_{16}=0,\ P_4=0$

	P_{16}	P_8	P_4	P_2	P_1
old Parity:	1	1	1	1	0
New —— :	0	1	0	1	0 ⊕
Sum :	1	0	1	0	0

Dec → 20, we managed to detect that the error is in bit 20

Q:2 we Can't do Hamming Code ; bcz Hamming code depends on the MSG size to Calc (r), which is now Corrupted error in the 1st byte ↵

Q:3

Checksum = 1's Comp($\sum_{i}^{N} W_i$) → Equall chunks

= 1's Comp(0001 + F203 + F4F5 + F6F7)

= 1's Comp(2DDF0) = 1's Comp(DDF2) = 220D

(wrap)

Q:4

$$x^3+1 \overline{) x^7 + x^5 + 1} \quad = x^4 + x^2 + x$$

$x^7 + x^4$

$x^5 + x^4 + 1$

$x^5 + x^2$

$x^4 + x^2 + 1$

$x^4 + x$

$x^2 + x + 1$ → remainder [111]

Q:5

$$x^3+1 \overline{) x^7 + x^4 + x^3 + x^2 + 1} \quad = x^4 + 1$$

1001 1101 (bit positions 7 6 5 4 3 2 1 0)

$x^7 + x^4$

$x^3 + x^2 + 1$

$x^3 + 1$

x^2 → remainder [100]

add 3 to the pw

MSG = M + r = 100 1110 1 100

$= x^{10} + x^7 + x^6 + x^5 + x^3 + x^2$

* 3rd bit was inverted [Add x^8] → if we divided again, $r \neq 0$ i.e. error!

Q:6 So, we Can start our Computations while still receiving and compare the results at the End!