1.)

Bits	1	0	0	1	1	1	1	1	0	0	0	1
NRZ												
				-								_
Bits	1	0	0	1	1	1	1	1	0	0	0	1
			1									
NRZI												
Bits	1	0	0	1	1	1	1	1	0	0	0	1
Man.												

2.)

5B with Stronger Restrictions = Total 5B – Leading Zero Violators – Trailing Zero Violators + Duplicates.

Total  $5B = 2^5 = 32$  Possible Encodings

Leading Zero Violators = 00 {Arbitary 3 Bits} = 8 Violators

Trailing Zero Violators = {Arbitrary 3 Bits} 00 = 8 Violators

Duplicates = 00 {Arbitary Bit} 00 = 2 Duplicates

5B with Stronger Restrictions = 32 - 8 - 8 + 2 = 18 Valid Encodings.

Total  $4B = 2^4 = 16$  Permutations

Since 18 (5B Encodings) > 16 (4B Encodings), there should be enough valid encodings to map, even with the stronger restrictions.

(NOTE: This answer does not account for the control signals, which may invalidate this answer)

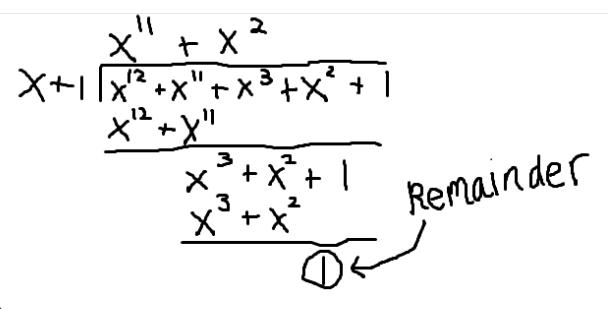
## 3.) 1101 0111 1100 1011 1110 1010 1111 1011 0

4.) For every DLE and ETX in the data body, add another DLE prior to it so that the actual data doesn't get interpreted incorrectly. Therefore, for this example, the bytes prior to the CRC is.

## **DLE DLE DLE ETX ETX CRC**

Where the highlighted DLEs are the bytes stuffed into the data.

5.) In this problem, C(x) does not contain the factor (x + 1). As such, it cannot check messages with an odd number of errors.



6.)

Message = **0011 1010 1011 0000 0000** (Take original message times  $2^8$ )

$$E(x) = (x^12 + ... + 1) = (x^8 + x^2 + x + 1) * (x^4 + ... + 1) =>$$

(Ignoring the arbitrary ellipses, we get the following E(x) polynomial that would cause a false positive)

$$(x^12 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1)$$