# Error detection: Cyclic Redundancy Check

Liceo G.B. Brocchi - Bassano del Grappa (VI) Liceo Scientifico - opzione scienze applicate Giovanni Mazzocchin

$$x-3 \quad x^3-2x^2+0x-4$$

$$\frac{x^2}{x-3} = \frac{x^3-2x^2+0x-4}{x^3-2x^2+0x-4}$$

$$\begin{array}{r}
x^{2} \\
x - 3 \\
x^{3} - 2x^{2} + 0x - 4 \\
x^{3} - 3x^{2} \\
x^{2} + 0x
\end{array}$$

$$\begin{array}{r}
x^{2} + x \\
x - 3 & x^{3} - 2x^{2} + 0x - 4 \\
x^{3} - 3x^{2} & \\
x^{2} + 0x
\end{array}$$

$$\begin{array}{r}
x^{2} + x \\
x - 3 \\
x^{3} - 2x^{2} + 0x - 4 \\
x^{3} - 3x^{2} \\
\hline
x^{2} + 0x \\
x^{2} - 3x \\
\hline
3x - 4
\end{array}$$

$$\begin{array}{r}
x^{2} + x + 3 \\
x - 3 \\
x^{3} - 2x^{2} + 0x - 4 \\
x^{3} - 3x^{2} \\
\hline
x^{2} + 0x \\
x^{2} - 3x \\
\hline
3x - 4 \\
3x - 9
\end{array}$$

$$\begin{array}{r}
x^{2} + x + 3 \\
x - 3 \overline{\smash)2222440x - 4} \\
x^{3} - 3x^{2} \\
\hline
x^{2} + 0x \\
x^{2} - 3x \\
\hline
3x - 4 \\
3x - 9 \\
\hline
5
\end{array}$$

# Cyclic Redundancy Check

- Il **CRC** è un esempio di polynomial code: un frame di k bit viene interpretato come polinomio di grado k-1
- Inventato da <u>W. Wesley Peterson</u>, viene utilizzato per l'error detection
- Esempio:

```
la stringa di bit 1 0 0 1 1 0 1 viene vista come: 1x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0
```

- Il CRC si basa sull'aritmetica in modulo 2: addizione e sottrazione sono la stessa operazione, ossia lo XOR. Quindi niente riporti o prestiti (no carries or borrows)
- Regola di divisibilità utilizzata: b divide a se a ha almeno tanti bit quanti b

## Cyclic Redundancy Check - algoritmo

- Il mittente e il destinatario si accordano su un polinomio generatore G(x), di grado inferiore rispetto al polinomio F(x) che rappresenta il frame
- Sia r il grado di G(x). Il mittente aggiunge in coda a F(x) r bit a 0. Chiamiamo il polinomio risultante P(x)
- Il mittente effettua la divisione in modulo 2 tra P(x) e G(x)
- Il mittente considera il resto della divisione (R(x)), e calcola:

$$T(x) = P(x) - R(x)$$

- Il mittente trasmette T(x)
- Il ricevitore calcola T(x) / G(x). Se il resto è diverso da 0, significa che il frame è corrotto

**Frame -** F(x): 1 1 0 1 0 1 1 1 1

**Generatore -** G(x): 1 0 0 1 1

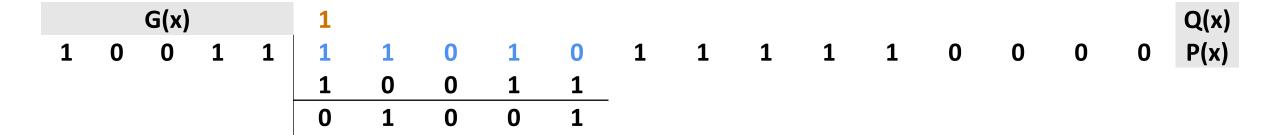
Il grado del generatore è 4. Quindi aggiungiamo 4 bit a 0 al frame, ottenendo:

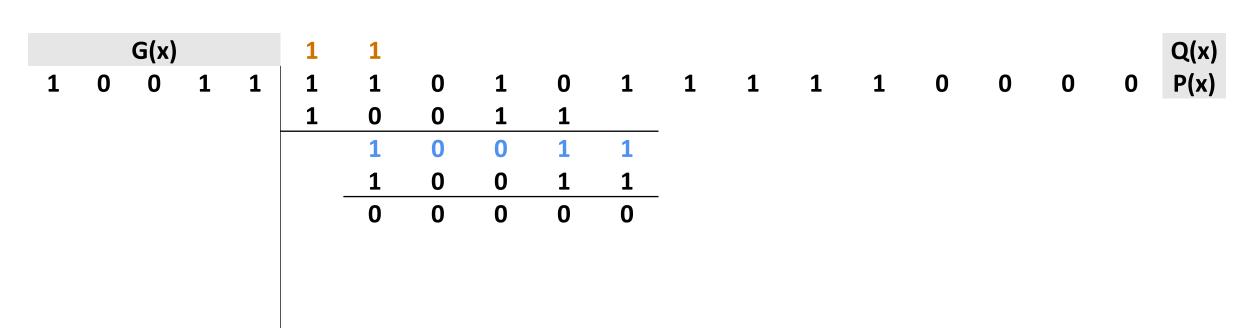
P(x): 1 1 0 1 0 1 1 1 1 1 0 0 0 0

i bit sottolineati sono quelli aggiunti

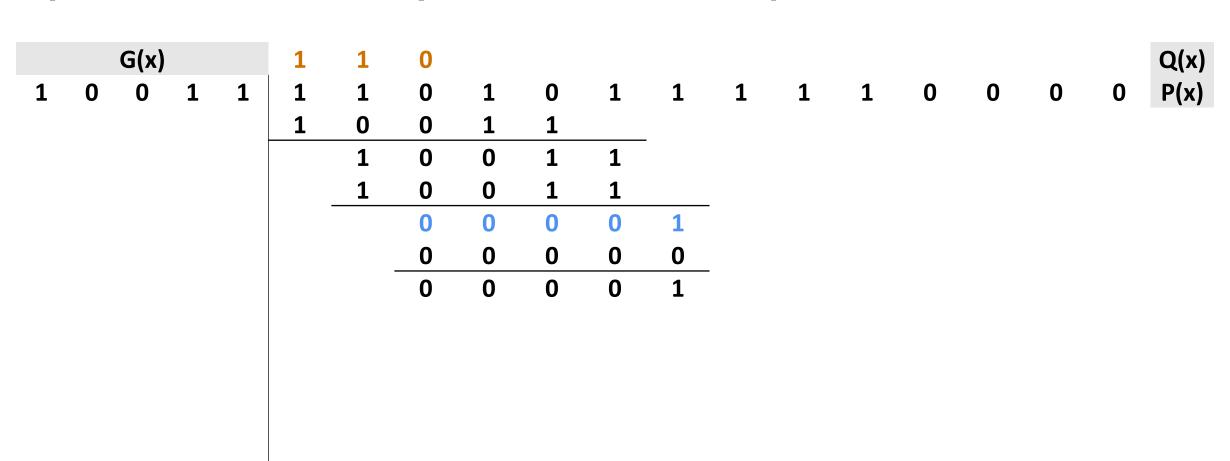


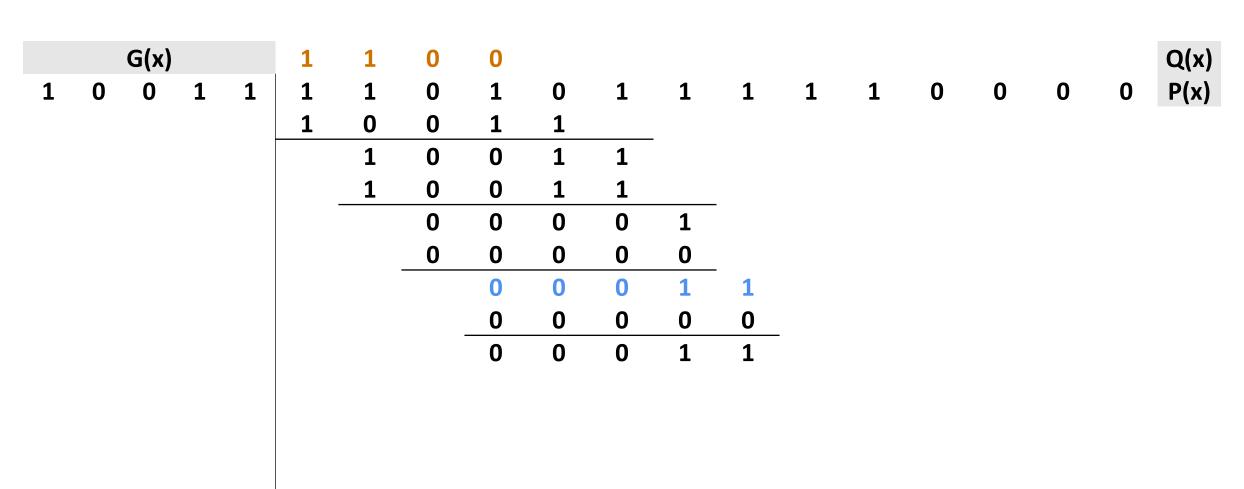
08/10/2023 Cyclic Redundancy Check

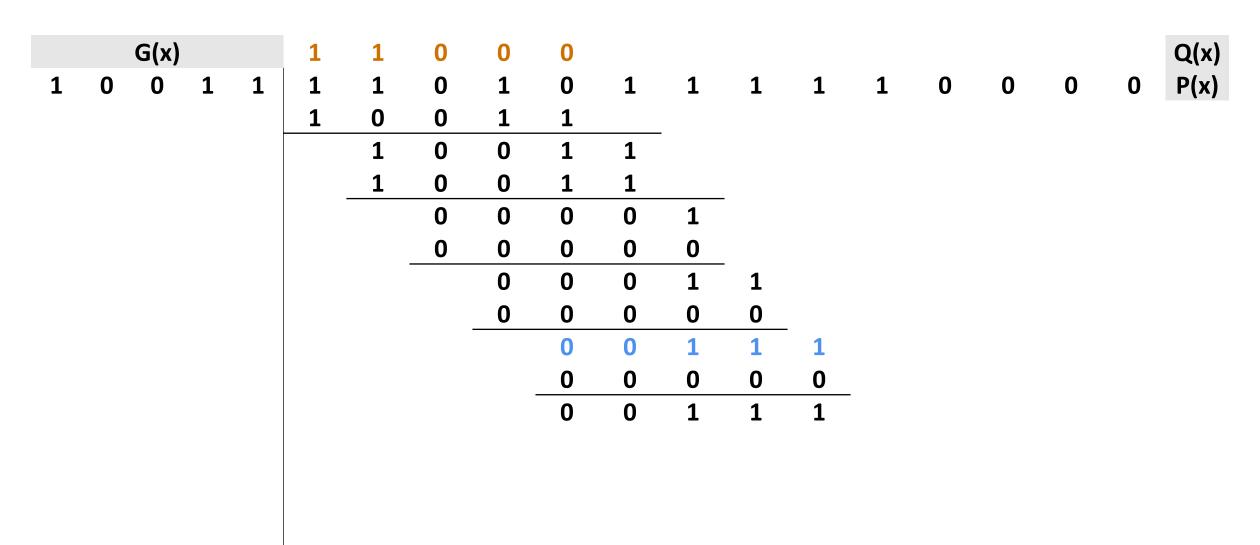


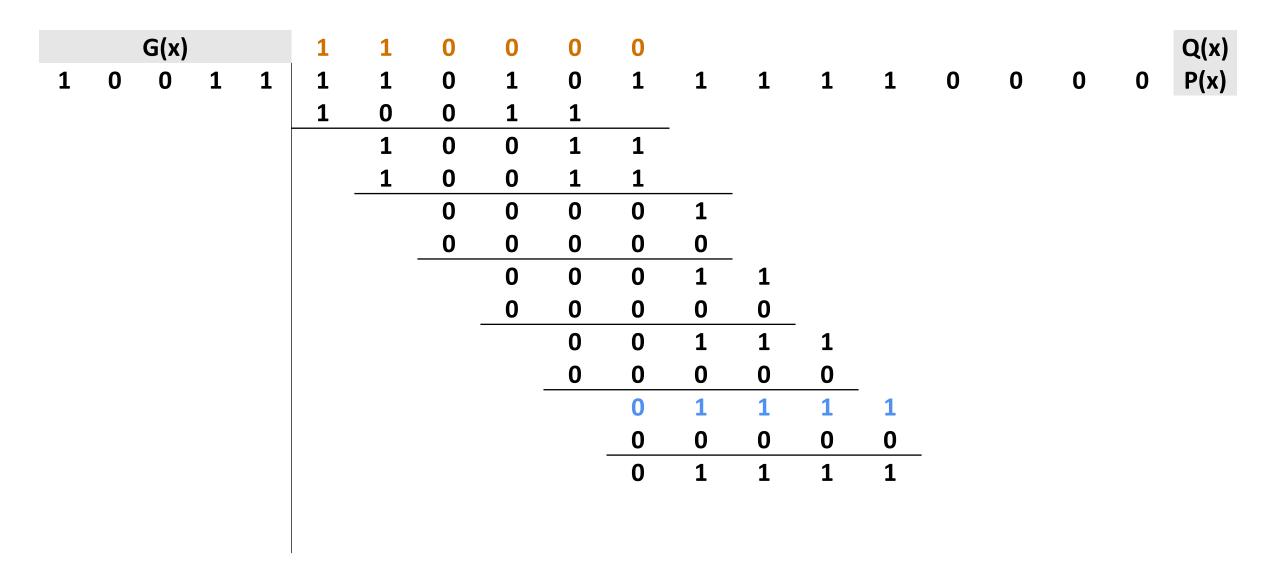


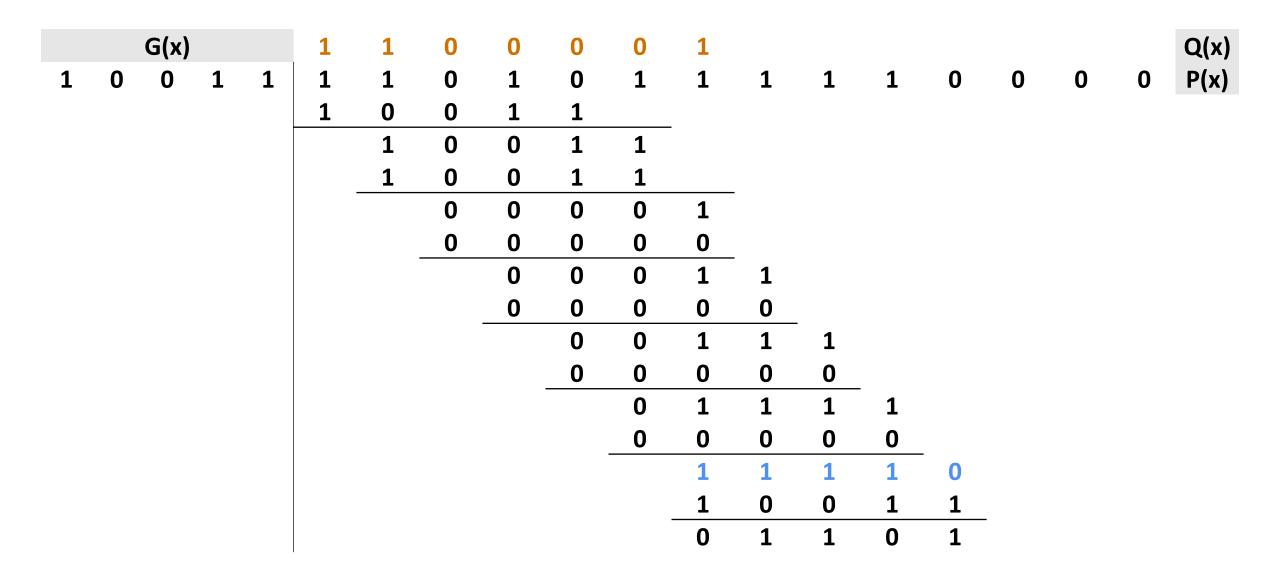
15

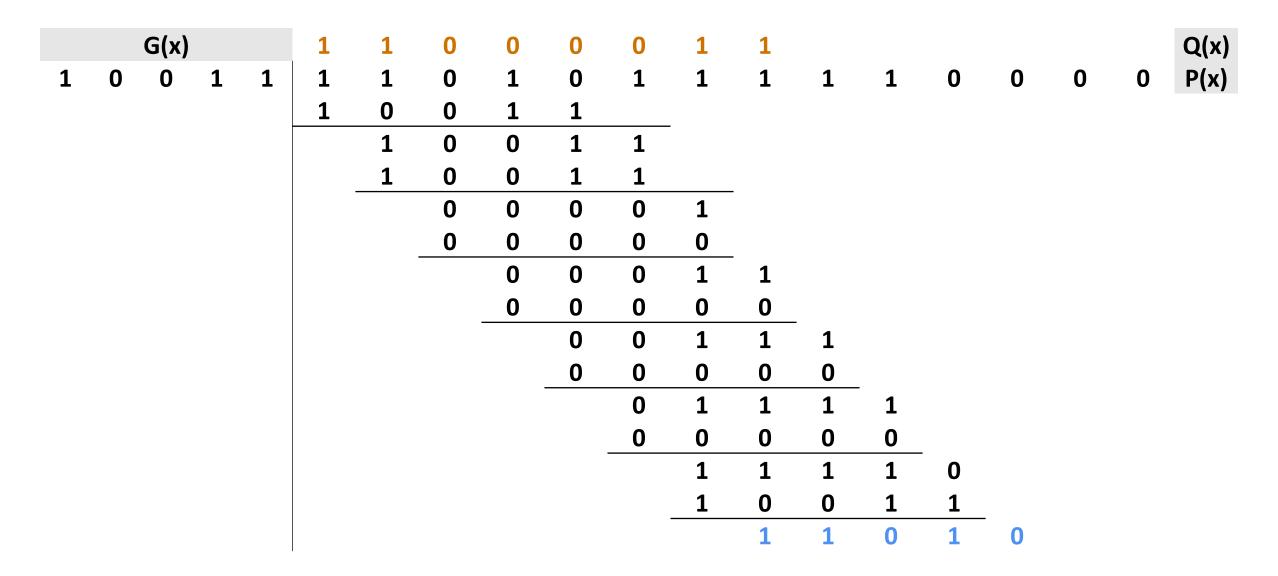


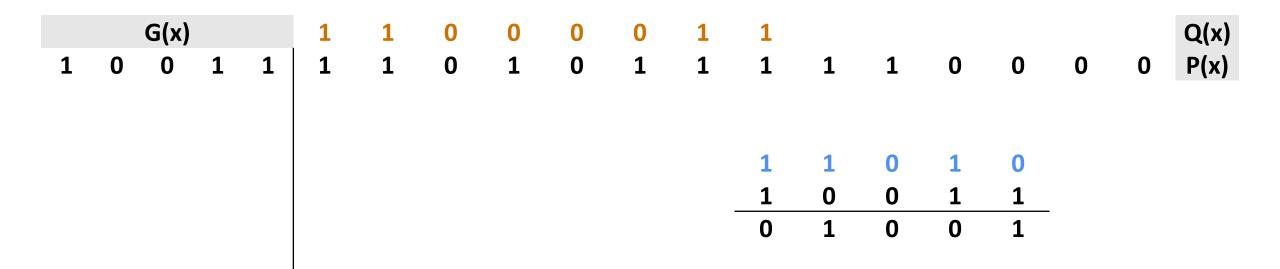


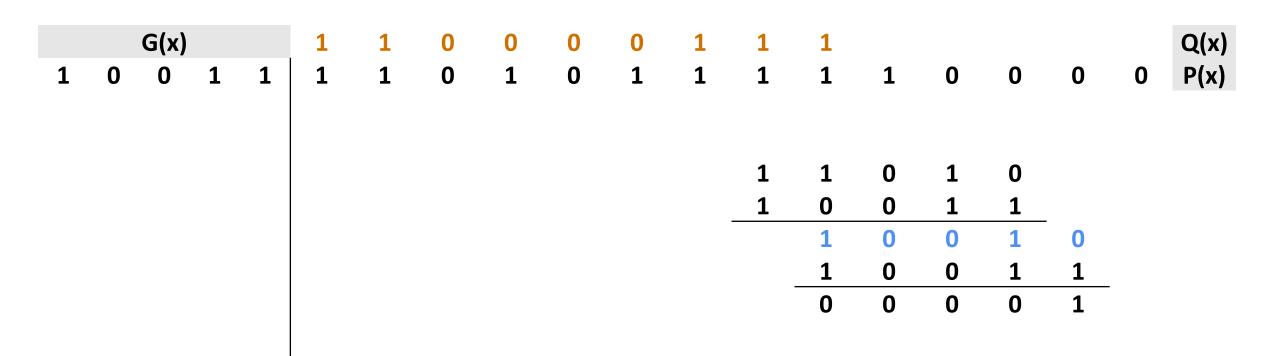


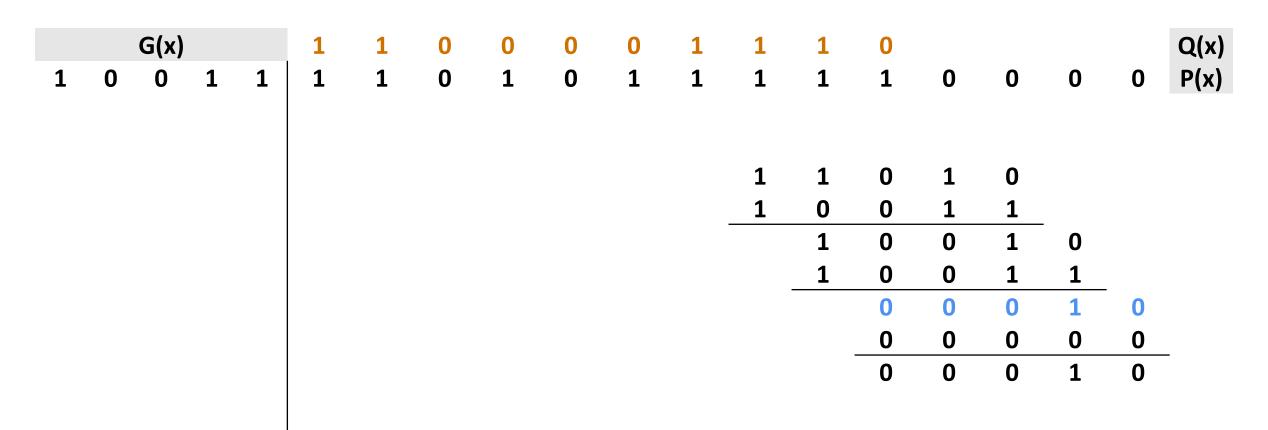




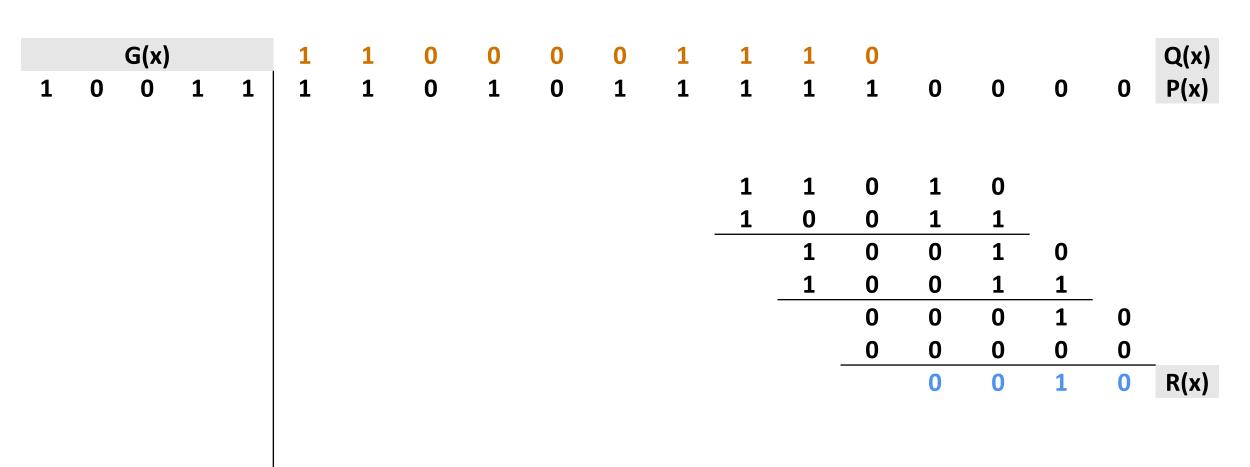






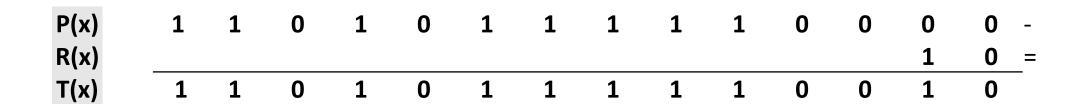


08/10/2023 Cyclic Redundancy Check 24

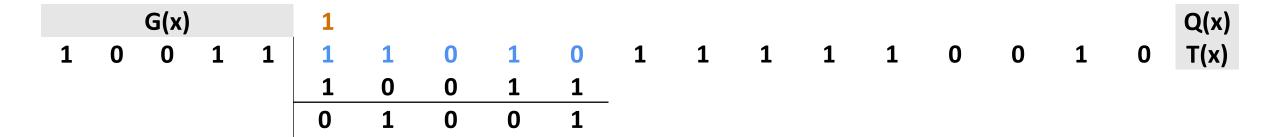


08/10/2023 Cyclic Redundancy Check

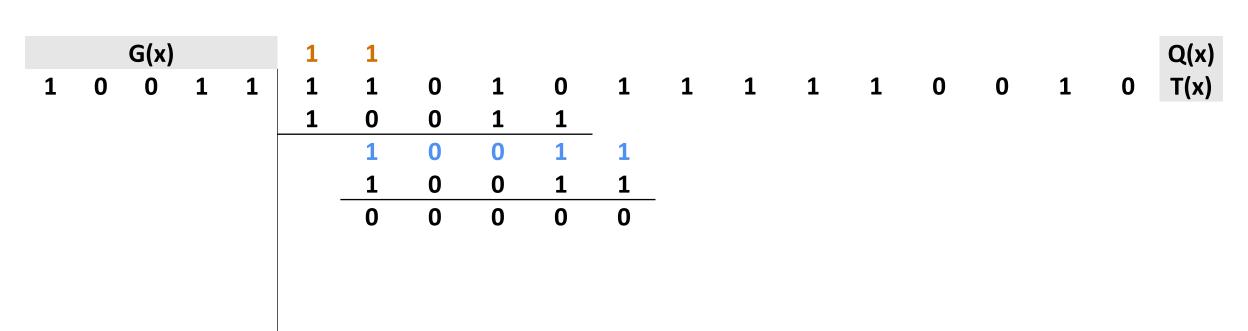
25



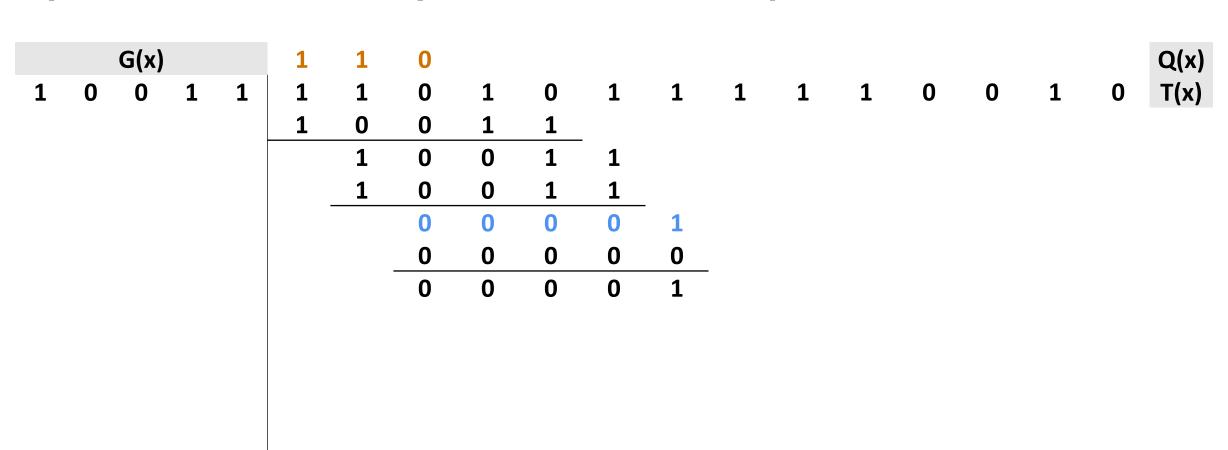
- il mittente trasmette T(x)
- il ricevitore calcola T(x) / G(x). Se il resto è diverso da 0, significa che il frame è corrotto
- calcoliamo T(x) / G(x)

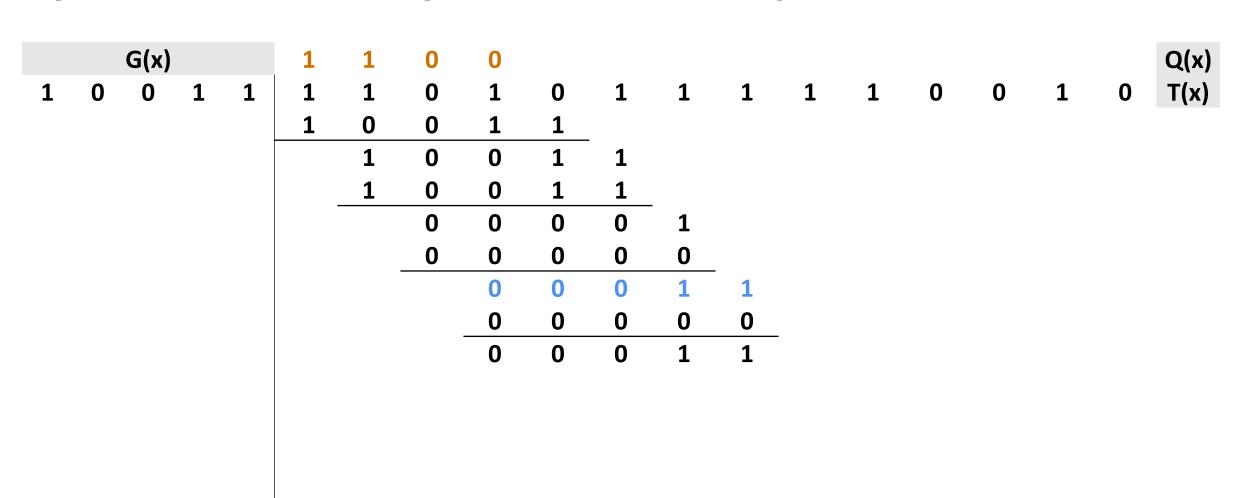


08/10/2023 Cyclic Redundancy Check 27

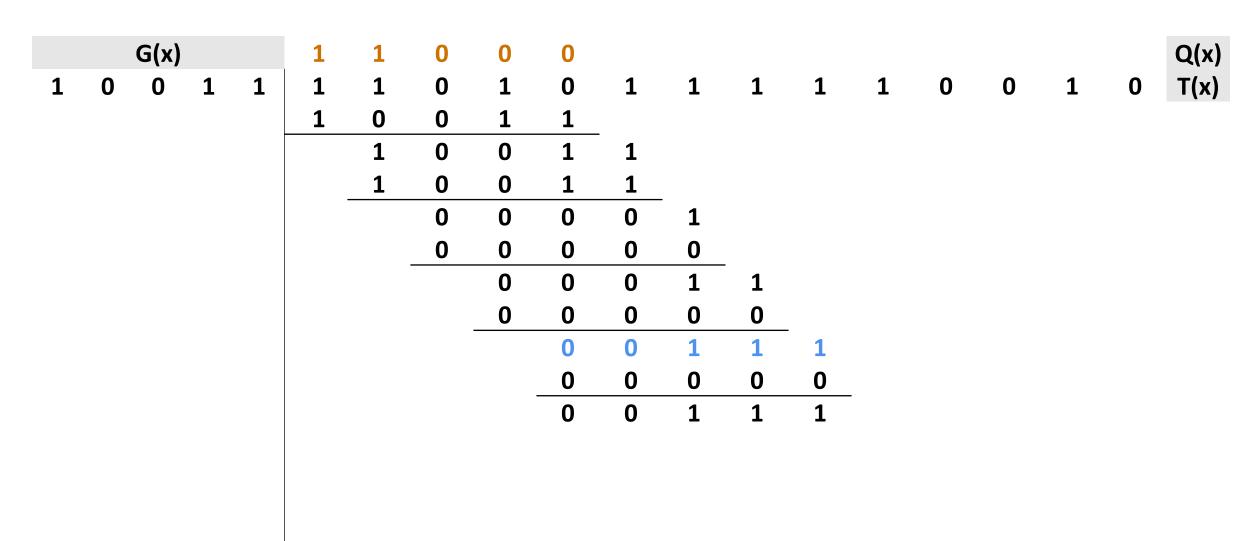


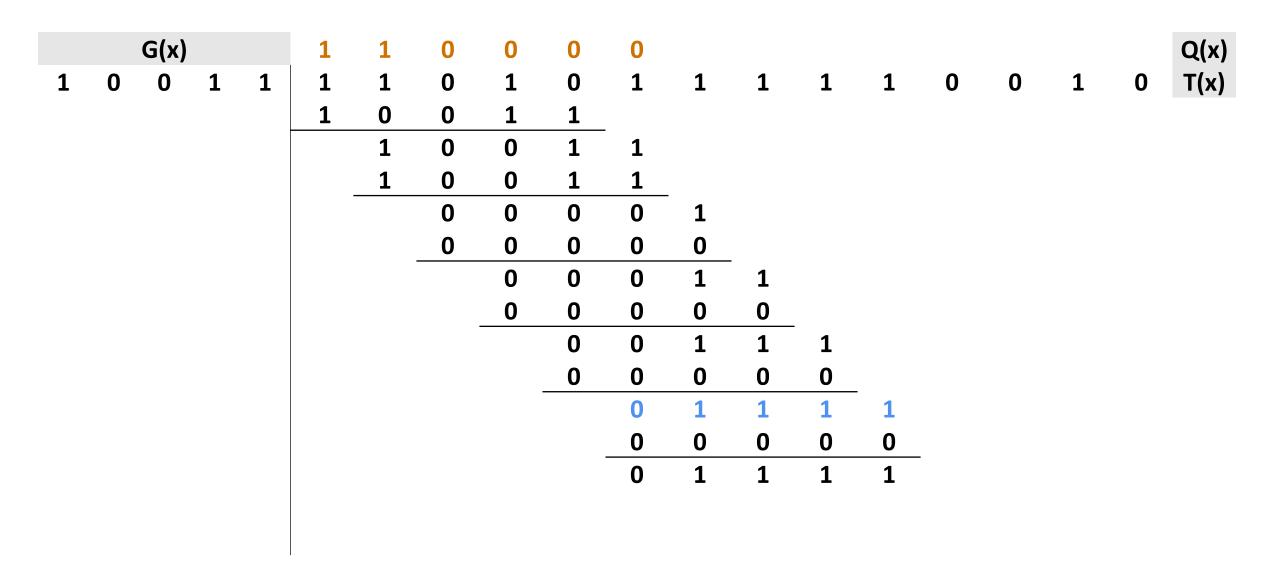
Cyclic Redundancy Check 28

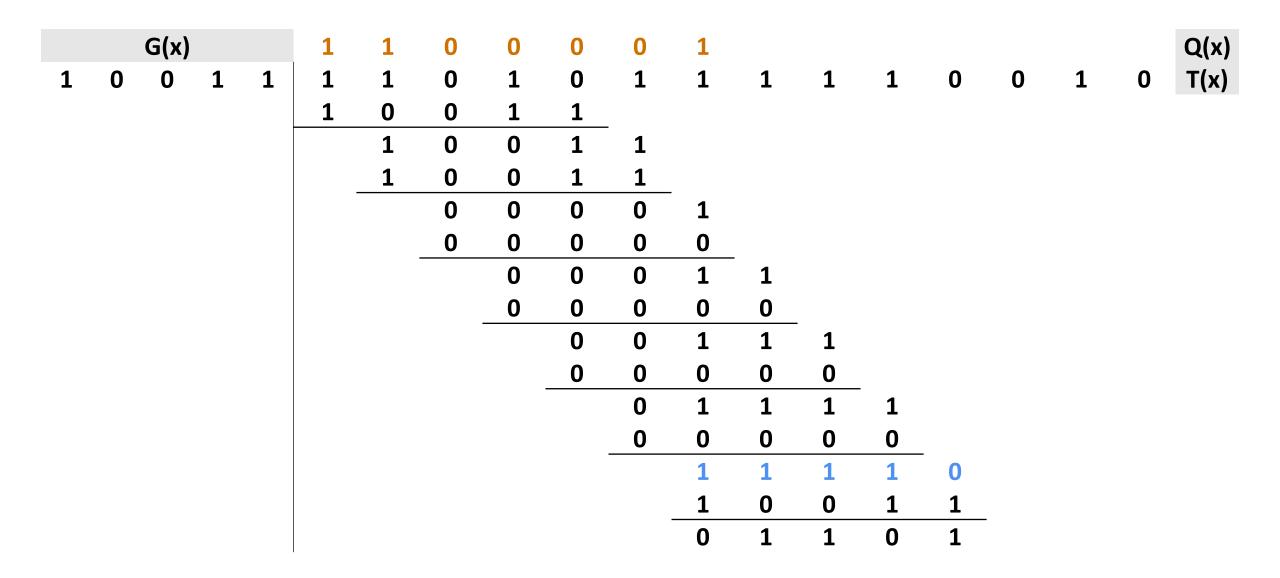


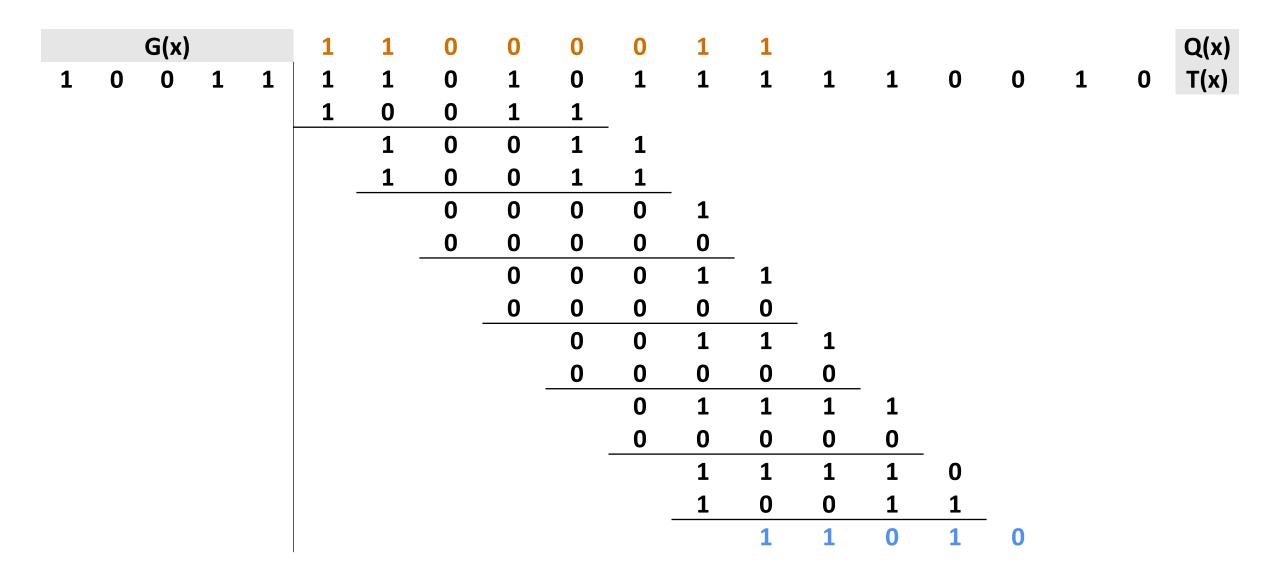


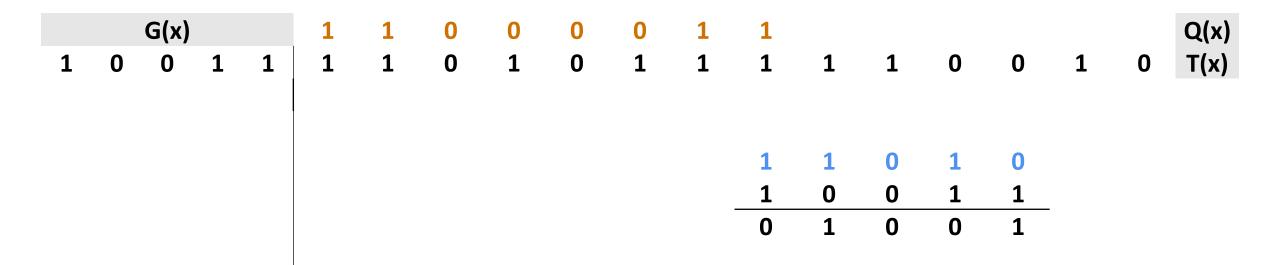
08/10/2023

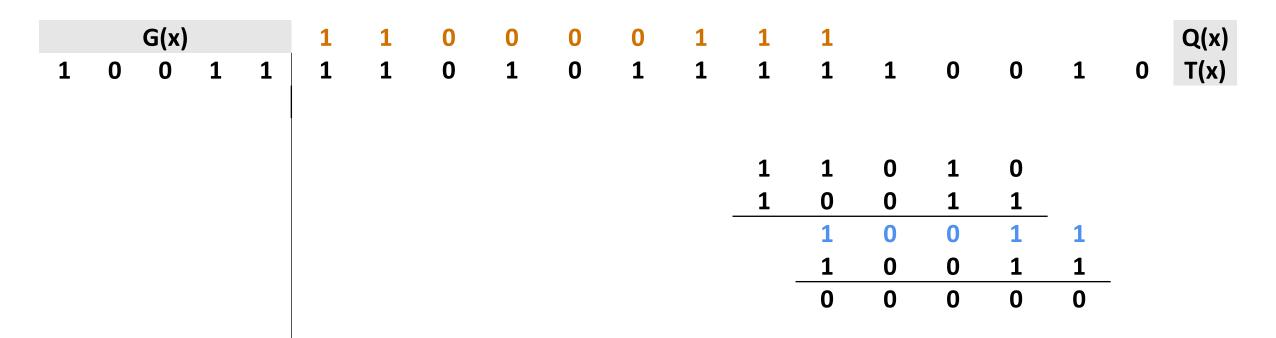


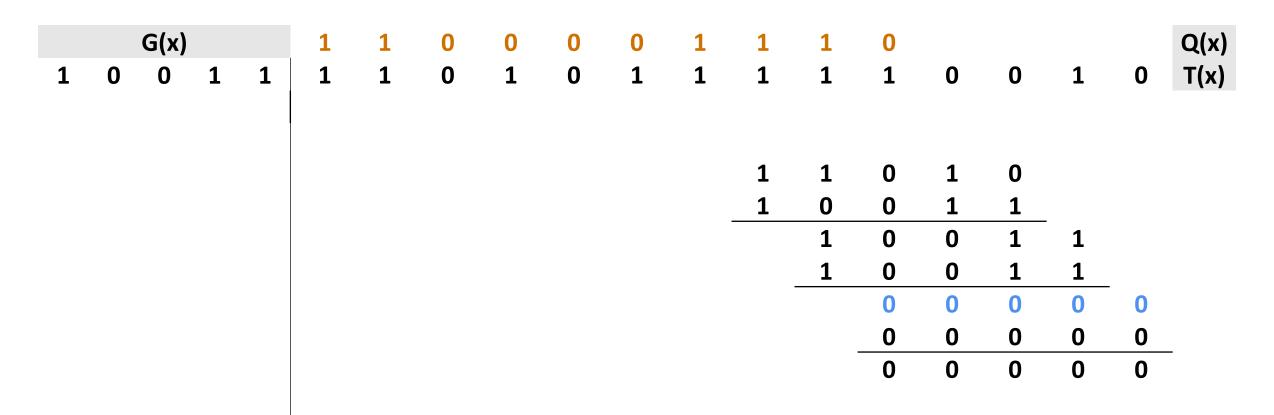




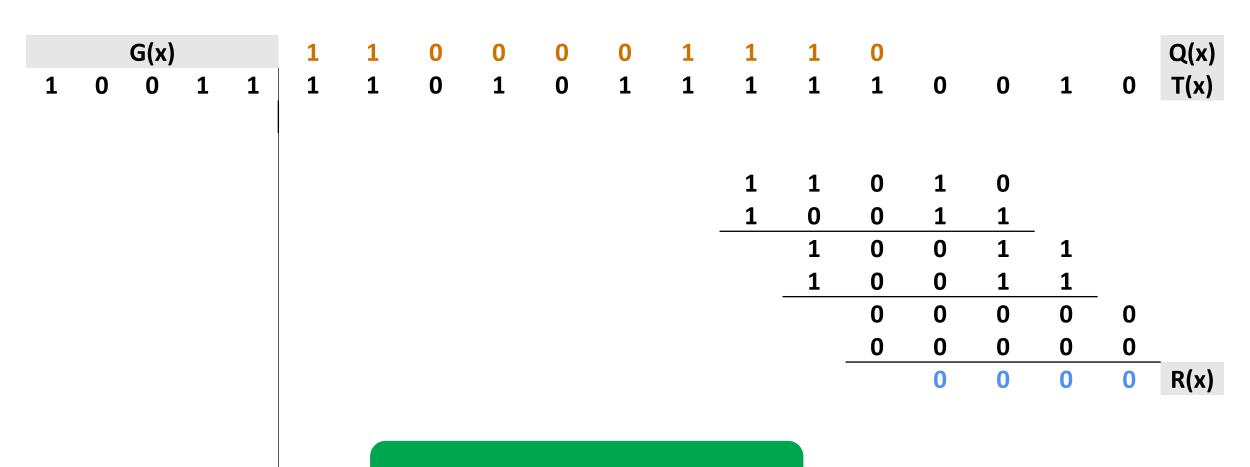




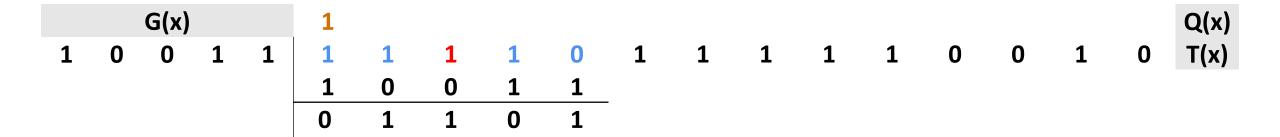




37

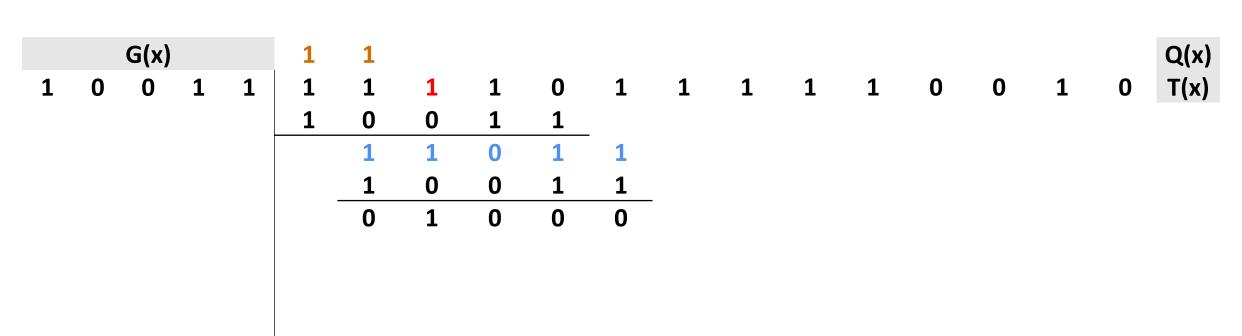


R(x): 0 -> ACCEPT

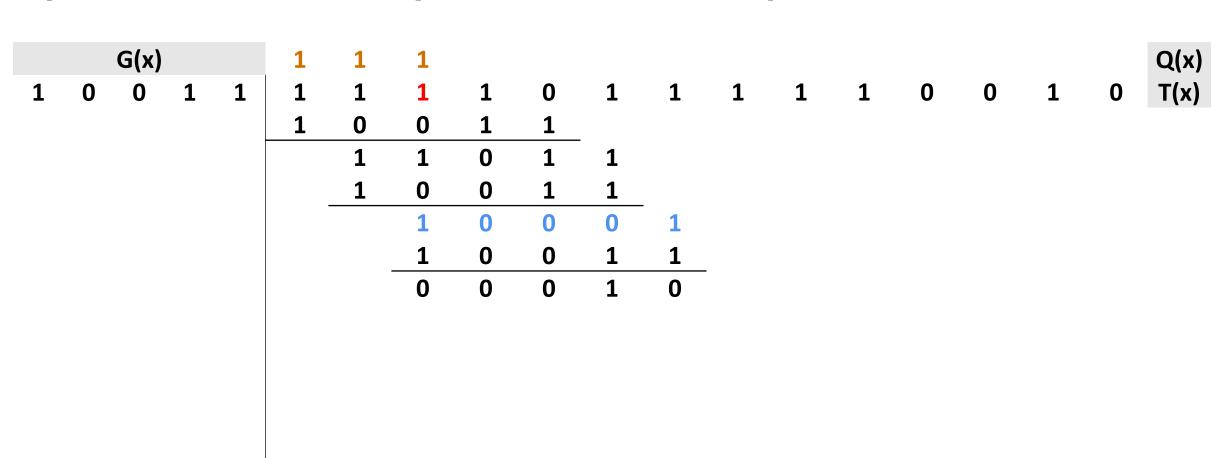


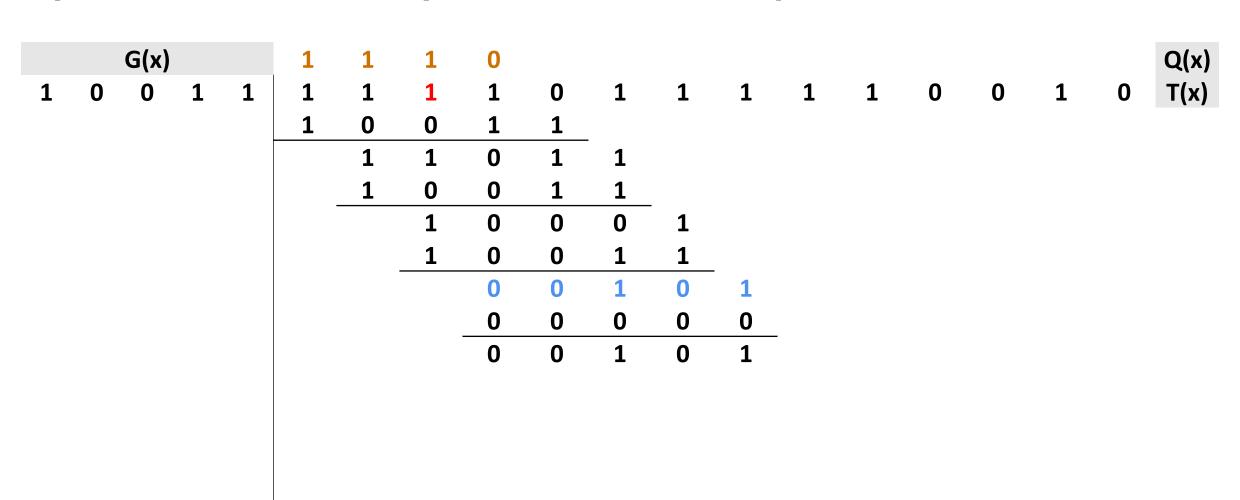
08/10/2023 Cyclic Redundancy Check

39

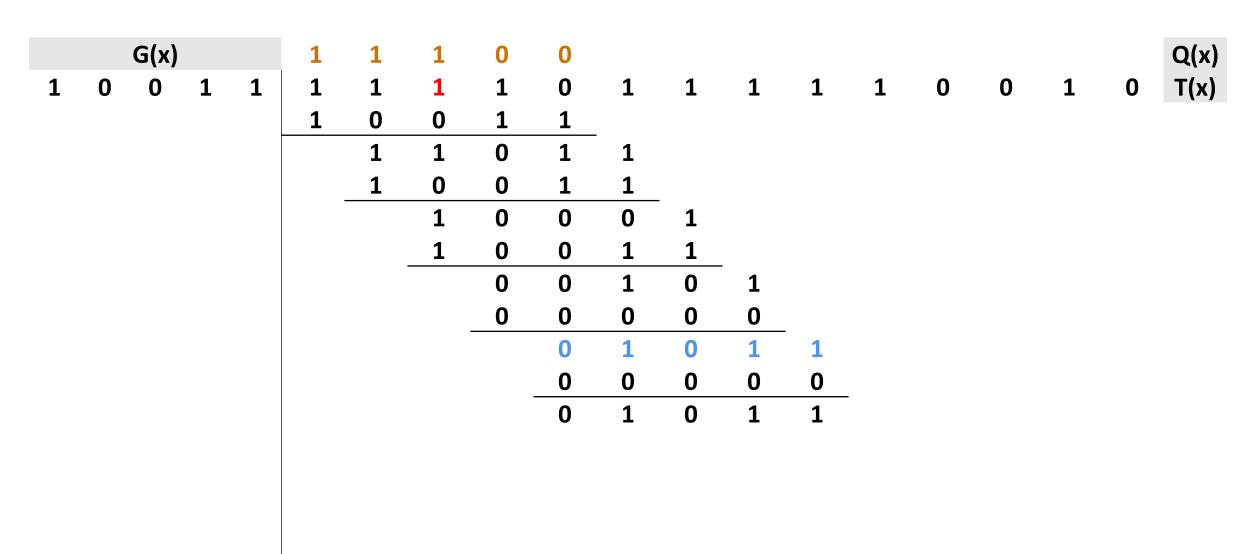


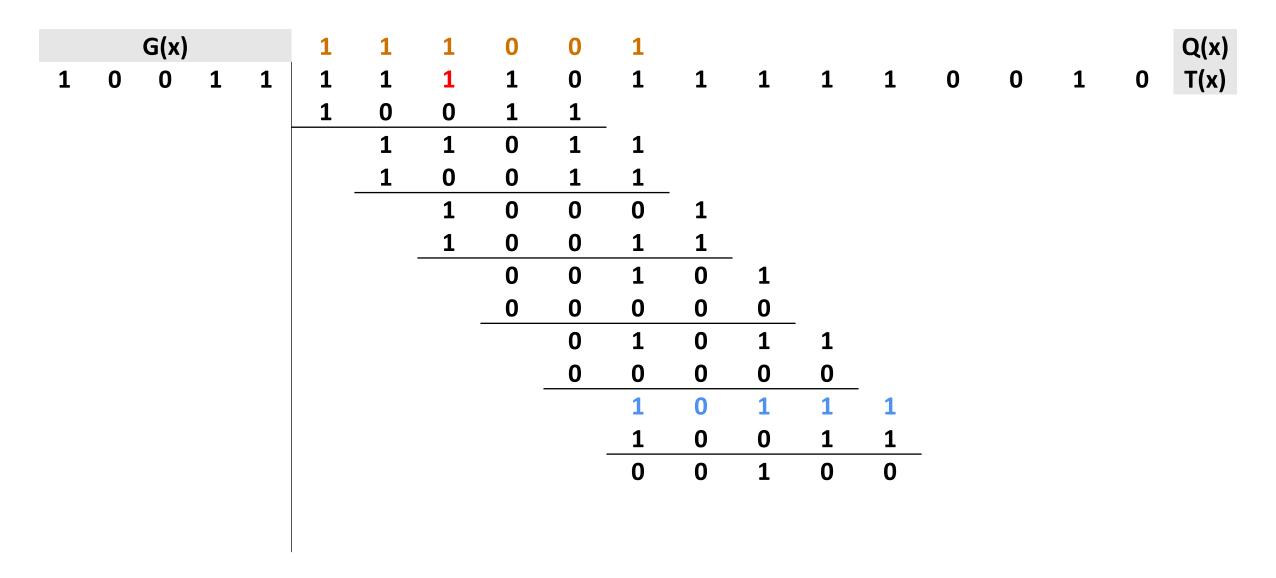
08/10/2023 Cyclic Redundancy Check 40



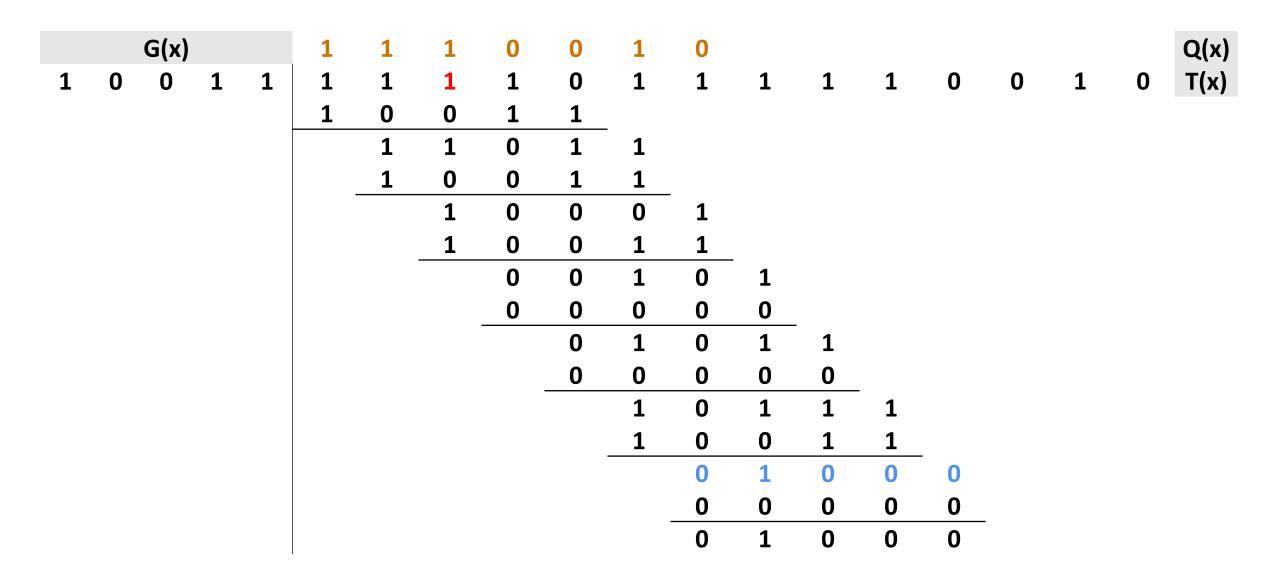


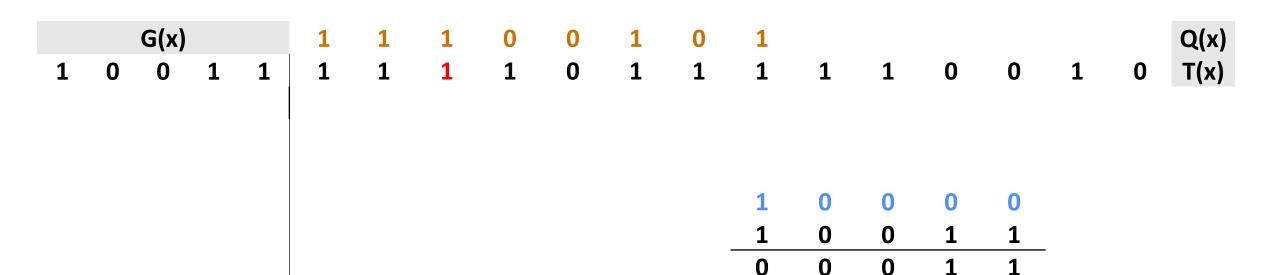
08/10/2023 Cyclic Redundancy Check 42

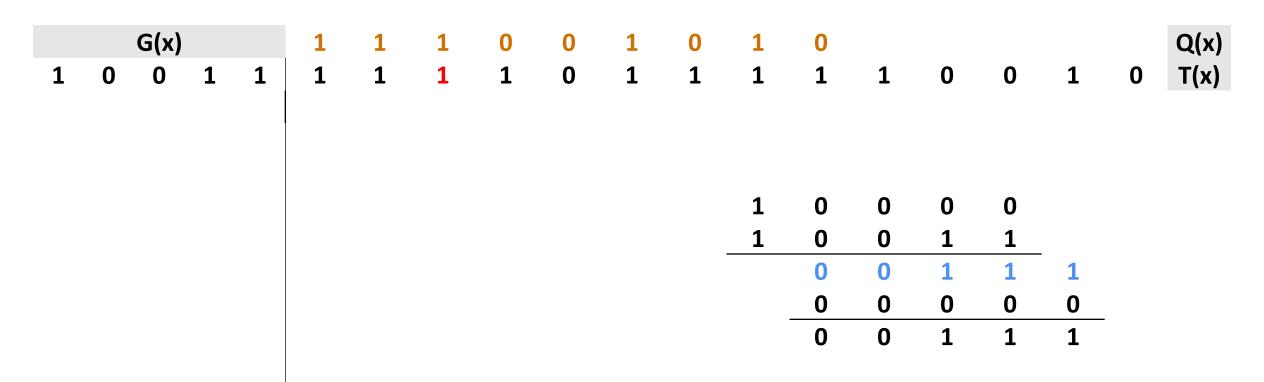


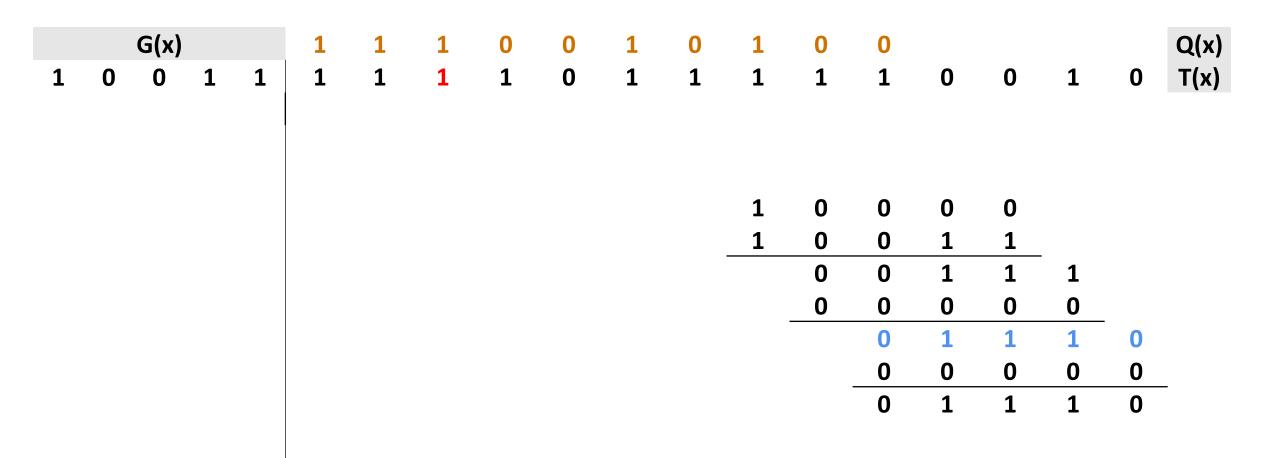


08/10/2023 Cyclic Redundancy Check 44

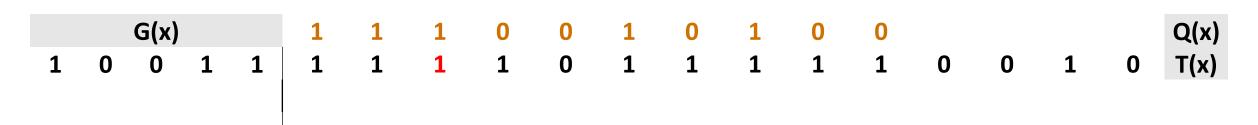








48



R(x): 1110 -> REJECT

#### Considerazioni

- Replicare l'esempio precedente con un foglio di calcolo
- Il polinomio generatore viene scelto in base al protocollo. Non viene scambiato prima di trasmettere il frame
- I polinomi CRC non sono scelti a caso
- Un polinomio scelto bene permette di rilevare molte più situazioni di errore rispetto ad un semplice checksum
- Ethernet 33-bit CRC (32-degree):

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

#### Da provare a casa

• CRC calculator